



Guía del usuario de

AWS IoT SiteWise



AWS IoT SiteWise: Guía del usuario de

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Las marcas comerciales y la imagen comercial de Amazon no se pueden utilizar en relación con ningún producto o servicio que no sea de Amazon, de ninguna manera que pueda causar confusión entre los clientes y que menosprecie o desacredite a Amazon. Todas las demás marcas registradas que no son propiedad de Amazon son propiedad de sus respectivos propietarios, que pueden o no estar afiliados, conectados o patrocinados por Amazon.

Table of Contents

¿Qué es AWS IoT SiteWise?	1
¿Cómo AWS IoT SiteWise funciona	1
Incorporación de datos industriales	2
Modelado de activos para contextualizar los datos recopilados	3
Analice mediante consultas, alarmas y predicciones	4
Visualización de operaciones	4
Almacenamiento de datos	4
Integración con otros servicios de	5
Casos de uso para AWS IoT SiteWise	5
Fabricación	5
Alimentos y bebidas	6
Energía y servicios públicos	6
Trabajando con AWS SDKs	6
Conceptos	7
Introducción	15
Requisitos	15
Configura una AWS cuenta	16
Inscríbese en una Cuenta de AWS	16
Creación de un usuario con acceso administrativo	16
Uso de la demostración de inicio rápido	19
Cree la AWS IoT SiteWise demostración	19
Elimine la AWS IoT SiteWise demostración	21
Tutoriales	23
Cálculo de la OEE	23
Requisitos previos	23
Cómo calcular la OEE	24
Ingesta de datos	26
Requisitos previos	28
Paso 1: Crear una AWS IoT política	28
Paso 2: Crea cualquier AWS IoT cosa	32
Paso 3: creación de un modelo de activos de dispositivo	36
Paso 4: creación de un modelo de activos de flota de dispositivos	38
Paso 5: creación y configuración de un activo de dispositivos	39
Paso 6: creación y configuración de un activo de flota de dispositivos	40

Paso 7: Crea una regla en AWS IoT Core para enviar datos a los activos del dispositivo	41
Paso 8: ejecución del script del cliente de dispositivo	44
Paso 9: limpieza de los recursos después del tutorial	52
Integre los datos con Edge SiteWise	54
Requisitos previos	55
Paso 1: Crear una AWS IoT política	56
Paso 2: Crea y configura cualquier cosa AWS IoT	57
Paso 3: Configure su puerta de enlace V3 habilitada para SiteWise Edge MQTT	58
Paso 4: Instale el software SiteWise Edge Gateway	60
Paso 5: Configure el broker EMQX para que se conecte a aplicaciones externas	61
Paso 6: Publica los datos con Mosquitto	64
Paso 7: Especifique los destinos	66
Paso 8: Especificar los filtros de ruta	68
Paso 9: Configura tus recursos AWS de IoT	72
Paso 10: visualice sus datos	72
Paso 11: Limpiar los recursos después del tutorial	74
Recursos adicionales	77
Visualice y comparta datos en Grafana	77
Requisitos previos	78
Paso 1: Configura tu espacio de trabajo de Grafana gestionado por Amazon	79
Paso 2: AWS IoT SiteWise Añádalo como fuente de datos	81
Paso 3: Crea un panel para explorar y visualizar tus datos	82
(opcional) Paso 4: Configurar alertas para supervisar el rendimiento	85
Paso 5: Limpiar los recursos después del tutorial	86
Recursos adicionales	77
Visualice y comparta datos en SiteWise Monitor	87
Requisitos previos	88
Paso 1: Crear un portal en Monitor SiteWise	89
Paso 2: inicio de sesión en un portal	93
Paso 3: creación de un proyecto de parque eólico	95
Paso 4: creación de un panel para visualizar datos de parques eólicos	99
Paso 5: exploración del portal	106
Paso 6: limpieza de los recursos después del tutorial	107
Publicar en Amazon DynamoDB	110
Requisitos previos	111

Paso 1: Configure AWS IoT SiteWise para publicar las actualizaciones del valor de la propiedad	112
Paso 2: Crea una regla en Core AWS IoT	113
Paso 3: Configurar la acción de la regla de DynamoDB	114
Paso 4: Explore los datos en DynamoDB	115
Eliminar recursos	116
Recursos adicionales	117
Ingiera datos para AWS IoT SiteWise	118
Administre los flujos de datos	119
Configuración de permisos y ajustes	120
Asocie un flujo de datos a una propiedad de un activo	122
Desasociar un flujo de datos de una propiedad de un activo	123
Eliminación de un flujo de datos	124
Actualizar un alias de propiedad de un activo	125
Escenarios habituales	126
Ingiera datos con AWS IoT SiteWise APIs	128
BatchPutAssetPropertyValue API	128
CreateBulkImportJob API	131
Usa reglas AWS IoT Core	139
Concesión del acceso requerido	140
Configuración de la acción de regla	141
Reducción de costos con Basic Ingest	150
Usa acciones AWS IoT Events	151
Usa el administrador de AWS IoT Greengrass transmisiones	152
Utilice las puertas de SiteWise enlace Edge	153
Conceptos clave de Gateway	153
Ventajas de implementar SiteWise Edge	154
Aloje usted mismo una puerta de enlace	155
Requisitos	156
Creación de una puerta de enlace	161
Instale el software de puerta de enlace	164
Puertas de enlace V3 habilitadas para MQTT	167
Transmisiones clásicas, pasarelas V2	236
Agregue fuentes de datos	254
Componentes para SiteWise Edge	301
Filtrado de activos de	303

Soporte de proxy y almacenes de confianza	304
Utilice APIs	311
Alojamiento de una puerta de enlace en Siemens Industrial Edge	329
Seguridad	330
Siemens Secure Storagey la aplicación AWS IoT SiteWise Edge	331
Destinos para Siemens Industrial Edge dispositivos	331
Migración desde la aplicación de vista previa	333
Resolución de problemas	334
AWS IoT SiteWise Registro de cambios de la aplicación Edge	334
Requisitos	336
Creación de una puerta de enlace	337
Creación de un Siemens Databus user	338
Acceda a la aplicación	339
Instalación de la aplicación	340
Actualización de la configuración de una aplicación instalada	342
Datos generados por el uso de este servicio	343
Filtros de destinos y rutas	345
Comprenda los destinos	345
Entendimiento de filtros de ruta	349
Agregue un destino en tiempo real	352
Añadir un destino almacenado en búfer mediante Amazon S3	361
Agregación de filtros de ruta	368
Gestione los destinos	371
Administración de puertas de enlace	377
Administre su puerta de enlace SiteWise Edge con la AWS IoT SiteWise consola	378
Administre las puertas de enlace SiteWise Edge mediante AWS OpsHub AWS IoT SiteWise	379
Acceda a su puerta de enlace SiteWise Edge con las credenciales del sistema operativo local	381
Administre el certificado de la puerta de enlace SiteWise Edge	383
Cambie la versión de los paquetes de componentes de SiteWise Edge Gateway	384
Enumere las pasarelas de SiteWise Edge	385
Describa una puerta de enlace Edge SiteWise	385
Cree una puerta de enlace Edge SiteWise	386
Actualice una puerta de enlace Edge SiteWise	387
Actualice la configuración de la capacidad del gateway	388

Etiquete los recursos de gateway	389
Enumere las etiquetas de una puerta de enlace	390
Eliminar etiquetas de una puerta de enlace	391
Actualice la versión de un componente AWS IoT SiteWise	392
Eliminar una puerta de enlace SiteWise Edge	393
Copia de seguridad y restauración de puertas de enlace	395
Copias de seguridad diarias de los datos de métricas	395
Restaura una puerta de enlace SiteWise Edge	396
Restaura AWS IoT SiteWise los datos	396
Validación de copias de seguridad y restauraciones correctas	398
Puertas de enlace antiguas (AWS IoT Greengrass Version 1)	398
Modelado de activos industriales	400
Descripción general de los activos	400
Los alias de propiedad identifican los flujos de datos de los equipos	400
Las jerarquías de activos representan las relaciones entre los equipos	400
Los modelos de activos estandarizan la representación de los equipos	401
Opciones de modelado para equipos industriales	401
Crear y administrar activos	401
Gestión de modelos de activos complejos	402
Estados de activos y modelos	402
Comprobación del estado de un activo	403
Comprobación del estado de un modelo de activos o de componentes	405
Versiones de modelos de activos	407
Recupere la versión activa de un modelo de activos o un modelo de componentes (con consola)	408
Recupere la versión activa de un modelo de activos o un modelo de componentes (AWS CLI)	409
Modelos compuestos personalizados (componentes)	410
Modelos compuestos personalizados en línea	410
Component-model-based modelos compuestos personalizados	412
Uso de rutas para hacer referencia a propiedades de modelos compuestos personalizados	414
Interfaces de modelos de activos	416
Caso de uso de estandarización de modelos de activos	416
Estructura y componentes	418
Consideraciones	419

Comprenda la relación entre la interfaz y el modelo de activos	419
Cree una interfaz	424
Aplicar una interfaz a un modelo de activos	426
Administre las interfaces	428
Ejemplos de interfaz adicionales	431
Configurar el objeto IDs	435
Trabaja con un objeto UUIDs	435
Usa un externo IDs	436
Creación de modelos de	438
Cree modelos de activos en AWS IoT SiteWise	439
Creación de modelos de componentes	454
Definición de propiedades de datos	458
Creación de modelos compuestos personalizados (componentes)	542
Creación de activos	547
Creación de un activo (consola)	547
Creación de un activo (AWS CLI)	548
Configuración de un nuevo activo	550
Elija activos de	550
Requisitos previos	550
Búsqueda avanzada en Consola de AWS IoT SiteWise	550
Actualización de valores de atributos	554
Asociación y desasociación de activos	557
Asociación y desasociación de activos (consola)	557
Asociación y desasociación de activos (AWS CLI)	559
Actualización de activos y modelos	560
Actualice los activos en AWS IoT SiteWise	561
Actualice los modelos de activos, los modelos de componentes y las interfaces	562
Actualización de modelos compuestos personalizados (componentes)	568
Bloqueo positivo de las escrituras del modelo de activos	571
Elimine activos y modelos en AWS IoT SiteWise	575
Eliminación de activos	576
Elimine modelos e interfaces	578
Operaciones masivas con activos y modelos	580
Conceptos y terminología clave	581
Funcionalidades compatibles	582
Requisitos previos para operaciones masivas	583

Ejecute un trabajo de importación masiva	586
Ejecución de un trabajo de exportación masiva	588
Seguimiento del progreso de los trabajos y gestión de errores	591
Ejemplos de metadatos de importación	597
Ejemplos de metadatos de exportación	613
Esquema del trabajo de transferencia de metadatos	615
Supervisión de datos con alarmas	635
Tipos de Alarmas	635
Estados de alarma	637
Propiedades del estado de alarma	638
Definición de alarmas en modelos de activos	641
Requisitos para las notificaciones de alarma	644
Defina AWS IoT Events las alarmas	645
Definición de alarmas externas	680
Configuración de alarmas en activos	682
Configuración de un valor de umbral (consola)	683
Configuración de un valor de umbral (AWS CLI)	683
Configuración de ajustes de notificaciones	685
Respuesta a las alarmas	688
Respuesta a una alarma (consola)	689
Respuesta a una alarma (API)	692
Ingesta de un estado de alarma externa	692
Asignación de flujos de estados de alarmas externas	693
Ingesta de datos de estados de alarma	695
AWS IoT SiteWise Asistente	697
Configure el AWS IoT SiteWise asistente	697
Creación de un conjunto de datos	699
Editar un conjunto de datos	704
Eliminar un conjunto de datos.	706
AWS IoT SiteWise Preguntas de asistente	707
Supervise los datos con AWS IoT SiteWise Monitor	708
SiteWise Supervise las funciones	709
Federación de SAML	711
SiteWise Conceptos de monitoreo	712
Comience con AWS IoT SiteWise Monitor (Classic)	714
Creación de un portal	715

Configure su portal	717
Invitación a administradores	721
Agregar usuarios al portal	724
Creación de paneles (CLI)	728
Habilitación de alarmas para sus portales	734
Habilitación del portal en la periferia	738
Administración de los portales	738
Comience con AWS IoT SiteWise Monitor (compatible con la IA)	750
Creación de un portal	751
Configure su portal	752
Administración de los portales	755
Eliminación de un portal	760
Cree cuadros de mando con AWS CLI	761
Acceso al portal	766
Crear un proyecto	766
Actualiza un proyecto.	767
Eliminación de un proyecto	768
Creación de panel	769
Actualización de un panel	771
Eliminación de un panel	772
Configurar el panel	772
Consulta datos de AWS IoT SiteWise	797
Consulta de valores de activos actuales	798
Consulta del valor actual de la propiedad de un activo (consola)	798
Consulta del valor actual de la propiedad de un activo (AWS CLI)	799
Consulta de valores históricos de propiedades de activos	800
Consulta de agregados de propiedades de activos	802
Agregaciones para la propiedad de un activo (API)	803
Agregaciones para la propiedad de un activo (AWS CLI)	804
AWS IoT SiteWise idioma de consulta	805
Referencia del lenguaje de consulta	806
Optimización de las consultas	832
Filtros de metadatos	832
Filtros de datos sin procesar	833
ÚNASE a la optimización	834
Consultas grandes	836

ODBC	836
Sintaxis de la cadena de conexión	837
Ejemplos de cadenas de conexión	842
Resolución de problemas	843
Interacción con otros servicios	844
Descripción de propiedades de activos en temas de MQTT	845
Uso de notificaciones	845
Activación de notificaciones sobre propiedades de activos (consola)	846
Activación de notificaciones sobre propiedades de activos (AWS CLI)	846
Consulta de notificaciones	848
Exportación de datos a Amazon S3	851
Integrar Grafana	852
Integre con AWS IoT TwinMaker	853
Habilitación de la integración	854
Integrar y AWS IoT SiteWise AWS IoT TwinMaker	855
Detección de anomalías de equipos	856
Adición de una definición de predicción (consola)	857
Entrenamiento de una predicción (consola)	860
Inicie o detenga la inferencia sobre una predicción (consola)	862
Adición de una definición de predicción (CLI)	863
Entrenamiento de una predicción e inicio de la inferencia (CLI)	866
Entrenamiento de una predicción (CLI)	867
Inicie o detenga la inferencia sobre una predicción (CLI)	869
Detección nativa de anomalías	872
Funciones nativas de detección de anomalías	873
Requisitos previos	874
Configuración del AWS CLI modelo de cálculo APIs	874
Requisitos de propiedad	874
Requisitos previos de etiquetado	875
Requisitos previos para la evaluación del modelo	875
Requisitos previos de las métricas del modelo	876
Habilite la detección de anomalías en los sensores de un activo	877
Crear un modelo de cálculo (AWS CLI)	877
ExecuteAction Preparación de la carga útil de la API	878
Entrenar el modelo (AWS CLI)	879
Iniciar y detener el reentrenamiento del modelo (AWS CLI)	882

Iniciar y detener la inferencia ()AWS CLI	885
Encuentre enlaces de datos	888
Habilite la detección de anomalías en los sensores de todos los activos	889
Crear un modelo de cálculo ()AWS CLI	889
ExecuteAction Preparación de la carga útil de la API	891
Entrenar el modelo (AWS CLI)	892
Iniciar y detener el reentrenamiento del modelo ()AWS CLI	894
Iniciar y detener la inferencia ()AWS CLI	895
Configuraciones de entrenamiento avanzadas	898
Configuración de frecuencia de muestreo	898
Etiquete sus datos	900
Evaluar el modelo	902
Genere métricas de modelos	903
Configuraciones de inferencia avanzadas	908
Inferencia de alta frecuencia (de 5 minutos a 1 hora)	908
Inferencia de baja frecuencia (2 horas — 1 día)	909
Programación flexible	910
Activación de la versión del modelo	912
Comprobación de las versiones del modelo	913
Revise los resultados de la inferencia	914
Recupere los resultados de la inferencia	914
Comprenda los resultados de las inferencias	915
Activa acciones personalizadas en caso de comportamiento anómalo ()Consola de administración de AWS	916
Prácticas recomendadas	917
Comprenda el intervalo de fechas mínimo	917
Toma de muestras para obtener datos de alta frecuencia y garantizar la coherencia entre el entrenamiento y la inferencia	917
Recomendaciones de etiquetado	918
Administración del almacenamiento de datos	919
Configuración de ajustes de almacenamiento	920
Impacto en la retención de datos	921
Configuración del nivel de acceso semiactivo (consola)	921
Configuración del nivel de acceso frecuente (AWS CLI)	923
Configuración para el nivel de acceso inactivo (consola)	926
Configuración para el nivel de acceso inactivo (AWS CLI)	929

Solución de problemas de configuración de almacenamiento	935
Error: el bucket no existe	935
Error: acceso denegado a la ruta de Amazon S3	935
Error: no se puede asumir el ARN del rol	936
Error: no se ha podido acceder al bucket de Amazon S3 entre regiones	936
Rutas de archivos y esquemas de datos guardados en el nivel inactivo	936
Datos del equipo (mediciones)	937
Métricas, transformaciones y agregados	941
Metadatos de los activos	947
Metadatos de jerarquía de los activos	951
Almacenamiento de archivos índice de datos	953
Ejemplos de código	954
Conceptos básicos	955
Hola AWS IoT SiteWise	955
Conceptos básicos	959
Acciones	1019
Seguridad	1087
Protección de datos	1088
Privacidad del tráfico entre redes	1089
AWS IoT SiteWise Asistente de mejora del servicio empresarial	1089
Cifrado de datos	1089
Cifrado en reposo	1090
Cifrado en tránsito	1093
Administración de claves	1095
Identity and Access Management	1097
Público	1097
Autenticarse con identidades	1097
¿Cómo AWS IoT SiteWise funciona con IAM	1099
Políticas administradas	1117
Roles vinculados a servicios	1125
Configuración de permisos para alarmas	1147
La prevención policial confusa entre servicios en AWS IoT SiteWise	1153
Solución de problemas de identidades y accesos en	1155
Validación de conformidad	1157
Resiliencia	1158
Seguridad de la infraestructura	1159

Configuración y análisis de vulnerabilidades	1159
Puntos de conexión de VPC	1160
Operaciones de la API de admitidas	1161
Creación de un punto de conexión de la VPC de tipo interfaz	1163
Acceso a AWS IoT SiteWise través de un punto final de VPC de interfaz	1164
Creación de una política de puntos de conexión de VPC	1165
Prácticas recomendadas de seguridad	1166
Usar credenciales de autenticación en los servidores OPC UA	1167
Usar modos de comunicación cifrados para los servidores OPC UA	1167
Mantener los componentes actualizados	1167
Cifre el sistema de archivos de su puerta de enlace SiteWise Edge	1167
Acceso seguro a la configuración de su periferia	1168
Protección de los datos en Siemens Industrial Edge Management	1168
Otorgue a los usuarios de SiteWise Monitor los permisos mínimos posibles	1168
No exponer información confidencial	1169
Siga las prácticas recomendadas de AWS IoT Greengrass seguridad	1169
Véase también	1169
Registro y supervisión	1170
Supervisión de registros de servicio	1171
Administra el inicio de sesión AWS IoT SiteWise	1172
Ejemplo: entradas de archivos de AWS IoT SiteWise registro	1174
Supervise los registros de SiteWise Edge Gateway	1174
Usa Amazon CloudWatch Logs	1175
Uso de los registros de servicio	1176
Supervisa con las CloudWatch métricas de Amazon	1178
AWS IoT Greengrass Version 2 métricas de puerta de enlace	1179
Registra las llamadas a la API con AWS CloudTrail	1189
AWS IoT SiteWise información en CloudTrail	1189
AWS IoT SiteWise eventos de datos en CloudTrail	1190
AWS IoT SiteWise eventos de gestión en CloudTrail	1193
Ejemplo: entradas de archivos de AWS IoT SiteWise registro	1193
Etiquetar los recursos	1195
Utilice etiquetas en AWS IoT SiteWise	1195
Etiquete con la Consola de administración de AWS	1195
Etiquete con la API AWS IoT SiteWise	1196
Uso de etiquetas con políticas de IAM	1197

Resolución de problemas	1199
Solución de problemas de una puerta de enlace	1199
Configure y acceda a los registros de la puerta de SiteWise enlace Edge	1199
Solución de problemas con la puerta de enlace SiteWise Edge	1200
Solución de problemas de la aplicación AWS IoT SiteWise Edge en Siemens Industrial Edge	1206
Solución de problemas de integraciones de código abierto en Edge	1208
Solución de problemas AWS IoT Greengrass	1210
Solución de problemas de un portal	1210
Los usuarios y los administradores no pueden acceder al portal AWS IoT SiteWise	1210
Solucionar problemas con una acción de AWS IoT SiteWise regla	1212
Configure AWS IoT Core los registros	1213
Configuración de una acción de error de volver a publicar	1213
Solución de problemas de reglas	1215
Solución de problemas de una regla (AWS IoT SiteWise)	1218
Solución de problemas de una regla (DynamoDB)	1219
Solución de importación y exportación masivas	1223
Cuotas y puntos de conexión	1225
Puntos de conexión	1225
Cuotas	1225
Cuotas de AWS IoT SiteWise activos y modelos de activos	1225
Cuotas de interfaz	1232
Cuotas de datos AWS IoT SiteWise de propiedades de activos	1233
Cuotas para las pasarelas SiteWise Edge	1240
Cuotas para AWS IoT SiteWise Monitor	1241
Cuotas de importación y exportación AWS IoT SiteWise masivas de metadatos	1243
Cuotas para la importación AWS IoT SiteWise masiva de datos	1244
AWS IoT SiteWise Límites de regulación de la API auxiliar	1246
Cuotas de cuotas de detección de anomalías	1247
Cuotas para SQL	1248
Historial de revisión	1250
.....	mccclxxv

¿Qué es AWS IoT SiteWise?

AWS IoT SiteWise es un servicio gestionado con el que puede recopilar, almacenar, organizar y supervisar los datos de los equipos industriales a escala para ayudarle a tomar mejores decisiones basadas en los datos. Se puede utilizar AWS IoT SiteWise para supervisar las operaciones en todas las instalaciones, calcular rápidamente las métricas de rendimiento industrial más comunes y crear aplicaciones que analicen los datos de los equipos industriales para evitar problemas costosos con los equipos y reducir las brechas en la producción.

Con AWS IoT SiteWise Monitor ello, sus usuarios operativos pueden crear aplicaciones web para ver y analizar sus datos industriales en tiempo real. Puede obtener información sobre sus operaciones industriales configurando y monitoreando métricas como el tiempo medio entre fallas y la efectividad general del equipo (OEE).

AWS IoT SiteWise Edge es un componente AWS IoT SiteWise que permite la recopilación, el almacenamiento y el procesamiento de datos en dispositivos locales. Resulta útil si tiene acceso limitado a Internet o si necesita mantener la privacidad de los datos.

Temas

- [¿Cómo AWS IoT SiteWise funciona](#)
- [Casos de uso para AWS IoT SiteWise](#)
- [Uso de este servicio con un AWS SDK](#)
- [AWS IoT SiteWise conceptos](#)

¿Cómo AWS IoT SiteWise funciona

AWS IoT SiteWise ofrece un marco de modelado de recursos que puede utilizar para crear representaciones de sus dispositivos, procesos e instalaciones industriales. Las representaciones de sus equipos y procesos se denominan modelos de activos en este documento AWS IoT SiteWise. Con los modelos de activos, puede definir qué datos sin formato consumir y cómo procesarlos en métricas complejas. Cree y visualice activos y modelos para su operación industrial en la [consola de AWS IoT SiteWise](#). También puede configurar modelos de activos para recopilar y procesar datos en la periferia o en la AWS nube.

Temas

- [Incorporación de datos industriales](#)
- [Modelado de activos para contextualizar los datos recopilados](#)
- [Analice mediante consultas, alarmas y predicciones](#)
- [Visualización de operaciones](#)
- [Almacenamiento de datos](#)
- [Integración con otros servicios de](#)

Incorporación de datos industriales

Comience a usarlo AWS IoT SiteWise ingiriendo datos industriales. La ingesta de datos se realiza de varias maneras:

- Ingesta directa desde servidores in situ: utilice protocolos como OPC UA para leer datos directamente desde dispositivos in situ. Implemente el software de puerta de enlace SiteWise Edge, compatible con AWS IoT Greengrass V2, en una amplia gama de plataformas, como las puertas de enlace industriales comunes o los servidores virtuales. Puede conectar hasta 100 servidores OPC UA a una sola AWS IoT SiteWise puerta de enlace. Para obtener más información, consulte [AWS IoT SiteWise Requisitos de la puerta de enlace autohospedada de Edge](#).

Tenga en cuenta que protocolos como Modbus TCP y Ethernet/IP (EIP) son compatibles con nuestra asociación Domatoma en el contexto de. AWS IoT Greengrass V2

- Procesamiento de datos perimetrales con paquetes: mejore su pasarela SiteWise perimetral añadiendo paquetes para disponer de funciones perimetrales integrales. Con SiteWise Edge, disponible en AWS IoT Greengrass V2, el procesamiento de datos se ejecuta directamente in situ antes de transmitirlos de forma segura a la AWS nube mediante una AWS IoT Greengrass transmisión. Para obtener más información, consulte [Configure una fuente OPC UA en Edge SiteWise](#).
- Ingesta adaptativa a través de Amazon S3 con operaciones masivas: cuando trabaje con un gran número de activos o modelos de activos, utilice las operaciones masivas para importar y exportar recursos de forma masiva desde buckets de Amazon S3. Para obtener más información, consulte [Operaciones masivas con activos y modelos](#).
- Mensajes MQTT con reglas AWS IoT básicas: en el caso de los dispositivos conectados a AWS IoT Core que envíen mensajes MQTT, utilice el motor de reglas AWS IoT Core para dirigir esos mensajes hacia ellos AWS IoT SiteWise. Si tiene dispositivos conectados a AWS IoT Core que

envían mensajes [MQTT](#), utilice el motor de reglas AWS IoT Core para enrutar esos mensajes.

AWS IoT SiteWise Para obtener más información, consulte [Ingiera datos para AWS IoT SiteWise usar reglas AWS IoT Core](#).

- Ingesta de datos activada por eventos: utilice AWS IoT Events acciones para configurar la SiteWise acción de IoT AWS IoT Events para enviar datos AWS IoT SiteWise cuando se produzcan eventos. Para obtener más información, consulte [Ingiera datos desde AWS IoT SiteWise AWS IoT Events](#).
- AWS IoT SiteWise API: sus aplicaciones en Edge o en la nube pueden enviar datos directamente a. AWS IoT SiteWise Para obtener más información, consulte [Ingiera datos con AWS IoT SiteWise APIs](#).

Modelado de activos para contextualizar los datos recopilados

Después de ingerir datos, puede usarlos para crear representaciones virtuales de sus activos, procesos e instalaciones mediante la creación de modelos de sus operaciones físicas. Un activo, que representa un dispositivo o un proceso, transmite flujos de datos a la AWS nube. Los activos también pueden significar agrupaciones lógicas de dispositivos. Las jerarquías se forman asociando activos a operaciones complejas de reflejo. Estas jerarquías permiten a los activos acceder a datos de activos secundarios asociados. Se crean activos a partir de modelos de activos. Los modelos de activos son estructuras declarativas que normalizan los formatos de activos. Reutilice componentes de activos para organizar y mantener sus modelos. Para obtener más información, consulte [Modelado de activos industriales](#).

Con AWS IoT SiteWise, puede configurar sus activos para transformar los datos entrantes en métricas y transformaciones contextuales.

- Las transformaciones funcionan al recibir datos del equipo.
- Se calculan métricas con los intervalos que usted defina.

Las métricas y las transformaciones se aplican tanto a activos individuales como a varios activos. AWS IoT SiteWise calcula automáticamente los agregados estadísticos más utilizados, como media, suma y recuento, en varios períodos de tiempo relevantes para datos, métricas y transformaciones de sus equipos.

Los activos se pueden sincronizar mediante AWS IoT TwinMaker. Para obtener más información, consulte [Integrar y AWS IoT SiteWise AWS IoT TwinMaker](#).

Analice mediante consultas, alarmas y predicciones

Analice la fecha recopilada AWS IoT SiteWise mediante la ejecución de consultas y la configuración de alarmas. También puede usar Amazon Lookout para detectar automáticamente anomalías en las métricas e identificar sus causas principales.

- Establezca alarmas específicas para alertar a sus colaboradores cuando los equipos o los procesos no funcionen de manera óptima, garantizando una identificación y resolución rápidas de los problemas. Para obtener más información, consulte [Supervise los datos con las alarmas integradas AWS IoT SiteWise](#).
- Utilice las operaciones de la AWS IoT SiteWise API para consultar los valores actuales, históricos y agregados de sus propiedades de activos en intervalos de tiempo específicos. Para obtener más información, consulte [Consulta datos de AWS IoT SiteWise](#).
- Utilice la detección de anomalías con Amazon Lookout for Equipment para identificar y visualizar los cambios en el equipo o en las condiciones de funcionamiento. Con la detección de anomalías, puede determinar medidas de mantenimiento preventivo para sus operaciones. Esta integración permite a los clientes sincronizar datos entre AWS IoT SiteWise y Amazon Lookout for Equipment. Para obtener más información, consulte [Detección de anomalías con Lookout for Equipment](#).

Visualización de operaciones

Configure SiteWise Monitor para crear aplicaciones web para sus empleados operativos. Las aplicaciones web ayudan a los empleados a visualizar sus operaciones. Gestione distintos niveles de acceso para sus empleados mediante IAM Identity Center o IAM. Configure inicios de sesión y permisos únicos para cada empleado a fin de ver subconjuntos específicos de toda una operación industrial. AWS IoT SiteWise proporciona una [guía de aplicación](#) para que estos empleados aprendan a usar SiteWise Monitor.

Para obtener más información sobre la visualización de sus operaciones, consulte [Supervise los datos con AWS IoT SiteWise Monitor](#).

Almacenamiento de datos

Puede integrar el almacenamiento de series temporales con su lago de datos industriales. AWS IoT SiteWise tiene tres niveles de almacenamiento para datos industriales:

- Un nivel de almacenamiento de acceso frecuente que está optimizado para aplicaciones en tiempo real.

- Un nivel de almacenamiento de acceso semiactivo optimizado para cargas de trabajo de análisis.
- Un nivel de almacenamiento de acceso inactivo administrado por el cliente que utiliza Amazon S3 para aplicaciones de datos operativos con alta tolerancia a la latencia.

AWS IoT SiteWise le ayuda a administrar los costos de almacenamiento al mantener los datos recientes en el nivel de almacenamiento activo. A continuación, defina las políticas de retención de datos para transferir los datos históricos a un almacenamiento de acceso semiactivo o inactivo. Para obtener más información, consulte [Gestione el almacenamiento de datos en AWS IoT SiteWise](#).

También puede importar y exportar metadatos de activos. Para obtener más información, consulte [Metadatos de los activos](#).

Integración con otros servicios de

AWS IoT SiteWise se integra con varios AWS servicios para desarrollar una AWS IoT solución completa en la AWS nube. Para obtener más información, consulte [Interactúa con otros AWS servicios](#).

Casos de uso para AWS IoT SiteWise

AWS IoT SiteWise se utiliza en una variedad de industrias para muchas aplicaciones de recopilación y análisis de datos industriales.

Recopile datos coherentes de todas sus fuentes para ayudar a resolver los problemas rápidamente. AWS IoT SiteWise ofrece monitoreo remoto para recopilar los datos directamente in situ o recopilarlos de múltiples fuentes en muchas instalaciones. AWS IoT SiteWise proporciona la flexibilidad necesaria para las soluciones de datos de IoT industrial.

Fabricación

AWS IoT SiteWise puede simplificar el proceso de recopilación y utilización de los datos de sus equipos para identificar y minimizar las ineficiencias, lo que mejora las operaciones industriales. AWS IoT SiteWise le ayuda a recopilar datos de las líneas y equipos de fabricación. Con AWS IoT SiteWise, puede transferir los datos a la AWS nube y crear métricas de rendimiento para sus equipos y procesos específicos. Puede usar las métricas obtenidas para comprender la efectividad general de sus operaciones e identificar oportunidades de innovación y mejora. También puede ver su proceso de fabricación e identificar deficiencias en los equipos y procesos, brechas de producción o defectos en los productos.

Alimentos y bebidas

Las instalaciones del sector de alimentos y bebidas manejan una amplia variedad de procesos, que incluyen moler granos para hacer harina, faenar y empacar carne, y preparar, cocinar y congelar alimentos para microondas. Las plantas de procesamiento de alimentos a menudo abarca muchas ubicaciones, con operadores de equipo e instalación que supervisan los procesos y el equipo desde una ubicación centralizada. Por ejemplo, las unidades de refrigeración evalúan la gestión y la caducidad de los ingredientes. Supervisan la generación de residuos en todas las instalaciones para garantizar la eficiencia operativa. Con él AWS IoT SiteWise, puede agrupar los flujos de datos de los sensores procedentes de varios lugares por línea de producción e instalaciones para que sus ingenieros de procesos puedan comprender mejor las instalaciones y realizar mejoras en ellas.

Energía y servicios públicos

Con él AWS IoT SiteWise, puede resolver los problemas de los equipos de forma más fácil y eficiente. Puede supervisar el rendimiento de los activos de forma remota y en tiempo real. Acceda a datos históricos de los equipos desde cualquier lugar para identificar posibles problemas, enviar activos precisos y prevenir y solucionar los problemas con mayor rapidez.

Uso de este servicio con un AWS SDK

AWS Los kits de desarrollo de software (SDKs) están disponibles para muchos lenguajes de programación populares. Cada SDK proporciona una API, ejemplos de código y documentación que facilitan a los desarrolladores la creación de aplicaciones en su lenguaje preferido.

Documentación de SDK	Ejemplos de código
AWS SDK para C++	AWS SDK para C++ ejemplos de código
AWS CLI	AWS CLI ejemplos de código
AWS SDK para Go	AWS SDK para Go ejemplos de código
AWS SDK para Java	AWS SDK para Java ejemplos de código
AWS SDK para JavaScript	AWS SDK para JavaScript ejemplos de código
AWS SDK para Kotlin	AWS SDK para Kotlin ejemplos de código

Documentación de SDK	Ejemplos de código
AWS SDK para .NET	AWS SDK para .NET ejemplos de código
AWS SDK para PHP	AWS SDK para PHP ejemplos de código
Herramientas de AWS para PowerShell	Herramientas de AWS para PowerShell ejemplos de código
AWS SDK para Python (Boto3)	AWS SDK para Python (Boto3) ejemplos de código
AWS SDK para Ruby	AWS SDK para Ruby ejemplos de código
AWS SDK para Rust	AWS SDK para Rust ejemplos de código
AWS SDK para SAP ABAP	AWS SDK para SAP ABAP ejemplos de código
AWS SDK para Swift	AWS SDK para Swift ejemplos de código

Ejemplo de disponibilidad

¿No encuentra lo que necesita? Solicite un ejemplo de código a través del enlace de Enviar comentarios que se encuentra al final de esta página.

AWS IoT SiteWise conceptos

Los siguientes son los conceptos básicos de AWS IoT SiteWise:

Agregado

Los agregados son métricas o mediciones fundamentales que AWS IoT SiteWise se calculan automáticamente para todos los datos de series temporales. Para obtener más información, consulte [La propiedad del activo de consulta se agrega en AWS IoT SiteWise](#).

activo

Cuando introduce o ingiere datos de su equipo industrial, cada uno AWS IoT SiteWise de sus dispositivos, equipos y procesos se muestra como activos. Cada activo tiene datos asociados.

Por ejemplo, un equipo podría tener un número de serie, una ubicación, una marca y un modelo y una fecha de instalación. También puede tener valores de series temporales para disponibilidad, rendimiento, calidad, temperatura, presión, etc. Agrupe los activos en jerarquías para permitir que los activos accedan a datos almacenados en sus activos secundarios. Para obtener más información, consulte [Modelado de activos industriales](#).

Jerarquía de activos

Configure jerarquías de activos para crear representaciones lógicas de sus operaciones industriales. Para ello, defina una jerarquía en un modelo de activos y asocie los activos creados a partir de ese modelo a la jerarquía especificada. Las métricas de los activos principales pueden combinar datos de las propiedades de los activos secundarios, lo que le permite calcular métricas que ofrecen información sobre sus operaciones generales o sobre una parte específica de ellas. Para obtener más información, consulte [Definición de jerarquías de modelos de activos](#).

Modelo de activos

Cada activo se crea utilizando un modelo de activos. Los modelos de activos son estructuras que definen y normalizan el formato de los activos. Garantizan la coherencia de la información en varios activos del mismo tipo, lo que permite gestionar datos en activos que representan grupos de dispositivos. En cada modelo de activos, puede definir [atributos](#), entradas de serie temporal ([medidas](#)), transformaciones de serie temporal ([transformaciones](#)), agregaciones de serie temporal ([métricas](#)) y [jerarquías de activos](#). Para obtener más información, consulte [Modelado de activos industriales](#).

Seleccione dónde desea que se procesen las propiedades de su modelo de activos configurando su modelo de activos para la periferia. Utilice esta característica para gestionar y supervisar los datos de los activos en los dispositivos locales.

Propiedad de activo

Las propiedades de activo son las estructuras dentro de cada activo que contienen datos industriales. Cada propiedad tiene un tipo de datos y también puede tener una unidad. Una propiedad puede ser un [atributo](#), una [medida](#), una [transformación](#) o una [métrica](#). Para obtener más información, consulte [Definición de propiedades de datos](#).

Configure las propiedades de los activos para que se procesen en la periferia. Para obtener más información sobre el procesamiento de datos en la periferia, consulte [Configure una fuente OPC UA en Edge SiteWise](#).

Atributo

Los atributos son propiedades de un activo que normalmente permanecen constantes, como el fabricante o la ubicación del dispositivo. Los atributos pueden tener valores predefinidos. Cada activo creado a partir de un modelo de activos incluye los valores predeterminados de los atributos definidos en dicho modelo. Para obtener más información, consulte [Definición de datos estáticos \(atributos\)](#).

Modelo de cómputo

A `ComputationModel` es una abstracción para ciertos tipos de cómputo que se puede aplicar a los datos. Define el modelo de un conjunto de cálculos, en el que los usuarios pueden describir la entrada, la salida y la configuración de un motor de cálculo específico. `ComputationModel` es un recurso nuevo con ARN y tiene estado y está versionado. Para obtener más información, consulte [Crear un modelo de cálculo \(AWS CLI\)](#).

Panel de control

Cada proyecto contiene un conjunto de paneles. Los paneles proporcionan un conjunto de visualizaciones para los valores de un conjunto de activos. Los propietarios de proyectos crean los paneles y las visualizaciones que contiene. Cuando un propietario de proyecto está listo para compartir el conjunto de paneles, el propietario puede invitar a observadores al proyecto, lo que les da acceso a todos los paneles del mismo. Si desea un conjunto distinto de observadores para distintos paneles, debe dividir los paneles entre proyectos. Cuando los observadores consultan paneles, pueden personalizar el intervalo de tiempo para buscar datos específicos.

Conjunto de datos

Los conjuntos de datos son conjuntos de datos que representan datos de series temporales, non-time-series datos y datos no relacionados con el equipo, como los horarios de los turnos, los registros de mantenimiento y las bases de datos de los empleados. Son compatibles con datos externos y utilizan AWS IoT SiteWise capacidades analíticas. Incluye las fuentes del conjunto de datos, el esquema del conjunto de datos y los parámetros del conjunto de datos. El AWS IoT SiteWise Asistente usa conjuntos de datos que consumen los índices de Amazon Kendra.

Flujo de datos

Introduzca o ingiera datos industriales AWS IoT SiteWise incluso antes de crear modelos y activos. AWS IoT SiteWise genera automáticamente flujos de datos para recopilar flujos de datos sin procesar de su equipo.

Alias de flujo de datos

Los alias de flujo de datos le ayudan a identificar fácilmente un flujo de datos. Por ejemplo, el alias `server1-windfarm/3/turbine/7/temperature` indica los valores de temperatura procedentes de la turbina #7 del parque eólico #3. El término `server1` es el nombre del origen de datos que ayuda a identificar el servidor OPC UA y `server1-` es un prefijo asociado a todos los flujos de datos notificados desde este servidor OPC UA.

Asociación de flujos de datos

Después de crear modelos de activos y activos, asocie los flujos de datos con las propiedades de los activos definidas en sus activos para estructurar los datos. AWS IoT SiteWise a continuación, puede utilizar modelos de activos y activos para gestionar los datos entrantes de sus flujos de datos. También puede disociar los flujos de datos de las propiedades de activo. Para obtener más información, consulte [Gestione los flujos de datos para AWS IoT SiteWise](#).

Destinos

Los destinos en SiteWise Edge representan los puntos finales a los que desea enviar la telemetría o los datos procesados. SiteWise Edge admite la capa AWS IoT SiteWise activa, la ingesta en búfer o un bucket de Amazon S3 como destinos. Puede configurar los destinos para suscribirse a temas específicos de MQTT mediante filtros de ruta. Para obtener más información, consulte [Comprenda los destinos de AWS IoT SiteWise Edge](#).

Fórmula

Cada propiedad de [transformación](#) y [métrica](#) tiene una fórmula que define de qué forma la propiedad transforma o agrega datos. Las fórmulas constan de entradas de propiedad, operadores y funciones que ofrece AWS IoT SiteWise. Para obtener más información, consulte [Uso de expresiones de fórmula](#).

Interfaz

Una interfaz es un tipo de modelo que define una estructura estándar que se puede aplicar a diferentes modelos de activos. Para obtener más información, consulte [Interfaces del modelo de activos](#).

Medida

Las medidas son propiedades de un activo que representan flujos de datos de serie temporal sin procesar de sensores procedentes de un dispositivo o equipo. Para obtener más información, consulte [Definición de flujos de datos procedentes del equipo \(mediciones\)](#).

Métrica

Las métricas son propiedades que representan datos de serie temporal agregados. Cada métrica va acompañada de una expresión matemática ([fórmula](#)) que describe cómo agregar puntos de datos y un intervalo de tiempo para calcular esa agregación. Las métricas generan un único punto de datos para cada intervalo de tiempo especificado. Para obtener más información, consulte [Agregación de datos desde propiedades y otros activos \(métricas\)](#).

MQTT

El MQTT (Message Queuing Telemetry Transport) es un protocolo de mensajería ligero para sensores y dispositivos.

Paquetes

SiteWise Las pasarelas perimetrales utilizan paquetes para determinar cómo recopilar, procesar y enrutar los datos. Para obtener más información sobre los paquetes disponibles para su puerta de enlace SiteWise Edge, consulte [the section called “Uso de paquetes”](#).

Paquete de recopilación de datos

Utilice el paquete de recopilación de datos para que su puerta de enlace SiteWise Edge pueda recopilar sus datos industriales y enviarlos al AWS destino que elija.

Paquete de procesamiento de datos

Utilice el paquete de procesamiento de datos para procesar, almacenar y recuperar sus datos en la periferia durante un máximo de 30 días. Intercambie datos procesados de forma perimetral hacia y desde aplicaciones locales a través SiteWise de Edge. APIs

OPC UA

OPC UA (Open Platform Communications Unified Architecture) es un protocolo de comunicación para la automatización industrial.

Filtros de ruta

Utilice filtros de ruta dentro de una puerta de enlace para suscribirse a los temas de MQTT y publicarlos en los destinos AWS IoT SiteWise compatibles. Todas las fuentes, los canales de procesamiento de datos y los destinos basados en MQTT intercambian datos mediante temas de MQTT en una pasarela V3 autohospedada y habilitada para MQTT. Puede definir filtros de temas para especificar los datos que desea ingerir o enrutar a diferentes destinos.

Portal

Un AWS IoT SiteWise Monitor portal es una aplicación web que puede utilizar para visualizar y compartir sus AWS IoT SiteWise datos. Un portal tiene uno o varios administradores y contiene cero o más proyectos.

Administrador del portal

Cada portal de SiteWise Monitor tiene uno o más administradores de portal. Los administradores del portal utilizan el portal para crear proyectos que contengan recopilaciones de activos y paneles. A continuación, el administrador del portal asigna activos y propietarios a cada proyecto. Al controlar el acceso al proyecto, los administradores del portal especifican los activos que los propietarios y visores de proyectos pueden ver.

Proyecto

Cada portal de SiteWise Monitor contiene un conjunto de proyectos. Cada proyecto tiene un subconjunto de sus activos de AWS IoT SiteWise asociados al mismo. Los propietarios de proyectos crean uno o varios paneles para proporcionar una forma coherente de ver los datos asociados a esos activos. Los propietarios del proyecto pueden invitar a los lectores al proyecto para permitirles ver los activos y paneles del proyecto. El proyecto es la unidad básica para compartir en SiteWise Monitor. Los propietarios del proyecto pueden invitar a los usuarios a los que el AWS administrador les dio acceso al portal. Un usuario debe tener acceso a un portal antes de que un proyecto de ese portal pueda compartirse con ese usuario.

Propietario del proyecto

Cada proyecto de SiteWise Monitor tiene propietarios. Los propietarios de proyectos crean visualizaciones en forma de paneles para representar los datos operativos de manera coherente. Cuando los paneles están listos para compartirse, el propietario del proyecto puede invitar a lectores al proyecto. Los propietarios de proyectos también pueden asignar otros propietarios al proyecto. Los propietarios del proyecto pueden configurar los umbrales y los ajustes de notificación de las alarmas.

Observador de proyectos

Cada proyecto de SiteWise Monitor tiene visores. Los observadores de proyectos pueden conectarse al portal para ver los paneles creados por los propietarios de proyectos. En cada panel de control, los observadores de proyectos pueden ajustar el intervalo de tiempo para comprender mejor los datos operativos. Los observadores de proyectos solo pueden ver los paneles de los proyectos a los que tienen acceso. Los observadores de proyectos pueden confirmar y posponer alarmas.

Alias de propiedad

Tiene la opción de crear alias en las propiedades de los activos, como la ruta de flujo de datos de un servidor OPC UA (por ejemplo, /company/windfarm/3/turbine/7/temperature), lo que simplifica la identificación de la propiedad de un activo durante la ingesta o la recuperación de los datos del activo. Cuando utiliza una [puerta de enlace SiteWise Edge](#) para ingerir datos de los servidores, los alias de sus propiedades deben coincidir con las rutas de sus flujos de datos sin procesar. Para obtener más información, consulte [Gestione los flujos de datos para AWS IoT SiteWise](#).

Notificación de propiedad

Al habilitar las notificaciones de propiedad para una propiedad de un activo, AWS IoT SiteWise publica un mensaje MQTT AWS IoT Core cada vez que esa propiedad recibe un nuevo valor. La carga útil del mensaje incluye detalles sobre la actualización de ese valor de propiedad. Utilice las notificaciones del valor de la propiedad para crear soluciones que conecten sus datos industriales AWS IoT SiteWise con otros AWS servicios. Para obtener más información, consulte [Interactúa con otros AWS servicios](#).

SiteWise Puerta de enlace Edge

Se instala una puerta de enlace SiteWise Edge en las instalaciones del cliente para recopilar, gestionar y dirigir los datos. Una puerta de enlace SiteWise Edge se conecta a sus fuentes de datos industriales a través de varios protocolos para recopilar y procesar datos y enviarlos a la AWS nube. SiteWise Las pasarelas perimetrales también se pueden conectar a fuentes de datos asociadas. Para obtener más información, consulte [Utilice puertas de enlace AWS IoT SiteWise Edge](#).

Transformación

Las transformaciones son propiedades de un activo que representan datos de serie temporal transformados. Cada transformación va acompañada de una expresión matemática ([fórmula](#)) que especifica cómo convertir puntos de datos de una forma a otra. Los puntos de datos transformados mantienen una one-to-one relación con los puntos de datos de entrada. Para obtener más información, consulte [Datos de transformación \(transformaciones\)](#).

Visualización

En cada panel de control, los propietarios de proyectos deciden cómo mostrar las propiedades y alarmas de los activos asociados al proyecto. La disponibilidad puede representarse como un gráfico de líneas, mientras que otros valores pueden mostrarse como gráficos de barras o indicadores clave de rendimiento (KPIs). Las alarmas se visualizan mejor como cuadrículas

de estado y líneas temporales de estado. Los propietarios de proyectos personalizan cada visualización para ofrecer la mejor comprensión de los datos de ese activo.

Comience con AWS IoT SiteWise

Con AWS IoT SiteWise, puede recopilar, organizar, analizar y visualizar sus datos.

AWS IoT SiteWise proporciona una demostración que puede utilizar para explorar el servicio sin configurar una fuente de datos real. Para obtener más información, consulte [Usa la AWS IoT SiteWise demo](#).

Puede completar los siguientes tutoriales para explorar determinadas funciones de AWS IoT SiteWise:

- [Transfiere datos AWS IoT SiteWise desde cosas AWS IoT](#)
- [Visualice y comparta datos de parques eólicos en SiteWise Monitor](#)
- [Publicar actualizaciones de valor de propiedad en Amazon DynamoDB](#)

Consulte los siguientes temas para obtener más información sobre AWS IoT SiteWise:

- [Ingiera datos para AWS IoT SiteWise](#)
- [Modelado de activos industriales](#)
- [Configure las capacidades perimetrales en AWS IoT SiteWise Edge](#)
- [Supervise los datos con AWS IoT SiteWise Monitor](#)
- [Consulta datos de AWS IoT SiteWise](#)
- [Interactúa con otros AWS servicios](#)

Temas

- [Requisitos](#)
- [Configura una AWS cuenta](#)

Requisitos

Debe tener una AWS cuenta para empezar AWS IoT SiteWise. Si no la tiene, consulte la sección siguiente para obtener instrucciones sobre cómo configurar una cuenta.

Usa una región donde AWS IoT SiteWise esté disponible. Para obtener más información, consulte [Puntos de conexión y cuotas de AWS IoT SiteWise](#). Puede utilizar el selector de regiones de la Consola de administración de AWS para cambiar a una de estas regiones.

Configura una AWS cuenta

Temas

- [Inscríbese en una Cuenta de AWS](#)
- [Creación de un usuario con acceso administrativo](#)

Inscríbese en una Cuenta de AWS

Si no tiene una Cuenta de AWS, complete los siguientes pasos para crearlo.

Para suscribirse a una Cuenta de AWS

1. Abrir <https://portal.aws.amazon.com/billing/registro>.
2. Siga las instrucciones que se le indiquen.

Parte del procedimiento de registro consiste en recibir una llamada telefónica o mensaje de texto e indicar un código de verificación en el teclado del teléfono.

Cuando te registras en una Cuenta de AWS, Usuario raíz de la cuenta de AWS se crea uno. El usuario raíz tendrá acceso a todos los Servicios de AWS y recursos de esa cuenta. Como práctica recomendada de seguridad, asigne acceso administrativo a un usuario y utilice únicamente el usuario raíz para realizar [tareas que requieren acceso de usuario raíz](#).

AWS te envía un correo electrónico de confirmación una vez finalizado el proceso de registro. En cualquier momento, puede ver la actividad de su cuenta actual y administrarla accediendo a <https://aws.amazon.com/> y seleccionando Mi cuenta.

Creación de un usuario con acceso administrativo

Después de crear un usuario administrativo Cuenta de AWS, asegúrelo Usuario raíz de la cuenta de AWS AWS IAM Identity Center, habilite y cree un usuario administrativo para no usar el usuario root en las tareas diarias.

Proteja su Usuario raíz de la cuenta de AWS

1. Inicie sesión [Consola de administración de AWS](#) como propietario de la cuenta seleccionando el usuario root e introduciendo su dirección de Cuenta de AWS correo electrónico. En la siguiente página, escriba su contraseña.

Para obtener ayuda para iniciar sesión con el usuario raíz, consulte [Iniciar sesión como usuario raíz](#) en la Guía del usuario de AWS Sign-In .

2. Active la autenticación multifactor (MFA) para el usuario raíz.

Para obtener instrucciones, consulte [Habilitar un dispositivo MFA virtual para el usuario Cuenta de AWS raíz \(consola\)](#) en la Guía del usuario de IAM.

Creación de un usuario con acceso administrativo

1. Activar IAM Identity Center.

Consulte las instrucciones en [Activar AWS IAM Identity Center](#) en la Guía del usuario de AWS IAM Identity Center .

2. En IAM Identity Center, conceda acceso administrativo a un usuario.

Para ver un tutorial sobre su uso Directorio de IAM Identity Center como fuente de identidad, consulte [Configurar el acceso de los usuarios con la configuración predeterminada Directorio de IAM Identity Center en la](#) Guía del AWS IAM Identity Center usuario.

Inicio de sesión como usuario con acceso de administrador

- Para iniciar sesión con el usuario de IAM Identity Center, use la URL de inicio de sesión que se envió a la dirección de correo electrónico cuando creó el usuario de IAM Identity Center.

Para obtener ayuda para iniciar sesión con un usuario del Centro de identidades de IAM, consulte [Iniciar sesión en el portal de AWS acceso](#) en la Guía del AWS Sign-In usuario.

Concesión de acceso a usuarios adicionales

1. En IAM Identity Center, cree un conjunto de permisos que siga la práctica recomendada de aplicar permisos de privilegios mínimos.

Para conocer las instrucciones, consulte [Create a permission set](#) en la Guía del usuario de AWS IAM Identity Center .

2. Asigne usuarios a un grupo y, a continuación, asigne el acceso de inicio de sesión único al grupo.

Para conocer las instrucciones, consulte [Add groups](#) en la Guía del usuario de AWS IAM Identity Center .

Usa la AWS IoT SiteWise demo

Puede explorarlo fácilmente AWS IoT SiteWise utilizando la AWS IoT SiteWise demostración. AWS IoT SiteWise incluye la demostración en forma de CloudFormation plantilla que puede utilizar para crear modelos de activos, activos y un portal de SiteWise monitorización, además de generar datos de muestra para un máximo de una semana.

Important

Cuando cree la demostración, se le empezará a cobrar por los recursos que esta demostración cree y consuma.

Temas

- [Cree la AWS IoT SiteWise demostración](#)
- [Elimine la AWS IoT SiteWise demostración](#)

Cree la AWS IoT SiteWise demostración

Puede crear la AWS IoT SiteWise demostración desde la AWS IoT SiteWise consola.


Note

La demostración crea las funciones Lambda, una regla de CloudWatch eventos y las funciones AWS Identity and Access Management (de IAM) necesarias para la demostración. Es posible que vea estos recursos en su cuenta. AWS Recomendamos que conserve estos recursos hasta que termine con la demostración. Si elimina los recursos, la demostración podría dejar de funcionar correctamente.

Para crear la demostración en la AWS IoT SiteWise consola


1. Ve a la [AWS IoT SiteWise consola](#) y busca la SiteWise demostración en la esquina superior derecha de la página.

2. (Opcional) En la sección de SiteWise demostración, cambia el campo Días para conservar los activos de la demostración para especificar cuántos días se conservará la demostración antes de eliminarla.
3. (Opcional) Para crear un portal de SiteWise monitoreo para monitorear los datos de muestra, haga lo siguiente.

 Note

Se le cobrará por los recursos de SiteWise Monitor que cree y consuma esta demostración. Para obtener más información, consulte [SiteWise Monitorear](#) en los AWS IoT SiteWise precios.

- a. Seleccione Recursos de monitorización.
- b. Seleccione Permiso.
- c. Elija un rol de IAM existente que conceda a sus usuarios de IAM federados acceso al portal.

 Important

Su rol de IAM debe tener los siguientes permisos.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iotsitewise:Describe*",
        "iotsitewise:List*",
        "iotsitewise:Get*",
        "cloudformation:DescribeStacks",
        "iam:GetPolicyVersion",
        "iam:GetPolicy",
        "iam:ListAttachedRolePolicies",
        "sso:DescribeRegisteredRegions",
        "organizations:DescribeOrganization"
      ]
    }
  ],
}
```

```
    "Resource": "*"
  }
]
}
```

Para obtener más información sobre cómo trabajar con SiteWise Monitor, consulte [¿Qué es AWS IoT SiteWise Monitor?](#) en la Guía AWS IoT SiteWise Monitor de aplicación.

4. Seleccione Crear demostración.

Se tarda unos 3 minutos en crear la demostración. Si no se puede crear la demostración, es posible que su cuenta no tenga permisos suficientes. Cambie a una cuenta que tenga permisos administrativos o use los siguientes pasos para eliminar la demostración e inténtelo de nuevo:

a. Seleccione Eliminar demostración.

La demostración tarda unos 15 minutos en borrarse.

b. Si la demostración no se elimina, abre la [CloudFormation consola](#), elige la pila llamada IoT Site Wise Demo Assets y selecciona Eliminar en la esquina superior derecha.

c. Si la demostración no se borra de nuevo, sigue los pasos de la CloudFormation consola para omitir los recursos que no se pudieron eliminar e inténtalo de nuevo.

5. Una vez que la demostración se haya creado correctamente, podrá explorar los activos y datos de la demostración en la [consola de AWS IoT SiteWise](#).

Elimine la AWS IoT SiteWise demostración

La AWS IoT SiteWise demo se borra automáticamente al cabo de una semana, o el número de días que hayas elegido si has creado la versión de demostración desde la CloudFormation consola. Puede eliminar la demostración si ha terminado de usar los recursos de demostración. También puede eliminar la demostración si no se puede crear. Siga los pasos siguientes para eliminar la demostración manualmente.

Para eliminar la demostración AWS IoT SiteWise

1. Vaya a la [consola de CloudFormation](#).
2. Seleccione IoT SiteWise Demo Assets de la lista de Stacks (Pilas).
3. Seleccione Eliminar (Delete).

Al eliminar la pila, se eliminan todos los recursos creados para la demostración.

4. En el cuadro de diálogo de confirmación, elija Delete stack (Eliminar pila).

La pila tarda unos 15 minutos en borrarse. Si la demostración no se elimina, vuelva a seleccionar Delete (Eliminar) en la esquina superior derecha. Si la demostración no se puede eliminar de nuevo, siga los pasos de la CloudFormation consola para omitir los recursos que no se pudieron eliminar e inténtelo de nuevo.

AWS IoT SiteWise tutoriales

Bienvenido a la página de AWS IoT SiteWise tutoriales. Esta colección en aumento de tutoriales le brinda los conocimientos y las habilidades necesarios para navegar por las complejidades de AWS IoT SiteWise. Estos tutoriales ofrecen una amplia gama de temas básicos para atender sus necesidades. A medida que profundice en los tutoriales, descubra información inestimable sobre varios aspectos de AWS IoT SiteWise.

Cada tutorial utiliza un ejemplo de equipo específico. Están pensados para entornos de prueba y usan nombres de empresa, activos, propiedades y otros elementos ficticios. Su finalidad es proporcionar orientación general. Los tutoriales no deben utilizarse directamente en un entorno de producción sin antes realizar una revisión y adaptación exhaustivas para satisfacer las necesidades únicas de la organización.

Temas

- [Calcule el OEE en AWS IoT SiteWise](#)
- [Transfiere datos AWS IoT SiteWise desde cosas AWS IoT](#)
- [Integre los datos en SiteWise Edge mediante una puerta de enlace V3 habilitada para MQTT](#)
- [Visualice y comparta datos en Grafana](#)
- [Visualice y comparta datos de parques eólicos en SiteWise Monitor](#)
- [Publicar actualizaciones de valor de propiedad en Amazon DynamoDB](#)

Calcule el OEE en AWS IoT SiteWise

En este tutorial se proporciona un ejemplo de cómo calcular la eficacia general de los equipos (OEE) de un proceso de producción. Como resultado, los cálculos o fórmulas de su OEE podrían diferir de los que se muestran aquí. En general, OEE se define como $\text{Availability} * \text{Quality} * \text{Performance}$. Para obtener más información sobre el cálculo de la OEE, consulte [Eficacia general de los equipos](#) en Wikipedia.

Requisitos previos

Para completar este tutorial, debe configurar la ingesta de datos para un dispositivo que tenga los tres flujos de datos siguientes:

- **Equipment_State**: un código numérico que representa el estado de la máquina como inactiva, averiada, parada planificada o funcionamiento normal.
- **Good_Count**: un flujo de datos donde cada punto de datos contiene el número de operaciones exitosas desde el último punto de datos.
- **Bad_Count**: un flujo de datos donde cada punto de datos contiene el número de operaciones fallidas desde el último punto de datos.

Para configurar la ingesta de datos, consulte [Ingiera datos para AWS IoT SiteWise](#). Si no tiene una operación industrial disponible, puede escribir un script que genere y cargue datos de muestra a través de la API de AWS IoT SiteWise .

Cómo calcular la OEE

En este aprendizaje, creará un modelo de activos que calcula la OEE a partir de tres flujos de entrada de datos: **Equipment_State**, **Good_Count** y **Bad_Count**. En este ejemplo, considere una máquina de envasado genérica, como una que se utiliza para empaquetar azúcar, patatas fritas o pintura. En la [AWS IoT SiteWise consola](#), cree un modelo de AWS IoT SiteWise activos con las siguientes medidas, transformaciones y métricas. A continuación, puede crear un activo para representar la máquina de embalaje y observar cómo AWS IoT SiteWise calcula la OEE.

Defina las siguientes [mediciones](#) para representar los flujos de datos sin formato de la máquina de envasado.

Mediciones

- **Equipment_State**: un flujo de datos (o medición) que proporciona el estado actual de la máquina de envasado en códigos numéricos:
 - **1024**: la máquina está inactiva.
 - **1020**: un fallo, como un error o un retraso.
 - **1000**: una parada planificada.
 - **1111**: funcionamiento normal.
- **Good_Count**: un flujo de datos donde cada punto de datos contiene el número de operaciones exitosas desde el último punto de datos.
- **Bad_Count**: un flujo de datos donde cada punto de datos contiene el número de operaciones fallidas desde el último punto de datos.

Con el flujo de datos de medida `Equipment_State` y los códigos que contiene, defina las siguientes [transformaciones](#) (o medidas derivadas). Las transformaciones tienen una one-to-one relación con las medidas sin procesar.

Transformaciones

- `Idle = eq(Equipment_State, 1024)`: un flujo de datos transformado que contiene el estado de inactividad de la máquina.
- `Fault = eq(Equipment_State, 1020)`: un flujo de datos transformado que contiene el estado de fallo de la máquina.
- `Stop = eq(Equipment_State, 1000)`: un flujo de datos transformado que contiene el estado de parada planificada de la máquina.
- `Running = eq(Equipment_State, 1111)`: un flujo de datos transformado que contiene el estado de funcionamiento normal de la máquina.

Utilizando las medidas sin formato y las medidas transformadas, defina las siguientes [métricas](#) que agreguen los datos de la máquina a lo largo de los intervalos de tiempo especificados. Seleccione el mismo intervalo de tiempo para cada métrica cuando defina las métricas en esta sección.

Métricas

- `Successes = sum(Good_Count)`: el número de envases llenados con éxito durante el intervalo de tiempo especificado.
- `Failures = sum(Bad_Count)`: el número de envases llenados sin éxito durante el intervalo de tiempo especificado.
- `Idle_Time = statetime(Idle)`: el tiempo total de inactividad de la máquina (en segundos) por intervalo de tiempo especificado.
- `Fault_Time = statetime(Fault)`: el tiempo total de fallo de la máquina (en segundos) por intervalo de tiempo especificado.
- `Stop_Time = statetime(Stop)`: el tiempo total de parada planificada de la máquina (en segundos) por intervalo de tiempo especificado.
- `Run_Time = statetime(Running)`: el tiempo total de la máquina (en segundos) funcionando sin problemas por intervalo de tiempo especificado.
- `Down_Time = Idle_Time + Fault_Time + Stop_Time`: el tiempo total de inactividad de la máquina (en segundos) durante el intervalo de tiempo especificado, calculado como la suma de los estados de la máquina distintos de `Run_Time`.

- $Availability = Run_Time / (Run_Time + Down_Time)$: el tiempo de actividad de la máquina o el porcentaje de tiempo programado que la máquina está disponible para funcionar durante el intervalo de tiempo especificado.
- $Quality = Successes / (Successes + Failures)$: el porcentaje de envases llenados con éxito de la máquina durante los intervalos de tiempo especificados.
- $Performance = ((Successes + Failures) / Run_Time) / Ideal_Run_Rate$: el rendimiento de la máquina a lo largo del intervalo de tiempo especificado como porcentaje de la tasa de ejecución ideal (en segundos) de su proceso.

Por ejemplo, su `Ideal_Run_Rate` podría ser de 60 envases por minuto (1 envase por segundo). Si su `Ideal_Run_Rate` es por minuto o por hora, deberá dividirlo por el factor de conversión de unidades apropiado porque `Run_Time` es en segundos.

- $OEE = Availability * Quality * Performance$: la eficacia general de los equipos de la máquina a lo largo del intervalo de tiempo especificado. Esta fórmula calcula la OEE como fracción de 1.

Note

Si la OEE se define como una transformación, los valores de salida se calculan para cada uno de los valores de entrada. Existe la posibilidad de que se generen valores inesperados, ya que la evaluación de la transformación considera los últimos valores disponibles para todas las propiedades que contribuyen a la fórmula. En el caso de las actualizaciones de propiedades con la misma marca de tiempo, es posible que las actualizaciones de otras propiedades entrantes sobrescriban los valores de salida. Por ejemplo, cuando se calculan disponibilidad, calidad y rendimiento, la OEE se calcula con los últimos puntos de datos disponibles para las otras dos propiedades. Estos valores contribuyentes comparten marcas de tiempo y provocan valores de salida incorrectos de la OEE. El orden no está garantizado para el cálculo de transformaciones.

Transfiere datos AWS IoT SiteWise desde cosas AWS IoT

En este tutorial, aprenda a ingerir datos AWS IoT SiteWise de una flota de AWS IoT cosas mediante el uso de sombras de dispositivos. Las sombras de los dispositivos son objetos JSON que almacenan información sobre el estado actual de un AWS IoT dispositivo. Para obtener más información, consulte [Servicio sombra de dispositivo](#) en la Guía para desarrolladores de AWS IoT .

Después de completar este tutorial, puede configurar una operación en AWS IoT SiteWise función de los AWS IoT elementos. Al usar AWS IoT cosas, puede integrar su operación con otras funciones útiles de AWS IoT. Por ejemplo, puede configurar AWS IoT funciones para realizar las siguientes tareas:

- Configurar reglas adicionales para transmitir datos a [AWS IoT Events](#), [Amazon DynamoDB](#), y otros servicios de AWS . Para obtener más información, consulte [Reglas](#) en la Guía para desarrolladores de AWS IoT .
- Indexe, busque y agregue los datos de sus dispositivos con el servicio de indexación de AWS IoT flotas. Para obtener más información, consulte [Servicio de indexación de flotas](#) en la Guía para desarrolladores de AWS IoT .
- Audite y proteja sus dispositivos con AWS IoT Device Defender. Para obtener más información, consulte [AWS IoT Device Defender](#) en la Guía para desarrolladores de AWS IoT .

En este tutorial, aprenderás a transferir datos desde dispositivos AWS IoT ocultos hasta activos internos. AWS IoT SiteWise Para ello, debe crear una o más AWS IoT cosas y ejecutar un script que actualice la sombra del dispositivo de cada una de ellas con los datos de uso de la CPU y la memoria. Utilice los datos de uso de CPU y memoria en este tutorial para imitar datos realistas del sensor. A continuación, se crea una regla con una AWS IoT SiteWise acción que envía estos datos a un activo AWS IoT SiteWise cada vez que se actualiza la sombra del dispositivo de una cosa. Para obtener más información, consulte [Ingiera datos para AWS IoT SiteWise usar reglas AWS IoT Core](#).

Temas

- [Requisitos previos](#)
- [Paso 1: Crear una AWS IoT política](#)
- [Paso 2: Crea y configura cualquier cosa AWS IoT](#)
- [Paso 3: creación de un modelo de activos de dispositivo](#)
- [Paso 4: creación de un modelo de activos de flota de dispositivos](#)
- [Paso 5: creación y configuración de un activo de dispositivos](#)
- [Paso 6: creación y configuración de un activo de flota de dispositivos](#)
- [Paso 7: Crea una regla en AWS IoT Core para enviar datos a los activos del dispositivo](#)
- [Paso 8: ejecución del script del cliente de dispositivo](#)
- [Paso 9: limpieza de los recursos después del tutorial](#)

Requisitos previos

Necesitará lo siguiente para completar este tutorial:

- Una AWS cuenta. Si no dispone de una, consulte [Configura una AWS cuenta](#).
- Un ordenador de desarrollo que ejecute Windows, macOS, Linux, o Unix para acceder al Consola de administración de AWS. Para obtener más información, consulte [Introducción a Consola de administración de AWS](#).
- Un usuario AWS Identity and Access Management (IAM) con permisos de administrador.
- Python3 instalado en tu ordenador de desarrollo o instalado en el dispositivo que quieres registrar como AWS IoT cosa.

Paso 1: Crear una AWS IoT política

En este procedimiento, cree una AWS IoT política que permita que sus AWS IoT cosas accedan a los recursos utilizados en este tutorial.

Console

Utilice el siguiente procedimiento para crear una AWS IoT política mediante la AWS IoT Core consola:

Para crear una AWS IoT política

1. Inicie sesión en la [Consola de administración de AWS](#).
2. Revise las [AWS regiones en las](#) que AWS IoT SiteWise está disponible. Cambie a una de estas regiones compatibles, si es necesario.
3. Vaya a la [consola de AWS IoT](#). Si aparece el botón Conectar dispositivo, elíjalo.
4. En el panel de navegación de la izquierda, elija Seguridad y, a continuación, elija Políticas.
5. Seleccione Crear.
6. Introduzca un nombre para la AWS IoT política (por ejemplo, **SiteWiseTutorialDevicePolicy**).
7. En Documento de política, elija JSON para introducir la siguiente política en formato JSON. Sustituya *region* y *account-id* por su región e ID de cuenta, como **us-east-1** y **123456789012**.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iot:Connect",
      "Resource": "arn:aws:iot:us-east-1:123456789012:client/SiteWiseTutorialDevice*"
    },
    {
      "Effect": "Allow",
      "Action": "iot:Publish",
      "Resource": [
        "arn:aws:iot:us-east-1:123456789012:topic/$aws/things/${iot:Connection.Thing.ThingName}/shadow/update",
        "arn:aws:iot:us-east-1:123456789012:topic/$aws/things/${iot:Connection.Thing.ThingName}/shadow/delete",
        "arn:aws:iot:us-east-1:123456789012:topic/$aws/things/${iot:Connection.Thing.ThingName}/shadow/get"
      ]
    },
    {
      "Effect": "Allow",
      "Action": "iot:Receive",
      "Resource": [
        "arn:aws:iot:us-east-1:123456789012:topic/$aws/things/${iot:Connection.Thing.ThingName}/shadow/update/accepted",
        "arn:aws:iot:us-east-1:123456789012:topic/$aws/things/${iot:Connection.Thing.ThingName}/shadow/delete/accepted",
        "arn:aws:iot:us-east-1:123456789012:topic/$aws/things/${iot:Connection.Thing.ThingName}/shadow/get/accepted",
        "arn:aws:iot:us-east-1:123456789012:topic/$aws/things/${iot:Connection.Thing.ThingName}/shadow/update/rejected",
        "arn:aws:iot:us-east-1:123456789012:topic/$aws/things/${iot:Connection.Thing.ThingName}/shadow/delete/rejected"
      ]
    },
    {
      "Effect": "Allow",
      "Action": "iot:Subscribe",
```

```

    "Resource": [
      "arn:aws:iot:us-east-1:123456789012:topicfilter/$aws/things/
      ${iot:Connection.Thing.ThingName}/shadow/update/accepted",
      "arn:aws:iot:us-east-1:123456789012:topicfilter/$aws/things/
      ${iot:Connection.Thing.ThingName}/shadow/delete/accepted",
      "arn:aws:iot:us-east-1:123456789012:topicfilter/$aws/things/
      ${iot:Connection.Thing.ThingName}/shadow/get/accepted",
      "arn:aws:iot:us-east-1:123456789012:topicfilter/$aws/things/
      ${iot:Connection.Thing.ThingName}/shadow/update/rejected",
      "arn:aws:iot:us-east-1:123456789012:topicfilter/$aws/things/
      ${iot:Connection.Thing.ThingName}/shadow/delete/rejected"
    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "iot:GetThingShadow",
      "iot:UpdateThingShadow",
      "iot>DeleteThingShadow"
    ],
    "Resource": "arn:aws:iot:us-
    east-1:123456789012:thing/SiteWiseTutorialDevice*"
  }
]
}

```

8. Seleccione Crear.

AWS CLI

Important

Esta política utiliza caracteres comodín para mantenerse dentro de los límites de tamaño de la AWS IoT SiteWise CLI. Para obtener permisos más restrictivos con rutas de temas explícitas, cree la política a través de la AWS IoT SiteWise consola. Para obtener más información, consulte el ejemplo de política de IoT que se proporciona en la pestaña.

Use el siguiente AWS CLI comando para crear una política de IoT:

```
aws iot create-policy \
```

```
--policy-name "SiteWiseTutorialDevicePolicy" \  
--policy-document '{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": "iot:Connect",  
      "Resource": "arn:aws:iot:region:account-id:client/SiteWiseTutorialDevice*"  
    },  
    {  
      "Effect": "Allow",  
      "Action": ["iot:Publish", "iot:Receive"],  
      "Resource": [  
        "arn:aws:iot:region:account-id:topic/$aws/things/  
${iot:Connection.Thing.ThingName}/shadow/*"  
      ]  
    },  
    {  
      "Effect": "Allow",  
      "Action": "iot:Subscribe",  
      "Resource": [  
        "arn:aws:iot:region:account-id:topicfilter/$aws/things/  
${iot:Connection.Thing.ThingName}/shadow/*"  
      ]  
    },  
    {  
      "Effect": "Allow",  
      "Action": [  
        "iot:GetThingShadow",  
        "iot:UpdateThingShadow",  
        "iot>DeleteThingShadow"  
      ],  
      "Resource": "arn:aws:iot:region:account-id:thing/SiteWiseTutorialDevice*"  
    }  
  ]  
}'
```

Para comprobar que la política se creó correctamente, utilice el siguiente comando:

```
aws iot get-policy --policy-name "SiteWiseTutorialDevicePolicy"
```

Esta política permite a sus AWS IoT dispositivos establecer conexiones y comunicarse con dispositivos ocultos mediante mensajes MQTT. Para obtener más información acerca de los mensajes de MQTT, consulte [What is MQTT?](#). Para interactuar con las sombras de los dispositivos, tus AWS IoT cosas publican y reciben mensajes MQTT sobre temas que comienzan con. `$aws/things/thing-name/shadow/` Esta política incorpora una variable de política de objetos denominada `${iot:Connection.Thing.ThingName}`. Esta variable sustituye el nombre del objeto conectado en cada tema. La instrucción de `iot:Connect` impone límites en cuanto a qué objetos pueden establecer conexiones, lo que garantiza que la variable de política de objeto solo puede sustituir los nombres que comienzan por `SiteWiseTutorialDevice`.

Para obtener más información, consulte [Variables de objetos de políticas](#) en la Guía para desarrolladores de AWS IoT .

Note

Esta política se aplica a objetos cuyos nombres comienzan por `SiteWiseTutorialDevice`. Para usar un nombre diferente para sus objetos, debe actualizar la política según corresponda.

Paso 2: Crea y configura cualquier cosa AWS IoT

En este procedimiento, se crea y configura cualquier AWS IoT cosa. Puede designar su ordenador de desarrollo como cualquier AWS IoT cosa. A medida que avance, recuerde que los principios que está aprendiendo aquí se pueden aplicar a proyectos reales. Tiene la flexibilidad de crear y configurar AWS IoT cosas en cualquier dispositivo capaz de ejecutar un AWS IoT SDK, incluido AWS IoT Greengrass FreeRTOS. Para obtener más información, consulte [AWS IoT SDKs](#) en la Guía para desarrolladores de AWS IoT .

Console

Para crear y configurar cualquier cosa AWS IoT

1. Abra una línea de comandos y ejecute el siguiente comando para crear un directorio para este tutorial.


```
mkdir iot-sitewise-rule-tutorial
cd iot-sitewise-rule-tutorial
```

2. Ejecute el siguiente comando para crear un directorio para los certificados de su objeto.

```
mkdir device1
```

Si está creando objetos adicionales, incremente el número en el nombre del directorio según corresponda para realizar un seguimiento de cuáles son los certificados que pertenecen a cada objeto.

3. Vaya a la [consola de AWS IoT](#).
4. En el panel de navegación de la izquierda, elija Todos los dispositivos en la sección Administrar. A continuación, elija Objetos.
5. Si aparece un cuadro de diálogo You don't have any things yet (Aún no tiene ningún objeto), elija Create a thing (Crear un objeto). De lo contrario, seleccione Crear objetos.
6. En la página Creación de objetos, elija Crear un solo objeto y luego elija Siguiente.
7. En la página Especificar propiedades de objeto, introduzca un nombre para el objeto AWS IoT (por ejemplo, **SiteWiseTutorialDevice1**) y, a continuación, seleccione Siguiente. Si está creando objetos adicionales, incremente el número en el nombre del objeto según corresponda.

 Important

El nombre de la cosa debe coincidir con el nombre utilizado en la política que creó en el paso 1: Creación de una AWS IoT política. De lo contrario, el dispositivo no podrá conectarse a AWS IoT.

8. En la página Configurar el certificado del dispositivo: opcional, selecciona Generar automáticamente un certificado nuevo (recomendado) y, a continuación, selecciona Siguiente. Los certificados permiten AWS IoT identificar sus dispositivos de forma segura.
9. En la página Adjuntar políticas al certificado (opcional), seleccione la política que creó en el paso 1: Crear una AWS IoT política y elija Crear cosa.
10. En el cuadro de diálogo Descargar certificados y claves, haga lo siguiente:
 - a. Elija los enlaces de Download (Descarga) para descargar el certificado, la clave pública y la clave privada de su objeto. Guarde los tres archivos en el directorio que creó para los certificados de su objeto (por ejemplo, **iot-sitewise-rule-tutorial/device1**).

⚠ Important

Esta es la única vez que puede descargar el certificado y las claves de su certificado, que necesita para que su dispositivo se conecte correctamente a AWS IoT.

- b. Seleccione el enlace Descargar para descargar un certificado de CA raíz. Guarde el certificado de entidad de certificación raíz en **iot-sitewise-rule-tutorial**. Recomendamos descargar Amazon Root CA 1.

11. Seleccione Listo.

AWS CLI

Siga estos pasos para crear y configurar AWS IoT cualquier cosa mediante AWS CLI:

1. Abre una línea de comandos y ejecuta el siguiente comando para crear un directorio para este tutorial:

```
mkdir iot-sitewise-rule-tutorial
```

2. Navegue hasta el directorio del tutorial:

```
cd iot-sitewise-rule-tutorial
```

3. Ejecuta el siguiente comando para crear un directorio para los certificados de tu dispositivo:

```
mkdir device1
```

Si está creando objetos adicionales, incremente el número en el nombre del directorio según corresponda para realizar un seguimiento de cuáles son los certificados que pertenecen a cada objeto.

4. Crea cualquier AWS IoT cosa:

```
aws iot create-thing --thing-name "SiteWiseTutorialDevice1"
```

⚠ Important

El nombre de la cosa debe coincidir con el patrón de nombres utilizado en la política que creó en el paso 1. De lo contrario, el dispositivo no podrá conectarse a AWS IoT.

5. Cree un certificado y guarde los archivos. Anote el ARN del certificado que aparece en el resultado; lo necesitará en los siguientes pasos:

```
aws iot create-keys-and-certificate \  
  --set-as-active \  
  --certificate-pem-outfile "device1/device.pem.crt" \  
  --public-key-outfile "device1/public.pem.key" \  
  --private-key-outfile "device1/private.pem.key"
```

6. Adjunte la política que creó en el paso 1 al certificado:

```
aws iot attach-policy \  
  --policy-name "SiteWiseTutorialDevicePolicy" \  
  --target "certificate-arn"
```

7. Adjunta el certificado a lo siguiente:

```
aws iot attach-thing-principal \  
  --thing-name "SiteWiseTutorialDevice1" \  
  --principal "certificate-arn"
```

8. Descarga el certificado de CA raíz de Amazon:

```
curl https://www.amazontrust.com/repository/AmazonRootCA1.pem >  
  AmazonRootCA1.pem
```

Este certificado es necesario para que el dispositivo se conecte correctamente AWS IoT.

⚠ Important

Guarde sus certificados y claves de forma segura. No puede volver a descargar estas credenciales después de crearlas.

Ya ha registrado AWS IoT algo en su ordenador. Siga uno de estos pasos:

- Continúe con el paso 3: Crear un modelo de activos de dispositivo sin crear AWS IoT cosas adicionales. Puede completar este tutorial con un solo objeto.
- Repita los pasos de esta sección en otro equipo o dispositivo para crear más objetos de AWS IoT . Para este tutorial, le recomendamos que siga esta opción para que pueda ingerir datos únicos de uso de memoria y CPU desde varios dispositivos.
- Repita los pasos de esta sección en el mismo dispositivo (su equipo) para crear más objetos de AWS IoT . Cada AWS IoT dispositivo recibe datos de uso de CPU y memoria similares de su ordenador, así que utilice este enfoque para demostrar la ingesta de datos no exclusivos de varios dispositivos.

Paso 3: creación de un modelo de activos de dispositivo

En este procedimiento, se crea un modelo de activos AWS IoT SiteWise para representar los dispositivos que transmiten los datos de uso de la CPU y la memoria. Para procesar datos en activos que representan grupos de dispositivos, los modelos de activos aplican información coherente en varios activos del mismo tipo. Para obtener más información, consulte [Modelado de activos industriales](#).

Para crear un modelo de activos que represente un dispositivo

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación izquierdo, elija Models (Modelos).
3. Elija Crear modelo de activos.
4. En Detalles del modelo, escriba un nombre para el modelo. Por ejemplo, **SiteWise Tutorial Device Model**.
5. En Measurement definitions (Definiciones de medida), haga lo siguiente:
 - a. En Name (Nombre), escriba **CPU Usage**.
 - b. En Unit (Unidad), escriba %.
 - c. Deje Data type (Tipo de datos) como Double (Doble).

Las propiedades de medición representan los flujos de datos sin procesar de un dispositivo. Para obtener más información, consulte [Definición de flujos de datos procedentes del equipo \(mediciones\)](#).

6. Elija Agregar nueva medida para agregar una segunda propiedad de medida.
7. En la segunda fila, en Measurement definitions (Definiciones de medida), haga lo siguiente:
 - a. En Name (Nombre), escriba **Memory Usage**.
 - b. En Unit (Unidad), escriba %.
 - c. Deje Data type (Tipo de datos) como Double (Doble).
8. En Metric definitions (Definiciones de métricas), haga lo siguiente:
 - a. En Name (Nombre), escriba **Average CPU Usage**.
 - b. En Formula (Fórmula), escriba **avg(CPU Usage)**. Elija CPU Usage en la lista de autocompletar cuando aparezca.
 - c. En Time interval (Intervalo de tiempo), escriba **5 minutes**.

Las propiedades de métrica definen cálculos de agregación que procesan todos los puntos de datos de entrada a lo largo de un intervalo y generan un único punto de datos por intervalo. Esta propiedad de métrica calcula el uso promedio de CPU de cada dispositivo cada 5 minutos. Para obtener más información, consulte [Agregación de datos desde propiedades y otros activos \(métricas\)](#).

9. Elija Agregar nueva métrica para agregar una segunda propiedad de métrica.
10. En la segunda fila en Metric definitions (Definiciones de métricas), haga lo siguiente:
 - a. En Name (Nombre), escriba **Average Memory Usage**.
 - b. En Formula (Fórmula), escriba **avg(Memory Usage)**. Elija Memory Usage en la lista de autocompletar cuando aparezca.
 - c. En Time interval (Intervalo de tiempo), escriba **5 minutes**.

Esta propiedad de métrica calcula el uso medio de memoria de cada dispositivo cada 5 minutos.

11. (Opcional) Agregue otras métricas adicionales que le interese calcular por dispositivo. Algunas funciones interesantes incluyen min y max. Para obtener más información, consulte [Uso de expresiones de fórmula](#). En el paso 4: crear un modelo de activos de flota de dispositivos, usted crea un activo principal que pueda calcular métricas utilizando datos de toda la flota de dispositivos.
12. Seleccione Crear modelo.

Paso 4: creación de un modelo de activos de flota de dispositivos

En este procedimiento, diseñará un modelo de activos AWS IoT SiteWise para simbolizar su colección de dispositivos. En este modelo de activos se establece una estructura que permite vincular varios activos de dispositivos a un activo general de la flota. A continuación, se describen las métricas del modelo de activos de la flota para consolidar los datos de todos los activos de dispositivos conectados. Este método proporciona información completa sobre el rendimiento colectivo de toda la flota.

Para crear un modelo de activos que represente una flota de dispositivos

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación izquierdo, elija Models (Modelos).
3. Elija Crear modelo de activos.
4. En Detalles del modelo, escriba un nombre para el modelo. Por ejemplo, **SiteWise Tutorial Device Fleet Model**.
5. En Hierarchy definitions (Definiciones de jerarquía), haga lo siguiente:
 - a. En Hierarchy name (Nombre de jerarquía), escriba **Device**.
 - b. En Hierarchy model (Modelo de jerarquía), elija su modelo de activos de dispositivo (**SiteWise Tutorial Device Model**).

Una jerarquía define una relación entre un modelo de activo principal (flota) y un modelo de activo secundario (dispositivo). Los activos principales pueden acceder a los datos de propiedades de los activos secundarios. Cuando cree activos más adelante, tiene que asociar los activos secundarios a activos principales según una definición de jerarquía en el modelo de activos principales. Para obtener más información, consulte [Las jerarquías de activos representan las relaciones entre los equipos](#).

6. En Metric definitions (Definiciones de métricas), haga lo siguiente:
 - a. En Name (Nombre), escriba **Average CPU Usage**.
 - b. En Formula (Fórmula), escriba **avg(Device | Average CPU Usage)**. Cuando aparezca la lista de autocompletar, elija Device para elegir una jerarquía y, a continuación, elija Average CPU Usage para elegir la métrica del activo de dispositivo que creó anteriormente.
 - c. En Time interval (Intervalo de tiempo), escriba **5 minutes**.

Esta propiedad de métrica calcula el uso medio de CPU de todos los activos de dispositivo asociados a un activo de flota a través de la jerarquía de **Device**.

7. Elija Agregar nueva métrica para agregar una segunda propiedad de métrica.
8. En la segunda fila en Metric definitions (Definiciones de métricas), haga lo siguiente:
 - a. En Name (Nombre), escriba **Average Memory Usage**.
 - b. En Formula (Fórmula), escriba **avg(Device | Average Memory Usage)**. Cuando aparezca la lista de autocompletar, elija Device para elegir una jerarquía y, a continuación, elija Average Memory Usage para elegir la métrica del activo de dispositivo que creó anteriormente.
 - c. En Time interval (Intervalo de tiempo), escriba **5 minutes**.

Esta propiedad de métrica calcula el uso medio de memoria de todos los activos de dispositivo asociados a un activo de flota a través de la jerarquía de **Device**.

9. (Opcional) Agregue otras métricas adicionales que le interese calcular en su flota de dispositivos.
10. Seleccione Crear modelo.

Paso 5: creación y configuración de un activo de dispositivos

En este procedimiento se genera un activo de dispositivos que se basa en el modelo de activos de dispositivos. A continuación, se definen los alias de propiedad para cada propiedad de medida. Un alias de propiedad es una cadena única que identifica una propiedad de activo. Más adelante podrá identificar una propiedad para la carga de datos utilizando los alias en lugar del ID del activo y el ID de la propiedad. Para obtener más información, consulte [Gestione los flujos de datos para AWS IoT SiteWise](#).

Para crear un activo de dispositivo y definir alias de propiedad

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación izquierdo, elija Assets (activos).
3. Elija Create asset (Crear activo).
4. En Información del modelo, seleccione el modelo de activo de su dispositivo, **SiteWise Tutorial Device Model**.

5. En Información del activo, introduzca un nombre para su activo. Por ejemplo, **SiteWise Tutorial Device 1**.
6. Elija Create asset (Crear activo).
7. Para el nuevo activo de dispositivo, elija Edit (Editar).
8. En Medidas:
 - a. En CPU Usage, escriba **/tutorial/device/SiteWiseTutorialDevice1/cpu** como alias de propiedad. Incluye el nombre de la AWS IoT cosa en el alias de la propiedad para poder ingerir datos de todos tus dispositivos con una sola AWS IoT regla.
 - b. En Memory Usage, escriba **/tutorial/device/SiteWiseTutorialDevice1/memory** como alias de propiedad.
9. Seleccione Save.

Si has creado varios AWS IoT elementos anteriormente, repite los pasos 3 a 10 para cada dispositivo e incrementa el número en el nombre del activo y en los alias de propiedad según corresponda. Por ejemplo, el nombre del segundo activo del dispositivo debe ser **SiteWise Tutorial Device 2** y sus alias de propiedad deben ser **/tutorial/device/SiteWiseTutorialDevice2/cpu** y **/tutorial/device/SiteWiseTutorialDevice2/memory**.

Paso 6: creación y configuración de un activo de flota de dispositivos

En este procedimiento se crea un activo de flota de dispositivos a partir del modelo de activos de flota de dispositivos. A continuación, se vinculan los activos de su dispositivo individual al activo de flota. Esta asociación permite que las propiedades métricas del activo de flota recopilen y analicen datos de varios dispositivos. Estos datos proporcionan una visión consolidada del rendimiento colectivo de toda la flota.

Para crear un activo de flota de dispositivos y asociar activos de dispositivo

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación izquierdo, elija Assets (activos).
3. Elija Create asset (Crear activo).
4. En Información de modelo, elija su modelo de activos de flota de dispositivos, **SiteWise Tutorial Device Fleet Model**.
5. En Información del activo, introduzca un nombre para su activo. Por ejemplo, **SiteWise Tutorial Device Fleet 1**.

6. Elija Create asset (Crear activo).
7. Para el nuevo activo de flota de dispositivos, elija Edit (Editar).
8. En Activos asociados a este activo, elija Agregar activo asociado y haga lo siguiente:
 - a. En Hierarchy (Jerarquía), elija Device. Esta jerarquía identifica la relación jerárquica entre el dispositivo y los activos de la flota de dispositivos. Ha definido esta jerarquía en el modelo de activos de flota de dispositivos anteriormente en este tutorial.
 - b. En Asset (activo), elija el activo del dispositivo, SiteWise Tutorial Device 1.
9. (Opcional) Si ha creado varios activos de dispositivo anteriormente, repita los pasos 8 a 10 para cada activo de dispositivo que haya creado.
10. Seleccione Save.

Ahora debería ver los activos de su dispositivo organizados como una jerarquía.

Paso 7: Crea una regla en AWS IoT Core para enviar datos a los activos del dispositivo

En este procedimiento se establece una regla en AWS IoT Core. La regla está diseñada para interpretar los mensajes de notificación de los dispositivos ocultos y transmitir los datos a los activos del dispositivo AWS IoT SiteWise. Cada vez que la sombra de tu dispositivo se actualiza, AWS IoT envía un mensaje MQTT. Puede crear una regla que actúe cuando cambien las sombras del dispositivo según el mensaje MQTT. En este caso, el objetivo es gestionar el mensaje de actualización, extraer los valores de propiedad y transmitirlos a los activos del dispositivo en AWS IoT SiteWise.

Para crear una regla con una acción AWS IoT SiteWise

1. Vaya a la [consola de AWS IoT](#).
2. En el panel de navegación de la izquierda, elija Redirección de mensajes y, a continuación, seleccione Reglas.
3. Seleccione Creación de regla.
4. Ingrese un nombre y una descripción para su regla y, a continuación, elija Siguiente.
5. Introduzca la siguiente instrucción SQL y, a continuación, seleccione Siguiente.

```
SELECT  
*
```

```
FROM
  '$aws/things/+/shadow/update/accepted'
WHERE
  startsWith(topic(3), "SiteWiseTutorialDevice")
```

Esta instrucción de consulta de regla funciona porque el servicio de sombras de dispositivo publica actualizaciones de sombras en `$aws/things/thingName/shadow/update/accepted`. Para obtener más información acerca de las sombras de dispositivos, consulte [Servicio de sombras de dispositivos](#) en la Guía de desarrolladores de AWS IoT .

En la cláusula WHERE, esta declaración de consulta de regla utiliza la función `topic(3)` para obtener el nombre de objeto del tercer segmento del tema. A continuación, la instrucción filtra los dispositivos que tienen nombres que no coinciden con los de los dispositivos del tutorial. Para obtener más información sobre AWS IoT SQL, consulte la [referencia de AWS IoT SQL](#) en la Guía para AWS IoT desarrolladores.

6. En Acciones de la regla, seleccione Enviar los datos del mensaje a las propiedades de activos en AWS IoT SiteWise y haga lo siguiente:
 - a. Elija By property alias (Por alias de propiedad).
 - b. En Property alias (Alias de propiedad), escriba `/tutorial/device/${topic(3)}/cpu`.

La `${...}` sintaxis es una plantilla de sustitución. AWS IoT evalúa el contenido de las llaves. Esta plantilla de sustitución extrae el nombre del objeto del tema para crear un alias único para cada objeto. Para obtener más información, consulte [Plantillas de sustitución](#) en la Guía para desarrolladores de AWS IoT .

Note

Dado que una expresión de una plantilla de sustitución se evalúa por separado de la instrucción SELECT, no se puede utilizar una plantilla de sustitución para hacer referencia a un alias creado mediante una cláusula AS. Solo puede hacer referencia a la información presente en la carga original, además de a las funciones y operadores compatibles.

- c. En ID de entrada: opcional, introduzca `${concat(topic(3), "-cpu-", floor(state.reported.timestamp))}`.


La entrada identifica de IDs forma única cada intento de entrada de valores. Si una entrada devuelve un error, puede encontrar el ID de entrada en la salida de error para solucionar el problema. La plantilla de sustitución en este ID de entrada combina el nombre del objeto y la marca temporal notificada por el dispositivo. Por ejemplo, el ID de entrada resultante podría tener un aspecto similar a `SiteWiseTutorialDevice1-cpu-1579808494`.

- d. En Time in seconds (Tiempo en segundos), escriba **`floor(state.reported.timestamp)`**.

Esta plantilla de sustitución calcula el tiempo en segundos a partir de la marca temporal notificada por el dispositivo. En este tutorial, los dispositivos notifican la marca temporal en segundos en formato de tiempo Unix como un número de punto flotante.

- e. En Desfase en nanosegundos opcional, escriba **`floor((state.reported.timestamp % 1) * 1E9)`**.

Esta plantilla de sustitución calcula el desfase en nanosegundos a partir de la hora en segundos convirtiendo la parte decimal de la marca temporal notificada por el dispositivo.

 Note

AWS IoT SiteWise requiere que sus datos tengan una marca de tiempo actual en Unix epoch Time. Si los dispositivos no notifican la hora con precisión, puede obtener la hora actual del motor de reglas de AWS IoT con [timestamp\(\)](#). Esta función notifica el tiempo en milisegundos, por lo que debe actualizar los parámetros de tiempo de la acción de regla a los siguientes valores:

- En Time in seconds (Tiempo en segundos), escriba **`floor(timestamp() / 1E3)`**.
- En Offset in nanos (Desfase en nanosegundos), escriba **`(timestamp() % 1E3) * 1E6`**.

- f. En Data type (Tipo de datos), elija Double (Doble).

Este tipo de datos debe coincidir con el tipo de datos de la propiedad de activo definida en el modelo de activos.

- g. En Valor, escriba **`state.reported.cpu`**. En las plantillas de sustitución, se utiliza el operador `.` para recuperar un valor desde una estructura JSON.

- h. Elija Add entry (Agregar entrada) para agregar una nueva entrada para la propiedad de uso de memoria y lleve a cabo los pasos siguientes para esa propiedad:
 - i. Elija By property alias (Por alias de propiedad).
 - ii. En Property alias (Alias de propiedad), escriba **`/tutorial/device/${topic(3)}/memory`**.
 - iii. En ID de entrada: opcional, introduzca **`${concat(topic(3), "-memory-", floor(state.reported.timestamp))}`**.
 - iv. En Time in seconds (Tiempo en segundos), escriba **`${floor(state.reported.timestamp)}`**.
 - v. En Desfase en nanosegundos opcional, escriba **`${floor((state.reported.timestamp % 1) * 1E9)}`**.
 - vi. En Data type (Tipo de datos), elija Double (Doble).
 - vii. En Valor, escriba **`${state.reported.memory}`**.
 - i. En Rol de IAM, elija Crear rol para crear un rol de IAM para esta acción de regla. Esta función le permite enviar datos AWS IoT a las propiedades del activo de su flota de dispositivos y a su jerarquía de activos.
 - j. Introduzca un nombre de rol y elija Crear.
7. (Opcional) Configure una acción de error que pueda utilizar para solucionar problemas de la regla. Para obtener más información, consulte [Solución de problemas de una regla \(AWS IoT SiteWise\)](#).
 8. Elija Siguiente.
 9. Revise la configuración y seleccione Crear.

Paso 8: ejecución del script del cliente de dispositivo

En este tutorial no se utiliza un dispositivo real para transmitir datos. En su lugar, ejecutas un script para actualizar la sombra AWS IoT del dispositivo con el uso de la CPU y la memoria para imitar los datos reales de los sensores. Para ejecutar el script, primero debe instalar los paquetes necesarios de Python. En este procedimiento, instale los paquetes de Python necesarios y, a continuación, ejecute el script del cliente del dispositivo.

Para configurar y ejecutar el script del cliente del dispositivo

1. Vaya a la [consola de AWS IoT](#).

2. En la parte inferior del panel de navegación izquierdo, elija Settings (Configuración).
3. Guarde el punto de enlace personalizado para usarlo con el script del cliente del dispositivo. Utiliza este punto de enlace para interactuar con las sombras de su objeto. Este punto de enlace es exclusivo de su cuenta en la región actual.

El punto de enlace personalizado debe ser similar al siguiente ejemplo.

```
identifier.iot.region.amazonaws.com
```

4. Abra una línea de comandos y ejecute el siguiente comando para acceder al directorio del tutorial que creó anteriormente.

```
cd iot-sitewise-rule-tutorial
```

5. Ejecute el siguiente comando para instalar AWS IoT Device SDK para Python.

```
pip3 install AWSIoTPythonSDK
```

Para obtener más información, consulte [AWS IoT Device SDK para Python](#) en la Guía para desarrolladores de AWS IoT

6. Ejecute el siguiente comando para instalar psutil, un proceso multiplataforma y una biblioteca de utilidades del sistema.

```
pip3 install psutil
```

Para obtener más información, consulte [psutil](#) en el Python Package Index.

7. Cree un archivo denominado `thing_performance.py` en el directorio `iot-sitewise-rule-tutorial` y, a continuación, copie el siguiente código Python en el archivo.

```
import AWSIoTPythonSDK.MQTTLib as AWSIoTPyMQTT

import json
import psutil
import argparse
import logging
import time

# Configures the argument parser for this program.
```

```
def configureParser():
    parser = argparse.ArgumentParser()
    parser.add_argument(
        "-e",
        "--endpoint",
        action="store",
        required=True,
        dest="host",
        help="Your AWS IoT custom endpoint",
    )
    parser.add_argument(
        "-r",
        "--rootCA",
        action="store",
        required=True,
        dest="rootCAPath",
        help="Root CA file path",
    )
    parser.add_argument(
        "-c",
        "--cert",
        action="store",
        required=True,
        dest="certificatePath",
        help="Certificate file path",
    )
    parser.add_argument(
        "-k",
        "--key",
        action="store",
        required=True,
        dest="privateKeyPath",
        help="Private key file path",
    )
    parser.add_argument(
        "-p",
        "--port",
        action="store",
        dest="port",
        type=int,
        default=8883,
        help="Port number override",
    )
    parser.add_argument(
```

```
        "-n",
        "--thingName",
        action="store",
        required=True,
        dest="thingName",
        help="Targeted thing name",
    )
    parser.add_argument(
        "-d",
        "--requestDelay",
        action="store",
        dest="requestDelay",
        type=float,
        default=1,
        help="Time between requests (in seconds)",
    )
    parser.add_argument(
        "-v",
        "--enableLogging",
        action="store_true",
        dest="enableLogging",
        help="Enable logging for the AWS IoT Device SDK for Python",
    )
    return parser

# An MQTT shadow client that uploads device performance data to AWS IoT at a
# regular interval.
class PerformanceShadowClient:
    def __init__(
        self,
        thingName,
        host,
        port,
        rootCAPath,
        privateKeyPath,
        certificatePath,
        requestDelay,
    ):
        self.thingName = thingName
        self.host = host
        self.port = port
        self.rootCAPath = rootCAPath
        self.privateKeyPath = privateKeyPath
```

```
self.certificatePath = certificatePath
self.requestDelay = requestDelay

# Updates this thing's shadow with system performance data at a regular
interval.
def run(self):
    print("Connecting MQTT client for {}".format(self.thingName))
    mqttClient = self.configureMQTTClient()
    mqttClient.connect()
    print("MQTT client for {} connected".format(self.thingName))
    deviceShadowHandler = mqttClient.createShadowHandlerWithName(
        self.thingName, True
    )

    print("Running performance shadow client for {}...
\n".format(self.thingName))
    while True:
        performance = self.readPerformance()
        print("[{}]" .format(self.thingName))
        print("CPU:\t{}%".format(performance["cpu"]))
        print("Memory:\t{}%\n".format(performance["memory"]))
        payload = {"state": {"reported": performance}}
        deviceShadowHandler.shadowUpdate(
            json.dumps(payload), self.shadowUpdateCallback, 5
        )
        time.sleep(args.requestDelay)

# Configures the MQTT shadow client for this thing.
def configureMQTTClient(self):
    mqttClient = AWSIoTPyMQTT.AWSIoTMQTTShadowClient(self.thingName)
    mqttClient.configureEndpoint(self.host, self.port)
    mqttClient.configureCredentials(
        self.rootCAPath, self.privateKeyPath, self.certificatePath
    )
    mqttClient.configureAutoReconnectBackoffTime(1, 32, 20)
    mqttClient.configureConnectDisconnectTimeout(10)
    mqttClient.configureMQTTOperationTimeout(5)
    return mqttClient

# Returns the local device's CPU usage, memory usage, and timestamp.
def readPerformance(self):
    cpu = psutil.cpu_percent()
    memory = psutil.virtual_memory().percent
    timestamp = time.time()
```

```
        return {"cpu": cpu, "memory": memory, "timestamp": timestamp}

# Prints the result of a shadow update call.
def shadowUpdateCallback(self, payload, responseStatus, token):
    print("{} {}".format(self.thingName))
    print("Update request {} {} \n".format(token, responseStatus))

# Configures debug logging for the AWS IoT Device SDK for Python.
def configureLogging():
    logger = logging.getLogger("AWSIoTPythonSDK.core")
    logger.setLevel(logging.DEBUG)
    streamHandler = logging.StreamHandler()
    formatter = logging.Formatter(
        "%(asctime)s - %(name)s - %(levelname)s - %(message)s"
    )
    streamHandler.setFormatter(formatter)
    logger.addHandler(streamHandler)

# Runs the performance shadow client with user arguments.
if __name__ == "__main__":
    parser = configureParser()
    args = parser.parse_args()
    if args.enableLogging:
        configureLogging()
    thingClient = PerformanceShadowClient(
        args.thingName,
        args.host,
        args.port,
        args.rootCAPath,
        args.privateKeyPath,
        args.certificatePath,
        args.requestDelay,
    )
    thingClient.run()
```

8. Ejecute **thing_performance.py** desde la línea de comandos con los siguientes parámetros:

- **-n, --thingName**: su nombre del objeto, como **SiteWiseTutorialDevice1**.
- **-e, --endpoint** — El AWS IoT terminal personalizado que guardaste anteriormente en este procedimiento.
- **-r, --rootCA** — La ruta a su certificado de CA AWS IoT raíz.

- `-c, --cert` — La ruta a su certificado de AWS IoT cosas.
- `-k, --key` — La ruta a la clave privada de su certificado de AWS IoT cosas.
- `-d, --requestDelay`: (opcional) el tiempo de espera en segundos entre cada actualización de la sombra de dispositivo. El valor predeterminado es 1 segundo.
- `-v, --enableLogging`: (opcional) si este parámetro está presente, el script imprime los mensajes de depuración desde el AWS IoT Device SDK para Python.

El comando debería ser similar al siguiente ejemplo.

```
python3 thing_performance.py \  
  --thingName SiteWiseTutorialDevice1 \  
  --endpoint identifier.iot.region.amazonaws.com \  
  --rootCA AmazonRootCA1.pem \  
  --cert device1/thing-id-certificate.pem.crt \  
  --key device1/thing-id-private.pem.key
```

Si está ejecutando el script para otras AWS IoT cosas, actualice el nombre de la cosa y el directorio del certificado en consecuencia.

9. Intente abrir y cerrar programas en su dispositivo para ver cómo cambian los usos de la CPU y la memoria. El script imprime cada lectura de uso de CPU y memoria. Si el script carga datos en el servicio de sombra del dispositivo correctamente, el resultado del script debe ser similar al siguiente ejemplo.

```
[SiteWiseTutorialDevice1]  
CPU:    24.6%  
Memory: 85.2%  
  
[SiteWiseTutorialDevice1]  
Update request e6686e44-fca0-44db-aa48-3ca81726f3e3 accepted
```

10. Siga estos pasos para comprobar que el script está actualizando la sombra del dispositivo:
 - a. Vaya a la [consola de AWS IoT](#).
 - b. En el panel de navegación de la izquierda, elija Todos los dispositivos y, a continuación, Objetos.
 - c. Elige lo tuyo, SiteWiseTutorialDevice.

- d. Seleccione la pestaña Sombras de dispositivo, seleccione Sombra clásica y compruebe que el Estado de la sombra es similar al del siguiente ejemplo.

```
{
  "reported": {
    "cpu": 24.6,
    "memory": 85.2,
    "timestamp": 1579567542.2835066
  }
}
```

Si el estado de sombra de tu objeto está vacío o no tiene el mismo aspecto que en el ejemplo anterior, comprueba que el script se esté ejecutando y se haya conectado correctamente AWS IoT. Si el script sigue agotándose al conectarse AWS IoT, compruebe que su [política de cosas](#) esté configurada de acuerdo con este tutorial.

11. Siga estos pasos para comprobar que la acción de regla está enviando datos a AWS IoT SiteWise:
- Vaya a la [consola de AWS IoT SiteWise](#).
 - En el panel de navegación izquierdo, elija Assets (activos).
 - Selecciona la flecha situada junto al activo de tu flota de dispositivos (SiteWise Tutorial Device Fleet 1) para ampliar su jerarquía de activos y, a continuación, selecciona el activo de tu dispositivo (SiteWise Tutorial Device1).
 - Elija Measurements (Medidas).
 - Compruebe que las celdas Latest value (Valor más reciente) tengan valores para las propiedades CPU Usage y Memory Usage.

Measurements				
Name	Alias	Notification status	Notification topic	Latest value
CPU Usage	/tutorial/device/SiteWiseTutorialDevice1/cpu	⊖ Disabled	-	24.6
Memory Usage	/tutorial/device/SiteWiseTutorialDevice1/memory	⊖ Disabled	-	85.2

- f. Si las propiedades CPU Usage y Memory Usage no tienen los valores más recientes, actualice la página. Si los valores no aparecen después de unos minutos, consulte [Solución de problemas de una regla \(AWS IoT SiteWise\)](#).

Ha completado este tutorial. Si desea explorar visualizaciones en directo de sus datos, puede configurar un portal en AWS IoT SiteWise Monitor. Para obtener más información, consulte [Supervise los datos con AWS IoT SiteWise Monitor](#). De lo contrario, puede pulsar CTRL+C en el símbolo del sistema para detener el script del cliente del dispositivo. Es poco probable que el programa Python envíe suficientes mensajes para incurrir en gastos, pero se recomienda detener el programa cuando haya terminado.

Paso 9: limpieza de los recursos después del tutorial

Note

Los recursos creados en este tutorial son necesarios para el tutorial sobre cómo [integrar datos en SiteWise Edge](#). No limpie los recursos en este paso si planea completarlo.

Después de completar el tutorial sobre la ingesta de datos de AWS IoT las cosas, limpie sus recursos para evitar incurrir en cargos adicionales.

Para eliminar activos jerárquicos en AWS IoT SiteWise

1. Vaya a la [consola AWS IoT SiteWise](#).
2. En el panel de navegación izquierdo, elija Assets (activos).
3. Al eliminar activos AWS IoT SiteWise, primero debe desasociarlos.

Complete los siguientes pasos para anular la asociación de los activos de su dispositivo de su activo de flota de dispositivos:

- a. Elige el activo de tu flota de dispositivos (SiteWise Tutorial Device Fleet 1).
- b. Elija Edit (Edición de).
- c. En Assets associated to this asset (activos asociados a este activo), elija Disassociate (Anular asociación) para cada activo de dispositivo asociado a este activo de flota de dispositivo.
- d. Seleccione Save.

Ahora ya no debería ver los activos de su dispositivo organizados como jerarquía.

4. Elija el activo de dispositivo (SiteWise Tutorial Device 1).
5. Elija Eliminar.

6. En el cuadro de diálogo de confirmación, escriba **Delete** y, a continuación, elija Delete (Eliminar).
7. Repita los pasos 4 a 6 para cada activo de dispositivo y el activo de flota de dispositivos (SiteWise Tutorial Device Fleet 1).

Para eliminar modelos de activos jerárquicos en AWS IoT SiteWise

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. Si aún no lo ha hecho, elimine el dispositivo y los activos de la flota de dispositivos. Para obtener más información, consulte [el procedimiento anterior](#). No puede eliminar un modelo si tiene activos creados a partir de ese modelo.
3. En el panel de navegación izquierdo, elija Models (Modelos).
4. Elija su modelo de activos de flota de dispositivos (SiteWise Tutorial Device Fleet Model).

Al eliminar modelos de activos jerárquicos, comience eliminando primero el modelo de activos principal.

5. Elija Eliminar.
6. En el cuadro de diálogo de confirmación, escriba **Delete** y, a continuación, elija Delete (Eliminar).
7. Repita los pasos 4 a 6 para el modelo de activos del dispositivo (SiteWise Tutorial Device Model).

Para deshabilitar o eliminar una regla en AWS IoT Core

1. Vaya a la [consola de AWS IoT](#).
2. En el panel de navegación de la izquierda, elija Redirección de mensajes y, a continuación, seleccione Reglas.
3. Seleccione su regla y seleccione Eliminar.
4. En el cuadro de diálogo de confirmación, ingrese el nombre de la regla y, a continuación, elija Eliminar.

Integre los datos en SiteWise Edge mediante una puerta de enlace V3 habilitada para MQTT

Este tutorial le guía sobre la integración de dispositivos y sensores de terceros que utilizan el protocolo de mensajería MQTT con la puerta de enlace V3 AWS IoT SiteWise habilitada para MQTT. Aprenderá a configurar una puerta de enlace AWS IoT SiteWise perimetral para recopilar y monitorear datos de sus dispositivos compatibles con MQTT. AWS IoT SiteWise le permite recopilar, procesar y monitorear datos de equipos industriales. Utilice las capacidades de SiteWise Edge para optimizar las operaciones de IoT industrial y transformar los datos sin procesar en información procesable.

En este tutorial, utilizamos los datos de una demostración de un parque eólico para ilustrar los conceptos clave. Una vez que se familiarice con el proceso, puede repetir el tutorial con sus propios datos.

Después de completar este tutorial, puede realizar las siguientes acciones:

- Instale y configure una puerta de enlace V3 habilitada para MQTT para recibir datos de dispositivos industriales
- Procese y valide los mensajes MQTT entrantes desde su equipo periférico
- Vea los datos del dispositivo AWS IoT SiteWise mediante una plataforma de visualización de terceros
- Envíe los datos procesados desde su puerta de enlace perimetral Nube de AWS a

Además, puede aprovechar las capacidades de su puerta de enlace perimetral conectándose a otros AWS IoT servicios para realizar las siguientes tareas:

- Configure AWS IoT reglas para enrutar los datos a servicios como [Amazon S3](#), [Amazon Timestream](#) y [AWS Lambda](#)
- Úselo [AWS IoT Device Defender](#) para administrar y actualizar de forma remota las configuraciones de su puerta de enlace.
- Implemente la autenticación y la autorización seguras de los dispositivos mediante funciones AWS IoT de seguridad. Para obtener más información, consulte la [AWS IoT seguridad](#) en la Guía para AWS IoT desarrolladores.
- Cree alertas y notificaciones automatizadas basadas en los datos del equipo. Para obtener más información, consulte [las reglas AWS IoT de la Guía para AWS IoT desarrolladores](#).

Note

Este tutorial hace referencia a servicios, herramientas y documentación de terceros. AWS no es vendedor ni proveedor de productos o servicios de terceros y no puede garantizar la precisión de la información proporcionada por proveedores externos. Evalúe y valide todas las herramientas de terceros antes de implementarlas.

Temas

- [Requisitos previos](#)
- [Paso 1: Crear una AWS IoT política](#)
- [Paso 2: Crea y configura cualquier cosa AWS IoT](#)
- [Paso 3: Configure su puerta de enlace V3 habilitada para SiteWise Edge MQTT](#)
- [Paso 4: Instale el software SiteWise Edge Gateway](#)
- [Paso 5: Configure el broker EMQX para que se conecte a aplicaciones externas](#)
- [Paso 6: Publica los datos con Mosquitto](#)
- [Paso 7: Especifique los destinos](#)
- [Paso 8: Especificar los filtros de ruta](#)
- [Paso 9: Configura tus recursos AWS de IoT](#)
- [Paso 10: visualice sus datos](#)
- [Paso 11: Limpiar los recursos después del tutorial](#)
- [Recursos adicionales](#)

Requisitos previos

Necesitará lo siguiente para completar este tutorial:

- Y Cuenta de AWS. Si no dispone de una, consulte [Configura una AWS cuenta](#).
- Un usuario AWS Identity and Access Management (IAM) con permisos de administrador. Para obtener más información, consulte [Gestión de identidades y accesos para AWS IoT SiteWise](#).
- La última versión de Python instalada en el dispositivo.

⚠ Important

Este tutorial requiere el uso de los recursos creados en el tutorial [sobre ingesta de datos](#). Debe completarlo antes de continuar con este tutorial.

Paso 1: Crear una AWS IoT política

En este tutorial, se utiliza la AWS IoT política que creó en el tutorial [sobre ingesta de datos](#). Esta política establece las reglas de seguridad para sus dispositivos y crea una representación digital de sus dispositivos y sensores externos. AWS IoT La política permite que sus dispositivos de terceros envíen datos AWS IoT Core mediante MQTT (Message Queuing Telemetry Transport). Para obtener más información acerca de los mensajes de MQTT, consulte [What is MQTT?](#).

Console

Asegúrese de que se complete una política. AWS IoT Para obtener instrucciones detalladas, consulta el [paso 1](#) del tutorial [sobre la ingesta de datos](#).

Para comprobar que tiene una política activa AWS IoT

1. Vaya a la [consola de AWS IoT](#).
2. En el panel de navegación izquierdo, seleccione Valores y, a continuación, Políticas.
3. Elija la política que ha creado. Por ejemplo, **SiteWiseTutorialDevicePolicy**.
4. Confirme que el estado de la política aparezca como Activo.

AWS CLI

Asegúrese de completar una AWS IoT política. Para obtener instrucciones detalladas, consulta el [paso 1](#) del tutorial [sobre la ingesta de datos](#).

Utilice el siguiente comando AWS CLI [get-policy](#) en la Referencia de AWS CLI comandos para comprobar que tiene una política activa: AWS IoT

```
aws iot get-policy --policy-name "SiteWiseTutorialDevicePolicy"
```

Esta política permite a sus AWS IoT dispositivos establecer conexiones y comunicarse con dispositivos ocultos mediante mensajes MQTT. Para interactuar con las sombras de los dispositivos,

tus AWS IoT cosas publican y reciben mensajes en MQTT sobre temas que comienzan con. `$aws/things/thing-name/shadow/` Esta política incorpora una variable de política de objetos denominada `#{iot:Connection.Thing.ThingName}`. Esta variable sustituye el nombre del objeto conectado en cada tema. La instrucción de `iot:Connect` impone límites en cuanto a qué objetos pueden establecer conexiones, lo que garantiza que la variable de política de objeto solo puede sustituir los nombres que comienzan por `SiteWiseTutorialDevice`.

Para obtener más información, consulte [Variables de objetos de políticas](#) en la Guía para desarrolladores de AWS IoT .

Note

Esta política se aplica a objetos cuyos nombres comienzan por `SiteWiseTutorialDevice`. Para usar un nombre diferente para sus objetos, debe actualizar la política según corresponda.

Paso 2: Crea y configura cualquier cosa AWS IoT

En este paso, registre su dispositivo perimetral como una AWS IoT cosa y genere los certificados y claves necesarios para una comunicación segura con AWS IoT SiteWise Edge. Este proceso sienta las bases para que su dispositivo envíe datos de terceros a través de su puerta de enlace V3 habilitada para MQTT.

Console

Asegúrese de completar los pasos de creación y configuración de cualquier cosa. AWS IoT Para obtener instrucciones detalladas, consulte el [paso 2](#) del tutorial [sobre la ingesta de datos](#).

Para comprobar que tienes algo activo AWS IoT

1. Vaya a la [consola de AWS IoT](#).
2. En el panel de navegación izquierdo, selecciona Todos los dispositivos y, a continuación, Cosas.
3. Elige lo que has creado. Por ejemplo, **SiteWiseTutorialDevice1**.
4. En Certificados, confirme que el estado aparece como activo.

AWS CLI

Asegúrese de completar los pasos de creación y configuración de cualquier AWS IoT cosa. Para obtener instrucciones detalladas, consulte el [paso 2](#) del tutorial [sobre la ingesta de datos](#).

Use el siguiente AWS CLI comando para comprobar que tiene una AWS IoT política activa:

```
aws iot describe-thing --thing-name "SiteWiseTutorialDevice1"
```

Tras completar estos pasos, podrás conectar tu dispositivo a AWS IoT SiteWise Edge de forma segura. Creó un directorio local para almacenar los certificados y las claves que generó para la autenticación en MQTT. El dispositivo está registrado como un AWS IoT objeto en la [AWS IoT consola](#) y está preparado para integrar los datos con SiteWise Edge. Puede conectar su equipo industrial u otros dispositivos a la AWS IoT plataforma y empezar a ingerir datos en SiteWise Edge.

Paso 3: Configure su puerta de enlace V3 habilitada para SiteWise Edge MQTT


En este paso, cree su puerta de enlace V3 habilitada para AWS IoT SiteWise Edge MQTT y configúrela para recibir datos del intermediario EMQX. La puerta de enlace actúa como un puente entre sus dispositivos y AWS IoT. Esto le permite procesar los datos localmente en el borde antes de enviarlos al Nube de AWS. Esta configuración reduce el ancho de banda y reduce los retrasos en el procesamiento en la nube.

Console

Para crear su puerta de enlace V3 AWS IoT SiteWise habilitada para MQTT

1. Inicie sesión en la [Consola de administración de AWS](#) y abra la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación izquierdo, selecciona Gateways Edge y, a continuación, selecciona Crear puerta de enlace.
3. En Objetivo de implementación, selecciona Puerta de enlace autohospedada.
4. En Opciones de puerta de enlace autohospedada, selecciona Puerta de enlace V3 habilitada para MQTT (recomendada).
5. En Configuración de puerta de enlace:

- a. En Nombre de la puerta de enlace, introduzca un nombre para la puerta de enlace. Por ejemplo, **SiteWise Tutorial Device Gateway**.
 - b. En el sistema operativo del dispositivo Greengrass, seleccione la opción adecuada para su dispositivo.
6. En Configuración avanzada:
- a. Seleccione Configuración predeterminada.
 - b. Introduzca un nombre para el dispositivo principal de Greengrass o utilice el nombre generado por. AWS IoT SiteWise
7. Seleccione Crear puerta de enlace.
8. En el cuadro de diálogo de confirmación, seleccione Generar y descargar para generar un instalador para su puerta de enlace SiteWise Edge. Para obtener más información, consulte [Crear una puerta de enlace SiteWise Edge autohospedada](#).

 Warning

Guarde el archivo de instalación en un lugar seguro. Este archivo no se puede regenerar y es necesario para completar la configuración de la puerta de enlace en pasos posteriores.

AWS CLI

Se utiliza AWS CLI para crear una puerta de enlace autohospedada. Debe proporcionar un nombre para la puerta de enlace, especificar la plataforma y la versión de la puerta de enlace. Para obtener más información, consulta [CreateGateway](#) en la AWS IoT SiteWise Referencia de la API de .

Para utilizar este ejemplo, reemplace los marcadores de posición de entrada del usuario con su propia información.

```
aws iotsitewise create-gateway \  
  --gateway-name SiteWise Tutorial Device Gateway \  
  --gateway-platform greengrassV2={coreDeviceThingName=your-core-device-thing-name, coreDeviceOperatingSystem=LINUX_AMD64} \  
  --gateway-version 3 \  
  \
```

```
[--cli-input-json your-configuration]
```

- `gateway-name`— Un nombre único para la puerta de enlace, por ejemplo, `SiteWiseTutorial Device Gateway`.
- `gateway-platform`— Introducir `greengrassV2`. Para obtener más información, consulta [CreateGateway](#) en la AWS IoT SiteWise Referencia de la API de .
 - `coreDeviceThingName`— El nombre del AWS IoT dispositivo AWS IoT Greengrass V2 principal. Por ejemplo, `SiteWiseTutorialDevice1`.
 - `coreDeviceOperatingSystem`— El sistema operativo del dispositivo principal en AWS IoT Greengrass V2. Para la versión 3 de la puerta de enlace es necesario especificar el sistema operativo. Las opciones incluyen: `LINUX_AARCH64`, `LINUX_AMD64`, `WINDOWS_AMD64`
- `gateway-version`— La versión de la puerta de enlace.
 - `3` Utilícela para la versión de puerta de enlace para crear una puerta de enlace V3 habilitada para MQTT.
- `cli-input-json`— Un archivo JSON que contiene los parámetros de la solicitud.

Utilice el siguiente AWS CLI comando para comprobar que la puerta de enlace se creó correctamente:

```
aws iotsitewise describe-gateway --gateway-id your-gateway-id
```

Paso 4: Instale el software SiteWise Edge Gateway

Para instalar el software de puerta de enlace, utilice el paquete de instalación que descargó en el paso anterior. El proceso de instalación configura los componentes necesarios, inicia el servicio principal de Greengrass y registra el dispositivo en él. AWS IoT Greengrass Una vez completada la instalación, compruebe que la puerta de enlace aparece en la [AWS IoT SiteWise consola](#), debajo de las puertas de enlace Edge, y que el servicio Greengrass se ejecuta correctamente en su dispositivo.

Para obtener instrucciones detalladas, consulte [Instale el software AWS IoT SiteWise Edge Gateway en su dispositivo local](#).

Paso 5: Configure el broker EMQX para que se conecte a aplicaciones externas

Note

Debe haber implementado su puerta de enlace V3 habilitada para SiteWise Edge MQTT antes de continuar. La puerta de enlace proporciona la infraestructura y los ajustes de seguridad necesarios para configurar el broker EMQX. La configuración del broker fallará sin un despliegue de puerta de enlace activo.

Configure el broker EMQX para permitir una comunicación segura entre sus dispositivos de IoT y aplicaciones externas. El intermediario EMQX funciona como un centro de mensajería central que enruta los datos entre sus dispositivos, puertas de enlace y aplicaciones de IoT. El intermediario EMQX garantiza una entrega fiable de los mensajes en su pasarela y las aplicaciones conectadas en la periferia. Para obtener más información, consulte [Connect aplicaciones externas al broker EMQX](#).

Para configurar el bróker EMQX

1. Configure el bróker EMQX. Para obtener instrucciones de configuración detalladas, siga los pasos 1 a 14 de la sección [Actualizar la configuración de despliegue de EMQX para la autenticación](#).
2. Configure los temas de MQTT para la supervisión de parques eólicos. Para obtener más información sobre los requisitos de MQTT, consulte. [Requisitos del tema MQTT](#)
 - a. Uso de la CPU: SiteWiseTutorialDevice/cpu
 - b. Uso de memoria: SiteWiseTutorialDevice/memory
 - c. Marca de tiempo: SiteWiseTutorialDevice/timestamp
3. Revise la configuración y complete la implementación.
 - a. Elija Confirmar para guardar la configuración.
 - b. Selecciona Siguiente hasta llegar al paso de revisión.
 - c. En la página Revisar, elija Implementar.
 - d. Espere a que la implementación se complete correctamente antes de continuar.

4. Prepare los mensajes con el formato de carga útil para enviarlos al intermediario EMQX. Para obtener más información sobre la estructuración de las cargas útiles, consulte [Actualizar la configuración de despliegue de EMQX para la autenticación](#).
5. Implemente las siguientes medidas de seguridad:
 - a. Utilice el cifrado Transport Layer Security (TLS) (puerto 8833) para proteger los datos en tránsito. Para obtener más información, consulte [Configure TLS para establecer conexiones seguras con el intermediario EMQX en Edge AWS IoT SiteWise](#).
 - b. Configure la autenticación con nombre de usuario y contraseña para verificar las identidades de los dispositivos. Esta medida de seguridad ayuda a proteger sus datos y garantiza que solo los dispositivos autorizados puedan conectarse a su sistema. Para obtener más información, consulte [Habilite la autenticación de nombre de usuario y contraseña](#).

EMQX le permite crear reglas de autorización basadas en identificadores como el nombre de usuario, la dirección IP o el ID de cliente. Esto es útil para controlar el acceso a sus datos. Para obtener más información, consulte [Configure las reglas de autorización para AWS IoT SiteWise Edge en EMQX](#).

Tras una implementación exitosa, su agente EMQX estará configurado y listo para conectarse de forma segura con aplicaciones externas.

Note

El formato de carga útil debe seguir una estructura específica para que AWS IoT SiteWise Edge procese e ingiera los datos correctamente. Para obtener más información sobre la estructura requerida, consulte [Estructura de carga útil de JSON](#)

Ejemplo: añadir cargas útiles JSON de CPU, memoria y marca de tiempo

Carga útil JSON de CPU

```
{
  "propertyAlias": "SiteWiseTutorialDevice/cpu",
  "propertyValues": [
    {
      "quality": "GOOD",
      "timestamp": {
```

```
    "offsetInNanos": 0,  
    "timeInSeconds": 1753206441  
  },  
  "value": {  
    "integerValue": 45.2  
  }  
}  
]  
}
```

Carga útil JSON de memoria

```
{  
  "propertyAlias": "SiteWiseTutorialDevice/memory",  
  "propertyValues": [  
    {  
      "quality": "GOOD",  
      "timestamp": {  
        "offsetInNanos": 0,  
        "timeInSeconds": 1753206441  
      },  
      "value": {  
        "integerValue": 67.8  
      }  
    }  
  ]  
}
```

Carga útil JSON con marca de tiempo

```
{  
  "propertyAlias": "SiteWiseTutorialDevice/timestamp",  
  "propertyValues": [  
    {  
      "quality": "GOOD",  
      "timestamp": {  
        "offsetInNanos": 0,  
        "timeInSeconds": 1753206441  
      },  
      "value": {  
        "integerValue": 23.5  
      }  
    }  
  ]  
}
```

```
]
}
```

Note

Cada carga útil JSON debe publicarse por separado como un mensaje individual. No combine varios valores de propiedades en un solo mensaje. Envíe cada carga útil de CPU, memoria y marca de tiempo como una publicación MQTT independiente.

La carga útil define la estructura JSON requerida que sus dispositivos de IoT deben usar para enviar los datos del dispositivo a Edge a través del intermediario EMQX. SiteWise Este formato garantiza que AWS IoT SiteWise pueda identificar sus dispositivos y procesar las lecturas de los sensores. Una vez implementadas estas configuraciones y estructuras de carga útil, el sistema de monitoreo de su parque eólico estará listo para recopilar y procesar datos.

Paso 6: Publica los datos con Mosquitto

Después de crear su puerta de enlace V3 habilitada para MQTT, configure Eclipse Mosquitto para enviar los datos de prueba a Edge. SiteWise Mosquitto es un intermediario de mensajes MQTT de código abierto que utiliza el protocolo MQTT para facilitar la mensajería entre dispositivos. El cliente Mosquitto permite publicar mensajes sobre temas de MQTT, simulando los datos de los sensores de los parques eólicos. Con Mosquitto, simule los datos del dispositivo sin necesidad de servicios de terceros ni equipos adicionales. Para obtener más información, consulte la [documentación](#) en el sitio web oficial de Eclipse Mosquitto. En este tutorial, los datos locales del tutorial de [ingestión de datos](#) y los datos ficticios se utilizan con fines de demostración.

Utilice el cliente CLI de Mosquitto para probar el broker SiteWise EDGE EMQX

1. Instale Mosquitto en su dispositivo local. Para obtener instrucciones detalladas, consulta [Descargar Mosquitto en el sitio web](#) oficial de Eclipse Mosquitto.
2. Para obtener más información sobre la conexión de aplicaciones externas para transferir datos industriales, consulte. [Connect aplicaciones externas al broker EMQX](#)

Important

Asegúrese de que los ajustes de conexión MQTT que configure aquí coincidan con los utilizados en el comando Mosquitto publish. El host debe ser la dirección IP o el nombre de

host de su SiteWise puerta de enlace Edge. El puerto suele ser el 1883 (o el 8883 si se utiliza SSL/TLS).

Use Mosquitto para publicar los datos de las pruebas. Abra una línea de comandos y ejecute los siguientes comandos:

Ejemplo: propiedad de la CPU

```
mosquitto_pub -h localhost -p 1883 -t "SiteWiseTutorialDevice/cpu" -m '{
  "propertyAlias": "SiteWiseTutorialDevice/cpu",
  "propertyValues": [
    {
      "quality": "GOOD",
      "timestamp": {
        "timeInSeconds": 1753206441,
        "offsetInNanos": 0
      },
      "value": {
        "integerValue": 45.2
      }
    }
  ]
}'
```

Ejemplo: propiedad de memoria

```
mosquitto_pub -h localhost -p 1883 -t "SiteWiseTutorialDevice/memory" -m '{
  "propertyAlias": "SiteWiseTutorialDevice/memory",
  "propertyValues": [
    {
      "quality": "GOOD",
      "timestamp": {
        "timeInSeconds": 1753206441,
        "offsetInNanos": 0
      },
      "value": {
        "integerValue": 72.1
      }
    }
  ]
}'
```

Ejemplo: propiedad Timestamp

```
mosquitto_pub -h localhost -p 1883 -t "SiteWiseTutorialDevice/timestamp" -m '{
  "propertyAlias": "SiteWiseTutorialDevice/timestamp",
  "propertyValues": [
    {
      "quality": "GOOD",
      "timestamp": {
        "timeInSeconds": 1753206441,
        "offsetInNanos": 0
      },
      "value": {
        "integerValue": 1683000000
      }
    }
  ]
}'
```

Note

El uso de localhost como dirección de intermediario de EMQX es únicamente con fines de demostración. En entornos de producción o cuando se conecta desde dispositivos externos, debe utilizar la dirección de intermediario EMQX adecuada para su configuración de despliegue específica. Para obtener instrucciones de conexión detalladas, consulte.

[Conecte una aplicación al bróker EMQX en Edge AWS IoT SiteWise](#)

Paso 7: Especifique los destinos

En este paso, especifique los destinos para determinar hacia dónde dirigir los datos de origen. Úselo AWS IoT SiteWise con el almacenamiento en búfer de Amazon S3 como destino. Esta opción proporciona una forma escalable de almacenar y procesar sus datos de IoT.

Console

Para añadir destinos

1. Navegue hasta la [AWS IoT SiteWise consola](#) y seleccione Edge Gateways.
2. En el SiteWise tutorial Device Gateway, selecciona Añadir destinos.

3. En Detalles del destino, selecciona AWS IoT SiteWise Almacenado en búfer mediante Amazon S3. Para obtener más información sobre los tipos de destinos, consulte los destinos de [AWS IoT SiteWise entrada](#).
4. En Nombre de destino, introduce un nombre para tu destino, por ejemplo, SiteWise Tutorial S3 Destination.
5. En la configuración de carga de S3, introduce la ubicación de tu depósito de S3. Por ejemplo, `s3://sitewise-tutorial-mqtt-data-[your-account-id]`. Para obtener más información sobre Amazon S3, consulte [Creación, configuración y uso de buckets de Amazon S3](#) en la Guía del usuario de Amazon Simple Storage Service.
6. En Frecuencia de carga de datos, introduzca un valor entre 1 minuto y 30 días. Por ejemplo, 1 minute.
7. En Configuración de almacenamiento de datos:
 - Deselecciona Copiar datos al almacenamiento. Si bien esta configuración se recomienda para entornos de producción, no la necesitará para este tutorial. Al deseleccionar esta opción, la opción Eliminar datos de S3 se deselecciona automáticamente.
8. Elija Add destination.

Note

En este tutorial se utiliza un intervalo de 1 minuto para realizar las pruebas. Después de completar el tutorial, puede ajustar este intervalo para que se adapte a sus necesidades de producción o eliminarlo para evitar cargos adicionales.

AWS CLI

Ejemplo: crear un nuevo AWS IoT SiteWise destino almacenado en búfer mediante Amazon S3

Utilice la referencia [update-gateway-capability-configuration](#) de AWS CLI comandos para configurar el publicador. Establezca el parámetro `capabilityNamespace` como `iotsitewise:publisher:3`.

```
{
  "sources": [
    {
      "type": "MQTT"
    }
  ]
}
```

```
    }
  ],
  "destinations": [
    {
      "type": "SITEWISE_BUFFERED",
      "name": "your-s3-destination-name",
      "config": {
        "targetBucketArn": "arn:aws:s3:::amzn-s3-demo-bucket/Optional/SomeFolder",
        "publishPolicy": {
          "publishFrequency": "1m",
          "localSizeLimitGB": 10
        },
        "siteWiseImportPolicy": {
          "enableSiteWiseStorageImport": true,
          "enableDeleteAfterImport": true,
          "bulkImportJobRoleArn": "arn:aws:iam::123456789012:role/your-role-name"
        }
      }
    },
    {
      "filters": [
        {
          "type": "PATH",
          "config": {
            "paths": [
              "#"
            ]
          }
        }
      ]
    }
  ]
}
```


Para obtener más información sobre los destinos, consulte [Añadir un destino almacenado en AWS IoT SiteWise búfer mediante Amazon S3](#).

Paso 8: Especificar los filtros de ruta

En este paso, configure los filtros de ruta para especificar qué temas de MQTT debe monitorizar para los datos de los dispositivos de su parque eólico.

Los filtros de ruta siguen la especificación comodín del tema MQTT, que admite dos caracteres especiales:

- +— Este símbolo representa un comodín de un solo nivel, que coincide con cualquier cadena de un solo nivel.
- #— Este símbolo representa un comodín de varios niveles, que coincide con cualquier número de niveles de la jerarquía de temas.

 Note

Para obtener más información sobre otros filtros de ruta, consulte [Caracteres especiales en los nombres de los filtros de ruta](#)

Console

Para configurar los filtros de ruta

En Filtros de ruta:

1. Ve a la [AWS IoT SiteWise consola](#) y selecciona Edge Gateways.
2. En el SiteWise tutorial Device Gateway, selecciona Añadir destinos.
3. Seleccione Añadir filtros de ruta para introducir los siguientes filtros de ruta manualmente:
 - SiteWiseTutorialDevice/#
 - windfarm/+/turbine/+/performance/#
 - cpu/+/idle-time
 - cpu/+/interruption-count/+
 - +/memory/consumption
 - timestamp/+/measurement
 - device/+/status/+
 - system/+/performance-log
4. Elija Add destination.

Para obtener más información sobre las prácticas recomendadas para los filtros de rutas, consulte [Prácticas recomendadas para los filtros de rutas](#).

AWS CLI

Utilice los siguientes AWS CLI comandos para configurar los filtros de ruta:

Ejemplo 1: Datos del dispositivo mediante caracteres comodín

```
{
  "destinations": [
    {
      "name": "All Device Data Destination"
    }
  ],
  "filters": [
    {
      "type": "PATH",
      "config": {
        "paths": [
          "SiteWiseTutorialDevice/#",
          "windfarm/+/turbine/+/performance/#"
        ]
      }
    }
  ]
}
```

Esta configuración de filtro de ruta utiliza caracteres comodín de varios niveles (#) para capturar todos los datos SiteWiseTutorialDevice y todos los datos de rendimiento de cualquier turbina del parque eólico.

Ejemplo 2: rendimiento de la CPU y la memoria

```
{
  "destinations": [
    {
      "name": "Performance Metrics Destination"
    }
  ],
  "filters": [
    {
      "type": "PATH",
      "config": {
        "paths": [
```

```

        "cpu/+/idle-time",
        "+/memory/consumption",
        "cpu/+/interruption-count/+"
    ]
}
]
}

```

Este ejemplo captura varias métricas de la CPU (tiempo de inactividad y recuento de interrupciones) y datos de consumo de memoria en todos los dispositivos.

Ejemplo 3: Diagnóstico de dispositivos

```

{
  "destinations": [
    {
      "name": "Device Diagnostics Destination"
    }
  ],
  "filters": [
    {
      "type": "PATH",
      "config": {
        "paths": [
          "device+/status/+",
          "system+/performance-log"
        ]
      }
    }
  ]
}

```

Esta configuración utiliza el + comodín para capturar los datos de diagnóstico de varios dispositivos, específicamente los registros de rendimiento del sistema y las actualizaciones de estado de los dispositivos.

Estos tres filtros de ruta coinciden con los temas de MQTT que se utilizan para publicar los datos de las pruebas con Mosquitto. Los filtros garantizan que su puerta de enlace SiteWise Edge capture y procese los mensajes MQTT relevantes. Para obtener más información sobre cómo añadir filtros de ruta, consulte [Agregue filtros de ruta a los destinos de AWS IoT SiteWise Edge](#).

Paso 9: Configura tus recursos AWS de IoT

En este paso, cree los modelos de activos y los AWS IoT SiteWise activos necesarios para representar sus dispositivos simulados de terceros y permitir la ingesta de datos a través de su puerta de enlace perimetral.

Antes de iniciar este paso, debería haber completado los pasos 3 a 8 del tutorial sobre la [ingesta de datos](#). Estos pasos establecen los componentes fundamentales para integrar sus datos de terceros a través de la puerta de enlace V3 habilitada para MQTT. También debe configurar reglas que definen cómo fluyen los datos de los sensores y ejecuta un script de cliente del dispositivo que simula los datos de un parque eólico industrial. AWS IoT SiteWise

Para validar la configuración de sus recursos de AWS IoT

1. Utilice el siguiente AWS CLI comando para comprobar que ha creado y configurado correctamente su modelo de dispositivo SiteWise tutorial y su modelo de flota de dispositivos SiteWise tutorial:

```
aws iotsitewise describe-asset-model --asset-model-id your-device-model-id
```

Utilice el siguiente AWS CLI comando para recuperar el ID de sus modelos de activos:

```
aws iotsitewise list-asset-models
```

2. Utilice el siguiente AWS CLI comando para comprobar que ha creado y configurado correctamente los activos SiteWise Tutorial Device 1 y SiteWise Tutorial Device Fleet 1:

```
aws iotsitewise describe-asset --asset-id your-asset-id
```

Utilice el siguiente AWS CLI comando para recuperar el ID de sus activos:

```
aws iotsitewise list-assets
```


Paso 10: visualice sus datos

Configure la versión de código abierto de Grafana para visualizar los datos de los dispositivos de su parque eólico. Grafana es una plataforma de visualización que muestra sus datos operativos en tiempo real. Estos paneles le ayudan a realizar un seguimiento de la eficiencia operativa e identificar

las necesidades de mantenimiento en toda su infraestructura. Para obtener más información sobre la integración, consulte [Integre AWS IoT SiteWise con Grafana](#).


Para configurar Grafana

1. Para obtener instrucciones para descargar e instalar la última versión de Grafana, consulte [Instalar Grafana en el sitio web oficial de Grafana](#).
2. Para obtener instrucciones de configuración detalladas específicas para su sistema operativo, consulte [Configurar Grafana en el sitio web oficial](#) de Grafana.
3. Configure la fuente de AWS IoT SiteWise datos. Esto le permite configurar el AWS IoT SiteWise complemento en su servidor Grafana. Para obtener instrucciones detalladas sobre cómo usar el complemento, consulte [Conectarse a una fuente de AWS IoT SiteWise datos](#) en la Guía del usuario de Amazon Managed Grafana.

 Important

Asegúrese de tener la última versión de Grafana para garantizar la compatibilidad con la fuente de AWS IoT SiteWise datos.

Después de completar estos pasos, puede crear y personalizar los paneles de Grafana para mostrar las métricas operativas de su parque eólico. Esto le permite realizar un seguimiento y analizar el rendimiento de su parque eólico en el perímetro en tiempo real.

 Note

Si bien este tutorial utiliza la versión de código abierto de Grafana AWS, también ofrece Amazon Managed Grafana para entornos de producción. Amazon Managed Grafana es un servicio totalmente gestionado que elimina la necesidad de instalar, configurar y mantener sus propios servidores Grafana.

Considere la posibilidad de pasarse a Amazon Managed Grafana cuando esté listo para escalar su solución. Para obtener instrucciones detalladas sobre cómo conectar sus SiteWise datos a Grafana, consulte el tutorial [Visualizar y compartir datos en Grafana](#).

Has completado el tutorial. En este procedimiento, configuró AWS IoT SiteWise Edge para integrar datos de dispositivos de terceros mediante una puerta de enlace V3 habilitada para MQTT. Esta

configuración le permite recopilar, procesar y visualizar datos de equipos industriales en la periferia, lo que reduce la latencia y los costes operativos. Al utilizar la demostración del parque eólico, recopiló y procesó métricas operativas, como los datos de uso de la CPU y la memoria, a través de su puerta de enlace V3 compatible con MQTT.

Para mejorar su solución de IoT, considere la posibilidad de explorar funciones avanzadas, como la detección de anomalías [Detección de anomalías con Lookout for Equipment](#), mediante el aprovechamiento o la integración con otros AWS servicios, como [Amazon Quick, en la Guía rápida](#) del usuario de Amazon para obtener análisis avanzados.

Paso 11: Limpiar los recursos después del tutorial

Después de completar este tutorial sobre la integración de datos en AWS IoT SiteWise Edge, limpie sus recursos para evitar incurrir en cargos adicionales.

Para eliminar activos jerárquicos en AWS IoT SiteWise

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación izquierdo, elija Assets (activos).
3. Al eliminar activos AWS IoT SiteWise, primero debe desasociarlos.

Complete los siguientes pasos para anular la asociación de los activos de su dispositivo de su activo de flota de dispositivos:

- a. Elige el activo de tu flota de dispositivos (SiteWise Tutorial Device Fleet 1).
- b. Elija Edit (Edición de).
- c. En Assets associated to this asset (activos asociados a este activo), elija Disassociate (Anular asociación) para cada activo de dispositivo asociado a este activo de flota de dispositivo.
- d. Seleccione Save.

Note

Los activos del dispositivo ya no están organizados en forma jerárquica.

4. Elija el activo de dispositivo (SiteWise Tutorial Device 1).
5. Elija Eliminar.

6. En el cuadro de diálogo de confirmación, introduzca **yDelete**, a continuación, seleccione Eliminar.
7. Repita los pasos 4 a 6 para cada activo de dispositivo y el activo de flota de dispositivos (SiteWise Tutorial Device Fleet 1).

Para eliminar modelos de activos jerárquicos en AWS IoT SiteWise

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. Elimine su dispositivo y los activos de la flota de dispositivos.
3. En el panel de navegación izquierdo, elija Models (Modelos).
4. Elija su modelo de activos de flota de dispositivos (SiteWise Tutorial Device Fleet Model). No puede eliminar un modelo si tiene activos creados a partir de ese modelo.

Al eliminar modelos de activos jerárquicos, comience eliminando primero el modelo de activos principal.

5. Elija Eliminar.
6. En el cuadro de diálogo de confirmación, ingresa **yDelete**, a continuación, selecciona Eliminar.
7. Repita los pasos 4 a 6 para el modelo de activos del dispositivo (SiteWise Tutorial Device Model).

Para deshabilitar o eliminar una regla en AWS IoT Core

1. Vaya a la [consola de AWS IoT](#).
2. En el panel de navegación izquierdo, selecciona Enrutamiento de mensajes y, a continuación, Reglas.
3. Seleccione su regla y seleccione Eliminar.
4. En el cuadro de diálogo de confirmación, ingrese el nombre de la regla y, a continuación, elija Eliminar.

Para eliminar un bucket de Amazon S3

1. Vaya a la [consola de Amazon IVS](#).
2. En el panel de navegación izquierdo, selecciona un depósito de uso general.

3. En la lista de cubos, selecciona el botón de opción situado junto al depósito que has creado y, a continuación, selecciona Vacío en la parte superior de la página.
4. En el cuadro de diálogo de confirmación, confirme la eliminación y, a continuación, seleccione Vacío.
5. Cuando el depósito esté vacío, selecciona Eliminar para eliminarlo.
6. En el cuadro de diálogo de confirmación, introduce el nombre del depósito para confirmar la eliminación.
7. Elija Delete bucket (Eliminar bucket).

Para eliminar una puerta de enlace SiteWise Edge

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación izquierdo, elija puertas de enlace Edge.
3. En Puertas de enlace, elija la puerta de enlace que creó para este tutorial. Por ejemplo, SiteWise Tutorial Device Gateway.
4. Elija Eliminar.
5. Para confirmar que desea eliminar la puerta de enlace, escriba **Delete** en el cuadro de diálogo de confirmación y, a continuación, seleccione Eliminar en la ventana que aparece.

Para eliminar tu cosa de IoT

1. Vaya a la [consola de AWS IoT](#).
2. En el panel de navegación izquierdo, selecciona Administrar y, a continuación, selecciona Cosas.
3. Selecciona el elemento de IoT que creaste para este tutorial. Por ejemplo, SiteWiseTutorialDevice1.
4. Elija Eliminar.
5. En el cuadro de diálogo de confirmación, introduce el nombre de la cosa y, a continuación, selecciona Eliminar.

Para desinstalar AWS IoT Greengrass Core

Desinstale el software AWS IoT Greengrass Core de su dispositivo local. Para obtener instrucciones detalladas, consulte [Desinstalar el software AWS IoT Greengrass principal](#) en la Guía para AWS IoT Greengrass desarrolladores, versión 2.

Important

Al desinstalar Greengrass, se eliminan todas las configuraciones y datos locales. Asegúrese de haber hecho una copia de seguridad de toda la información importante antes de continuar.

(Opcional) Para eliminar recursos de terceros

Tras completar este tutorial, considere la posibilidad de cerrar todos los recursos externos que haya creado. Esto ayuda a evitar incurrir en cargos por parte de proveedores externos.

Recursos adicionales

Consulte los siguientes recursos para obtener más información:

- [Interactúa con otros AWS servicios](#)
- [Utilice puertas de enlace AWS IoT SiteWise Edge](#)
- [Solución de problemas de una puerta de enlace SiteWise Edge](#)
- [Mejores prácticas de seguridad para AWS IoT SiteWise](#)
- [AWS Precios de IoT](#)
- [Ingiera datos para AWS IoT SiteWise](#)
- [Utilice etiquetas en AWS IoT SiteWise](#)

Visualice y comparta datos en Grafana

Este tutorial lo guía a través de la configuración del complemento de fuente de AWS IoT SiteWise datos con Grafana, una plataforma de visualización de datos. Con Grafana, puede crear cuadros de mando que visualicen y supervisen sus datos industriales. En este tutorial, se utiliza un conjunto de datos de muestra de una demostración de un parque eólico para ilustrar los conceptos clave. Una vez que se familiarice con el proceso, puede repetir el tutorial con sus propios datos.

Tras completar este tutorial, puede hacer lo siguiente:

- Recopile, consulte y analice datos de equipos industriales
- Cree paneles de Grafana interactivos para visualizar las métricas de rendimiento de los activos
- Supervise los datos operativos a través de una interfaz unificada
- Comparta información con su equipo mediante las funciones de colaboración de Grafana
- Combine AWS IoT SiteWise datos con otras fuentes AWS de datos, como [Amazon CloudWatch](#) o [Amazon Timestream](#)

Temas

- [Requisitos previos](#)
- [Paso 1: Configura tu espacio de trabajo de Grafana gestionado por Amazon](#)
- [Paso 2: AWS IoT SiteWise Añádalo como fuente de datos](#)
- [Paso 3: Crea un panel para explorar y visualizar tus datos](#)
- [\(opcional\) Paso 4: Configurar alertas para supervisar el rendimiento](#)
- [Paso 5: Limpiar los recursos después del tutorial](#)
- [Recursos adicionales](#)

Requisitos previos

Necesitará lo siguiente para completar este tutorial:

- Y. Cuenta de AWS Si no dispone de una, consulte [Configura una AWS cuenta](#).
- Un usuario AWS Identity and Access Management (IAM) con permisos de administrador. Para obtener instrucciones detalladas, consulte [the section called “¿Cómo AWS IoT SiteWise funciona con IAM”](#).
- Una AWS IoT SiteWise demostración en ejecución.

Note

Este tutorial requiere el uso de los recursos creados en la versión de [AWS IoT SiteWise demostración Use the](#). Debe completarlo antes de continuar con este tutorial.

La creación de la demostración suele tardar unos 3 minutos. Si la demostración no se crea, es posible que los permisos de tu AWS cuenta sean insuficientes. En este caso, cambie a una cuenta

con acceso administrativo. Para obtener más información acerca de los permisos requeridos, consulte [¿Cómo AWS IoT SiteWise funciona con IAM.](#)

⚠ Important

Conserve todos los recursos de demostración hasta que complete este tutorial. Eliminar cualquier componente podría interrumpir la funcionalidad de la demostración y afectar a su capacidad para mostrar datos en Grafana.


Paso 1: Configura tu espacio de trabajo de Grafana gestionado por Amazon

En este procedimiento, cree y configure un espacio de trabajo de Grafana gestionado por Amazon para visualizar los datos de su parque eólico.

1. Inicia sesión en la consola de [Grafana gestionada por Amazon](#).
2. Elija Crear espacio de trabajo.
3. En Detalles del espacio de trabajo, introduce un nombre para tu espacio de trabajo, por ejemplo. **SiteWiseTutorialDemo**
4. En Versión de Grafana, selecciona la última versión. Elija esta versión para obtener la mayoría de las up-to-date funciones y capacidades. Para obtener más información sobre los diferentes conjuntos de versiones, consulte [Diferencias entre las versiones de Grafana](#) en la Guía del usuario de Amazon Managed Grafana.
5. Elija Siguiente.
6. En Acceso de autenticación, selecciona AWS IAM Identity Center.
 - Si el centro de identidad de AWS IAM de su cuenta no está activado, se le pedirá que lo configure primero. Para obtener instrucciones detalladas sobre cómo configurar el acceso de los usuarios, consulta los [ejemplos de políticas basadas en identidad para Amazon Managed Grafana en](#) la Guía del usuario de Amazon Managed Grafana.
7. En Tipo de permiso, selecciona Servicio gestionado. Amazon Managed Grafana crea y configura automáticamente las funciones y permisos de IAM necesarios para cualquier fuente de AWS datos que decida utilizar en este espacio de trabajo. En el caso de las cuentas de los miembros de la organización, la opción de administración del servicio solo está disponible si la cuenta está designada como administradora delegada. Para obtener información sobre la configuración de

cuentas de administrador delegado, consulte [Registrar una cuenta de miembro de administrador delegado](#) en la Guía del CloudFormation usuario.

8. En las opciones de configuración del espacio de trabajo, lleve a cabo las siguientes acciones:
 - a. Selecciona Activar las alertas de Grafana. Con esta configuración, puede crear y administrar alertas a través de una interfaz centralizada en su espacio de trabajo. Para obtener más información, consulta Cómo [trabajar con alertas de Grafana en la Guía del usuario de Amazon Managed Grafana](#).
 - b. Selecciona Activar la administración de complementos. Esto te permite instalar, actualizar y desinstalar complementos en tu espacio de trabajo. Para obtener más información, consulte [Amplíe su espacio de trabajo con complementos](#) en la Guía del usuario de Amazon Managed Grafana.


 Important

Asegúrese de habilitar la administración de complementos. Si no seleccionas esta opción, no podrás añadirla AWS IoT SiteWise como fuente de datos en el siguiente paso.

9. En Control de acceso a la red, selecciona Acceso abierto. En este tutorial, utiliza datos de demostración para que el espacio de trabajo esté disponible públicamente.
 - Acceso abierto: permite que tu espacio de trabajo sea de acceso público.
 - Acceso restringido: limita el acceso a rangos de IP o puntos finales de VPC específicos. Para obtener más información, consulte [Cómo funciona la conectividad de VPC](#) en la Guía del usuario de Amazon Managed Grafana.
10. Elija Siguiente.
11. En la página de configuración de permisos gestionados por el servicio, selecciona Cuenta corriente para que Amazon Managed Grafana cree automáticamente políticas y permisos para acceder a los AWS datos de tu cuenta.
12. En Fuentes de datos, selecciona AWS IoT SiteWise. Para obtener más información, consulte [Conectarse a una fuente de AWS IoT SiteWise datos](#) en la Guía del usuario de Amazon Managed Grafana.
13. (Opcional) En Canales de notificación, selecciona Amazon SNS para permitir que las alertas de Grafana se envíen a través de Amazon SNS. Esto crea una política de IAM que permite publicar en Amazon SNS temas con nombres que comiencen por Grafana. Deberás completar la

configuración del canal de notificaciones más adelante en tu consola Grafana dentro del espacio de trabajo.

14. Confirme los detalles del espacio de trabajo y elija Crear espacio de trabajo. Este proceso tarda un par de minutos.
15. En la pestaña Autenticación, en el Centro de identidades de AWS IAM, asigne usuarios o grupos a su espacio de trabajo de la siguiente manera:
 - Para asignar el usuario que administrará AWS IoT SiteWise los datos, seleccione Asignar un nuevo usuario o grupo. A continuación, selecciona Convertir en administrador en la lista desplegable Acciones para otorgarles privilegios administrativos.

 Important

Para administrar el espacio de trabajo de Grafana, debes asignar el Admin rol a al menos un usuario. Este usuario tendrá acceso completo a la consola del espacio de trabajo de Grafana.

Ya ha creado y configurado su espacio de trabajo de Grafana. En el siguiente paso, puede añadirlo AWS IoT SiteWise como fuente de datos y empezar a crear visualizaciones para los datos de su parque eólico. Desde su espacio de trabajo, puede consultar, visualizar y analizar sus datos industriales en tiempo real. Para obtener más información sobre los espacios de trabajo de Grafana gestionados por Amazon, consulte Utilice [su espacio de trabajo de Grafana en la Guía](#) del usuario de Amazon Managed Grafana.

Paso 2: AWS IoT SiteWise Añádalo como fuente de datos

Para ayudarle a visualizar sus datos, Amazon Managed Grafana incluye el complemento AWS Data Sources, que simplifica el proceso de conexión a. Servicios de AWS Este complemento viene preinstalado en su espacio de trabajo y proporciona una interfaz unificada para descubrir y configurar AWS los recursos como fuentes de datos. Para la visualización de su parque eólico, utilizará este complemento para conectarse a AWS IoT SiteWise. Para obtener más información, consulte [Conectarse a una fuente de AWS IoT SiteWise datos](#) en la Guía del usuario de Amazon Managed Grafana.

Antes de empezar a consultar los datos de su parque eólico, el complemento AWS Data Sources necesita los permisos adecuados para acceder a sus AWS IoT SiteWise recursos. Estos permisos se configuraron automáticamente cuando los seleccionaste AWS IoT SiteWise como fuente de datos

en el paso anterior. Para obtener más información sobre los permisos de los complementos, consulte [Permisos necesarios](#) en la Guía del usuario de Amazon Managed Grafana.

Para conectarse AWS IoT SiteWise a su espacio de trabajo de Grafana


1. Abra la [consola de Amazon Managed Grafana](#). En la página de detalles de tu espacio de trabajo, elige la URL que aparece debajo de la URL del espacio de trabajo de Grafana. La URL del espacio de trabajo abre la página de inicio de sesión de la consola del espacio de trabajo de Grafana.
2. Seleccione Iniciar sesión con AWS IAM Identity Center e introduzca sus credenciales. Estas credenciales solo funcionan si respondiste al correo electrónico de Grafana gestionado por Amazon en el que se te pedía que crearas una contraseña para el Centro de Identidad de IAM.
3. En el panel de navegación izquierdo, selecciona Aplicaciones, luego Fuentes de AWS datos y, a continuación, selecciona la Servicios de AWSpestaña.
4. En AWS IoT SiteWise, selecciona Instalar ahora para instalar la última versión del AWS IoT SiteWise complemento.
5. Ve a la pestaña Fuentes de datos y selecciona IoT SiteWise como servicio.
6. En Región predeterminada, selecciona la región de la que quieres recuperar los datos, por ejemplo, EE. UU. Este (Norte de Virginia).
7. Tras especificar los parámetros del complemento, seleccione Añadir fuente de datos.
8. Seleccione Ir a la configuración.
9. En Detalles de la conexión, seleccione Guardar y probar para comprobar que el servicio funciona.

Paso 3: Crea un panel para explorar y visualizar tus datos

En este paso, cree un panel de Grafana para visualizar los datos de demostración del parque eólico que creó anteriormente. Los paneles le ayudan a monitorear sus datos al mostrar múltiples visualizaciones en una sola vista. Puede usar los paneles de control para realizar un seguimiento de las métricas, analizar patrones y obtener información a partir de sus datos industriales. Para obtener más información, consulta Cómo [crear tu primer panel](#) en la Guía del usuario de Amazon Managed Grafana.

Para crear tu primer panel de control en Grafana

1. En el panel de navegación izquierdo, selecciona Paneles y, a continuación, selecciona Crear panel para empezar a crear tu primer panel.
2. Seleccione Añadir visualización. Se abre el editor del panel, donde puede configurar las fuentes de datos, las consultas y los ajustes de visualización.
3. En la pestaña Consulta, selecciona la fuente de AWS IoT SiteWise datos en el menú desplegable.
4. En Tipo de consulta, seleccione Obtener agregados del valor de las propiedades en el menú desplegable para recuperar los valores agregados de las propiedades de los activos a lo largo del tiempo.
5. Seleccione Explorar para ver los activos disponibles en su jerarquía. En la pestaña Jerarquía, seleccione Demo Wind Farm Asset y, a continuación, Demo Turbine Asset 1.
6. En Propiedad, seleccione Potencia media entre las propiedades disponibles. Seleccione Ejecutar consultas para ejecutar la consulta y obtener una vista previa del resultado. La visualización se actualizará para mostrar los datos de potencia media de Demo Turbine Asset 1.
7. En el panel de navegación derecho, asigne un título al nuevo panel, como **Turbine Demo 1 (Average Power)**. Elija Aplicar para guardar los cambios.

 Warning

Siempre que realices cambios en el panel de control, guárdalo antes de actualizar la página o salir de ella. De lo contrario, perderás tu progreso.

8. En la esquina superior derecha, selecciona Guardar panel. Se le pedirá que introduzca un nombre para su panel de control, por ejemplo, **SiteWise Wind Farm Demo Dashboard**.
9. Seleccione Save.

Para obtener información sobre cómo compartir paneles, consulte [Compartir paneles y paneles](#) en la Guía del usuario de Amazon Managed Grafana.

Para añadir otro panel para visualizar la velocidad del viento

1. Seleccione Añadir visualización para abrir un panel en blanco.
2. En la pestaña Consulta, selecciona la fuente de AWS IoT SiteWise datos en el menú desplegable.

3. En Tipo de consulta, selecciona Obtener el valor de la propiedad en el menú desplegable y, en Activo, selecciona Demo Wind Farm Asset y, a continuación, Demo Turbine Asset 1
4. En Propiedad, selecciona Velocidad del viento entre las propiedades disponibles. Seleccione Ejecutar consultas para actualizar los cambios.
5. En Visualización, selecciona Calibre. Los medidores funcionan mejor para mostrar métricas únicas y en tiempo real, como la velocidad del viento.
6. En el panel de navegación derecho, asigne un título al nuevo panel, como **Turbine Demo 1 (Wind Speed)**.
7. En Opciones estándar, en las opciones del panel, selecciona Unidad. Elija Velocidad y, a continuación, elija meters/second (m/s).
8. Elija Aplicar para guardar los cambios.

La siguiente imagen muestra el aspecto que podrían tener los paneles de Grafana al completar este paso.



(opcional) Paso 4: Configurar alertas para supervisar el rendimiento

Las alertas indican los cambios de estado una vez que se producen para identificar problemas de rendimiento en sus equipos industriales. Para obtener más información, consulta las [alertas de Amazon Managed Grafana en la Guía del usuario de Amazon Managed Grafana](#).

Para configurar alertas en Grafana

1. En la pestaña ReglaTurbine Demo 1 (Average Power), establece Evaluar todos a 5m y Para15m. Esta configuración evalúa la potencia media cada 5 minutos y activa una alerta si la condición persiste durante más de 15 minutos.

2. En Condiciones, seleccione IS a continuación e introduzca los **7,020** vatios. Esta configuración le notificará si las condiciones medias de la turbina caen por debajo de los 7.020 vatios durante más de 5 minutos. Para obtener más información sobre la creación de alertas, consulte [los campos de reglas de alerta](#) en la Guía del usuario de Amazon Managed Grafana.

Has completado el tutorial. En este procedimiento, creó un espacio de trabajo de Grafana y lo configuró para visualizar los datos del parque eólico. AWS IoT SiteWise Creó un panel interactivo con varios tipos de widgets, incluido un gráfico de series temporales para la potencia media y un indicador de la velocidad del viento. También configuraste alertas para monitorear el rendimiento de la turbina, lo que te permitió identificar posibles problemas antes de que interrumpan la producción. Puede seguir mejorando su panel de control añadiendo más visualizaciones, creando alertas adicionales o conectando otras fuentes de AWS datos para obtener información más detallada sobre sus operaciones industriales.

Paso 5: Limpiar los recursos después del tutorial

Después de completar este tutorial sobre la visualización de datos con Grafana, limpie sus recursos para evitar incurrir en cargos adicionales.

Para eliminar la demostración AWS IoT SiteWise

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En la esquina superior derecha de la página, selecciona Eliminar demostración.
3. En el cuadro de diálogo de confirmación, escriba **DELETE** y, a continuación, elija Delete (Eliminar).

Para obtener más información, consulte [Elimine la AWS IoT SiteWise demostración](#).

Si eliminas un Grafana gestionado por Amazon, también se eliminarán todos los datos de configuración de ese espacio de trabajo. Esto incluye paneles, configuración de orígenes de datos, alertas e instantáneas.

Para eliminar un espacio de trabajo de Grafana gestionado por Amazon

1. Abra la [consola de Amazon Managed Grafana](#).
2. En el panel de navegación de la izquierda, elija el icono del menú.
3. Elija Todos los espacios de trabajo.

4. Elija el nombre del espacio de trabajo que desea eliminar.
5. Elija Eliminar.
6. Para confirmar la eliminación, ingrese el nombre del espacio de trabajo y elija Eliminar.

Note

Este procedimiento elimina un espacio de trabajo. Es posible que no se eliminen otros recursos. Por ejemplo, los roles de IAM que estaban en uso en el espacio de trabajo no se eliminan (pero se pueden desbloquear si ya no se utilizan).

Para obtener más información, consulte [Eliminar un espacio de trabajo de Grafana gestionado por Amazon](#) en la Guía del usuario de Amazon Managed Grafana.

Recursos adicionales

Para obtener más información sobre la visualización de datos, consulte los siguientes recursos:

- [Solución de problemas de acceso e identidad de Amazon Managed Grafana](#) en la Guía del usuario de Amazon Managed Grafana
- [Mejores prácticas de seguridad](#) en la Guía del usuario de Amazon Managed Grafana
- [Integre AWS IoT SiteWise con Grafana](#)
- [Procese y visualice datos con SiteWise Edge y herramientas de código abierto](#)
- [Usuarios, equipos y permisos](#) en la Guía del usuario de Amazon Managed Grafana
- [Permisos y políticas de Amazon Managed Grafana para fuentes de AWS datos](#) en la Guía del usuario de Amazon Managed Grafana

Visualice y comparta datos de parques eólicos en SiteWise Monitor

En este tutorial se explica cómo visualizar AWS IoT SiteWise Monitor y compartir datos industriales a través de aplicaciones web gestionadas, conocidas como portales. Cada portal abarca proyectos, lo que le proporciona la flexibilidad de elegir los datos a los que se puede acceder en cada proyecto. A continuación, especifique las personas de su organización que pueden acceder a cada portal. Sus usuarios inician sesión en portales mediante cuentas de AWS IAM Identity Center, por lo que puede utilizar su almacén de identidades existente o uno administrado por AWS.

Usted y sus usuarios con permisos suficientes pueden crear paneles en cada proyecto para visualizar sus datos industriales de manera significativa. A continuación, los usuarios pueden ver estos paneles para obtener rápidamente información sobre sus datos y monitorizar su operación. Puede configurar permisos administrativos o de solo lectura para cada proyecto para cada usuario de su empresa. Para obtener más información, consulte [Supervise los datos con AWS IoT SiteWise Monitor](#).

A lo largo del tutorial, mejorará la AWS IoT SiteWise demostración y proporcionará un conjunto de datos de muestra para un parque eólico. Configura un portal en SiteWise Monitor, crea un proyecto y paneles para visualizar los datos del parque eólico. El tutorial también incluye la creación de usuarios adicionales, junto con la asignación de permisos para controlar o visualizar el proyecto y sus paneles asociados.

Note

Al usar SiteWise Monitor, se le cobra por cada usuario que inicie sesión en un portal (por mes). En este tutorial, crea tres usuarios, pero solo tiene que iniciar sesión con uno. Después de completar este tutorial, generará cargos por un usuario. Para obtener más información, consulte [AWS IoT SiteWise Precios](#).

Temas

- [Requisitos previos](#)
- [Paso 1: Crear un portal en Monitor SiteWise](#)
- [Paso 2: inicio de sesión en un portal](#)
- [Paso 3: creación de un proyecto de parque eólico](#)
- [Paso 4: creación de un panel para visualizar datos de parques eólicos](#)
- [Paso 5: exploración del portal](#)
- [Paso 6: limpieza de los recursos después del tutorial](#)

Requisitos previos

Necesitará lo siguiente para completar este tutorial:

- Un Cuenta de AWS. Si no dispone de una, consulte [Configura una AWS cuenta](#).

- Un ordenador de desarrollo en ejecución Windows, macOS, Linux, o Unix para acceder al Consola de administración de AWS. Para obtener más información, consulte [¿Qué es Consola de administración de AWS?](#)
- Un usuario AWS Identity and Access Management (IAM) con permisos de administrador.
- Una demostración de un AWS IoT SiteWise parque eólico en funcionamiento. Al configurar la demostración, esta define los modelos y los activos AWS IoT SiteWise y les transmite datos para representar un parque eólico. Para obtener más información, consulte [Usa la AWS IoT SiteWise demo](#).
- Si ha activado el Centro de identidad de IAM en su cuenta, inicie sesión en su cuenta AWS Organizations de gestión. Para obtener más información, consulte [Terminología y conceptos de AWS Organizations](#). Si no ha habilitado el Centro de identidades de IAM, lo habilitará en este tutorial y configurará su cuenta como cuenta de administración.

Si no puede iniciar sesión en su cuenta AWS Organizations de administración, puede completar parcialmente el tutorial siempre que tenga un usuario del IAM Identity Center en su organización. En este caso, podrá crear el portal y los paneles de control, pero no podrá crear nuevos usuarios del Centro de identidades de IAM para asignarlos a los proyectos.

Paso 1: Crear un portal en Monitor SiteWise

En este procedimiento, crea un portal en AWS IoT SiteWise Monitor. Cada portal es una aplicación web administrada a la que usted y sus usuarios pueden acceder con cuentas de AWS IAM Identity Center. El Centro de identidades de IAM le permite utilizar el almacén de identidades existente en su empresa o crear uno administrado por AWS. Los empleados de su empresa pueden iniciar sesión sin crear Cuentas de AWS independientes.

Para crear un portal

1. Inicie sesión en la [consola de AWS IoT SiteWise](#).
2. Revise los [AWS IoT SiteWise puntos finales y las cuotas](#) compatibles y cambie de región, si AWS IoT SiteWise es necesario. Debe ejecutar la AWS IoT SiteWise demostración en la misma región.
3. En el panel de navegación izquierdo, elija Portales.
4. Elija Create portal (Crear portal).
5. Si ya ha habilitado el Centro de identidades de IAM, vaya al paso 6. Caso contrario, complete los siguientes pasos para habilitar el Centro de identidades de IAM:

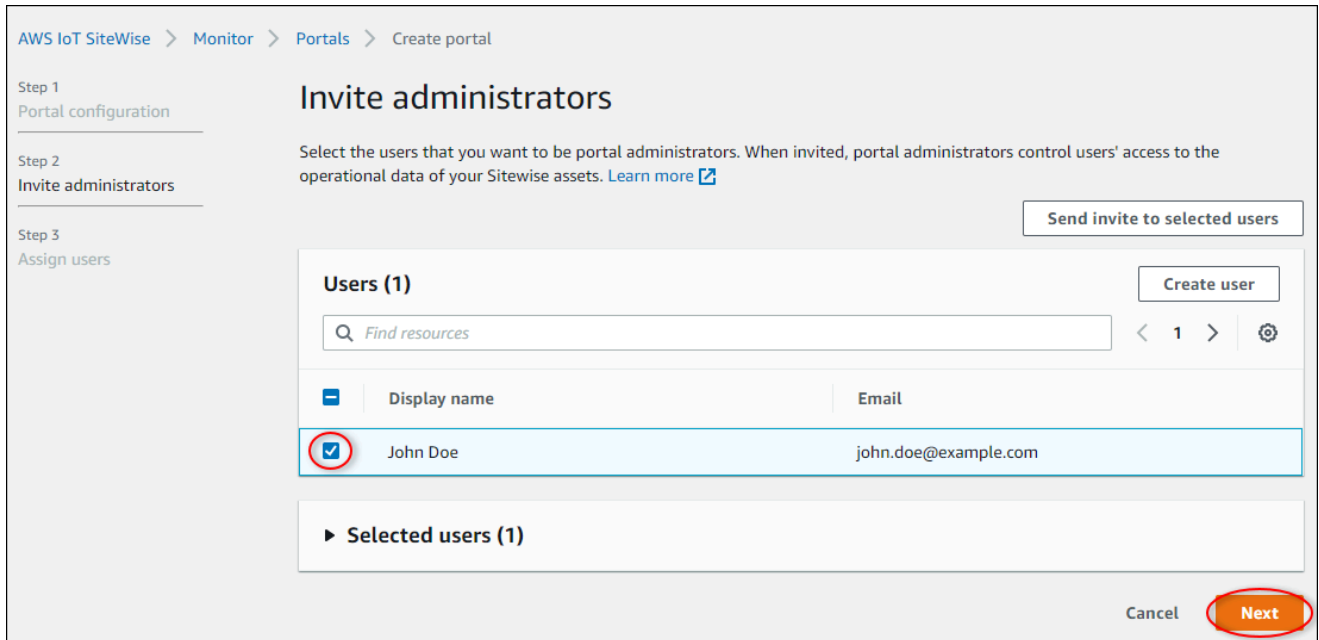
- a. En la página Habilitar AWS IAM Identity Center (SSO), introduzca su dirección de correo electrónico, nombre y apellidos para crear un usuario del IAM Identity Center que será el administrador del portal. Utilice una dirección de correo electrónico a la que pueda acceder para recibir un mensaje con el fin de establecer una contraseña para su nuevo usuario del Centro de identidades de IAM.

En un portal, el administrador del portal crea proyectos y asigna usuarios a proyectos. Puede crear más usuarios más adelante.

The screenshot shows the 'Enable AWS Single Sign-On (SSO)' page in the AWS IoT SiteWise Monitor console. The breadcrumb trail is 'AWS IoT SiteWise > Monitor > Portals > Create portal'. The page is divided into four steps: Step 1 (Enable SSO), Step 2 (Portal configuration), Step 3 (Invite administrators), and Step 4 (Assign users). The main heading is 'Enable AWS Single Sign-On (SSO)'. Below the heading, there is a paragraph explaining that SSO is required to create a portal and invite users, and a 'Learn more' link. The 'Create a user' form is the central focus, with fields for 'Email address' (john.doe@example.com), 'First name' (John), and 'Last name' (Doe). A 'Create user' button is highlighted in orange at the bottom right of the form.

- b. Seleccione la opción Crear un usuario.
6. En la página Configuración del portal, siga los pasos que se describen a continuación:
 - a. Escriba un nombre para el portal, como **WindFarmPortal**.
 - b. (Opcional) Escriba una descripción para el portal. Si tiene varios portales, utilice descripciones significativas para realizar un seguimiento de lo que contiene cada portal.
 - c. (Opcional) Cargue una imagen para mostrarla en el portal.
 - d. Introduzca una dirección de correo electrónico con la que los usuarios del portal puedan ponerse en contacto cuando tengan un problema con el portal y necesiten la ayuda del AWS administrador de la empresa para resolverlo.
 - e. Elija Create portal (Crear portal).

7. En la página Invitar administradores, puede asignar usuarios del Centro de identidades de IAM al portal como administradores. Los administradores del portal administran los permisos y los proyectos dentro de un portal. En esta página, haga lo siguiente:
 - a. Seleccione un usuario que vaya a ser el administrador del portal. Si habilitó el Centro de identidades de IAM anteriormente en este tutorial, seleccione el usuario que creó.



- b. (Opcional) Elija Enviar invitación a los usuarios seleccionados. Se abrirá su cliente de correo electrónico y aparecerá una invitación en el cuerpo del mensaje. Puede personalizar el correo electrónico antes de enviarlo a los administradores del portal. También puede enviar el correo electrónico a los administradores de su portal más tarde. Si es la primera vez que prueba SiteWise Monitor y va a ser el administrador del portal, no es necesario que se envíe un correo electrónico.
 - c. Elija Siguiente.
8. En la página Asignar usuarios, puede asignar usuarios del Centro de identidades de IAM al portal. Posteriormente, los administradores del portal podrán asignar a estos usuarios como propietarios u observadores de proyectos. Los propietarios de proyectos pueden crear paneles de control en los proyectos. Los observadores de proyectos tienen acceso de solo lectura a los proyectos que tengan asignados. En esta página puede crear usuarios del Centro de identidades de IAM para añadirlos al portal.

Note

Si no ha iniciado sesión en su cuenta AWS Organizations de administración, no podrá crear usuarios del IAM Identity Center. Seleccione Asignar usuarios para crear el portal sin usuarios del portal y, a continuación, omita este paso.

En esta página, haga lo siguiente:

- a. Complete los siguientes pasos dos veces para crear dos usuarios del Centro de identidades de IAM:
 - i. Seleccione Crear usuario para abrir un cuadro de diálogo en el que introducirá los detalles del nuevo usuario.
 - ii. Introduzca Dirección de correo electrónico, Nombre y Apellido para el nuevo usuario. El Centro de identidades de IAM enviará al usuario un correo electrónico para que establezca su contraseña. Si desea iniciar sesión en el portal como estos usuarios, elija una dirección de correo electrónico a la que pueda acceder. Cada dirección de correo electrónico debe ser única. Sus usuarios inician sesión en el portal utilizando su dirección de correo electrónico como nombre de usuario.

Create user [X]

Create a new AWS user. You can assign this user access to AWS applications and services

Email address
mary.major@example.com

First name: Mary Last name: Major

Cancel **Create user**

- iii. Seleccione la opción Crear un usuario.
- b. Seleccione los dos usuarios del Centro de identidades de IAM que creó en el paso anterior.

AWS IoT SiteWise > Monitor > Portals > WindFarmPortal > Assign users

Assign users

Users (3) Create user

Find resources

	Display name	Email
<input type="checkbox"/>	John Doe	john.doe@example.com
<input checked="" type="checkbox"/>	Mary Major	mary.major@example.com
<input checked="" type="checkbox"/>	Mateo Jackson	mateo.jackson@example.com

Selected users (2)

Cancel Assign users

- c. Seleccione Asignar usuarios para añadir estos usuarios al portal.

La página de portales se abre con su nuevo portal mostrado.

Paso 2: inicio de sesión en un portal

En este procedimiento, inicia sesión en el nuevo portal con el AWS IAM Identity Center usuario que ha agregado al portal.

Para iniciar sesión en un portal

1. En la página Portals (Portales), elija el Link (Enlace) del nuevo portal para abrirlo en una nueva pestaña.

The screenshot shows the AWS IoT SiteWise 'Portals' management interface. At the top, there are navigation links for 'AWS IoT SiteWise', 'Monitor', and 'Portals'. Below the navigation, there's a section titled 'Portals (1)' with buttons for 'Delete', 'View details', and 'Create portal'. A descriptive text explains that employees can use web portals to access AWS IoT SiteWise asset data. Below this is a search bar labeled 'Filter portals'. A table lists the portals with columns: Name, Link, Date last modified, Date created, and Status. One portal is listed: 'WindFarmPortal' with a link 'https://a1b2c3d4-5678-90ab-cdef-1111EXAMPLE.app.iotsitewise.aws' circled in red, a date last modified of '04-28-2020', a date created of '04-20-2020', and a status of 'Active'.

2. Si ha creado su primer usuario del Centro de identidades de IAM anteriormente en el tutorial, realice los pasos siguientes para crear una contraseña para su usuario:
 - a. Revise su correo electrónico para ver el asunto Invitation to join AWS IAM Identity Center.
 - b. Abra ese correo electrónico de invitación y elija Accept invitation.
 - c. En la nueva ventana, establezca una contraseña para su usuario del Centro de identidades de IAM.

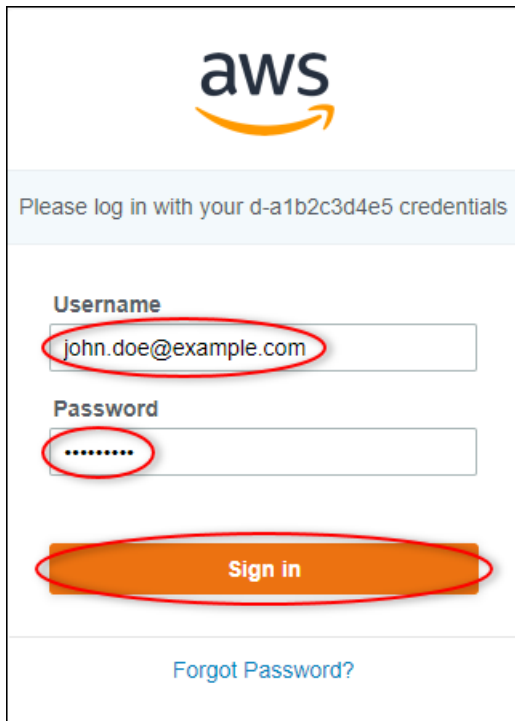
Si desea iniciar sesión más adelante en el portal como el segundo y tercer usuario del Centro de identidades de IAM que creó anteriormente, también puede completar estos pasos para establecer las contraseñas de dichos usuarios.

Note

Si no ha recibido un correo electrónico, puede generar una contraseña para su usuario en la consola del Centro de identidades de IAM. Para obtener más información, consulte [Restablecimiento de la contraseña de usuario de IAM Identity Center para un usuario final](#) en la Guía del usuario de AWS IAM Identity Center .

3. Introduzca su Username y Password del Centro de identidades de IAM. Si creó su usuario del Centro de identidades de IAM anteriormente en este tutorial, su Username es la dirección de correo electrónico del usuario administrador del portal que creó.

Todos los usuarios del portal, incluyendo el administrador del portal, deben iniciar sesión con sus credenciales de usuario del Centro de identidades de IAM. Normalmente, estas credenciales no son las mismas que se utilizan para iniciar sesión en la Consola de administración de AWS.



aws

Please log in with your d-a1b2c3d4e5 credentials

Username
john.doe@example.com

Password
.....

Sign in

[Forgot Password?](#)

4. Elija Sign in.

Se abrirá su portal.

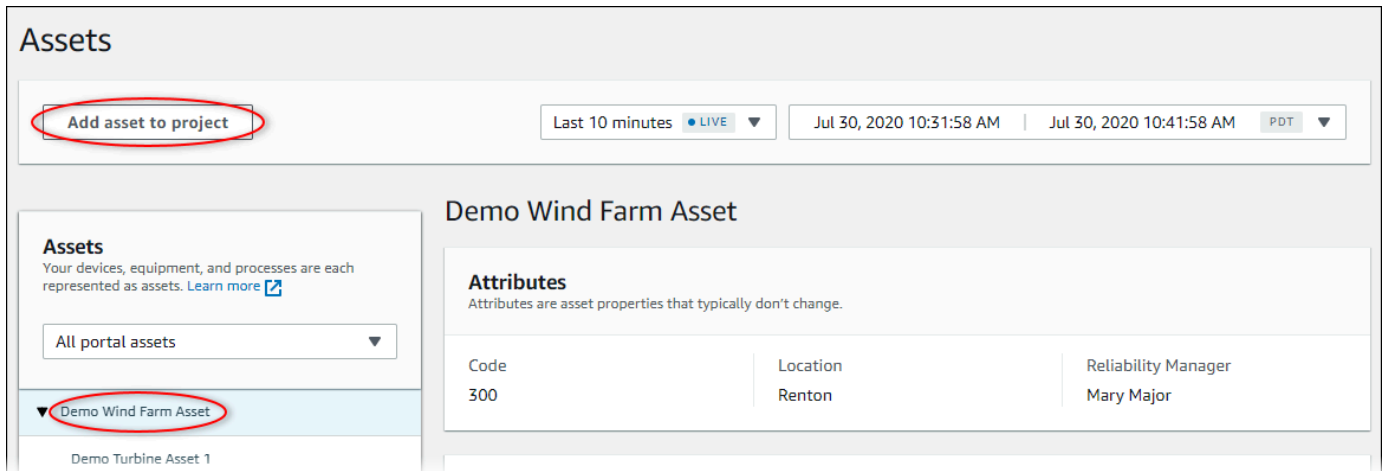
Paso 3: creación de un proyecto de parque eólico

En este procedimiento, crea un proyecto en el portal. Los proyectos son recursos que definen un conjunto de permisos, activos y paneles, que se pueden configurar para visualizar datos de activos en ese proyecto. Con los proyectos, define quién tiene acceso a qué subconjuntos de su operación y cómo se visualizan los datos de esos subconjuntos. Puede asignar a los usuarios del portal como propietarios u observadores de cada proyecto. Los propietarios de proyectos pueden crear paneles de control para visualizar datos y compartir el proyecto con otros usuarios. Los observadores de proyectos pueden ver los paneles de control, pero no editarlos. Para obtener más información sobre las funciones en SiteWise Monitor, consulte [SiteWise Supervise las funciones](#).

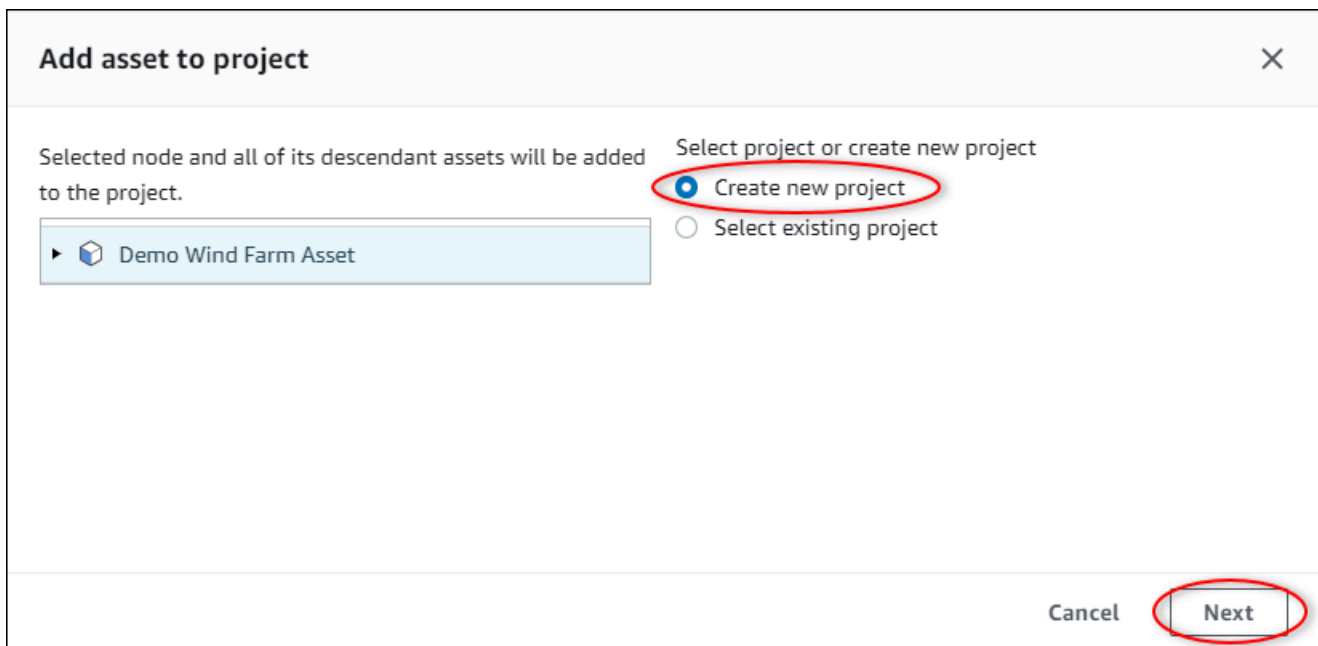
Para crear un proyecto de parque eólico

1. En el panel de navegación izquierdo de su portal, seleccione la pestaña Activos. En la página Activos, puede explorar todos los activos disponibles en el portal y añadirlos a proyectos.


2. En el navegador de activos, elija Demo Wind Farm Asset. Al elegir un activo, puede explorar los datos históricos y en directo del mismo. También puede presionar Shift para seleccionar varios activos y comparar sus datos side-by-side.
3. Seleccione Añadir activo a proyecto en la parte superior izquierda. Los proyectos contienen paneles que pueden ver los usuarios del portal para explorar los datos. Cada proyecto tiene acceso a un subconjunto de tus activos en AWS IoT SiteWise. Cuando añade un activo a un proyecto, todos los usuarios con acceso a ese proyecto también pueden acceder a los datos de ese activo y sus elementos secundarios.



4. En el cuadro de diálogo Añadir activo al proyecto, seleccione Crear nuevo proyecto y, a continuación, Siguiente.



5. En el cuadro de diálogo Crear nuevo proyecto, introduzca el Nombre del proyecto y una Descripción del proyecto para su proyecto y, a continuación, seleccione Añadir activo al proyecto.



The screenshot shows a dialog box titled "Create new project" with a close button (X) in the top right corner. It contains two input fields: "Project name" with the text "Wind Farm 1" and "Project description" with the text "A project that contains dashboards for wind farm #1.". Below each field is a character limit note: "The project name can have up to 256 characters." and "The project description can have up to 2048 characters.". At the bottom right, there are three buttons: "Cancel", "Previous", and "Add asset to project". The "Add asset to project" button is highlighted with a red oval.

Se abrirá la página de su nuevo proyecto.

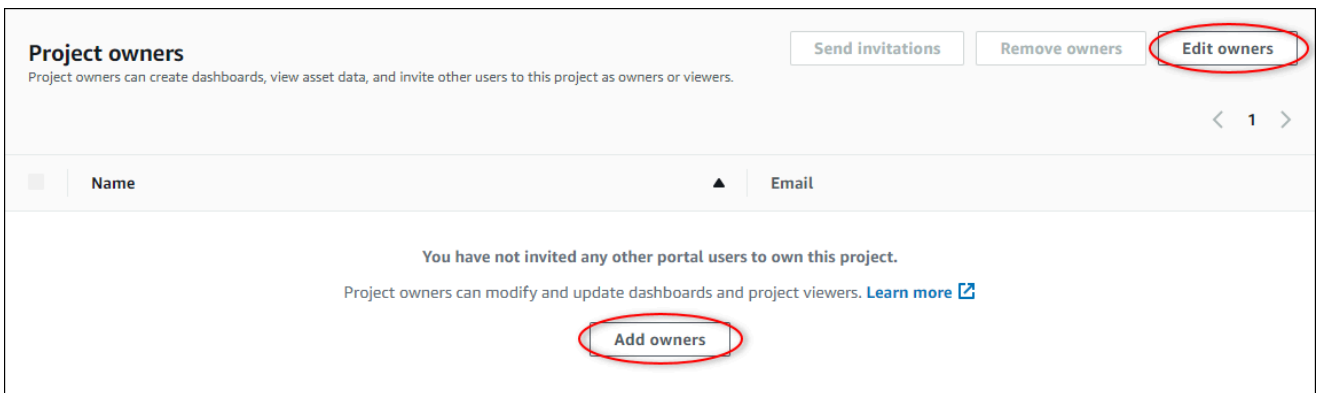
6. En la página del proyecto, puede añadir usuarios del portal como propietarios u observadores de este proyecto.

Note

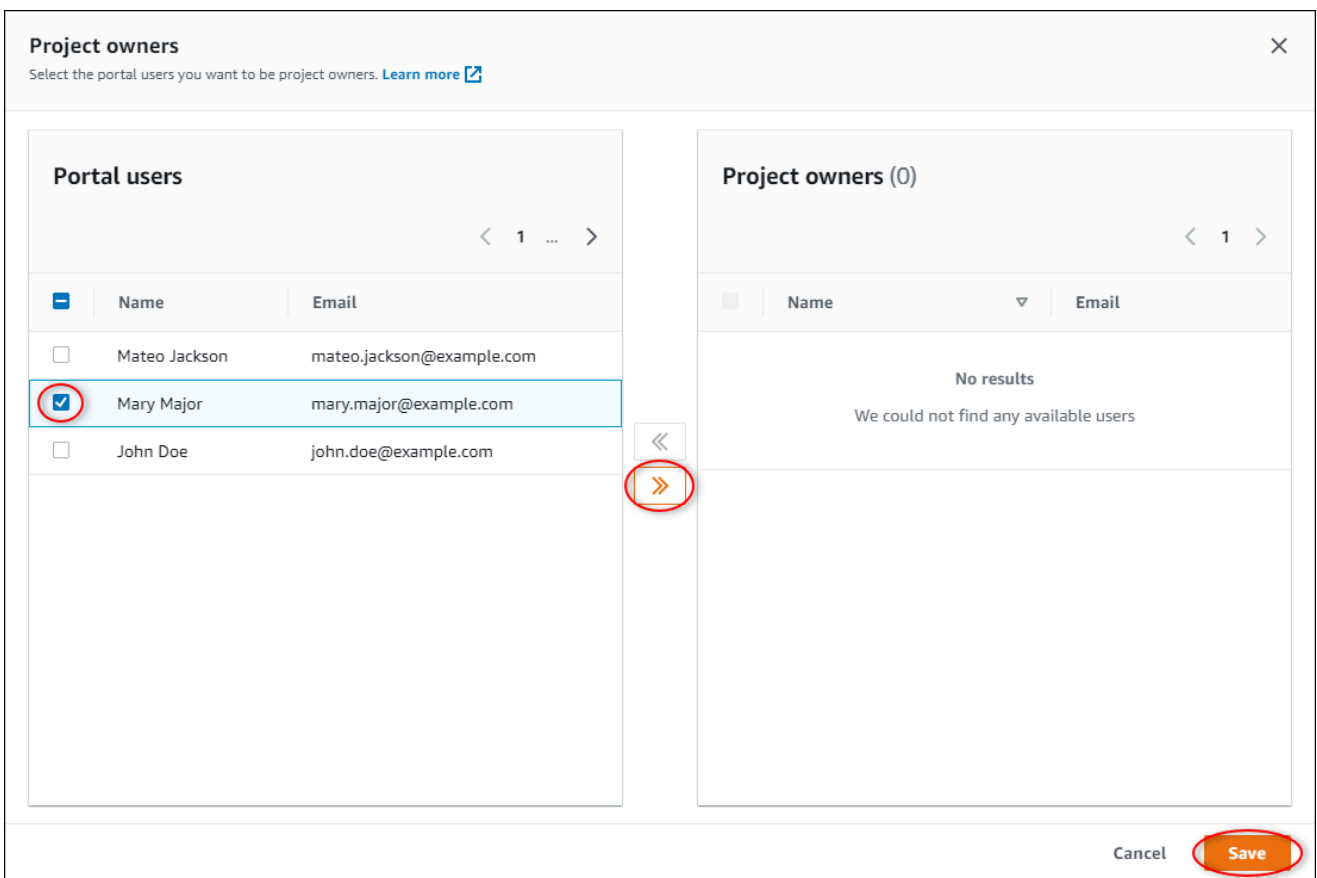
Si no ha iniciado sesión en su cuenta de AWS Organizations administración, es posible que no tenga usuarios del portal para asignar a este proyecto, por lo que puede omitir este paso.

En esta página, haga lo siguiente:

- a. En Propietarios del proyecto, elija Añadir propietarios o Editar usuarios.



- b. Elija el usuario que se va a añadir como propietario del proyecto (por ejemplo, Mary Major) y, a continuación, elija el icono >>.



- c. Seleccione Guardar.

Su usuario Mary Major del Centro de identidades de IAM puede iniciar sesión en este portal para editar los paneles de control de este proyecto y compartirlo con otros usuarios de este portal.

- d. En Observadores del proyecto, elija Añadir observadores o Editar usuarios.

- e. Elija el usuario que desee añadir como observador del proyecto (por ejemplo, Mateo Jackson) y, a continuación, seleccione el icono >>.
- f. Seleccione Save.

Su usuario Mateo Jackson del Centro de identidades de IAM puede acceder a este portal para ver, pero no editar, los paneles de control del proyecto del parque eólico.

Paso 4: creación de un panel para visualizar datos de parques eólicos

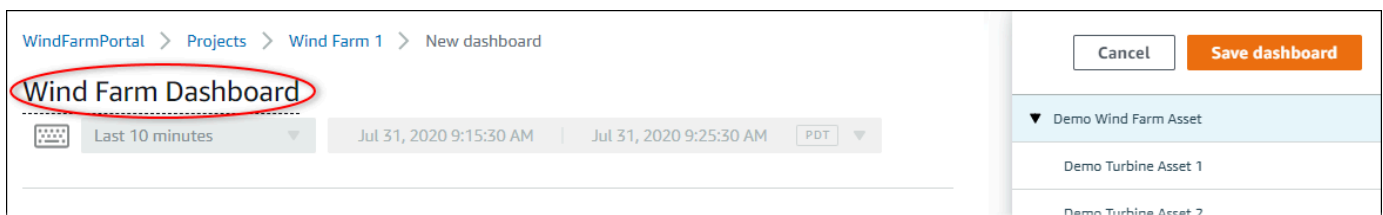
En este procedimiento, creará paneles para visualizar los datos de parques eólicos de demostración. Los paneles contienen visualizaciones personalizables de los datos de activos del proyecto. Cada visualización puede tener un tipo diferente, como un gráfico de líneas, un gráfico de barras o una visualización de indicador de rendimiento clave (KPI). Puede elegir el tipo de visualización que mejor se adapte a sus datos. Los propietarios de proyectos pueden editar paneles, mientras que los observadores de proyectos solo pueden verlos para obtener información.

Para crear un panel con visualizaciones

1. En la página de su nuevo proyecto, seleccione Crear panel de control para crear un panel de control y abrir su página de edición.

En la página de edición de un panel, puede arrastrar las propiedades de los activos desde la jerarquía de activos hasta el panel para crear visualizaciones. A continuación, puede editar el título, los títulos de leyenda, el tipo, el tamaño y la ubicación de cada visualización en el panel.

2. Introduzca un nombre para su panel de control.



3. Arrastre Total Average Power desde el Demo Wind Farm Asset hasta el panel para crear una visualización.

WindFarmPortal > Projects > Wind Farm 1 > New dashboard

Wind Farm Dashboard

Last 10 minutes | Jul 31, 2020 9:15:30 AM | Jul 31, 2020 9:25:30 AM | PDT

Total Average Power 24038 Watts

Properties for "Demo Wind Farm Asset"

Code 300

Total Overdrive State Time 0 seconds

4. Seleccione Demo Turbine Asset 1 para mostrar las propiedades de ese activo y, a continuación, arrastre Wind Speed al panel de control para crear una visualización de velocidad del viento.

WindFarmPortal > Projects > Wind Farm 1 > New dashboard

Wind Farm Dashboard

Last 10 minutes | Jul 31, 2020 9:15:30 AM | Jul 31, 2020 9:25:30 AM | PDT

Total Average Po...

26,000
25,500
25,000
24,500
24,000
23,500
23,000
22,500
22,000

09:20 09:25

— Total Average Power (Demo Wind Farm Asset)
23420 Watts

Wind Speed 14.753 m/s

Cancel Save dashboard

▼ Demo Wind Farm Asset

- Demo Turbine Asset 1
- Demo Turbine Asset 2
- Demo Turbine Asset 3
- Demo Turbine Asset 4

Properties for "Demo Turbine Asset 1"

Overdrive State 0

Overdrive State Time 0 Seconds

RotationsPerMinute 27.143 RPM

RotationsPerSecond 4.524e-1 RPS

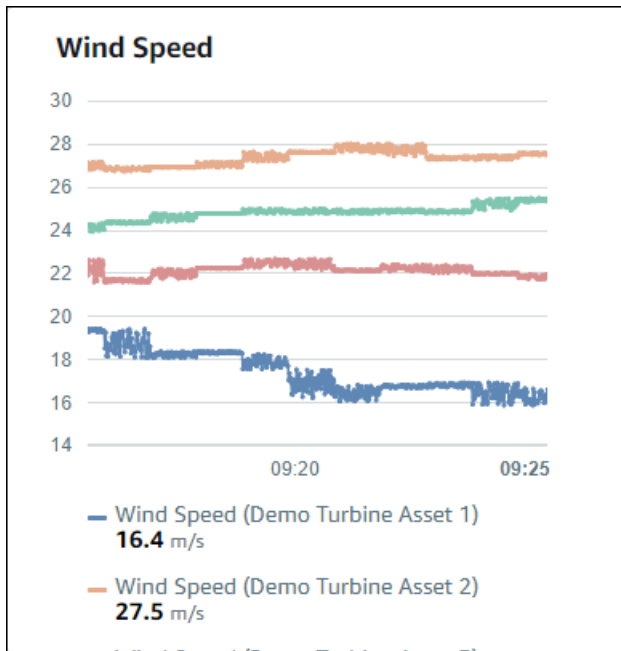
Torque (KiloNewton Meter) 2.5261 kNm

Torque (Newton Meter) 2526.1 Nm

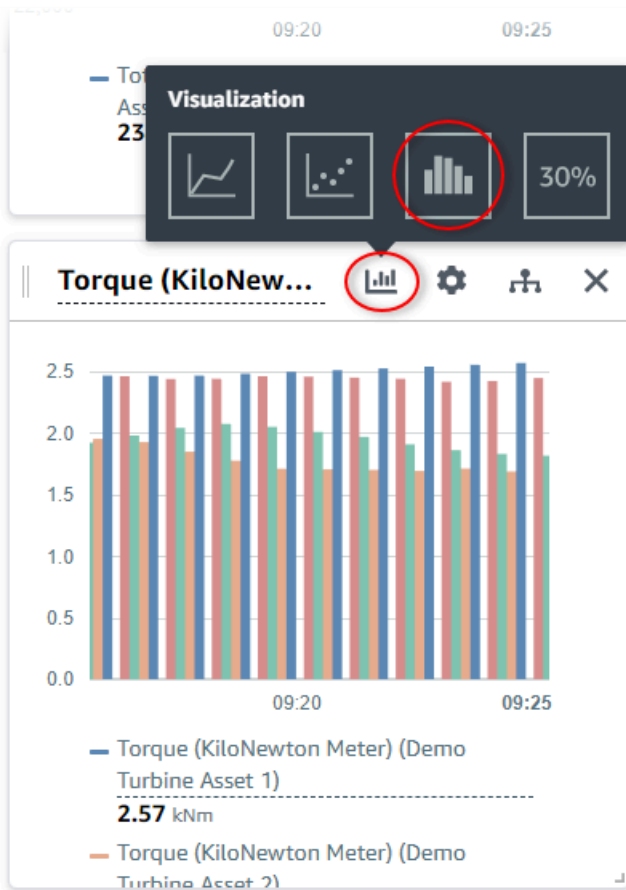
Wind Direction 7.4587 Degrees

5. Añada Wind Speed a la nueva visualización de la velocidad del viento para cada Demo Turbine Asset 2, 3 y 4 (en ese orden).

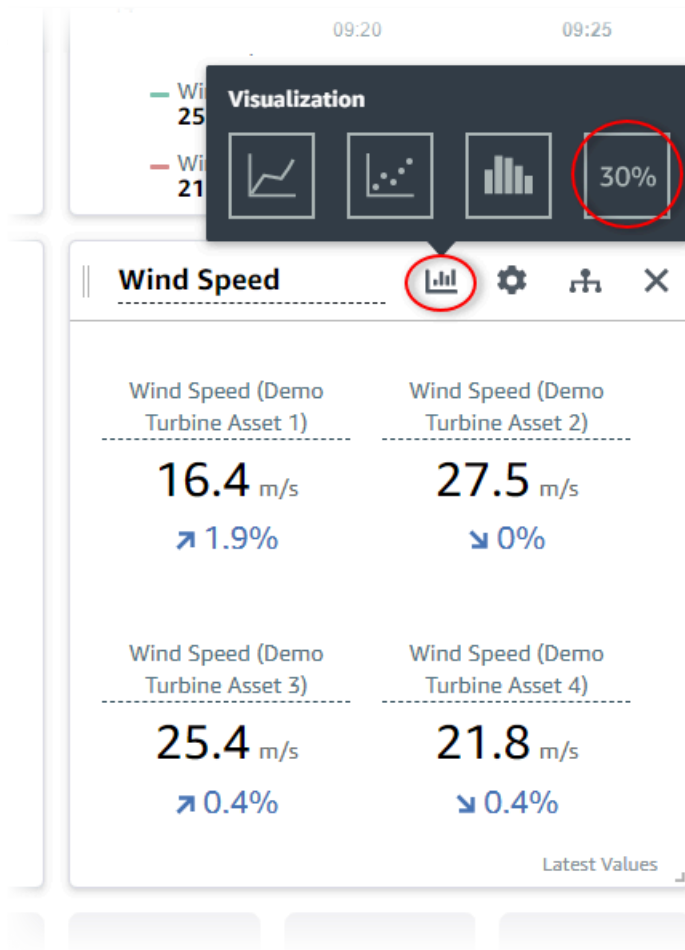
Su visualización Wind Speed debería ser similar a la siguiente captura de pantalla.



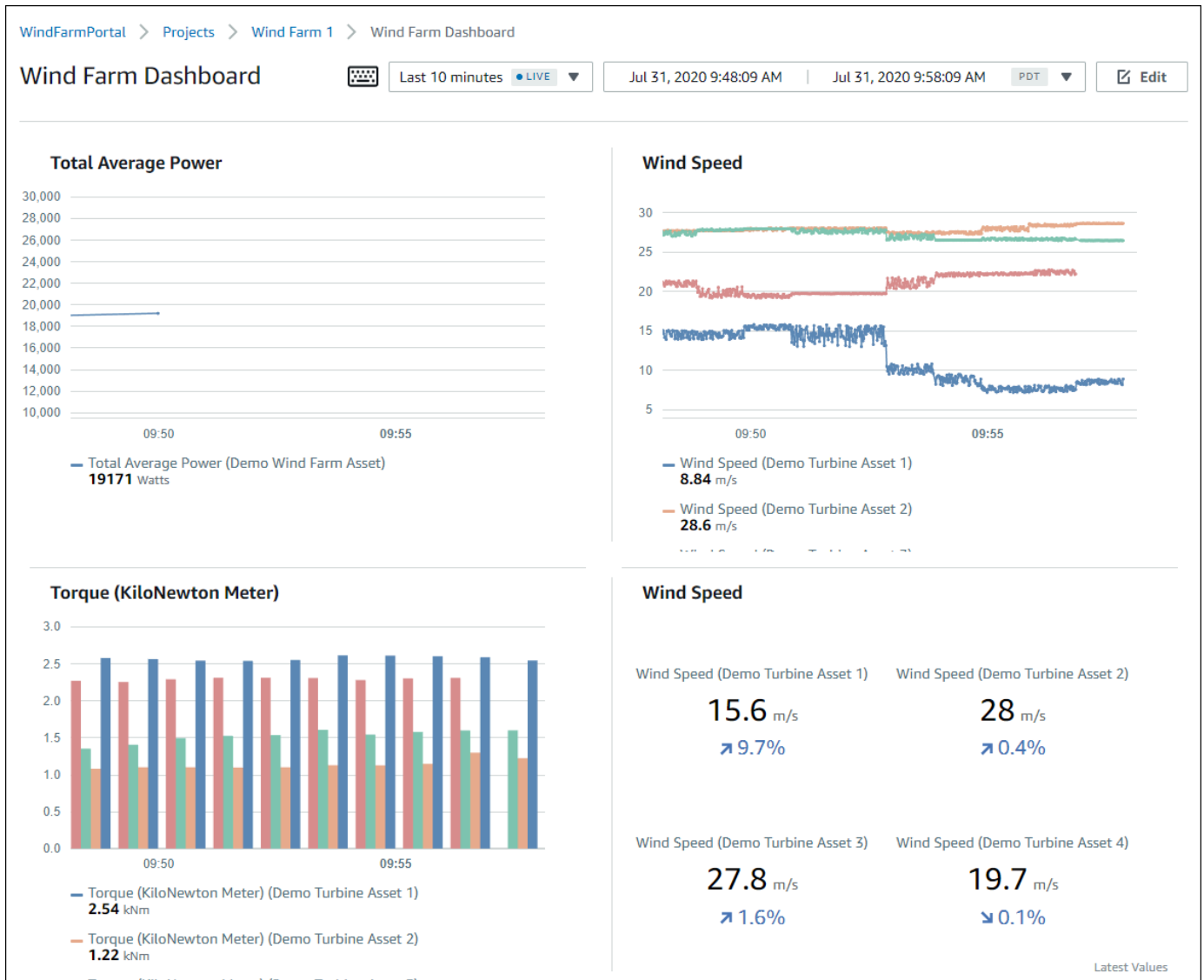
6. Repita los pasos 4 y 5 para las propiedades de Torque (KiloNewton Meter) de las turbinas eólicas a fin de crear una visualización de par de las turbinas eólicas.
7. Seleccione el icono de tipo de visualización para la visualización de Torque (KiloNewton Meter) y, a continuación, el icono de gráfico de barras.



8. Repita los pasos 4 y 5 para las propiedades de Wind Direction de las turbinas eólicas a fin de crear una visualización de dirección del viento.
9. Seleccione el icono de tipo de visualización para la visualización de Wind Direction y, a continuación, el icono de gráfico de KPI (30%).



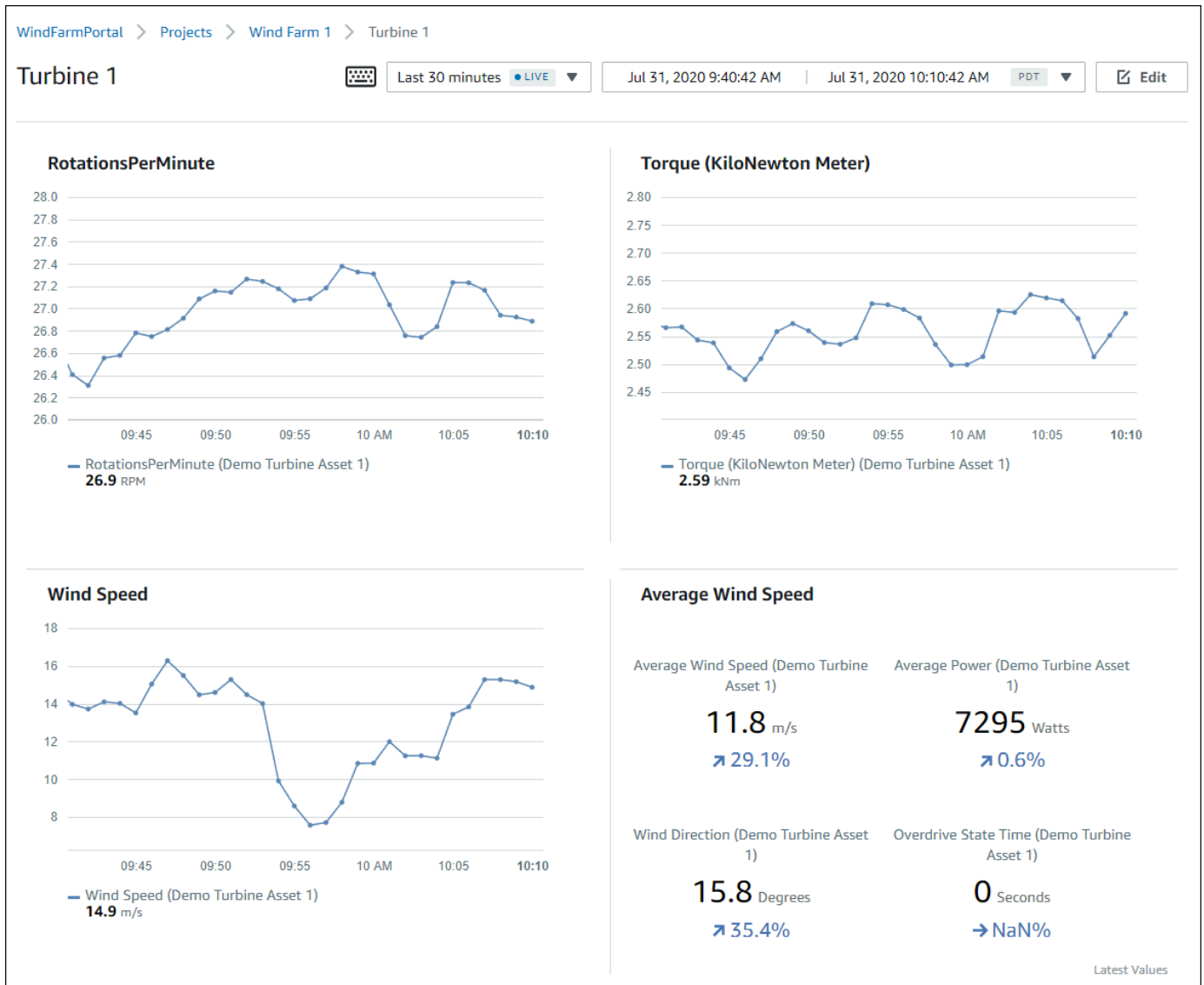
10. (Opcional) Realice otros cambios en el título, los títulos de leyenda, el tipo, el tamaño y la ubicación de cada visualización según sea necesario.
 11. Seleccione Guardar panel de control en la esquina superior derecha para guardar su panel.
- El panel debe tener un aspecto similar a la siguiente captura de pantalla.



12. (Opcional) Cree un panel adicional para cada activo de turbina eólica.

Como práctica recomendada, se recomienda crear un panel para cada activo de modo que los observadores del proyecto puedan investigar cualquier problema con cada activo individual. Solo puede añadir hasta 5 activos a cada visualización, por lo que debe crear varios paneles para los activos jerárquicos en muchos escenarios.

Un panel para una turbina eólica de demostración podría tener un aspecto similar a la siguiente captura de pantalla.



- (Opcional) Cambie la línea temporal o seleccione puntos de datos en una visualización para explorar los datos del panel. Para obtener más información, consulte [Visualización de paneles de control](#) en la Guía de la aplicación AWS IoT SiteWise Monitor .

Paso 5: exploración del portal

En este procedimiento puede explorar el portal como usuario con menos permisos que un administrador del portal de AWS IoT SiteWise .

Para explorar el portal y finalizar el tutorial

- (Opcional) Si ha añadido otros usuarios al proyecto como propietarios u observadores, puede iniciar sesión en el portal como estos usuarios. Esto le permite explorar el portal como un usuario con menos permisos que un administrador del portal.

Important

Se le cobra por cada usuario que inicie sesión en el portal. Para obtener más información, consulte [AWS IoT SiteWise Precios](#).

Para explorar el portal como otros usuarios, haga lo siguiente:

- a. Seleccione Cerrar sesión en la parte inferior izquierda del portal para salir de la aplicación web.
- b. Seleccione Cerrar sesión en la parte superior derecha del portal de la aplicación del Centro de identidades de IAM para cerrar sesión como usuario de IAM.
- c. Inicie sesión en el portal como el usuario de Centro de identidades de IAM que asignó como propietario del proyecto u observador del proyecto. Para obtener más información, consulte [Paso 2: inicio de sesión en un portal](#).

Ha completado este tutorial. Cuando termine de explorar su parque eólico de demostración en SiteWise Monitor, siga el siguiente procedimiento para limpiar sus recursos.

Paso 6: limpieza de los recursos después del tutorial

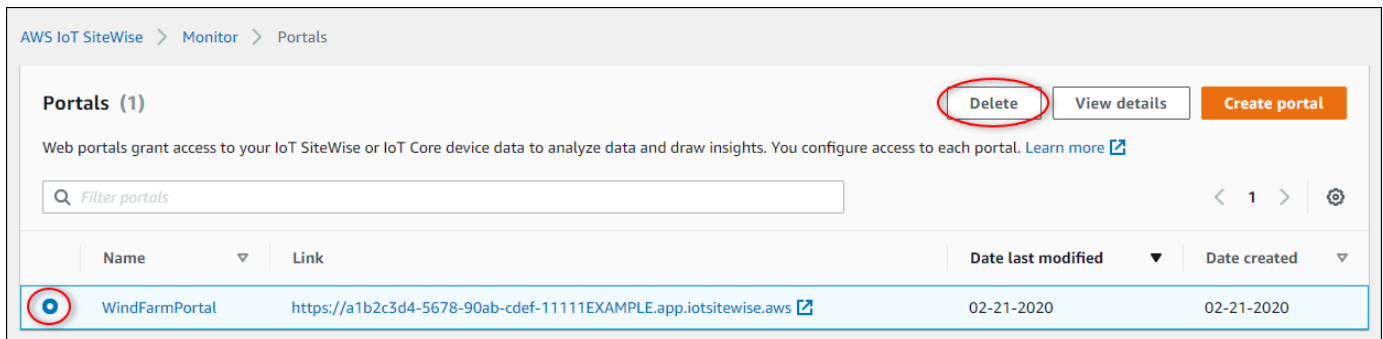
Una vez que complete el tutorial, puede sanear sus recursos. No se le cobrará por AWS IoT SiteWise si los usuarios no inician sesión en el portal, pero puede eliminar este y los usuarios de Directorio de AWS IAM Identity Center . Sus activos de parque eólico de demostración se eliminan al final de la duración que eligió al crear la demostración, o puede eliminar la demostración manualmente. Para obtener más información, consulte [Elimine la AWS IoT SiteWise demostración](#).

Utilice los siguientes procedimientos para eliminar su portal y los usuarios del Centro de identidades de IAM.

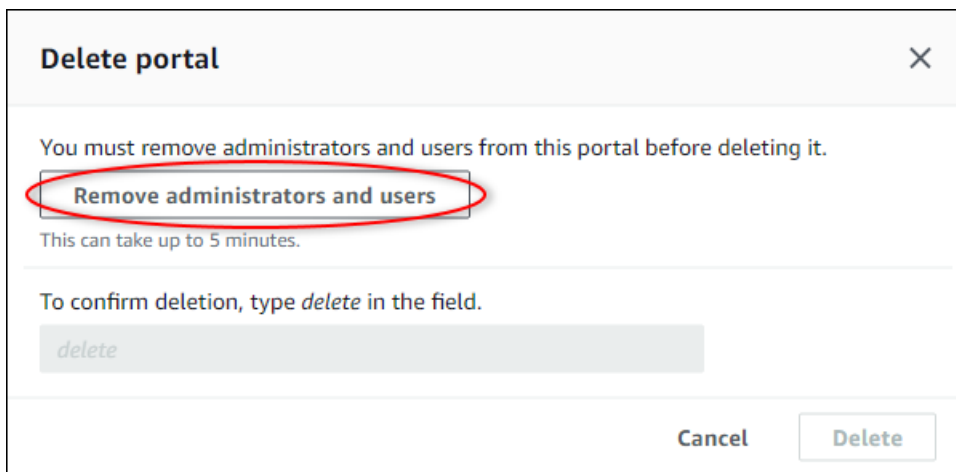
Para eliminar un portal

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación izquierdo, elija Portales.
3. Elija su portal y WindFarmPortal, a continuación, elija Eliminar.

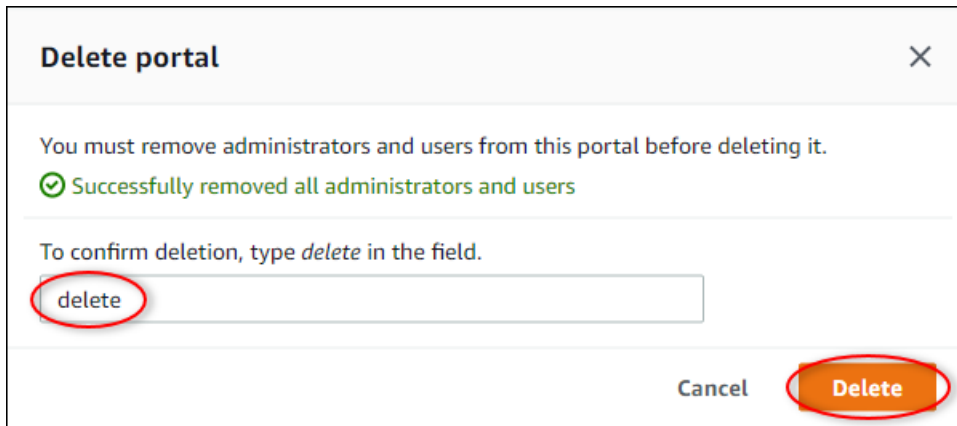
Al eliminar un portal o proyecto, los activos asociados a los proyectos eliminados no se ven afectados.



4. En el cuadro de diálogo Eliminar portal, seleccione Eliminar administradores y usuarios.

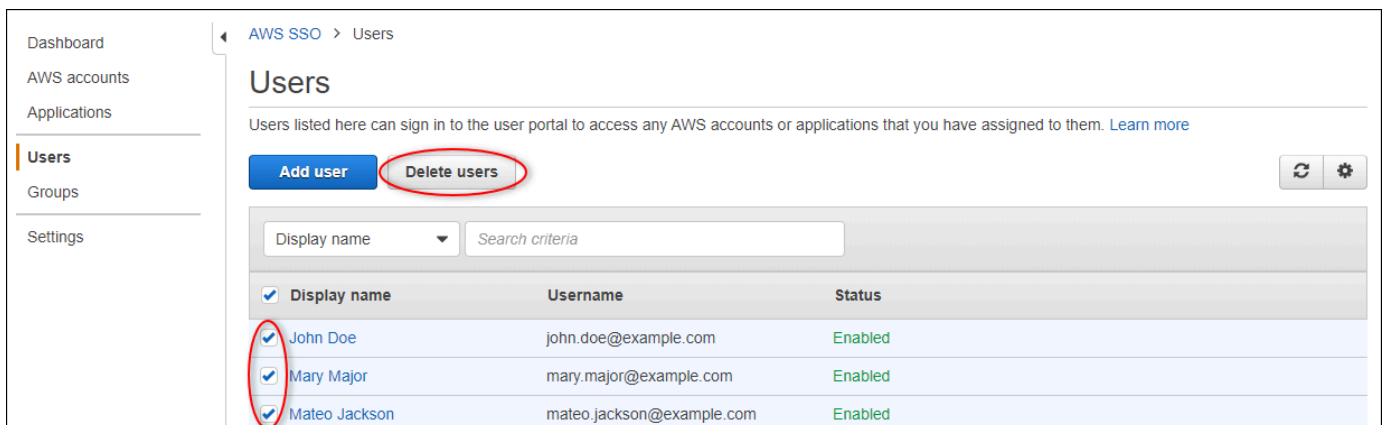


5. Escriba **delete** para confirmar la eliminación y, a continuación, elija Delete (Eliminar).



Para eliminar usuarios del Centro de identidades de IAM

1. Navegue hasta la [consola del Centro de identidades de IAM](#).
2. En el panel de navegación izquierdo, elija Usuarios.
3. Active la casilla de verificación de cada usuario que desee eliminar y, a continuación, elija Delete users (Eliminar usuarios).



4. En el cuadro de diálogo Eliminar usuarios, introduzca **DELETE** y, a continuación, seleccione Eliminar usuarios.

Delete users ✕

Deleting the following users will remove access to AWS accounts and applications.

This action cannot be undone.

Display name	Username
John Doe	john.doe@example.com
Mary Major	mary.major@example.com
Mateo Jackson	mateo.jackson@example.com

Are you sure you want to delete these users?
Type 'DELETE' to confirm

Cancel Delete users

Publicar actualizaciones de valor de propiedad en Amazon DynamoDB

En este tutorial, se presenta una forma práctica de almacenar los datos mediante [Amazon DynamoDB](#), lo que facilita el acceso a los datos históricos de los activos sin tener que consultar repetidamente la API. AWS IoT SiteWise Después de completar este tutorial, puede crear software personalizado que consuma los datos de sus activos, como un mapa en vivo de la velocidad y la dirección del viento en todo un parque eólico. Si desea supervisar y visualizar los datos sin implementar una solución de software personalizada, consulte [Supervise los datos con AWS IoT SiteWise Monitor](#).

En este tutorial, se basa en la AWS IoT SiteWise demostración que proporciona un conjunto de datos de muestra para un parque eólico. Puede configurar actualizaciones de valores de propiedad desde la demostración del parque eólico para enviar datos, a través de las reglas de AWS IoT Core, a una tabla de DynamoDB que cree. Cuando habilita las actualizaciones del valor de la propiedad, AWS IoT SiteWise envía sus datos a AWS IoT Core mensajes MQTT. A continuación, defina las reglas

AWS IoT principales que realicen acciones, como la acción de DynamoDB, en función del contenido de esos mensajes. Para obtener más información, consulte [Interactúa con otros AWS servicios](#).

Temas

- [Requisitos previos](#)
- [Paso 1: Configure AWS IoT SiteWise para publicar las actualizaciones del valor de la propiedad](#)
- [Paso 2: Crea una regla en Core AWS IoT](#)
- [Paso 3: Configurar la acción de la regla de DynamoDB](#)
- [Paso 4: Explore los datos en DynamoDB](#)
- [Eliminar recursos](#)
- [Recursos adicionales](#)

Requisitos previos

Necesitará lo siguiente para completar este tutorial:

- Una Cuenta de AWS Si no dispone de una, consulte [Configura una AWS cuenta](#).
- Un ordenador de desarrollo que ejecute Windows, macOS, Linux o Unix para acceder al Consola de administración de AWS. Para obtener más información, consulte [¿Qué es Consola de administración de AWS?](#)
- Un usuario AWS Identity and Access Management (IAM) con permisos de administrador. Para obtener instrucciones detalladas, consulte [the section called “¿Cómo AWS IoT SiteWise funciona con IAM”](#).
- Una AWS IoT SiteWise demostración en ejecución. Para obtener más información, consulte [Usa la AWS IoT SiteWise demo](#).

Note

Este tutorial requiere el uso de los recursos creados en el [Usa la AWS IoT SiteWise demo](#). Debe completarlo antes de continuar con este tutorial.

⚠ Important

Conserve todos los recursos de demostración hasta completar este tutorial. Eliminar cualquier componente podría interrumpir la funcionalidad de la demostración y afectar a su capacidad para completar el tutorial.

Paso 1: Configure AWS IoT SiteWise para publicar las actualizaciones del valor de la propiedad

En este procedimiento, se habilitan las notificaciones de valores de propiedad en las propiedades Wind Speed de los activos de la turbina de demostración. Tras activar las notificaciones del valor de las propiedades, AWS IoT SiteWise publica cada actualización de valores en un mensaje MQTT en AWS IoT Core.

Para habilitar las notificaciones de actualización de valores de propiedad en propiedades de activos

1. Inicie sesión en la [consola de AWS IoT SiteWise](#).
2. Revise los [AWS IoT SiteWise puntos finales y las cuotas](#) compatibles y cambie de AWS región si AWS IoT SiteWise es necesario. Cambie a la región en la que esté realizando la AWS IoT SiteWise demostración.
3. En el panel de navegación izquierdo, elija Assets (activos).
4. Elija la flecha situada junto a Demo Wind Farm Asset para ampliar la jerarquía del activo del parque eólico.
5. Elija una turbina de demostración y elija Edit (Editar).
6. Elija Measurements (Medidas).
7. Actualice el estado de la notificación MQTT de la Wind Speed propiedad a ACTIVO.
8. Elija Guardar en la parte inferior de la página.
9. Repita los pasos 5 a 7 para cada activo de turbina de demostración.
10. Elija una turbina de demostración (por ejemplo, Demo Turbine Asset 1).
11. Elija Measurements (Medidas).
12. Elija el icono copiar junto a la propiedad Wind Speed para copiar el tema de notificación en el portapapeles. Guarde el tema de notificación para utilizarlo más adelante en este tutorial. Solo necesita registrar el tema de notificación desde una turbina.

El tema de notificación debe ser similar al siguiente ejemplo.

```
$aws/sitewise/asset-models/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE/  
assets/a1b2c3d4-5678-90ab-cdef-22222EXAMPLE/properties/a1b2c3d4-5678-90ab-  
cdef-33333EXAMPLE
```

Paso 2: Crea una regla en Core AWS IoT

En este paso, cree una regla en AWS IoT Core que analice los mensajes de notificación del valor de la propiedad e inserte los datos en una tabla de Amazon DynamoDB. AWS IoT Las reglas básicas analizan los mensajes de MQTT y realizan acciones en función del contenido y el tema de cada mensaje. A continuación, cree una regla con una acción de DynamoDB para insertar datos en una tabla de DynamoDB que cree como parte de este tutorial.

Para crear una regla con una acción de DynamoDB

1. Vaya a la [consola de AWS IoT](#).
2. En el panel de navegación izquierdo, selecciona Enrutamiento de mensajes y, a continuación, Reglas.
3. Seleccione Creación de regla.
4. En Especificar las propiedades de la regla, introduzca un nombre y una descripción para la regla.
5. Busque el tema de notificación que guardó anteriormente en este tutorial.

```
$aws/sitewise/asset-models/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE/  
assets/a1b2c3d4-5678-90ab-cdef-22222EXAMPLE/properties/a1b2c3d4-5678-90ab-  
cdef-33333EXAMPLE
```

Sustituya el ID del activo (el ID después de assets/) del tema por un +. Esto selecciona la propiedad de velocidad del viento para todos los activos de aerogeneradores de demostración. El filtro de temas + acepta todos los nodos de un solo nivel de un tema. El tema debe ser similar al siguiente ejemplo.

```
$aws/sitewise/asset-models/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE/assets/+/  
properties/a1b2c3d4-5678-90ab-cdef-33333EXAMPLE
```

- Introduzca la siguiente instrucción de consulta de reglas. Reemplace el tema de la sección FROM por el tema de notificación.

```
SELECT
  payload.assetId AS asset,
  (SELECT VALUE (value.doubleValue) FROM payload.values) AS windspeed,
  timestamp() AS timestamp
FROM
  '$aws/sitewise/asset-models/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE/assets/+/
properties/a1b2c3d4-5678-90ab-cdef-33333EXAMPLE'
WHERE
  type = 'PropertyValueUpdate'
```

- En Acciones de la regla, vaya a la Acción 1.
- En la página Seleccione una acción, elija Dynamo. DBv2 Esto divide el mensaje en varias columnas de una tabla de DynamoDB.
- En Nombre de la tabla, elija Crear tabla nueva. Cree una tabla de Amazon DynamoDB para recibir los datos de velocidad del viento de la acción de la regla.
- En Nombre de tabla en la consola de [DynamoDB](#), introduzca un nombre para la tabla.
- Para la clave de partición, haga lo siguiente:
 - Escriba **timestamp** como clave de partición.
 - Seleccione el tipo Number (Número) .
 - Marque la casilla Add sort key (Añadir clave de ordenación).
 - Escriba **asset** como clave de ordenación y deje el tipo de clave de ordenación predeterminado de String (Cadena).
- Seleccione Create table (Creación de tabla).
- Vuelva a la pestaña con la página Configure action (Configurar acción) .
- En la página de acción Adjuntar regla, actualice la lista de nombres de tabla y elija la nueva tabla de DynamoDB que creó en el paso anterior.

Paso 3: Configurar la acción de la regla de DynamoDB

En este paso, configure la acción de la regla de Amazon DynamoDB para insertar datos de las actualizaciones de valores de propiedades en la nueva tabla de DynamoDB.

Para configurar la acción de regla de DynamoDB

1. Seleccione Crear rol para crear un rol de IAM que otorgue a AWS IoT Core acceso para realizar la acción de regla.
2. Introduzca el nombre de un rol, por ejemplo, WindSpeedDataRole. Elija Crear rol.
3. Elija Siguiente.
4. Elija Crear en la parte inferior de la página para terminar de crear la regla.

Los datos de los activos de demostración deberían comenzar a aparecer en la tabla de DynamoDB.

Paso 4: Explore los datos en DynamoDB

En este paso, explore los datos de velocidad del viento de los activos de demostración en su nueva tabla de Amazon DynamoDB.

Para explorar datos de activos en DynamoDB.

1. Vuelva a la pestaña con la tabla de DynamoDB abierta.
2. En la tabla que creó anteriormente, seleccione la pestaña Explorar los elementos de la tabla para ver los datos de la tabla. Actualice la página si no ve filas en la tabla. Si las filas no aparecen después de unos minutos, consulte [Solución de problemas de una regla \(DynamoDB\)](#).
3. En una fila de la tabla, elija el icono de edición para expandir los datos.
4. Seleccione la flecha situada junto a la estructura de windspeed para ampliar la lista de puntos de datos de velocidad del viento. Cada lista refleja un lote de puntos de datos de velocidad del viento enviados AWS IoT SiteWise por la demostración del parque eólico. Es posible que desee un formato de datos diferente si configura una acción de regla para su propio uso. Para obtener más información, consulte [Consulte las notificaciones de propiedades de activos en AWS IoT SiteWise](#).

Ahora que ha completado el tutorial, puede deshabilitar o eliminar la regla y eliminar la tabla de DynamoDB para evitar incurrir en cargos adicionales. Para limpiar los recursos, consulte [Eliminar recursos](#).

También puede aprender a crear aplicaciones personalizadas para consumir y visualizar estos datos. Para ver un tutorial guiado sobre la visualización de AWS IoT SiteWise datos, consulte [Visualice y comparta datos en Grafana](#).

Eliminar recursos

Después de completar el tutorial, limpie los recursos para evitar incurrir en cargos adicionales.

Para eliminar la demostración AWS IoT SiteWise

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En la esquina superior derecha de la página, selecciona Eliminar demostración.
3. En el cuadro de diálogo de confirmación, escriba **DELETE** y, a continuación, elija Delete (Eliminar).

Para obtener más información, consulte [Elimine la AWS IoT SiteWise demostración](#).

Utilice los siguientes procedimientos para deshabilitar las notificaciones de actualización del valor de las propiedades (si no ha eliminado la demostración), deshabilitar o eliminar la AWS IoT regla y eliminar la tabla de DynamoDB.

Para deshabilitar las notificaciones de actualización de valores de propiedad en propiedades de activos

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación izquierdo, elija Assets (activos).
3. Elija la flecha situada junto a Demo Wind Farm Asset para ampliar la jerarquía del activo del parque eólico.
4. Elija una turbina de demostración y elija Edit (Editar).
5. Actualice el estado de notificación de la Wind Speed propiedad a INACTIVO.
6. Elija Save asset (Guardar activo) en la parte inferior de la página.
7. Repita los pasos 4 a 6 para cada activo de turbina de demostración.

Para deshabilitar o eliminar una regla en AWS IoT Core

1. Vaya a la [consola de AWS IoT](#).
2. En el panel de navegación de la izquierda, elija Redirección de mensajes y, a continuación, seleccione Reglas.
3. Seleccione su regla y seleccione Eliminar.

4. En el cuadro de diálogo de confirmación, ingrese el nombre de la regla y, a continuación, elija Eliminar.

Para eliminar una tabla de DynamoDB

1. Navegue hasta la [consola de DynamoDB](#).
2. En el panel de navegación izquierdo, elija Tables (Tablas).
3. Elija la tabla que creó anteriormente, por ejemplo, WindSpeedData.
4. Elija Eliminar.
5. En el cuadro de diálogo de confirmación, introduzca **confirm** para eliminar la tabla.

Recursos adicionales

Para obtener más información sobre cómo trabajar con DynamoDB y supervisar los datos, consulte los siguientes recursos:

- [Supervisión de las métricas incluidas CloudWatch en](#) la Guía para desarrolladores de DynamoDB
- [Prácticas recomendadas para diseñar y utilizar claves de partición de forma eficaz en](#) la Guía para desarrolladores de DynamoDB
- [Reglas incluidas AWS IoT en la Guía para](#) desarrolladores AWS IoT
- [Visualice y comparta datos en Grafana](#)

Ingiera datos para AWS IoT SiteWise

AWS IoT SiteWise está diseñado para recopilar y correlacionar de manera eficiente los datos industriales con los activos correspondientes, que representan varios aspectos de las operaciones industriales. Esta documentación se centra en los aspectos prácticos de la ingesta de datos y ofrece varios métodos adaptados a diversos casos de uso industrial. AWS IoT SiteWise Para obtener instrucciones sobre cómo crear su operación industrial virtual, consulte [Modelado de activos industriales](#).

Puede enviar datos industriales a AWS IoT SiteWise través de cualquiera de las siguientes opciones:

- AWS IoT SiteWise Edge: utilice la [puerta de enlace SiteWise Edge](#) como intermediario entre AWS IoT SiteWise y sus servidores de datos. AWS IoT SiteWise proporciona AWS IoT Greengrass componentes que puede implementar en cualquier plataforma que se pueda ejecutar AWS IoT Greengrass para configurar una puerta de enlace SiteWise Edge. Esta opción admite el enlace con el protocolo del servidor [OPC UA](#).
- AWS IoT SiteWise API: utilice la [AWS IoT SiteWise API](#) para cargar datos desde cualquier otra fuente. Utilice nuestra [BatchPutAssetPropertyValueAPI](#) de streaming para la ingesta en cuestión de segundos, o la [CreateBulkImportJobAPI](#) orientada a lotes para facilitar la ingesta rentable en lotes más grandes.
- AWS IoT Reglas básicas: utilice las [reglas AWS IoT básicas](#) para cargar datos de los mensajes MQTT publicados por una cosa u otro servicio. AWS IoT AWS
- AWS IoT Events acciones: utilice [AWS IoT Events acciones](#) activadas por eventos específicos en. AWS IoT Events Este método es adecuado para situaciones en las que la carga de datos esté vinculada a eventos.
- AWS IoT Greengrass administrador de transmisiones: utilice el [administrador de AWS IoT Greengrass transmisiones](#) para cargar datos de fuentes de datos locales mediante un dispositivo perimetral. Esta opción es válida para situaciones en las que los datos se originen en ubicaciones en las instalaciones o periféricas.

Estos métodos ofrecen una gama de soluciones para administrar datos de diferentes orígenes. Profundice en los detalles de cada opción para obtener una comprensión completa de las capacidades de ingesta de datos que ofrece. AWS IoT SiteWise

Gestione los flujos de datos para AWS IoT SiteWise

Un flujo de datos es el recurso que contiene datos históricos de series temporales. Cada flujo de datos se identifica con un alias único, lo que facilita el seguimiento del origen de cada dato. Los flujos de datos se crean automáticamente AWS IoT SiteWise cuando se reciben los primeros datos de series temporales. Si los datos de la primera serie temporal se identifican con un alias, AWS IoT SiteWise crea un nuevo flujo de datos con ese alias, siempre que a ninguna propiedad de activo se le haya asignado ese alias. Como alternativa, si los datos de la primera serie temporal se identifican con un identificador de activo y un identificador de propiedad, AWS IoT SiteWise crea un nuevo flujo de datos y lo asocia a la propiedad del activo.

Hay dos formas de asignar un alias a una propiedad de un activo. El método utilizado depende de si los datos se envían AWS IoT SiteWise primero o si se crea primero un activo.

- Si los datos se envían AWS IoT SiteWise primero, se crea automáticamente un flujo de datos con el alias asignado. Cuando el activo se cree más adelante, utilice la [AssociateTimeSeriesToAssetProperty](#) API para asociar el flujo de datos y su alias a la propiedad del activo.
- Si primero se crea un activo, utilice la [UpdateAssetProperty](#) API para asignar un alias a una propiedad del activo. Cuando los datos se envíen posteriormente a AWS IoT SiteWise, el flujo de datos se crea automáticamente y se asocia a la propiedad del activo.

Actualmente, solo puede asociar flujos de datos con mediciones. Las Mediciones son un tipo de propiedad de activo que representan flujos de datos de sensores sin procesar de los dispositivos, como valores de temperatura con marca de tiempo o valores de revoluciones por minuto (RPM) con marca de tiempo.

Cuando estas mediciones definen métricas o transformaciones, los datos entrantes desencadenan cálculos específicos. Es importante tener en cuenta que una propiedad de activo solo se puede enlazar a un flujo de datos a la vez.

AWS IoT SiteWise utiliza `TimeSeries` el recurso Amazon Resource Name (ARN) para determinar los cargos de almacenamiento. Para obtener más información, consulte [AWS IoT SiteWise Precios](#).

En las siguientes secciones, se muestra cómo utilizar la AWS IoT SiteWise consola o la API para gestionar los flujos de datos.

Temas

- [Configuración de permisos y ajustes](#)
- [Asocie un flujo de datos a una propiedad de un activo](#)
- [Desasocie un flujo de datos de una propiedad de activo](#)
- [Eliminación de un flujo de datos](#)
- [Actualice el alias de una propiedad de un activo](#)
- [Escenarios habituales](#)

Configuración de permisos y ajustes

Los flujos de datos se crean automáticamente AWS IoT SiteWise cuando se reciben los primeros datos de series temporales. Si los datos ingeridos no están asociados a una propiedad de activo, AWS IoT SiteWise crea un nuevo flujo de datos disociado que se puede configurar para asociarse a una propiedad de activo. Configure el control de acceso de la puerta de enlace a AWS IoT SiteWise la que se envían los datos mediante políticas de IAM para especificar el tipo de datos que se van a ingerir.

La siguiente política de IAM desactiva la ingesta de datos disociados de la puerta de enlace y, al mismo tiempo, permite la ingesta de datos a los flujos de datos asociados a una propiedad de activo:

Example Política de usuario de IAM que desactiva la ingesta disociada de datos desde la puerta de enlace

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowPutAssetPropertyValuesUsingAssetIdAndPropertyId",
      "Effect": "Allow",
      "Action": "iotsitewise:BatchPutAssetPropertyValue",
      "Resource": "arn:aws:iotsitewise:*:*:asset/*"
    },
    {
      "Sid": "AllowPutAssetPropertyValuesUsingAliasWithAssociatedAssetProperty",
      "Effect": "Allow",
      "Action": "iotsitewise:BatchPutAssetPropertyValue",
      "Resource": "arn:aws:iotsitewise:*:*:time-series/*",
    }
  ]
}
```

```

    "Condition": {
      "StringLikeIfExists": {
        "iotsitewise:isAssociatedWithAssetProperty": "true"
      }
    },
    {
      "Sid": "DenyPutAssetPropertyValuesUsingAliasWithNoAssociatedAssetProperty",
      "Effect": "Deny",
      "Action": "iotsitewise:BatchPutAssetPropertyValue",
      "Resource": "arn:aws:iotsitewise:*:*:time-series/*",
      "Condition": {
        "StringLikeIfExists": {
          "iotsitewise:isAssociatedWithAssetProperty": "false"
        }
      }
    }
  ]
}

```

Example Política de usuario de IAM que desactiva toda la ingesta de datos desde la puerta de enlace

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyPutAssetPropertyValues",
      "Effect": "Deny",
      "Action": "iotsitewise:BatchPutAssetPropertyValue",
      "Resource": [
        "arn:aws:iotsitewise:*:*:asset/*",
        "arn:aws:iotsitewise:*:*:time-series/*"
      ]
    }
  ]
}

```

Asocie un flujo de datos a una propiedad de un activo

Administre sus flujos de datos con Consola de AWS IoT SiteWise o AWS CLI.

Console

Utilice la AWS IoT SiteWise consola para gestionar sus flujos de datos.

Para administrar flujos de datos (consola)

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, seleccione Flujos de datos.
3. Elija un flujo de datos filtrando por el alias del flujo de datos o seleccionando Flujos de datos diasociados en el menú desplegable de filtros.
4. Seleccione el flujo de datos que desee actualizar. Puede seleccionar varios flujos de datos. Haga clic en Administrar flujos de datos en la esquina superior derecha.
5. Seleccione el flujo de datos que desee asociar en Actualizar asociaciones de flujos de datos y haga clic en el botón Elegir medición.
6. En la sección Elegir medición, busque la propiedad de medición de activos correspondiente. Seleccione la medida y, a continuación, haga clic en Elegir.
7. Realice los pasos 4 y 5 para otros flujos de datos seleccionados en el paso 3. Asigne propiedades de activos a todos los flujos de datos.
8. Elija Actualizar para confirmar los cambios. Aparece un cartel de confirmación de que la actualización se ha realizado correctamente.

AWS CLI

Para asociar un flujo de datos (identificado por su alias) a una propiedad de activo (identificada por su IDs), ejecute el siguiente comando:

```
aws iotsitewise associate-time-series-to-asset-property \  
  --alias <data-stream-alias> \  
  --assetId <asset-ID> \  
  --propertyId <property-ID>
```

Desasocie un flujo de datos de una propiedad de activo

Console

Utilice la AWS IoT SiteWise consola para desasociar el flujo de datos de una propiedad de activo.

Para desasociar los flujos de datos de una propiedad de activo (consola)

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, seleccione Flujos de datos.
3. Elija un flujo de datos filtrando por el alias del flujo de datos o seleccionando Flujos de datos asociados en el menú desplegable del filtro.
4. Seleccione el flujo de datos que desee desasociar. La columna de alias del flujo de datos debe contener un alias. Las columnas Nombre del activo y Nombre de la propiedad del activo deben contener los valores de la propiedad del activo a la que está asociado el flujo de datos. Puede seleccionar varios flujos de datos.
5. Haga clic en Administrar flujos de datos en la esquina superior derecha.
6. En la sección Actualizar asociaciones de flujos de datos, haga clic en la X de la columna Nombre de la medición. Debería aparecer un `submitted` estado en la columna Estado.
7. Seleccione Actualizar para confirmar los cambios. El flujo de datos ahora está disociado de la propiedad del activo y el alias se utiliza ahora para identificar el flujo de datos.

AWS CLI

Para desasociar un flujo de datos de una propiedad de activo (identificada por su ID s y su alias), ejecute el siguiente comando:

```
aws iotsitewise disassociate-time-series-from-asset-property \  
  --alias <asset-property-alias> \  
  --assetId <asset-ID> \  
  --propertyId <property-ID>
```

El flujo de datos ahora está disociado de la propiedad del activo y el alias se utiliza para identificar el flujo de datos. El alias ya no está asociado a la propiedad del activo, ya que ahora está asociado al flujo de datos.

Eliminación de un flujo de datos

Cuando se elimina una propiedad de un modelo de activos, AWS IoT SiteWise elimina las propiedades y sus flujos de datos de todos los activos gestionados por el modelo de activos. También elimina todas las propiedades y sus flujos de datos de un activo cuando se elimina el activo. Si se deben conservar los datos de un flujo de datos, se deben disociar de la propiedad del activo antes de eliminarlos.

Warning

Cuando se elimina una propiedad de un activo, también se elimina el flujo de datos asociado. Para conservar el flujo de datos, desasócielo primero de la propiedad del activo, antes de eliminar la propiedad del modelo de activo o eliminar el activo.

Console

Utilice la AWS IoT SiteWise consola para desasociar el flujo de datos de una propiedad de activo.

Para eliminar un flujo de datos (consola)

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, seleccione Flujos de datos.
3. Elija un flujo de datos filtrando el alias del flujo de datos.
4. Seleccione el flujo de datos que desee eliminar. Puede seleccionar varios flujos de datos.
5. Pulse el botón Eliminar para eliminar el flujo de datos.

AWS CLI

Utilice la [DeleteTimeSeries](#) API para eliminar un flujo de datos específico por su alias.

```
aws iotsitewise delete-time-series \  
  --alias <data-stream-alias>
```

Actualice el alias de una propiedad de un activo

Los alias deben ser únicos dentro de una AWS región. Esto incluye los alias de las propiedades de los activos y de los flujos de datos. No asigne un alias a una propiedad de un activo si otra propiedad o flujo de datos utiliza ese alias.

Console

Utilice la AWS IoT SiteWise consola para actualizar el alias de una propiedad de un activo.

Para actualizar el alias de una propiedad de un activo (consola)

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, elija Activos.
3. Seleccione el activo de la tabla.
4. Haga clic en el botón Edit (Editar).
5. Seleccione el tipo de propiedad en la tabla de propiedades.
6. Busque la propiedad y escriba el nuevo alias en el campo de texto del alias de la propiedad.
7. Haga clic en el botón Guardar para guardar los cambios.

AWS CLI

Para actualizar un alias en una propiedad de un activo, ejecute el siguiente comando:

```
aws iotsitewise update-asset-property \  
  --asset-id <asset-ID> \  
  --property-id <property-ID> \  
  --property-alias <asset-property-alias> \  
  --property-notification-state <ENABLED|DISABLED>
```

Note

Si las notificaciones de propiedades están habilitadas actualmente, se deben volver a proporcionar para garantizar que sigan activadas.

Escenarios habituales

Mover un flujo de datos

Para cambiar la asociación de un flujo de datos con otra propiedad del activo, primero desasocie el flujo de datos de la propiedad del activo actual. Al disociar un flujo de datos de una propiedad de activo, debe haber un alias asignado a esa propiedad de activo.

```
aws iotsitewise disassociate-time-series-from-asset-property \  
  --alias <asset-property-alias> \  
  --assetId <asset-ID> \  
  --propertyId <property-ID>
```

Ahora reasigne el flujo de datos a la nueva propiedad del activo.

```
aws iotsitewise associate-time-series-from-asset-property \  
  --alias <data-stream-alias> \  
  --assetId <new-asset-ID> \  
  --propertyId <new-property-ID>
```

Error al asignar un alias a una propiedad de activo

Al utilizar la UpdateAssetProperty API para asignar un alias a una propiedad, es posible que aparezca el siguiente mensaje de error:

```
Given alias <data-stream-alias> for property <property-name> with ID <property-ID>  
already in use by another property or data stream
```

Este mensaje de error indica que el alias no está asignado a la propiedad porque lo usa actualmente otra propiedad o un flujo de datos.

Esto ocurre si los datos se ingieren AWS IoT SiteWise con un alias. Cuando los datos se envían con un alias que no está siendo utilizado por otro flujo de datos o propiedad de activo, se crea un nuevo flujo de datos con ese alias. Las dos opciones siguientes resuelven el problema.

- Utilice la AssociateTimeSeriesToAssetProperty API para asociar el flujo de datos con su alias a la propiedad del activo.

- Detenga temporalmente la ingesta de datos y elimine el flujo de datos. Utilice la `UpdateAssetProperty` API para asignar el alias a la propiedad del activo y, a continuación, vuelva a activar la ingesta de datos.

Error al asociar un flujo de datos a una propiedad de activo

Al asociar un flujo de datos a una propiedad de un activo, aparece el siguiente mensaje de error.

```
assetProperty <property-name> with assetId <asset-ID> propertyId <property-ID> contains data
```

Este mensaje de error indica que la propiedad del activo ya está asociada a un flujo de datos que contiene datos. Ese flujo de datos debe disociarse o eliminarse antes de asociar otro flujo de datos a esa propiedad del activo.

Note

Al disociar un flujo de datos de una propiedad de activo, el alias asignado a la propiedad se asigna al flujo de datos. Para que ese alias permanezca asignado a la propiedad, asigne un nuevo alias a esa propiedad antes de disociar el flujo de datos.

Para conservar los datos almacenados en la propiedad del activo, haga lo siguiente:

- Asegúrese de que no se ingieran datos en la propiedad del activo para evitar la creación de un nuevo flujo de datos.
- Utilice la `UpdateAssetProperty` API para establecer un nuevo alias que se asigne al flujo de datos actualmente asignado.
- Utilice la `DisassociateTimeSeriesFromAssetProperty` API para desasociar el flujo de datos actual de la propiedad del activo.
- Utilice la `AssociateTimeSeriesToAssetProperty` API para asociar el flujo de datos deseado a la propiedad del activo.

Si se deben eliminar los datos almacenados en la propiedad del activo, haga lo siguiente:

- Asegúrese de que no se ingiera ningún dato a la propiedad del activo para evitar la creación de un nuevo flujo de datos.

- Utilice `DeleteTimeSeries` la API para eliminar el flujo de datos actualmente asignado.
- Utilice la `AssociateTimeSeriesToAssetProperty` API para asociar el flujo de datos deseado a la propiedad del activo.

Ingiera datos con AWS IoT SiteWise APIs

AWS IoT SiteWise APIs Utilícelo para enviar datos industriales con fecha y hora a las propiedades de medición y atributos de sus activos. APIs Acepta estructuras que contienen cargas útiles (TQV).
timestamp-quality-value

BatchPutAssetPropertyValue API

Utilice la operación [BatchPutAssetPropertyValue](#) para cargar sus datos. Con esta operación, puede cargar varias entradas de datos a la vez para recopilar datos de varios dispositivos y enviarlos todos en una sola solicitud.

Important

La [BatchPutAssetPropertyValue](#) operación está sujeta a las siguientes cuotas:

- Hasta 10 [entradas](#) por solicitud.
- Hasta 10 [valores de propiedad](#) (puntos de datos TQV) por entrada.
- AWS IoT SiteWise rechaza cualquier dato con una marca de tiempo fechada en más de 7 días o más de 10 minutos en el futuro.

Para obtener más información acerca de estas cuotas, consulte [BatchPutAssetPropertyValue](#) en la Referencia de API de AWS IoT SiteWise .

Para identificar la propiedad de un activo, especifique una de las siguientes opciones:

- `assetId` y `propertyId` de la propiedad del activo a la que está enviando datos.
- El `propertyAlias`, que es un alias de flujo de datos (por ejemplo, `/company/windfarm/3/turbine/7/temperature`). Para utilizar esta opción, primero debe establecer el alias de la propiedad del activo. Para establecer alias de propiedades, consulte [Gestione los flujos de datos para AWS IoT SiteWise](#).

El siguiente ejemplo muestra cómo enviar lecturas de temperatura y rotaciones por minuto (RPM) de una turbina eólica desde una carga útil almacenada en un archivo JSON.

```
aws iotsitewise batch-put-asset-property-value --cli-input-json file://batch-put-payload.json
```

La carga útil de ejemplo `batch-put-payload.json` tiene el siguiente contenido.

```
{
  "enablePartialEntryProcessing": true,
  "entries": [
    {
      "entryId": "unique entry ID",
      "propertyAlias": "/company/windfarm/3/turbine/7/temperature",
      "propertyValues": [
        {
          "value": {
            "integerValue": 38
          },
          "timestamp": {
            "timeInSeconds": 1575691200
          }
        }
      ]
    },
    {
      "entryId": "unique entry ID",
      "propertyAlias": "/company/windfarm/3/turbine/7/rpm",
      "propertyValues": [
        {
          "value": {
            "doubleValue": 15.09
          },
          "timestamp": {
            "timeInSeconds": 1575691200
          },
          "quality": "GOOD"
        }
      ]
    },
    {
      "entryId": "unique entry ID",
      "propertyAlias": "/company/windfarm/3/turbine/7/rpm",
```

```

    "propertyValues": [
      {
        "value": {
          "nullValue":{"valueType": "D"}
        },
        "timestamp": {
          "timeInSeconds": 1575691200
        },
        "quality": "BAD"
      }
    ]
  }
}

```

Si `enablePartialEntryProcessing` se especifica como `true` permite la ingesta de todos los valores que no provoquen errores. El comportamiento predeterminado es `false`. Si un valor no es válido, no se podrá ingerir toda la entrada.

Cada entrada de la carga contiene un `entryId` que puede definir como una única cadena. Si la entrada de la solicitud no se realiza correctamente, cada error contendrá el `entryId` de la solicitud correspondiente para que sepa qué solicitudes deben volver a intentarse.

Cada estructura de la lista de `propertyValues` es una estructura `timestamp-quality-value (TQV)` que contiene `value`, `timestamp`, y, opcionalmente `quality`

- `value`: una estructura que contiene uno de los siguientes campos, en función del tipo de propiedad que se establezca:
 - `booleanValue`
 - `doubleValue`
 - `integerValue`
 - `stringValue`
 - `nullValue`
- `nullValue`— Una estructura con el siguiente campo que indica el tipo de valor de la propiedad con un valor nulo y una calidad de `BAD` o `UNCERTAIN`
 - `valueType`— Enuma de `{"B", "D", "S", "I"}`
- `timestamp`: una estructura que contiene el tiempo en segundos en formato de tiempo Unix, `timeInSeconds`. También puede configurar la `offsetInNanos` clave en la `timestamp`

estructura si tiene datos precisos en el momento. AWS IoT SiteWise rechaza cualquier punto de datos con marcas de tiempo anteriores a 7 días o posteriores a 10 minutos en el futuro.

- `quality`: (opcional) una de las siguientes cadenas de calidad:
 - `GOOD`: (predeterminado) los datos no se ven afectados por ningún problema.
 - `BAD`: los datos se ven afectados por un problema, como un fallo del sensor.
 - `UNCERTAIN`: los datos se ven afectados por un problema, como la falta de precisión de un sensor.

Para obtener más información sobre cómo se AWS IoT SiteWise gestiona la calidad de los datos en los cálculos, consulte Calidad de [los datos en las expresiones de fórmulas](#).

CreateBulkImportJob API

Utilice la API `CreateBulkImportJob` para importar grandes cantidades de datos de Amazon S3. Los datos deben guardarse en formato CSV en Amazon S3. Los archivos de datos pueden tener las siguientes columnas.

Note

No se admiten datos anteriores a las 00:00:00 UTC del 1 de enero de 1970.

Para identificar la propiedad de un activo, especifique una de las siguientes opciones.

- La `ASSET_ID` y `PROPERTY_ID` de la propiedad del activo al que va a enviar los datos.
- El `ALIAS`, que es un alias de flujo de datos (por ejemplo, `/company/windfarm/3/turbine/7/temperature`). Para utilizar esta opción, primero debe establecer el alias de la propiedad del activo. Para aprender a configurar los alias de las propiedades, consulte [the section called “Administre los flujos de datos”](#).

- `ALIAS`: el alias que identifica la propiedad, como una ruta de flujo de datos del servidor OPC UA (por ejemplo, `/company/windfarm/3/turbine/7/temperature`). Para obtener más información, consulte [Gestione los flujos de datos para AWS IoT SiteWise](#).
- `ASSET_ID`: el ID del activo.
- `PROPERTY_ID`: el ID de la propiedad del activo.
- `DATA_TYPE`: el tipo de datos de la propiedad puede ser uno de los siguientes.
 - `STRING` – Una cadena con hasta 1024 bytes.

- INTEGER: un entero de 32 bits con signo con rango [-2.147.483.648, 2.147.483.647].
- DOUBLE: un número de punto flotante con rango [-10¹⁰⁰, 10¹⁰⁰] e IEEE 754 doble precisión.
- BOOLEAN – true o false.
- TIMESTAMP_SECONDS: la marca temporal del punto de datos, con la hora en formato Unix.
- TIMESTAMP_NANO_OFFSET: el desplazamiento de nanosegundos convertido de TIMESTAMP_SECONDS.
- QUALITY: (opcional) La calidad del valor de la propiedad del activo. El valor puede ser uno de los siguientes:
 - GOOD: (predeterminado) los datos no se ven afectados por ningún problema.
 - BAD: los datos se ven afectados por un problema, como un fallo del sensor.
 - UNCERTAIN: los datos se ven afectados por un problema, como la falta de precisión de un sensor.

Para obtener más información sobre cómo se AWS IoT SiteWise gestiona la calidad de los datos en los cálculos, consulte Calidad de [los datos en las expresiones de fórmulas](#).

- VALUE: el valor de la propiedad del activo.

Example archivo(s) de datos en formato .csv

```
asset_id,property_id,DOUBLE,1635201373,0,GOOD,1.0
asset_id,property_id,DOUBLE,1635201374,0,GOOD,2.0
asset_id,property_id,DOUBLE,1635201375,0,GOOD,3.0
```

```
unmodeled_alias1,DOUBLE,1635201373,0,GOOD,1.0
unmodeled_alias1,DOUBLE,1635201374,0,GOOD,2.0
unmodeled_alias1,DOUBLE,1635201375,0,GOOD,3.0
unmodeled_alias1,DOUBLE,1635201376,0,GOOD,4.0
unmodeled_alias1,DOUBLE,1635201377,0,GOOD,5.0
unmodeled_alias1,DOUBLE,1635201378,0,GOOD,6.0
unmodeled_alias1,DOUBLE,1635201379,0,GOOD,7.0
unmodeled_alias1,DOUBLE,1635201380,0,GOOD,8.0
unmodeled_alias1,DOUBLE,1635201381,0,GOOD,9.0
unmodeled_alias1,DOUBLE,1635201382,0,GOOD,10.0
```

AWS IoT SiteWise proporciona las siguientes operaciones de API para crear un trabajo de importación masiva y obtener información sobre un trabajo existente.

- [CreateBulkImportJob](#)— Crea un nuevo trabajo de importación masiva.
- [DescribeBulkImportJob](#)— Recupera información sobre un trabajo de importación masiva.
- [ListBulkImportJob](#)— Recupera una lista paginada de resúmenes de todos los trabajos de importación masiva.

Crea un trabajo de importación AWS IoT SiteWise masiva ()AWS CLI

Utilice la operación de [CreateBulkImportJob](#) API para transferir datos de Amazon S3 a AWS IoT SiteWise. La [CreateBulkImportJob](#) API permite la ingesta de grandes volúmenes de datos históricos y la ingesta en búfer de flujos de datos analíticos en lotes pequeños. Proporciona un método básico rentable para la ingesta de datos. El siguiente ejemplo utiliza AWS CLI.

Important

Antes de crear un trabajo de importación masiva, debe habilitar el nivel AWS IoT SiteWise cálido o el nivel frío. AWS IoT SiteWise Para obtener más información, consulte [Configure los ajustes de almacenamiento en AWS IoT SiteWise](#).

La [CreateBulkImportJob](#) API admite la ingesta de datos históricos AWS IoT SiteWise con la opción de configurar el `adaptive-ingestion-flag` parámetro.

- Cuando se configura en `false`, la API ingiere datos históricos sin activar cálculos ni notificaciones.
- Cuando se configura en `true`, la API ingiere nuevos datos, calculando métricas y transformando los datos para optimizar los análisis y las notificaciones continuos en un plazo de siete días.

Ejecute el comando siguiente. *file-name* Sustitúyalo por el nombre del archivo que contiene la configuración del trabajo de importación masiva.

```
aws iotsitewise create-bulk-import-job --cli-input-json file://file-name.json
```

Example Configuración de trabajos de importación masiva

A continuación se ofrecen ejemplos de configuración:

- Reemplace *adaptive-ingestion-flag* por `true` o `false`.

- Si se establece en `false`, el trabajo de importación masiva ingiere datos históricos en AWS IoT SiteWise.
- Si se establece en `true`, el trabajo de importación masiva hace lo siguiente:
 - Ingiere nuevos datos en. AWS IoT SiteWise
 - Calcula métricas y transformaciones, y admite notificaciones de datos con una marca de tiempo de siete días.
- *delete-files-after-import-flag* Sustitúyalos por `true` para eliminar los datos del depósito de datos de Amazon S3 después de ingerirlos en un almacenamiento de nivel AWS IoT SiteWise cálido.
- Sustituya `amzn-s3-demo-bucket` por el *for-errors* nombre del bucket de Amazon S3 al que se envían los errores asociados a este trabajo de importación masiva.
- Sustituya `amzn-s3-demo-bucket` por el *for-errors-prefix* prefijo del bucket de Amazon S3 al que se envían los errores asociados a este trabajo de importación masiva.

Amazon S3 usa el prefijo como nombre de carpeta para organizar los datos del bucket. Cada objeto de Amazon S3 tiene una clave que es su identificador único en el bucket. Cada objeto de un bucket tiene exactamente una clave. El prefijo debe terminar con una barra diagonal: (/). Para obtener más información, consulte [Organizar objetos usando prefijos](#) en la Guía para usuarios de Amazon Simple Storage Service.

- Sustituya `amzn-s3-demo-bucket` por el *data* nombre del bucket de Amazon S3 desde el que se importan los datos.
- *data-bucket-key* Sustitúyala por la clave del objeto de Amazon S3 que contiene los datos. Cada objeto tiene una clave que es un identificador único. Cada objeto tiene exactamente una clave.
- *data-bucket-version-id* Sustitúyalo por el ID de versión para identificar una versión específica del objeto de Amazon S3 que contiene sus datos. Este parámetro es opcional.
- *column-name* Sustitúyalo por el nombre de la columna especificado en el archivo.csv.
- *job-name* Sustitúyalo por un nombre único que identifique el trabajo de importación masiva.
- *job-role-arn* Sustitúyalo por el rol de IAM que AWS IoT SiteWise permite leer los datos de Amazon S3.

Note

Asegúrese de que el rol tenga los permisos que se muestran en el siguiente ejemplo. Sustituya `amzn-s3-demo-bucket` por el *data* nombre del bucket de Amazon S3 que

contiene los datos. Además, *amzn-s3-demo-bucket-for-errors* sustitúyalo por el nombre del bucket de Amazon S3 al que se envían los errores asociados a este trabajo de importación masiva.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3:GetObject",
        "s3:GetBucketLocation"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket-data",
        "arn:aws:s3:::amzn-s3-demo-bucket-data/*"
      ],
      "Effect": "Allow"
    },
    {
      "Action": [
        "s3:PutObject",
        "s3:GetObject",
        "s3:GetBucketLocation"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket-for-errors",
        "arn:aws:s3:::amzn-s3-demo-bucket-for-errors/*"
      ],
      "Effect": "Allow"
    }
  ]
}
```

```
{
  "adaptiveIngestion": adaptive-ingestion-flag,
  "deleteFilesAfterImport": delete-files-after-import-flag,
  "errorReportLocation": {
    "bucket": "amzn-s3-demo-bucket-for-errors",
```

```

    "prefix": "amzn-s3-demo-bucket-for-errors-prefix"
  },
  "files": [
    {
      "bucket": "amzn-s3-demo-bucket-data",
      "key": "data-bucket-key",
      "versionId": "data-bucket-version-id"
    }
  ],
  "jobConfiguration": {
    "fileFormat": {
      "csv": {
        "columnNames": [ "column-name" ]
      }
    }
  },
  "jobName": "job-name",
  "jobRoleArn": "job-role-arn"
}

```

Example response

```

{
  "jobId": "f8c031d0-01d1-4b94-90b1-afe8bb93b7e5",
  "jobStatus": "PENDING",
  "jobName": "myBulkImportJob"
}

```

Describa un trabajo de importación AWS IoT SiteWise masiva (AWS CLI)

Utilice la operación de la [DescribeBulkImportJob](#) API para recuperar información sobre un trabajo de importación masiva específico en AWS IoT SiteWise. Esta operación devuelve detalles como estado del trabajo, hora de creación e información de error si el trabajo ha fallado. Puede utilizar esta operación para supervisar el progreso del trabajo y solucionar problemas. Para utilizar `DescribeBulkImportJob` necesita el ID del trabajo de la operación de `CreateBulkImportJob`. La API devuelve la siguiente información:

- Lista de archivos que se están importando, incluidas sus ubicaciones y claves de los buckets de Amazon S3
- Ubicación del informe de errores (si corresponde)
- Detalles de configuración del trabajo, como formato de archivo y nombres de columnas del CSV

- Marcas de tiempo de creación de trabajo y última actualización
- Estado actual del trabajo (por ejemplo, si el trabajo está en curso, se ha completado o ha fallado)
- ARN de rol de IAM para el trabajo de importación

En el caso de los trabajos finalizados, revise los resultados para confirmar que la integración de datos se ha realizado correctamente. Si un trabajo falla, examine los detalles del error para diagnosticar y resolver los problemas.

job-ID Sustitúyalo por el ID del trabajo de importación masiva que deseas recuperar.

```
aws iotsitewise describe-bulk-import-job --job-id job-ID
```

Example response

```
{
  "files": [
    {
      "bucket": "amzn-s3-demo-bucket1",
      "key": "100Tags12Hours.csv"
    },
    {
      "bucket": "amzn-s3-demo-bucket2",
      "key": "BulkImportData1MB.csv"
    },
    {
      "bucket": " amzn-s3-demo-bucket3",
      "key": "UnmodeledBulkImportData1MB.csv"
    }
  ],
  "errorReportLocation": {
    "prefix": "errors/",
    "bucket": "amzn-s3-demo-bucket-for-errors"
  },
  "jobConfiguration": {
    "fileFormat": {
      "csv": {
        "columnNames": [
          "ALIAS",
          "DATA_TYPE",
          "TIMESTAMP_SECONDS",
          "TIMESTAMP_NANO_OFFSET",
```

```
        "QUALITY",
        "VALUE"
    ]
}
},
"jobCreationDate":1645745176.498,
"jobStatus":"COMPLETED",
"jobName":"myBulkImportJob",
"jobLastUpdateDate":1645745279.968,
"jobRoleArn":"arn:aws:iam::123456789012:role/DemoRole",
"jobId":"f8c031d0-01d1-4b94-90b1-afe8bb93b7e5"
}
```

Enumere los trabajos de importación AWS IoT SiteWise masiva (AWS CLI)

Utilice la operación de la [ListBulkImportJobs](#) API para recuperar una lista de resúmenes de los trabajos de importación masiva en AWS IoT SiteWise. Esta operación proporciona una forma eficaz de supervisar y administrar los procesos de importación de datos. Devuelve la siguiente información clave para cada trabajo:

- ID del trabajo. Identificador único para cada trabajo de importación masiva
- Nombre del trabajo. Nombre que asignó al trabajo cuando lo creó.
- Estado actual. Estado actual del trabajo (por ejemplo, COMPLETED, RUNNING, FAILED)

ListBulkImportJobs es especialmente útil para obtener una visión general completa de todos sus trabajos de importación masiva. Puede ayudarlo a realizar un seguimiento de las múltiples importaciones de datos, identificar los trabajos que requieren atención y mantener un flujo de trabajo organizado. La operación admite paginación, lo que permite recuperar un gran número de resúmenes de trabajos de manera eficiente. Puede utilizar el trabajo IDs devuelto por esta operación con la [DescribeBulkImportJob](#) operación para recuperar información más detallada sobre trabajos específicos. Este proceso de dos pasos le permite obtener primero una vista de alto nivel de todos los trabajos y, a continuación, profundizar en los detalles de los trabajos de interés. Cuando utilice ListBulkImportJobs, puede aplicar filtros para acotar los resultados. Por ejemplo, puede filtrar los trabajos en función de su estado para recuperar solo los trabajos finalizados o solo los trabajos en ejecución. Esta característica ayuda a centrarse en la información más relevante para su tarea actual. La operación también devuelve un nextToken si hay más resultados disponibles. Puede usar este token en llamadas posteriores para recuperar el siguiente conjunto de resúmenes de trabajos, lo que le permitirá iterar por todos los trabajos de importación masiva, aunque tenga

muchos. En el siguiente ejemplo, se muestra cómo `ListBulkImportJobs` utilizarla AWS CLI para recuperar una lista de trabajos finalizados.

```
aws iotsitewise list-bulk-import-jobs --filter COMPLETED
```

Example Filtro de respuesta para trabajos completados

```
{
  "jobSummaries":[
    {
      "id":"bdbbfa52-d775-4952-b816-13ba1c7cb9da",
      "name":"myBulkImportJob",
      "status":"COMPLETED"
    },
    {
      "id":"15ffc641-dbd8-40c6-9983-5cb3b0bc3e6b",
      "name":"myBulkImportJob2",
      "status":"COMPLETED"
    }
  ]
}
```

Este comando muestra cómo se utiliza `ListBulkImportJobs` para recuperar una lista de trabajos que se han completado con errores. El máximo está establecido en 50 resultados y utilizamos el siguiente token para los resultados paginados.

```
aws iotsitewise list-bulk-import-jobs --filter COMPLETED_WITH_FAILURES --max-results 50
--next-token "string"
```

Ingiera datos para AWS IoT SiteWise usar reglas AWS IoT Core

Envíe datos AWS IoT SiteWise desde AWS IoT objetos y otros AWS servicios mediante las reglas de AWS IoT Core. Las reglas transforman los mensajes de MQTT y realizan acciones para interactuar con AWS los servicios. La acción de la AWS IoT SiteWise regla reenvía los datos de los mensajes a la [BatchPutAssetPropertyValue](#) operación desde la AWS IoT SiteWise API. Para obtener más información, consulte [Reglas](#) y [Acción AWS IoT SiteWise](#) en la Guía para desarrolladores de AWS IoT .

Si desea consultar un tutorial con los pasos necesarios para configurar una regla que ingiera datos a través de sus sombras de dispositivo, vea [Transfiere datos AWS IoT SiteWise desde cosas AWS IoT](#).

También puedes enviar datos desde AWS IoT SiteWise otros AWS servicios. Para obtener más información, consulte [Interactúa con otros AWS servicios](#).

Temas

- [Conceda AWS IoT el acceso requerido](#)
- [Configure la acción de la AWS IoT SiteWise regla](#)
- [Reduzca los costes con Basic Ingest in AWS IoT SiteWise](#)

Conceda AWS IoT el acceso requerido

Las funciones de IAM se utilizan para controlar los AWS recursos a los que tiene acceso cada regla. Antes de crear una regla, debe crear una función de IAM con una política que permita a la regla realizar acciones en el recurso requerido AWS . AWS IoT asume esta función al ejecutar una regla.

Si crea la acción de la regla en la AWS IoT consola, puede elegir un activo raíz para crear un rol que tenga acceso a una jerarquía de activos seleccionada. Para obtener más información sobre cómo definir manualmente un rol para una regla, consulte [Otorgar AWS IoT el acceso necesario](#) y [transferir los permisos de rol](#) en la Guía para AWS IoT desarrolladores.

Para la acción de la AWS IoT SiteWise regla, debe definir una función que permita el `iotsitewise:BatchPutAssetPropertyValue` acceso a las propiedades de los activos a las que la regla envía los datos. Para mejorar la seguridad, puede especificar una ruta jerárquica de AWS IoT SiteWise activos en la `Condition` propiedad.

La siguiente política de confianza de ejemplo permite el acceso a un activo específico y a sus secundarios.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iotsitewise:BatchPutAssetPropertyValue",
      "Resource": "*",
      "Condition": {
        "StringLike": {
```

```

        "iotsitewise:assetHierarchyPath": [
            "/root node asset ID",
            "/root node asset ID/*"
        ]
    }
}
]
}

```

Quite `Condition` de la política para permitir el acceso a todos sus activos. La política de confianza de ejemplo siguiente permite el acceso a todos los activos de la región actual.

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iotsitewise:BatchPutAssetPropertyValue",
      "Resource": "*"
    }
  ]
}

```

Configure la acción de la AWS IoT SiteWise regla

La acción de la AWS IoT SiteWise regla envía los datos del mensaje MQTT que inició la regla a las propiedades del activo. AWS IoT SiteWise Puede cargar varias entradas de datos a diferentes propiedades de activos al mismo tiempo para enviar actualizaciones para todos los sensores de un dispositivo en un solo mensaje. También puede cargar varios puntos de datos a la vez para cada entrada de datos.

Note

Al enviar datos a AWS IoT SiteWise con la acción de la regla, estos deben cumplir todos los requisitos de la `BatchPutAssetPropertyValue` operación. Por ejemplo, los datos no

pueden tener una marca temporal anterior a 7 días a partir de la hora actual en formato Unix. Para obtener más información, consulte [Ingesta de datos con la API AWS IoT SiteWise](#).

Para cada entrada de datos de la acción de regla, se identifica una propiedad de activo y se especifica la marca temporal, la calidad y el valor de cada punto de datos para esa propiedad de activo. La acción de regla espera cadenas para todos los parámetros.

Para identificar una propiedad de activo en una entrada, puede especificar uno de los elementos siguientes:

- El Asset ID (ID de activo) (`assetId`) y Property ID (ID de propiedad) (`propertyId`) de la propiedad del activo al que está enviando los datos. Puede encontrar el ID de activo y el ID de propiedad utilizando Consola de AWS IoT SiteWise. Si conoce el identificador del activo, puede utilizar el AWS CLI para llamar y [DescribeAsset](#) buscar el identificador de la propiedad.
- El Property alias (Alias de propiedad) (`propertyAlias`), que es un alias de flujo de datos (por ejemplo, `/company/windfarm/3/turbine/7/temperature`). Para utilizar esta opción, primero debe establecer el alias de la propiedad del activo. Para aprender a configurar los alias de las propiedades, consulte [Gestione los flujos de datos para AWS IoT SiteWise](#).

Para la marca temporal de cada entrada, use la marca temporal proporcionada por AWS IoT Core o que notificó su equipo. La marca temporal tiene dos parámetros:

- Tiempo en segundos (`timeInSeconds`): la hora en formato Unix, en segundos, a la que el sensor o el equipo notificó los datos.
- Desfase en nanosegundos (`offsetInNanos`): (opcional) el desfase en nanosegundos respecto al tiempo en segundos.

Important

Si la marca temporal es una cadena, tiene una parte decimal o no está en segundos, AWS IoT SiteWise rechaza la solicitud. Debe convertir la marca temporal en segundos y el desfase en nanosegundos. Utilice las funciones del motor de AWS IoT reglas para convertir la marca de tiempo. Para obtener más información, consulte los siguientes temas:

- [Obtener marcas temporales para dispositivos que no indican la hora exacta](#)

- [Convertir las marcas temporales que están en formato de cadena](#)

Puede utilizar plantillas de sustitución para varios parámetros para realizar cálculos, invocar funciones y extraer valores de la carga útil del mensaje. Para obtener más información, consulte [Plantillas de sustitución](#) en la Guía para desarrolladores de AWS IoT .

Note

Dado que una expresión de una plantilla de sustitución se evalúa por separado de la instrucción SELECT, no se puede utilizar una plantilla de sustitución para hacer referencia a un alias creado mediante una cláusula AS. Solo puede hacer referencia a la información presente en la carga original, además de a las funciones y operadores compatibles.

Temas

- [Obtener marcas temporales para dispositivos que no indican la hora exacta](#)
- [Convertir las marcas temporales que están en formato de cadena](#)
- [Convertir cadenas de marcas temporales con una precisión de nanosegundos](#)
- [Ejemplo de configuraciones de regla](#)
- [Solución de problemas de la acción de regla de](#)

Obtener marcas temporales para dispositivos que no indican la hora exacta

[Si su sensor o equipo no proporciona datos de tiempo precisos, obtenga la época actual de Unix desde el motor de AWS IoT reglas con timestamp \(\)](#). Esta función muestra la hora en milisegundos, por lo que debe convertir el valor a tiempo en segundos y desfase en nanosegundos. Para ello, utilice las siguientes conversiones:

- En Time in seconds (Hora en segundos) (timeInSeconds), utilice `$\{\text{floor}(\text{timestamp}() / 1E3)\}$` para convertir la hora de milisegundos a segundos.
- En Offset in nanos (Desfase en nanosegundos) (offsetInNanos), utilice `$\{(\text{timestamp}() \% 1E3) * 1E6\}$` para calcular el desfase en nanosegundos de la marca temporal.

Convertir las marcas temporales que están en formato de cadena

Si su sensor o equipo notifica los datos de tiempo en formato de cadena (por ejemplo, `2020-03-03T14:57:14.699Z`), use [time_to_epoch\(String, String\)](#). Esta función introduce la marca temporal y el patrón de formato como parámetros y genera el tiempo en milisegundos. A continuación, debe convertir el tiempo en tiempo en segundos y el desfase en nanosegundos. Para ello, utilice las siguientes conversiones:

- En Hora en segundos (`timeInSeconds`), utilice $\{\{\text{floor}(\text{time_to_epoch}("2020-03-03T14:57:14.699Z", "yyyy-MM-dd'T'HH:mm:ss'Z'") / 1E3)\}$ para convertir la cadena de marca temporal a milisegundos, y luego a segundos.
- En Desfase en nanosegundos (`offsetInNanos`), utilice $\{\{(\text{time_to_epoch}("2020-03-03T14:57:14.699Z", "yyyy-MM-dd'T'HH:mm:ss'Z'") \% 1E3) * 1E6\}$ para calcular el desfase en nanosegundos de la cadena de la marca temporal.

Note

La función `time_to_epoch` admite cadenas de marcas temporales con una precisión de hasta milisegundos. Para convertir cadenas con una precisión de microsegundos o nanosegundos, configura una AWS Lambda función a la que tu regla llame para convertir la marca de tiempo en valores numéricos. Para obtener más información, consulte [Convertir cadenas de marcas temporales con una precisión de nanosegundos](#).

Convertir cadenas de marcas temporales con una precisión de nanosegundos

Si su dispositivo envía información de marca temporal en formato de cadena (por ejemplo, `2020-03-03T14:57:14.699728491Z`), siga este procedimiento para configurar su acción de la regla. Puedes crear una AWS Lambda función que convierta la marca temporal de una cadena en Tiempo en segundos (`timeInSeconds`) y Offset en nanos (`offsetInNanos`). A continuación, use [aws_lambda\(functionArn, inputJson\)](#) en sus parámetros de acción de regla para invocar esa función de Lambda y usar la salida en su regla.

Note

En esta sección se incluyen instrucciones avanzadas que suponen que está familiarizado con la forma de crear estos recursos:

- Funciones de Lambda. Para obtener más información, consulte [Creación de su primera función de Lambda](#) en la Guía para desarrolladores de AWS Lambda .
- AWS IoT gobierna con la acción de la AWS IoT SiteWise regla. Para obtener más información, consulte [Ingiera datos para AWS IoT SiteWise usar reglas AWS IoT Core](#).

Para crear una acción de AWS IoT SiteWise regla que analice cadenas de marcas de tiempo

1. Cree una función de Lambda con las siguientes propiedades:

- Nombre de la función: utilice un nombre de función descriptivo (por ejemplo, **ConvertNanosecondTimestampFromString**).
- Tiempo de ejecución: utilice un tiempo de ejecución de Python 3, como Python 3.11 (python3.11).
- Permisos: cree un rol con permisos Lambda básicos () AWS LambdaBasicExecutionRole.
- Capas: añada la capa AWS SDKPandas-Python311 para que la utilice la función Lambda. numpy
- Código de función: utilice el siguiente código de función, que consume un argumento de cadena denominado `timestamp` y las salidas `timeInSeconds` y `offsetInNanos` los valores de esa marca temporal.

```
import json
import math
import numpy

# Converts a timestamp string into timeInSeconds and offsetInNanos in Unix epoch
time.
# The input timestamp string can have up to nanosecond precision.
def lambda_handler(event, context):
    timestamp_str = event['timestamp']
    # Parse the timestamp string as nanoseconds since Unix epoch.
    nanoseconds = numpy.datetime64(timestamp_str, 'ns').item()
    time_in_seconds = math.floor(nanoseconds / 1E9)
    # Slice to avoid precision issues.
```

```

offset_in_nanos = int(str(nanoseconds)[-9:])
return {
    'timeInSeconds': time_in_seconds,
    'offsetInNanos': offset_in_nanos
}

```

[Esta función Lambda introduce cadenas de marcas de tiempo en formato ISO 8601 mediante `datetime64` from. NumPy](#)

Note

Si las cadenas de marca temporal no están en formato ISO 8601, puede implementar una solución con pandas que defina el formato de la marca temporal. Para obtener más información, consulte [pandas.to_datetime..](#)

- Al configurar la AWS IoT SiteWise acción de la regla, utilice las siguientes plantillas de sustitución para Time in seconds (**timeInSeconds**) y Offset in nanos (**offsetInNanos**). Estas plantillas de sustitución suponen que su carga de mensaje contiene la cadena de marca temporal en `timestamp`. La función `aws_lambda` consume una estructura JSON para su segundo parámetro, por lo que puede modificar las siguientes plantillas de sustitución en caso necesario.
 - En Time in seconds (Tiempo en segundos) (`timeInSeconds`), use la siguiente plantilla de sustitución.

```

${aws_lambda('arn:aws:lambda:region:account-
id:function:ConvertNanosecondTimestampFromString', {'timestamp':
timestamp}).timeInSeconds}

```

- En Offset in nanos (Desfase en nanosegundos) (`offsetInNanos`), use la siguiente plantilla de sustitución.

```

${aws_lambda('arn:aws:lambda:region:account-
id:function:ConvertNanosecondTimestampFromString', {'timestamp':
timestamp}).offsetInNanos}

```

Para cada parámetro, sustituye *region* y *account-id* por tu región e ID de AWS cuenta. Si ha usado otro nombre para su función de Lambda, cámbielo también.

3. Concede AWS IoT permisos para invocar tu función con el `lambda:InvokeFunction` permiso. Para obtener más información, consulte [aws_lambda\(functionArn, inputJson\)](#).
4. Pruebe la regla (por ejemplo, utilice el cliente de prueba AWS IoT MQTT) y compruebe que AWS IoT SiteWise recibe los datos que envía.

Si su regla no funciona según lo previsto, consulte [Solucionar problemas con una acción de AWS IoT SiteWise regla](#).

Note

Esta solución invoca la función de Lambda dos veces para cada cadena de marca temporal. Puede crear otra regla para reducir el número de invocaciones de funciones de Lambda si su regla gestiona varios puntos de datos que tienen la misma marca temporal en cada carga. Para ello, cree una regla con una acción de volver a publicar que invoca Lambda y publica la carga original con la cadena de marca temporal convertida en `timeInSeconds` y `offsetInNanos`. A continuación, cree una regla con una acción de AWS IoT SiteWise regla para consumir la carga útil convertida. Con este enfoque, se reduce el número de veces que la regla invoca la Lambda, pero se aumenta el número de acciones AWS IoT de la regla que se ejecutan. Considere los precios de cada servicio si aplica esta solución a su caso de uso.

Ejemplo de configuraciones de regla

Esta sección contiene ejemplos de configuraciones de reglas para crear una regla con una AWS IoT SiteWise acción.

Example Acción de regla de ejemplo que utiliza alias de propiedad como temas de mensaje

En el siguiente ejemplo, se crea una regla con una AWS IoT SiteWise acción que utiliza el tema (a través del [tema \(\)](#)) como alias de la propiedad para identificar las propiedades de los activos. Utilice este ejemplo para definir una regla que incorpore datos de tipo doble a todos los aerogeneradores de todos los parques eólicos. Este ejemplo requiere que defina alias de propiedad en todas las propiedades de los activos de la turbina. Necesitaría definir una segunda regla similar para ingerir datos de tipo entero.

```
aws iot create-topic-rule \  
  --rule-name SiteWiseWindFarmRule \  
  --topic-rule-payload file://sitewise-rule-payload.json
```

La carga de ejemplo en `sitewise-rule-payload.json` contiene el siguiente contenido.

```
{
  "sql": "SELECT * FROM '/company/windfarm/+/turbine/+/+' WHERE type = 'double'",
  "description": "Sends data to the wind turbine asset property with the same alias as the topic",
  "ruleDisabled": false,
  "awsIotSqlVersion": "2016-03-23",
  "actions": [
    {
      "iotSiteWise": {
        "putAssetPropertyValueEntries": [
          {
            "propertyAlias": "${topic()}",
            "propertyValues": [
              {
                "timestamp": {
                  "timeInSeconds": "${timeInSeconds}"
                },
                "value": {
                  "doubleValue": "${value}"
                }
              }
            ]
          }
        ],
        "roleArn": "arn:aws:iam::account-id:role/MySiteWiseActionRole"
      }
    }
  ]
}
```

Con esta acción de regla, envíe el siguiente mensaje a un alias de propiedad de turbina eólica (por ejemplo, `/company/windfarm/3/turbine/7/temperature`) como tema para ingerir datos.

```
{
  "type": "double",
  "value": "38.3",
  "timeInSeconds": "1581368533"
}
```

Example Ejemplo de acción de regla que utiliza timestamp() para determinar la hora

En el siguiente ejemplo, se crea una regla con una AWS IoT SiteWise acción que identifica una propiedad de activo mediante timestamp () IDs y utiliza la [marca timestamp \(\)](#) para determinar la hora actual.

```
aws iot create-topic-rule \
  --rule-name SiteWiseAssetPropertyRule \
  --topic-rule-payload file://sitewise-rule-payload.json
```

La carga de ejemplo en sitewise-rule-payload.json contiene el siguiente contenido.

```
{
  "sql": "SELECT * FROM 'my/asset/property/topic'",
  "description": "Sends device data to an asset property",
  "ruleDisabled": false,
  "awsIotSqlVersion": "2016-03-23",
  "actions": [
    {
      "iotSiteWise": {
        "putAssetPropertyValueEntries": [
          {
            "assetId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
            "propertyId": "a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
            "propertyValues": [
              {
                "timestamp": {
                  "timeInSeconds": "${floor(timestamp() / 1E3)}",
                  "offsetInNanos": "${(timestamp() % 1E3) * 1E6}"
                },
                "value": {
                  "doubleValue": "${value}"
                }
              }
            ]
          }
        ]
      },
      "roleArn": "arn:aws:iam::account-id:role/MySiteWiseActionRole"
    }
  ]
}
```

Con esta acción de regla, envíe el siguiente mensaje a `my/asset/property/topic` para ingerir datos.

```
{
  "type": "double",
  "value": "38.3"
}
```

Solución de problemas de la acción de regla de

Para solucionar la acción de la AWS IoT SiteWise regla AWS IoT Core, configura CloudWatch los registros o configura una acción de error al volver a publicar la regla. Para obtener más información, consulte [Solucionar problemas con una acción de AWS IoT SiteWise regla](#).

Reduzca los costes con Basic Ingest in AWS IoT SiteWise

AWS IoT Core [proporciona una función denominada ingesta básica que puede utilizar para enviar datos AWS IoT Core sin incurrir AWS IoT en costes de mensajería](#). Basic Ingest optimiza el flujo de datos para cargas de trabajo de ingestión de datos de gran volumen al eliminar el agente de publish/subscribe mensajes de la ruta de ingestión. Puede utilizar Basic Ingest si sabe a qué reglas deben dirigirse sus mensajes.

Para utilizar Basic Ingest, se envían mensajes directamente a una regla específica utilizando un tema especial, `$aws/rules/rule-name`. Por ejemplo, para enviar un mensaje a una regla denominada `SiteWiseWindFarmRule`, envíe un mensaje al tema `$aws/rules/SiteWiseWindFarmRule`.

Si la acción de regla utiliza plantillas de sustitución que contengan [topic\(Decimal\)](#), puede transferir el tema original al final del tema especial de Basic Ingest, como `$aws/rules/rule-name/original-topic`. Por ejemplo, para utilizar Basic Ingest con el ejemplo de alias de propiedad del parque eólico de la sección anterior, puede enviar mensajes al tema siguiente.

```
$aws/rules/SiteWiseWindFarmRule//company/windfarm/3/turbine/7/temperature
```

Note

El ejemplo anterior incluye una segunda barra (//) porque AWS IoT elimina el prefijo Basic Ingest () del tema que está visible para la `$aws/rules/rule-name`/ acción de la regla. En este ejemplo, la regla recibe el tema `/company/windfarm/3/turbine/7/temperature`.

Para obtener más información, consulte [Reducción de los costos de mensajería con Basic Ingest](#) en la Guía para desarrolladores de AWS IoT .

Ingiera datos desde AWS IoT SiteWise AWS IoT Events

Con AWS IoT Events, puede crear aplicaciones de monitoreo de eventos complejas para su flota de IoT en la AWS nube. Utilice la SiteWise acción de IoT AWS IoT Events para enviar datos a las propiedades de los activos AWS IoT SiteWise cuando se produzca un evento.

Note

Aviso de fin de soporte: se AWS ha finalizado el soporte para AWS IoT Events. Para obtener más información, consulte [AWS IoT Events el fin del soporte](#).

AWS IoT Events está diseñado para agilizar el desarrollo de aplicaciones de monitoreo de eventos para dispositivos y sistemas de IoT dentro de la AWS nube. Si lo usa AWS IoT Events, puede:

- Detectar cambios, anomalías o condiciones específicas en la flota de IoT y responder a ellos.
- Mejorar la eficiencia operativa y habilitar la administración proactiva del ecosistema de IoT.

Al integrarse a AWS IoT SiteWise través de la AWS IoT SiteWise acción, AWS IoT Events amplía sus capacidades, lo que le permite actualizar automáticamente las propiedades de los activos AWS IoT SiteWise en respuesta a eventos específicos. Esta interacción puede simplificar la ingesta y la administración de datos. También puede proporcionar información útil.

Para obtener más información, consulte los siguientes temas de la Guía para desarrolladores de AWS IoT Events :

- [¿Qué es AWS IoT Events?](#)
- [Acciones de AWS IoT Events](#)
- [SiteWise Acción de IoT](#)

Utilice el administrador de AWS IoT Greengrass transmisiones en AWS IoT SiteWise

AWS IoT Greengrass El administrador de flujos es una función de integración que facilita la transferencia de flujos de datos de fuentes locales a la AWS nube. Actúa como una capa intermedia que gestiona los flujos de datos, lo que permite a los dispositivos que funcionan en la periferia recopilar y almacenar datos antes de enviarlos AWS IoT SiteWise, para su posterior análisis y procesamiento.

Agregue un destino de datos configurando un origen local en la consola de AWS IoT SiteWise . También puedes usar el administrador de flujos en tu AWS IoT Greengrass solución personalizada para ingerir datos. AWS IoT SiteWise

Note

Para ingerir datos de fuentes OPC UA, configure una puerta de enlace AWS IoT SiteWise Edge que se ejecute en AWS IoT Greengrass Para obtener más información, consulte [Utilice puertas de enlace AWS IoT SiteWise Edge](#).

Para obtener más información acerca de cómo configurar un destino para datos de origen local, consulte [Comprenda los destinos de AWS IoT SiteWise Edge](#).

Para obtener más información sobre cómo ingerir datos mediante el administrador de flujos en una solución de AWS IoT Greengrass personalizada, consulte los siguientes temas en la Guía para desarrolladores de AWS IoT Greengrass Version 2 :

- [¿Qué es? AWS IoT Greengrass](#)
- [Administrar secuencias de datos en el núcleo de AWS IoT Greengrass](#)
- [Exportación de datos a propiedades AWS IoT SiteWise de activos](#)

Utilice puertas de enlace AWS IoT SiteWise Edge

AWS IoT SiteWise Edge extiende las capacidades de la nube a los entornos industriales periféricos, lo que permite el procesamiento, el análisis y la toma de decisiones de datos locales. SiteWise Edge se integra con AWS IoT SiteWise y otros AWS servicios para proporcionar soluciones integrales de IoT industrial. Las pasarelas sirven de intermediario entre su equipo industrial y AWS IoT SiteWise.

SiteWise Las puertas de enlace perimetrales se ejecutan en dos objetivos de implementación diferentes:

- AWS IoT Greengrass V2
- Siemens Industrial Edge

Puede usar una puerta de enlace SiteWise Edge para recopilar datos en la periferia y publicarlos en la nube. En el caso de las pasarelas que se ejecutan en ellas AWS IoT Greengrass, también puede procesar los datos en la periferia mediante modelos de activos y activos.

La aplicación AWS IoT SiteWise Edge Siemens Industrial Edge permite la integración entre los equipos industriales y permite agregar y AWS IoT SiteWise procesar los datos sin procesar de las máquinas y realizar análisis de forma local antes de enviar los datos refinados a la AWS nube.

Conceptos clave de las pasarelas SiteWise Edge

SiteWise Edge tiene varias funciones útiles para la computación periférica en entornos industriales.

Recopilación y procesamiento de datos locales

Soporta la recopilación de datos de activos industriales mediante protocolos como OPC-UA y MQTT. Las pasarelas se ejecutan en AWS IoT Greengrass dispositivos Core o. Siemens Industrial Edge

Funcionamiento sin conexión

Continúa recopilando y procesando datos durante las interrupciones de Internet y se sincroniza con la nube cuando se restablece la conectividad.

Computación perimetral con componentes AWS IoT Greengrass

Utiliza IoT SiteWise Publisher para reenviar datos a la nube y al AWS IoT SiteWise procesador para realizar transformaciones y cálculos locales. Tanto el editor como el procesador son AWS

IoT Greengrass V2 componentes. Para obtener más información sobre AWS IoT Greengrass los componentes, consulte [AWS-provided components](#).

Integración con las funciones AWS IoT SiteWise de la nube para ampliar

Funciona con las funciones de la AWS IoT SiteWise nube, lo que amplía los modelos de activos, los análisis y los paneles de control hasta el límite.

En el caso de las pasarelas con un paquete de procesamiento de datos habilitado, puede utilizarlas AWS OpsHub AWS IoT SiteWise para gestionar centralmente las pasarelas SiteWise Edge. AWS OpsHub proporciona capacidades de administración y monitoreo remotos. Para obtener más información, consulte [Administre las puertas de enlace SiteWise Edge mediante AWS OpsHub AWS IoT SiteWise](#).

Integración de fuentes de datos de socios

Conecte una fuente de datos de un socio a su puerta de enlace y reciba datos del socio en su puerta de enlace SiteWise Edge y en la AWS nube. Para obtener más información, consulte [Fuentes de datos asociadas en las puertas de enlace SiteWise Edge](#).

Visualización local en la periferia

Proporciona paneles personalizados para obtener información en tiempo real desde la periferia.

Supervise los datos de forma local en sus instalaciones mediante los portales SiteWise Monitor de sus dispositivos locales. Para obtener más información, consulte [Habilitar su AWS IoT SiteWise portal en la periferia](#).

Ventajas de implementar SiteWise Edge

SiteWise Edge ofrece numerosas ventajas que pueden mejorar significativamente las operaciones industriales y los procesos de toma de decisiones.

- Información operativa en tiempo real sin demoras en el procesamiento en la nube
- Continuidad operativa en entornos desconectados
- Reducción de los costes de ancho de banda y almacenamiento mediante el preprocesamiento perimetral
- Mayor confiabilidad con la capacidad de tomar decisiones locales basadas en datos

Aloje automáticamente una puerta de enlace AWS IoT SiteWise Edge con AWS IoT Greengrass V2

Configure AWS IoT SiteWise Edge para recopilar, procesar y visualizar los datos de los equipos industriales a nivel local antes de enviarlos a la nube. Autohospedaje mediante AWS IoT Greengrass Version 2.

Una puerta de enlace AWS IoT SiteWise Edge actúa como intermediaria entre su equipo industrial y AWS IoT SiteWise. Si se ejecuta AWS IoT Greengrass Version 2, la puerta de enlace SiteWise Edge permite la recopilación y el procesamiento de datos in situ.

Hay dos tipos de puertas de enlace autohospedadas:

Puerta de enlace V3 habilitada para MQTT

La arquitectura de puerta de enlace V3 habilitada para MQTT proporciona capacidades mejoradas de ingesta de datos. Utiliza el protocolo MQTT para una comunicación de datos eficiente y ofrece destinos de datos configurables. Estas incluyen opciones para la ingesta de datos en búfer mediante Amazon S3, así como la ingesta de datos en tiempo real. Puede implementar filtros de ruta para suscribirse a temas específicos de MQTT, lo que permite una recopilación de datos específica. Tenga en cuenta que la puerta de enlace V3 habilitada para MQTT no admite la función de paquete de procesamiento de datos. Para obtener más información, consulte [Puertas de enlace V3 habilitadas para MQTT para Edge AWS IoT SiteWise](#).

Puerta de enlace V2 de transmisiones clásicas

La puerta de enlace Classic Streams, V2, representa la arquitectura de puerta de enlace Edge tradicional AWS IoT SiteWise. Es ideal para las implementaciones de SiteWise Edge existentes y para los usuarios acostumbrados al flujo de trabajo establecido. Si bien la puerta de enlace Classic Streams V2 admite el paquete de procesamiento de datos, tenga en cuenta que los datos generados por el paquete de procesamiento de datos no se pueden ingerir a través de Amazon S3. Utilice la puerta de enlace Classic Streams V2 si necesita mantener la compatibilidad con las implementaciones existentes o si necesita la funcionalidad del paquete de procesamiento de datos. Para obtener más información, consulte [Transmisiones clásicas, pasarelas V2 para Edge AWS IoT SiteWise](#).

Temas

- [AWS IoT SiteWise Requisitos de la puerta de enlace autohospedada de Edge](#)

- [Cree una puerta de enlace SiteWise Edge autohospedada](#)
- [Instale el software AWS IoT SiteWise Edge Gateway en su dispositivo local](#)
- [Puertas de enlace V3 habilitadas para MQTT para Edge AWS IoT SiteWise](#)
- [Transmisiones clásicas, pasarelas V2 para Edge AWS IoT SiteWise](#)
- [Agregue fuentes de datos a su puerta de enlace AWS IoT SiteWise Edge](#)
- [AWS IoT Greengrass componentes para AWS IoT SiteWise Edge](#)
- [Filtre los activos en una puerta de enlace SiteWise Edge](#)
- [Configura el soporte de proxy y administra los almacenes de confianza para AWS IoT SiteWise Edge](#)
- [AWS IoT SiteWise APIs Úselo en el borde](#)

AWS IoT SiteWise Requisitos de la puerta de enlace autohospedada de Edge

AWS IoT SiteWise Las puertas de enlace Edge se ejecutan AWS IoT Greengrass V2 como un conjunto de AWS IoT Greengrass componentes que permiten la recopilación, el procesamiento y la publicación de datos in situ. Para configurar una puerta de enlace SiteWise Edge que se ejecute AWS IoT Greengrass V2, cree una puerta de enlace en Nube de AWS y ejecute el software de puerta de enlace SiteWise Edge para configurar su dispositivo local. Cuando se utiliza Consola de administración de AWS para crear la puerta de enlace SiteWise Edge, se proporciona un script de instalación. Ejecute este script en el dispositivo de puerta de enlace de destino para configurar el software y las dependencias necesarios.

Requisitos del dispositivo local

Los dispositivos locales deben cumplir los siguientes requisitos para instalar y ejecutar el software SiteWise Edge Gateway.

- Es compatible con la versión [2.3.0](#) o posterior del software AWS IoT Greengrass V2 Core. Para obtener más información, consulte [Requisitos](#) en la Guía para desarrolladores de AWS IoT Greengrass Version 2 .
- Una de las siguientes plataformas admitidas:
 - OS: Ubuntu 20.04 o posterior

Arquitectura: x86_64 () o (Aarch64) AMD64 ARMv8

- OS: Red Hat Enterprise Linux (RHEL) 8
Arquitectura: x86_64 () o (Aarch64) AMD64 ARMv8
- OS: Amazon Linux 2
Arquitectura: x86_64 () o (Aarch64) AMD64 ARMv8
- OS: Debian 11
Arquitectura: x86_64 () o (Aarch64) AMD64 ARMv8
- OS: Windows Server 2019 o posterior
Arquitectura: x86_64 () AMD64

Note

Las plataformas ARM solo admiten puertas de enlace SiteWise Edge con el paquete de recopilación de datos. El paquete de procesamiento de datos no es compatible.

- Mínimo de 4 GB de RAM.
- Dispone de un mínimo de 10 GB de espacio en disco para el software SiteWise Edge Gateway.
- Configure su dispositivo local para asegurarse de que se pueda acceder a los puertos adecuados. Para obtener una lista completa de los puntos finales de servicio salientes necesarios, consulte Puntos de enlace de [servicio necesarios para AWS IoT SiteWise las puertas de enlace Edge](#).
- Entorno de ejecución de Java (JRE) versión 11 o superior. Java debe estar disponible en la variable de entorno PATH en el dispositivo. Para utilizar Java para desarrollar componentes personalizados, debe instalar un kit de desarrollo de Java (JDK). Le recomendamos que utilice [Amazon Corretto](#) u [OpenJDK](#).

Lista de buckets de Amazon S3 para dispositivos locales

Configure su dispositivo local para proporcionar acceso al firewall al siguiente bucket de Amazon S3. Configure el acceso de sus dispositivos en función de las regiones respectivas.

Region	Punto de conexión
Asia-Pacífico (Tokio)	https://iot-sitewise-gateway-ap-northeast-1-785558802005.s3.ap-northeast-1.amazonaws.com

Region	Punto de conexión
Asia-Pacífico (Seúl)	https://iot-sitewise-gateway-ap-northeast-2-310055672453.s3.ap-northeast-2.amazonaws.com
Asia-Pacífico (Mumbai)	https://iot-sitewise-gateway-ap-south-1-677656657204.s3.ap-south-1.amazonaws.com
Asia-Pacífico (Singapur)	https://iot-sitewise-gateway-ap-southeast-1-475191558554.s3.ap-southeast-1.amazonaws.com
Asia-Pacífico (Sidney)	https://iot-sitewise-gateway-ap-southeast-2-396319432685.s3.ap-southeast-2.amazonaws.com
Canadá (centro)	https://iot-sitewise-gateway-ca-central-1-842060018567.s3.ca-central-1.amazonaws.com
China (Pekín)	https://iot-sitewise-gateway-cn-north-1-237124890262.s3.cn-north-1.amazonaws.com.cn
Europa (Fráncfort)	https://iot-sitewise-gateway-eu-central-1-748875242063.s3.eu-central-1.amazonaws.com
Europa (Irlanda)	https://iot-sitewise-gateway-eu-west-1-383414315062.s3.eu-west-1.amazonaws.com
Este de EE. UU. (Norte de Virginia)	https://iot-sitewise-gateway-us-east-1-223558168232.s3.us-east-1.amazonaws.com / https://iot-sitewise-gateway-us-east-1-223558168232.s3.amazonaws.com
Este de EE. UU. (Ohio)	https://iot-sitewise-gateway-us-east-2-005072661813.s3.us-east-2.amazonaws.com
AWS GovCloud (EE. UU.-Oeste)	https://iot-sitewise-gateway-us-gov-west-1-599984565679.s3.us-gov-west-1.amazonaws.com/
Oeste de EE. UU. (Oregón)	https://iot-sitewise-gateway-us-west-2-502577205460.s3.us-west-2.amazonaws.com

Requisitos del paquete de procesamiento de datos

Note

La característica de paquete de procesamiento de datos (DPP) ya no está disponible para los nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte [Cambio en la disponibilidad del paquete de procesamiento de datos](#).

- Si planea utilizar el paquete de procesamiento de datos de forma perimetral AWS IoT SiteWise, su dispositivo local también debe cumplir los siguientes requisitos:
 - Tener un procesador x86 de 64 bits y cuatro núcleos.
 - Tener al menos 16 GB de RAM.
 - Tiene al menos 32 GB de RAM si lo usa Microsoft Windows.
 - Tener al menos 256 GB de espacio libre en disco.
 - El dispositivo local debe permitir tráfico de red entrante en el puerto 443.
 - Los siguientes puertos están reservados para su uso por AWS IoT SiteWise: 80, 443, 3001, 4569, 4572, 8000, 8081, 8082, 8084, 8085, 8445, 8086, 9000, 9500, 11080 y 50010. El uso de un puerto reservado para el tráfico puede causar la terminación de la conexión.

Note

AWS IoT Greengrass V2 El componente Stream Manager tiene sus propios requisitos. Para obtener más información, consulte [Configuración](#) en la Guía para desarrolladores de AWS IoT Greengrass Version 2 .

- Los requisitos mínimos de espacio en disco y capacidad de procesamiento dependen de una variedad de factores que son exclusivos de su implementación y caso de uso.
 - El espacio en disco necesario para almacenar en caché los datos para una conectividad a Internet intermitente depende de los siguientes factores:
 - Número de flujos de datos cargados
 - Puntos de datos por flujo de datos por segundo
 - Tamaño de cada punto de datos

- Tiempo de inactividad de red esperado
- La capacidad de cómputo necesaria para sondear y cargar los datos depende de los siguientes factores:
 - Número de flujos de datos cargados
 - Puntos de datos por flujo de datos por segundo

Configure los permisos para usar las puertas de enlace SiteWise Edge

Debe tener los siguientes permisos para usar las puertas de enlace SiteWise Edge:

Note

Si usa la AWS IoT SiteWise consola para crear su puerta de enlace SiteWise Edge, estos permisos se añaden automáticamente.

- La función de IAM de su puerta de enlace SiteWise Edge debe permitirle utilizar una puerta de enlace SiteWise Edge en un AWS IoT Greengrass V2 dispositivo para procesar los datos del modelo de activos y los datos de los activos.

El rol permite que el siguiente servicio asuma el rol: `credentials.iot.amazonaws.com`.

Detalles de los permisos

El rol debe tener los siguientes permisos:

- `iotsitewise`: permite a las entidades principales recuperar datos de modelos de activos y datos de activos en la periferia.
- `iot`— Permite que sus AWS IoT Greengrass V2 dispositivos interactúen con AWS IoT.
- `logs`— Permite que tus AWS IoT Greengrass V2 dispositivos envíen registros a Amazon CloudWatch Logs.
- `s3`— Permite que sus AWS IoT Greengrass V2 dispositivos descarguen artefactos de componentes personalizados de Amazon S3.

JSON

```
{  
  "Version": "2012-10-17",
```

```

"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "iotsitewise:BatchPutAssetPropertyValue",
      "iotsitewise:List*",
      "iotsitewise:Describe*",
      "iotsitewise:Get*"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "iot:DescribeCertificate",
      "logs:CreateLogGroup",
      "logs:CreateLogStream",
      "logs:PutLogEvents",
      "logs:DescribeLogStreams",
      "s3:GetBucketLocation",
      "s3:GetObject",
      "iot:Connect",
      "iot:Publish",
      "iot:Subscribe",
      "iot:Receive",
      "iot:DescribeEndpoint"
    ],
    "Resource": "*"
  }
]
}

```

Cree una puerta de enlace SiteWise Edge autohospedada

Use la AWS IoT SiteWise consola o AWS CLI cree una puerta de enlace SiteWise Edge autohospedada. Este procedimiento detalla cómo crear una puerta de enlace SiteWise Edge autohospedada que instalará en su propio hardware. Para obtener información sobre cómo crear una puerta de enlace SiteWise Edge que se ejecute en Siemens Industrial Edge, consulte [Aloje una puerta de enlace SiteWise Edge en Siemens Industrial Edge](#).

Cree una puerta de enlace SiteWise Edge

Console

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, seleccione Puertas de enlace Edge.
3. Seleccione Crear puerta de enlace.
4. En Elegir destino de implementación, elija una puerta de enlace autohospedada.
5. Seleccione una puerta de enlace V3 habilitada para MQTT o una puerta de enlace V2 Classic Streams. Para obtener más información sobre cada opción, consulte. [Aloje automáticamente una puerta de enlace AWS IoT SiteWise Edge con AWS IoT Greengrass V2](#) Se recomienda la puerta de enlace V3 habilitada para MQTT por sus funciones preparadas para el futuro.
6. En la sección de configuración de la puerta de enlace, introduzca un nombre para la puerta de enlace SiteWise Edge o utilice el nombre generado por. AWS IoT SiteWise
7. En el sistema operativo del dispositivo Greengrass, selecciona el sistema operativo del dispositivo en el que instalarás esta puerta de enlace SiteWise Edge.

Note

El paquete de procesamiento de datos solo está disponible en plataformas x86. Solo está disponible en la puerta de enlace Classic Streams V2


8. (Opcional) Para procesar y organizar los datos en la periferia, en Capacidades de periferia, seleccione Paquete de procesamiento de datos.

Note

Para conceder a los grupos de usuarios de su directorio corporativo acceso a esta puerta de enlace SiteWise Edge, consulte [Configure la capacidad SiteWise perimetral en Edge](#)

9. (Opcional) En Configuración avanzada, haga lo siguiente:
 - En Dispositivo central de Greengrass, elija una de las siguientes opciones:

- Configuración predeterminada: utiliza AWS automáticamente la configuración predeterminada para crear un dispositivo principal de Greengrass. AWS IoT Greengrass V2
 1. Introduzca un nombre para el dispositivo principal de Greengrass o utilice el nombre generado por. AWS IoT SiteWise
 - Configuración avanzada: elija esta opción si desea utilizar un dispositivo de núcleo de Greengrass existente o crear uno manualmente.
 1. Elija un dispositivo principal de Greengrass o elija Crear dispositivo principal de Greengrass para crear uno en la consola. AWS IoT Greengrass V2 Para obtener más información, consulte [Configuración de dispositivos centrales de AWS IoT Greengrass V2](#) en la Guía para desarrolladores de AWS IoT Greengrass Version 2 .
10. Seleccione Crear puerta de enlace.
 11. En el cuadro de diálogo del instalador de la puerta de enlace Generate SiteWise Edge, seleccione Generar y descargar. AWS IoT SiteWise genera automáticamente un instalador que puede usar para configurar su dispositivo local.

 Important

No puede volver a generar este archivo. Asegúrese de guardar el archivo del instalador en una ubicación segura porque lo utilizará más adelante.

AWS CLI

Para crear una puerta de enlace autohospedada mediante AWS CLI, proporcione un nombre para la puerta de enlace, especifique la plataforma y la versión de la puerta de enlace. Hay muchas otras opciones que puede especificar al crear una puerta de enlace. Para obtener más información, consulte [create-gateway](#) en la Referencia de AWS CLI comandos de AWS IoT SiteWise

Para utilizar este ejemplo, reemplace los marcadores de posición de entrada del usuario con su propia información.

```
aws iotsitewise create-gateway \  
  --gateway-name your-gateway-name \  
  --gateway-platform greengrassV2={coreDeviceThingName=your-core-device-thing-name, coreDeviceOperatingSystem=LINUX_AMD64} \  
  \
```

```
--gateway-version 3 \  
[--cli-input-json your-configuration]
```

- `gateway-name`— Un nombre único para la puerta de enlace.
- `gateway-platform`— Especifica la configuración de la plataforma de la puerta de enlace. Para las puertas de enlace autohospedadas, introduzca `greengrassV2`. Para obtener más información, consulte [las opciones](#) en la sección `create-gateway` de la Referencia de AWS CLI comandos para `AWS IoT SiteWise`.
- `coreDeviceThingName`— El nombre del AWS IoT para tu dispositivo AWS IoT Greengrass V2 principal.
- `coreDeviceOperatingSystem`— El sistema operativo del dispositivo principal en AWS IoT Greengrass V2. Especificar el sistema operativo es obligatorio `gateway-version 3` y no aplicable para `gateway-version 2`. Las opciones incluyen: `LINUX_AARCH64`, `LINUX_AMD64`, y `WINDOWS_AMD64`.
- `gateway-version`— La versión de la puerta de enlace.
 - Para crear una puerta de enlace V3 compatible con MQTT, utilícela `3` para la versión de puerta de enlace.
 - Para crear una puerta de enlace Classic Streams, V2, utilícela `2` para la versión de puerta de enlace.
- `cli-input-json`— Un archivo JSON que contiene los parámetros de la solicitud.

Ahora que ha creado la puerta de enlace SiteWise Edge, [instale el software AWS IoT SiteWise Edge Gateway en su dispositivo local](#).

Instale el software AWS IoT SiteWise Edge Gateway en su dispositivo local

Después de crear una puerta de enlace AWS IoT SiteWise Edge, instale el software de puerta de enlace SiteWise Edge en su dispositivo local. El software de puerta de enlace Edge se puede instalar en dispositivos locales que tengan instalados sistemas operativos Linux o de Microsoft Windows servidor.

Important

Asegúrese de que su dispositivo local se conecte a Internet.

Linux

El siguiente procedimiento utiliza SSH para conectarse a su dispositivo local. También puede utilizar una memoria flash USB u otra herramienta para transferir el archivo del instalador a su dispositivo local. Si no quieres usar SSH, ve al paso 2: instala el software de puerta de enlace SiteWise Edge que se muestra a continuación.

Requisitos previos de SSH

Antes de conectarse a su dispositivo mediante SSH debe cumplir los siguientes requisitos previos.

- Linux y macOS: descargue e instale OpenSSH. Para obtener más información, consulte <https://www.openssh.com>.

Paso 1: Copie el instalador en su dispositivo de puerta de enlace SiteWise Edge

En las siguientes instrucciones se explica cómo conectarse al dispositivo local mediante un cliente SSH.

1. Para conectarse a su dispositivo, ejecute el siguiente comando en una ventana de terminal de su ordenador *username* y sustituya y por un nombre de usuario y una dirección IP *IP* con privilegios elevados.

```
ssh username@IP
```

2. Para transferir el archivo de instalación que AWS IoT SiteWise se generó a su dispositivo de puerta de enlace SiteWise Edge, ejecute el siguiente comando.

Note

- *path-to-saved-installer* Reemplácelo por la ruta de su computadora que utilizó para guardar el archivo de instalación y el nombre del archivo de instalación.
- *IP-address* Sustitúyala por la dirección IP de tu dispositivo local.
- *directory-to-receive-installer* Sustitúyala por la ruta del dispositivo local que utilizaste para recibir el archivo de instalación.

```
scp path-to-saved-installer.sh user-name@IP-address:directory-to-receive-installer
```

Paso 2: Instale el software SiteWise Edge Gateway

En los siguientes procedimientos, ejecute los comandos en una ventana de terminal de su dispositivo de puerta de enlace SiteWise Edge.

1. Otorgue al archivo instalador el permiso de ejecución.

```
chmod +x path-to-installer.sh
```

2. Ejecute el instalador.

```
sudo ./path-to-installer.sh
```

Windows Server

Requisitos previos

Debe cumplir los siguientes requisitos previos para instalar el software de puerta de enlace SiteWise Edge:

- Microsoft Windows Servidor 2019 o posterior instalado
- Privilegios de administrador
- PowerShell instalada la versión 5.1 o posterior
- SiteWise El instalador de Edge Gateway se descargó en el servidor Windows donde se aprovisionará

Paso 1: Ejecute PowerShell como administrador

1. En el servidor Windows en el que desee instalar SiteWise Edge Gateway, inicie sesión como administrador.
2. Ingrese PowerShell en la barra de búsqueda de Windows.

3. En los resultados de la búsqueda, abre el menú contextual (con el botón derecho) de la PowerShell aplicación de Windows. Seleccione Ejecutar como administrador.

Paso 2: Instale el software SiteWise Edge Gateway

Ejecute los siguientes comandos en una ventana de terminal de su dispositivo SiteWise Edge Gateway.

1. Desbloquee el instalador de SiteWise Edge Gateway.

```
unblock-file path-to-installer.ps1
```

2. Ejecute el instalador.

```
./path-to-installer.ps1
```

Note

Si la ejecución del script está deshabilitada en el sistema, cambie la política de ejecución del script a RemoteSigned.

```
Set-ExecutionPolicy RemoteSigned
```

El siguiente paso depende del tipo de puerta de enlace autohospedada que necesite. Continúe hasta [Puertas de enlace V3 habilitadas para MQTT para Edge AWS IoT SiteWise](#) o [Transmisiones clásicas, pasarelas V2 para Edge AWS IoT SiteWise](#)

Puertas de enlace V3 habilitadas para MQTT para Edge AWS IoT SiteWise

AWS IoT SiteWise puede utilizar puertas de enlace V3 habilitadas para MQTT, lo que representa un avance significativo en la arquitectura de puerta de enlace Edge. SiteWise Este tipo de puerta de enlace aprovecha el protocolo MQTT (Message Queuing Telemetry Transport) para la comunicación de datos, lo que proporciona una mayor flexibilidad y eficiencia en las implementaciones de IoT industrial.

La puerta de enlace V3, compatible con MQTT, utiliza MQTT para la transferencia de datos, lo que permite un protocolo de red ligero de publicación y suscripción que transporta los mensajes de

manera eficiente entre los dispositivos y la nube. Puede configurar varios destinos de datos, incluida la ingesta de datos en tiempo real directamente AWS IoT SiteWise y la ingesta de datos en búfer mediante Amazon S3. Para permitir una recopilación de datos precisa, puede implementar filtros de ruta para suscribirse a temas específicos de MQTT.

Las pasarelas V3, habilitadas para MQTT, vienen con un destino en tiempo real preconfigurado con filtros configurados en «#» (todos los temas), que puede personalizar o eliminar según sea necesario. Para agilizar la administración de datos, solo puede existir un destino en tiempo real en cada puerta de enlace.

La arquitectura compatible con MQTT difiere considerablemente de la pasarela Classic Streams, V2. Mientras que la V2 utiliza un enfoque basado en flujos, la V3 emplea MQTT, que ofrece destinos de datos y opciones de filtrado más configurables. Sin embargo, tenga en cuenta que la V3 no es compatible con el paquete de procesamiento de datos, que está disponible en la V2.

La puerta de enlace V3 habilitada para MQTT ofrece varias ventajas:

- Escalabilidad mejorada, debido a la naturaleza liviana del MQTT, que permite un mejor manejo de numerosos dispositivos y una transmisión de datos de alta frecuencia.
- Control de datos mejorado mediante filtros de ruta, lo que permite una gestión pormenorizada de la recopilación de datos y reduce la transferencia y el procesamiento innecesarios de datos.
- Gestión flexible de los datos, que permite configurar entre el procesamiento en tiempo real y el almacenamiento en búfer en función de las necesidades específicas.
- Alineación con los estándares modernos de comunicación de IoT, preparando el terreno para futuras mejoras e integraciones.

Considere la posibilidad de adoptar la puerta de enlace V3 compatible con MQTT para las nuevas implementaciones, especialmente cuando necesite opciones flexibles de ingesta de datos y un control preciso de la recopilación de datos.

Note

Para las implementaciones o escenarios existentes que requieren el paquete de procesamiento de datos, la pasarela Classic Streams y V2 sigue siendo una opción viable.

Al ofrecer ambos tipos de puertas de enlace, AWS IoT SiteWise garantiza que pueda elegir la solución que mejor se adapte a sus necesidades específicas de IoT industrial, ya sea que priorice las capacidades avanzadas de MQTT o la compatibilidad con los sistemas existentes.

Filtros de destinos y rutas

Consulte los siguientes temas para obtener más información sobre los filtros de destinos y rutas en las pasarelas habilitadas para MQTT:

- [Comprenda los destinos de AWS IoT SiteWise Edge](#)
- [Agregue un destino de AWS IoT SiteWise Edge en tiempo real](#)
- [Añadir un destino almacenado en AWS IoT SiteWise búfer mediante Amazon S3](#)
- [Conozca los filtros de ruta para los destinos de AWS IoT SiteWise Edge](#)
- [Agregue filtros de ruta a los destinos de AWS IoT SiteWise Edge](#)
- [Administre los destinos de AWS IoT SiteWise Edge](#)

Connect aplicaciones externas al broker EMQX

Esta guía explica cómo conectar aplicaciones externas a su puerta de enlace AWS IoT SiteWise Edge a través de un intermediario EMQX en su puerta de enlace V3 implementada y habilitada para MQTT. Las aplicaciones externas pueden incluir herramientas de monitoreo personalizadas, software de visualización de terceros o sistemas heredados que necesitan interactuar con sus datos industriales en la periferia.

Abordaremos los pasos de configuración tanto para Linux como para Microsoft Windows entornos, incluida la configuración de implementación de EMQX, la configuración de TLS para conexiones seguras y las reglas de autorización para controlar el acceso a temas específicos.

Note

EMQX no es un vendedor ni proveedor de Edge. AWS IoT SiteWise

Important

Para proteger las conexiones a su puerta de enlace, le recomendamos encarecidamente que utilice la autenticación basada en certificados a través de la función de autenticación del dispositivo AWS IoT Greengrass cliente. Este método proporciona una seguridad sólida

mediante la autenticación TLS mutua (mTLS). Para obtener más información, consulte [Conectar dispositivos cliente a dispositivos principales](#) en la Guía para AWS IoT Greengrass Version 2 desarrolladores.

Si no puede utilizar la autenticación basada en certificados, siga esta guía para configurar la autenticación mediante nombres de usuario y contraseñas.

Requisitos previos

- Una puerta de enlace V3 habilitada para SiteWise Edge MQTT que se ha implementado y está en línea
- Acceso al host de la puerta de enlace
- Acceso a las AWS IoT Greengrass consolas AWS IoT SiteWise y

Temas

- [Formato de carga útil de mensajes para el broker EMQX en Edge AWS IoT SiteWise](#)
- [Configure el bróker EMQX](#)
- [Conecte una aplicación al bróker EMQX en Edge AWS IoT SiteWise](#)
- [Configure las reglas de autorización para AWS IoT SiteWise Edge en EMQX](#)

Formato de carga útil de mensajes para el broker EMQX en Edge AWS IoT SiteWise

Para que el componente de SiteWise editor de IoT consuma datos de su aplicación externa y los publique AWS IoT SiteWise en la nube, la carga útil enviada al intermediario debe cumplir requisitos específicos.

Comprender el formato de la carga útil es clave para una comunicación exitosa de MQTT con Edge. AWS IoT SiteWise Si bien el proceso de configuración de la conexión se explica en secciones posteriores, primero presentamos los requisitos de carga útil para ayudarlo a planificar su implementación.

Requisitos del tema MQTT

No hay restricciones en la estructura de los temas de MQTT, incluida la cantidad de niveles o caracteres utilizados. Sin embargo, recomendamos que el tema coincida con el `propertyAlias` campo de la carga útil.

Example Ejemplo de alias de propiedad

Si el tema de MQTT es `site1/line1/compressor1/temperature`, asegúrese de que `propertyAlias` coincidan.

```
{
  "assetId": "compressor_asset_01",
  "propertyAlias": "site1/line1/compressor1/temperature",
  "propertyId": "temperature_sensor_01",
  "propertyValues": [
    {
      "quality": "GOOD",
      "timestamp": {
        "offsetInNanos": 0,
        "timeInSeconds": 1683000000
      },
      "value": {
        "doubleValue": 23.5
      }
    }
  ]
}
```

Estructura de carga útil de JSON

[La carga útil de los mensajes MQTT está escrita en JSON y sigue el formato de PutAssetPropertyValueEntry](#) mensaje definido en la referencia de la AWS IoT SiteWise API.

```
{
  "assetId": "string",
  "propertyAlias": "string",
  "propertyId": "string",
  "propertyValues": [
    {
      "quality": "string",
      "timestamp": {
        "offsetInNanos": number,
        "timeInSeconds": number
      },
      "value": {
        "booleanValue": boolean,
        "doubleValue": number,
        "integerValue": number,

```

```
        "stringValue": "string"
      }
    }
  ]
}
```

Note

Para que un mensaje se considere válido, solo se puede cumplir una de las siguientes condiciones:

- El `propertyAlias` está establecido, o
- Ambos `assetId` y `propertyId` están configurados

`PutAssetPropertyValueEntry` tiene un `entryId` campo que no es obligatorio en este contexto.

Configure el bróker EMQX

En esta sección se explica cómo añadir nombres de usuario y contraseñas. También explica cómo establecer una conexión TLS desde una fuente externa utilizando el nombre de usuario y la contraseña agregados. Puede configurar el bróker EMQX mediante Linux o Microsoft Windows

Note

Para configurar el bróker, necesita un dispositivo central que esté configurado con la configuración EMQX predeterminada en su puerta de enlace V3 habilitada para MQTT.

Important

Tras completar este procedimiento, le recomendamos encarecidamente que configure las reglas de autorización. Para obtener más información, consulte [Configure las reglas de autorización para AWS IoT SiteWise Edge en EMQX](#). Las reglas de autorización para los usuarios adicionales mejoran la seguridad.

Actualice la configuración de despliegue de EMQX para la autenticación

Para actualizar la configuración de despliegue de EMQX para la autenticación

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación de la izquierda, elija las puertas de enlace Edge en la sección Edge.
3. Elija la puerta de enlace que desee configurar.
4. En la sección de configuración de la puerta de enlace Edge, copia el valor de tu dispositivo principal de Greengrass. Guárdelo para usarlo más adelante.
5. Abra la [consola de AWS IoT](#).
6. En el menú de navegación de la izquierda, en la sección Administrar, selecciona dispositivos Greengrass y, a continuación, Implementaciones.
7. Busque el valor del dispositivo principal que guardó anteriormente y elija ese enlace para abrir la implementación.
8. Selecciona el botón desplegable Acciones y, a continuación, Revisa.
9. Lea el mensaje que aparece y, a continuación, seleccione Revisar el despliegue. Aparece la página Especificar el destino.
10. Elija Siguiente hasta llegar al paso Configurar componentes.
11. Seleccione el botón de opción `aws.greengrass.clientdevices.mqtt.EMQX`.
12. Pulse el botón Configurar componente. Aparece una página de configuración para el componente.
13. En Actualización de la configuración, seleccione Restablecer la configuración predeterminada para la versión del componente: `2.*.*`.
14. Introduzca la siguiente configuración en la sección Configuración para fusionar en función de su sistema operativo.

Linux

```
{
  "emqxConfig": {
    "authorization": {
      "no_match": "allow"
    },
    "listeners": {
      "tcp": {
        "default": {
```

```

        "enabled": true,
        "enable_authn": false
    }
},
"ssl": {
    "default": {
        "enabled": true,
        "enable_authn": true,
        "ssl_options": {
            "verify": "verify_none",
            "fail_if_no_peer_cert": false
        }
    }
},
},
"authentication": {
    "enable": true,
    "backend": "built_in_database",
    "mechanism": "password_based",
    "password_hash_algorithm": {
        "iterations": 210000,
        "mac_fun": "sha512",
        "name": "pbkdf2"
    },
    "user_id_type": "username"
},
"dashboard": {
    "listeners": {
        "http": {
            "bind": 18083
        }
    }
},
},
"authMode": "bypass",
"dockerOptions": "-p 8883:8883 -p 127.0.0.1:1883:1883
-p 127.0.0.1:18083:18083 -v emqx-data:/opt/emqx/data -e
EMQX_NODE_NAME=emqx@local",
"requiresPrivilege": "true"
}

```

Windows

```
{
  "emqxConfig": {
    "authorization": {
      "no_match": "allow"
    },
    "listeners": {
      "tcp": {
        "default": {
          "enabled": true,
          "enable_authn": false
        }
      },
      "ssl": {
        "default": {
          "enabled": true,
          "enable_authn": true,
          "ssl_options": {
            "verify": "verify_none",
            "fail_if_no_peer_cert": false
          }
        }
      }
    },
    "authentication": {
      "enable": true,
      "backend": "built_in_database",
      "mechanism": "password_based",
      "password_hash_algorithm": {
        "iterations": 210000,
        "mac_fun": "sha512",
        "name": "pbkdf2"
      },
      "user_id_type": "username"
    },
    "dashboard": {
      "listeners": {
        "http": {
          "bind": 18083
        }
      }
    }
  }
}
```

```
  },  
  "authMode": "bypass",  
  "requiresPrivilege": "true"  
}
```

El `dockerOptions` campo es solo para pasarelas de Linux.

15. Elija Confirmar.
16. Seleccione Siguiente hasta llegar al paso de revisión.
17. Elija Implementar.
18. Una vez que la implementación se haya realizado correctamente, continúe con el siguiente paso.

Habilite la autenticación de nombre de usuario y contraseña

En esta sección, se muestra cómo añadir nombres de usuario y contraseñas a través de la GUI del panel de EMQX.

Note


Las instrucciones relacionadas con EMQX que se proporcionan son solo de referencia. Como la documentación y las funciones de EMQX pueden cambiar con el tiempo y no mantenemos su documentación, recomendamos consultar la documentación [oficial de EMQX para obtener la](#) información más actualizada.

EMQX Dashboard

Para habilitar la autenticación de nombre de usuario y contraseña a través del panel de EMQX

1. Asegúrese de estar dentro del host de la puerta de enlace.
2. Abra una ventana del navegador y visite <http://localhost:18083/>.
3. Introduzca el nombre de usuario **admin** y la contraseña predeterminados de **public**. Para obtener más información, consulte el [panel de control de EMQX](#) en los documentos de EMQX.
4. Después de iniciar sesión, se le solicitará que cambie su contraseña. Actualice su contraseña para continuar con el panel de control de EMQX.
5. En el menú de navegación de la izquierda, seleccione el icono del escudo y, a continuación, Autenticación.

6. En la fila Base de datos integrada, pulse el botón Usuarios.
7. Pulse el botón con el icono del signo más para añadir usuarios. Aparece la pantalla Añadir.
8. Introduzca un nombre de usuario y una contraseña para el usuario de la aplicación externa.
9. Seleccione Save. El nombre de usuario que eligió aparece en la tabla de la página de autenticación.

 Note

Las reglas de autorización existentes o predeterminadas se aplican al nuevo usuario. Se recomienda revisarlas y ajustarlas a las necesidades de las aplicaciones externas.

EMQX Management with Linux

Utilice la herramienta AWS IoT SiteWise CLI de EMQX en. `/greengrass/v2/bin/swe-emqx-cli`

Para habilitar la autenticación de nombre de usuario y contraseña mediante EMQX Management mediante Linux

1. Cambie la contraseña de administrador ejecutando el siguiente comando:

```
/greengrass/v2/bin/swe-emqx-cli admin change-pwd
```

2. Cuando se le solicite, haga lo siguiente:
 1. Introduzca su usuario administrador actual (el predeterminado `esadmin`) y la contraseña (el predeterminado `espublic`).
 2. Introduzca y confirme su nueva contraseña.

Si se ejecuta correctamente, aparecerá el siguiente mensaje:

```
admin password changed successfully
```

3. Añada usuarios para aplicaciones externas ejecutando el siguiente comando:

```
/greengrass/v2/bin/swe-emqx-cli users add
```

4. Cuando se le solicite, haga lo siguiente:
 1. Introduzca el nombre de usuario del nuevo usuario.
 2. Introduzca y confirme la contraseña del nuevo usuario.

Si se ejecuta correctamente, aparecerá el siguiente mensaje:

```
User '[username]' created successfully
```

5. Compruebe la configuración del usuario ejecutando el siguiente comando:

```
/greengrass/v2/bin/swe-emqx-cli users list
```

El resultado muestra todos los usuarios configurados:

```
Users:  
- [your-added-username]  
  
Total users: 1
```

EMQX Management with Windows

Utilice la herramienta AWS IoT SiteWise CLI de EMQX en una de las siguientes ubicaciones:

- PowerShell: `C:\greengrass\v2\bin\swe-emqx-cli.ps1`
- Línea de comandos: `C:\greengrass\v2\bin\swe-emqx-cli.bat`

Para habilitar la autenticación de nombre de usuario y contraseña mediante la administración de EMQX mediante Windows

1. Cambie la contraseña de administrador ejecutando el siguiente comando:

```
C:\greengrass\v2\bin\swe-emqx-cli.ps1 admin change-pwd
```

2. Cuando se le solicite, haga lo siguiente:
 1. Introduzca su usuario administrador actual (el predeterminado `esadmin`) y la contraseña (el predeterminado `espublic`).

2. Introduzca y confirme su nueva contraseña.

Si se ejecuta correctamente, aparecerá el siguiente mensaje:

```
admin password changed successfully
```

3. Añada usuarios para aplicaciones externas ejecutando el siguiente comando:

```
C:\greengrass\v2\bin\swe-emqx-cli.ps1 users add
```

4. Cuando se le solicite, haga lo siguiente:

1. Introduzca el nombre de usuario del nuevo usuario.
2. Introduzca y confirme la contraseña del nuevo usuario.

Si se ejecuta correctamente, aparecerá el siguiente mensaje:

```
User '[username]' created successfully
```

5. Compruebe la configuración del usuario ejecutando el siguiente comando:

```
C:\greengrass\v2\bin\swe-emqx-cli.ps1 users list
```

El resultado muestra todos los usuarios configurados:

```
Users:  
- [your-added-username]  
  
Total users: 1
```

Conecte una aplicación al bróker EMQX en Edge AWS IoT SiteWise

El broker EMQX utiliza Transport Layer Security (TLS) en el puerto 8883 para cifrar todas las comunicaciones, lo que garantiza que sus datos permanezcan protegidos durante la transmisión. En esta sección, se explican los pasos para establecer conexiones entre sus aplicaciones y el bróker EMQX. Seguir estos pasos ayuda a mantener la integridad y confidencialidad de sus datos industriales. El proceso de conexión implica dos enfoques principales: utilizar la detección automática de IP a través de componentes o configurar manualmente los nombres DNS y las direcciones

IP como nombres alternativos del sujeto (SANs) en los certificados TLS. Cada método tiene sus propias ventajas en función de la configuración de la red y de los requisitos de seguridad. Esta documentación lo guiará a través de ambas opciones.

Temas

- [Configure TLS para establecer conexiones seguras con el intermediario EMQX en Edge AWS IoT SiteWise](#)
- [Pruebe la conexión del broker EMQX en Edge AWS IoT SiteWise](#)
- [Utilice su propia CA](#)
- [Abra el puerto 8883 para las conexiones de firewall externas](#)

Configure TLS para establecer conexiones seguras con el intermediario EMQX en Edge AWS IoT SiteWise

De forma predeterminada, AWS IoT Greengrass genera un certificado de servidor TLS para el broker EMQX firmado por la autoridad de certificación (CA) del dispositivo principal. Para obtener más información, consulte [Conectar dispositivos cliente a un dispositivo AWS IoT Greengrass Core con un agente MQTT](#).

Recupere el certificado TLS

Para obtener el certificado de CA, ejecute el siguiente comando en el host de la puerta de enlace:

Linux

Ejecute el siguiente comando en una sesión de shell en el host de la puerta de enlace:

```
/greengrass/v2/bin/swe-emqx-cli cert
```

Este comando muestra la ubicación del certificado e imprime su contenido.

También puede guardar el certificado en un archivo mediante este comando:

```
/greengrass/v2/bin/swe-emqx-cli cert --output /path/to/certificate.pem
```

Windows

Ejecute el siguiente comando en una PowerShell sesión en el host de la puerta de enlace:

```
C:\greengrass\v2\bin\swe-emqx-cli.ps1 cert
```

Este comando muestra la ubicación del certificado e imprime su contenido.

También puede guardar el certificado en un archivo mediante este comando:

```
C:\greengrass\v2\bin\swe-emqx-cli.ps1 cert --output C:\path\to\certificate.pem
```

La CLI localiza automáticamente el certificado independientemente de la ruta exacta del sistema.

Copie el contenido del archivo `ca.pem` a la aplicación externa que va a conectar al agente. Guárdelo como `BrokerCoreDeviceCA.pem`.

Agregue nombres/IP direcciones DNS personalizadas al certificado del servidor TLS

El nombre alternativo (SAN) del sujeto del certificado generado por AWS IoT Greengrass es `localhost`. Al establecer una conexión TLS desde fuera del host de la puerta de enlace, se produce un error en el paso de verificación de TLS porque el nombre de host del intermediario no coincide con el nombre de host del certificado del `localhost` servidor.

Para solucionar el problema de los nombres de host no coincidentes, AWS IoT Greengrass proporciona dos formas de administrar los puntos finales de los dispositivos principales. En esta sección se describen ambas opciones. Para obtener información más detallada, consulte [Administrar los terminales de los dispositivos principales](#) en la Guía para AWS IoT Greengrass Version 2 desarrolladores.

- Para conectarse al agente EMQX mediante la dirección IP del dispositivo principal, utilice la sección [Detección automática de IP](#).
- Para conectarse al agente EMQX mediante un nombre DNS en lugar de una dirección IP, utilice la sección [de administración manual](#).

Automated IP discovery

Esta opción permite que el dispositivo principal descubra automáticamente su dirección IP y la añada como nombre alternativo del sujeto (SAN) al certificado del agente.

1. Añada el `aws.greengrass.clientdevices.IPDetector` componente a la implementación de su dispositivo principal.

2. Implemente los cambios en su dispositivo
3. Espere a que se complete el despliegue.

Una vez completada la implementación, puede establecer una conexión TLS segura mediante la dirección IP del intermediario.

La dirección IP se agrega automáticamente como SAN al certificado del corredor.

Manual DNS and IP Configuration

Puede agregar manualmente nombres DNS y direcciones IP como nombres alternativos del asunto (SANs) a su certificado TLS. Este método resulta útil cuando ha configurado un nombre DNS para el host de la puerta de enlace.

Important

Si utiliza el IPDetector componente, elimínelo de la implementación antes de continuar. El IPDetector componente anula las configuraciones manuales de los puntos finales.

Para configurar manualmente los puntos finales

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación de la izquierda, elija las puertas de enlace Edge en la sección Edge.
3. Elija la puerta de enlace que desee configurar.
4. En la sección de configuración de la puerta de enlace Edge, elija la URL de su dispositivo principal de Greengrass. Aparece la página del dispositivo principal.
5. Seleccione la pestaña Dispositivos cliente.
6. Seleccione Administrar puntos de conexión.
7. En el cuadro de diálogo Administrar puntos de conexión, introduzca los nombres DNS y las direcciones IP que desee SANs añadir. Utilice el puerto 8883.
8. Elija Actualizar.

El certificado del servidor TLS del bróker se actualiza automáticamente para incluir sus nuevos puntos de conexión.

Para verificar la actualización del certificado del servidor TLS mediante Linux

1. Inicie una sesión de shell en el host de la puerta de enlace.

```
docker exec emqx openssl x509 -in ./data/cert.pem -text -noout | grep -A1
"Subject Alternative Name"
```

2. El comando devuelve un resultado similar al siguiente:

```
X509v3 Subject Alternative Name:
DNS: endpoint_you_added, DNS:localhost
```

3. Compruebe que su punto final aparezca en la lista de SANs.

Para comprobar la actualización del certificado del servidor TLS mediante Windows

1. Inicie una sesión de shell en el host de la puerta de enlace.

```
(Get-PfxCertificate -FilePath "C:\greengrass\v2\work
\aws.greengrass.clientdevices.mqtt.EMQX\v2\data\cert.pem").Extensions | Where-
Object { $_.Oid.FriendlyName -eq "Subject Alternative Name" } | ForEach-Object
{ "Subject Alternative Name:", ($_.Format($true) -split "`n")[0..1] }
```

2. El comando devuelve un resultado similar al siguiente:

```
Subject Alternative Name:
DNS Name=your-endpoint
DNS Name=localhost
```

3. Compruebe que el punto final que agregó esté en la lista de SANs.

Pruebe la conexión del broker EMQX en Edge AWS IoT SiteWise

Tras configurar su agente EMQX con certificados TLS y credenciales de autenticación, es importante comprobar que la configuración funciona correctamente. Probar la conexión ayuda a garantizar que las configuraciones de seguridad se implementen correctamente y que los clientes puedan establecer correctamente conexiones cifradas con el intermediario. En esta sección se muestra cómo probar la conexión de su intermediario mediante el cliente de interfaz de línea de comandos (CLI) de Mosquitto, una herramienta de cliente MQTT muy utilizada que admite el cifrado y la autenticación TLS.

Utilice el cliente CLI de Mosquitto para probar la conexión del broker EMQX

En este paso, utilizaremos el cliente CLI mosquitto para probar nuestra configuración y asegurarnos de que podemos conectarnos correctamente al corredor con el nombre de usuario y la contraseña que creamos anteriormente. Para obtener los `BrokerCoreDeviceCA.pem` siguientes pasos, consulte el Paso 3: Configuración de TLS.

```
mosquitto_sub -h hostname/ip address \  
-p 8883 \  
-t "#" \  
-q 1 \  
-u username -P password \  
--cafile BrokerCoreDeviceCA.pem
```

Note

Es posible que aparezca un error `SSL:Verify` si la `hostname/IP` dirección a la que se está conectando no coincide con el nombre alternativo (SAN) del sujeto que figura en el certificado de CA que está transmitiendo al cliente. Consulte la sección «Añadir names/IP direcciones DNS personalizadas al certificado del servidor TLS» en el Paso 3: Configuración de TLS para obtener un certificado con la SAN correcta.

En este punto, todos los usuarios tienen acceso a publicar y suscribirse a todos los temas del intermediario. Continúe en [Configure las reglas de autorización para AWS IoT SiteWise Edge en EMQX](#).

Utilice su propia CA

AWS IoT Greengrass describe cómo configurar su propio componente de autenticación del dispositivo cliente para que utilice su propia autoridad de certificación (CA). El componente de autenticación del dispositivo de cliente (`aws.greengrass.clientdevices.Auth`) autentica los dispositivos de cliente y autoriza las acciones de los dispositivos de cliente. Para obtener más información, consulte [Uso de su propia autoridad de certificación](#) en la Guía para AWS IoT Greengrass Version 2 desarrolladores.

Para usar su propia CA, añada el `aws.greengrass.clientdevices.Auth` componente a su implementación para poder especificar una configuración personalizada.

Abra el puerto 8883 para las conexiones de firewall externas

Linux

En la regla de firewall de su host Linux, añada una regla de entrada para el puerto 8883 a fin de permitir las conexiones entrantes desde fuera del host de la puerta de enlace. Si hay algún firewall instalado, asegúrese de que se permitan las conexiones TLS entrantes en el puerto 8883.

Windows

En la regla de firewall del Microsoft Windows host, añada una regla de entrada para el puerto 8883 a fin de permitir las conexiones entrantes desde fuera del host de la puerta de enlace. Asegúrese de que la regla sea una regla de permiso, del tipo puerto, que especifique el puerto 8883. Puede configurarlo de acuerdo con la configuración de su red para permitir las conexiones desde sus aplicaciones externas al intermediario.

Configure las reglas de autorización para AWS IoT SiteWise Edge en EMQX

EMQX permite agregar reglas de autorización basadas en identificadores como el nombre de usuario, la dirección IP o el ID de cliente. Esto resulta útil si desea limitar el número de aplicaciones externas que se conectan a diversas operaciones o temas.

Temas

- [Configure la autorización mediante la base de datos integrada con Linux](#)
- [Configure la autorización mediante la base de datos integrada con Windows](#)
- [Actualice la configuración de despliegue del EMQX para su autorización](#)
- [Agregue reglas de autorización a través del panel de control de EMQX para los usuarios](#)

Configure la autorización mediante la base de datos integrada con Linux

Al configurar las reglas de autorización, hay dos opciones de configuración que dependen de la configuración de la implementación.

- Docker— Si ejecuta una Docker instalación estándar sin ella Litmus Edge, utilice la configuración de puerta de enlace Docker Bridge. Este suele ser el caso cuando solo se han implementado AWS IoT SiteWise componentes.
- Litmus Edge— Si los ha Litmus Edge instalado en su puerta de enlace, utilice la configuración de Litmus Edgesubred de la red.

Note

Si configura inicialmente la puerta de enlace Docker puente y la instala posteriormente Litmus Edge, vuelva a configurar las reglas de autorización mediante la opción de Litmus Edge subred de red para garantizar una comunicación adecuada entre todos los componentes.

Para agregar reglas de autorización básicas

1. Compruebe que el bróker EMQX esté desplegado y en funcionamiento.
2. Inicie una sesión de shell en el host de su puerta de enlace.

Docker without Litmus Edge

Para una Docker instalación estándar sin Litmus Edge ella, ejecute:

```
/greengrass/v2/bin/swe-emqx-cli acl init
```

Litmus Edge network subnet

Si estás utilizando Litmus Edge, determina la IP de la subred de la red de Litmus Edge:

```
docker network inspect LitmusNetwork | grep IPAM -A9
```

Anota el valor de subred del resultado y ejecuta el siguiente comando.

`litmus_subnet_ip` Sustitúyalo por el valor de subred del paso anterior.

```
/greengrass/v2/bin/swe-emqx-cli acl init litmus_subnet_ip
```

La herramienta crea y aplica automáticamente reglas de autorización para permitir las conexiones desde la dirección IP proporcionada al intermediario. Permite el acceso a todos los temas. Esto incluye el recopilador IoT SiteWise OPC UA y el SiteWise editor de IoT.

3. Continúe en [Actualice la configuración de despliegue del EMQX para su autorización.](#)

Configure la autorización mediante la base de datos integrada con Windows

En esta sección se describe la configuración de las reglas de autorización mediante la base de datos integrada para las implementaciones de Windows.

Para agregar reglas de autorización básicas

1. Compruebe que el bróker EMQX esté desplegado y en funcionamiento.
2. Ejecute la herramienta AWS IoT SiteWise CLI de EMQX:

```
C:\greengrass\v2\bin\swe-emqx-cli.ps1 acl init
```

La herramienta crea y aplica automáticamente las reglas de ACL que permiten las conexiones desde el servidor local (127.0.0.1) al intermediario. Permite el acceso a todos los temas. Esto incluye el recopilador IoT SiteWise OPC UA y el SiteWise editor de IoT.

3. Continúe en [Actualice la configuración de despliegue del EMQX para su autorización](#).

Actualice la configuración de despliegue del EMQX para su autorización

Para actualizar la configuración de despliegue de EMQX para su autorización

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación de la izquierda, elija las puertas de enlace Edge en la sección Edge.
3. Elija la puerta de enlace que desee configurar.
4. En la sección de configuración de la puerta de enlace Edge, copia el valor de tu dispositivo principal de Greengrass. Guárdelo para usarlo más adelante.
5. Abra la [consola de AWS IoT](#).
6. En el menú de navegación de la izquierda, en la sección Administrar, selecciona dispositivos Greengrass y, a continuación, Implementaciones.
7. Busque el valor del dispositivo principal que guardó anteriormente y elija ese enlace para abrir la implementación.
8. Selecciona el botón desplegable Acciones y, a continuación, Revisa.
9. Lea el mensaje que aparece y, a continuación, seleccione Revisar el despliegue. Aparece la página Especificar el destino.
10. Elija Siguiente hasta llegar al paso Configurar componentes.

11. Seleccione el botón de opción `aws.greengrass.clientdevices.mqtt.EMQX`.
12. Pulse el botón Configurar componente. Aparece una página de configuración para el componente.
13. En Actualización de la configuración, seleccione Restablecer la configuración predeterminada para la versión del componente: `2.*.*`.
14. Pegue el siguiente contenido en la sección Configuración para fusionar en función de su sistema operativo.

Linux

```
{
  "emqxConfig": {
    "authorization": {
      "no_match": "deny",
      "sources": [
        {
          "type": "built_in_database"
        },
        {
          "type": "file",
          "path": "data/authz/acl.conf"
        }
      ]
    },
    "listeners": {
      "tcp": {
        "default": {
          "enabled": true,
          "enable_authn": false
        }
      },
      "ssl": {
        "default": {
          "enabled": true,
          "enable_authn": true,
          "ssl_options": {
            "verify": "verify_none",
            "fail_if_no_peer_cert": false
          }
        }
      }
    }
  },
}
```

```

    "authentication": {
      "enable": true,
      "backend": "built_in_database",
      "mechanism": "password_based",
      "password_hash_algorithm": {
        "iterations": 210000,
        "mac_fun": "sha512",
        "name": "pbkdf2"
      },
      "user_id_type": "username"
    },
    "dashboard": {
      "listeners": {
        "http": {
          "bind": 18083
        }
      }
    }
  },
  "authMode": "bypass",
  "dockerOptions": "-p 8883:8883 -p 127.0.0.1:1883:1883
-p 127.0.0.1:18083:18083 -v emqx-data:/opt/emqx/data -e
EMQX_NODE__NAME=emqx@local",
  "requiresPrivilege": "true"
}

```

Windows

```

{
  "emqxConfig": {
    "authorization": {
      "no_match": "deny",
      "sources": [
        {
          "type": "built_in_database"
        },
        {
          "type": "file",
          "path": "C:\\greengrass\\v2\\work\\
\\aws.greengrass.clientdevices.mqtt.EMQX\\v2\\data\\authz\\acl.conf"
        }
      ]
    }
  },
}

```

```
"listeners": {
  "tcp": {
    "default": {
      "enabled": true,
      "enable_authn": false
    }
  },
  "ssl": {
    "default": {
      "enabled": true,
      "enable_authn": true,
      "ssl_options": {
        "verify": "verify_none",
        "fail_if_no_peer_cert": false
      }
    }
  }
},
"authentication": {
  "enable": true,
  "backend": "built_in_database",
  "mechanism": "password_based",
  "password_hash_algorithm": {
    "iterations": 210000,
    "mac_fun": "sha512",
    "name": "pbkdf2"
  },
  "user_id_type": "username"
},
"dashboard": {
  "listeners": {
    "http": {
      "bind": 18083
    }
  }
},
"authMode": "bypass",
"requiresPrivilege": "true"
}
```

15. Elija Confirmar.

16. Elija Siguiente hasta llegar al paso Revisar.

17. Elija Implementar.

Note

A partir de este momento, no podrá editar el archivo ACL para actualizar las reglas de autorización. Como alternativa, puede continuar [Agregue reglas de autorización a través del panel de control de EMQX para los usuarios](#) después de una implementación exitosa.

Agregue reglas de autorización a través del panel de control de EMQX para los usuarios

Puede añadir o actualizar las reglas de autorización mediante el panel de control de EMQX o la herramienta CLI de AWS IoT SiteWise EMQX. La herramienta AWS IoT SiteWise CLI de EMQX gestiona la autorización mediante la base de datos integrada de EMQX.

Note

Añadir reglas de autorización es un paso de configuración avanzada que requiere comprender los patrones de temas de MQTT y el control de acceso. Para obtener más información sobre cómo crear reglas de autorización utilizando la base de datos integrada de EMQX, consulte [Uso de una base de datos integrada en los documentos de EMQX](#).

Note

Las instrucciones relacionadas con el EMQX que se proporcionan son solo de referencia. Como la documentación y las funciones de EMQX pueden cambiar con el tiempo y no mantenemos su documentación, recomendamos consultar la documentación [oficial de EMQX para obtener la](#) información más actualizada.

EMQX dashboard

Este procedimiento muestra cómo añadir reglas de autorización en el panel de EMQX.

Solo se puede acceder al panel de EMQX desde el host de la puerta de enlace. Si intenta conectarse desde fuera del host de la puerta de enlace, no podrá acceder al panel de control.

Para añadir reglas de autorización mediante el panel de EMQX

1. Asegúrese de estar dentro del host de la puerta de enlace.
2. Abra una ventana del navegador y visite <http://localhost:18083/>.
3. Inicie sesión en el panel de EMQX. En este procedimiento se presupone que ha cambiado sus credenciales de inicio de sesión predeterminadas por las que ha elegido. Para obtener más información sobre la configuración inicial, consulte [Habilite la autenticación de nombre de usuario y contraseña](#).
4. Seleccione el icono del escudo y, a continuación, Autorización en el menú desplegable.
5. Pulse el botón Permisos en la fila de la base de datos integrada.
6. En la sección de autorización de la base de datos integrada, añada o actualice las reglas de autorización de usuario según las necesidades de su empresa. Para obtener más información sobre la creación de reglas, consulte la sección [Uso de una base de datos integrada](#) en los documentos de EMQX.

AWS IoT SiteWise CLI tool using Linux

Para administrar las reglas de autorización mediante la herramienta AWS IoT SiteWise CLI de EMQX en Linux:

- Agregue reglas de autorización para un usuario mediante el siguiente formato:

```
/greengrass/v2/bin/swe-emqx-cli auth add your-username your-action your-permission your-topic [your-action-permission-topic]
```

Example Agregue reglas de autorización para un usuario

En este ejemplo se muestra cómo añadir reglas para un usuario llamado `system1`:

```
/greengrass/v2/bin/swe-emqx-cli auth add system1 \  
  publish allow "sensors/#" \  
  subscribe allow "control/#" \  
  all deny "#"
```

Example: Ver las reglas de autorización de un usuario

Para ver las reglas de autorización de los `system1` usuarios, ejecute el siguiente comando:

```
/greengrass/v2/bin/swe-emqx-cli auth list system1
```

Example: Vea todas las reglas de autorización existentes

Para ver todas las reglas de autorización que tiene actualmente, ejecute el siguiente comando:

```
/greengrass/v2/bin/swe-emqx-cli auth list
```

Example: Elimine todas las reglas de autorización de un usuario

Para eliminar todas las reglas de autorización aplicadas a un usuario concreto, ejecute el siguiente comando:

```
/greengrass/v2/bin/swe-emqx-cli auth delete system1
```

Se le pedirá que confirme la eliminación.

AWS IoT SiteWise CLI tool using Windows

Para administrar las reglas de autorización mediante la herramienta AWS IoT SiteWise CLI de EMQX en: Windows PowerShell

- Agregue reglas de autorización para un usuario mediante el siguiente formato:

```
C:\greengrass\v2\bin\swe-emqx-cli.ps1 auth add your-username your-action your-permission your-topic [your-action-permission-topic]
```

Example: Agregue reglas de autorización para un usuario

En este ejemplo se muestra cómo añadir reglas para un usuario llamado `system1`:

```
C:\greengrass\v2\bin\swe-emqx-cli.ps1 auth add system1 `
  publish allow "sensors/#" `
  subscribe allow "control/#" `
  all deny "#"
```

Example: Ver las reglas de autorización de un usuario

Para ver las reglas de autorización de los `system1` usuarios, ejecute el siguiente comando:

```
C:\greengrass\v2\bin\swe-emqx-cli.ps1 auth list system1
```

Example: Vea todas las reglas de autorización existentes

Para ver todas las reglas de autorización que tiene actualmente, ejecute el siguiente comando:

```
C:\greengrass\v2\bin\swe-emqx-cli.ps1 auth list
```

Example: Elimine todas las reglas de autorización de un usuario

Para eliminar todas las reglas de autorización aplicadas a un usuario concreto, ejecute el siguiente comando:

```
C:\greengrass\v2\bin\swe-emqx-cli.ps1 auth delete system1
```

Se le pedirá que confirme la eliminación.

Procese y visualice datos con SiteWise Edge y herramientas de código abierto

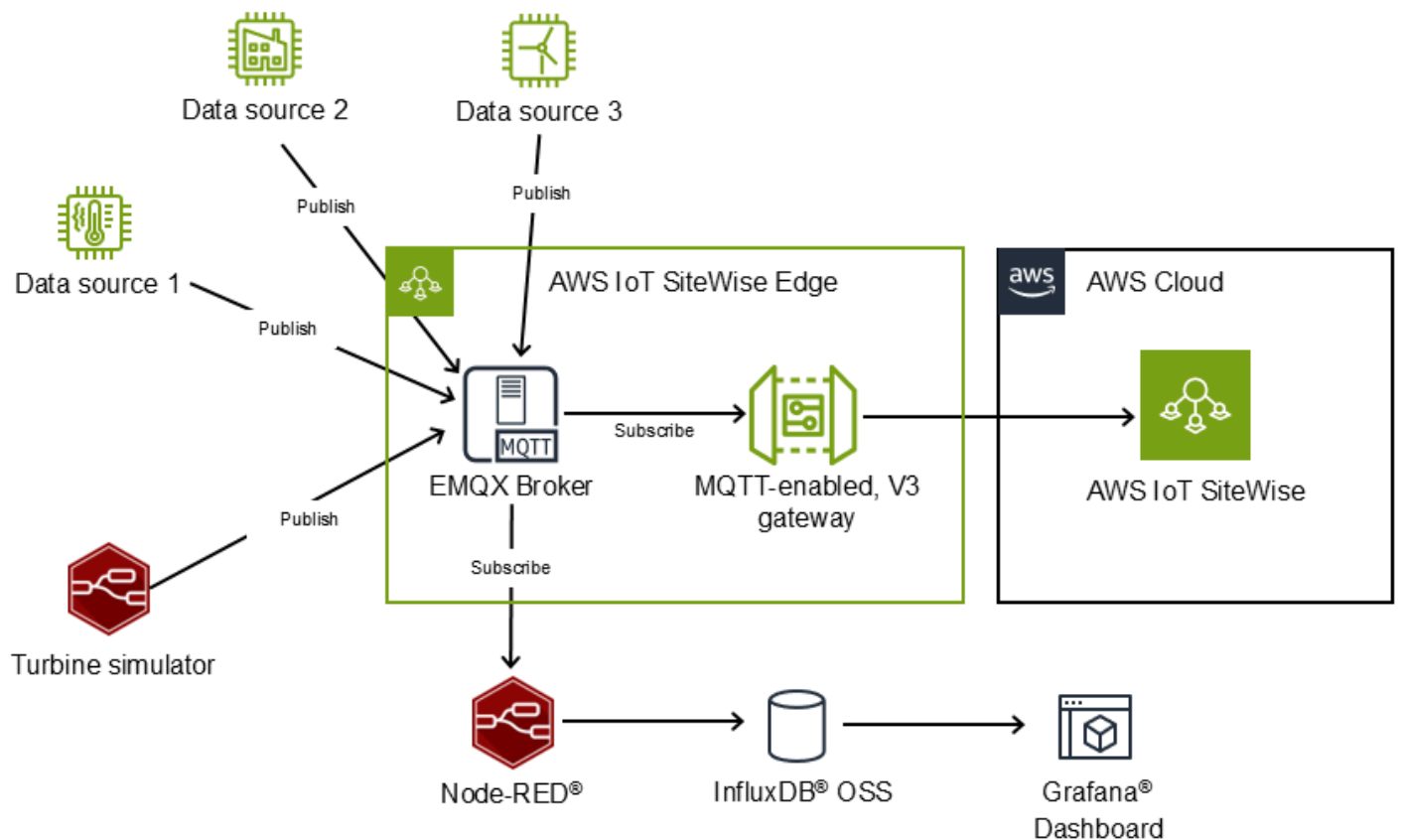
Configure las puertas de enlace compatibles con AWS IoT SiteWise Edge MQTT con herramientas de código abierto para el procesamiento y la visualización locales a fin de mejorar sus capacidades de gestión de datos industriales.

Con SiteWise Edge, puede crear una canalización de procesamiento de datos local mediante herramientas externas de código abierto. [Utilice Node-RED® para almacenar datos de series temporales con InfluxDB® y supervise las operaciones a través de los paneles de Grafana®.](#)

Node-RED procesa y transforma sus flujos de datos, mientras que InfluxDB proporciona almacenamiento de datos de series temporales. Grafana muestra sus datos operativos en tiempo real. Utilice estas herramientas con SiteWise Edge para sincronizar los datos entre su entorno local y el entorno local Nube de AWS, lo que le proporcionará información local inmediata y funciones de análisis a largo plazo basadas en la nube.

Note

Node-RED®, InfluxDB® y Grafana® no son vendedores ni proveedores de Edge. SiteWise



Note

En esta guía, utilizamos la versión de código abierto de [Grafana SiteWise for Edge](#) en lugar del servicio [Amazon Managed Grafana](#).

Opciones de implementación

Puede implementar esta solución mediante uno de estos dos enfoques. Con una configuración Microsoft Windows manual, usted controla la configuración y la integración de los componentes con su infraestructura. Con Linux, puede usar Docker para implementar componentes preconfigurados en contenedores.

Elija el método que se adapte a sus requisitos operativos.

- [Configure las integraciones de código abierto manualmente \(Windows\)](#)— Para configuraciones personalizadas o infraestructuras existentes
- [Configure integraciones de código abierto con Docker \(Linux\)](#)— Para un despliegue rápido con componentes preconfigurados

Descripción general de un ejemplo de parque eólico

Esta guía utiliza un ejemplo de parque eólico para demostrar cómo se puede monitorizar la velocidad del viento de una turbina en un parque eólico. Este escenario práctico ilustra las necesidades comunes de monitoreo industrial, en las que tanto la visibilidad local como la basada en la nube son valiosas para la eficiencia operativa.

Con esta integración, puede:

- Recopile datos de equipos industriales mediante una puerta de enlace AWS IoT SiteWise Edge
- Procese datos localmente con Node-RED, InfluxDB y Grafana
- Almacene datos localmente usando InfluxDB
- Supervise los datos en tiempo real mediante los paneles de control de Grafana

A lo largo de esta guía, utilizamos el ejemplo de un parque eólico. Usamos Node-RED para simular una turbina que genera datos de velocidad del viento. Node-RED traduce la carga útil de datos, los publica en el broker MQTT de SiteWise Edge, se suscribe para recibir datos del bróker y los almacena localmente en InfluxDB. Este enfoque garantiza que todos los datos operativos estén disponibles tanto localmente para su acceso inmediato como en la nube para su posterior análisis. Al implementar este patrón, gana resiliencia ante las interrupciones de la red y, al mismo tiempo, mantiene la capacidad de realizar análisis avanzados en la Nube de AWS red. Grafana se conecta a InfluxDB para el monitoreo local, lo que proporciona a los operadores visibilidad en tiempo real de las métricas sin depender de la nube. Una pasarela habilitada para MQTT de SiteWise Edge se conecta al mismo intermediario de MQTT al que enviar los datos AWS IoT SiteWise, lo que crea un puente entre sus operaciones perimetrales y los servicios basados en la nube.

Puede utilizar sus propios datos y configuraciones para crear un flujo de trabajo similar adaptado a sus requisitos industriales específicos, ya sea que esté supervisando equipos de fabricación, infraestructuras de servicios públicos u otros activos industriales.

Requisitos para las integraciones de código abierto

Antes de implementar integraciones de código abierto con SiteWise Edge, asegúrese de que su entorno cumpla con los requisitos necesarios.

- Requisitos de hardware: el hardware de su puerta de enlace debe cumplir los requisitos de las puertas de enlace SiteWise Edge. Para obtener más información, consulte las puertas de enlace V3 habilitadas [AWS IoT SiteWise Requisitos de la puerta de enlace autohospedada de Edge](#) para MQTT y [Requisitos para la aplicación AWS IoT SiteWise Edge](#)

⚠ Important

[Al implementar componentes de código abierto adicionales, asegúrese de que su hardware cumpla con los requisitos de InfluxDB, Node-RED y Grafana.](#)

- La configuración de su red debe admitir tanto la comunicación local entre los componentes como la conectividad a la nube para Edge. SiteWise
- Todos los servicios deben ejecutarse en el mismo host.

Consideraciones de seguridad

Se recomienda cifrar todas las comunicaciones entre los componentes, especialmente al acceder a las interfaces desde redes no locales. Implemente los controles de acceso adecuados para cada componente y siga las prácticas AWS recomendadas para la configuración de la puerta de enlace AWS IoT SiteWise Edge y la seguridad de las AWS cuentas.

Entorno de desarrollo

Esta guía muestra cómo Node-RED, InfluxDB y Grafana se ejecutan y se accede localmente en un host de puerta de enlace. Para las implementaciones de producción que requieren acceso externo, implemente medidas de seguridad que incluyan el cifrado, la autenticación y la autorización TLS. Siga las prácticas recomendadas de seguridad de cada aplicación.

Software de terceros

Esta solución utiliza software de terceros no mantenido por AWS, incluidos InfluxDB, Node-RED, Grafana y el complemento. `node-red-contrib-influxdb` Antes de la implementación, asegúrese de que estos componentes cumplan con los requisitos de seguridad, las normas de cumplimiento y las políticas de gobierno de su organización.

⚠ Important

Esta guía hace referencia y utiliza software de terceros que no es propiedad ni mantenido por él AWS. Antes de la implementación, asegúrese de que todos los componentes cumplan con sus requisitos de seguridad, cumplimiento y gobierno. Mantenga todo el software actualizado con los parches de seguridad más recientes y siga las mejores prácticas para proteger su implementación perimetral.

InfluxDB, Node-RED y Grafana no son vendedores ni proveedores de Edge. SiteWise

Otras consideraciones

Tenga en cuenta estos factores adicionales al implementar integraciones de código abierto con Edge. SiteWise

- Usa las versiones más recientes de todos los servicios, herramientas y componentes.
- Filtre y agregue los datos localmente antes de transmitirlos a la nube para reducir AWS IoT SiteWise los costos de ingesta de datos. Configure los períodos de retención de datos adecuados en InfluxDB y dimensione correctamente el hardware de su puerta de enlace. Para obtener más información, consulte [Precios de AWS IoT SiteWise](#).
- Implemente procedimientos de respaldo periódicos para todos los datos.
- Supervise el uso de los recursos en su puerta de enlace y configure los límites de recursos adecuados para cada componente. Implemente políticas de retención de datos en InfluxDB para administrar el uso del disco.

Configure las integraciones de código abierto manualmente (Windows)

Utilice esta guía para crear manualmente un depósito de series temporales para los datos de velocidad del viento que se conecte con Grafana® y Node-RED®.

Instale y configure manualmente Node-RED, InfluxDB® y Grafana para controlar la configuración de la Microsoft Windows implementación. Puede almacenar y administrar los datos de series temporales de sus dispositivos mediante InfluxDB.

Requisitos previos de configuración manual

Antes de empezar, complete estos requisitos:

Note

Ejecute todos los servicios (SiteWise Edge, InfluxDB, Node-RED y Grafana) en el mismo host.

- Instale una puerta de enlace V3 habilitada para MQTT. Para obtener más información, consulte [Puertas de enlace V3 habilitadas para MQTT para Edge AWS IoT SiteWise](#).

- Instale y ejecute estos servicios localmente:
 - InfluxDB OSS v2. Para ver los pasos de instalación, consulte [Instalar InfluxDB](#).
 - Node-RED. Para ver los pasos de instalación, consulte [Instalar Node-RED](#) localmente.
 - Grafana. Para ver los pasos de instalación, consulte [Instalación de Grafana](#).

Configure el almacenamiento local con InfluxDB

Con InfluxDB®, puede almacenar datos de series temporales de sus dispositivos de forma local. El objetivo de la capacidad de almacenamiento local es mantener la visibilidad operativa durante las interrupciones de la red y reducir la latencia de las aplicaciones en las que el tiempo es crucial. Puede realizar análisis y visualizaciones en la periferia sin dejar de tener la opción de reenviar los datos a la nube de forma selectiva.

En esta sección, creará un depósito de series temporales para los datos de velocidad del viento de las turbinas y generará un token de API para la conectividad de Grafana® y Node-RED®. El depósito de InfluxDB sirve como un contenedor de almacenamiento dedicado para los datos de series temporales, similar a una base de datos en los sistemas tradicionales. El token de API permite un acceso programático seguro a sus datos.

Para configurar InfluxDB

1. [Tras completar los pasos previos y asegurarse de que todas las herramientas se ejecutan en el mismo servidor, abra su navegador web y vaya a `http://127.0.0.1:8086`.](#)
2. (Opcional) Habilite el cifrado TLS para mejorar la seguridad. Para obtener más información, consulte [Habilitar el cifrado TLS](#) en InfluxData Documentation
3. Cree un depósito de InfluxDB de series temporales para almacenar datos de Node-RED. El depósito servirá como un contenedor específico para los datos de su parque eólico, lo que le permitirá organizar y administrar las políticas de retención específicas de este conjunto de datos. Para obtener más información, consulte [Administrar depósitos](#) en InfluxData Documentation
4. (Opcional) Configure el período de retención de datos para su ubicación perimetral. Establecer períodos de retención adecuados ayuda a administrar los recursos de almacenamiento de manera eficiente al eliminar automáticamente los datos antiguos que ya no son necesarios para las operaciones locales.

Para obtener información sobre la retención de datos, consulte [Retención de datos en InfluxDB](#) en InfluxData Documentation

5. Genere un token de API para el depósito. Este token permitirá una comunicación segura entre InfluxDB y otros componentes como Node-RED y Grafana. De esta forma, solo los servicios autorizados pueden leer o escribir en su almacén de datos. Para obtener más información, consulte [Crear un token](#) en InfluxData Documentation.

Tras completar estos pasos, puede almacenar datos de series temporales en su instancia de InfluxDB, lo que proporciona una base para la persistencia y el análisis de los datos locales en su entorno perimetral.

Configure los flujos de Node-RED para la integración de datos AWS IoT SiteWise

Con Node-RED®, puede implementar dos flujos para administrar los datos entre sus dispositivos y AWS IoT SiteWise. Estos flujos funcionan en conjunto para crear una solución de administración de datos integral que aborde el flujo de datos local y en la nube.

- Flujo de publicación de datos: se publica en la nube. El flujo de publicación de datos envía datos a AWS IoT SiteWise. Este flujo simula un dispositivo de turbina generando datos de sensores, traduciéndolos al AWS IoT SiteWise formato y publicándolos en el broker MQTT de SiteWise Edge. Esto le permite aprovechar las capacidades AWS IoT SiteWise de la nube para el almacenamiento, el análisis y la integración con otros servicios. AWS

Para obtener más información, consulte [Configure el flujo de publicación de datos](#).

- Flujo de retención de datos: almacena los datos en la periferia. El flujo de retención de datos se suscribe al broker MQTT de SiteWise Edge para recibir datos, traducirlos al formato InfluxDB® y almacenarlos localmente para su supervisión. Este almacenamiento local proporciona acceso inmediato a los datos operativos, reduce la latencia de las aplicaciones en las que el tiempo es crucial y garantiza la continuidad durante las interrupciones de la red.

Para obtener más información, consulte [Configure el flujo de retención de datos](#).

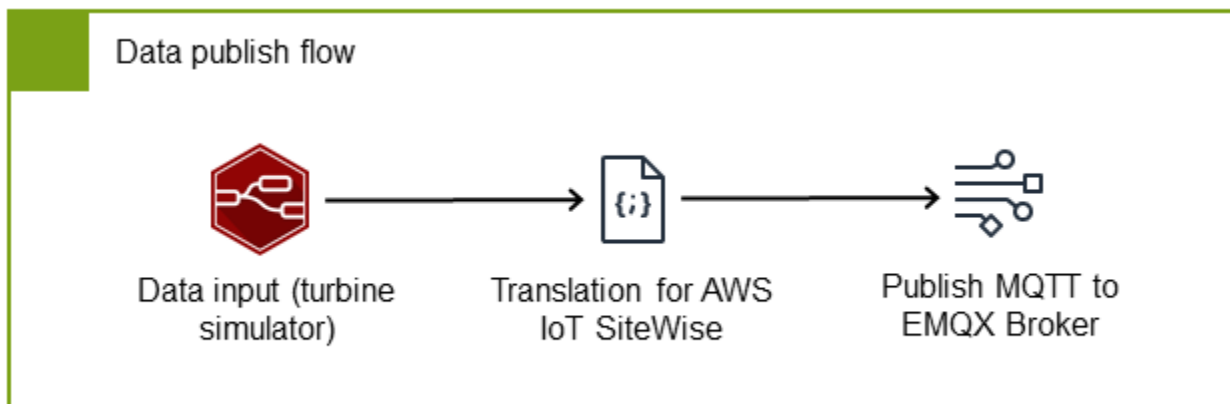
Estos dos flujos funcionan juntos para garantizar que los datos se envíen a AWS IoT SiteWise y almacenen localmente para su acceso inmediato.

[Para acceder a la consola Node-RED, vaya a http://127.0.0.1:1880](#). Para obtener información sobre cómo habilitar el cifrado TLS, consulte [Habilitar](#) el cifrado TLS.

Configure el flujo de publicación de datos

El flujo de publicación de datos utiliza tres nodos para crear una canalización que envía los datos industriales a la nube. Este flujo es esencial para permitir el análisis basado en la nube, el almacenamiento a largo plazo y la integración con otros AWS servicios. En primer lugar, los datos simulados del dispositivo se envían al intermediario MQTT de SiteWise Edge. La puerta de enlace recoge los datos del intermediario, lo que permite su transmisión a la AWS IoT SiteWise nube, donde puede aprovechar las potentes funciones de análisis y visualización.

- Entrada de datos: recibe datos de dispositivos de sus equipos industriales o simuladores
- Traductor de datos para AWS IoT SiteWise: traduce los datos a un AWS IoT SiteWise formato para garantizar la compatibilidad con la puerta de enlace SiteWise Edge
- Publicador de MQTT: publica los datos en el broker MQTT de SiteWise Edge, lo que los pone a disposición de los consumidores locales y de la nube



Configure el nodo de entrada de datos

En este ejemplo, el nodo de entrada de datos usa un dispositivo de turbina eólica simulado que genera datos de velocidad del viento. Este nodo sirve como punto de entrada para sus datos industriales, ya provengan de fuentes simuladas (como en nuestro ejemplo) o de equipos industriales reales en entornos de producción.

Usamos un formato JSON personalizado para la carga útil de datos a fin de proporcionar una estructura estandarizada que funcione de manera eficiente tanto con las herramientas de procesamiento locales como con el servicio AWS IoT SiteWise en la nube. Este formato incluye metadatos esenciales, como marcas de tiempo e indicadores de calidad, junto con los valores de

medición reales, lo que permite una gestión integral de los datos y un seguimiento de la calidad durante todo el proceso. Importe el nodo de inyección para recibir datos simulados en este formato JSON estandarizado con marcas de tiempo, indicadores de calidad y valores.

[Para obtener más información sobre el nodo de inyección Node-RED, consulte la sección Inject de la documentación. Node-RED](#)

El simulador de turbina genera datos de velocidad del viento cada segundo en este formato JSON estandarizado:

Example: Carga útil de datos de la turbina

```
{
  name: string,          // Property name/identifier
  timestamp: number,    // Epoch time in nanoseconds
  quality: "GOOD" | "UNCERTAIN" | "BAD",
  value: number | string | boolean
}
```

Este formato ofrece varias ventajas:

- El `name` campo identifica la propiedad o medida específica, lo que le permite rastrear varios puntos de datos desde el mismo dispositivo
- El valor `timestamp` en nanosegundos garantiza un seguimiento preciso del tiempo para un análisis histórico preciso
- El `quality` indicador le ayuda a filtrar y gestionar los datos en función de su fiabilidad
- El `value` campo flexible admite diferentes tipos de datos para adaptarse a las distintas salidas de los sensores

Example: nodo de inyección de un simulador de turbina

```
[
  {
    "id": "string",
    "type": "inject",
    "z": "string",
    "name": "Turbine Simulator",
    "props": [
      {
```

```

        "p": "payload.timestamp",
        "v": "",
        "vt": "date"
    },
    {
        "p": "payload.quality",
        "v": "GOOD",
        "vt": "str"
    },
    {
        "p": "payload.value",
        "v": "$random()",
        "vt": "jsonata"
    },
    {
        "p": "payload.name",
        "v": "/Renton/WindFarm/Turbine/WindSpeed",
        "vt": "str"
    }
],
"repeat": "1",
"crontab": "",
"once": false,
"onceDelay": "",
"topic": "",
"x": 270,
"y": 200,
"wires": [
    [
        "string"
    ]
]
}
]

```

Configure un nodo para la traducción de datos

La puerta de enlace SiteWise Edge requiere datos en un formato específico para garantizar la compatibilidad con AWS IoT SiteWise la nube. El nodo traductor es un componente importante que convierte los datos de entrada al formato de AWS IoT SiteWise carga útil requerido. Este paso de traducción garantiza que sus datos industriales se puedan procesar, almacenar y analizar posteriormente de forma adecuada en el entorno de AWS IoT SiteWise nube.

Al estandarizar el formato de los datos en esta etapa, posibilita la integración entre sus dispositivos periféricos y el servicio en la nube, donde puede utilizar las capacidades de modelado, análisis y visualización de activos. Utilice esta estructura:

Example: Estructura de carga útil para el análisis de datos de SiteWise Edge

```
{
  "propertyAlias": "string",
  "propertyValues": [
    {
      "value": {
        "booleanValue": boolean,
        "doubleValue": number,
        "integerValue": number,
        "stringValue": "string"
      },
      "timestamp": {
        "timeInSeconds": number,
        "offsetInNanos": number
      },
      "quality": "GOOD" | "UNCERTAIN" | "BAD",
    }
  ]
}
```

Note

Haga coincidir el `propertyAlias` tema con su jerarquía de temas de MQTT (por ejemplo, `/Renton/WindFarm/Turbine/WindSpeed`). Esto garantiza que sus datos estén asociados correctamente a la propiedad de activo correcta en AWS IoT SiteWise. Para obtener más información, consulte el concepto de «alias de flujo de datos» en [AWS IoT SiteWise conceptos](#).

1. Importe el nodo de función de ejemplo para la traducción de la AWS IoT SiteWise carga útil. Esta función gestiona la conversión del formato de entrada estandarizado al formato AWS IoT SiteWise compatible, lo que garantiza que el formato de la marca de tiempo, los indicadores de calidad y la escritura de valores sean correctos.

```
[
  {
```

```

    "id": "string",
    "type": "function",
    "z": "string",
    "name": "Translate to SiteWise payload",
    "func": "let input = msg.payload;\nlet output = {};\n
\noutput[\"propertyAlias\"] = input.name;\n\nlet propertyVal = {}\n\nlet
  timeInSeconds = Math.floor(input.timestamp / 1000);\nlet offsetInNanos =
  (input.timestamp % 1000) * 1000000;\n\npropertyVal[\"timestamp\"] = {\n
  \"timeInSeconds\": timeInSeconds,\n  \"offsetInNanos\": offsetInNanos,\n};
\n\npropertyVal[\"quality\"] = input.quality\n\nlet typeNameConverter = {\n
  \"number\": (x) => Number.isInteger(x) ? \"integerValue\" : \"doubleValue
\", \n  \"boolean\": (x) => \"booleanValue\", \n  \"string\": (x) =>
  \"stringValue\", \n}\n\nlet typeName = typeNameConverter[typeof input.value]
(input.value)\n\npropertyVal[\"value\"] = {}\n\npropertyVal[\"value\"][typeName]
= input.value;\n\noutput[\"propertyValues\"] = [propertyVal]\n\nreturn {\n
  payload: JSON.stringify(output)\n};",
    "outputs": 1,
    "timeout": "",
    "noerr": 0,
    "initialize": "",
    "finalize": "",
    "libs": [],
    "x": 530,
    "y": 200,
    "wires": [
      [
        "string"
      ]
    ]
  }
]

```

2. Compruebe que el JavaScript código traduce los datos de velocidad del viento correctamente. La función realiza varias tareas importantes:
 - Extrae el nombre de la propiedad de la entrada y lo establece como PropertyAlias
 - Convierte la marca de tiempo de milisegundos al formato de segundos y nanosegundos requerido
 - Conserva el indicador de calidad para el seguimiento de la fiabilidad de los datos
 - Detecta automáticamente el tipo de valor y lo formatea según AWS IoT SiteWise los requisitos
3. Conecte el nodo a su flujo, vinculándolo entre el nodo de entrada de datos y el editor MQTT.

Para obtener orientación sobre cómo escribir una función específica para las necesidades de su empresa, consulte Cómo [escribir funciones](#) en la documentación Node-RED

Configure el editor MQTT

Tras la traducción, los datos están listos para su publicación en el broker MQTT de SiteWise Edge.

Configure el editor MQTT con estos ajustes para enviar datos al agente MQTT de SiteWise Edge:

Para importar el nodo de salida MQTT

1. Importe un nodo de configuración de salida MQTT mediante. "type": "mqtt out" Los nodos de salida MQTT le permiten compartir la configuración de un corredor.
2. Introduzca los pares clave-valor para obtener información relevante sobre la conexión del bróker MQTT y el enrutamiento de mensajes.

Importe el nodo de ejemplo. `mqtt out`

Example

```
[
  {
    "id": "string",
    "type": "mqtt out",
    "z": "string",
    "name": "Publish to MQTT broker",
    "topic": "/Renton/WindFarm/Turbine/WindSpeed",
    "qos": "1",
    "retain": "",
    "respTopic": "",
    "contentType": "",
    "userProps": "",
    "correl": "",
    "expiry": "",
    "broker": "string",
    "x": 830,
    "y": 200,
    "wires": []
  },
  {
    "id": "string",
    "type": "mqtt-broker",
```

```
    "name": "emqx",
    "broker": "127.0.0.1",
    "port": "1883",
    "clientid": "",
    "autoConnect": true,
    "usetls": false,
    "protocolVersion": "5",
    "keepalive": 15,
    "cleansession": true,
    "autoUnsubscribe": true,
    "birthTopic": "",
    "birthQos": "0",
    "birthPayload": "",
    "birthMsg": {},
    "closeTopic": "",
    "closePayload": "",
    "closeMsg": {},
    "willTopic": "",
    "willQos": "0",
    "willPayload": "",
    "willMsg": {},
    "userProps": "",
    "sessionExpiry": ""
  }
]
```

El nodo de salida MQTT de ejemplo crea la conexión MQTT con la siguiente información:

- Servidor: 127.0.0.1
- Puerto: 1883
- Protocolo: MQTT V5

A continuación, el nodo de salida MQTT configura el enrutamiento de mensajes con la siguiente información:

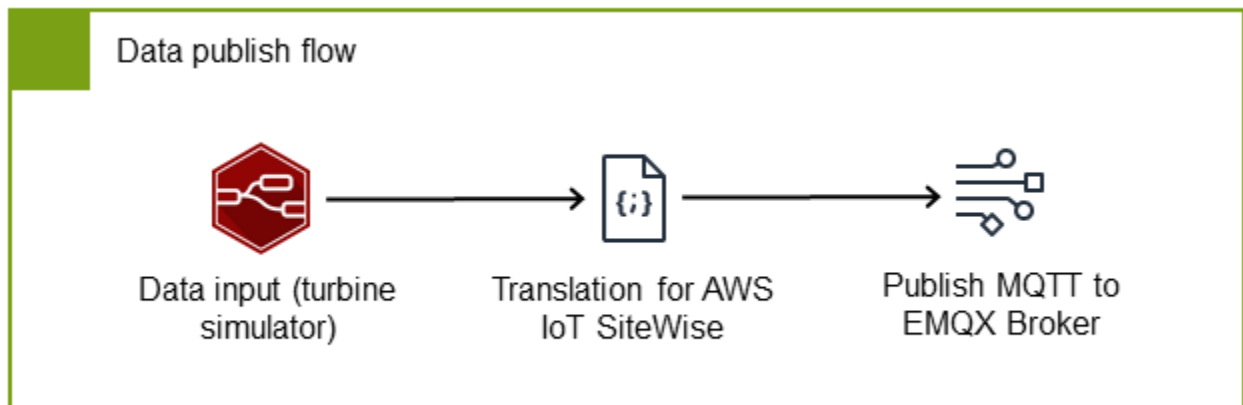
- Tema: /Renton/WindFarm/Turbine/WindSpeed
- QoS: 1

Implemente y verifique los nodos

Tras configurar los tres nodos del flujo de publicación de datos, siga estos pasos para implementar el flujo y comprobar que los datos se transmiten correctamente a AWS IoT SiteWise

Para implementar y verificar las conexiones

1. Conecte los tres nodos como se muestra en el flujo de publicación de datos.



2. Elija Implementar para aplicar todos los cambios en la conexión de los nodos.
3. Navegue hasta la [AWS IoT SiteWise consola](#) y elija Flujos de datos.
4. Asegúrese de seleccionar el prefijo Alias en el menú desplegable. A continuación, busca el / Renton/WindFarm/Turbine/WindSpeed alias.

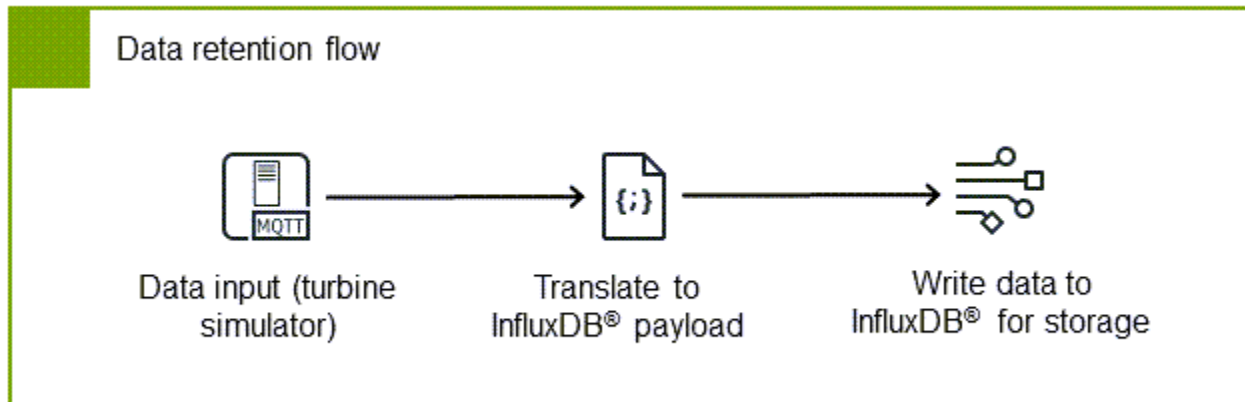
Si ve el alias correcto en la búsqueda, significa que ha desplegado el flujo y verificado la transmisión de datos.

Configure el flujo de retención de datos

El flujo de retención de datos se puede utilizar para mantener la visibilidad operativa en la periferia. Esto resulta útil durante las interrupciones de la red o cuando se necesita acceso inmediato a los datos. Este flujo se suscribe al intermediario MQTT para recibir los datos del dispositivo, los convierte al formato InfluxDB® y los almacena localmente. Al implementar este flujo, se crea un almacén de datos local resiliente al que los operadores pueden acceder sin depender de la nube, lo que permite la supervisión en tiempo real y la toma de decisiones en la periferia.

El flujo consta de tres componentes clave que trabajan juntos para garantizar que sus datos se capturen y almacenen correctamente:

- Cliente de suscripción a MQTT: recibe los datos del intermediario, lo que garantiza la captura de todos los datos industriales relevantes
- Traductor InfluxDB: convierte la AWS IoT SiteWise carga útil al formato InfluxDB y prepara los datos para un almacenamiento eficiente de series temporales
- Escritor InfluxDB: gestiona el almacenamiento local, lo que garantiza la persistencia de los datos y la disponibilidad de las aplicaciones locales



Configure el cliente de suscripción MQTT

- Configure el cliente de suscripción MQTT en Node-RED para recibir datos del broker EMQX de MQTT importando el siguiente ejemplo. AWS IoT SiteWise

Example: MQTT en el nodo

```
[
  {
    "id": "string",
    "type": "mqtt in",
    "z": "string",
    "name": "Subscribe to MQTT broker",
    "topic": "/Renton/WindFarm/Turbine/WindSpeed",
    "qos": "1",
    "datatype": "auto-detect",
    "broker": "string",
    "nl": false,
    "rap": true,
    "rh": 0,
    "inputs": 0,
  }
]
```

```
    "x": 290,
    "y": 340,
    "wires": [
      [
        "string"
      ]
    ]
  },
  {
    "id": "string",
    "type": "mqtt-broker",
    "name": "emqx",
    "broker": "127.0.0.1",
    "port": "1883",
    "clientId": "",
    "autoConnect": true,
    "usetls": false,
    "protocolVersion": "5",
    "keepalive": 15,
    "cleansession": true,
    "autoUnsubscribe": true,
    "birthTopic": "",
    "birthQos": "0",
    "birthPayload": "",
    "birthMsg": {},
    "closeTopic": "",
    "closePayload": "",
    "closeMsg": {},
    "willTopic": "",
    "willQos": "0",
    "willPayload": "",
    "willMsg": {},
    "userProps": "",
    "sessionExpiry": ""
  }
]
```

Esta suscripción garantiza que todos los datos relevantes publicados en el corredor se recopilen para su almacenamiento local, lo que proporciona un registro completo de sus operaciones industriales. El nodo utiliza los mismos parámetros de conexión MQTT que la [Configurar el editor MQTT](#) sección, con los siguientes ajustes de suscripción:

- Tema — /Renton/WindFarm/Turbine/WindSpeed
- QoS — 1

Para obtener más información, consulte [Conectarse a un broker de MQTT](#) en la Node-RED documentación.

Configure el traductor InfluxDB

[InfluxDB organiza los datos mediante etiquetas para la indexación y campos para los valores.](#) Esta organización optimiza el rendimiento de las consultas y la eficiencia del almacenamiento de los datos de series temporales. Importe el nodo de función de ejemplo que contiene el JavaScript código para convertir la AWS IoT SiteWise carga útil al formato InfluxDB. El traductor divide las propiedades en dos grupos:

- Etiquetas: propiedades de calidad y nombre para una indexación eficiente
- Campos: marca de tiempo (en milisegundos desde la época) y valor

Example: Nodo de función de traducción a una carga útil de InfluxDB

```
[
  {
    "id": "string",
    "type": "function",
    "z": "string",
    "name": "Translate to InfluxDB payload",
    "func": "let data = msg.payload;\n\nlet timeInSeconds =
data.propertyValues[0].timestamp.timeInSeconds;\nlet offsetInNanos =
data.propertyValues[0].timestamp.offsetInNanos;\nlet timestampInMilliseconds =
(timeInSeconds * 1000) + (offsetInNanos / 1000000);\n\nmsg.payload = [\n  {\n
    \"timestamp(milliseconds_since_epoch)\": timestampInMilliseconds,\n
    \"value\": data.propertyValues[0].value.doubleValue\n  },\n  {\n
    \"name\": data.propertyAlias,\n
    \"quality\": data.propertyValues[0].quality\n  }]\n\nreturn msg",
    "outputs": 1,
    "timeout": "",
    "noerr": 0,
    "initialize": "",
    "finalize": "",
    "libs": [],
    "x": 560,
```

```
    "y": 340,  
    "wires": [  
      [  
        "string"  
      ]  
    ]  
  }  
]
```

Para ver opciones de configuración adicionales, consulte el [node-red-contrib-influxdb](#) repositorio de Node-RED. GitHub

Configure el escritor InfluxDB

El nodo de escritura de InfluxDB es el componente final de su flujo de retención de datos, responsable de almacenar sus datos industriales en la base de datos local de InfluxDB. Este almacenamiento local es importante para mantener la visibilidad operativa durante las interrupciones de la red y proporcionar acceso inmediato a los datos para las aplicaciones en las que el tiempo es crucial.

1. Instale el `node-red-contrib-influxdb` paquete mediante la opción Administrar paleta. Este paquete proporciona los nodos necesarios para conectar Node-RED con InfluxDB.
2. Agregue un nodo de salida de InfluxDB a su flujo. Este nodo se encargará de la escritura real de los datos en su base de datos de InfluxDB.
3. Configure las propiedades del servidor para establecer una conexión segura con su instancia de InfluxDB:
 - a. Establezca la versión en 2.0: esto especifica que se está conectando a InfluxDB v2.x, que usa una API diferente a la de las versiones anteriores
 - b. Establece la URL en `http://127.0.0.1:8086`: esto apunta a tu instancia local de InfluxDB
 - c. Introduzca su token de autenticación de InfluxDB. Este token seguro autoriza la conexión a su base de datos. Generó el token durante el [Configure el almacenamiento local con InfluxDB](#) procedimiento.
4. Especifique los parámetros de la ubicación de almacenamiento para definir dónde y cómo se almacenarán los datos:

- a. Introduzca el nombre de su organización de InfluxDB: la organización es un espacio de trabajo para un grupo de usuarios, al que pertenecen sus grupos y paneles. Para obtener más información, consulte [Administrar](#) organizaciones en. InfluxData Documentation
- b. Especifique el depósito de InfluxDB (por ejemplo, WindFarmData): el depósito equivale a una base de datos en los sistemas tradicionales y sirve como contenedor para los datos de sus series temporales
- c. Establezca la medición de InfluxDB (por ejemplo, TurbineData): la medición es similar a una tabla en las bases de datos relacionales y organiza los puntos de datos relacionados

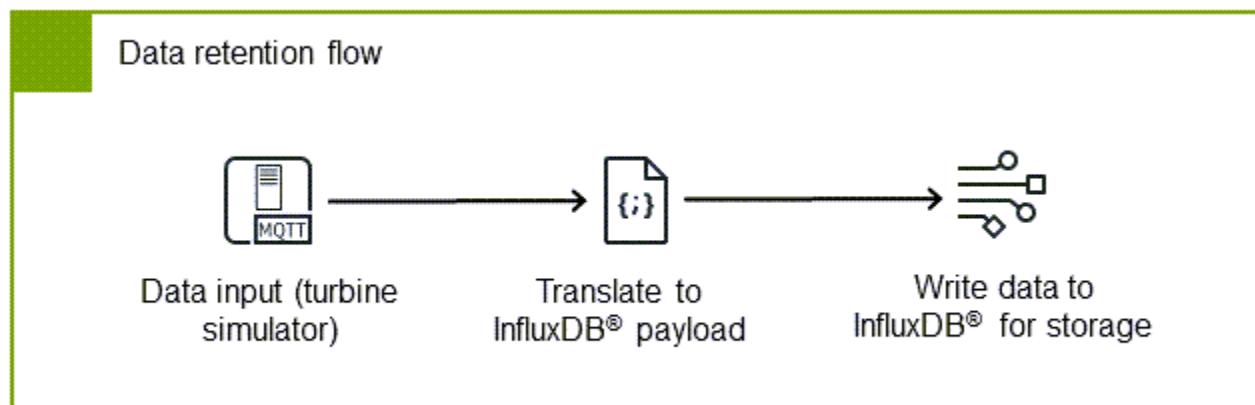
Note

Busca el nombre de tu organización en la barra lateral izquierda de la instancia de InfluxDB. Los conceptos de organización, grupo y medición son fundamentales para el modelo de organización de datos de InfluxDB. Para obtener más información, consulte la documentación de [InfluxDB](#).

Implemente y verifique el flujo de retención

Tras configurar todos los componentes del flujo de retención de datos, debe implementar y comprobar que el sistema funciona correctamente. Esta verificación garantiza que sus datos industriales se almacenen adecuadamente a nivel local para su acceso y análisis inmediatos.

1. Conecte los tres nodos como se muestra en el diagrama de flujo de retención de datos. Esto crea un proceso completo desde la suscripción de datos hasta el almacenamiento local.



2. Elija Deploy para aplicar los cambios y activar el flujo. Esto inicia el proceso de recopilación y almacenamiento de datos.
3. Utilice el explorador de datos de InfluxDB para consultar y visualizar sus datos. Esta herramienta le permite verificar que los datos se almacenan correctamente y crear visualizaciones iniciales de los datos de sus series temporales.

En el explorador de datos, debería poder ver cómo se registran las mediciones de la velocidad del viento a lo largo del tiempo, lo que confirma que todo el proceso, desde la generación de datos hasta el almacenamiento local, funciona correctamente.

Para obtener más información, consulte [Consulta en el explorador de datos](#) en el InfluxData Documentation.

Con el flujo de publicación de datos y el flujo de retención de datos implementados, ahora tiene un sistema completo que envía los datos a la AWS IoT SiteWise nube y, al mismo tiempo, mantiene una copia local para un acceso inmediato y una mayor resiliencia. Este enfoque de doble vía garantiza que obtenga los beneficios del análisis y el almacenamiento basados en la nube y, al mismo tiempo, mantenga la visibilidad operativa en la periferia.

Configurar Grafana para Edge SiteWise

Grafana® le permite crear paneles de monitoreo locales en tiempo real para sus datos industriales. Al visualizar los datos almacenados en InfluxDB®, puede proporcionar a los operadores información inmediata sobre el rendimiento del equipo, la eficiencia del proceso y los posibles problemas. Esta visibilidad perimetral es importante para las operaciones urgentes y para mantener la continuidad durante las interrupciones de la red.

Configuración de los orígenes de datos

Al conectar Grafana a su base de datos de InfluxDB, se crea una potente capa de visualización para sus datos industriales. Esta conexión permite disponer de paneles de control en tiempo real que los operadores pueden utilizar para tomar decisiones informadas sin depender de la nube.

1. Acceda a su instancia de Grafana de forma local desde <http://127.0.0.1:3000> en su navegador. Si es necesario habilitar TLS, puede consultar [Configurar Grafana HTTPS para un tráfico web seguro](#) en Grafana Labs la documentación.
2. Agregue una fuente de datos de InfluxDB que apunte al segmento de series temporales de InfluxDB donde Node-RED escribe los datos. Por ejemplo, WindFarmData. Esta conexión establece el vínculo entre los datos almacenados y la plataforma de visualización.

3. Para obtener instrucciones detalladas, consulte [Configurar la fuente de datos de InfluxDB](#) en la Grafana Labs documentación.

Cree un panel de Grafana para SiteWise los datos de Edge

La creación de un panel es el paso final para crear su solución de monitoreo local. Los paneles proporcionan representaciones visuales de sus datos industriales, lo que facilita la identificación de tendencias, anomalías y posibles problemas de un vistazo.

- Siga la guía para crear un panel de control. Para obtener más información, consulte [Crear su primer panel](#) en la Grafana Labs documentación. En esta plantilla se presupone que tu depósito tiene un nombre WindFarmData y una medida TurbineData.

También puede utilizar la guía de inicio rápido importando la plantilla de panel de ejemplo proporcionada para crear rápidamente un panel con un gráfico de series temporales para los datos que Node-RED generó en la sección anterior. Esta plantilla proporciona un punto de partida que puede personalizar para satisfacer sus necesidades específicas de supervisión.

```
{
  "__inputs": [
    {
      "name": "DS_WINDFARM-DEMO",
      "label": "windfarm-demo",
      "description": "",
      "type": "datasource",
      "pluginId": "influxdb",
      "pluginName": "InfluxDB"
    }
  ],
  "__elements": {},
  "__requires": [
    {
      "type": "grafana",
      "id": "grafana",
      "name": "Grafana",
      "version": "11.6.0-pre"
    },
    {
      "type": "datasource",
      "id": "influxdb",
      "name": "InfluxDB",
```

```
    "version": "1.0.0"
  },
  {
    "type": "panel",
    "id": "timeseries",
    "name": "Time series",
    "version": ""
  }
],
"annotations": {
  "list": [
    {
      "builtIn": 1,
      "datasource": {
        "type": "grafana",
        "uid": "-- Grafana --"
      },
      "enable": true,
      "hide": true,
      "iconColor": "rgba(0, 211, 255, 1)",
      "name": "Annotations & Alerts",
      "type": "dashboard"
    }
  ]
},
"editable": true,
"fiscalYearStartMonth": 0,
"graphTooltip": 0,
"id": null,
"links": [],
"panels": [
  {
    "datasource": {
      "type": "influxdb",
      "uid": "${DS_WINDFARM-DEMO}"
    },
    "fieldConfig": {
      "defaults": {
        "color": {
          "mode": "palette-classic"
        },
        "custom": {
          "axisBorderShow": false,
          "axisCenteredZero": false,
```

```
    "axisColorMode": "text",
    "axisLabel": "",
    "axisPlacement": "auto",
    "barAlignment": 0,
    "barWidthFactor": 0.6,
    "drawStyle": "line",
    "fillOpacity": 0,
    "gradientMode": "none",
    "hideFrom": {
      "legend": false,
      "tooltip": false,
      "viz": false
    },
    "insertNulls": false,
    "lineInterpolation": "linear",
    "lineWidth": 1,
    "pointSize": 5,
    "scaleDistribution": {
      "type": "linear"
    },
    "showPoints": "auto",
    "spanNulls": false,
    "stacking": {
      "group": "A",
      "mode": "none"
    },
    "thresholdsStyle": {
      "mode": "off"
    }
  },
  "mappings": [],
  "thresholds": {
    "mode": "absolute",
    "steps": [
      {
        "color": "green"
      },
      {
        "color": "red",
        "value": 80
      }
    ]
  }
},
```

```

    "overrides": []
  },
  "gridPos": {
    "h": 8,
    "w": 12,
    "x": 0,
    "y": 0
  },
  "id": 1,
  "options": {
    "legend": {
      "calcs": [],
      "displayMode": "list",
      "placement": "bottom",
      "showLegend": true
    },
    "tooltip": {
      "hideZeros": false,
      "mode": "single",
      "sort": "none"
    }
  },
  "pluginVersion": "11.6.0-pre",
  "targets": [
    {
      "datasource": {
        "type": "influxdb",
        "uid": "${DS_WINDFARM-DEMO}"
      },
      "query": "from(bucket: \"WindFarmData\")\n |> range(start:
v.timeRangeStart, stop: v.timeRangeStop)\n |> filter(fn: (r) => r[\"_measurement
\"] == \"TurbineData\")\n |> filter(fn: (r) => r[\"_field\"] == \"value\")\n
|> filter(fn: (r) => r[\"name\"] == \"/Renton/WindFarm/Turbine/WindSpeed\")\n
|> filter(fn: (r) => r[\"quality\"] == \"GOOD\")\n |> aggregateWindow(every:
v.windowPeriod, fn: mean, createEmpty: false)\n |> yield(name: \"mean\")",
      "refId": "A"
    }
  ],
  "title": "Panel Title",
  "type": "timeseries"
}
],
"schemaVersion": 41,
"tags": [],

```

```
"templating": {
  "list": []
},
"time": {
  "from": "now-6h",
  "to": "now"
},
"timepicker": {},
"timezone": "browser",
"title": "demo dashboard",
"uid": "fejc0t08o6d4wb",
"version": 1,
"weekStart": ""
}
```

Configure integraciones de código abierto con Docker (Linux)

Para agilizar el proceso de implementación, puede usar Docker para configurar Node-RED®, InfluxDB® y Grafana® en un entorno Linux. Este método utiliza contenedores preconfigurados, lo que permite una implementación rápida y una administración más sencilla de los componentes.

Requisitos previos de configuración de Docker

Antes de empezar, compruebe que tiene lo siguiente:

- Una puerta de enlace V3 habilitada para MQTT. Para obtener más información, consulte [Puertas de enlace V3 habilitadas para MQTT para Edge AWS IoT SiteWise](#).
- El complemento Docker Compose. Para ver los pasos de instalación, consulta [Instalar el Docker Compose complemento](#) en la documentación de los manuales de Docker.

Implemente los servicios

Esta implementación ejecuta SiteWise Edge, InfluxDB, Node-RED y Grafana en el mismo host.

Configuración del entorno

1. Obtenga acceso root:

```
sudo -i
```

2. Cree un archivo.env o exporte estas variables de entorno:

```
export INFLUXDB_PASSWORD=your-secure-influxdb-password
export INFLUXDB_TOKEN=your-secure-influxdb-token
export GRAFANA_PASSWORD=your-secure-grafana-password
```

Configure la red Docker

- Cree una red puente con el nombre `SiteWiseEdgeNodeRedDemoNetwork`.

```
docker network create --driver=bridge SiteWiseEdgeNodeRedDemoNetwork
```

Prepare el Docker Compose archivo

Copie el contenido del siguiente archivo YAML en su dispositivo de puerta de enlace SiteWise Edge.

Amplíe para ver el ejemplo del archivo Docker Compose YAML

```
services:
  influxdb:
    image: influxdb:latest
    container_name: influxdb
    ports:
      - "127.0.0.1:8086:8086"
    volumes:
      - influxdb-storage:/influxdbv2
    environment:
      - DOCKER_INFLUXDB_INIT_MODE=setup
      - DOCKER_INFLUXDB_INIT_USERNAME=admin
      - DOCKER_INFLUXDB_INIT_PASSWORD=${INFLUXDB_PASSWORD}
      - DOCKER_INFLUXDB_INIT_ORG=iot-sitewise-edge
      - DOCKER_INFLUXDB_INIT_BUCKET=WindFarmData
      - DOCKER_INFLUXDB_INIT_RETENTION=0
      - DOCKER_INFLUXDB_INIT_ADMIN_TOKEN=${INFLUXDB_TOKEN}
    networks:
      - SiteWiseEdgeNodeRedDemoNetwork
    restart: unless-stopped

  grafana:
    image: grafana/grafana:latest
    container_name: grafana
    ports:
```

```
- "127.0.0.1:3000:3000"
volumes:
- grafana-storage:/var/lib/grafana
- ./grafana/provisioning:/etc/grafana/provisioning
environment:
- GF_SECURITY_ADMIN_USER=admin
- GF_SECURITY_ADMIN_PASSWORD=${GRAFANA_PASSWORD}
- GF_INSTALL_PLUGINS=grafana-clock-panel,grafana-simple-json-datasource
- GF_PATHS_PROVISIONING=/etc/grafana/provisioning
- GF_PATHS_CONFIG=/etc/grafana/grafana.ini
- GF_LOG_LEVEL=info
configs:
- source: grafana_datasource
  target: /etc/grafana/provisioning/datasources/influxdb.yaml
- source: grafana_preload_dashboard_config
  target: /etc/grafana/provisioning/dashboards/dashboard.yml
- source: grafana_preload_dashboard
  target: /etc/grafana/provisioning/dashboards/demo_dashboard.json
depends_on:
- influxdb
networks:
- SiteWiseEdgeNodeRedDemoNetwork
restart: unless-stopped

nodered:
build:
  context: .
  dockerfile_inline: |
    FROM nodered/node-red:latest
    RUN npm install node-red-contrib-influxdb
container_name: nodered
ports:
- "127.0.0.1:1880:1880"
volumes:
- node_red_data:/data
environment:
- NODE_RED_ENABLE_SAFE_MODE=false
- NODE_RED_ENABLE_PALETTE_EDIT=true
- NODE_RED_AUTO_INSTALL_MODULES=true
configs:
- source: nodered_flows
  target: /data/flows.json
- source: nodered_settings
  target: /data/settings.js
```

```
- source: nodered_flows_cred
  target: /data/flows_cred.json
depends_on:
- influxdb
networks:
- SiteWiseEdgeNodeRedDemoNetwork
restart: unless-stopped

volumes:
influxdb-storage:
grafana-storage:
node_red_data:

networks:
SiteWiseEdgeNodeRedDemoNetwork:
  external: true

configs:
grafana_datasource:
  content: |
    apiVersion: 1
    datasources:
      - name: windfarm-demo
        type: influxdb
        access: proxy
        url: http://influxdb:8086
        jsonData:
          version: Flux
          organization: iot-sitewise-edge
          defaultBucket: WindFarmData
          tlsSkipVerify: true
        secureJsonData:
          token: ${INFLUXDB_TOKEN}
        editable: false

grafana_preload_dashboard_config:
  content: |
    apiVersion: 1
    providers:
      - name: "Dashboard provider"
        orgId: 1
        type: file
        options:
          path: /etc/grafana/provisioning/dashboards
```

```
grafana_preload_dashboard:
  content: |
    {
      "annotations": {
        "list": [
          {
            "builtIn": 1,
            "datasource": {
              "type": "grafana",
              "uid": "-- Grafana --"
            },
            "enable": true,
            "hide": true,
            "iconColor": "rgba(0, 211, 255, 1)",
            "name": "Annotations & Alerts",
            "type": "dashboard"
          }
        ]
      },
      "editable": true,
      "fiscalYearStartMonth": 0,
      "graphTooltip": 0,
      "id": 1,
      "links": [],
      "panels": [
        {
          "datasource": {
            "type": "influxdb",
            "uid": "PEB0DCBF338B3CEB2"
          },
          "fieldConfig": {
            "defaults": {
              "color": {
                "mode": "palette-classic"
              },
              "custom": {
                "axisBorderShow": false,
                "axisCenteredZero": false,
                "axisColorMode": "text",
                "axisLabel": "",
                "axisPlacement": "auto",
                "barAlignment": 0,
                "barWidthFactor": 0.6,
```

```
    "drawStyle": "line",
    "fillOpacity": 0,
    "gradientMode": "none",
    "hideFrom": {
      "legend": false,
      "tooltip": false,
      "viz": false
    },
    "insertNulls": false,
    "lineInterpolation": "linear",
    "lineWidth": 1,
    "pointSize": 5,
    "scaleDistribution": {
      "type": "linear"
    },
    "showPoints": "auto",
    "spanNulls": false,
    "stacking": {
      "group": "A",
      "mode": "none"
    },
    "thresholdsStyle": {
      "mode": "off"
    }
  },
  "mappings": [],
  "thresholds": {
    "mode": "absolute",
    "steps": [
      {
        "color": "green"
      },
      {
        "color": "red",
        "value": 80
      }
    ]
  }
},
"overrides": []
},
"gridPos": {
  "h": 8,
  "w": 12,
```

```

        "x": 0,
        "y": 0
    },
    "id": 1,
    "options": {
        "legend": {
            "calcs": [],
            "displayMode": "list",
            "placement": "bottom",
            "showLegend": true
        },
        "tooltip": {
            "hideZeros": false,
            "mode": "single",
            "sort": "none"
        }
    },
    "pluginVersion": "11.6.0",
    "targets": [
        {
            "datasource": {
                "type": "influxdb",
                "uid": "PEB0DCBF338B3CEB2"
            },
            "query": "from(bucket: \"WindFarmData\")\n  |> range(start:
v.timeRangeStart, stop: v.timeRangeStop)\n  |> filter(fn: (r) => r[\"_measurement
\"] == \"TurbineData\")\n  |> filter(fn: (r) => r[\"_field\"] == \"value\")\n
  |> filter(fn: (r) => r[\"name\"] == \"/Renton/WindFarm/Turbine/WindSpeed\")\n
  |> filter(fn: (r) => r[\"quality\"] == \"GOOD\")\n  |> aggregateWindow(every:
v.windowPeriod, fn: mean, createEmpty: false)\n  |> yield(name: \"mean\")",
            "refId": "A"
        }
    ],
    "title": "Wind Speed",
    "type": "timeseries"
}
],
"preload": false,
"schemaVersion": 41,
"tags": [],
"templating": {
    "list": []
},
"time": {

```

```
    "from": "now-6h",
    "to": "now"
  },
  "timepicker": {},
  "timezone": "browser",
  "title": "Demo Dashboard",
  "uid": "eejtureqjo9a8c",
  "version": 2
}
```

nodered_flows:

```
content: |
  [
    {
      "id": "95fce448fdd43b47",
      "type": "tab",
      "label": "Demo Flow",
      "disabled": false,
      "info": ""
    },
    {
      "id": "5f63740b66af3386",
      "type": "mqtt out",
      "z": "95fce448fdd43b47",
      "name": "Publish to MQTT broker",
      "topic": "/Renton/WindFarm/Turbine/WindSpeed",
      "qos": "1",
      "retain": "",
      "respTopic": "",
      "contentType": "",
      "userProps": "",
      "correl": "",
      "expiry": "",
      "broker": "5744207557fa19be",
      "x": 830,
      "y": 200,
      "wires": []
    },
    {
      "id": "8f2eb590d596679b",
      "type": "function",
      "z": "95fce448fdd43b47",
      "name": "Translate to SiteWise payload",
```

```

    "func": "let input = msg.payload;\nlet output = {};\n\noutput[\"propertyAlias\n\"] = input.name;\n\nlet propertyVal = {}\n\nlet timeInSeconds =
  Math.floor(input.timestamp / 1000);\nlet offsetInNanos = (input.timestamp % 1000) *
  1000000;\n\npropertyVal[\"timestamp\"] = {\n  \"timeInSeconds\": timeInSeconds,\n  \"offsetInNanos\": offsetInNanos,\n};\n\npropertyVal[\"quality\"] = input.quality
\n\nlet typeNameConverter = {\n  \"number\": (x) => Number.isInteger(x) ?
  \"integerValue\" : \"doubleValue\", \n  \"boolean\": (x) => \"booleanValue\", \n
  \"string\": (x) => \"stringValue\", \n}\n\nlet typeName = typeNameConverter[typeof
input.value](input.value)\n\npropertyVal[\"value\"] = {}\n\npropertyVal[\"value\"]
[typeName] = input.value;\n\noutput[\"propertyValues\"] = [propertyVal]\n\nreturn {\n
  payload: JSON.stringify(output)\n};",
    "outputs": 1,
    "timeout": "",
    "noerr": 0,
    "initialize": "",
    "finalize": "",
    "libs": [],
    "x": 530,
    "y": 200,
    "wires": [
      [
        "5f63740b66af3386"
      ]
    ]
  },
  {
    "id": "4b78cbdea5e3258c",
    "type": "inject",
    "z": "95fce448fdd43b47",
    "name": "Turbine Simulator",
    "props": [
      {
        "p": "payload.timestamp",
        "v": "",
        "vt": "date"
      },
      {
        "p": "payload.quality",
        "v": "GOOD",
        "vt": "str"
      },
      {
        "p": "payload.value",
        "v": "$$random()",

```

```
    "vt": "jsonata"
  },
  {
    "p": "payload.name",
    "v": "/Renton/WindFarm/Turbine/WindSpeed",
    "vt": "str"
  }
],
"repeat": "1",
"crontab": "",
"once": false,
"onceDelay": "",
"topic": "",
"x": 270,
"y": 200,
"wires": [
  [
    "8f2eb590d596679b"
  ]
]
},
{
  "id": "b658bf337ea2e316",
  "type": "influxdb out",
  "z": "95fce448fdd43b47",
  "influxdb": "2f1c38495035d2e4",
  "name": "Store data in InfluxDB",
  "measurement": "TurbineData",
  "precision": "",
  "retentionPolicy": "",
  "database": "",
  "retentionPolicyV18Flux": "",
  "org": "iot-sitewise-edge",
  "bucket": "WindFarmData",
  "x": 840,
  "y": 340,
  "wires": []
},
{
  "id": "9432d39af35b202f",
  "type": "function",
  "z": "95fce448fdd43b47",
  "name": "Translate to InfluxDB payload",
```

```

    "func": "let data = msg.payload;\n\nlet timeInSeconds =
data.propertyValues[0].timestamp.timeInSeconds;\nlet offsetInNanos =
data.propertyValues[0].timestamp.offsetInNanos;\nlet timestampInMilliseconds =
(timeInSeconds * 1000) + (offsetInNanos / 1000000);\n\nmsg.payload = [\n  {\n
    \"timestamp(milliseconds_since_epoch)\": timestampInMilliseconds,\n
    \"value\": data.propertyValues[0].value.doubleValue\n  },\n  {\n
    \"name\": data.propertyAlias,\n
    \"quality\": data.propertyValues[0].quality\n  }]\n\nreturn msg",
  "outputs": 1,
  "timeout": "",
  "noerr": 0,
  "initialize": "",
  "finalize": "",
  "libs": [],
  "x": 560,
  "y": 340,
  "wires": [
    [
      "b658bf337ea2e316"
    ]
  ]
},
{
  "id": "b689403d2c80816b",
  "type": "mqtt in",
  "z": "95fce448fdd43b47",
  "name": "Subscribe to MQTT broker",
  "topic": "/Renton/WindFarm/Turbine/WindSpeed",
  "qos": "1",
  "datatype": "auto-detect",
  "broker": "5744207557fa19be",
  "nl": false,
  "rap": true,
  "rh": 0,
  "inputs": 0,
  "x": 290,
  "y": 340,
  "wires": [
    [
      "9432d39af35b202f"
    ]
  ]
},
{

```

```
"id": "4f59bed8e829fc35",
"type": "comment",
"z": "95fce448fdd43b47",
"name": "Data Publish Flow",
"info": "dfgh",
"x": 270,
"y": 160,
"wires": []
},
{
  "id": "b218c7fc58c8b6e7",
  "type": "comment",
  "z": "95fce448fdd43b47",
  "name": "Data Retention flow",
  "info": "",
  "x": 270,
  "y": 300,
  "wires": []
},
{
  "id": "5744207557fa19be",
  "type": "mqtt-broker",
  "name": "emqx",
  "broker": "emqx",
  "port": "1883",
  "clientid": "",
  "autoConnect": true,
  "usetls": false,
  "protocolVersion": "5",
  "keepalive": 15,
  "cleansession": true,
  "autoUnsubscribe": true,
  "birthTopic": "",
  "birthQos": "0",
  "birthPayload": "",
  "birthMsg": {},
  "closeTopic": "",
  "closePayload": "",
  "closeMsg": {},
  "willTopic": "",
  "willQos": "0",
  "willPayload": "",
  "willMsg": {},
  "userProps": ""
```

```
    "sessionExpiry": ""
  },
  {
    "id": "2f1c38495035d2e4",
    "type": "influxdb",
    "hostname": "influxdb",
    "port": 8086,
    "protocol": "http",
    "database": "",
    "name": "InfluxDB",
    "usetls": false,
    "tls": "",
    "influxdbVersion": "2.0",
    "url": "http://influxdb:8086",
    "timeout": "",
    "rejectUnauthorized": false
  }
]
```

nodered_flows_cred:

```
content: |
  {
    "2f1c38495035d2e4": {
      "token": "${INFLUXDB_TOKEN}"
    }
  }
```

nodered_settings:

```
content: |
  module.exports = {
    flowFile: 'flows.json',
    credentialSecret: false,
    adminAuth: null,
    editorTheme: {
      projects: {
        enabled: false
      }
    }
  }
```

Actualice la implementación de SiteWise Edge

1. Vaya a la [consola AWS IoT](#).

2. Elija dispositivos Greengrass en el menú de navegación izquierdo, en la sección Administrar, y luego Dispositivos principales.
3. Seleccione el dispositivo principal conectado a su SiteWise Edge Gateway.
4. Elija la pestaña Implementaciones y, a continuación, seleccione el valor del ID de implementación.
5. Elija Acciones y, a continuación, seleccione Revisar.
6. Lea el mensaje emergente y, a continuación, seleccione Revisar Deployment.
7. En el paso 2: Seleccione los componentes, seleccione los siguientes componentes y, a continuación, elija Siguiente.
 - `aws.greengrass.clientdevices.mqtt.EMQX`
 - `aws.iot.SiteWiseEdgePublisher`
8. En el paso 3: Configurar los componentes, seleccione el valor del `aws.greengrass.clientdevices.mqtt.EMQX` componente y añada la siguiente configuración de red:

```
{
  "emqxConfig": {
    "authorization": {
      "no_match": "allow"
    },
    "listeners": {
      "tcp": {
        "default": {
          "enabled": true,
          "enable_authn": false
        }
      }
    }
  },
  "authMode": "bypass",
  "dockerOptions": "-p 127.0.0.1:1883:1883 --
network=SiteWiseEdgeNodeRedDemoNetwork",
  "requiresPrivilege": "true"
}
```

9. Elija Siguiente.
10. En el paso 4: Configurar los ajustes avanzados, seleccione Siguiente.

11. Elija Implementar.

Inicie los servicios

1. Inicie los servicios con el archivo Docker Compose. Ejecute el siguiente comando en el directorio que contiene el `compose.yaml` archivo.

```
docker compose up -d
```

2. Cree un túnel SSH para acceder a los servicios:

```
ssh -i path_to_your_ssh_key -L 1880:127.0.0.1:1880 -L 3000:127.0.0.1:3000 -L 8086:127.0.0.1:8086 username@gateway_ip_address
```

Esta implementación crea los siguientes servicios en: `SiteWiseEdgeNodeRedDemoNetwork` network

InfluxDB v2 (puerto 8086)

Incluye una organización preconfigurada (`iot-sitewise-edge`), un depósito de `WindFarmData` InfluxDB y credenciales de administrador

Node-RED (puerto 1880)

Incluye nodos InfluxDB y flujos preconfigurados para la integración AWS IoT SiteWise

Grafana (puerto 3000)

Incluye un usuario administrador, una fuente de datos de InfluxDB y un panel de monitoreo

Acceda a los servicios

Tras la implementación, acceda a los servicios con URLs las siguientes credenciales:

Note

Puede acceder a cada servicio desde su host o desde la máquina de puerta de enlace.

Detalles de acceso al servicio

Servicio	URL	Credenciales
Node-RED	http://127.0.0.1:1880	No se requieren credenciales
InfluxDB	http://127.0.0.1:8086	Nombre de usuario: admin Contraseña: \$INFLUXDB_PASSWORD
Grafana	http://127.0.0.1:3000	Nombre de usuario: admin Contraseña: \$GRAFANA_PASSWORD

Verifique la implementación

Para garantizar que la implementación se realice correctamente, realice las siguientes comprobaciones:

1. En el caso de Node-RED, compruebe la presencia de dos flujos precargados:
 - Flujo de publicación de datos
 - Flujo de retención de datos
2. Para AWS IoT SiteWise, en la AWS IoT SiteWise consola, confirme la presencia de un flujo de datos con el alias/Renton/WindFarm/Turbine/WindSpeed.
3. En el caso de InfluxDB, utilice el explorador de datos para verificar el almacenamiento de los datos en la TurbineData medición dentro del WindFarmData depósito.
4. En el caso de Grafana, consulta el panel para confirmar la visualización de los datos de series temporales generados por Node-RED.

Procesa datos para integraciones de código abierto

Los datos se pueden procesar (por ejemplo, mediante la transformación o la agregación) en diferentes etapas y utilizando diversas herramientas, cada una de las cuales cumple diferentes requisitos de monitoreo.

Procese los datos con los nodos Node-RED

Transforme sus datos en tiempo real utilizando los nodos de procesamiento integrados de Node-RED®. Configure estos nodos a través de la consola Node-RED para crear su canalización de datos.

Nodos de transformación de datos

Transforme puntos de datos individuales, de forma similar a Transforms in AWS IoT SiteWise, mediante estos nodos:

- cambiar de nodo: realiza modificaciones de valores sencillas en los datos.
- nodo de función: permite JavaScript transformaciones personalizadas para el procesamiento de datos complejos.

Nodos de cálculo de métricas

Combine varios puntos de datos en una sola salida, de forma similar a Metrics in AWS IoT SiteWise, mediante estos nodos:

- nodo por lotes: agrupa varios mensajes para su procesamiento por lotes.
- nodo de unión: combina varios flujos de datos en una sola salida.
- nodo agregador: calcula las métricas agregadas a partir de varios puntos de datos.

Para ver opciones de nodos adicionales, consulte la biblioteca [Node-RED](#).

Cree tareas de InfluxDB

Si bien Node-RED destaca en el procesamiento básico de datos con una configuración rápida, los cálculos métricos complejos pueden resultar complicados en la programación basada en flujos. Las tareas de InfluxDB® ofrecen una alternativa a través de los scripts Flux programados para necesidades de procesamiento avanzadas.

Utilice las tareas de InfluxDB para:

- Agregaciones estadísticas en grandes conjuntos de datos
- Operaciones matemáticas en múltiples propiedades
- Mediciones derivadas de múltiples fuentes

Características de la tarea

- Ejecución programada: ejecuta tareas basadas en expresiones cron
- Procesamiento por lotes: optimice las operaciones para datos de series temporales
- Recuperación de errores: reintente automáticamente las operaciones fallidas
- Supervisión: realice un seguimiento de la ejecución mediante registros detallados

Administre las tareas a través de la interfaz de usuario, la API o la CLI de InfluxDB. Para obtener más información, consulte [Procesar datos con tareas de InfluxDB](#).

Usa las transformaciones de Grafana

Transforme la visualización de datos en Grafana® sin modificar los datos de origen en InfluxDB. Las transformaciones de Grafana se aplican solo a la capa de visualización.

- Visual Builder: cree transformaciones sin escribir código
- Vista previa en vivo: vea los resultados de la transformación en tiempo real
- Fuentes múltiples: procese datos de múltiples fuentes de bases de datos
- Almacenamiento eficiente: transforme los datos en el momento de la visualización sin almacenar resultados intermedios

Para obtener más información, consulte [Transformar datos](#).

Solución de problemas de integraciones de código abierto

Para obtener más información sobre la solución de problemas relacionados con las integraciones de código abierto para las puertas de enlace SiteWise Edge, consulte. [Solución de problemas de integraciones de código abierto en Edge](#)

Transmisiones clásicas, pasarelas V2 para Edge AWS IoT SiteWise

Conozca las características y limitaciones de las transmisiones clásicas, las pasarelas V2 para AWS IoT SiteWise Edge.

La pasarela V2 Classic Streams mantiene la funcionalidad tradicional que ya existía en las AWS IoT SiteWise implementaciones anteriores antes de la introducción de las pasarelas V3 habilitadas para MQTT. Estas pasarelas SiteWise Edge se consideran pasarelas clásicas de transmisiones V2. Mantienen la compatibilidad con versiones anteriores y funcionan con el paquete

de procesamiento de datos. Si bien la pasarela Classic Streams V2 ofrece un rendimiento fiable para las configuraciones existentes, presenta limitaciones en comparación con las nuevas opciones de pasarela. En concreto, este tipo de puerta de enlace no es totalmente compatible con las funciones avanzadas disponibles en el destino de puerta de enlace V3 habilitado para MQTT. Para utilizar el protocolo de mensajería MQTT, puede crear una nueva puerta de enlace V3 habilitada para MQTT. Para obtener más información, consulte [Puertas de enlace V3 habilitadas para MQTT para Edge AWS IoT SiteWise](#).

Temas

- [Utilice paquetes para recopilar y procesar datos en Edge SiteWise](#)
- [Configure el AWS IoT SiteWise componente de publicación](#)
- [Gestor de destinos y AWS IoT Greengrass transmisiones](#)
- [Configure las capacidades perimetrales en AWS IoT SiteWise Edge](#)
- [Configure el procesamiento de datos AWS IoT SiteWise perimetrales para modelos y activos](#)

Utilice paquetes para recopilar y procesar datos en Edge SiteWise

Note

La característica de paquete de procesamiento de datos (DPP) ya no está disponible para los nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte [Cambio en la disponibilidad del paquete de procesamiento de datos](#).

AWS IoT SiteWise Las pasarelas Edge utilizan diferentes paquetes para determinar cómo recopilar y procesar los datos.

Actualmente están disponibles los siguientes paquetes:

- Paquete de recopilación de datos: utilice este paquete para recopilar sus datos industriales y enviarlos a destinos AWS en la nube. De forma predeterminada, este paquete se habilita automáticamente para su puerta de enlace SiteWise Edge.
- Paquete de procesamiento de datos: utilice este paquete para habilitar la comunicación de la puerta de enlace SiteWise Edge con modelos de activos y activos configurados de forma perimetral. Puede utilizar la configuración de periferia para controlar qué datos de activos se van a computar y procesar en las instalaciones. A continuación, puede enviar sus datos a AWS IoT

SiteWise u otros AWS servicios. Para obtener más información sobre el paquete de procesamiento de datos, consulte [the section called “Configure el procesamiento de datos perimetrales”](#).

Actualización de paquetes

Important

La actualización de las versiones de los paquetes de procesamiento de datos anteriores a la 2.0.x (incluidas) a la versión 2.1.x provocará la pérdida de datos de las mediciones almacenadas localmente.

SiteWise Las pasarelas perimetrales utilizan diferentes paquetes para determinar cómo recopilar y procesar los datos. Puede utilizar la AWS IoT SiteWise consola para actualizar los paquetes.

Para actualizar paquetes (consola)

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación de la izquierda, selecciona las puertas de enlace Edge en la sección Edge.
3. En la lista de puertas de enlace, elija la puerta de enlace SiteWise Edge con los paquetes que desee actualizar.
4. En la sección de Configuración de puerta de enlace, seleccione Actualizaciones de software disponibles.
5. En la página Editar versiones de software, selecciona Actualizaciones.

Note

Solo puede actualizar los paquetes que estén habilitados. Para ver la lista de paquetes que están habilitados para esta puerta de enlace de SiteWise Edge, seleccione Descripción general y, a continuación, consulte la sección de capacidades de Edge.

6. En la página de versiones del software de edición, en la sección Actualizaciones de componentes de puerta de enlace, haga lo siguiente:
 - Para actualizar el Recopilador OPC UA, elija una versión y, a continuación, Implementar.
 - Para actualizar el Publicador, elija una versión y, a continuación, Implementar.

- Para actualizar el Paquete de procesamiento de datos, elija una versión y, a continuación, Implementar.
7. Cuando haya terminado de implementar las nuevas versiones, elija Listo.

Si tiene problemas para actualizar los paquetes, consulte [No se han podido implementar paquetes en las puertas de enlace de SiteWise Edge](#).

Cambio en la disponibilidad de los paquetes de procesamiento de datos

Note

La característica de paquete de procesamiento de datos (DPP) ya no está disponible para los nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte [Cambio en la disponibilidad del paquete de procesamiento de datos](#).

Para obtener capacidades similares a la función del paquete de procesamiento de datos de AWS IoT, explore las [alternativas de código abierto o las integraciones de nuestros socios](#). El paquete de procesamiento de AWS IoT SiteWise datos es una función AWS IoT SiteWise que proporciona transformaciones de datos, métricas, filtrado, almacenamiento local y visualización en la periferia.

Note

AWS IoT SiteWise y la función del paquete de recopilación de datos de AWS IoT SiteWise Edge sigue estando disponible, pero la función del paquete de procesamiento de datos está entrando en modo de mantenimiento.

Opciones de migración

Explore estas opciones de migración para reemplazar la funcionalidad del paquete de procesamiento de datos.

Alternativas de código abierto

Cree canalizaciones de procesamiento de datos locales con Node-RED para la transformación de datos, InfluxDB para el almacenamiento de series temporales y Grafana para la visualización. Estas herramientas funcionan con pasarelas V3 habilitadas para MQTT a través de MQTT para

proporcionar procesamiento perimetral e información local, a la vez que sincronizan los datos con la nube. AWS

Para obtener más información, consulte [Procesar y visualizar datos con Edge y herramientas](#) de código abierto. SiteWise

Integraciones de socios

Conecte equipos y sensores industriales a través de fuentes de datos de socios externos CloudRail EasyEdge, como Litmus Edge. Estos componentes de Greengrass se han desarrollado en colaboración con más de 200 protocolos industriales AWS y son compatibles con ellos para la recopilación y el procesamiento integrales de datos.

Para obtener más información, consulte [Añadir una fuente de datos](#).

Puertas de enlace V3 habilitadas para MQTT

Las pasarelas V3 habilitadas para MQTT utilizan un protocolo MQTT ligero para una comunicación de datos eficiente y ofrecen destinos de datos flexibles, incluida la ingesta en tiempo real y la ingesta en búfer de Amazon S3. Puede implementar filtros de ruta para una recopilación de datos precisa y beneficiarse de una escalabilidad mejorada y de la alineación de los estándares de IoT. Las pasarelas V3 habilitadas para MQTT proporcionan un procesamiento de datos basado en la nube a través de servicios AWS IoT SiteWise básicos que incluyen modelos de activos, propiedades computadas, alarmas y consultas de datos históricos.

Para obtener más información, consulte [Puertas de enlace V3 habilitadas para MQTT para Edge AWS IoT SiteWise](#).

Preguntas frecuentes

¿Puedo migrar gradualmente?

Sí, puede migrar gradualmente mediante cualquier combinación de opciones de migración. Puede implementar pasarelas V3 compatibles con MQTT, alternativas de código abierto o integraciones de socios junto con las transmisiones clásicas existentes o las pasarelas V2 con el paquete de procesamiento de datos. Todas las opciones pueden enviar datos al mismo entorno. AWS IoT SiteWise

¿Cuánto tiempo puedo seguir utilizando el paquete de procesamiento de datos?

El paquete de procesamiento de datos permanece disponible para los clientes actuales en modo de mantenimiento. Recibirás un aviso anticipado si está previsto algún cambio en la

disponibilidad. Supervisa los anuncios AWS de servicio y las notificaciones de tu cuenta para ver si hay actualizaciones.

Configure el AWS IoT SiteWise componente de publicación

Tras crear una puerta de enlace de AWS IoT SiteWise Edge e instalar el software, puede configurar el componente de publicación para que la puerta de enlace de SiteWise Edge pueda exportar datos a la AWS nube. Utilice el componente de publicador para habilitar características adicionales o configurar ajustes predeterminados. Para obtener más información, consulte [AWS IoT SiteWise publisher](#) en la Guía para desarrolladores de AWS IoT Greengrass Version 2 .

Note

La configuración del editor varía según el tipo de puerta de enlace que utilice. En el caso de las pasarelas Classic Stream V2, utilice el `iotsitewise:publisher:2` espacio de nombres. Para las puertas de enlace V3 habilitadas para MQTT, utilice el espacio de nombres. `iotsitewise:publisher:3`

Console

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, seleccione Puertas de enlace Edge.
3. Seleccione la puerta de enlace SiteWise Edge para la que desee configurar el publicador.
4. En la sección Configuración del editor, elija Editar.
5. En Orden de publicación, elija una de las siguientes opciones.
 - Publique primero los datos más antiguos: la puerta de enlace SiteWise Edge publica primero los datos más antiguos en la nube de forma predeterminada.
 - Publique primero los datos más recientes: la puerta de enlace SiteWise Edge publica primero los datos más recientes en la nube.
6. (Opcional) Si no desea que la puerta de enlace SiteWise Edge comprima sus datos, anule la selección de Activar la compresión al cargar datos.
7. (Opcional) Si no desea publicar datos antiguos, seleccione Excluir los datos caducados y haga lo siguiente:

- En Periodo límite, introduzca un valor y elija una unidad. El periodo límite debe ser de entre cinco minutos y siete días. Por ejemplo, si el periodo límite es de tres días, los datos anteriores a tres días no se publicarán en la nube.
8. (Opcional) Para establecer una configuración personalizada sobre cómo se gestionan los datos en el dispositivo local, elija Configuración de almacenamiento local y haga lo siguiente:
 - a. En Periodo de retención, introduzca un número y elija una unidad. El periodo de retención debe ser de entre un minuto y 30 días, y mayor o igual que el periodo de rotación. Por ejemplo, si el período de retención es de 14 días, la puerta de enlace de SiteWise Edge elimina todos los datos del borde que hayan superado el período límite especificado después de haberlos almacenado durante 14 días.
 - b. En Periodo de rotación, introduzca un número y elija una unidad. El periodo de rotación debe ser mayor que un minuto y menor o igual que el periodo de retención. Por ejemplo, supongamos que el período de rotación es de dos días, la puerta de enlace SiteWise Edge agrupa los datos anteriores al período límite y los guarda en un solo archivo. Para las puertas de enlace autohospedadas AWS IoT Greengrass V2, la puerta de enlace SiteWise Edge transfiere un lote de datos al siguiente directorio local una vez cada dos días: `./greengrass/v2/work/aws.iot.SiteWiseEdgePublisher/exports`
 - c. En Capacidad de almacenamiento, introduzca un valor que sea mayor o igual a 1. Si la capacidad de almacenamiento es de 2 GB, la puerta de enlace SiteWise Edge comienza a eliminar datos cuando hay más de 2 GB de datos almacenados localmente.
 9. Seleccione Save.

AWS CLI

Use la [UpdateGatewayCapabilityConfiguration](#) API para configurar el editor.

Establezca el parámetro `capabilityNamespace` como `iotsitewise:publisher:2`.

Example: Configuración de editor para las pasarelas Classic Stream V2

Espacio de nombres del publicador: `iotsitewise:publisher:2`

```
{
  "SiteWisePublisherConfiguration": {
    "publishingOrder": "TIME_ORDER",
    "enableCompression": true,
```

```

    "dropPolicy": {
      "cutoffAge": "7d",
      "exportPolicy": {
        "retentionPeriod": "7d",
        "rotationPeriod": "6h",
        "exportSizeLimitGB": 10
      }
    }
  },
  "SiteWiseS3PublisherConfiguration": {
    "accessRoleArn": "arn:aws:iam:123456789012:role/roleName",
    "streamToS3ConfigMapping": [
      {
        "streamName": "S3_OPC-UA_Data_Collector",
        "targetBucketArn": "arn:aws:s3:::amzn-s3-demo-bucket/dataCollector",
        "publishPolicy": {
          "publishFrequency": "10m",
          "localSizeLimitGB": 10
        },
        "siteWiseImportPolicy": {
          "enableSiteWiseStorageImport": true,
          "enableDeleteAfterImport": true
        }
      }
    ]
  }
}

```

El publicador proporciona los siguientes parámetros de configuración que puede personalizar.

SiteWisePublisherConfiguration

publishingOrder

El orden en el que se publican los datos en la nube. El valor de este parámetro puede ser uno de los siguientes.

- **TIME_ORDER** (Publicar primero los datos más antiguos): De forma predeterminada, los datos más antiguos se publican primero en la nube.
- **RECENT_DATA** (Publicar primero los datos más recientes): los datos más recientes se publican primero en la nube.

`enableCompression`

Configúrelo `true` para comprimir los datos antes de publicarlos. La compresión de datos puede reducir el uso del ancho de banda.

`dropPolicy`

(Opcional) Una política que controle qué datos se publican en la nube.

`cutoffAge`

La antigüedad máxima de los datos que se van a publicar se especifica en días, horas y minutos. Por ejemplo, `7d` o `1d7h16m`. Los datos anteriores a los especificados no se envían a AWS IoT SiteWise.

Los datos anteriores al período límite no se publican en la nube. La antigüedad límite debe ser de entre cinco minutos y siete días.

Puede utilizar `m`, `h` y `d` cuando especifique una edad límite. Tenga en cuenta que `m` representa minutos, `h` representa horas y `d` representa días.

`exportPolicy`

(Opcional) Una política que administra el almacenamiento de datos en la periferia. Esta política se aplica a los datos que son anteriores a la antigüedad límite.

`retentionPeriod`

La puerta de enlace SiteWise Edge elimina del almacenamiento local cualquier dato periférico anterior al período límite una vez que se haya almacenado durante el período de retención especificado. El periodo de retención debe ser de entre un minuto y 30 días, y mayor o igual que el periodo de rotación.

Puede utilizar `m`, `h` y `d` cuando especifique un periodo de retención. Tenga en cuenta que `m` representa minutos, `h` representa horas y `d` representa días.

`rotationPeriod`

El intervalo de tiempo en que los datos se procesan por lotes y se guardan en un único archivo que es anterior al periodo límite. La puerta de enlace SiteWise Edge transfiere un lote de datos al siguiente directorio local al final de cada período de rotación: `./greengrass/v2/work/aws.iot.SiteWiseEdgePublisher/`

`exports` El periodo de rotación debe ser mayor que un minuto y menor o igual que el periodo de retención.

Puede utilizar m, h y d cuando especifique un periodo de rotación. Tenga en cuenta que m representa minutos, h representa horas y d representa días.

`exportSizeLimitGB`

El tamaño máximo permitido de datos almacenados localmente, en GB. Si se supera esta cuota, la puerta de enlace SiteWise Edge comienza a eliminar los primeros datos hasta que el tamaño de los datos almacenados localmente sea igual o inferior a la cuota. El valor de este parámetro debe ser mayor o igual que 1.

`SiteWiseS3PublisherConfiguration`

`accessRoleArn`

El rol de acceso que da AWS IoT SiteWise permiso para administrar el bucket de Amazon S3 en el que está publicando.

`streamToS3ConfigMapping`

Conjunto de configuraciones que asigna un flujo a una configuración de Amazon S3.

`streamName`

El flujo que se va a leer y publicar en la configuración de Amazon S3.

`targetBucketArn`

El ARN del bucket en el que se va a publicar.

`publishPolicy`

`publishFrequency`

La frecuencia con la que la puerta de enlace SiteWise Edge publica en el bucket de Amazon S3.

`localSizeLimitGB`

El tamaño máximo de los archivos escritos en el disco local. Si se supera este umbral, el publicador publica todos los datos almacenados en el búfer en su destino.

`siteWiseImportPolicy`

`enableSiteWiseStorageImport`

`true` Configúrelo para importar datos de un bucket de Amazon S3 al AWS IoT SiteWise almacenamiento.

`enableDeleteAfterImport`

Configure como `true` para eliminar el archivo del bucket de Amazon S3 después de su ingesta en el almacenamiento de AWS IoT SiteWise .

Gestor de destinos y AWS IoT Greengrass transmisiones

AWS IoT Greengrass El administrador de transmisiones le permite enviar datos a los siguientes Nube de AWS destinos: canales entrantes AWS IoT Analytics, transmisiones en Amazon Kinesis Data Streams, propiedades de activos u objetos AWS IoT SiteWise en Amazon Simple Storage Service (Amazon S3). Para obtener más información, consulte [Administración de flujos de datos en AWS IoT Greengrass Core](#) en la Guía para desarrolladores de AWS IoT Greengrass Version 2 .

Example: Estructura de mensajes de transmisión de datos

El siguiente ejemplo muestra la estructura de mensajes de flujo de datos requerida transmitida por el administrador de AWS IoT Greengrass flujo.

```
{
  "assetId": "string",
  "propertyAlias": "string",
  "propertyId": "string",
  "propertyValues": [
    {
      "quality": "string",
      "timestamp": {
        "offsetInNanos": number,
        "timeInSeconds": number
      },
      "value": {
        "booleanValue": boolean,
        "doubleValue": number,
        "integerValue": number,
        "stringValue": "string"
      }
    }
  ]
}
```

```
}
```

Note

El mensaje de flujo de datos debe incluir (`assetId` y `propertyId`) o `propertyAlias` en su estructura.

assetId

(Opcional) ID del activo que se desea actualizar.

propertyAlias

(Opcional) Alias que identifica la propiedad, como una ruta de flujo de datos del servidor OPC UA. Por ejemplo:

```
/company/windfarm/3/turbine/7/temperature
```

Para obtener más información, consulte [Administrar flujos de datos](#) en la Guía del AWS IoT SiteWise usuario.

propertyId

(Opcional) ID de la propiedad de activo de esta entrada.

propertyValues

(Obligatorio) Lista de valores de propiedad que se va a cargar. Puede especificar hasta 10 elementos de la matriz `propertyValues`.

quality

(Opcional) La calidad del valor de la propiedad de activo.

timestamp

(Obligatorio) Marca de tiempo del valor de propiedad del activo.

offsetInNanos

(Opcional) Desfase en nanosegundos respecto a `timeInSeconds`.

`timeInSeconds`

(Obligatorio) Marca de tiempo, en segundos, en formato epoch de Unix. Los datos fraccionarios de nanosegundos los proporciona `offsetInNanos`.

`value`

(Obligatorio) Valor de la propiedad del activo.

Note

Solo puede haber uno de los siguientes valores en el campo `value`.

`booleanValue`

(Opcional) Datos de propiedades del activo de tipo booleano (`true` o `false`).

`doubleValue`

(Opcional) Datos de propiedad del activo de tipo doble (número de coma flotante).

`integerValue`

(Opcional) Datos de propiedad del activo de tipo entero (número entero).

`stringValue`

(Opcional) Datos de propiedad del activo de tipo cadena (secuencia de caracteres).

Configure las capacidades perimetrales en AWS IoT SiteWise Edge

Note

La característica de paquete de procesamiento de datos (DPP) ya no está disponible para los nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte [Cambio en la disponibilidad del paquete de procesamiento de datos](#).

Puede usar AWS IoT SiteWise Edge para recopilar y almacenar datos temporalmente para poder organizar y procesar los datos del dispositivo de forma local. Al habilitar el procesamiento perimetral, puede optar por enviar solo datos agregados Nube de AWS para optimizar el uso del ancho de

banda y los costos de almacenamiento en la nube. Al utilizar AWS IoT SiteWise componentes con AWS IoT Greengrass, puede recopilar y procesar datos en la periferia antes de enviarlos a ella Nube de AWS, o bien gestionarlos de forma local mediante SiteWise Edge APIs.

La recopilación de datos se realiza mediante paquetes de datos y AWS IoT SiteWise componentes que se ejecutan en AWS IoT Greengrass ellos.

Note

- AWS IoT SiteWise conserva sus datos perimetrales en sus gateways SiteWise Edge durante un máximo de 30 días. El periodo de retención de los datos depende del espacio de disco disponible en el dispositivo.
- Si su puerta de enlace SiteWise Edge ha estado desconectada de ella Nube de AWS durante 30 días, el [paquete de procesamiento de datos](#) se desactiva automáticamente.

Temas

- [Configure la capacidad SiteWise perimetral en Edge](#)

Configure la capacidad SiteWise perimetral en Edge

AWS IoT SiteWise proporciona los siguientes paquetes que su puerta de enlace SiteWise Edge puede usar para determinar cómo recopilar y procesar sus datos. Seleccione paquetes para habilitar las capacidades perimetrales de su puerta de enlace SiteWise perimetral.

- El paquete de recopilación de datos permite a su puerta de enlace SiteWise Edge recopilar datos de varios servidores OPC UA y luego exportarlos desde el borde a la AWS nube. Se activa una vez que haya agregado fuentes de datos a su puerta de enlace SiteWise Edge.
- El paquete de procesamiento de datos permite que su pasarela SiteWise Edge procese los datos de su equipo en la periferia. Por ejemplo, puede utilizar modelos de activos para calcular métricas y transformaciones. Para obtener más información sobre modelos de activos y los activos, consulte [Modelado de activos industriales](#).

Note

- El paquete de procesamiento de datos solo está disponible en plataformas x86.

Configuración de capacidades de periferia

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, seleccione Puertas de enlace Edge.
3. Seleccione la puerta de enlace SiteWise Edge para la que desee activar las funciones perimetrales.
4. En la sección Capacidades de periferia, elija Editar.
5. En la sección Capacidades de periferia, seleccione Paquete de procesamiento de datos (se aplican cargos adicionales).
6. (Opcional) En la sección de conexión LDAP de Edge, puede conceder a los grupos de usuarios de su directorio corporativo acceso a esta puerta de enlace de SiteWise Edge. Los grupos de usuarios pueden usar las credenciales del Protocolo ligero de acceso a directorios (LDAP) para acceder a la puerta de enlace Edge. SiteWise Luego, pueden usarlas AWS OpsHub para la AWS IoT SiteWise aplicación, las operaciones de la AWS IoT SiteWise API u otras herramientas para administrar la puerta de enlace SiteWise Edge. Para obtener más información, consulte [Administre las puertas de enlace SiteWise Edge](#).

Note

También puede usar Linux o las Microsoft Windows credenciales para acceder a la puerta de enlace SiteWise Edge. Para obtener más información, consulte [Acceda a su puerta de enlace SiteWise Edge con las credenciales del sistema operativo Linux](#).

- a. Seleccione Activado.
- b. En Nombre del proveedor, introduzca un nombre para su proveedor de LDAP.
- c. En Nombre de host o dirección IP, introduzca el nombre de host o la dirección IP de su servidor LDAP.
- d. En Puerto, introduzca un número de puerto.
- e. En Nombre distinguido (DN) de base, introduzca un nombre distinguido (DN) para la base.


Se admiten los siguientes tipos de atributos: CommonName (CN), LocalityName (L), Name (ST), stateOrProvince OrganizationName (O), (OU), CountryName (C), organizationalUnitName StreetAddress (STREET), DomainComponent (DC) e ID de usuario (UID).

- f. En DN de grupo de administradores, introduzca un DN.
 - g. En DN de grupo de usuarios, introduzca un DN.
7. Seleccione Save.

Ahora que ha activado las funciones perimetrales en su puerta de enlace perimetral, debe configurar su modelo de activos para la SiteWise periferia. La configuración de periferia de su modelo de activos especifica dónde se calculan las propiedades de los activos. Puede calcular todas las propiedades en la periferia o puede configurar las propiedades de su modelo de activos por separado. Las propiedades del modelo de activos incluyen [métricas](#), [transformaciones](#) y [mediciones](#).


Para obtener más información sobre las propiedades de activos, consulte [the section called “Definición de propiedades de datos”](#).

Después de crear el modelo de activo, puede configurarlo para la periferia. Para obtener más información sobre cómo configurar su modelo de activos para la periferia, consulte [the section called “Creación de un modelo de activos \(consola\)”](#).

 Note

Los modelos de activos y los paneles de control se sincronizan automáticamente entre la AWS nube y su puerta de enlace SiteWise Edge cada 10 minutos. También puedes sincronizarlos manualmente desde la aplicación SiteWise Edge Gateway local.

Configure el procesamiento de datos AWS IoT SiteWise perimetrales para modelos y activos

 Note

La característica de paquete de procesamiento de datos (DPP) ya no está disponible para los nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte [Cambio en la disponibilidad del paquete de procesamiento de datos](#).

Puede usar AWS IoT SiteWise Edge para recopilar, almacenar, organizar y monitorear los datos del equipo a nivel local. Puede usar SiteWise Edge para modelar sus datos industriales y visualizarlos

localmente. Puede procesar sus datos de forma local y enviarlos a la AWS nube, o bien procesarlos de forma local mediante la AWS IoT SiteWise API.

Con AWS IoT SiteWise Edge, puede procesar los datos sin procesar de forma local y elegir enviar solo los datos agregados a la AWS nube para optimizar el uso del ancho de banda y los costos de almacenamiento en la nube.

Note

- AWS IoT SiteWise conserva los datos perimetrales en sus gateways SiteWise Edge durante un máximo de 30 días. El periodo de retención de los datos depende del espacio de disco disponible en el dispositivo.
- Si su puerta de enlace SiteWise Edge ha estado desconectada de la AWS nube durante 30 días, [Configure una fuente OPC UA en Edge SiteWise](#) se deshabilita automáticamente.

Configure un modelo de activos para el procesamiento de datos en SiteWise Edge

Debe configurar su modelo de activos para la periferia antes de poder procesar los datos de la puerta de enlace SiteWise perimetral en la periferia. La configuración de periferia de su modelo de activos especifica dónde se calculan las propiedades de los activos. Puede elegir entre calcular todas las propiedades en la periferia y enviar los resultados a la Nube de AWS o personalizar el lugar donde se calculará cada propiedad de los activos por separado. Para obtener más información, consulte [Configure el procesamiento de datos AWS IoT SiteWise perimetrales para modelos y activos](#).


Las propiedades de los activos incluyen métricas, transformaciones y mediciones:

- Las métricas son los datos agregados del activo en un periodo de tiempo determinado. Puede calcular nuevas métricas utilizando los datos de métricas existentes. AWS IoT SiteWise siempre envía tus métricas a la AWS nube para almacenarlas a largo plazo. AWS IoT SiteWise calcula las métricas en la AWS nube de forma predeterminada. Puede configurar su modelo de activos para calcular sus métricas en la periferia. AWS IoT SiteWise envía los resultados procesados a la AWS nube.
- Las transformaciones son expresiones matemáticas que asignan puntos de datos de la propiedad de un activo de un formulario a otro. Las transformaciones pueden utilizar métricas como datos de entrada y deben computarse y almacenarse en la misma ubicación que sus entradas. Si configura una entrada de métrica para que se calcule en la periferia, AWS IoT SiteWise también calcula su transformación asociada en la periferia.

- Las mediciones se formatean como datos sin procesar que su dispositivo recopila y envía a la AWS nube de forma predeterminada. Puede configurar su modelo de activos para almacenar estos datos en su dispositivo local.

Para obtener más información sobre las propiedades de activos, consulte [the section called “Definición de propiedades de datos”](#).

Después de crear el modelo de activo, puede configurarlo para la periferia. Para obtener más información sobre cómo configurar su modelo de activos para la periferia, consulte [the section called “Creación de un modelo de activos \(consola\)”](#).

 Note

Los modelos de activos y los paneles de control se sincronizan automáticamente entre la AWS nube y su puerta de enlace SiteWise Edge cada 10 minutos. También puede sincronizarlos manualmente desde [Administre las puertas de enlace SiteWise Edge](#).

Puede usar el AWS IoT SiteWise REST APIs y el AWS Command Line Interface (AWS CLI) para consultar los datos de la periferia en su puerta de enlace SiteWise Edge. Antes de consultar los datos de la SiteWise periferia en la puerta de enlace de Edge, debe cumplir los siguientes requisitos previos:

- Sus credenciales deben estar configuradas para el REST APIs. Para obtener más información acerca de cómo configurar las credenciales, consulte [the section called “Administración de puertas de enlace”](#).
- El punto final del SDK debe apuntar a la dirección IP de su puerta de enlace SiteWise Edge. Puede encontrar más información en la documentación de su SDK. Por ejemplo, consulte [Especificar puntos de conexión personalizados](#) en la Guía para desarrolladores de AWS SDK for Java 2.x .
- Su certificado de puerta de enlace SiteWise Edge debe estar registrado. Puedes encontrar más información sobre cómo registrar tu certificado de puerta de enlace SiteWise Edge en la documentación de tu SDK. Por ejemplo, consulte el [Registro de paquetes de certificados en Node.js](#) en la Guía para desarrolladores de AWS SDK for Java 2.x .

Para obtener más información sobre cómo consultar datos con AWS IoT SiteWise, consulte [Consulta datos de AWS IoT SiteWise](#).

Agregue fuentes de datos a su puerta de enlace AWS IoT SiteWise Edge

Después de configurar una puerta de enlace AWS IoT SiteWise Edge, puede agregar y configurar fuentes de datos para ingerir datos desde equipos industriales locales. AWS IoT SiteWise SiteWise Edge admite varios protocolos, incluido el OPC UA, y muchos otros protocolos disponibles a través de fuentes de datos asociadas. Estas fuentes permiten que su puerta de enlace se conecte con los servidores locales y recupere sus datos industriales. Al configurar las fuentes de datos, puede ingerir datos de diversas fuentes de datos y, a continuación, asociar los flujos de datos con las propiedades de los activos, lo que permite modelar y mapear los datos de forma integral de los activos industriales. AWS IoT SiteWise

Temas

- [Fuentes de datos OPC UA para AWS IoT SiteWise pasarelas Edge](#)
- [Fuentes de datos asociadas en las puertas de enlace SiteWise Edge](#)

Fuentes de datos OPC UA para AWS IoT SiteWise pasarelas Edge

Después de configurar una puerta de enlace AWS IoT SiteWise Edge, puede configurar las fuentes de datos para que la puerta de enlace SiteWise Edge pueda ingerir datos de equipos industriales locales a AWS IoT SiteWise. Cada fuente representa un servidor local, como un servidor OPC UA, al que la puerta de enlace SiteWise Edge conecta y recupera los flujos de datos industriales. Para obtener más información sobre la configuración de una puerta de enlace SiteWise Edge, consulte.

[Cree una puerta de enlace SiteWise Edge autohospedada](#)

El tipo de puerta de enlace, las puertas de enlace V3 habilitadas para MQTT frente a las puertas de enlace V2 de transmisión clásica, influye en la forma en que se gestionan los datos del OPC UA. En Classic Stream, las puertas de enlace V2 y las fuentes de datos OPC UA se agregan directamente a la configuración del SiteWise editor de IoT de la puerta de enlace. Cada fuente de datos está acoplada a la puerta de enlace y el enrutamiento de datos se configura individualmente para cada fuente. Por el contrario, al utilizar pasarelas V3 habilitadas para MQTT, las fuentes de datos OPC UA se convierten en temas MQTT y se gestionan a través de destinos centralizados. Para obtener más información sobre cada tipo, consulte y. [Puertas de enlace V3 habilitadas para MQTT para Edge AWS IoT SiteWise Transmisiones clásicas, pasarelas V2 para Edge AWS IoT SiteWise](#)

Note

AWS IoT SiteWise reinicia la puerta de enlace SiteWise Edge cada vez que agrega o edita una fuente. La puerta de enlace SiteWise Edge no ingiere datos mientras actualiza la configuración de la fuente. El tiempo necesario para reiniciar la puerta de enlace SiteWise Edge depende de la cantidad de etiquetas que haya en las fuentes de la puerta de enlace SiteWise Edge. El tiempo de reinicio puede oscilar entre unos segundos (para una puerta de enlace SiteWise Edge con pocas etiquetas) y varios minutos (para una puerta de enlace SiteWise Edge con muchas etiquetas).

Después de crear los orígenes, puede asociar sus flujos de datos a las propiedades de los activos. Para obtener más información acerca de cómo crear y utilizar activos, consulte [Modelado de activos industriales](#).

Puede ver CloudWatch las métricas para comprobar a qué fuente de datos está conectada AWS IoT SiteWise. Para obtener más información, consulte [AWS IoT Greengrass Version 2 métricas de puerta de enlace](#).

Actualmente, AWS IoT SiteWise es compatible con los siguientes protocolos de fuente de datos:

- [OPC UA](#): un protocolo de comunicación machine-to-machine (M2M) para la automatización industrial.

Support para protocolos industriales adicionales

SiteWise Edge admite una amplia gama de protocolos industriales mediante la integración con los socios de fuentes de datos. Estas asociaciones permiten la conectividad con más de 200 protocolos diferentes, que se adaptan a diversos sistemas y dispositivos industriales.

Para obtener una lista de los socios de fuentes de datos disponibles, consulte [SiteWise Opciones de fuentes de datos de los socios de Edge Gateway](#).

Configure una fuente OPC UA en Edge SiteWise

Puede usar la AWS IoT SiteWise consola o una función de puerta de enlace SiteWise Edge para definir y agregar una fuente OPC UA a su puerta de enlace SiteWise Edge para representar un servidor OPC UA local.

Temas

- [Configuración de un origen OPC UA \(consola\)](#)
- [Configure una fuente OPC UA \(AWS CLI\)](#)

Configuración de un origen OPC UA (consola)

Puede utilizar la consola para configurar el origen OPC UA mediante el siguiente procedimiento.

Note

Advertencia: si se duplica, se TQVs puede duplicar la carga.

Para configurar una fuente OPC UA mediante la consola AWS IoT SiteWise

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación de la izquierda, selecciona las puertas de enlace Edge en la sección Edge.
3. Seleccione la puerta de enlace SiteWise Edge para añadir una fuente OPC UA.
4. Seleccione Añadir origen de datos.
5. Introduzca un nombre para el origen.
6. Escriba el Punto de conexión local del servidor de origen de datos. El punto de conexión puede ser la dirección IP o el nombre de host. También puede añadir un número de puerto al punto de conexión local. Por ejemplo, su punto de conexión local podría tener el siguiente aspecto:
opc.tcp://203.0.113.0:49320
7. (Opcional) En ID de nodo de selección, añada filtros de nodo para limitar los flujos de datos que se ingieren en la nube de AWS . De forma predeterminada, las puertas de enlace SiteWise Edge utilizan el nodo raíz de un servidor para ingerir todos los flujos de datos. Puede usar filtros de nodos para reducir el tiempo de inicio de la puerta de enlace SiteWise Edge y el uso de la CPU al incluir únicamente las rutas a los datos que usted modele. AWS IoT SiteWise De forma predeterminada, las pasarelas SiteWise Edge cargan todas las rutas OPC UA excepto las que comienzan por. /Server/ Para definir filtros de nodo OPC UA, puede utilizar rutas de nodo y los caracteres comodín * y **. Para obtener más información, consulte [Utilice filtros de nodo OPC UA en Edge SiteWise](#).
8. Los destinos varían entre las pasarelas V3 habilitadas para MQTT y las pasarelas V2 para transmisiones clásicas.

- En las transmisiones clásicas, los destinos de las pasarelas V2 tienen una relación 1:1 con la fuente. Cada fuente envía datos a un destino concreto.
- Los destinos de puerta de enlace V3 habilitados para MQTT se configuran por separado, ya que el modelo de concentrador y radio permite centralizar la configuración y la administración de múltiples fuentes de datos en diferentes puertas de enlace. Para configurar los destinos en una puerta de enlace V3, consulte. [Comprenda los destinos de AWS IoT SiteWise Edge](#)

Classic streams, V2 gateway destinations

- AWS IoT SiteWise en tiempo real: elija esta opción para enviar los datos directamente al AWS IoT SiteWise almacenamiento. Ingera y supervise los datos en tiempo real en la periferia.
- AWS IoT SiteWise Almacenados en búfer mediante Amazon S3: envíe datos en formato Parquet a Amazon S3 y, a continuación, impórtelos al AWS IoT SiteWise almacenamiento. Elija esta opción para ingerir datos en lotes y almacenar los datos históricos de forma rentable. Puede configurar la ubicación del bucket de Amazon S3 que prefiera y la frecuencia con la que desea que se carguen los datos en Amazon S3. También puede elegir qué hacer con los datos después de su ingesta en AWS IoT SiteWise. Puede elegir que los datos estén disponibles AWS IoT SiteWise tanto en Amazon S3 como en Amazon S3 o puede optar por eliminarlos automáticamente de Amazon S3 después de importarlos AWS IoT SiteWise.
 - El bucket de Amazon S3 es un mecanismo de almacenamiento y almacenamiento en búfer y admite archivos en formato Parquet.
 - Si selecciona la casilla Importar datos al AWS IoT SiteWise almacenamiento, los datos se cargan primero en Amazon S3 y, después, en el AWS IoT SiteWise almacenamiento.
 - Si selecciona la casilla Eliminar datos de Amazon S3, los datos se eliminarán de Amazon S3 después de importarlos al SiteWise almacenamiento.
 - Si desactiva la casilla Eliminar datos de Amazon S3, los datos se almacenan tanto en Amazon S3 como en SiteWise almacenamiento.
 - Si desactiva la casilla Importación de datos al almacenamiento de AWS IoT SiteWise, los datos solo se almacenarán en Amazon S3. No se importan al SiteWise almacenamiento.

Visite [Administración del almacenamiento de datos](#) para obtener más información sobre las distintas opciones de almacenamiento que ofrece AWS IoT SiteWise . Para obtener más información sobre precios, consulte [Precios de AWS IoT SiteWise](#).

- AWS IoT Greengrass administrador de transmisiones: utilice el administrador de AWS IoT Greengrass transmisiones para enviar datos a los siguientes destinos en la AWS nube: canales AWS IoT Analytics entrantes, transmisiones en Amazon Kinesis Data Streams, propiedades de activos u objetos AWS IoT SiteWise en Amazon Simple Storage Service (Amazon S3). Para obtener más información, consulte [Administrar flujos de datos en la Guía AWS IoT Greengrass básica](#) para AWS IoT Greengrass Version 2 desarrolladores.

Introduzca un nombre para la AWS IoT Greengrass transmisión.


MQTT-enabled, V3 gateway destinations

1. Consulta [Puertas de enlace V3 habilitadas para MQTT para Edge AWS IoT SiteWise](#) para obtener información sobre cómo añadir tus destinos relevantes.
 2. Vuelva a este procedimiento después de añadir sus destinos de origen.
9. En el panel Configuración avanzada puede realizar lo siguiente:
- a. Elija un modo de seguridad de mensajes para las conexiones y los datos en tránsito entre el servidor de origen y la puerta de enlace SiteWise Edge. Este campo es la combinación de la política de seguridad OPC UA y el modo de seguridad de mensajes. Elija la misma política de seguridad y el mismo modo de seguridad de mensajes que especificó para su servidor OPC UA.
 - b. Si su origen requiere autenticación, elija un secreto de AWS Secrets Manager en la lista Configuración de autenticación. La puerta de enlace SiteWise Edge utiliza las credenciales de autenticación que figuran en este secreto cuando se conecta a esta fuente de datos. Debe adjuntar los secretos al AWS IoT Greengrass componente de la puerta de enlace SiteWise Edge para utilizarlos en la autenticación de la fuente de datos. Para obtener más información, consulte [the section called “Configure la autenticación de la fuente de datos”](#).

 Tip

Es posible que el servidor de datos tenga una opción denominada Permitir inicio de sesión anónimo. Si esta opción es Sí, entonces el origen no requerirá autenticación.

- c. (Opcional) Puede activar un prefijo de flujo de datos seleccionando Activar prefijo de flujo de datos: opcional.
 - Introduzca un Prefijo de flujo de datos. La puerta de enlace SiteWise Edge agrega este prefijo a todos los flujos de datos de esta fuente. Utilice un prefijo de flujo de datos para distinguir entre flujos de datos que tienen el mismo nombre y orígenes diferentes. Cada flujo de datos debe tener un nombre único en su cuenta.
- d. (Opcional) Elija una opción Conversión de tipos de datos para convertir los tipos de datos OPC UA no compatibles en cadenas antes de ingerirlos en AWS IoT SiteWise. Convierta valores de matriz con tipos de datos simples en cadenas JSON y tipos de DateTime datos en cadenas ISO 8601. Para obtener más información, consulte [Conversión de tipos de datos no admitidos](#).
- e. Elija un activador de cambio de datos predeterminado para los nodos que no estén incluidos en un grupo de propiedades definido por el usuario. El activador de cambio de datos predeterminado determina cuándo el servidor OPC UA envía los valores actualizados a la puerta de enlace. Puede elegir una de las siguientes opciones:
 - Estado: para recibir datos solo cuando cambia un estado.
 - StatusValue— recibir datos cuando cambia un estado o un valor.
 - StatusValueTimestamp— recibir datos cuando cambia un estado, un valor o una marca de tiempo.
- f. (Opcional) En una puerta de enlace V3 habilitada para MQTT, puede usar la configuración Discovery para configurar el proceso de descubrimiento de nodos OPC UA. La configuración de Discovery reemplaza el sistema de archivos de anulación de la configuración anterior para estas opciones por ajustes basados en la consola que se actualizan dinámicamente sin necesidad de reiniciar la puerta de enlace.

 Note

El activador de cambio de datos predeterminado requiere la versión 3.1.0 o posterior del componente recopilador IoT SiteWise OPC UA. Para obtener más información, consulte [Actualice la versión de un componente AWS IoT SiteWise](#).

- i. Para obtener el número máximo de solicitudes de navegación simultáneas, introduzca el número máximo de solicitudes de navegación que el servidor OPC UA puede


- gestionar simultáneamente. Puede configurar hasta 500 solicitudes de navegación simultáneas por fuente de datos.
- ii. Para obtener el número máximo de nodos por solicitud de navegación, introduzca el número máximo de nodos que se van a enviar en cada solicitud de navegación al servidor OPC UA. Puede enviar hasta 1000 nodos por solicitud de navegación.
 - iii. Seleccione Evitar los bucles del árbol de nodos para evitar que la puerta de enlace se quede atascada en referencias circulares al navegar por la estructura del servidor OPC UA. Cuando está seleccionada, la puerta de enlace rastrea las ubicaciones visitadas para evitar los bucles infinitos que pueden producirse cuando los nodos del servidor se refieren entre sí siguiendo un patrón circular.
 - iv. Seleccione Habilitar el recorrido de nodos para permitir que la puerta de enlace explore la estructura completa de su servidor OPC UA y descubra todos los puntos de datos disponibles en sus equipos y dispositivos. Cuando se selecciona, la pasarela navega por la organización de datos del equipo más allá del nivel raíz para encontrar automáticamente todos los sensores, controles y puntos de medición.
 - v. Seleccione Activar la detección periódica para ejecutar automáticamente las operaciones de detección a intervalos regulares a fin de detectar cambios en la estructura del servidor OPC UA. Cuando se selecciona, la puerta de enlace monitorea continuamente los equipos o puntos de datos recién agregados, lo que garantiza que se detecten automáticamente y estén disponibles para la recopilación de datos.
 - A. En Intervalo de detección periódica, establezca el intervalo de tiempo entre las operaciones de detección automática cuando se ejecute la detección periódica. El intervalo mínimo de detección periódica es de 30 segundos y el máximo de 30 días.
 - B. En el número máximo de nodos descubiertos por intervalo, establezca el número máximo de nodos que se deben descubrir por intervalo de descubrimiento. Esto ayuda a controlar la carga tanto en la puerta de enlace como en el servidor OPC UA durante las operaciones de descubrimiento.
- g. (Opcional) En Grupos de propiedades, seleccione Añadir nuevo grupo.
- i. Introduzca un Nombre para el grupo de propiedades.
 - ii. En Propiedades:

1. (Opcional) En Rutas de nodo, añada filtros de nodo OPC UA para limitar las rutas OPC UA que se cargan en AWS IoT SiteWise. El formato es similar a ID de nodo para su selección.
- iii. En Configuración, realice lo siguiente:
1. En la configuración de calidad de los datos, elija el tipo de calidad de datos que desea que AWS IoT SiteWise incorpore Collector.
 2. En Configuración del modo de análisis, configure las propiedades de la suscripción estándar mediante Modo de análisis. Puede seleccionar Suscribirse o Sondeo. Para obtener más información sobre el modo de escaneo, consulte [the section called “Filtrado de rangos de ingesta de datos”](#).

Subscribe

Envío de todos los puntos de datos

- i. Seleccione Suscribirse y configure lo siguiente:
 - A. [Desencadenador de cambio de datos](#): condición que inicia una alerta de cambio de datos.
 - B. [Tamaño de la cola de suscripciones](#): profundidad de la cola en un servidor OPC UA para una métrica concreta en la que se ponen en cola las notificaciones de los elementos monitorizados.
 - C. [Intervalo de publicación de suscripciones](#): intervalo (en milisegundos) del ciclo de publicación especificado al crearse la suscripción.
 - D. Intervalo de instantáneas: opcional: se ajusta el tiempo de espera de las instantáneas para garantizar que AWS IoT SiteWise Edge ingiera un flujo constante de datos.
 - E. Velocidad de escaneo: la velocidad a la que desea que la puerta de enlace SiteWise Edge lea sus registros. AWS IoT SiteWise calcula automáticamente la velocidad de escaneo mínima permitida para su puerta de enlace SiteWise Edge.
 - F. Marca de tiempo: marca de tiempo que se debe incluir en los puntos de datos OPC UA. Puede utilizar la marca de tiempo del servidor o de su dispositivo.

 Note

Utilice la versión 2.5.0 o posterior del componente recopilador IoT SiteWise OPC UA. Si utiliza la función de marca de tiempo con versiones anteriores, las actualizaciones de configuración fallarán. Para obtener más información, consulte [Actualice la versión de un componente AWS IoT SiteWise](#).


- ii. En Configuración de la zona neutra, configure un Tipo de zona neutra. El tipo de banda inactiva controla los datos que la fuente le envía y los datos que descarta. AWS IoT SiteWise Para obtener más información sobre configuración de la banda muerta, consulte [the section called “Filtrado de rangos de ingesta de datos”](#).
 - Ninguno: el servidor asociado envía todos los puntos de datos de este grupo de propiedades.
 - Porcentaje: el servidor asociado solo envía datos que se encuentran fuera de un porcentaje específico del rango de datos. El servidor calcula este rango en función de la unidad de ingeniería mínima y máxima definida para cada nodo. Si el servidor no admite zonas neutras de porcentaje o carece de unidades de ingeniería definidas, la puerta de enlace calcula el rango utilizando los valores mínimo y máximo que se indican a continuación.
 - Absoluto: el servidor asociado solo envía datos que se encuentran fuera de un rango específico.
 - A. Establezca el Valor de la zona neutra como el porcentaje del rango de datos con respecto a la zona neutra.
 - B. (Opcional) Especifique un mínimo y un máximo para el rango de zona neutra utilizando Rango mínimo: opcional y Rango máximo: opcional.

Poll


Envío de puntos de datos en un intervalo específico

- Seleccione Suscribirse y configure lo siguiente:

- A. Velocidad de escaneo: la velocidad a la que quieres que la puerta de enlace SiteWise Edge lea tus registros. AWS IoT SiteWise calcula automáticamente la velocidad de escaneo mínima permitida para su puerta de enlace SiteWise Edge.
- B. Marca de tiempo: marca de tiempo que se debe incluir en los puntos de datos OPC UA. Puede utilizar la marca de tiempo del servidor o de su dispositivo.

 Note

Utilice la versión 3.1.0 o posterior del componente recopilador IoT SiteWise OPC UA. Si utiliza la función de marca de tiempo con versiones anteriores, las actualizaciones de configuración fallarán. Para obtener más información, consulte [Actualice la versión de un componente AWS IoT SiteWise](#).

 Note

Configuración de la zona neutra se aplica cuando ha seleccionado Suscribirse en Configuración del modo de análisis.

10. Seleccione Save.

Configure una fuente OPC UA ()AWS CLI

Puede definir las fuentes de datos OPC UA para una puerta de enlace SiteWise Edge mediante AWS CLI. Para ello, cree un archivo JSON de configuración de capacidades OPC UA y utilice el [update-gateway-capability-configuration](#) comando para actualizar la configuración de la puerta de enlace SiteWise Edge. Debe definir todos los orígenes OPC UA en una única configuración de capacidad.

MQTT-enabled, V3 gateway

Esta capacidad tiene el siguiente espacio de nombres.

- `iotsitewise:opcuacollector:3`

```

{
  "sources": [
    {
      "name": "string",
      "endpoint": {
        "certificateTrust": {
          "type": "TrustAny" | "X509",
          "certificateBody": "string",
          "certificateChain": "string",
        },
        "endpointUri": "string",
        "securityPolicy": "NONE" | "BASIC128_RSA15" | "BASIC256" | "BASIC256_SHA256"
| "AES128_SHA256_RSAOAEP" | "AES256_SHA256_RSAPSS",
        "messageSecurityMode": "NONE" | "SIGN" | "SIGN_AND_ENCRYPT",
        "identityProvider": {
          "type": "Anonymous" | "Username",
          "usernameSecretArn": "string"
        },
        "nodeFilterRules": [
          {
            "action": "INCLUDE",
            "definition": {
              "type": "OpcUaRootPath",
              "rootPath": "string"
            }
          }
        ]
      },
      "measurementDataStreamPrefix": "string",
      "typeConversions": {
        "array": "JsonArray",
        "datetime": "ISO8601String"
      },
      "destination": {
        {
          "type": "MQTT"
        }
      },
      "defaultPropertyGroupConfig": {
        "dataChangeTrigger": "STATUS" | "STATUS_VALUE" | "STATUS_VALUE_TIMESTAMP"
      },
      "discoveryConfig": {
        "enableNodeTraversal": true | false,

```

```

    "avoidNodeTreeLoops": true | false,
    "maxConcurrentBrowseRequests": integer,
    "maxNodesPerBrowseRequest": integer,
    "periodicDiscovery": {
      "interval": "string",
      "maxNodesDiscoveredPerInterval": integer
    }
  },
  "propertyGroups": [
    {
      "name": "string",
      "nodeFilterRuleDefinitions": [
        {
          "type": "OpcUaRootPath",
          "rootPath": "string"
        }
      ],
      "deadband": {
        "type": "PERCENT" | "ABSOLUTE",
        "value": double,
        "eguMin": double,
        "eguMax": double,
        "timeoutMilliseconds": integer
      },
      "scanMode": {
        "type": "EXCEPTION" | "POLL",
        "rate": integer,
        "timestampToReturn": "SOURCE_TIME" | "SERVER_TIME"
      },
      "dataQuality": {
        "allowGoodQuality": true | false,
        "allowBadQuality": true | false,
        "allowUncertainQuality": true | false
      },
      "subscription": {
        "dataChangeTrigger": "STATUS" | "STATUS_VALUE" |
"STATUS_VALUE_TIMESTAMP",
        "queueSize": integer,
        "publishingIntervalMilliseconds": integer,
        "snapshotFrequencyMilliseconds": integer
      }
    }
  ]
}

```

```

]
}

```

Classic streams, V2 gateway

Esta capacidad tiene el siguiente espacio de nombres.

- `iotsitewise:opcuacollector:2`

Sintaxis de la solicitud

```

{
  "sources": [
    {
      "name": "string",
      "endpoint": {
        "certificateTrust": {
          "type": "TrustAny" | "X509",
          "certificateBody": "string",
          "certificateChain": "string",
        },
        "endpointUri": "string",
        "securityPolicy": "NONE" | "BASIC128_RSA15" | "BASIC256" | "BASIC256_SHA256"
| "AES128_SHA256_RSAOAEP" | "AES256_SHA256_RSAPSS",
        "messageSecurityMode": "NONE" | "SIGN" | "SIGN_AND_ENCRYPT",
        "identityProvider": {
          "type": "Anonymous" | "Username",
          "usernameSecretArn": "string"
        },
        "nodeFilterRules": [
          {
            "action": "INCLUDE",
            "definition": {
              "type": "OpcUaRootPath",
              "rootPath": "string"
            }
          }
        ]
      },
      "measurementDataStreamPrefix": "string",
      "typeConversions": {
        "array": "JsonArray",
        "datetime": "ISO8601String"
      }
    }
  ]
}

```

```

    },
    "destination": {
      "type": "StreamManager",
      "streamName": "string",
      "streamBufferSize": integer,
    },
    "propertyGroups": [
      {
        "name": "string",
        "nodeFilterRuleDefinitions": [
          {
            "type": "OpcUaRootPath",
            "rootPath": "string"
          }
        ],
        "deadband": {
          "type": "PERCENT" | "ABSOLUTE",
          "value": double,
          "eguMin": double,
          "eguMax": double,
          "timeoutMilliseconds": integer
        },
        "scanMode": {
          "type": "EXCEPTION" | "POLL",
          "rate": integer,
          "timestampToReturn": "SOURCE_TIME" | "SERVER_TIME"
        },
        "dataQuality": {
          "allowGoodQuality": true | false,
          "allowBadQuality": true | false,
          "allowUncertainQuality": true | false
        },
        "subscription": {
          "dataChangeTrigger": "STATUS" | "STATUS_VALUE" |
"STATUS_VALUE_TIMESTAMP",
          "queueSize": integer,
          "publishingIntervalMilliseconds": integer,
          "snapshotFrequencyMilliseconds": integer
        }
      }
    ]
  }
]

```

```
}
```

Cuerpo de la solicitud

`sources`

Una lista de estructuras de definición de origen OPC UA que contienen la siguiente información:

`name`

Un nombre único y sencillo para el origen.

`endpoint`

Una estructura de punto de conexión que contiene la siguiente información:

`certificateTrust`

Una estructura de política de confianza de certificados que contiene la siguiente información:

`type`

El modo de confianza del certificado para el origen. Seleccione una de las siguientes opciones:

- `TrustAny`— La puerta de enlace SiteWise Edge confía en cualquier certificado cuando se conecta a la fuente OPC UA.
- `X509`— La puerta de enlace SiteWise Edge confía en un certificado X.509 cuando se conecta a la fuente OPC UA. Si elige esta opción, debe definir `certificateBody` en `certificateTrust`. También puede definir `certificateChain` en `certificateTrust`.

`certificateBody`

(Opcional) Cuerpo de un certificado X.509.

Este campo es obligatorio si elige `X509` para `type` en `certificateTrust`.

`certificateChain`

(Opcional) Cadena de confianza para un certificado X.509.

Este campo solo se utiliza si elige `X509` para `type` en `certificateTrust`.

endpointUri

Punto de conexión local del origen OPC UA. Por ejemplo, su punto de conexión local podría tener el siguiente aspecto: `opc.tcp://203.0.113.0:49320`.

securityPolicy

Política de seguridad que se debe utilizar para proteger los mensajes que se leen desde el origen OPC UA. Seleccione una de las siguientes opciones:

- **NONE**— La puerta de enlace SiteWise Edge no protege los mensajes de la fuente OPC UA. Le recomendamos que elija una política de seguridad diferente. Si elige esta opción, también debe elegir **NONE** para `messageSecurityMode`.
- **BASIC256_SHA256**: la política de seguridad `Basic256Sha256`.
- **AES128_SHA256_RSA0AEP**: la política de seguridad `Aes128_Sha256_Rsa0aep`.
- **AES256_SHA256_RSAPSS**: la política de seguridad `Aes256_Sha256_RsaPss`.
- **BASIC128_RSA15**: (obsoleta) la política de seguridad `Basic128Rsa15` está en desuso en la especificación OPC UA porque ya no se considera segura. Le recomendamos que elija una política de seguridad diferente. Para obtener más información, consulte [Profile SecurityPolicy — Basic128Rsa15](#).
- **BASIC256**: (obsoleta) la política de seguridad `Basic256` está en desuso en la especificación OPC UA porque ya no se considera segura. Le recomendamos que elija una política de seguridad diferente. [Para obtener más información, consulte — Basic256. SecurityPolicy](#)

Important

Si elige una política de seguridad distinta a **NONE**, debe elegir **SIGN** o **SIGN_AND_ENCRYPT** para `messageSecurityMode`. También debe configurar el servidor de origen para que confíe en la puerta de enlace SiteWise Edge. Para obtener más información, consulte [Configure los servidores OPC UA para que confíen en la puerta de enlace Edge AWS IoT SiteWise](#).

messageSecurityMode

Modo de seguridad de mensajes que se va a utilizar para proteger las conexiones con el origen OPC UA. Seleccione una de las siguientes opciones:

- NONE— La puerta de enlace SiteWise Edge no protege las conexiones a la fuente OPC UA. Le recomendamos que elija un modo de seguridad de mensajes diferente. Si elige esta opción, también debe elegir NONE para `securityPolicy`.
- SIGN— Los datos en tránsito entre la puerta de enlace SiteWise Edge y la fuente OPC UA están firmados pero no cifrados.
- SIGN_AND_ENCRYPT: los datos en tránsito entre la puerta de enlace y el origen OPC UA se firman y cifran.

 Important

Si elige un modo de seguridad de mensajes que no sea NONE, debe elegir una `securityPolicy` distinta de NONE. También debe configurar el servidor de origen para que confíe en la puerta de enlace SiteWise Edge. Para obtener más información, consulte [Configure los servidores OPC UA para que confíen en la puerta de enlace Edge AWS IoT SiteWise](#).

`identityProvider`

Estructura de proveedor de identidades que contiene la siguiente información:

`type`

Tipo de credenciales de autenticación requeridas por el origen. Seleccione una de las siguientes opciones:

- Anonymous: el origen no requiere autenticación para conectarse.
- Username: el origen requiere un nombre de usuario y una contraseña para conectarse. Si elige esta opción, debe definir `usernameSecretArn` en `identityProvider`.

`usernameSecretArn`

(Opcional) El ARN de un AWS Secrets Manager secreto. La puerta de enlace SiteWise Edge usa las credenciales de autenticación de este secreto cuando se conecta a esta fuente. Debe adjuntar los secretos al SiteWise conector IoT de su puerta de enlace SiteWise Edge para usarlos en la autenticación de origen. Para obtener más información, consulte [Configurar la autenticación de fuentes de datos para SiteWise Edge](#).

Este campo es obligatorio si elige Username para type en identityProvider.

nodeFilterRules

Una lista de estructuras de reglas de filtrado de nodos que definen las rutas de flujo de datos OPC UA que se van a enviar a la AWS nube. Puedes usar filtros de nodos para reducir el tiempo de inicio de la puerta de enlace SiteWise Edge y el uso de la CPU al incluir únicamente las rutas a los datos que modelos. AWS IoT SiteWise De forma predeterminada, las pasarelas SiteWise Edge cargan todas las rutas OPC UA excepto las que comienzan por. /Server/ Para definir filtros de nodo OPC UA, puede utilizar rutas de nodo y los caracteres comodín * y **. Para obtener más información, consulte [Utilice filtros de nodo OPC UA en Edge SiteWise](#).

Cada estructura de la lista debe contener la siguiente información:

action

Es la acción de esta regla de filtro de nodo. Puede elegir la siguiente opción:

- INCLUDE— La puerta de enlace SiteWise Edge incluye solo los flujos de datos que cumplen esta regla.

definition

Estructura de reglas de filtros de nodo que contiene la siguiente información:

type

Es el tipo de ruta de filtro de nodo para esta regla. Puede elegir la siguiente opción:

- OpcUaRootPath— La puerta de enlace SiteWise Edge evalúa esta ruta de filtro de nodos comparándola con la raíz de la jerarquía de rutas OPC UA.

rootPath

Ruta de filtro de nodo que se va a evaluar con la raíz de la jerarquía de rutas de OPC UA. Esta ruta debe comenzar con /.

measurementDataStreamPrefix

Cadena que se debe anteponer a todos los flujos de datos del origen. La puerta de enlace SiteWise Edge agrega este prefijo a todos los flujos de datos de esta fuente. Utilice un prefijo de flujo de datos para distinguir entre flujos de datos que tienen el mismo nombre y orígenes diferentes. Cada flujo de datos debe tener un nombre único en su cuenta.

typeConversions

Tipos de conversiones disponibles para tipos de datos OPC UA no compatibles. Cada tipo de datos se convierte en cadenas. Para obtener más información, consulte [Conversión de tipos de datos no admitidos](#).

array

Tipo de datos de matriz simple que se convierte en cadenas. Puede elegir la siguiente opción:

- `JsonArray`: indica que ha elegido convertir los tipos de datos de matriz simple en cadenas.

datetime

El tipo de DateTime datos que se convierte en cadenas. Puede elegir la siguiente opción:

- `ISO8601String`: indica que ha elegido convertir los tipos de datos ISO 8601 en cadenas.

destination

Configuración para el destino de las etiquetas OPC UA. Las pasarelas V3 de transmisión clásica, las versiones 2 y las habilitadas para MQTT tienen diferentes configuraciones para los destinos.

type

El tipo del destino.

`streamName`— solo para transmisiones clásicas y pasarelas V2

El nombre del flujo de . El nombre del flujo debe ser único.

`streamBufferSize`— solo para transmisiones clásicas y pasarelas V2

Tamaño de búfer del flujo. Es importante para administrar el flujo de datos de orígenes OPC UA.

`defaultPropertyGroupConfig`— Compatible con MQTT, solo pasarelas V3

(Opcional) Configuración para el grupo de propiedades predeterminado. El grupo de propiedades predeterminado contiene todos los nodos que de otro modo no estarían incluidos en un grupo de propiedades definido por el usuario.

`dataChangeTrigger`

El activador de cambios de datos predeterminado que se utilizará en el grupo de propiedades predeterminado. Los valores válidos son `STATUS_VALUE_TIMESTAMP`, `STATUS_VALUE` o `STATUS`.

Note

`defaultPropertyGroupConfig` requiere la versión 3.1.0 o posterior del componente recopilador IoT SiteWise OPC UA. Para obtener más información, consulte [Actualice la versión de un componente AWS IoT SiteWise](#).

`discoveryConfig`— Solo pasarelas V3 habilitadas para MQTT

(Opcional) Configuración para el proceso de descubrimiento de nodos OPC UA.

`enableNodeTraversal`

Especifica si se deben seguir recorriendo los nodos secundarios del nodo raíz definidos por el filtro de nodos de la fuente de datos. Cuando se establece en `false`, la detección se detiene en el nodo raíz.

`avoidNodeTreeLoops`

Especifica si se deben evitar los bucles infinitos durante el proceso de navegación del nodo OPC UA. Cuando se establece en `true`, la puerta de enlace rastrea los nodos visitados para evitar referencias circulares.

`maxConcurrentBrowseRequests`

El número máximo de solicitudes de navegación simultáneas que el servidor OPC UA puede gestionar simultáneamente. El rango válido es de 1 a 500.

`maxNodesPerBrowseRequest`

El número máximo de nodos que se deben enviar en cada solicitud de navegación al servidor OPC UA. El rango válido es de 1 a 1000.

`periodicDiscovery`

Configuración para ejecutar el descubrimiento periódicamente a intervalos fijos. La detección periódica está habilitada cuando se proporciona esta configuración.

interval

El tiempo que transcurre entre las operaciones de detección periódicas. Se puede utilizar m durante minutos, h horas y d días. Por ejemplo, 90m o 1h. El intervalo mínimo es de 30 segundos.

maxNodesDiscoveredPerInterval

El número máximo de nodos que se deben descubrir por intervalo de descubrimiento. Esto ayuda a controlar la carga tanto en la puerta de enlace como en el servidor OPC UA.

Note

`periodicDiscovery` requiere la versión 3.1.0 o posterior del componente recopilador IoT SiteWise OPC UA. Para obtener más información, consulte [Actualice la versión de un componente AWS IoT SiteWise](#).

Note

Si la detección se repite de forma infinita, habilítela. `avoidNodeTreeLoops`. Supervise el progreso de la detección en CloudWatch los registros del `aws.iot.SiteWiseOpcUaCollector` componente.

propertyGroups

(Opcional) La lista de grupos de propiedades que definen la deadband y el scanMode solicitados por el protocolo.

name

El nombre del grupo de propiedades. Debe ser un identificador único.

deadband

El valor de deadband define el cambio mínimo en el valor de un punto de datos que debe producirse antes de que los datos se envíen a la nube. Contiene la siguiente información:

type

Los tipos de banda muerta admitidos. Puede elegir las opciones siguientes:

- **ABSOLUTE**: valor fijo que especifica el cambio absoluto mínimo necesario para considerar un punto de datos lo suficientemente significativo como para su envío a la nube.
- **PERCENT**: valor dinámico que especifica el cambio mínimo requerido como porcentaje del valor del último punto de datos enviado. Este tipo de zona neutra resulta útil cuando los valores de los datos varían considerablemente a lo largo del tiempo.

value

El valor de la banda muerta. Cuando `type` es **ABSOLUTE**, este valor es un doble sin unidades. Cuando `type` es **PERCENT**, este valor es un doble entre 1 y 100.

eguMin

(Opcional) La unidad de ingeniería mínima al utilizar una banda muerta **PERCENT**. Establézcala si el servidor OPC UA no tiene unidades de ingeniería configuradas.

eguMax

(Opcional) La unidad de ingeniería máxima al utilizar una banda muerta **PERCENT**. Establézcala si el servidor OPC UA no tiene unidades de ingeniería configuradas.

timeoutMilliseconds

La duración en milisegundos antes del tiempo de espera. El mínimo es 100.

scanMode

La estructura de `scanMode` que contiene la siguiente información:

type

Los tipos admitidos de `scanMode`. Los valores aceptados son **POLL** y **EXCEPTION**.

rate

El intervalo de muestreo para el modo de escaneo.

timestampToReturn

Origen de la marca de tiempo. Puede elegir las opciones siguientes:

- **SOURCE_TIME**: utiliza la marca de tiempo del dispositivo.
- **SERVER_TIME**: utiliza la marca de tiempo del servidor.

Note

Úselo `TimestampToReturn` con la versión 2.5.0 o posterior del componente recopilador IoT SiteWise OPC UA. Si utiliza esta función con versiones anteriores, se produce un error en las actualizaciones de configuración. Para obtener más información, consulte [Actualice la versión de un componente AWS IoT SiteWise](#).

nodeFilterRuleDefinitions

(Opcional) Una lista de rutas de nodos para incluir en el grupo de propiedades. Los grupos de propiedades no pueden solaparse. Si no especifica un valor para este campo, el grupo contendrá todas las rutas bajo la raíz y no podrá crear grupos de propiedades adicionales. La estructura `nodeFilterRuleDefinitions` contiene la siguiente información:

type

`OpcUaRootPath` es el único tipo admitido. Especifica que el valor de `rootPath` es una ruta relativa a la raíz del espacio de navegación de OPC UA.

rootPath

Lista delimitada por comas que especifica las rutas (relativas a la raíz) por incluir en el grupo de propiedades.

Ejemplos de configuración de capacidades adicionales para transmisiones clásicas y puertas de enlace V2 ()AWS CLI

El siguiente ejemplo define la configuración de la capacidad de una puerta de enlace OPC UA SiteWise Edge a partir de una carga útil almacenada en un archivo JSON.

```
aws iotsitewise update-gateway-capability-configuration \
--capability-namespace "iotsitewise:opcuacollector:2" \
--capability-configuration file://opc-ua-configuration.json
```

Example: configuración de origen OPC UA

El siguiente archivo `opc-ua-configuration.json` define una configuración de origen OPC UA básica y no segura.

```
{
```

```

"sources": [
  {
    "name": "Wind Farm #1",
    "endpoint": {
      "certificateTrust": {
        "type": "TrustAny"
      },
      "endpointUri": "opc.tcp://203.0.113.0:49320",
      "securityPolicy": "NONE",
      "messageSecurityMode": "NONE",
      "identityProvider": {
        "type": "Anonymous"
      },
      "nodeFilterRules": []
    },
    "measurementDataStreamPrefix": ""
  }
]
}

```

Example: configuración de origen OPC UA con grupos de propiedades definidos

El siguiente archivo `opc-ua-configuration.json` define una configuración de origen OPC UA básica y no segura con grupos de propiedades definidos.

```

{
  "sources": [
    {
      "name": "source1",
      "endpoint": {
        "certificateTrust": {
          "type": "TrustAny"
        },
        "endpointUri": "opc.tcp://10.0.0.9:49320",
        "securityPolicy": "NONE",
        "messageSecurityMode": "NONE",
        "identityProvider": {
          "type": "Anonymous"
        },
        "nodeFilterRules": [
          {
            "action": "INCLUDE",
            "definition": {

```

```

        "type": "OpcUaRootPath",
        "rootPath": "/Utilities/Tank"
    }
}
],
},
"measurementDataStreamPrefix": "propertyGroups",
"propertyGroups": [
    {
        "name": "Deadband_Abs_5",
        "nodeFilterRuleDefinitions": [
            {
                "type": "OpcUaRootPath",
                "rootPath": "/Utilities/Tank/Temperature/TT-001"
            },
            {
                "type": "OpcUaRootPath",
                "rootPath": "/Utilities/Tank/Temperature/TT-002"
            }
        ],
        "deadband": {
            "type": "ABSOLUTE",
            "value": 5.0,
            "timeoutMilliseconds": 120000
        }
    },
    {
        "name": "Polling_10s",
        "nodeFilterRuleDefinitions": [
            {
                "type": "OpcUaRootPath",
                "rootPath": "/Utilities/Tank/Pressure/PT-001"
            }
        ],
        "scanMode": {
            "type": "POLL",
            "rate": 10000
        }
    },
    {
        "name": "Percent_Deadband_Timeout_90s",
        "nodeFilterRuleDefinitions": [
            {
                "type": "OpcUaRootPath",

```

```

        "rootPath": "/Utilities/Tank/Flow/FT-*"
      }
    ],
    "deadband": {
      "type": "PERCENT",
      "value": 5.0,
      "eguMin": -100,
      "eguMax": 100,
      "timeoutMilliseconds": 90000
    }
  }
]
}

```

Example: configuración de origen OPC UA con propiedades

El siguiente JSON de ejemplo de `opc-ua-configuration.json` define una configuración de origen OPC UA con las siguientes propiedades:

- Confía en cualquier certificado.
- Utiliza la política de seguridad BASIC256 para proteger los mensajes.
- Utiliza el modo SIGN_AND_ENCRYPT para proteger las conexiones.
- Utiliza credenciales de autenticación almacenadas en un secreto del administrador de secretos.
- Filtra los flujos de datos, excepto aquellos cuya ruta comienza por `/WindFarm/2/WindTurbine/`.
- Añade `/Washington` al inicio de cada ruta de flujo de datos para distinguir entre este “Wind Farm #2” y un “Wind Farm #2” en otra área.

```

{
  "sources": [
    {
      "name": "Wind Farm #2",
      "endpoint": {
        "certificateTrust": {
          "type": "TrustAny"
        },
        "endpointUri": "opc.tcp://203.0.113.1:49320",
        "securityPolicy": "BASIC256",

```

```

        "messageSecurityMode": "SIGN_AND_ENCRYPT",
        "identityProvider": {
            "type": "Username",
            "usernameSecretArn":
"arn:aws:secretsmanager:region:123456789012:secret:greengrass-windfarm2-auth-1ABCDE"
        },
        "nodeFilterRules": [
            {
                "action": "INCLUDE",
                "definition": {
                    "type": "OpcUaRootPath",
                    "rootPath": "/WindFarm/2/WindTurbine/"
                }
            }
        ],
        "measurementDataStreamPrefix": "/Washington"
    }
}

```

Example: configuración de origen OPC UA con certificado de confianza

El siguiente JSON de ejemplo de `opc-ua-configuration.json` define una configuración de origen OPC UA con las siguientes propiedades:

- Confiar en un certificado X.509 dado.
- Utiliza la política de seguridad BASIC256 para proteger los mensajes.
- Utiliza el modo SIGN_AND_ENCRYPT para proteger las conexiones.

```

{
  "sources": [
    {
      "name": "Wind Farm #3",
      "endpoint": {
        "certificateTrust": {
          "type": "X509",
          "certificateBody": "-----BEGIN CERTIFICATE-----
MIICiTCCAfICCQD6m7oRw0uX0jANBgkqhkiG9w
0BAQUFADCBiDELMAKGA1UEBhMVCVVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwdTZ
WF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xZDAsBgNVBAsTC01BTSBDb25zb2x1MRIw

```

```

EAYDVQQDEw1UZsXN0Q21sYWMxHzAdBgkqhkiG9w0BCQEWEG5vb251QGfTYXpvi5
jb20wHhcNMTEwNDI1MjA0NTIxWhcNMTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBh
MCVVMxMjA0NTIxWjCBiDELMAkGA1UEBhMCAwZG9wYDQwYDQwYDQwYDQwYDQwYDQw
WF6b24xZDAsBgNVBAgTQ01BTSBDb25zb2x1MRIwEAYDVQQDEw1UZsXN0Q21sYWMx
HzAdBgkqhkiG9w0BCQEWEG5vb251QGfTYXpvi5jb20wgZ8wDQYJKoZIhvcNAQE
BBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ21uUSfwfEvySWtC2XADZ4nB+BLYgVI
k60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9TrDHudUZg3qX4waLG5M43q7Wgc/MbQ
ITx0USQv7c7ugFFDzQGBzZswY6786m86gpEibb30hjZnzcVQAaRHhd1QWIMm2nr
AgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4nUhVvxYUntneD9+h8Mg9q6q+auN
KyExzyLwax1Aoo7TJHidbtS4J5iNmZgXL0FkbFFBjvSfpJI1J00zbhNYS5f6Guo
EDmFJl0ZxBHjJnyp3780D8uTs7fLvJx79LjSTbNYiytVbZPQUQ5Yaxu2jXnimvw
3rrszlaEXAMPLE=
    -----END CERTIFICATE-----",
        "certificateChain": "-----BEGIN CERTIFICATE-----
MIICiTCCAfICCD6m7oRw0uX0jANBgkqhkiG9w
0BAQUFADCBIDELMAkGA1UEBhMCAwZG9wYDQwYDQwYDQwYDQwYDQwYDQwYDQw
WF6b24xZDAsBgNVBAgTQ01BTSBDb25zb2x1MRIwEAYDVQQDEw1UZsXN0Q21sYWMx
EAYDVQQDEw1UZsXN0Q21sYWMxHzAdBgkqhkiG9w0BCQEWEG5vb251QGfTYXpvi5
jb20wHhcNMTEwNDI1MjA0NTIxWhcNMTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBh
MCVVMxMjA0NTIxWjCBiDELMAkGA1UEBhMCAwZG9wYDQwYDQwYDQwYDQwYDQwYDQw
WF6b24xZDAsBgNVBAgTQ01BTSBDb25zb2x1MRIwEAYDVQQDEw1UZsXN0Q21sYWMx
HzAdBgkqhkiG9w0BCQEWEG5vb251QGfTYXpvi5jb20wgZ8wDQYJKoZIhvcNAQE
BBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ21uUSfwfEvySWtC2XADZ4nB+BLYgVI
k60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9TrDHudUZg3qX4waLG5M43q7Wgc/MbQ
ITx0USQv7c7ugFFDzQGBzZswY6786m86gpEibb30hjZnzcVQAaRHhd1QWIMm2nr
AgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4nUhVvxYUntneD9+h8Mg9q6q+auN
KyExzyLwax1Aoo7TJHidbtS4J5iNmZgXL0FkbFFBjvSfpJI1J00zbhNYS5f6Guo
EDmFJl0ZxBHjJnyp3780D8uTs7fLvJx79LjSTbNYiytVbZPQUQ5Yaxu2jXnimvw
3rrszlaEXAMPLE=
    -----END CERTIFICATE-----"
    },
    "endpointUri": "opc.tcp://203.0.113.2:49320",
    "securityPolicy": "BASIC256",
    "messageSecurityMode": "SIGN_AND_ENCRYPT",
    "identityProvider": {
        "type": "Anonymous"
    },
    "nodeFilterRules": []
},
"measurementDataStreamPrefix": ""
}
]

```

```
}
```

Configure los servidores OPC UA para que confíen en la puerta de enlace Edge AWS IoT SiteWise

Si elige una opción `messageSecurityMode` distinta de Ninguna al configurar su fuente OPC UA, debe permitir que sus servidores de origen confíen en la puerta de enlace AWS IoT SiteWise Edge. La puerta de enlace SiteWise Edge genera un certificado que puede necesitar su servidor de origen. El proceso varía en función de los servidores de origen. Para obtener más información, consulte la documentación de los servidores.

El siguiente procedimiento describe los pasos básicos.

Para permitir que un servidor OPC UA confíe en la puerta de enlace SiteWise Edge

1. Abra la interfaz para configurar el servidor OPC UA.
2. Escriba el nombre de usuario y la contraseña del administrador del servidor OPC UA.
3. Localice Clientes de confianza en la interfaz y, a continuación, seleccione Cliente de puerta de enlace de AWS IoT SiteWise para conectarse.
4. Elija Confiar.

Exportación del certificado de cliente OPC UA

Algunos servidores OPC UA requieren acceso al archivo de certificado de cliente OPC UA para poder confiar en la SiteWise puerta de enlace Edge. Si esto se aplica a sus servidores OPC UA, puede utilizar el siguiente procedimiento para exportar el certificado de cliente OPC UA desde la puerta de enlace Edge. SiteWise A continuación, puede importar el certificado en su servidor OPC UA.

Exportación del archivo de certificado de cliente OPC UA de un origen

1. Ejecute el siguiente comando para ir al directorio que contiene el archivo de certificado.
`sitewise-work` Sustitúyala por la ruta de almacenamiento local de la carpeta de trabajo de `aws.iot.SiteWiseEdgeCollectorOpcua` Greengrass y `source-name` sustitúyala por el nombre de la fuente de datos.

De forma predeterminada, la carpeta de trabajo de Greengrass está `/greengrass/v2/work/aws.iot.SiteWiseEdgeCollectorOpcua` en Linux y `C:/greengrass/v2/work/aws.iot.SiteWiseEdgeCollectorOpcua` activada. Microsoft Windows

```
cd /sitewise-work/source-name/opcua-certificate-store
```

2. El certificado de cliente OPC UA de la puerta de enlace SiteWise Edge para esta fuente está en el `aws-iot-opcua-client.pfx` archivo.

Ejecute el siguiente comando para exportar el certificado a un archivo `.pem` llamado `aws-iot-opcua-client-certificate.pem`.

```
keytool -exportcert -v -alias aws-iot-opcua-client -keystore aws-iot-opcua-client.pfx -storepass amazon -storetype PKCS12 -rfc > aws-iot-opcua-client-certificate.pem
```

3. Transfiera el archivo de certificado desde la puerta de enlace SiteWise Edge al servidor OPC UA. `aws-iot-opcua-client-certificate.pem`

Para ello, puede utilizar un software común como el programa `scp` para transferir el archivo utilizando el protocolo SSH. Para obtener más información, consulte [Copia segura](#) en Wikipedia.

Note

Si su puerta de enlace SiteWise Edge se ejecuta en Amazon Elastic Compute Cloud (Amazon EC2) y se conecta a ella por primera vez, debe configurar los requisitos previos para conectarse. Para obtener más información, consulte [Conexión a la instancia de Linux con SSH](#) en la Guía del usuario de Amazon EC2.

4. Importe el archivo de certificado en el servidor OPC UA para confiar en la SiteWise puerta de enlace Edge. `aws-iot-opcua-client-certificate.pem` Los pasos pueden variar en función del servidor de origen que utilice. Consulte la documentación de cada servidor.

Filtrado de rangos de ingesta de datos con OPC UA

Puede controlar la forma de ingerir datos con un origen OPC UA mediante rangos de zona neutra y modo de análisis. Estas funciones le permiten controlar qué tipo de datos desea ingerir y cómo y cuándo su servidor y la puerta de enlace SiteWise Edge intercambian esta información.

Recopile o filtre los datos en función de la calidad

Puede configurar los ajustes de calidad de los datos para controlar qué datos se recopilan del origen OPC UA. El origen de datos incluye la calificación de calidad como metadatos cuando los envía.

Puede seleccionar una o todas las opciones siguientes:

- Good
- Bad
- Uncertain

Manejar NaN o valores nulos

SiteWise Edge admite la recopilación y el manejo de NaN y valores nulos.

- NaN (no es un número): representa resultados numéricos indefinidos o irrepresentables.
- Nulo: indica que faltan datos.

El recopilador IoT SiteWise OPC UA captura los valores NaN y Null con una calidad MALA o INCIERTA. Estos valores especiales se escriben en la transmisión local, lo que permite una recopilación de datos más completa.

Control de la frecuencia de recopilación de datos con el modo de escaneo

Puede configurar el modo de análisis OPC UA para controlar el modo de recopilación de datos desde su origen OPC UA. Puede elegir el modo de suscripción o de sondeo.

- Modo de suscripción: la fuente OPC UA recopila datos para enviarlos a su puerta de enlace SiteWise Edge con la frecuencia definida por su velocidad de escaneo. El servidor solo envía datos cuando el valor ha cambiado, por lo que esta es la frecuencia máxima con la que su puerta de enlace SiteWise Edge recibe datos.
- Modo de sondeo: su puerta de enlace SiteWise Edge sondea la fuente OPC UA a una frecuencia establecida definida por su velocidad de escaneo. El servidor envía los datos independientemente de si el valor ha cambiado, por lo que su puerta de enlace SiteWise Edge siempre recibe los datos en este intervalo.

Note

La opción de modo de sondeo anula sus ajustes de banda muerta para este origen.

Filtrado de la ingesta de datos OPC UA con intervalos de zona neutra

Puede aplicar una zona neutra a sus grupos de propiedades de origen OPC UA para filtrar y descartar determinados datos en lugar de enviarlos a la Nube de AWS . Una zona neutra especifica una ventana de fluctuaciones esperadas en los valores de datos entrantes desde su origen OPC UA. Si los valores se encuentran dentro de este intervalo, su servidor OPC UA no los enviará a la AWS nube. Puedes usar el filtrado de banda muerta para reducir la cantidad de datos que procesas y envías a la AWS nube. Para obtener información sobre cómo configurar las fuentes OPC UA para su puerta de enlace SiteWise Edge, consulte. [Fuentes de datos OPC UA para AWS IoT SiteWise pasarelas Edge](#)

Note

Su servidor elimina todos los datos que caen dentro de la ventana especificada por su banda muerta. Los datos descartados no se pueden recuperar.

Tipos de bandas muertas

Puede especificar dos tipos de zonas neutras para su grupo de propiedades del servidor OPC UA. Estas le permiten elegir cuántos datos se envían a la nube de AWS y cuántos se descartan.

- **Porcentaje:** usted especifica una ventana utilizando un porcentaje de fluctuación esperada en el valor de medición. El servidor calcula el intervalo exacto a partir de este porcentaje y envía a la AWS nube los datos que exceden el intervalo que queda fuera del intervalo. Por ejemplo, si se especifica un valor de banda muerta del 2% en un sensor con un rango de -100 grados Fahrenheit a +100 grados Fahrenheit, se indica al servidor que envíe los datos a la AWS nube cuando el valor cambie 4 grados Fahrenheit o más.

Note

Si lo desea, puede especificar un valor mínimo y máximo de banda muerta para esta ventana si su servidor de origen no define unidades de ingeniería. Si no se proporciona un rango de unidades de ingeniería, el servidor OPC UA utilizará de forma predeterminada el rango completo del tipo de datos de medición.

- **Absoluto:** usted especifica una ventana utilizando unidades exactas. Por ejemplo, si se especifica un valor de banda inactiva de 2 en un sensor, se indica al servidor que envíe los datos a la AWS nube cuando su valor cambie al menos 2 unidades. Puede utilizar la banda muerta absoluta

para entornos dinámicos en los que se esperan fluctuaciones regulares durante las operaciones normales.

Tiempos de espera de banda muerta

Si lo desea, puede configurar un tiempo de espera de banda muerta. Una vez transcurrido este tiempo de espera, el servidor OPC UA envía el valor de medición actual aunque se encuentre dentro de la fluctuación de zona neutra esperada. Puede usar la configuración de tiempo de espera para asegurarse de que AWS IoT SiteWise se ingiere un flujo constante de datos en todo momento, incluso cuando los valores no superen el intervalo de tiempo definido.

Utilice filtros de nodo OPC UA en Edge SiteWise

Al definir las fuentes de datos OPC UA para una puerta de enlace SiteWise Edge, puede definir filtros de nodos. Los filtros de nodos le permiten limitar las rutas de flujo de datos que la puerta de enlace SiteWise Edge envía a la nube. Puedes usar filtros de nodos para reducir el tiempo de inicio de la puerta de enlace SiteWise Edge y el uso de la CPU al incluir únicamente las rutas a los datos que modelos AWS IoT SiteWise. De forma predeterminada, las pasarelas SiteWise Edge cargan todas las rutas OPC UA excepto las que comienzan por `/Server/`. Puede utilizar los caracteres comodín `*` y `**` en los filtros de nodos para incluir varios flujos de datos con un solo filtro. Para obtener información sobre cómo configurar las fuentes OPC UA para su puerta de enlace SiteWise Edge, consulte [Fuentes de datos OPC UA para AWS IoT SiteWise pasarelas Edge](#)

Note

AWS IoT SiteWise reinicia la puerta de enlace SiteWise Edge cada vez que añada o edite una fuente. La puerta de enlace SiteWise Edge no ingiere datos mientras actualiza la configuración de la fuente. El tiempo necesario para reiniciar la puerta de enlace SiteWise Edge depende de la cantidad de etiquetas que haya en las fuentes de la puerta de enlace SiteWise Edge. El tiempo de reinicio puede oscilar entre unos segundos (para una puerta de enlace SiteWise Edge con pocas etiquetas) y varios minutos (para una puerta de enlace SiteWise Edge con muchas etiquetas).

En la tabla siguiente se enumeran los caracteres comodín que se pueden utilizar para filtrar orígenes de datos OPC UA.

Comodines para filtros de nodos OPC UA

Comodín	Description (Descripción)
*	Coincide con un solo nivel en una ruta de flujo de datos.
**	Coincide con varios niveles en una ruta de flujo de datos.

Note

Si configura una fuente con un filtro amplio y luego cambia la fuente para usar un filtro más restrictivo, AWS IoT SiteWise deja de almacenar los datos que no coinciden con el nuevo filtro.

Example: Escenario que utiliza filtros de nodos

Piense en los siguientes flujos de datos hipotéticos:

- /WA/Factory 1/Line 1/PLC1
- /WA/Factory 1/Line 1/PLC2
- /WA/Factory 1/Line 2/Counter1
- /WA/Factory 1/Line 2/PLC1
- /OR/Factory 1/Line 1/PLC1
- /OR/Factory 1/Line 2/Counter2

Con los flujos de datos anteriores, puede definir filtros de nodos para limitar qué datos se van a incluir de su origen OPC UA.

- Para seleccionar todos los nodos en este ejemplo, utilice / o /**/. Puede incluir varios directorios o carpetas con los caracteres comodín **.
- Para seleccionar todos los flujos de datos de PLC, utilice /*/*/*/PLC* o /**/PLC*.
- Para seleccionar todos los contadores en este ejemplo, utilice /**/Counter* o /*/*/*/Counter*.

- Para seleccionar todos los contadores de Line 2, utilice `/**/Line 2/Counter*`.

Conversión de tipos de datos no admitidos

Si lo desea, habilite la conversión de tipos AWS IoT SiteWise de datos en matrices y tipos de DateTime datos simples. AWS IoT SiteWise no es compatible con todos los tipos de datos OPC UA. Cuando envías datos no compatibles a tu flujo de AWS IoT Greengrass datos, esos datos se pierden. Sin embargo, al convertir los tipos de datos nativos no compatibles en cadenas, puede ingerir los datos en AWS IoT SiteWise lugar de descartarlos. AWS IoT SiteWise serializa los datos convertidos para que luego pueda utilizar sus propias funciones para volver a convertir las cadenas a su tipo de datos original en sentido descendente, si es necesario.

Puede actualizar la configuración de conversión de tipos de datos para un origen de datos en cualquier momento y cada origen de datos puede tener su propia configuración.

Al añadir fuentes de datos en Consola de AWS IoT SiteWise, hay dos casillas de verificación en la sección Conversión de tipos de datos en la configuración avanzada. Puede indicar qué tipos de datos desea convertir en cadenas.

Además, el recopilador IoT SiteWise OPC UA puede aceptar NaN o valores nulos en el borde.

- Conversión de valores de matriz con tipos de datos simples en cadenas JSON
- Convierta DateTime valores en cadenas ISO 8601

Requisito previo

- Utilice la versión 2.5.0 o posterior del recopilador [IoT SiteWise OPC UA](#).

Limitaciones

Estas son las limitaciones de la conversión de tipos de datos OPC UA a cadenas en AWS IoT SiteWise.

- No se admite la conversión de tipos de datos complejos.
- El límite de cadena después de la conversión es de 1024 bytes. Si la cadena tiene más de 1024 bytes, se rechaza por. AWS IoT SiteWise

Configurar la autenticación de fuentes de datos para SiteWise Edge

Si su servidor OPC UA requiere credenciales de autenticación para conectarse, puede utilizarlas AWS Secrets Manager para crear e implementar un secreto en su puerta de enlace SiteWise Edge. AWS Secrets Manager cifra los secretos del dispositivo para proteger su nombre de usuario y contraseña hasta que necesite usarlos. Para obtener más información sobre el componente de administrador de AWS IoT Greengrass secretos, consulte [Secret Manager](#) en la Guía para AWS IoT Greengrass Version 2 desarrolladores.

Para obtener información sobre la administración del acceso a los secretos de Secrets Manager, consulte:

- [Quién tiene permisos para acceder a tus AWS Secrets Manager secretos.](#)
- [Determinar si una solicitud está permitida o denegada en una cuenta.](#)

Paso 1: Crear secretos de autenticación de origen

Se puede utilizar AWS Secrets Manager para crear un secreto de autenticación para la fuente de datos. En el secreto, defina pares clave-valor, **username** y **password**, que contengan detalles de autenticación para su origen de datos.

Para crear un secreto (consola)

1. Vaya a la [consola de AWS Secrets Manager](#).
2. Seleccione Almacenar un nuevo secreto.
3. En Tipo de secreto, seleccione Otro tipo de secretos.
4. En Pares clave/valor, haga lo siguiente:
 1. En el primer cuadro de entrada, introduzca **username** y, en el segundo cuadro de entrada, introduzca el nombre de usuario.
 2. Seleccione Agregar regla.
 3. En el primer cuadro de entrada, introduzca **password** y, en el segundo cuadro de entrada, introduzca la contraseña.
5. En Clave de cifrado, seleccione aws/secretsmanager y, a continuación, elija Siguiente.
6. En la página Guardar un nuevo secreto, introduzca un Nombre secreto.
7. (Opcional) Introduzca una Descripción que le ayude a identificar este secreto y, a continuación, seleccione Siguiente.

8. (Opcional) En la página Guardar un nuevo secreto, active Rotación automática. Para obtener más información, consulte [Rotación de secretos](#) en la Guía del usuario de AWS Secrets Manager .
9. Especifique un calendario de rotación.
10. Elija una función de Lambda que pueda rotar este secreto y, a continuación, seleccione Siguiente.
11. Revise la configuración de sus secretos y, a continuación, seleccione Guardar.

Para autorizar la interacción con la puerta de enlace de SiteWise Edge AWS Secrets Manager, la función de IAM de la puerta de enlace de SiteWise Edge debe permitir esta `secretsmanager:GetSecretValue` acción. Puede utilizar el dispositivo principal de Greengrass para buscar la política de IAM. Para obtener más información sobre la actualización de una política de IAM, consulte [Edición de políticas de IAM](#) en la Guía del usuario.AWS Identity and Access Management

Example policy

*secret-arn*Sustitúyalo por el nombre de recurso de Amazon (ARN) del secreto que creaste en el paso anterior. Para obtener más información sobre cómo obtener el ARN de un secreto, consulte [Buscar secretos AWS Secrets Manager en](#) la Guía del AWS Secrets Manager usuario.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "secretsmanager:GetSecretValue"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:secretsmanager:us-east-1:123456789012:secret/*"
      ]
    }
  ]
}
```

Paso 2: Implemente los secretos en su dispositivo de puerta de enlace SiteWise Edge

Puede usar la AWS IoT SiteWise consola para implementar datos secretos en su puerta de enlace SiteWise Edge.

Para implementar un secreto (consola)

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, seleccione Puertas de enlace.
3. En la lista de puertas de enlace, elija la puerta de enlace SiteWise Edge de destino.
4. En la sección de configuración de la puerta de enlace, elija el enlace del dispositivo principal de Greengrass para abrir el AWS IoT Greengrass núcleo asociado a la puerta de enlace SiteWise Edge.
5. En el panel de navegación, seleccione Implementaciones.
6. Seleccione la implementación de destino y, a continuación, Revisar.
7. En la página Especificar destino, seleccione Siguiente.
8. En la página Seleccionar componentes, en la sección Componentes públicos, desactive la opción Mostrar solo componentes seleccionados.
9. Busque y elija `aws.greengrass.SecretManager` componente y, a continuación, elija Siguiente.
10. En la lista de componentes seleccionados, elija `aws.greengrass.SecretManager` componente y, a continuación, elija Configurar componente.
11. En el campo Configuración por fusionar, añada el siguiente objeto JSON.

Note

`secret-arn` Sustitúyalo por el ARN del secreto que creó en el paso anterior. Para obtener más información sobre cómo obtener el ARN de un secreto, consulte [Buscar secretos AWS Secrets Manager en la Guía del AWS Secrets Manager usuario](#).

```
{
  "cloudSecrets": [
    {
      "arn": "secret-arn"
    }
  ]
}
```

```
}
```

12. Seleccione Confirmar.
13. Elija Siguiente.
14. En la página Configurar opciones avanzadas, seleccione Siguiente.
15. Revise las configuraciones de su implementación y, a continuación, seleccione Implementar.

Paso 3: Añadir configuraciones de autenticación

Puede usar la AWS IoT SiteWise consola para agregar configuraciones de autenticación a su puerta de enlace SiteWise Edge.

Para añadir configuraciones de autenticación (consola)

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En la lista de puertas de enlace, elija la puerta de enlace SiteWise Edge de destino.
3. En la lista Orígenes de datos, seleccione el origen de datos de destino y, a continuación, Editar.
4. En la página Añadir un origen de datos, seleccione Configuración avanzada.
5. En Configuración de autenticación, elija el secreto que implementó en el paso anterior.
6. Seleccione Save.

Fuentes de datos asociadas en las puertas de enlace SiteWise Edge

Al utilizar una puerta de enlace de AWS IoT SiteWise Edge, puede conectar una fuente de datos asociada a su puerta de enlace de SiteWise Edge y recibir datos del socio en su puerta de enlace de SiteWise Edge y en la AWS nube. Estas fuentes de datos de los socios son AWS IoT Greengrass componentes que se desarrollan en colaboración entre el socio AWS y el socio. Cuando añada una fuente de datos asociada, AWS IoT SiteWise creará este componente y lo implementará en su puerta de enlace SiteWise Edge.

Note

Puede agregar una fuente de datos para cada socio en cada puerta de enlace.

Para añadir un origen de datos de un socio, realice lo siguiente:

1. [Agregue una fuente de datos asociada en Edge SiteWise](#)
2. Vaya al portal web del socio, si corresponde, y configure la fuente de datos del socio para que se conecte a la puerta de enlace SiteWise Edge.

Temas

- [Seguridad](#)
- [Configúrelo Docker en su puerta de enlace SiteWise Edge](#)
- [Agregue una fuente de datos asociada en Edge SiteWise](#)
- [SiteWise Opciones de fuentes de datos de los socios de Edge Gateway](#)

Seguridad

Como parte del [modelo de responsabilidad compartida](#) entre AWS nuestros clientes y nuestros socios, a continuación se describe quién es responsable de los distintos aspectos de la seguridad:

Responsabilidad del cliente

- Investigar al socio.
- Configurar el acceso a la red otorgado al socio.
- Supervisión para garantizar un uso razonable de los recursos de la máquina SiteWise Edge Gateway (CPU, memoria y sistema de archivos).

AWS responsabilidad

- Aislar al socio de los recursos AWS en la nube del cliente, excepto los que necesite el socio. En este caso, ingesta de AWS IoT SiteWise .
- Restringir la solución del socio a un uso razonable de los recursos de la máquina de SiteWise Edge Gateway (CPU y memoria).

Responsabilidad del socio

- Uso de valores predeterminados seguros.
- Mantener la solución segura en el tiempo mediante parches y otras actualizaciones apropiadas.
- Mantener la confidencialidad de los datos de los clientes.

Configúrelo Docker en su puerta de enlace SiteWise Edge

AWS IoT SiteWise proporciona una Docker imagen que le permite ejecutar la aplicación SiteWise Edge en varias plataformas y entornos. Esta Docker imagen resume todos los componentes y

dependencias necesarios para recopilar, procesar y enviar datos de sus equipos industriales a la AWS nube. Con la imagen de Docker, puede implementar y ejecutar la aplicación SiteWise Edge en hosts compatibles con Docker, como servidores, dispositivos perimetrales o servicios de contenedores basados en la nube.

Para añadir un origen de datos de socios debe instalar [Docker Engine](#) 1.9.1 o posterior en su dispositivo local.

Note

La versión 20.10 es la última que se ha comprobado que funciona con el software Edge Gateway. SiteWise

Verifica que Docker esté instalada

Para comprobar si Docker está instalado, ejecute el siguiente comando desde un terminal conectado a la puerta de enlace SiteWise Edge:

```
docker info
```

Si el comando devuelve un `docker is not recognized` resultado o si Docker hay instalada una versión anterior de, [instale Docker Engine](#) antes de continuar.

Configuración de Docker

El usuario del sistema que ejecuta un componente Docker contenedor debe tener permisos de administrador o root, o bien debe configurarlo Docker para ejecutarlo como usuario no root o no administrador.

En los dispositivos Linux, debe añadir un `ggc_user` usuario al `docker` grupo para poder ejecutar comandos sin él. Docker sudo

Para añadir `ggc_user` al `docker` grupo al usuario no root que utiliza para ejecutar los componentes del Docker contenedor, ejecute el siguiente comando:

```
sudo usermod -aG docker ggc_user
```

Para obtener más información, consulte los [pasos posteriores a la instalación de Linux para Docker Engine](#).

Agregue una fuente de datos asociada en Edge SiteWise

Para conectar una fuente de datos asociada a su puerta de enlace SiteWise Edge, agréguela como fuente de datos. Cuando lo añada como fuente de datos, AWS IoT SiteWise implementará un AWS IoT Greengrass componente privado en su puerta de enlace SiteWise Edge.

Requisitos previos

Para añadir un origen de datos de un socio, debe hacer lo siguiente:

- Para EasyEdge y CloudRail, cree una cuenta con el socio y, a continuación, vincule las cuentas.
- [Configúrelo Docker en su puerta de enlace SiteWise Edge](#)

Cree una puerta de enlace SiteWise Edge con una fuente de datos asociada

Si desea crear una nueva puerta de enlace SiteWise Edge, complete los pasos que se indican a continuación [Cree una puerta de enlace SiteWise Edge autohospedada](#). Una vez que haya creado la puerta de enlace SiteWise Edge, siga los pasos que se indican [Agregue una fuente de datos asociada a una puerta de enlace de SiteWise Edge existente](#) para agregar una fuente de datos asociada.

Agregue una fuente de datos asociada a una puerta de enlace de SiteWise Edge existente

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación de la izquierda, selecciona las puertas de enlace Edge en la sección Edge.
3. Elija la puerta de enlace SiteWise Edge a la que desee conectar la fuente de datos del socio.
4. En Orígenes de datos, elija Añadir origen de datos.
5. En la pantalla Agregar fuente de datos, elija un tipo de fuente para seleccionar el socio que conecta su puerta de enlace SiteWise Edge. Cada origen de datos tiene sus propias opciones de configuración. Hay dos categorías de orígenes de datos: orígenes de AWS y orígenes de socios.

Con un origen de datos de socios puede seleccionar un origen por puerta de enlace. Para obtener una lista de opciones de integración de socios de orígenes de datos, consulte [SiteWise Opciones de fuentes de datos de los socios de Edge Gateway](#). Tenga en cuenta que puede añadir hasta 100 fuentes de datos OPC UA (AWS fuentes). Para empezar a utilizar orígenes de datos OPC UA, consulte [Fuentes de datos OPC UA para AWS IoT SiteWise pasarelas Edge](#).

6. Introduzca un nombre para el origen.

7. Seleccione la pestaña del origen de datos que aparece a continuación y siga el procedimiento de configuración.

CloudRail

Gran parte de la CloudRail configuración se realiza en el CloudRail portal después de guardar la fuente de datos para la puerta de enlace SiteWise Edge. Sin embargo, es necesario autorizar la conexión.

Note

La conexión de CloudRail solo está disponible en Linux.

1. [Cree una cuenta de CloudRail](#) para empezar a conectarse a AWS IoT SiteWise.
2. Asegúrese de que Docker esté instalado en su puerta de enlace. Para obtener más información, consulte [Configúrelo Docker en su puerta de enlace SiteWise Edge](#).
3. Lea el acuerdo de Autorizar el acceso y la implementación y, a continuación, seleccione Autorizar. Al marcar esta casilla, el AWS socio tiene acceso a su fuente de datos y AWS permite la implementación en el componente del socio.

Note

El Prefijo de medición: opcional se establece en el portal de CloudRail.

Note

El software del socio es desarrollado, mantenido y respaldado por el AWS socio. AWS no es responsable de la interfaz, la configuración o el software.

Para obtener más información, consulte [CloudRail](#).

EasyEdge

Gran parte de la EasyEdge configuración se realiza en el EasyEdge portal después de guardar la fuente de datos para la puerta de enlace SiteWise Edge. Sin embargo, es necesario autorizar la conexión.

Note

La conexión de EasyEdge solo está disponible en Linux.

1. [Cree una cuenta de EasyEdge](#) para empezar a conectarse a AWS IoT SiteWise.
2. Asegúrese de que Docker esté instalado en su puerta de enlace. Para obtener más información, consulte [Configúrelo Docker en su puerta de enlace SiteWise Edge](#).
3. Lea el acuerdo de Autorizar el acceso y la implementación y, a continuación, seleccione Autorizar. Al marcar esta casilla, el AWS socio tiene acceso a su fuente de datos y AWS permite la implementación en el componente del socio.

Note

El Prefijo de medición: opcional se establece en el portal de EasyEdge.

Note

El software del socio es desarrollado, mantenido y respaldado por el AWS socio. AWS no es responsable de la interfaz, la configuración o el software.

Para obtener más información, consulte [EasyEdge](#).

Litmus Edge

Puede activar la configuración de Litmus de dos formas. Active Litmus Edge directamente mediante AWS IoT SiteWise utilizando información del portal de Litmus Edge Manager. También puede activar Litmus Edge manualmente para AWS IoT SiteWise a través de Litmus Edge Manager.

Note

La conexión de Litmus Edge solo está disponible en Linux.

Para activarlo mediante un código Litmus Edge de activación en AWS IoT SiteWise

Utilice este procedimiento cuando añada un origen de datos de Litmus Edge con un código de activación de Litmus Edge en Consola de AWS IoT SiteWise.

1. Seleccione **Actívalo ahora** con un código. Aparecen opciones de configuración adicionales.
2. Introduce el Litmus Edge Manager para conectarte Litmus Edge a tu puerta de enlace SiteWise Edge. Para obtener más información, consulte [Step 3a: Set Data and Device Management Endpoint](#) en la documentación de Litmus Edge Manager.
3. Proporcione el código de Litmus Edge Manager activación para Litmus Edge activarlo AWS IoT SiteWise
4. Si lo desea, AWS IoT SiteWise proporcione el certificado de Litmus Edge Manager CA. El certificado impide que Litmus Edge se active en un Litmus Edge Manager no autorizado.
5. Asegúrese de que Docker esté instalado en su puerta de enlace. Para obtener más información, consulte [Configúrelo Docker en su puerta de enlace SiteWise Edge](#).

Note

AWS IoT SiteWise despliega la aplicación asociada como un contenedor de Docker. La aplicación se implementa con capacidad NET_ADMIN para que se pueda administrar el contenedor de Docker de Litmus Edge a través de Litmus Edge Manager. Litmus Edge requiere que este acceso privilegiado se ejecute en los dispositivos. Para obtener más información sobre los requisitos de Litmus Edge Docker, consulta la sección [Instalación de Docker](#) en la QuickStart guía de la documentación de Litmus Edge.

6. Lea el acuerdo de Autorizar el acceso y la implementación y, a continuación, seleccione Autorizar. Al marcar esta casilla, el AWS socio tiene acceso a tu fuente de datos y permite AWS realizar la implementación en el componente del socio.

Activación manual mediante Litmus Edge

1. Seleccione Actívelo más tarde en Litmus Edge.
2. Asegúrese de que Docker esté instalado en su puerta de enlace. Para obtener más información, consulte [Configúrelo Docker en su puerta de enlace SiteWise Edge](#).

Note

AWS IoT SiteWise despliega la aplicación asociada como un contenedor de Docker. La aplicación se implementa con capacidad NET_ADMIN para que se pueda administrar el contenedor de Docker de Litmus Edge a través de Litmus Edge Manager. Litmus Edge requiere que este acceso privilegiado se ejecute en los dispositivos. Para obtener más información sobre los requisitos de Litmus Edge Docker, consulta la sección [Instalación de Docker](#) en la QuickStart guía de la documentación de Litmus Edge.

3. Lea el acuerdo de Autorizar el acceso y la implementación y, a continuación, seleccione Autorizar. Al marcar esta casilla, el AWS socio tiene acceso a tu fuente de datos y permite AWS realizar la implementación en el componente del socio.
4. Una vez completada la implementación, siga las instrucciones de [Access the Litmus Edge Web UI](#) en la documentación de Litmus Edge QuickStart Guide.

Note

El software del socio es desarrollado, mantenido y respaldado por el AWS socio. AWS no es responsable de la interfaz, la configuración o el software.

Para obtener más información, consulte [Litmus Edge](#).

8. Seleccione Save.

SiteWise Opciones de fuentes de datos de los socios de Edge Gateway

AWS IoT SiteWise le permite conectar e ingerir datos de varias fuentes de datos de socios, como equipos industriales, sensores y otros sistemas de terceros. Para conectar una fuente de datos asociada, debe seguir algunos pasos, como configurar la fuente de datos a la que enviar los datos AWS IoT SiteWise, configurar los permisos y la autenticación necesarios y asignar los datos a sus modelos de activos. Este proceso garantiza que sus datos de socios se integren sin problemas en su entorno de AWS IoT SiteWise, lo que le permite supervisarlos y analizarlos junto con otros orígenes de datos.

En esta sección se enumeran los socios disponibles para la integración de fuentes de datos de terceros en las puertas de enlace SiteWise Edge. Utilice la siguiente información para configurar un origen de datos de un socio.

Note

Puede agregar una fuente de datos para cada socio en cada puerta de enlace

CloudRail

Portal:

<https://devices.cloudrail.com/>

Requisitos

Para obtener más información sobre los requisitos de CloudRail, consulte [FAQS](#) en el sitio web de CloudRail.

Documentación de CloudRail:

[Computación perimetral: SiteWise perimetral](#)

EasyEdge

Portal:

<https://studio.easyedge.io/>

Requisitos

[Requisitos de EasyEdge](#): información sobre los requisitos de EasyEdge, incluidos los puntos de conexión y los puertos necesarios para configurar el firewall. Nota: Necesitará una cuenta de EasyEdge para acceder a esta documentación.

Documentación de EasyEdge:

[EasyEdge para AWS](#)

Litmus Edge

Acceda a Litmus Edge Manager:

Para acceder a Litmus Edge, configure una cuenta del [Litmus Edge Manager](#).

Requisitos

[Requisitos de Litmus Edge](#): configuraciones recomendadas y requisitos del sistema para implementar Litmus Edge.

Documentación de Litmus:

- [Integración a AWS IoT SiteWise](#)
- [Documentación de Litmus Edge](#)

AWS IoT Greengrass componentes para AWS IoT SiteWise Edge

SiteWise Edge utiliza AWS IoT Greengrass componentes para recopilar, procesar y transmitir datos industriales en la periferia. Estos componentes funcionan juntos para permitir el procesamiento local de datos y una integración perfecta con el servicio AWS IoT SiteWise en la nube.

SiteWise Publicador de IoT

El componente de SiteWise editor de IoT (`aws.iot.SiteWiseEdgePublisher`) es responsable de:

- Transmitir de forma segura los datos recopilados al servicio AWS IoT SiteWise en la nube
- Administrar el almacenamiento en búfer y los reintentos de datos durante problemas de conectividad

Para obtener más información sobre la configuración del editor para SiteWise Edge, consulte [Configure el AWS IoT SiteWise componente de publicación](#) Y, para obtener más información

sobre el componente de editor, consulte [IoT SiteWise Publisher](#) en la Guía para AWS IoT Greengrass Version 2 desarrolladores.

SiteWise Procesador IoT

El componente SiteWise procesador de IoT (`aws.iot.SiteWiseEdgeProcessor`) realiza las siguientes tareas:

- Ejecuta transformaciones y cálculos de datos en la periferia
- Implementación local de definiciones y cálculos de propiedades de activos
- Reducir el volumen de datos mediante la agregación o el filtrado de los datos antes de la transmisión

Para obtener más información sobre el componente del procesador, consulte el [SiteWise procesador IoT](#) en la Guía para AWS IoT Greengrass Version 2 desarrolladores.

Colector IoT SiteWise OPC UA

El componente IoT SiteWise OPC UA collector (`aws.iot.SiteWiseEdgeCollectorOpcua`) está diseñado para:

- Conéctese a servidores OPC UA en entornos industriales
- Recopile datos de fuentes de datos OPC UA de forma eficiente
- Transforma los datos del OPC UA en un formato compatible con AWS IoT SiteWise

Para obtener más información sobre el componente recopilador OPC UA, consulte el [recopilador SiteWise OPC UA de IoT](#) en la Guía AWS IoT Greengrass Version 2 para desarrolladores.

Simulador de fuente de datos IoT SiteWise OPC UA

El componente simulador de fuente de datos IoT SiteWise OPC UA (`aws.iot.SiteWiseEdgeOpcuaDataSourceSimulator`) proporciona la siguiente funcionalidad:

- Inicia un servidor OPC UA local que genera datos de muestra
- Simula una fuente de datos que el componente recopilador AWS IoT SiteWise OPC UA puede leer en una puerta de enlace AWS IoT SiteWise
- Permite la exploración de AWS IoT SiteWise características utilizando los datos de muestra generados

Este componente es particularmente útil para fines de prueba y desarrollo, ya que permite simular fuentes de datos industriales sin necesidad de equipo físico.

Para obtener más información sobre el componente de simulación de fuentes de datos, consulte el [simulador de fuentes de datos IoT SiteWise OPC UA](#) en la Guía para AWS IoT Greengrass Version 2 desarrolladores.

Estos AWS IoT Greengrass componentes funcionan para habilitar la funcionalidad SiteWise Edge. El SiteWise editor de IoT garantiza que los datos se envíen de forma fiable a la nube, el SiteWise procesador de IoT gestiona los cálculos locales y la optimización de los datos, y el recopilador IoT SiteWise OPC UA facilita la integración con los protocolos industriales habituales.

Note

Para utilizar estos componentes, debe disponer de sus dispositivos AWS IoT Greengrass V2 periféricos o instalarlos posteriormente en ellos. La configuración adecuada de cada componente es importante para un rendimiento óptimo de SiteWise Edge.

Filtre los activos en una puerta de enlace SiteWise Edge

Puede usar el filtrado perimetral para administrar sus activos de manera más eficiente enviando solo un subconjunto de activos a una puerta de enlace de SiteWise Edge específica para su uso en el procesamiento de datos. Si sus activos están organizados en forma de árbol o estructura principal secundaria, puede configurar una política de IAM asociada a la función de IAM de una puerta de enlace de SiteWise Edge que solo permita enviar la raíz del árbol, o principal, y sus elementos secundarios a una puerta de enlace de Edge específica. SiteWise

Note

Si está organizando los activos existentes en una estructura de árbol, después de crear la estructura, vaya a cada activo existente que haya agregado a la estructura y elija Editar y, a continuación, seleccione Guardar para asegurarse de que AWS IoT SiteWise reconoce la nueva estructura.

Configuración del filtrado de periferia

Para configurar el filtrado de borde en la puerta de enlace de SiteWise Edge, añada la siguiente política de IAM a la función de IAM de la puerta de enlace de SiteWise Edge y `<root-asset-id>` reemplácela por el ID del activo raíz que desee enviar a la puerta de enlace de SiteWise Edge.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "iotsitewise:DescribeAsset",
        "iotsitewise:ListAssociatedAssets"
      ],
      "Resource": "arn:aws:iotsitewise:*:*:asset/*",
      "Condition": {
        "StringNotLike": {
          "iotsitewise:assetHierarchyPath": "/<root-asset-id>*"
        }
      }
    }
  ]
}
```

Si actualmente hay activos en su puerta de enlace de SiteWise Edge que le gustaría eliminar, inicie sesión en su puerta de enlace de SiteWise Edge y ejecute el siguiente comando para forzar a la puerta de enlace de SiteWise Edge a sincronizarse con ellos AWS IoT SiteWise mediante la eliminación de la memoria caché.

```
sudo rm /greengrass/v2/work/aws.iot.SiteWiseEdgeProcessor/sync-app/
sync_resource_bundles/edge.json
```

Configura el soporte de proxy y administra los almacenes de confianza para AWS IoT SiteWise Edge

En AWS IoT SiteWise Edge, configure y administre los almacenes de confianza para configurar el soporte de proxy para sus dispositivos perimetrales. En primer lugar, configure el proxy y, a continuación, configure los almacenes de confianza. Puede configurar los almacenes de confianza durante la instalación de la puerta de enlace o manualmente una vez establecida la puerta de enlace.

- **Proxies:** facilitan la conectividad entre sus dispositivos y AWS servicios periféricos en varios entornos de red.

- Almacenes de confianza: garantice la seguridad de las conexiones mediante la administración de certificados de confianza. Las configuraciones adecuadas le ayudan a cumplir con las políticas de seguridad de la red, permiten la comunicación en entornos de red restringidos y optimizan la transferencia de datos entre los dispositivos periféricos y los servicios en la nube.

SiteWise Edge utiliza varios almacenes de confianza para diferentes tipos de componentes, lo que garantiza un flujo de datos seguro y eficiente desde los dispositivos periféricos a la nube. Puede configurar los almacenes de confianza y los proxies en una puerta de enlace existente o durante el proceso de instalación al crear una nueva puerta de enlace.

Requisitos para las configuraciones de almacenes de confianza y proxy

Antes de configurar un almacén de confianza o instalar SiteWise Edge con la configuración de proxy, asegúrese de cumplir los requisitos previos. Existen diversos requisitos de implementación según el uso de los componentes y los requisitos de funcionalidad.

Requisitos de soporte de proxy

- La URL de su servidor proxy. La URL debe incluir la información del usuario y el número de puerto del host. Por ejemplo, `scheme://[userinfo@]host[:port]`.
 - `scheme`— Debe ser HTTP o HTTPS
 - (Opcional)`userinfo`: información de nombre de usuario y contraseña
 - `host`— El nombre de host o la dirección IP del servidor proxy
 - `port`— El número de puerto
- Una lista de direcciones para evitar el proxy.
- (Opcional) El archivo de certificado de CA del proxy si utiliza un proxy HTTPS con un certificado autofirmado.

Requisitos del almacén de confianza

- Para que el paquete de procesamiento de datos funcione plenamente con el proxy HTTPS, debe actualizar los tres almacenes de confianza.
- Si solo usa el compilador SiteWise OPC UA de IoT y el SiteWise editor de IoT, actualice los almacenes de confianza AWS IoT Greengrass Core y Java de certificados a la última versión.

Prácticas recomendadas para las configuraciones perimetrales de servidores proxy y almacenes fiduciarios

Para un mantenimiento continuo y para mantener el más alto nivel de seguridad en su entorno perimetral:

- Revise y actualice periódicamente la configuración del proxy para adaptarla a los requisitos de seguridad de su red.
- Supervise la conectividad de la puerta de enlace y el flujo de datos para garantizar una comunicación proxy adecuada
- Mantenga y actualice los almacenes de confianza de acuerdo con las políticas de administración de certificados de su organización
- Puede implementar y seguir nuestras prácticas recomendadas para una comunicación segura en entornos periféricos, como:
 - Documente las configuraciones de su proxy y almacén de confianza para obtener visibilidad operativa
 - Siga las prácticas de seguridad de su organización para la administración de credenciales

Estas prácticas ayudan a mantener las operaciones seguras y confiables de sus puertas de enlace SiteWise Edge y, al mismo tiempo, se mantienen alineadas con sus políticas de seguridad más amplias.

Configure los ajustes del proxy durante la instalación de la puerta de enlace AWS IoT SiteWise Edge

Puede configurar AWS IoT SiteWise Edge para que funcione con un servidor proxy durante la instalación de la puerta de enlace. El script de instalación admite proxies HTTP y HTTPS y puede configurar automáticamente almacenes de confianza para conexiones proxy seguras.

Al ejecutar el script de instalación con la configuración del proxy, realiza varias tareas importantes:

- Valida el formato y los parámetros de la URL del proxy para garantizar que estén correctamente especificados.
- Descarga e instala las dependencias necesarias a través del proxy configurado.
- Si se proporciona un certificado de CA proxy, se adjunta al certificado de CA AWS IoT Greengrass raíz y se importa a Java. KeyStore

- Se configura AWS IoT Greengrass (lo que usa SiteWise Edge) para usar el proxy en todas las conexiones salientes.
- Completa la instalación de SiteWise Edge con las configuraciones de proxy y almacén de confianza adecuadas.

Para configurar los ajustes del proxy al instalar el software de puerta de enlace

1. Cree una puerta de enlace SiteWise Edge. Para obtener más información, consulte [Cree una puerta de enlace SiteWise Edge autohospedada](#) y [Instale el software AWS IoT SiteWise Edge Gateway en su dispositivo local](#).
2. Ejecute el script de instalación con la configuración de proxy adecuada para su entorno. Sustituya los marcadores de posición por su información de proxy específica

Sustituya cada uno de los siguientes elementos:

- `-p, --proxy-url` — La URL del servidor proxy. La URL debe ser una de las dos `httphttps`.
- `-n, --no-proxy` — Una lista de direcciones separadas por comas para evitar el proxy.
- (Opcional) `-c, --proxy-ca-cert` — Ruta al archivo de certificado de CA del proxy.
- (Opcional) `-j, --javastorepass` — La KeyStore contraseña de Java. La contraseña predeterminada es `changeit`.

Linux

Para los sistemas Linux, utilice la siguiente estructura de comandos:

```
sudo ./install.sh -p proxy-url -n no-proxy-addresses [-c proxy-ca-cert-path] [-j javastorepass]
```

Windows

Para Microsoft Windows los sistemas que utilizan PowerShell, utilice esta estructura de comandos:

```
.\install.ps1 -ProxyUrl proxy-url -NoProxyAddresses no-proxy-addresses [-ProxyCaCertPath proxy-ca-cert-path] [-JavaStorePass javastorepass]
```

Solución de problemas durante una instalación con proxy

Para obtener más información sobre cómo resolver los problemas del almacén de confianza relacionados con una puerta de enlace SiteWise Edge, consulte [Problemas de instalación con proxy](#).

Configure manualmente los almacenes de confianza para que sean compatibles con el proxy HTTPS en AWS IoT SiteWise Edge

Al configurar los componentes de AWS IoT SiteWise Edge para que se conecten a través de un proxy HTTPS, añada el certificado del servidor proxy a los almacenes de confianza correspondientes. SiteWise Edge usa varios almacenes de confianza para proteger las comunicaciones. Hay tres almacenes de confianza y el uso que haga de ellos depende del tipo de componente de SiteWise Edge que utilice en la implementación de la puerta de enlace.

Los almacenes de confianza se actualizan automáticamente durante el proceso de instalación cuando se proporciona la configuración del proxy.

- [Configure un almacén de confianza de componentes AWS IoT Greengrass principales](#)— El certificado AWS IoT Greengrass raíz de la CA se incluye en los almacenes de confianza para comprobar la autenticidad de AWS los servicios.

Este almacén de confianza ayuda a AWS IoT Greengrass los componentes a comunicarse de forma segura con AWS los servicios a través del proxy y, al mismo tiempo, verifica la autenticidad de esos servicios.

- [Configure un almacén de confianza de componentes basado en Java](#)— Java KeyStore (JKS) es el principal almacén de confianza que utilizan los componentes basados en Java para las conexiones. SSL/TLS

Las aplicaciones Java se basan en el JKS para establecer conexiones seguras. Por ejemplo, si utilizas el SiteWise editor de IoT o el recopilador SiteWise OPC UA de IoT, que están basados en Java, tendrás que configurar este almacén de confianza. Esto garantiza que estos componentes puedan comunicarse de forma segura a través del proxy HTTPS al enviar datos a la nube o recopilar datos de los servidores OPC UA.

- [Configuración del almacén de confianza de componentes a nivel del sistema](#)— Cuando se utilizan proxies HTTPS, sus certificados deben añadirse a los almacenes de confianza adecuados para permitir conexiones seguras.

Cuando se utilizan proxies HTTPS, sus certificados deben añadirse a los almacenes de confianza adecuados para permitir conexiones seguras. Esto es necesario porque los componentes

a nivel del sistema, que suelen estar escritos en lenguajes como Rust o Go, se basan en el almacén de confianza del sistema y no en el JKS de Java. Por ejemplo, si utilizas utilidades del sistema que necesitan comunicarse a través del proxy (como las actualizaciones de software o la sincronización horaria), tendrás que configurar el almacén de confianza a nivel del sistema. Esto garantiza que estos componentes y utilidades puedan establecer conexiones seguras a través del proxy.

Configure un almacén de confianza de componentes AWS IoT Greengrass principales

Para las funciones AWS IoT Greengrass principales que utilizan la CA raíz de Amazon:

1. Localice el archivo del certificado en `/greengrass/v2/AmazonRootCA1.pem`
2. Añada el certificado raíz del proxy HTTPS (autofirmado) a este archivo.

```
-----BEGIN CERTIFICATE-----
MIIEFTCCA v2gAwIQWgIVAMHSAzWG/5YVRYtRQ0xXUTEpHuEmApzGCSqGSIb3DQEK
\nCwUAhuL9MQswCQwJVUzEPMAVUzEYMBYGA1UECgwP1hem9uLmNvbSBJbmMuMRww
... content of proxy CA certificate ...
+vHIR1t0e5JAm5\noTIZGoFbK82A0/n07f/t5PSIDAim9V3Gc3pSXxCCAQoFYnui
GaPULGk1gCE84a0X\n7Rp/1ND/PuMZ/s8Yj1kY2NmYmNjMCAXDTE5MTEyN2cM216
gJMIADggEPADf2/m45hzEXAMPLE=
-----END CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIDQTCCAimgF6AwIBAgITBmyfz/5mjAo54vB4ikPm1jZKyjANJmApzyMZFo6qBg
ADA5MQswCQYDVQQGEwJVUzEPMA0tMVT8QtPHRh8jrdkGA1UEChMGRGV3ZQQDExBBKW
... content of root CA certificate ...
o/ufQJQWUCyziar1hem9uMRkwFwYVPSHCb2XV4cdFyQzR1K1dZwgJcIQ6XUDgHaa
5MsI+yMRQ+hDaXJioblDxgjUka642M4UwtBV8oK2xJNDd2ZhwLnOqdeXeGADKkpy
rqXRfKoQnoZsG4q5WTP46EXAMPLE
-----END CERTIFICATE-----
```

Configure el proxy HTTPS en una puerta de enlace establecida

Puede agregar soporte de proxy a una puerta de enlace establecida conectándose al puerto 443 en lugar del puerto 8883. Para obtener más información sobre el uso de un servidor proxy, consulte [Conectarse en el puerto 443 o mediante un proxy de red](#) en la Guía para AWS IoT Greengrass Version 2 desarrolladores. Si crea una nueva puerta de enlace, puede establecer la configuración

del proxy durante la instalación de la puerta de enlace. Para obtener más información, consulte [Configure los ajustes del proxy durante la instalación de la puerta de enlace AWS IoT SiteWise Edge](#).

Cuando utilizas un proxy HTTPS con AWS IoT Greengrass on SiteWise Edge, el software elige automáticamente entre HTTP y HTTPS para las conexiones proxy en función de la URL proporcionada.

 Important


Actualiza todos los almacenes de confianza necesarios antes de intentar conectarte a través de un proxy HTTPS.

Configure un almacén de confianza de componentes basado en Java

Para el SiteWise editor de IoT, el recopilador SiteWise OPC UA de IoT y los servicios Java del paquete de procesamiento de datos, la ubicación predeterminada del almacén de confianza de Java es `$JAVA_HOME/jre/lib/security/cacerts`

Para añadir un certificado

1. Cree un archivo para almacenar el certificado del servidor proxy, por ejemplo `proxy.crt`.

 Note

Cree el archivo con antelación utilizando el certificado del servidor proxy.

2. Añada el archivo al almacén de confianza de Java mediante el siguiente comando:

```
sudo keytool -import -alias proxyCert -keystore /usr/lib/jvm/java-11-openjdk-amd64/lib/security/cacerts -file proxy.crt
```

3. Cuando se le pida, utilice la contraseña predeterminada: `changeit`

Configuración del almacén de confianza de componentes a nivel del sistema

Para los componentes escritos en Rust, Go y otros lenguajes que utilizan el almacén de confianza del sistema:

Linux

Sistemas Linux: añada certificados a `/etc/ssl/certs/ca-certificates.crt`

Windows

Microsoft Windows sistemas: para configurar el almacén de confianza, siga el procedimiento del [almacén de certificados](#) de la documentación de Microsoft Ignite.

Windows ofrece varios almacenes de certificados, incluidos almacenes independientes para usuarios y ordenadores, cada uno con varios subalmacenes. Para la mayoría de las configuraciones de SiteWise Edge, recomendamos añadir certificados al almacén. `COMPUTER | Trusted Root Certification Authorities` Sin embargo, en función de sus requisitos específicos de configuración y seguridad, es posible que necesite utilizar un almacén diferente.

Solución de problemas con los almacenes de confianza

Para obtener más información sobre cómo resolver los problemas del almacén de confianza relacionados con una puerta de enlace de SiteWise Edge, consulte [Problemas con el almacén de confianza](#).

AWS IoT SiteWise APIs Úselo en el borde

AWS IoT SiteWise proporciona un subconjunto de ellos APIs, junto con otros específicos para cada entorno APIs, lo que permite una interacción fluida con los modelos de activos y sus activos asociados desplegados en la periferia. Estos modelos de activos deben configurarse para ejecutarse en la periferia. Para obtener más información, consulte instrucciones detalladas sobre este proceso de configuración en [Configure un modelo de activos para el procesamiento de datos en SiteWise Edge](#).

Después de configurar los APIs, puede recuperar datos completos sobre sus modelos de activos y activos individuales. La recuperación de información sobre modelos de activos, activos, paneles, portales y proyectos puede ayudarlo a supervisar los portales y paneles implementados y a acceder a los datos de los activos recopilados en la periferia. Esto proporciona un host central en la red para las interacciones AWS IoT SiteWise sin necesidad de una llamada a la API web.

Temas

- [Todos los dispositivos AWS IoT SiteWise Edge disponibles APIs](#)
- [Solo Edge APIs para su uso con dispositivos periféricos AWS IoT SiteWise](#)
- [Habilite CORS en Edge AWS IoT SiteWise APIs](#)

- [Configura los tiempos de espera de las sesiones para Edge AWS IoT SiteWise](#)
- [Tutorial: Enumere los modelos de activos en una puerta de enlace de AWS IoT SiteWise Edge](#)

Todos los dispositivos AWS IoT SiteWise Edge disponibles APIs

AWS IoT SiteWise ofrece una variedad de dispositivos periféricos APIs para que pueda completar las tareas de forma local en el dispositivo. Algunas de las ventajas disponibles APIs incluyen la recuperación de modelos de activos, la creación y actualización de las propiedades de los activos y el envío de flujos de datos a la nube. Al aprovecharlos APIs, puede crear soluciones que puedan funcionar en entornos con conectividad de red intermitente o limitada.

Disponible AWS IoT SiteWise APIs

Los siguientes AWS IoT SiteWise APIs están disponibles en los dispositivos periféricos:

- [ListAssetModels](#)
- [DescribeAssetModel](#)
- [ListAssets](#)
- [DescribeAsset](#)
- [DescribeAssetProperty](#)
- [ListAssociatedAssets](#)
- [GetAssetPropertyAggregates](#)
- [GetAssetPropertyValue](#)
- [GetAssetPropertyValueHistory](#)
- [ListDashboards](#)
- [ListPortals](#)
- [ListProjectAssets](#)
- [ListProjects](#)
- [DescribeDashboard](#)
- [DescribePortal](#)
- [DescribeProject](#)

Disponible solo en Edge APIs

Lo siguiente APIs se usa localmente en los dispositivos periféricos:

- [Autenticación](#): utilice esta API para obtener las credenciales temporales SigV4 que utilizará para realizar llamadas a la API.

Solo Edge APIs para su uso con dispositivos periféricos AWS IoT SiteWise

Además de los AWS IoT SiteWise APIs que están disponibles en el borde, hay otros específicos para el borde. Los específicos de los bordes se describen APIs a continuación.

Autenticación

Obtiene las credenciales de la puerta de enlace Edge SiteWise . Tendrá que añadir usuarios locales o conectarse a su sistema utilizando LDAP o un grupo de usuarios de Linux. Para obtener más información sobre cómo añadir usuarios, consulte [LDAP](#) o [Grupo de usuarios de Linux](#).

Sintaxis de la solicitud

```
POST /authenticate HTTP/1.1
Content-type: application/json
{
  "username": "string",
  "password": "string",
  "authMechanism": "string"
}
```

Parámetros de solicitud del URI

La solicitud no utiliza ningún parámetro de URI.

Cuerpo de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

nombre de usuario

El nombre de usuario utilizado para validar la llamada de solicitud.

Tipo: cadena

Obligatorio: sí

contraseña

La contraseña del usuario que solicita las credenciales.

Tipo: cadena

Obligatorio: sí

authMechanism

El método de autenticación para validar a este usuario en el host.

Tipo: cadena

Valores válidos: ldap, linux, winnt

Obligatorio: sí

Sintaxis de la respuesta

```
HTTP/1.1 200
Content-type: application/json
{
  "accessKeyId": "string",
  "secretAccessKey": "string",
  "sessionToken": "string",
  "region": "edge"
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

Los siguientes datos se devuelven en formato JSON.

accessKeyId

El ID de clave de acceso que identifica las credenciales de seguridad temporales.

Restricciones de longitud: longitud mínima de 16. Longitud máxima de 128.

Patrón: `[\w]*`

secretAccessKey

La clave de acceso secreta que se puede utilizar para firmar las solicitudes.

Tipo: cadena

sessionToken

El token que los usuarios deben pasar a la API de servicio para utilizar las credenciales temporales.

Tipo: cadena

region

La región a la que se dirigen las llamadas a la API.

Tipo: CONSTANT - edge

Errores

IllegalArgumentException

La solicitud ha sido rechazada porque el formato del cuerpo del documento proporcionado no era correcto. El mensaje de error describe el error específico.

Código de estado HTTP: 400

AccessDeniedException

El usuario no tiene credenciales válidas basadas en el proveedor de identidad actual. El mensaje de error describe el mecanismo de autenticación.

Código de estado HTTP: 403

TooManyRequestsException


La solicitud ha alcanzado su límite de intentos de autenticación. El mensaje de error contiene la cantidad de tiempo que debe esperarse hasta que se realicen nuevos intentos de autenticación.

Código de estado HTTP: 429

Habilite CORS en Edge AWS IoT SiteWise APIs

Al habilitar el CORS (intercambio de recursos entre orígenes) en AWS IoT SiteWise Edge APIs, las aplicaciones web se comunican directamente con ellos APIs en diferentes dominios. Esto permite una integración perfecta, el intercambio de datos en tiempo real y el acceso a los datos entre dominios sin servidores intermediarios ni soluciones alternativas. Los ajustes del CORS se pueden

configurar para especificar los orígenes permitidos, lo que garantiza un acceso controlado desde distintos orígenes.

 Note

CORS está disponible para la versión 3.3.1 y posteriores del componente. Esta función está disponible para la versión 3.3.1 y posteriores del componente.

`aws.iot.SiteWiseEdgeProcessor` Para obtener más información, consulte el [AWS IoT SiteWise procesador](#) en la AWS IoT Greengrass Version 2 Guía para desarrolladores.

Para habilitar CORS en Edge SiteWise APIs

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, seleccione Puertas de enlace Edge.
3. Seleccione la puerta de enlace SiteWise Edge para la que desee habilitar CORS. Puede habilitar CORS en el tipo de AWS IoT Greengrass V2 implementación.
4. En la sección de configuración de la puerta de enlace, elija el dispositivo principal de Greengrass asociado.
5. En la pestaña Implementaciones, en Dispositivos Greengrass, seleccione el enlace de implementación correspondiente.
6. En Acciones, elija Revisar y, a continuación, Revisar el despliegue.

 Important

Al crear una configuración revisada y habilitada para CORS, se reemplaza la configuración actual del dispositivo.

7. En el paso 1, especifique el destino, proporcione un nombre opcional para identificar la implementación.
8. En el paso 2, Seleccionar los componentes (opcional), puede dejar todas las selecciones actuales tal como están y elegir Siguiente.
9. En el paso 3, Configurar los componentes (opcional), seleccione `aws.iot.SiteWiseEdgeProcessor` elija Configurar componente.
10. En la sección Actualización de la configuración, en Configuración para fusionar, introduzca el siguiente JSON:

```
{  
  "AWS_SITWISE_EDGE_ACCESS_CONTROL_ALLOW_ORIGIN": "*" }  
}
```

Note

Si * se utiliza como valor para, se `AWS_SITWISE_EDGE_ACCESS_CONTROL_ALLOW_ORIGIN` permiten todos los orígenes. Para los entornos de producción, se recomienda especificar el origen exacto URLs para una mayor seguridad.

11. Elija Confirmar.
12. Seleccione Siguiente para continuar con los pasos restantes hasta llegar al paso 5, Revisar.
13. Revise los cambios de configuración y, a continuación, seleccione Deploy para aplicar los cambios a su puerta de enlace SiteWise Edge.

Note

Como alternativa, puede habilitar CORS configurando globalmente la variable `AWS_SITWISE_EDGE_ACCESS_CONTROL_ALLOW_ORIGIN` de entorno * en su AWS IoT SiteWise puerta de enlace.

Note


En el caso del proxy autenticado, `userinfo` debe incluirse en el `url` campo de la configuración del proxy `username` y `password` no como campos separados.

Una vez completada la implementación, CORS se habilita en la API de SiteWise Edge, lo que permite a los orígenes específicos realizar solicitudes de origen cruzado a la API.

Configura los tiempos de espera de las sesiones para Edge AWS IoT SiteWise

SiteWise Edge le permite configurar los tiempos de espera de sesión para la API de SiteWise Edge. Esta función mejora la seguridad al finalizar automáticamente las sesiones inactivas después de un


período de tiempo específico. Esta sección le guía a través del proceso de configuración del tiempo de espera de la sesión mediante el. Consola de AWS IoT SiteWise

 Note

La configuración del tiempo de espera de la sesión está disponible para la versión 3.4.0 y versiones posteriores del componente. `aws.iot.SiteWiseEdgeProcessor` Para obtener más información, consulte el [AWS IoT SiteWise procesador](#) en la Guía AWS IoT Greengrass Version 2 para desarrolladores.


Para configurar un tiempo de espera de sesión para una puerta de enlace SiteWise Edge

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, seleccione Puertas de enlace Edge.
3. Elija la puerta de enlace SiteWise Edge en la que desee configurar el tiempo de espera de la sesión.

 Note

Puede configurar el tiempo de espera de la sesión según el tipo de AWS IoT Greengrass V2 implementación.

4. En la sección de configuración de la puerta de enlace, elija el dispositivo principal de Greengrass asociado.
5. En la pestaña Implementaciones, en Dispositivos Greengrass, seleccione el enlace de implementación correspondiente.
6. En Acciones, elija Revisar. Lea la advertencia y, a continuación, seleccione Revisar el despliegue.

 Important

Al crear una configuración de tiempo de espera de sesión revisada, se reemplaza la configuración actual del dispositivo.

7. En el paso 1, especifique el destino, proporcione un nombre opcional para identificar la implementación revisada y, a continuación, elija Siguiente.

8. En el paso 2, Seleccionar los componentes (opcional), puede dejar todas las selecciones actuales tal como están y seleccionar Siguiente.
9. En el paso 3, Configurar los componentes (opcional), seleccione aws.iot.SiteWiseEdgeProcessory elija Configurar componente.
10. En la sección Actualización de la configuración, en Configuración para fusionar, introduzca el siguiente JSON:

```
{  
  "AWS_SITWISE_EDGE_SESSION_TIMEOUT_MINUTES": "240"  
}
```

11. Defina el valor `AWS_SITWISE_EDGE_SESSION_TIMEOUT_MINUTES` en minutos. Los valores de tiempo de espera de la sesión pueden oscilar entre 1 minuto y 10080 minutos (7 días). El valor predeterminado es 240 minutos (4 horas).
12. Elija Confirmar.
13. Seleccione Siguiente para continuar con los pasos restantes hasta llegar al paso 5, Revisar.
14. Revise los cambios de configuración y, a continuación, seleccione Deploy para aplicar los cambios a su puerta de enlace SiteWise Edge.

Note

Como alternativa, puede configurar el tiempo de espera de la sesión configurando la variable de entorno global `AWS_SITWISE_EDGE_SESSION_TIMEOUT_MINUTES` en el valor deseado (en minutos) en su puerta de enlace Edge. SiteWise

Una vez completada la implementación, la nueva configuración de tiempo de espera de la sesión se aplica a la API de Edge. SiteWise

Tutorial: Enumere los modelos de activos en una puerta de enlace de AWS IoT SiteWise Edge

Puede utilizar un subconjunto de los disponibles AWS IoT SiteWise APIs junto con los específicos de Edge APIs para interactuar con los modelos de activos y sus activos en Edge. Este tutorial le

explicará cómo obtener credenciales temporales para una puerta de enlace de AWS IoT SiteWise Edge y cómo obtener una lista de los modelos de activos de la puerta de enlace de Edge. SiteWise

Requisitos previos

En los pasos de este tutorial puede utilizar diversas herramientas. Para utilizar estas herramientas, asegúrese de tener instalados los requisitos previos correspondientes.

Necesitará lo siguiente para completar este tutorial:

- Un [AWS IoT SiteWise Requisitos de la puerta de enlace autohospedada de Edge](#) implementado y en ejecución
- Acceda a su puerta de enlace SiteWise Edge en la misma red a través del puerto 443.
- [OpenSSL](#) instalado
- (AWS OpsHub para AWS IoT SiteWise) La [AWS IoT SiteWise aplicación AWS OpsHub para](#)
- (curl) [curl](#) instalado
- (Python) [urllib3](#) instalado
- (Python) [Python3](#) instalado
- (Python) [Boto3](#) instalado
- (Python) [BotoCore](#) instalado

Paso 1: Obtenga un certificado firmado por el servicio SiteWise Edge Gateway

Para establecer una conexión TLS con la APIs disponible en la puerta de enlace SiteWise Edge, necesita un certificado de confianza. Puede generar este certificado mediante OpenSSL AWS OpsHub o for. AWS IoT SiteWise

OpenSSL

Note

Necesita tener instalado [OpenSSL](#) para ejecutar este comando.

Abre una terminal y ejecuta el siguiente comando para obtener un certificado firmado desde la puerta de enlace de SiteWise Edge. <sitewise_gateway_ip>Sustitúyala por la IP de la puerta de enlace SiteWise Edge.

```
openssl s_client -connect <sitewise_gateway_ip>:443 </dev/null 2>/dev/null | openssl
x509 -outform PEM > GatewayCert.pem
```

AWS OpsHub for AWS IoT SiteWise

Se puede usar AWS OpsHub para AWS IoT SiteWise. Para obtener más información, consulte [Administre las puertas de enlace SiteWise Edge](#).

En este tutorial se utiliza la ruta absoluta al certificado de puerta de enlace SiteWise Edge descargado. Ejecute el siguiente comando para exportar la ruta completa de su certificado, sustituyendo <absolute_path_to_certificate> por la ruta al certificado:

```
export PATH_TO_CERTIFICATE='<absolute_path_to_certificate>'
```

Paso 2: Obtenga el nombre de host de la puerta de enlace SiteWise Edge

Note

Necesita tener instalado [OpenSSL](#) para ejecutar este comando.

Para completar el tutorial, necesitará el nombre de host de su puerta de enlace SiteWise Edge. Para obtener el nombre de host de la puerta de enlace SiteWise Edge, ejecute lo siguiente y sustitúyalo por <sitewise_gateway_ip> la IP de la puerta de enlace SiteWise Edge:

```
openssl s_client -connect <sitewise_gateway_ip>:443 </dev/null 2>/dev/null | grep -Po
'CN = \K.*' | head -1
```


Ejecute el siguiente comando para exportar el nombre de host para usarlo más adelante y <your_edge_gateway_hostname> sustitúyalo por el nombre de host de su SiteWise puerta de enlace Edge:

```
export GATEWAY_HOSTNAME='<your_edge_gateway_hostname>'
```

Paso 3: Obtenga credenciales temporales para su SiteWise puerta de enlace Edge


Ahora que tiene el certificado firmado y el nombre de host de su puerta de enlace SiteWise Edge, necesita obtener credenciales temporales para poder ejecutar APIs en la puerta de enlace. Puede

obtener estas credenciales a través de la puerta AWS OpsHub de enlace SiteWise Edge AWS IoT SiteWise o directamente desde ella mediante APIs.

 Important

Las credenciales caducan cada 4 horas, por lo que debe obtenerlas justo antes de utilizarlas APIs en su puerta de enlace SiteWise Edge. No almacene las credenciales en caché durante más de 4 horas.

Obtenga credenciales temporales utilizando AWS OpsHub para AWS IoT SiteWise

 Note

Necesita tener instalada la [AWS IoT SiteWise aplicación AWS OpsHub for](#).

Para utilizar AWS IoT SiteWise la aplicación AWS OpsHub para obtener sus credenciales temporales, haga lo siguiente:

1. Inicie sesión en la aplicación.
2. Seleccione Configuración.
3. En Autenticación, elija Copiar credenciales.
4. Amplíe la opción que se adapte a su entorno y elija Copiar.
5. Guarde las credenciales para utilizarlas más tarde.

Obtenga credenciales temporales mediante la API de SiteWise Edge Gateway

Para usar la API de puerta de enlace de SiteWise Edge para obtener las credenciales temporales, puede usar un script de Python o un curl, primero necesitará tener un nombre de usuario y una contraseña para su puerta de enlace de SiteWise Edge. Las puertas de enlace SiteWise Edge utilizan la autenticación y la autorización SigV4. Para obtener más información sobre cómo añadir usuarios, consulte [LDAP](#) o [Grupo de usuarios de Linux](#). Estas credenciales se utilizarán en los siguientes pasos para obtener las credenciales locales de la puerta de enlace SiteWise Edge necesarias para utilizarla. AWS IoT SiteWise APIs

Python

Note

Necesita tener [urllib3](#) y [Python3](#) instalados.

Para obtener las credenciales mediante Python

1. Cree un archivo llamado `get_credentials.py` y copie en él el siguiente código.

```
...
The following demonstrates how to get the credentials from the SiteWise Edge
gateway. You will need to add local users or connect your system to LDAP/AD
https://docs.aws.amazon.com/iot-sitewise/latest/userguide/manage-gateways-
ggv2.html#create-user-pool

Example usage:
    python3 get_credentials.py -e https://<gateway_hostname> -c
    <path_to_certificate> -u '<gateway_username>' -p '<gateway_password>' -m
    '<method>'
...
import urllib3
import json
import urllib.parse
import sys
import os
import getopt

"""
This function retrieves the AWS IoT SiteWise Edge gateway credentials.
"""
def get_credentials(endpoint, certificatePath, user, password, method):
    http = urllib3.PoolManager(cert_reqs='CERT_REQUIRED', ca_certs=
    certificatePath)
    encoded_body = json.dumps({
        "username": user,
        "password": password,
        "authMechanism": method,
    })

    url = urllib.parse.urljoin(endpoint, "/authenticate")
```

```
response = http.request('POST', url,
                        headers={'Content-Type': 'application/json'},
                        body=encoded_body)

if response.status != 200:
    raise Exception(f'Failed to authenticate! Response status
{response.status}')

auth_data = json.loads(response.data.decode('utf-8'))

accessKeyId = auth_data["accessKeyId"]
secretAccessKey = auth_data["secretAccessKey"]
sessionToken = auth_data["sessionToken"]
region = "edge"

return accessKeyId, secretAccessKey, sessionToken, region

def print_help():
    print('Usage:')
    print(f'{os.path.basename(__file__)} -e <endpoint> -c <path/to/certificate>
-u <user> -p <password> -m <method> -a <alias>')
    print('')
    print('-e, --endpoint    edge gateway endpoint. Usually the Edge gateway
hostname.')
    print('-c, --cert_path path to downloaded gateway certificate')
    print('-u, --user        Edge user')
    print('-p, --password    Edge password')
    print('-m, --method      (Optional) Authentication method (linux, winnt,
ldap), default is linux')
    sys.exit()

def parse_args(argv):
    endpoint = ""
    certificatePath = None
    user = None
    password = None
    method = "linux"

    try:
        opts, args = getopt.getopt(argv, "he:c:u:p:m:",
["endpoint=", "cert_path=", "user=", "password=", "method="])
    except getopt.GetoptError:
        print_help()
```

```
for opt, arg in opts:
    if opt == '-h':
        print_help()
    elif opt in ("-e", "--endpoint"):
        endpoint = arg
    elif opt in ("-u", "--user"):
        user = arg
    elif opt in ("-p", "--password"):
        password = arg
    elif opt in ("-m", "--method"):
        method = arg.lower()
    elif opt in ("-c", "--cert_path"):
        certificatePath = arg

if method not in ['ldap', 'linux', 'winnt']:
    print("not valid method parameter, required are ldap, linux, winnt")
    print_help()

if (user == None or password == None):
    print("To authenticate against edge user, password have to be passed
together, and the region has to be set to 'edge'")
    print_help()

if(endpoint == ""):
    print("You must provide a valid and reachable gateway hostname")
    print_help()

return endpoint,certificatePath, user, password, method

def main(argv):
    # get the command line args
    endpoint, certificatePath, user, password, method = parse_args(argv)

    accessKeyId, secretAccessKey, sessionToken, region=get_credentials(endpoint,
certificatePath, user, password, method)

    print("Copy and paste the following credentials into the shell, they are
valid for 4 hours:")
    print(f"export AWS_ACCESS_KEY_ID={accessKeyId}")
    print(f"export AWS_SECRET_ACCESS_KEY={secretAccessKey}")
    print(f"export AWS_SESSION_TOKEN={sessionToken}")
    print(f"export AWS_REGION={region}")
```


```
print()
```

```
if __name__ == "__main__":  
    main(sys.argv[1:])
```

2. Ejecute `get_credentials.py` desde el terminal sustituyendo `<gateway_username>` y `<gateway_password>` por las credenciales que ha creado.

```
python3 get_credentials.py -e https://$GATEWAY_HOSTNAME -c $PATH_TO_CERTIFICATE  
-u '<gateway_username>' -p '<gateway_password>' -m 'linux'
```

curl

 Note

Necesita tener [curl](#) instalado.

Para obtener las credenciales mediante curl

1. Ejecute el siguiente comando desde el terminal sustituyendo `<gateway_username>` y `<gateway_password>` por las credenciales que ha creado.

```
curl --cacert $PATH_TO_CERTIFICATE --location \  
-X POST https://$GATEWAY_HOSTNAME:443/authenticate \  
--header 'Content-Type: application/json' \  
--data-raw '{  
    "username": "<gateway_username>",  
    "password": "<gateway_password>",  
    "authMechanism": "linux"  
}'
```

La respuesta debe ser similar a la siguiente:

```
{  
    "username": "sweuser",  
    "accessKeyId": "<accessKeyId>",
```

```
"secretAccessKey": "<secretAccessKey>",
"sessionToken": "<sessionToken>",
"sessionExpiryTime": "2022-11-17T04:51:40.927095Z",
"authMechanism": "linux",
"role": "edge-user"
}
```


2. Ejecute el siguiente comando desde el terminal.

```
export AWS_ACCESS_KEY_ID=<accessKeyId>
export AWS_SECRET_ACCESS_KEY=<secretAccessKey>
export AWS_SESSION_TOKEN=<sessionToken>
export AWS_REGION=edge
```

Paso 4: Obtenga una lista de los modelos de activos de la puerta de enlace SiteWise Edge

Ahora que tiene un certificado firmado, el nombre de host de la puerta de enlace SiteWise Edge y las credenciales temporales de la puerta de enlace SiteWise Edge, puede usar la `ListAssetModels` API para obtener una lista de los modelos de activos de la puerta de enlace SiteWise Edge.

Python

 Note

Necesitas tener instalado [Python3](#) y [Boto3](#). [BotoCore](#)

Para obtener la lista de modelos de activos mediante Python

1. Cree un archivo llamado `list_asset_model.py` y copie el siguiente código en él.

```
import json
import boto3
import botocore
import os

# create the client using the credentials
client = boto3.client("iotsitewise",
    endpoint_url= "https://" + os.getenv("GATEWAY_HOSTNAME"),
    region_name=os.getenv("AWS_REGION"),
    aws_access_key_id=os.getenv("AWS_ACCESS_KEY_ID"),
```

```
aws_secret_access_key=os.getenv("AWS_SECRET_ACCESS_KEY"),
aws_session_token=os.getenv("AWS_SESSION_TOKEN"),
verify=os.getenv("PATH_TO_CERTIFICATE"),
config=botocore.config.Config(inject_host_prefix=False))

# call the api using local credentials
response = client.list_asset_models()
print(response)
```

2. Ejecute `list_asset_model.py` desde el terminal.

```
python3 list_asset_model.py
```

curl

Note

Necesita tener [curl](#) instalado.

Para obtener la lista de modelos de activos mediante curl

Ejecute el siguiente comando desde el terminal.

```
curl \
  --request GET https://$GATEWAY_HOSTNAME:443/asset-models \
  --cacert $PATH_TO_CERTIFICATE \
  --aws-sigv4 "aws:amz:edge:iotsitewise" \
  --user "$AWS_ACCESS_KEY_ID:$AWS_SECRET_ACCESS_KEY" \
  -H "x-amz-security-token:$AWS_SESSION_TOKEN"
```

La respuesta debe ser similar a la siguiente:

```
{
  "assetModelSummaries": [
    {
      "arn": "arn:aws:iotsitewise:{region}:{account-id}:asset-model/{asset-
model-id}",
      "creationDate": 1.669245291E9,
      "description": "This is a small example asset model",
```

```
    "id": "{asset-model-id}",
    "lastUpdateDate": 1.669249038E9,
    "name": "Some Metrics Model",
    "status": {
      "error": null,
      "state": "ACTIVE"
    }
  },
  .
  .
  .
],
"nextToken": null
}
```

Aloje una puerta de enlace SiteWise Edge en Siemens Industrial Edge

Aloje su puerta de enlace en Siemens Industrial Edge mediante la aplicación AWS IoT SiteWise Edge. Al igual que con AWS IoT Greengrass V2 ella, puede optimizar los procesos de fabricación o mejorar los flujos de trabajo operativos con el SiteWise Edge onSiemens Industrial Edge.

Puede transferir los datos de su dispositivo Siemens Industrial Edge a su AWS cuenta ejecutando una puerta de enlace SiteWise Edge en el dispositivo. Para ello, solicite acceso a la aplicación AWS IoT SiteWise Edge al equipo de soporte de SiteWise Edge. A continuación, cree un recurso de puerta de enlace SiteWise Edge con un objetivo de despliegue del dispositivo Siemens Industrial Edge (nuevo). A continuación, descargue el archivo de configuración y cárguelo en su aplicación a través del portal de Siemens Industrial Edge Management. Para obtener más información sobre la ejecución de aplicaciones en Siemens Industrial Edge, incluida la configuración de los recursos de Siemens necesarios, consulte [What is Industrial Edge?](#) en la documentación de Siemens.

Note

Siemens no es un vendedor o proveedor de SiteWise Edge. The Siemens Industrial Edge Marketplace es un mercado independiente.

Temas

- [Seguridad](#)
- [Siemens Secure Storagey la aplicación AWS IoT SiteWise Edge](#)
- [Destinos para Siemens Industrial Edge dispositivos](#)
- [Migración desde la aplicación de vista previa](#)
- [Resolución de problemas](#)
- [AWS IoT SiteWise Registro de cambios de la aplicación Edge](#)
- [Requisitos para la aplicación AWS IoT SiteWise Edge](#)
- [Creación de una puerta de enlace para Siemens Industrial Edge](#)
- [Creación de un Siemens Databus user para la aplicación](#)
- [Acceda a la aplicación AWS IoT SiteWise Edge](#)
- [Instale la aplicación en un Siemens dispositivo](#)
- [Actualice la configuración de la aplicación AWS IoT SiteWise Edge](#)
- [AWS IoT SiteWise — Datos generados por el uso de este servicio](#)

Seguridad

Como parte del [modelo de responsabilidad compartida](#) entre AWS nuestros clientes y nuestros socios, a continuación se describe quién es responsable de los diferentes aspectos de la seguridad:

Responsabilidad del cliente

- Investigar al socio.
- Configurar el acceso a la red otorgado al socio.
- Proteger físicamente el dispositivo en el que se ejecuta SiteWise Edge.

AWS responsabilidad

- Aislar al socio de los recursos de la AWS nube del cliente.

Responsabilidad del socio

- Uso de valores predeterminados seguros.
- Mantener la solución segura en el tiempo mediante parches y otras actualizaciones apropiadas.
- Mantener la confidencialidad de los datos de los clientes.
- Examinar otras aplicaciones disponibles en el mercado de socios.

Siemens Secure Storagey la aplicación AWS IoT SiteWise Edge

Para proteger las credenciales y los secretos necesarios para ejecutar la aplicación AWS IoT SiteWise Edge, Siemens Industrial Edge proporciona mecanismos para almacenar de forma segura las credenciales en el dispositivo. La aplicación AWS IoT SiteWise Edge no se ejecutará en un dispositivo si este no admite el almacenamiento seguro de estas credenciales. Los errores de ejecución causados por la falta de compatibilidad con almacenamiento seguro se registran en los archivos de registro.

Se requieren las siguientes versiones mínimas del sistema operativo para instalar y ejecutar la aplicación AWS IoT SiteWise Edge. Actualice sus dispositivos a las versiones más recientes para instalar la aplicación.

- Para dispositivos virtuales: IEVD versión 1.19 o superior
- Para dispositivos físicos: IED-OS versión 2.2 o superior

La aplicación AWS IoT SiteWise Edge no Siemens Industrial Edge se ejecutará hasta que haya actualizado el dispositivo.

Destinos para Siemens Industrial Edge dispositivos

Cuando se utiliza la aplicación AWS IoT SiteWise EdgeSiemens Industrial Edge, los destinos ayudan a preparar los datos antes de enviarlos AWS IoT SiteWise para su posterior análisis y distribución. Puede configurar los ajustes de destino de los datos para la ingesta de datos en búfer mediante Amazon S3 o utilizar la ingesta de datos en tiempo real. Ambos le permiten suscribirse a los temas de MQTT mediante filtros de ruta en el destino de despliegue del dispositivo. Siemens Industrial Edge

El objetivo Siemens Industrial Edge de despliegue de su puerta de enlace admite dos métodos principales de gestión de datos:

- AWS IoT SiteWise configuración en tiempo real: los datos se envían directamente a AWS IoT SiteWise medida que se recopilan
- AWS IoT SiteWise almacenados en búfer mediante la configuración de Amazon S3: los datos se recopilan y almacenan temporalmente en lotes antes de enviarlos a Amazon S3

Para obtener más información sobre la configuración de estas opciones, consulte y. [Añadir un destino almacenado en AWS IoT SiteWise búfer mediante Amazon S3](#) [Agregue un destino de AWS IoT SiteWise Edge en tiempo real](#)

Prefijos para filtros de ruta

Los filtros de ruta para las puertas de enlace que utilizan objetivos de Siemens Industrial Edge despliegue combinan el tema y el nombre del flujo de datos para crear un identificador único para los datos. El tema combinado con el nombre del flujo de datos se denomina prefijo en Siemens Industrial Edge las pasarelas. Esto difiere de las pasarelas autohospedadas, en las que los filtros de ruta se basan únicamente en temas de MQTT.

Example Estructura de filtros de rutas para flujos de datos de Siemens

Un filtro de ruta típico para un flujo de datos de Siemens incluye tanto la ruta del tema como el nombre del flujo de datos:

```
ie/d/device1/application1/datastream1
```

Donde:

- `ie/d/` es el prefijo obligatorio para los flujos de datos de Siemens
- `device1/application1` representa la ruta jerárquica
- `datastream1` es el nombre del flujo de datos específico

Note

Cuando trabaje con flujos de Siemens Industrial Edge datos, asegúrese de incluir los temas de metadatos (`ie/m/`) y datos (`ie/d/`) en los filtros de ruta para recibir información completa sobre los flujos de datos.

Filtros de destinos y rutas

Consulte los siguientes temas para obtener más información sobre los filtros de destinos y rutas en las pasarelas habilitadas para MQTT:

- [Comprenda los destinos de AWS IoT SiteWise Edge](#)
- [Agregue un destino de AWS IoT SiteWise Edge en tiempo real](#)

- [Añadir un destino almacenado en AWS IoT SiteWise búfer mediante Amazon S3](#)
- [Conozca los filtros de ruta para los destinos de AWS IoT SiteWise Edge](#)
- [Agregue filtros de ruta a los destinos de AWS IoT SiteWise Edge](#)
- [Administre los destinos de AWS IoT SiteWise Edge](#)

Migración desde la aplicación de vista previa

Si utilizaste SiteWise Edge Siemens Industrial Edge durante la fase de vista previa, tendrás que actualizar de la versión preliminar, la versión 1.0.1, a la versión más reciente. Haga lo siguiente para migrar:

1. Crea nuevas puertas de enlace SiteWise Edge. Para obtener más información, consulte [Creación de una puerta de enlace para Siemens Industrial Edge](#).
2. Cree un nuevo Siemens Databus user para cada nueva puerta de enlace. Para obtener más información, consulte [Creación de un Siemens Databus user para la aplicación](#).
3. Desinstale la versión 1.0.1 de la aplicación AWS IoT SiteWise Edge gateway en su IED.

Note

Prepárese para las interrupciones en el flujo de datos mientras reconfigura los AWS IoT SiteWise activos que utilizaba anteriormente la versión preliminar de la AWS IoT SiteWise aplicación Edge. Si bien se conserva el historial de datos, existe la posibilidad de que se pierdan datos al volver a instalar la nueva puerta de enlace.

4. Elimine las pasarelas de SiteWise Edge que creó durante la vista previa en [Consola de AWS IoT SiteWise](#)
5. Instale la aplicación AWS IoT SiteWise Edge Gateway en el IED mediante el nuevo archivo de configuración de puerta de enlace. Para obtener más información, consulte [Instale la aplicación en un Siemens dispositivo](#).

Important

La instalación de la nueva puerta de enlace sobrescribe la versión preliminar de la aplicación SiteWise Edge. No es posible volver a la versión 1.0.1 después de instalar la versión actual.

Tras configurar la nueva puerta de enlace y Siemens Databus user, los datos fluirán a sus propiedades.

También puedes actualizar tu aplicación SiteWise Edge directamente desde la versión. Sin embargo, sigue siendo necesaria una nueva configuración de la puerta de enlace.

Resolución de problemas

Para solucionar los problemas de la puerta de enlace SiteWise Edge de su Siemens Industrial Edge dispositivo, consulte [Solución de problemas de la aplicación AWS IoT SiteWise Edge en Siemens Industrial Edge](#).

También puede acceder a [AWS re:post](#) para buscar respuestas a sus preguntas.

AWS IoT SiteWise Registro de cambios de la aplicación Edge

En la siguiente tabla se describen los cambios en cada versión de la aplicación AWS IoT SiteWise Edge.

Versión	Cambios	
3.0.1	<p>Mejoras y correcciones de errores</p> <ul style="list-style-type: none"> • Aplica el máximo de alias únicos por archivo de ingesta AWS IoT SiteWise almacenado en búfer para igualar el límite de nubes de 10 000. • Soluciona un problema que provocaba la ingesta de datos duplicada. • Soluciona un problema relacionado con la sincronización de grandes configuraciones. • Resuelve los retrasos en la inicialización de los puntos de control que anteriormente afectaban al proceso de inicio. 	
3.0.0	<p>Nuevas características</p> <ul style="list-style-type: none"> • Añade opciones de configuración adicionales en la AWS IoT SiteWise consola para las puertas de enlace instaladas en: Siemens Industrial Edge <ul style="list-style-type: none"> • Añada y configure los destinos de Amazon S3. 	

Versión	Cambios	
	<ul style="list-style-type: none"> • Elimine el destino AWS IoT SiteWise en tiempo real. • Añada filtros para cada destino para controlar el enrutamiento de los datos. • Añade la opción de una configuración de prefijo global para enviar todos los datos del SiteWise editor de IoT al Siemens IE instalador. El prefijo se aplica después del filtrado. <p>Mejoras y correcciones de errores</p> <ul style="list-style-type: none"> • Añade la posibilidad de volver a intentar las conexiones al bus de datos Siemens Industrial Edge si se produce un error en el intento inicial de la aplicación AWS IoT SiteWise Edge. 	
2.0.1	<p>Mejoras y correcciones de errores</p> <ul style="list-style-type: none"> • Soluciona un problema por el que la aplicación entraba en estado de error y se cerraba si no podía recuperar AWS las credenciales al iniciarse. Añade la posibilidad de volver a intentarlo hasta que la recuperación de las credenciales se realice correctamente. 	

Versión	Cambios	
2.0.0	<ul style="list-style-type: none"> • La aplicación AWS IoT SiteWise Edge ya está disponible de forma general. • La aplicación requiere Siemens IEVD versión 1.19 o Siemens IED-OS versión 2.2. • Mejoras de rendimiento: reducción del uso de memoria y CPU. • Mejoras en la depuración: ahora puede cargar un archivo de configuración opcional para habilitar los registros de depuración. • Mejoras de seguridad: la aplicación utiliza SecureStorage API para almacenar de forma segura las credenciales en el dispositivo. • Dockervalue resumido: sha256:4a960f29234a190ebb5224c1fd0f3e99faafccc4cb3d93ca13fef247b6656d18 	
1.0.1	Versión inicial	

Requisitos para la aplicación AWS IoT SiteWise Edge

Para ejecutar AWS IoT SiteWise Edge onSiemens Industrial Edge, necesitas lo siguiente:

- Una cuenta de [Siemens Digital Exchange Platform](#).
- Una cuenta de Siemens Industrial Edge Hub (iehub).
- Una instancia de Siemens Industrial Edge Management.
 - El servicio de configuración de aplicaciones IE. Para obtener más información, consulte [Installing the IE App Configuration Service manually](#) en la documentación de Siemens Industrial Edge Management.
- Acceso a la versión 2.0.1 o superior de la aplicación AWS IoT SiteWise Edge. Para obtener más información, consulte [Acceda a la aplicación AWS IoT SiteWise Edge](#).
- Un dispositivo Siemens Industrial Edge (IED) o un dispositivo virtual Siemens Industrial Edge (IEVD).

- Un mínimo de 15 GB de espacio en disco para los requisitos de hardware.
- 1 GB de RAM con 1 GB adicional de memoria de intercambio.
- Configuración del dispositivo para permitir tráfico saliente en los puertos 443 y 8883.
- Un procesador de x86-64 bits.
- Siemens Industrial Edge Management versión 1.13.10 o superior.
- Conformidad del dispositivo con los requisitos de Siemens Secure Storage.
 - En dispositivos virtuales, IEVD versión 1.19 o superior.
 - Para dispositivos físicos, IED-OS versión 2.2 o superior.
- La versión más reciente de la Docker Compose.
- Docker Engine versión 18.091 o superior.
- Acceso al dominio obligatorio. Para obtener más información, consulte [AWS IoT SiteWise puntos finales](#).

Creación de una puerta de enlace para Siemens Industrial Edge

Una vez que tenga las cuentas de Siemens y las instancias de IEM adecuadas, puede crear una pasarela SiteWise Edge de tipo dispositivo Siemens Industrial Edge de tipo despliegue.

Note

Asegúrese de cumplir todos los requisitos para ejecutar un dispositivo en Siemens Industrial Edge Management. Para obtener más información, consulte [Requisitos para la aplicación AWS IoT SiteWise Edge](#).

Creación del archivo de configuración

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, seleccione Puertas de enlace Edge.
3. Seleccione Crear puerta de enlace.
4. Para el tipo de implementación, elija el dispositivo Siemens Industrial Edge.
5. Introduzca un nombre para su puerta de enlace SiteWise Edge o utilice el nombre generado por AWS IoT SiteWise.
6. (Opcional) En Configuración avanzada, haga lo siguiente:

- Introduce un nombre para tu AWS IoT Core Thing o usa el nombre generado por AWS IoT SiteWise.
7. Seleccione Crear puerta de enlace.
 8. En el cuadro de diálogo Generar el archivo de configuración de la puerta de enlace SiteWise Edge, seleccione Generar y descargar. AWS IoT SiteWise genera automáticamente un archivo de configuración que utilizará para configurar la aplicación AWS IoT SiteWise Edge.

Important

Conserve el archivo de configuración de la puerta de enlace como copia de seguridad en caso de que necesite restaurar la instancia de la aplicación AWS IoT SiteWise Edge. [AWS Secrets Manager](#) Para ello, puede guardar de forma segura el archivo de configuración de la puerta de enlace SiteWise Edge. Secrets Manager almacena, administra y recupera información confidencial de forma segura. Si pierdes o eliminas este archivo de configuración, no podrás volver a conectar la instancia de la aplicación AWS IoT SiteWise Edge a su puerta de enlace original si necesitas recuperarlo. Deberás crear una nueva puerta de enlace y un nuevo archivo de configuración.

Creación de un Siemens Databus user para la aplicación

AWS IoT SiteWise Edge on Siemens Industrial Edge ingiere los datos de la Siemens Databus aplicación. Para conectar SiteWise Edge a Siemens Databus, necesitas una Siemens Databus user que te dé acceso a los datos a los que deseas transferirlos AWS IoT SiteWise de forma segura. Para empezar, cree una aplicación SiteWise Edge Siemens Databus user y, a continuación, proporcione las credenciales a la misma.

Para crear una Siemens Databus user

1. En su instancia de Siemens Industrial Edge Management, elija Administración de Edge en la sección Aplicaciones de plataforma.
2. Elija el icono Conexiones de datos.
3. Seleccione Databus. Aparece una lista de sus dispositivos conectados.
4. Seleccione el dispositivo para conectarse a la aplicación AWS IoT SiteWise Edge.
5. Elija Iniciar. Aparece Databus Configurator correspondiente al dispositivo seleccionado.

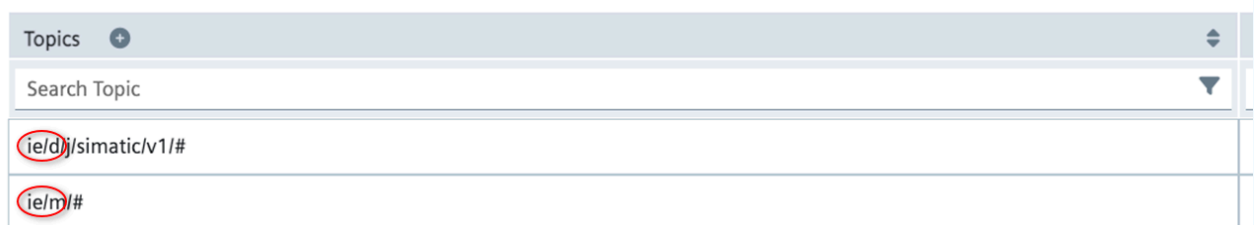
6. Cree un usuario para su dispositivo Edge en Usuarios. Para obtener más información sobre la creación de un usuario, consulte [Usuarios](#) en la documentación de Siemens Industrial Edge Management.
7. Seleccione los temas a los que este Siemens Databus debería tener acceso. Estos temas restringen a qué puede acceder AWS IoT SiteWise Edge.

⚠ Important

Todos los temas a los que Siemens Databus user tiene acceso se publican en AWS IoT SiteWise.

ℹ Note

Los Siemens Databus user necesitan acceso a los temas de datos y metadatos. Los temas que comienzan por `ie/d` son temas de datos. Y los temas que comienzan por `ie/m` son temas de metadatos. Comparta los temas por parejas para que SiteWise Edge tenga acceso a los datos y metadatos de cada tema respectivo.



8. Establezca los permisos adecuados para su configuración de Siemens Databus.

Tras crear Siemens Databus la configuración, puede instalar la aplicación AWS IoT SiteWise Edge en su Siemens Industrial Edge Management. Para obtener más información, consulte [Instale la aplicación en un Siemens dispositivo](#).

Si lo desea, también puede configurar filtros de destinos y rutas para su Siemens Industrial Edge puerta de enlace. Para obtener más información, consulte [Filtros de destinos y rutas](#).

Acceda a la aplicación AWS IoT SiteWise Edge

Para acceder a la aplicación AWS IoT SiteWise Edge en Siemens Industrial Edge, [envíe un correo electrónico](#) solicitando el acceso al equipo de soporte de SiteWise Edge.

En el mensaje, incluya la información siguiente:

- Su nombre e información de contacto
- Nombre de la empresa
- ID de inquilino de Siemens Industrial Edge

Instale la aplicación en un Siemens dispositivo

Después de acceder a la aplicación AWS IoT SiteWise Edge enviando un correo electrónico al equipo de soporte de SiteWise EdgeSiemens Industrial Edge, asigne la aplicación a una instancia deSiemens Industrial Edge Management. A continuación, puede instalar la aplicación AWS IoT SiteWise Edge en su dispositivo.

Para instalar la aplicación AWS IoT SiteWise Edge

1. Compruebe que el Docker resumen incluido en él Siemens Industrial Edge Management coincide con la última versión que aparece en el [AWS IoT SiteWise Registro de cambios de la aplicación Edge](#).

Para obtener más información sobre cómo encontrar el valor de Docker resumen de Siemens, consulte la Siemens documentación sobre cómo [administrar una aplicación](#) en el Siemens Industrial Edgedispositivo.

Siemens Industrial Edge Managementadmite una versión de la aplicación AWS IoT SiteWise Edge a la vez. Realice este paso para asegurarse de que está utilizando la última versión de la aplicación antes de instalar la aplicación AWS IoT SiteWise Edge en su Siemens Industrial Edge dispositivo.

2. Asigne la aplicación AWS IoT SiteWise Edge a Siemens Industrial Edge Management. Para obtener más información, consulte [Managing an app](#) en la sección Industrial Edge Management de la documentación de Siemens.
3. En Edge Management, explore el catálogo de AWS IoT SiteWise Edge y selecciónela.
4. Elija Instalar.

Note

Si aparece un botón de contacto, elíjalo y sigue los pasos para solicitar el acceso a la aplicación AWS IoT SiteWise Edge en ellaSiemens Industrial Edge. Para obtener más información, consulte [Acceda a la aplicación AWS IoT SiteWise Edge](#).

5. Seleccione Databus_Configuration en las opciones de Configuración del esquema.
6. Introduzca el Nombre de usuario y la Contraseña de la configuración de Databus. Para obtener más información sobre la creación de un Siemens Databus user, consulte [Creación de un Siemens Databus user para la aplicación](#).
7. Seleccione el icono de marca de verificación pequeño y redondo de color gris situado junto a Databus_Configuration para cambiar el color del icono a verde.

Note

Las configuraciones de entrada solo se aplican si el icono de la marca de verificación cambia de gris a verde. De lo contrario, se ignora la configuración de entrada.

Databus_Configuration Invalid Configuration

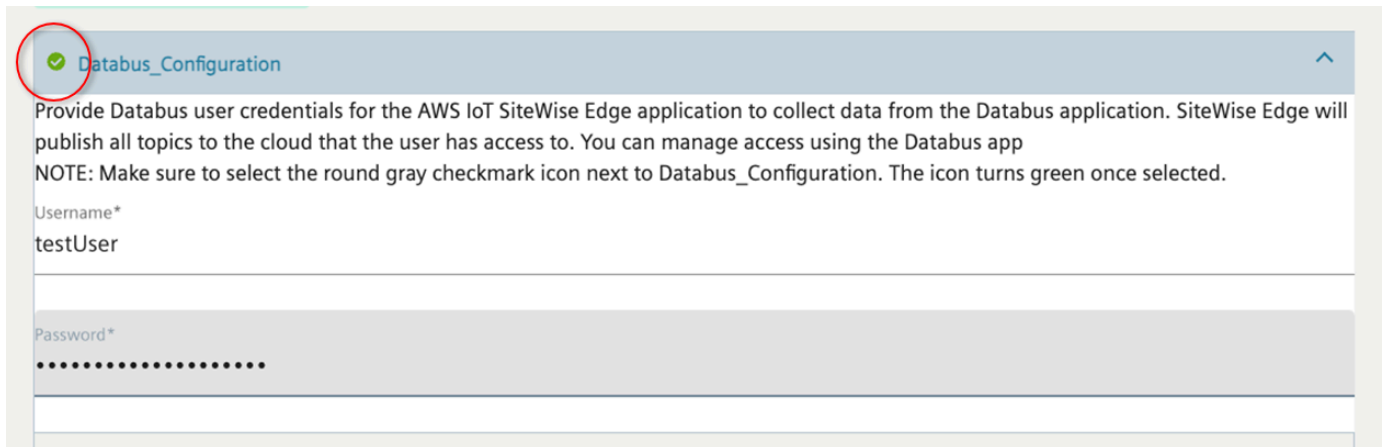
Provide Databus user credentials for the AWS IoT SiteWise Edge application to collect data from the Databus application. SiteWise Edge will publish all topics to the cloud that the user has access to. You can manage access using the Databus app
NOTE: Make sure to select the round gray checkmark icon next to Databus_Configuration. The icon turns green once selected.

Username*

is a required property

Password*

is a required property



Provide Databus user credentials for the AWS IoT SiteWise Edge application to collect data from the Databus application. SiteWise Edge will publish all topics to the cloud that the user has access to. You can manage access using the Databus app

NOTE: Make sure to select the round gray checkmark icon next to Databus_Configuration. The icon turns green once selected.

Username*

testUser

Password*

.....

8. Seleccione Siguiente para pasar a Otras configuraciones, donde podrá cargar el archivo de configuración de la puerta de enlace.
9. Elija SiteWise_Edge_Gateway_Config como ubicación para cargar el archivo de configuración de la puerta de enlace.

Note

Asegúrese de elegir `_Edge_Gateway_Config` en lugar de `_Edge_Support_Config_Optional`. SiteWise SiteWise

10. Seleccione el dispositivo para instalar la aplicación.
11. Seleccione Install Now (Instalar ahora).

Si lo desea, puede configurar el componente de publicación para exportar datos a la nube. AWS Para obtener más información, consulte [configurar el componente de publicador de AWS IoT SiteWise](#).

Para configurar los destinos de su Siemens Industrial Edge puerta de enlace, consulte [Filtros de destinos y rutas](#).

Actualice la configuración de la aplicación AWS IoT SiteWise Edge

Hay algunos aspectos que se deben tener en cuenta al actualizar la configuración de una aplicación AWS IoT SiteWise Edge a Siemens Industrial Edge.

Note

Cualquier cambio en la configuración de la aplicación AWS IoT SiteWise Edge requiere el reinicio de la aplicación.

Motivos para reiniciar la aplicación AWS IoT SiteWise Edge

- Una novedad Siemens Databus user para la aplicación AWS IoT SiteWise Edge.
- Un cambio en el archivo de configuración de la puerta de enlace (su archivo SiteWise_Edge_Gateway_Config).
- Una actualización de la configuración del proxy (que también requiere un reinicio completo del IEVD)
- Habilitación de registros de depuración para solucionar problemas

Reinicio de la aplicación

1. En su instancia de Siemens Industrial Edge Management, elija Administración de Edge en la sección Aplicaciones de plataforma.
2. Seleccione Mis aplicaciones instaladas.
3. Seleccione la aplicación Edge. AWS IoT SiteWise
4. Elija Reiniciar.

AWS IoT SiteWise — Datos generados por el uso de este servicio

AWS IoT SiteWise on Siemens Industrial Edge extiende las capacidades de la nube a los entornos industriales periféricos, lo que permite el procesamiento, el análisis y la toma de decisiones de datos a nivel local. SiteWise Edge se integra con AWS IoT SiteWise otros AWS servicios para proporcionar AWS IoT soluciones industriales integrales.

Tipos de datos

SiteWise La aplicación Edge en Siemens Industrial Edge genera datos sobre el rendimiento, el uso y las interacciones de las aplicaciones con otros AWS servicios, específicamente Amazon S3.

Volumen y recopilación de datos

La cantidad de datos generados varía en función de cómo utilice la aplicación y los servicios.

Almacenamiento de datos

Los datos de su aplicación se almacenan de forma segura en AWS los servidores. Se almacenan en formatos legibles por máquina.

Acceso a los datos

Puede acceder a sus datos a través de su cuenta. AWS Puede descargar una copia de sus datos siguiendo las instrucciones que se indican en esta guía del usuario de [SiteWise Edge on Siemens](#). Para obtener los datos de configuración de la aplicación, puede seguir las instrucciones de esta guía del usuario [Query Industrial Data](#). Encontrará más instrucciones para el acceso y la exportación masivos de datos en [Running Bulk Operations Export](#).

Administración de datos

Para obtener más información sobre las prácticas de datos de sus aplicaciones, consulte nuestras [condiciones de servicio](#), el [aviso de privacidad](#) y la [documentación del servicio](#), que incluye materiales sobre cómo administrar sus datos.

Eliminación de datos

Para obtener información sobre las opciones de retención y eliminación de datos, visite estas páginas de la guía del usuario: [administre el almacenamiento de datos](#), [elimine los flujos](#) de datos y [elimine modelos y activos](#).

Intercambio de datos con otras personas

Puede autorizar a terceros a acceder a sus AWS recursos a través de nuestros procesos de [gestión de identidad y acceso](#). AWS comparte datos personales con terceros en los casos limitados incluidos en el [Aviso AWS de privacidad](#).

¿Necesitas ayuda?

Visite [Customer Support](#) para comunicarse con nuestro equipo de soporte. Esto se entiende sin perjuicio de su derecho a presentar una reclamación en virtud de la legislación aplicable.

Titular de los datos

Amazon Web Services EMEA SARL, 38 Avenue John F. Kennedy, L-1855, Luxemburgo

Filtros de destinos y rutas

Los destinos en AWS IoT SiteWise Edge proporcionan una forma flexible y eficiente de gestionar la forma en que los datos industriales fluyen desde los dispositivos periféricos a la nube. En esta sección se explica cómo configurar los destinos, usar filtros de ruta para enrutar flujos de datos específicos y elegir el tipo de destino adecuado para su caso de uso.

Puede usar filtros de destinos y rutas en las puertas de enlace V3 autohospedadas y habilitadas para MQTT y las que se utilizan junto con la aplicación Edge alojada en ellas. AWS IoT SiteWise Siemens Industrial Edge Los filtros de destinos y rutas no funcionan con las pasarelas Classic Streams y V2.

Temas

- [Comprenda los destinos de AWS IoT SiteWise Edge](#)
- [Conozca los filtros de ruta para los destinos de AWS IoT SiteWise Edge](#)
- [Agregue un destino de AWS IoT SiteWise Edge en tiempo real](#)
- [Añadir un destino almacenado en AWS IoT SiteWise búfer mediante Amazon S3](#)
- [Agregue filtros de ruta a los destinos de AWS IoT SiteWise Edge](#)
- [Administre los destinos de AWS IoT SiteWise Edge](#)

Comprenda los destinos de AWS IoT SiteWise Edge

Utilice los destinos de AWS IoT SiteWise Edge para determinar dónde enviar los datos de origen. Puede elegir el destino de sus datos en función de las características específicas que necesite, como la rentabilidad, la baja latencia o los requisitos de almacenamiento. Integre los datos de los dispositivos capturados por AWS IoT SiteWise nuestros socios o aplicaciones personalizadas para publicar y suscribirse a los filtros de rutas (temas) periféricos. A continuación, puede modelar, transferir y almacenar los datos de su dispositivo en la nube.

Note

Para aprovechar al máximo todas las funciones de destino en las pasarelas autohospedadas, actualice a las versiones más recientes del SiteWise editor de IoT y del recopilador SiteWise OPC UA de IoT. La compatibilidad con las transmisiones clásicas continúa en las pasarelas V2 para mantener la compatibilidad con las configuraciones existentes. Para obtener más información, consulte [Transmisiones clásicas, pasarelas V2 para Edge AWS IoT SiteWise](#)

Temas

- [Cómo los destinos SiteWise Edge mejoran la gestión de datos](#)
- [Tipos de destino](#)
- [Compare la funcionalidad de destino entre las versiones de puerta de enlace](#)
- [Limitaciones de destino](#)
- [Casos de uso para destinos de SiteWise Edge](#)

Cómo los destinos SiteWise Edge mejoran la gestión de datos

Exporte datos desde la periferia a AWS IoT SiteWise en tiempo real o por lotes con Amazon S3.

Los destinos mejoran la flexibilidad y la escalabilidad de su entorno. AWS IoT SiteWise Los destinos implementan un modelo de administración de datos centralizado, en el que las fuentes publican los datos en un sistema central. Los destinos determinan dónde se envían los datos mediante filtros de ruta. Los destinos pueden suscribirse a varios filtros de ruta.

Las pasarelas habilitadas para MQTT, ya sean autohospedadas o en ejecución Siemens Industrial Edge, utilizan MQTT para la comunicación local e incluyen un destino en tiempo real predeterminado con filtros configurados en. # Esto significa que, de forma predeterminada, todos los mensajes sobre todos los temas se publican en el destino en tiempo real. AWS IoT SiteWise Para obtener más información, consulte [Conozca los filtros de ruta para los destinos de AWS IoT SiteWise Edge](#). Puede añadir un destino en tiempo real en cada puerta de enlace.

Tipos de destino

Al configurar un destino para su puerta de enlace, tiene dos opciones principales: configuración en tiempo real mediante AWS IoT SiteWise Amazon S3 y configuración en búfer. Cada tipo de destino tiene su propio conjunto de ajustes y consideraciones.

AWS IoT SiteWise ajustes en tiempo real

Elija esta opción para enviar los datos directamente al almacenamiento de AWS IoT SiteWise nivel avanzado para facilitar la ingesta y el monitoreo de los datos en tiempo real. La configuración en tiempo real gestiona el flujo de datos, especialmente cuando una puerta de enlace tiene problemas de conectividad con la nube. Durante la pérdida de la conexión, los datos se almacenan temporalmente de forma local en la puerta de enlace. Una vez que se restablece la conexión, los datos almacenados se envían automáticamente a la nube.

Puede ajustar varios aspectos del proceso de publicación de datos, como la cantidad máxima de datos que se deben almacenar localmente, la velocidad a la que se envían los datos a la nube al volver a conectarse y cuándo se eliminan los datos una vez que el almacenamiento alcanza su capacidad máxima.

Para obtener más información sobre los niveles AWS IoT SiteWise de almacenamiento, consulte [Gestione el almacenamiento de datos en AWS IoT SiteWise](#).

AWS IoT SiteWise almacenado en búfer mediante la configuración de Amazon S3

Este tipo de destino le permite almacenar datos en búfer local en la puerta de enlace y enviarlos periódicamente a un bucket de Amazon S3 por lotes. Los datos se almacenan en el eficiente formato Parquet, que está optimizado para las cargas de trabajo analíticas. Una vez que los datos estén en Amazon S3, podrá importarlos AWS IoT SiteWise para almacenarlos, procesarlos y analizarlos.

Elija esta opción para ingerir datos en lotes y almacenar los datos históricos de forma rentable. Puede configurar la ubicación del bucket de Amazon S3 que prefiera y la frecuencia con la que desea que se carguen los datos en Amazon S3. También puede elegir qué hacer con los datos después de su ingesta en AWS IoT SiteWise. Puede elegir que los datos estén disponibles tanto SiteWise en Amazon S3 como en Amazon S3 o puede optar por eliminarlos automáticamente de Amazon S3.

Compare la funcionalidad de destino entre las versiones de puerta de enlace

La función de destinos de las pasarelas habilitadas para MQTT agiliza la gestión del flujo de datos. Los destinos simplifican la administración de datos mediante la configuración centralizada del enrutamiento de datos a varios puntos finales. Este enfoque elimina la necesidad de configuraciones complejas de flujos individuales, lo que hace que el sistema en general sea más flexible y fácil de administrar.

En comparación, SiteWise Edge, la pasarela Classic Streams, V2, transmite los datos de las fuentes de datos a los editores mediante AWS IoT Greengrass transmisiones, configurando los destinos de los datos de forma individual para cada fuente de datos.

Con la función de AWS IoT SiteWise destino, se consolida la configuración de enrutamiento del editor. La configuración de destinos le permite gestionar los destinos y los filtros de rutas de forma centralizada. Puede añadir fácilmente un destino, gestionar los filtros de ruta y eliminar filtros o destinos innecesarios, según sus necesidades.

Además, la función de destinos utiliza MQTT (Message Queuing Telemetry Transport), un protocolo estándar del sector que se utiliza ampliamente en aplicaciones industriales de IoT. La adopción del MQTT ayuda AWS IoT SiteWise a facilitar la integración con varios dispositivos y sistemas.

Limitaciones de destino

Las limitaciones actuales para los destinos de las pasarelas SiteWise Edge incluyen:

- El paquete de procesamiento de datos no es compatible con las pasarelas habilitadas para MQTT.
- La compatibilidad con los tipos de datos se limita a los tipos de datos. AWS IoT SiteWise Para obtener información sobre cómo habilitar la conversión de tipos de datos, consulte [Conversión de tipos de datos no admitidos](#).

Casos de uso para destinos de SiteWise Edge

SiteWise Los destinos periféricos se utilizan en diversas aplicaciones. Estos son algunos ejemplos clave:

Automatización industrial, Supervisión en tiempo real y mantenimiento predictivo

En entornos industriales, los sensores y dispositivos de la fábrica pueden publicar datos en SiteWise Edge. Los destinos se pueden configurar para filtrar y enrutar los datos relevantes, lo que permite monitorear y analizar el rendimiento de la máquina en tiempo real. Puede suscribirse a los temas relevantes de MQTT mediante filtros de ruta, procesar los datos y, a continuación, publicar los datos procesados. De esta forma, puede enrutar de forma selectiva los datos procesados a servicios analíticos AWS en la nube o a sistemas locales. De este modo, los fabricantes pueden implementar estrategias de mantenimiento predictivo, optimizar los procesos de producción y reducir el tiempo de inactividad.

Edificios inteligentes, Eficiencia energética y optimización de la ocupación

Los sistemas de automatización de edificios generan flujos de datos para monitorear y controlar varios aspectos de un edificio, como los sistemas de climatización, la iluminación y el control de acceso. Con SiteWise Edge, estos flujos de datos se pueden ingerir, procesar y enrutar a diferentes destinos. Los administradores de las instalaciones pueden configurar los destinos para filtrar y reenviar los datos relevantes, lo que permite funciones avanzadas, como las medidas de eficiencia energética y la optimización de la ocupación, al tiempo que garantizan la privacidad de los datos y el cumplimiento.

Estos casos de uso demuestran cómo se puede aprovechar la función de destinos de SiteWise Edge en varios sectores para ingerir, procesar y enrutar datos de manera eficiente. Esto permite funciones avanzadas, como la supervisión en tiempo real, el mantenimiento predictivo, la eficiencia energética y el diagnóstico remoto, al tiempo que garantiza la privacidad de los datos y el cumplimiento.

Conozca los filtros de ruta para los destinos de AWS IoT SiteWise Edge

Temas

- [Requisitos de filtro de ruta](#)
- [Prácticas recomendadas para los filtros de rutas](#)
- [Filtros de ruta para servidores OPC UA](#)
- [Caracteres especiales en los nombres de los filtros de ruta](#)

Cada destino está configurado para enrutar los datos a Amazon S3 AWS IoT SiteWise o a Amazon S3. Los filtros de ruta le permiten seleccionar datos específicos para filtrarlos al recibir mensajes MQTT para un destino. Los filtros de ruta representan los nombres lógicos de sus flujos de datos y actúan como suscripciones a los temas de MQTT que desee.

En MQTT, los datos se organizan en temas, que son cadenas jerárquicas separadas por barras diagonales (/). Por ejemplo, un dispositivo podría publicar datos de temperatura en el tema. `home/livingroom/sensor1/temperature` Aquí, `home/livingroom/sensor1` representa la ruta o el nombre lógico del sensor y `temperature` es el tipo de datos que se va a publicar.

Puede usar filtros de ruta para suscribirse a temas específicos o a una variedad de temas mediante caracteres comodín (+y#). El + comodín coincide con un único nivel de la jerarquía de temas. Por ejemplo, `home/+sensor1/temperature` coincidiría con `home/livingroom/sensor1/temperature` y `home/bedroom/sensor1/temperature` El # comodín, cuando se usa al final de un filtro, coincide con varios niveles.

También puede utilizar una variedad de caracteres que normalmente no están permitidos en la especificación MQTT dentro del nombre de un filtro de ruta. Estos caracteres no funcionan como caracteres comodín cuando se utilizan dentro de un nombre. AWS IoT SiteWise convierte estos caracteres mediante codificación para garantizar el cumplimiento de MQTT y, al mismo tiempo, conservar la estructura de nomenclatura original. Esta función es especialmente útil para adaptarse a las convenciones de nomenclatura existentes en otros sistemas. Para obtener más información, consulte [Caracteres especiales en los nombres de los filtros de ruta](#).

Al seleccionar cuidadosamente los filtros de ruta adecuados, puede controlar qué datos se envían a un destino específico. Adapte el flujo de datos a los requisitos de su sistema de IoT mediante filtros de ruta.

Requisitos de filtro de ruta

Al introducir filtros de ruta mediante el Consola de AWS IoT SiteWise, tenga en cuenta lo siguiente:

- Los filtros de ruta están delimitados por una nueva línea, y cada línea representa un filtro de ruta independiente.
- Los filtros de ruta individuales pueden tener entre 1 y 65.535 bytes.
- Un filtro de ruta no puede estar en blanco.
- No se permiten valores nulos (U+0000).
- Puede introducir hasta 100 filtros de ruta o 65 535 caracteres a la vez, el límite que se alcance primero.
- El límite total es de 20 000 filtros de ruta para todos los destinos de una pasarela juntos.
- Puede utilizar los \$ caracteres%, #+, y en los nombres de los filtros de ruta, pero los convierte AWS IoT SiteWise automáticamente en codificación URI.

Prácticas recomendadas para los filtros de rutas

Al crear filtros de ruta para sus AWS IoT SiteWise destinos, tenga en cuenta las siguientes estrategias para gestionar sus datos de forma eficaz.

- Estructura los filtros para que reflejen la jerarquía de dispositivos. Por ejemplo, en un entorno de fabricación `factory/+/machine/#`, captura datos de todas las máquinas de diferentes líneas de producción.
- Utilice niveles específicos para los tipos, ubicaciones o funciones de los dispositivos. Por ejemplo, `factory/assembly-line/robot/temperature`. O, en la agricultura inteligente `farm/+ /crop/+ /moisture`, para monitorear los niveles de humedad de varios cultivos en diferentes campos.
- Aprovecha los comodines de forma estratégica: + úsalos para las variaciones de un solo nivel y # para capturar todos los niveles subsiguientes. Por ejemplo `building/+ /+ /energy-consumption`, rastrea el consumo de energía en diferentes zonas y pisos de un edificio. Esto supone que la primera + captura todos los pisos y la segunda + captura todas las zonas.

- Equilibre la especificidad y la flexibilidad mediante la creación de filtros que sean lo suficientemente específicos como para capturar datos relevantes, pero lo suficientemente flexibles como para adaptarse a futuros cambios. Por ejemplo, `site/+ /equipment-type/+ /measurement` permite añadir nuevos sitios o tipos de equipos sin cambiar la estructura del filtro.

Pruebe sus filtros minuciosamente para asegurarse de que capturan los datos previstos y se alinean con la arquitectura y los objetivos de su sistema de IoT.

Filtros de ruta para servidores OPC UA

En el caso de los servidores OPC UA, los filtros de ruta deben corresponder a los nombres de las etiquetas OPC UA. El nivel final del filtro de ruta debe coincidir exactamente con el nombre de la etiqueta OPC UA. Por ejemplo, si tu etiqueta OPC UA lo es `Device1.Temperature`, es posible que tu filtro de ruta lo sea `factory/line1/Device1.Temperature`. Puede utilizar caracteres comodín en los niveles anteriores, por ejemplo, `factory/+ /Device1.Temperature` para capturar la etiqueta en varias líneas de producción. Si tiene caracteres especiales en los nombres de los filtros de ruta, consulte [Caracteres especiales en los nombres de los filtros de ruta](#) para obtener más información.

Caracteres especiales en los nombres de los filtros de ruta

AWS IoT SiteWise admite caracteres que se utilizan habitualmente en protocolos industriales como OPC UA, que normalmente no están permitidos en los nombres de temas estándar de MQTT. Esta función facilita una integración más fluida de los sistemas industriales con las arquitecturas basadas en MQTT.

Note

Si bien nuestro manejo de caracteres especiales es útil para la integración y la migración, se recomienda seguir las convenciones de nomenclatura estándar de MQTT para las nuevas implementaciones siempre que sea posible, a fin de garantizar una compatibilidad más amplia.

Al recibir datos de fuentes industriales, AWS IoT SiteWise normaliza los nombres de los temas mediante la codificación URI para los caracteres especiales:

- `%` se convierte en `%25` (codificado primero como el carácter de escape)

- # se convertirá en %23
- + se convertirá en %2B
- \$se convierte %24 (solo cuando está al principio de un tema)

Esta codificación garantiza que los datos de origen que contienen estos caracteres especiales de MQTT se puedan utilizar de forma segura como nombres de temas de MQTT, conservando al mismo tiempo las convenciones de nomenclatura industrial originales.

Example: Caracteres especiales en los nombres de los filtros de rutas

Estos son algunos ejemplos de cómo pueden aparecer los nombres de temas industriales en los filtros de AWS IoT SiteWise rutas:

- Factory1/Line#2/Sensor+3 se convertirá en Factory1/Line%232/Sensor%2B3
- Plant%A/Unit\$1/Temp se convertirá en Plant%25A/Unit%241/Temp
- Site1/#Section/+Node se convertirá en Site1/%23Section/%2BNode

Al crear suscripciones o consultar los nombres de los temas AWS IoT SiteWise, verás las versiones originales sin codificar. La codificación se gestiona automáticamente para garantizar la conformidad con el MQTT.

Agregue un destino de AWS IoT SiteWise Edge en tiempo real

El tipo de destino en tiempo real le permite transmitir datos de IoT directamente desde sus dispositivos y pasarelas al AWS IoT SiteWise almacenamiento en tiempo real. Esta opción es ideal para casos de uso que requieren la ingesta y el procesamiento inmediatos de los datos a medida que se generan, sin necesidad de almacenarlos en lotes ni almacenarlos en búfer. Solo puede configurar un destino en tiempo real en cada puerta de enlace, ya que transmite datos de forma continua. AWS IoT SiteWise

Note

Si se duplica, se TQVs puede duplicar la carga.

Para añadir un destino en tiempo real

Utilice la AWS IoT SiteWise consola o añada un destino en tiempo real AWS CLI a su puerta de enlace habilitada para MQTT de SiteWise Edge.

Console

1. Abra la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, seleccione Puertas de enlace Edge.
3. Seleccione la puerta de enlace a la que desee añadir un destino.
4. En la sección Destinos, selecciona Añadir destino.
5. En la página Añadir destino, introduce los detalles del destino:
 - a. Un nombre para el destino en el campo Nombre del destino.
 - b. Selecciona el AWS IoT SiteWise tiempo real para el tipo de destino.
6. Configure el orden de publicación de la pasarela configurando el orden de publicación para Publicar primero los datos más antiguos o Publicar primero los datos más recientes. De forma predeterminada, la puerta de enlace publica primero los datos más antiguos.
7. Utilice el tiempo máximo de espera por lotes para establecer un tiempo máximo de espera para el publicador antes de enviar un lote de datos AWS IoT SiteWise. Esta configuración se aplica a cada alias. Los datos se almacenan localmente hasta que:
 - Ha transcurrido el tiempo establecido, o
 - Se reciben 10 entradas time-quality-value (TQV) para el alias

La condición que se cumpla primero hace que el lote se envíe a la nube.

8. Para comprimir los datos cargados, active la casilla Activar la compresión al cargar datos. Dejar que la puerta de enlace comprima los datos antes de subirlos a la nube reduce el uso de ancho de banda.
9. Para filtrar los datos del editor caducados, selecciona la casilla Excluir los datos caducados. Esta selección solo envía los datos activos y actuales a AWS IoT SiteWise.
10. En el campo Período límite, introduce la frecuencia con la que los datos deben considerarse caducados en tu conjunto de datos. Puedes determinar si los datos se cuentan en minutos o en días. El período límite mínimo es de cinco minutos. El período límite máximo es de siete días.
11. Si lo desea, configure los ajustes de almacenamiento local:

- a. Establezca la frecuencia del período de retención: la cantidad de tiempo que la puerta de enlace almacena localmente los datos que son anteriores al período límite. El período mínimo de retención es de un minuto.

El período máximo de retención es de 30 días y es superior o igual al período de rotación.

- b. Establezca el período de rotación: el intervalo de tiempo que se debe especificar al guardar datos anteriores al período límite de un solo archivo. La puerta de enlace transfiere un lote de datos al siguiente directorio local al final de cada periodo de rotación: `/greengrass/v2/work/aws.iot.SiteWiseEdgePublisher/exports`.

La retención debe ser superior a un minuto e igual al período de retención.

- c. Indique el valor de la capacidad de almacenamiento (GB) para establecer el tamaño máximo de los datos almacenados localmente en GB. Si los datos superan el tamaño máximo de almacenamiento local determinado, la puerta de enlace comienza a eliminar primero los datos más antiguos. La puerta de enlace continúa eliminando hasta que el tamaño de los datos almacenados localmente sea igual o inferior a la cuota.

La capacidad de almacenamiento debe ser superior o igual a un GB.

12. Añada filtros de ruta a su destino. Para obtener más información, consulte, [Agregue filtros de ruta a los destinos de AWS IoT SiteWise Edge](#).

Para obtener más información, consulte [Tipos de destino](#).

AWS CLI

Example: Crea un nuevo destino AWS IoT SiteWise en tiempo real

Utilice la [UpdateGatewayCapabilityConfiguration](#) API para configurar el publicador.

Establezca el parámetro `capabilityNamespace` como `iotsitewise:publisher:3`.

```
{
  "sources": [
    {
      "type": "MQTT"
    }
  ],
  "destinations": [
```

```

{
  "type": "SITEWISE_REALTIME",
  "name": "your-destination-name",
  "config": {
    "publishingOrder": "TIME_ORDER",
    "enableCompression": true,
    "maxBatchWaitTime": "10s"
  },
  "filters": [
    {
      "type": "PATH",
      "config": {
        "paths": [
          "#"
        ]
      }
    }
  ]
}

```

Para actualizar un destino AWS IoT SiteWise en tiempo real existente, utilice primero la `DescribeGatewayCapabilityConfiguration` API para encontrar el `destinationId`.

Example: Actualiza un destino AWS IoT SiteWise en tiempo real

Utilice la [UpdateGatewayCapabilityConfiguration](#) API para configurar el publicador.

Establezca el parámetro `capabilityNamespace` como `iotsitewise:publisher:3`.

```

{
  "sources": [
    {
      "type": "MQTT"
    }
  ],
  "destinations": [
    {
      "id": "your-existing-destination-id",
      "type": "SITEWISE_REALTIME",
      "name": "your-destination-name",
      "config": {

```

```

    "publishingOrder": "TIME_ORDER",
    "enableCompression": true,
    "dropPolicy": {
      "cutoffAge": "7d",
      "exportPolicy": {
        "retentionPeriod": "7d",
        "rotationPeriod": "6h",
        "exportSizeLimitGB": 10
      }
    },
    "maxBatchWaitTime": "10s"
  },
  "filters": [
    {
      "type": "PATH",
      "config": {
        "paths": [
          "#"
        ]
      }
    }
  ]
}

```

Las siguientes opciones de configuración son específicas de las puertas de enlace que utilizan el `iotsitewise:publisher:3` espacio de nombres.

sources

Define las fuentes de datos a las que se transferirán los datos de su equipo industrial. AWS IoT SiteWise Para las pasarelas habilitadas para MQTT, utilice. MQTT

Tipo: matriz de objetos

Obligatorio: sí

destinations

Define dónde enviar los datos. Los destinos se transmiten en tiempo real o se almacenan en búfer mediante Amazon S3. Se requiere al menos un objeto de destino, pero puede añadir una matriz vacía. Puede tener un destino en tiempo real para cada puerta de enlace. Para obtener más información, consulte [Comprenda los destinos de AWS IoT SiteWise Edge](#).

Tipo: matriz de objetos

Obligatorio: sí

`id`

El identificador único del destino. Puede proporcionar un identificador de destino existente o dejarlo en blanco. Si no especificas un ID, se generará un UUID de forma predeterminada.

Tipo: cadena

Requerido: no

`type`

Tipo de destino. Las opciones incluyen: `SITWISE_REALTIME` y `SITWISE_BUFFERED`

- `SITWISE_REALTIME`— Envíe los datos directamente al AWS IoT SiteWise almacenamiento en tiempo real.
- `SITWISE_BUFFERED`— Envíe los datos a Amazon S3 en lotes en formato Parquet y, a continuación, impórtelos al AWS IoT SiteWise almacenamiento.

Tipo: cadena

Obligatorio: sí

`name`

Un nombre único para el destino.

Tipo: cadena

Obligatorio: sí

`config`

Configuración específica para el tipo de destino en formato JSON. La configuración varía entre destinos en tiempo real y en búfer.

Tipo: objeto

Obligatorio: sí

Orden de publicación

Determina el orden en el que se publican los datos. Los datos se publican en función de su marca de tiempo. Las opciones incluyen `TIME_ORDER` y `RECENT_DATA`

- `TIME_ORDER`(predeterminado): publica primero los datos más antiguos.
- `RECENT_DATA`— Publica primero los datos más recientes.

Tipo: cadena

Requerido: no

Habilite la compresión

Si se establece en `true`, habilita la compresión de datos antes de enviarlos a AWS IoT SiteWise. Dejar que la puerta de enlace comprima los datos antes de subirlos a la nube reduce el uso de ancho de banda. El valor predeterminado es `true`.

Tipo: Booleano

Obligatorio: no

Política de abandono

Define cómo gestionar los datos antiguos.

Tipo: objeto

Obligatorio: no

- `cutoffAge`

La antigüedad máxima de los datos que se van a publicar se especifica en días, horas y minutos. Por ejemplo, `7d` o `1d7h16m`. Los datos anteriores a los especificados no se envían a AWS IoT SiteWise.

Los datos anteriores al período límite no se publican en la nube. La antigüedad límite debe ser de entre cinco minutos y siete días.

Puede utilizar `m`, `h` y `d` cuando especifique una edad límite. Tenga en cuenta que `m` representa minutos, `h` representa horas y `d` representa días.

Tipo: cadena

Obligatorio: sí

- `exportPolicy`

Define cómo gestionar los datos que superan la edad límite.

Tipo: objeto

Obligatorio: no

- `retentionPeriod`

La puerta de enlace SiteWise Edge elimina del almacenamiento local cualquier dato periférico anterior al período límite una vez que se haya almacenado durante el período de retención especificado. El periodo de retención debe ser de entre un minuto y 30 días, y mayor o igual que el periodo de rotación.

Puede utilizar m, h y d cuando especifique un periodo de retención. Tenga en cuenta que m representa minutos, h representa horas y d representa días.

Tipo: cadena

Requerido: no

- `rotationPeriod`

El intervalo de tiempo en que los datos se procesan por lotes y se guardan en un único archivo que es anterior al periodo límite. La puerta de enlace SiteWise Edge transfiere un lote de datos al siguiente directorio local al final de cada período de rotación: `./greengrass/v2/work/aws.iot.SiteWiseEdgePublisher/exports` El periodo de rotación debe ser mayor que un minuto y menor o igual que el periodo de retención.

Puede utilizar m, h y d cuando especifique un periodo de rotación. Tenga en cuenta que m representa minutos, h representa horas y d representa días.

Tipo: cadena

Requerido: no

- `exportSizeLimitGB`

El tamaño máximo permitido de datos almacenados localmente, en GB. Si se supera esta cuota, la puerta de enlace SiteWise Edge comienza a eliminar los primeros datos hasta que el tamaño de los datos almacenados localmente sea igual o inferior a la cuota. El valor de este parámetro debe ser mayor o igual que 1.

Tipo: entero

Obligatorio: no

`maxBatchWaitTime`

Establece un tiempo máximo que debe esperar el publicador antes de enviar un lote de datos a AWS IoT SiteWise. Esta configuración se aplica a cada alias. Los datos se almacenan localmente hasta que:

- Ha transcurrido el tiempo establecido, o
- Se reciben 10 entradas time-quality-value (TQV) para el alias

Utilice mh, y d para especificar una hora límite. Tenga en cuenta que m representa minutos, h representa horas y d representa días.

Tipo: cadena

Requerido: no

`filters`

Filtros que se van a aplicar a los datos. Se requiere al menos un filtro.

Tipo: cadena

Obligatorio: sí

`type`

Tipo de filtro. Utilice PATH.

Tipo: cadena

Obligatorio: sí

`config`

Configuración específica del tipo de filtro en formato JSON. Se requiere al menos un objeto, pero la matriz puede estar vacía.

Tipo: objeto

Obligatorio: sí

- `paths`

Conjunto de filtros de ruta. Para obtener más información, consulte [Conozca los filtros de ruta para los destinos de AWS IoT SiteWise Edge](#). La ruta predeterminada es #.

Tipo: matriz de cadenas

Obligatorio: sí

Añadir un destino almacenado en AWS IoT SiteWise búfer mediante Amazon S3

El tipo de destino almacenado en búfer te permite ahorrar en costes de ingesta AWS IoT SiteWise si no necesitas los datos en tiempo real. Le permite almacenar temporalmente sus datos de IoT en un bucket de Amazon S3 antes de importarlos AWS IoT SiteWise. O simplemente puede cargar sus datos a S3 para almacenarlos, independientemente de si planea importarlos o no AWS IoT SiteWise. Esto resulta útil para agrupar y almacenar en búfer los datos de sus dispositivos y pasarelas antes de incorporarlos. AWS IoT SiteWise Con esta opción, los datos se cargan en el bucket S3 especificado en formato Parquet con una frecuencia configurada. A continuación, puede importar estos datos al AWS IoT SiteWise almacenamiento para su posterior análisis y procesamiento.

Para añadir un destino almacenado en búfer mediante Amazon S3

Utilice la AWS IoT SiteWise consola o AWS CLI añada un destino que almacene datos en búfer mediante Amazon S3 a su puerta de enlace SiteWise Edge.

Console

Úselo Consola de administración de AWS para añadir un AWS IoT SiteWise destino almacenado en búfer mediante Amazon S3.

1. Abra la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, seleccione Puertas de enlace Edge.
3. Seleccione la puerta de enlace a la que quiere añadir un destino.
4. En la sección Destinos, selecciona Añadir destino.
5. En la página Añadir destino, introduce los detalles del destino:
 - a. Un nombre para el destino en el campo Nombre del destino.
 - b. Seleccione AWS IoT SiteWise Almacenado en búfer mediante Amazon S3 como Tipo de destino. AWS IoT SiteWise almacenado en búfer mediante Amazon S3 envía los datos a Amazon Simple Storage Service por lotes, en formato Parquet, y después los importa al almacenamiento. AWS IoT SiteWise

6. Introduzca la URL de Amazon S3 de la ubicación en la que desee almacenar los datos de la puerta de enlace. Para buscar la ruta, seleccione Browse S3. Una vez agregado un depósito, también puede verlo seleccionando Ver.
7. Especifique la frecuencia con la que su puerta de enlace debe cargar datos a Amazon S3 introduciendo un período de tiempo y seleccionando un incremento de tiempo para la frecuencia de carga de datos. El valor de frecuencia debe ser superior a 0 e inferior o igual a 30 días.
8. En Configuración de almacenamiento de datos, determine qué hacer con los datos de la puerta de enlace después de importarlos a AWS IoT SiteWise. Hay dos decisiones que se deben tomar en relación con el almacenamiento de datos:
 - Si desea copiar los datos importados al AWS IoT SiteWise almacenamiento, active la casilla Copiar los datos al almacenamiento. Esta opción duplica los datos importados del bucket de Amazon S3 configurado en el AWS IoT SiteWise almacenamiento.
 - Si decide importar los datos de su bucket de Amazon S3 al AWS IoT SiteWise almacenamiento, también puede especificar si los datos importados se deben eliminar una vez finalizada la importación. Seleccione la casilla Eliminar datos de Amazon S3 para eliminar la fecha importada del bucket de Amazon S3 configurado después de importarla al AWS IoT SiteWise almacenamiento.
9. Añada filtros de ruta a su destino. Para obtener más información, consulte, [Agregue filtros de ruta a los destinos de AWS IoT SiteWise Edge](#).

AWS CLI

Example: Cree un nuevo AWS IoT SiteWise destino almacenado en búfer mediante Amazon S3

Utilice la [UpdateGatewayCapabilityConfiguration](#) API para configurar el publicador.

Establezca el parámetro capabilityNamespace como `iotsitewise:publisher:3`.

```
{
  "sources": [
    {
      "type": "MQTT"
    }
  ],
  "destinations": [
    {
      "type": "SITEWISE_BUFFERED",
```

```

"name": "your-s3-destination-name",
"config": {
  "targetBucketArn": "arn:aws:s3:::amzn-s3-demo-bucket/Optional/SomeFolder",
  "publishPolicy": {
    "publishFrequency": "15m",
    "localSizeLimitGB": 10
  },
  "siteWiseImportPolicy": {
    "enableSiteWiseStorageImport": true,
    "enableDeleteAfterImport": true,
    "bulkImportJobRoleArn": "arn:aws:iam::123456789012:role/your-role-name"
  }
},
"filters": [
  {
    "type": "PATH",
    "config": {
      "paths": [
        "#"
      ]
    }
  }
]
}

```

Example: Actualizar un AWS IoT SiteWise destino almacenado en búfer mediante Amazon S3

Para actualizar un destino AWS IoT SiteWise en tiempo real existente, utilice primero la DescribeGatewayCapabilityConfiguration API para encontrar el destinationId

Espacio de nombres del publicador: iotsitewise:publisher:3

```

{
  "sources": [
    {
      "type": "MQTT"
    }
  ],
  "destinations": [
    {

```

```

    "id": "your-existing-destination-id",
    "type": "SITEWISE_BUFFERED",
    "name": "your-s3-destination-name",
    "config": {
      "targetBucketArn": "arn:aws:s3:::amzn-s3-demo-bucket/Optional/SomeFolder",
      "publishPolicy": {
        "publishFrequency": "15m",
        "localSizeLimitGB": 10
      },
      "siteWiseImportPolicy": {
        "enableSiteWiseStorageImport": true,
        "enableDeleteAfterImport": true,
        "bulkImportJobRoleArn": "arn:aws:iam::123456789012:role/your-role-name"
      }
    },
    "filters": [
      {
        "type": "PATH",
        "config": {
          "paths": [
            "#"
          ]
        }
      }
    ]
  }
]
}

```

Las siguientes opciones de configuración son específicas de las puertas de enlace habilitadas para MQTT que utilizan el espacio de nombres. `iotsitewise:publisher:3`

sources

Define las fuentes de datos a las que se transferirán los datos de su equipo industrial. AWS IoT SiteWise Para las pasarelas habilitadas para MQTT, utilice. MQTT

Tipo: matriz de objetos

Obligatorio: sí

destinations

Define dónde enviar los datos. Los destinos se transmiten en tiempo real o se almacenan en búfer mediante Amazon S3. Se requiere al menos un objeto de destino, pero puede añadir una matriz vacía. Puede tener un destino en tiempo real para cada puerta de enlace. Para obtener más información, consulte [Comprenda los destinos de AWS IoT SiteWise Edge](#).

Tipo: matriz de objetos

Obligatorio: sí

id

El identificador único del destino. Puede proporcionar un identificador de destino existente o dejarlo en blanco para que se genere automáticamente un nuevo identificador para el destino.

Tipo: cadena

Requerido: no

type

Tipo de destino. Las opciones incluyen: SITEWISE_REALTIME y SITEWISE_BUFFERED. Elija SITEWISE_BUFFERED.

- SITEWISE_REALTIME(predeterminado): envía los datos directamente al AWS IoT SiteWise almacenamiento en tiempo real. Para obtener más información, consulte [Agregue un destino de AWS IoT SiteWise Edge en tiempo real](#).
- SITEWISE_BUFFERED— Envíe los datos a Amazon S3 en lotes en formato Parquet y, a continuación, impórtelos al AWS IoT SiteWise almacenamiento.

Tipo: cadena

Obligatorio: sí

name

Un nombre único para el destino.

Tipo: cadena

Obligatorio: sí

config

Configuración específica para el tipo de destino en formato JSON. La configuración varía entre destinos en tiempo real y en búfer.

Tipo: objeto

Obligatorio: sí

`targetBucketArn`

El ARN del bucket en el que se va a publicar. Elija lo mismo Región de AWS para Amazon S3 AWS IoT SiteWise y para Amazon S3. Si elige un prefijo, debe tener entre 1 y 255 caracteres.

Note

AWS IoT SiteWise, incluida la puerta de enlace, tendrá acceso a todo el bucket de S3 especificado. Recomendamos utilizar un depósito dedicado para la ingesta de datos almacenados en búfer.

Tipo: cadena

Obligatorio: sí

`publishPolicy`

Detalles de la política de publicación.

Tipo: objeto

Obligatorio: sí

`publishFrequency`

La frecuencia con la que la puerta de enlace SiteWise Edge publica en el bucket de Amazon S3. La frecuencia de carga de datos a Amazon S3 debe ser superior a 0 minutos e inferior o igual a 30 días. Puede usar mh, y d cuando especifique una frecuencia de publicación, la antigüedad. Tenga en cuenta que m representa minutos, h representa horas y d representa días. El valor predeterminado es 15 minutos.

Tipo: cadena

Obligatorio: sí

`localSizeLimitGB`

El tamaño máximo de los archivos escritos en el disco local en GB. Si se supera este umbral, el publicador publica todos los datos almacenados en el búfer en su destino.

Tipo: entero

Obligatorio: sí

`siteWiseImportPolicy`

Detalles de la política de importación para importar datos a AWS IoT SiteWise.

Tipo: objeto

Obligatorio: sí

`enableSiteWiseStorageImport`

`true` Configúrelo para importar datos de un bucket de Amazon S3 al AWS IoT SiteWise almacenamiento. Inicialmente hace una copia de los datos en AWS IoT SiteWise. A continuación, si se establece en `enableDeleteAfterImport true`, los datos de S3 se eliminan después de copiarlos a AWS IoT SiteWise. Se aplican las implicaciones de precio. El valor predeterminado es `true`.

Tipo: Booleano

Obligatorio: sí

`enableDeleteAfterImport`

`true` Configúrelo para eliminar el archivo del bucket de Amazon S3 después de introducirlo en el AWS IoT SiteWise almacenamiento. El valor predeterminado es `true`.


Tipo: Booleano

Obligatorio: sí

`bulkImportJobRoleArn`

El ARN de la función de IAM que AWS IoT SiteWise asume leer los datos almacenados en búfer de Amazon S3 durante la ingesta de datos. Esta función

se utiliza cuando un dispositivo perimetral solicita iniciar el proceso de AWS IoT SiteWise APIs importación masiva.

 Note

Si `enableSiteWiseStorageImport` se establece en `true`, este parámetro es obligatorio.

Tipo: cadena

Requerido: no

Añada filtros de ruta para su destino. Para obtener más información, consulte [Agregue filtros de ruta a los destinos de AWS IoT SiteWise Edge](#).

Agregue filtros de ruta a los destinos de AWS IoT SiteWise Edge

Agregue filtros de ruta a un destino. Los filtros de ruta utilizan la sintaxis de temas de MQTT, donde # es un carácter comodín que coincide con cualquier número de niveles y + es un carácter comodín que coincide con un único nivel. Puede añadir varios destinos a una puerta de enlace, cada uno con su propio conjunto de filtros de ruta suscritos a la telemetría de su equipo.

Siemens Industrial Edgela pasarelas utilizan un prefijo para garantizar la compatibilidad. Para obtener más información, consulte [Prefijos para filtros de ruta](#).

Console

Para añadir filtros de ruta

1. Abra la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, seleccione Puertas de enlace Edge.
3. Seleccione la puerta de enlace a la que desee añadir filtros de ruta.
4. En la sección Filtros de ruta, en Agregar destino, elija Agregar filtro de ruta.
5. Introduce el filtro de ruta al que quieres que se suscriba este destino. Puede utilizar caracteres comodín (#y+) para suscribirse a varias rutas.
6. Seleccione Añadir filtro de ruta para añadir el filtro de ruta a la lista.

7. Repita los pasos para añadir filtros de ruta adicionales, si es necesario.
8. Una vez que haya agregado todos los filtros de ruta necesarios, elija Crear.

AWS CLI for self-hosted gateways

Example: Configuración del filtro de ruta

```
{
  "destinations": [
    {
      ...
    }
  ],
  "filters": [
    {
      "type": "PATH",
      "config": {
        "paths": [
          "home/+/sensor1/temperature",
          "home/livingroom/sensor1/temperature",
          "home/livingroom/sensor1/temperature",
          "building/#"
        ]
      }
    }
  ]
}
```

AWS CLI for Siemens IEGateways

Example: Configuración de prefijos para filtros de ruta

Capture todos los datos utilizando los prefijos de datos (ie/d) y metadatos (ie/m) de cada filtro de ruta.

```
{
  "destinations": [
    {
      ...
    }
  ]
}
```

```
],
"filters": [
  {
    "type": "PATH",
    "config": {
      "paths": [
        "ie/d/home/+sensor12/temperature",
        "ie/m/home/livingroom/sensor12/temperature",
        "ie/d/home/livingroom/sensor13/temperature2",
        "ie/m/home/livingroom/sensor13/temperature2",
        "ie/d/building/#",
        "ie/m/building/#"
      ]
    }
  }
]
```

Note

Copie los filtros de ruta entre destinos descargando la lista de filtros de ruta. Para obtener más información, consulte [Descargue todos los filtros de rutas de un destino \(consola\)](#).

Cargue filtros de ruta de forma masiva

Para cargar filtros de ruta de forma masiva, usa un archivo CSV o de texto. AWS IoT SiteWise elimina automáticamente los duplicados exactos al subir archivos. Por ejemplo, `windfarm/site1/` y `windfarm/site1/` son duplicados exactos los que AWS IoT SiteWise captan porque la cadena es exactamente la misma. Los duplicados parciales no se eliminan y conllevan cargos adicionales. Por ejemplo, `windfarm/#` y `windfarm/site1` son temas que se superponen porque `windfarm/site1` ya están abarcados por `windfarm/#`

Note

Evite los duplicados para evitar cargos adicionales. El archivo cargado debe estar en formato.csv o.txt. No puede contener ningún encabezado y debe constar de una sola columna. En la columna, haz una lista de los filtros de ruta, con cada filtro en una línea independiente. No se debe incluir ninguna otra información en el archivo.

Requisitos de carga de archivos

Estos son requisitos de filtro de rutas adicionales.

- Puede cargar un archivo.csv o.txt. No se admite ningún otro formato de archivo.
- Los archivos CSV (.csv) no pueden tener encabezados y solo deben contener una columna.
- Puede tener un filtro de ruta en cada línea.
- Los archivos cargados no pueden estar vacíos.
- Cuando se usa # como comodín, debe ser el último carácter del filtro de temas. Por ejemplo, topic/# o como un carácter independiente por tema en particular. Sin embargo, tenga en cuenta que # también se puede usar como un carácter normal dentro de un nombre por tema, como factory/machine#1/topic. Para obtener más información consulte [Caracteres especiales en los nombres de los filtros de ruta](#)
- También puedes usar el + personaje. Por ejemplo, úselo factory/+ /temp para obtener todas las temperaturas de las fábricas en lugar de hacerlo factory/machine2/temp de factory/machine3/temp forma individual.

Administre los destinos de AWS IoT SiteWise Edge

Tras añadir destinos, puede realizar varias operaciones para gestionarlos, como editar las configuraciones de los destinos, eliminar destinos y gestionar los filtros de rutas.

Editar un destino

Seleccione el botón de radio situado junto al destino en la tabla y pulse el botón Editar para editar un destino.

Console

Para editar un destino mediante la AWS IoT SiteWise consola

1. Abra la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación de la izquierda, selecciona las puertas de enlace Edge en la sección Edge.
3. Seleccione la puerta de enlace adecuada.
4. En la sección Destinos, elija el destino que desee editar y, a continuación, elija Editar.
5. Modifica el destino y, a continuación, selecciona Guardar.

AWS CLI

Para editar un destino mediante AWS CLI

- Puede editar un destino modificando la información de configuración de la capacidad de JSON.

```
aws iotsitewise update-gateway-capability-configuration \  
--gateway-id your-gateway-id \  
--capability-namespace "iotsitewise:publisher:3" \  
--capability-configuration '{  
  "sources": [  
    {  
      "type": "MQTT"  
    }  
  ],  
  "destinations": [  
    {  
      "id": "your-existing-destination-id",  
      "type": "SITEWISE_REALTIME",  
      "name": "your-updated-destination-name",  
      "config": {  
        "publishingOrder": "TIME_ORDER",  
        "enableCompression": true,  
        "dropPolicy": {  
          "cutoffAge": "10d",  
          "exportPolicy": {  
            "retentionPeriod": "10d",  
            "rotationPeriod": "6h",  
            "exportSizeLimitGB": 10  
          }  
        }  
      },  
      "maxBatchWaitTime": "15s"  
    },  
    {  
      "filters": [  
        {  
          ...  
        }  
      ]  
    }  
  ]  
}'
```

Note

No puedes actualizar el destino `type` `ocapability-namespace`. Por ejemplo, no puedes cambiar de un tipo de `SITewise_REALTIME` a `SITewise_BUFFERED`. Puede tener un destino en tiempo real para cada puerta de enlace habilitada para MQTT.

Eliminar un destino

Si ya no necesita un destino, puede eliminarlo de su puerta de enlace SiteWise Edge.

Console

Para eliminar un destino mediante la AWS IoT SiteWise consola

1. Abra la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación de la izquierda, selecciona las puertas de enlace Edge en la sección Edge.
3. Seleccione la puerta de enlace adecuada.
4. En la sección Destinos, elija el destino que desee eliminar y, a continuación, elija Eliminar. Aparece una pantalla de confirmación.
5. Para confirmar su elección de eliminar el destino, escriba «eliminar» en el cuadro de confirmación.

AWS CLI

Para eliminar un destino mediante AWS CLI

- Elimine la configuración de la capacidad de la puerta de enlace especificando el ID de la puerta de enlace y modificando la configuración de la capacidad para eliminar el destino que desea eliminar.

```
aws iotsitewise update-gateway-capability-configuration \  
  --gateway-id your-gateway-id \  
  --capability-namespace "iotsitewise:publisher:3" \  
  --capability-configuration '{  
    "sources": [  
      {
```

```
        "type": "MQTT"  
      }  
    ],  
    "destinations": []  
  }'
```

Note

La matriz de destinos puede estar vacía ([]), pero el propio objeto de destino debe incluirse en la configuración de la capacidad.

Descargue todos los filtros de rutas de un destino (consola)

Descarga un archivo CSV que contenga todos los filtros de rutas de la AWS IoT SiteWise consola. Puede utilizar una lista descargada de filtros de rutas para compartir fácilmente las listas de filtros de rutas entre los destinos de las puertas de enlace.

Para descargar un archivo CSV de todos los filtros de rutas mediante la AWS IoT SiteWise consola

1. Abra la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación de la izquierda, selecciona las puertas de enlace Edge en la sección Edge.
3. Seleccione la puerta de enlace que contiene los filtros de ruta.
4. Elija Agregar destino o Editar destino.
5. Ve a la sección Filtros de ruta y selecciona Descargar CSV.

Note

El archivo CSV incluye todos los filtros de ruta de un destino concreto, independientemente de los que haya seleccionado de la lista de filtros de ruta.

Edita un filtro de ruta

Puede editar filtros de ruta individuales para refinar los datos que recibe su destino.

Console

Con la AWS IoT SiteWise consola, puede editar cada filtro de ruta individual dentro de cada cuadro de texto correspondiente.

Para editar un filtro de ruta mediante la AWS IoT SiteWise consola

1. Abra la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación de la izquierda, selecciona las puertas de enlace Edge en la sección Edge.
3. Seleccione la puerta de enlace que contiene los filtros de ruta.
4. Seleccione el destino apropiado.
5. Elija Edit (Edición de).
6. Seleccione el cuadro de texto de la fila que contiene el filtro de ruta que desee editar.
7. Actualice el texto del filtro de ruta y asegúrese de que la casilla de verificación del filtro de ruta editado esté seleccionada.
8. Seleccione Save.

AWS CLI

Para editar los filtros de ruta de un destino mediante el AWS CLI, primero recupere la configuración actual, modifíquela y, a continuación, actualícela mediante el `update-gateway-capability-configuration` comando.

Para editar un filtro de ruta mediante AWS CLI

1. Recupere la configuración de capacidad actual:

```
aws iotsitewise describe-gateway-capability-configuration \  
  --gateway-id your-gateway-id \  
  --capability-namespace "iotsitewise:publisher:3" \  
  --query "capabilityConfiguration"
```

2. Edite el JSON para modificar los filtros de ruta según sea necesario.
3. Actualice la configuración de la capacidad con los filtros de ruta modificados:

```
aws iotsitewise update-gateway-capability-configuration \  
  --gateway-id your-gateway-id \  
  --capability-namespace "iotsitewise:publisher:3" \  
  --query "capabilityConfiguration"
```

```
--capability-namespace "iotsitewise:publisher:3" \  
--capability-configuration json-containing-your-updated-path-filters
```

Elimine un filtro de ruta

Puede eliminar los filtros de ruta de un destino para controlar los datos que recibe de las fuentes de MQTT y de las canalizaciones de procesamiento de datos.

Console

Para eliminar un filtro de ruta mediante la consola AWS IoT SiteWise

1. Abra la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación de la izquierda, selecciona las puertas de enlace Edge en la sección Edge.
3. Seleccione la puerta de enlace que contiene los filtros de ruta.
4. Seleccione el destino apropiado.
5. Elija Edit (Edición de).
6. En la pantalla Editar destino, en la sección Filtros de ruta, seleccione uno o más filtros de ruta que desee eliminar.
7. Elija Eliminar. Aparece un mensaje de confirmación de eliminación. Si desea continuar con la eliminación de los filtros de ruta, seleccione Eliminar en la pantalla de confirmación.

AWS CLI

Para eliminar un destino mediante AWS CLI

- Elimine un filtro de ruta eliminándolo de la configuración de capacidades.

```
aws iotsitewise update-gateway-capability-configuration \  
  --gateway-id your-gateway-id \  
  --capability-namespace "iotsitewise:publisher:3" \  
  --capability-configuration '{  
    "sources": [  
      {  
        "type": "MQTT"  
      }  
    ],
```

```

"destinations": [
  {
    "id": "your-destination-id",
    "type": "SITEWISE_REALTIME",
    "name": "your-destination-name",
    "config": {
      ...
    },
    "filters": [
      {
        "type": "PATH",
        "config": {
          "paths": [
            "/path1",
            "/path2",
            "/delete-a-path-to-remove-it"
          ]
        }
      }
    ]
  }
]
}

```

Note

La matriz de filtros puede estar vacía ([]), pero el propio objeto de filtros debe incluirse en la configuración de la capacidad.

Administre las puertas de enlace SiteWise Edge

Puede usar la AWS IoT SiteWise consola y las operaciones de la API para administrar las puertas de enlace AWS IoT SiteWise Edge. También puede usar la aplicación [AWS OpsHub para Windows](#) [AWS IoT SiteWise para](#) administrar algunos aspectos de su puerta de enlace SiteWise Edge desde su dispositivo local.

Le recomendamos encarecidamente que utilice la AWS IoT SiteWise aplicación AWS OpsHub for para supervisar el uso del disco en su dispositivo local. También puedes monitorear las CloudWatch métricas Gateway.AvailableDiskSpace y las de Gateway.UsedPercentageDiskSpace

Amazon y crear alarmas para recibir notificaciones cuando se agote el espacio en disco. Para obtener más información sobre CloudWatch las alarmas de Amazon, consulta [Crear una CloudWatch alarma basada en un umbral estático](#).

Asegúrese de que su dispositivo disponga de espacio suficiente para los datos venideros. Cuando está a punto de quedarse sin espacio en su dispositivo local, el servicio elimina automáticamente una pequeña cantidad de datos con las marcas de tiempo más antiguas para dejar espacio para nuevos datos.

Para comprobar si el servicio ha eliminado sus datos, haga lo siguiente:

1. Inicie sesión en la AWS OpsHub AWS IoT SiteWise solicitud.
2. Elija Configuración.
3. En Registros, especifique un intervalo de tiempo y, a continuación, seleccione Descargar.
4. Descomprima el archivo de registro.
5. Si el archivo de registro contiene el siguiente mensaje, el servicio ha eliminado los datos: se han eliminado *number* bytes de datos para evitar que el almacenamiento de SiteWise Edge Gateway se quede sin espacio.

Administre su puerta de enlace SiteWise Edge con la AWS IoT SiteWise consola

Puede usar la AWS IoT SiteWise consola para configurar, actualizar y monitorear todas las puertas de enlace SiteWise Edge de su AWS cuenta.

[Puede ver sus puertas de enlace de SiteWise Edge navegando a la página de puertas de enlace de Edge de la consola.AWS IoT SiteWise](#) Para acceder a la página Detalles de la puerta de enlace Edge correspondiente a una puerta de enlace específica, elija el nombre de una puerta de enlace de Edge.

En la pestaña Descripción general de la página Detalles de la puerta de enlace de Edge, puede hacer lo siguiente:

- En la sección Orígenes de datos, actualice la configuración de orígenes de datos y configure orígenes de datos adicionales
- Seleccione Abrir CloudWatch métricas para ver la cantidad de puntos de datos ingeridos por fuente de datos en la consola de métricas CloudWatch

- En la sección de capacidades de Edge, añada paquetes de datos a su puerta de enlace de SiteWise Edge haciendo clic en Editar
- En la sección de configuración de la puerta de enlace, consulte el estado de conectividad de las puertas de enlace SiteWise Edge
- En la sección de configuración del editor, consulte el estado de sincronización y la configuración de la puerta de enlace SiteWise Edge del componente del AWS IoT SiteWise editor

En la pestaña Actualizaciones de la página Detalles de la puerta de enlace Edge puede ver las versiones actuales del componente y del paquete que están implementadas en la puerta de enlace de Edge. Aquí también se implementan las nuevas versiones cuando están disponibles.

Administre las puertas de enlace SiteWise Edge mediante AWS OpsHub AWS IoT SiteWise

Utiliza la AWS IoT SiteWise aplicación AWS OpsHub for para administrar y monitorear sus puertas de enlace SiteWise Edge autohospedadas. Esta aplicación proporciona las siguientes opciones de monitoreo y administración:

- En Información general puede hacer lo siguiente:
 - Consulte los detalles de la puerta de enlace SiteWise Edge que le ayudarán a obtener información sobre los datos de sus dispositivos de puerta de enlace SiteWise Edge, identificar problemas y mejorar el rendimiento de la puerta de enlace SiteWise Edge.
 - Vea los portales SiteWise Monitor que monitorean los datos de los servidores y equipos locales ubicados en la periferia. Para obtener más información, consulte [Qué es AWS IoT SiteWise Monitor](#) en la Guía de la aplicación AWS IoT SiteWise Monitor .
- En Salud, hay un panel de control que muestra los datos de su puerta de enlace SiteWise Edge. Los expertos en el campo, como los ingenieros de procesos, pueden usar el panel para ver una descripción general del comportamiento de las puertas de enlace SiteWise Edge.
- En Activos puede ver los activos implementados en el dispositivo local y el último valor recopilado o calculado para las propiedades de los activos.
- En Configuración puede hacer lo siguiente:
 - Si el paquete de procesamiento de datos está instalado, consulte la información de configuración de la puerta de enlace SiteWise Edge y sincronice los recursos con la AWS nube.
 - Descargue los archivos de autenticación que puede usar para acceder a la puerta de enlace SiteWise Edge mediante otras herramientas.

- Descarga los registros que puedes usar para solucionar problemas con la puerta de enlace SiteWise Edge.
- Vea los AWS IoT SiteWise componentes implementados en la puerta de enlace SiteWise Edge.

 Important

Se requiere lo siguiente AWS OpsHub para su uso AWS IoT SiteWise:

- El dispositivo local y la AWS IoT SiteWise aplicación AWS OpsHub for deben estar conectados a la misma red.
- El paquete de procesamiento de datos debe estar habilitado.

Para administrar las puertas de enlace SiteWise Edge mediante AWS OpsHub

1. Descargue e instale la aplicación [AWS OpsHubAWS IoT SiteWise para Windows](#).
2. Abra la aplicación.
3. Si no ha configurado credenciales locales para su puerta de enlace, siga los pasos de [Acceda a su puerta de enlace SiteWise Edge con las credenciales del sistema operativo local](#) para configurarlas.
4. Puede iniciar sesión en su puerta de enlace SiteWise Edge con sus credenciales de Linux o del Protocolo ligero de acceso a directorios (LDAP). Para iniciar sesión en su puerta de enlace SiteWise Edge, realice una de las siguientes acciones:

Linux

1. En Nombre de host o dirección IP, introduzca el nombre de host o la dirección IP de su dispositivo local.
2. En Autenticación, elija Linux.
3. En Nombre de usuario, introduzca el nombre de usuario de su sistema operativo Linux.
4. En Contraseña, introduzca la contraseña de su sistema operativo Linux.
5. Elija Iniciar sesión.

LDAP

1. En Nombre de host o dirección IP, introduzca el nombre de host o la dirección IP de su dispositivo local.
2. En Autenticación, elija LDAP.
3. En Nombre de usuario, introduzca su nombre de usuario de LDAP.
4. En Contraseña, introduzca su contraseña de LDAP.
5. Seleccione Iniciar sesión.

Acceda a su puerta de enlace SiteWise Edge con las credenciales del sistema operativo local

Además del Protocolo ligero de acceso a directorios (LDAP), puede utilizar las credenciales de Linux o Windows para acceder a su puerta de enlace Edge autohospedada SiteWise .

Important

Para acceder a su puerta de enlace SiteWise Edge con credenciales de Linux, debe activar el paquete de procesamiento de datos de su SiteWise puerta de enlace Edge.

Acceda a su puerta de enlace SiteWise Edge con las credenciales del sistema operativo Linux

En los siguientes pasos se asume que utiliza un dispositivo con Ubuntu. Si utiliza una distribución de Linux diferente, consulte la documentación correspondiente a su dispositivo.

Para crear un grupo de usuarios de Linux

1. Para crear un grupo de administradores, ejecute el siguiente comando.

```
sudo groupadd --system SWE_ADMIN_GROUP
```

Los usuarios del SWE_ADMIN_GROUP grupo pueden permitir el acceso de administrador a la puerta de enlace SiteWise Edge.

2. Para crear un grupo de usuarios, ejecute el siguiente comando.

```
sudo groupadd --system SWE_USER_GROUP
```

Los usuarios del SWE_USER_GROUP grupo pueden permitir el acceso de solo lectura a la puerta de enlace SiteWise Edge.

3. Para añadir un usuario al grupo de administradores, ejecute el siguiente comando. Sustituya *user-name* y *password* por el nombre de usuario y la contraseña que desee agregar.

```
sudo useradd -p $(openssl passwd -1 password) user-name
```

4. Para añadir un usuario a uno de SWE_ADMIN_GROUP o SWE_USER_GROUP, *user-name* sustitúyalo por el nombre de usuario que añadió en el paso anterior.

```
sudo usermod -a -G SWE_ADMIN_GROUP user-name
```

Ahora puede usar el nombre de usuario y la contraseña para iniciar sesión en la puerta de enlace SiteWise Edge de la AWS IoT SiteWise aplicación AWS OpsHub for.

Acceda a su puerta de enlace SiteWise Edge con las credenciales de Windows

En los pasos siguientes se asume que utiliza un dispositivo con Windows.

Important

La seguridad es una responsabilidad compartida entre usted AWS y usted. Cree una política de contraseña segura con al menos 12 caracteres y una combinación de mayúsculas, minúsculas, números y símbolos. Además, configure las reglas del cortafuegos de Windows para permitir el tráfico entrante en el puerto 443 y bloquear el tráfico entrante en todos los demás puertos.

Para crear un grupo de usuarios de Windows Server

1. Ejecute PowerShell como administrador.
 - a. En el servidor Windows en el que desee instalar SiteWise Edge Gateway, inicie sesión como administrador.
 - b. Ingresa PowerShell en la barra de búsqueda de Windows.

- c. En los resultados de la búsqueda, haz clic con el botón derecho en la PowerShell aplicación de Windows. Seleccione Ejecutar como administrador.
2. Para crear un grupo de administradores, ejecute el siguiente comando.

```
net localgroup SWE_ADMIN_GROUP /add
```

Debe ser un usuario del SWE_ADMIN_GROUP grupo para permitir el acceso de administrador a la puerta de enlace SiteWise Edge.

3. Para crear un grupo de usuarios, ejecute el siguiente comando.

```
net localgroup SWE_USER_GROUP /add
```

Debe ser un usuario del SWE_USER_GROUP grupo para permitir el acceso exclusivo a la puerta de enlace SiteWise Edge.

4. Para añadir un usuario, ejecute el siguiente comando. Sustituya *user-name* y *password* por el nombre de usuario y la contraseña que desee crear.

```
net user user-name password /add
```

5. Para añadir un usuario al grupo de administradores, ejecute el siguiente comando. *user-name* Sustitúyalo por el nombre de usuario que desee añadir.

```
net localgroup SWE_ADMIN_GROUP user-name /add
```

Ahora puede usar el nombre de usuario y la contraseña para iniciar sesión en la puerta de enlace SiteWise Edge en la AWS IoT SiteWise aplicación AWS OpsHub for.

Administre el certificado de la puerta de enlace SiteWise Edge

Puede usar SiteWise Monitor y aplicaciones de terceros, como Grafana, en sus dispositivos SiteWise Edge Gateway. Estas aplicaciones requieren una conexión TLS al servicio. SiteWise En la actualidad, las pasarelas Edge utilizan un certificado autofirmado. Si utiliza un navegador para abrir las aplicaciones, como un portal de SiteWise Monitor, es posible que reciba una advertencia sobre un certificado que no es de confianza.

A continuación, se muestra cómo descargar el certificado de confianza de AWS OpsHub la AWS IoT SiteWise aplicación.

1. Inicie sesión en la aplicación.
2. Seleccione Configuración.
3. En Autenticación, elija Descargar certificado.

A continuación se supone que utiliza Google Chrome o FireFox. Si utiliza un navegador diferente, consulte la documentación correspondiente al mismo. Para añadir el certificado que ha descargado en el paso anterior a un navegador, realice una de las siguientes acciones:

- Si utiliza Google Chrome, siga las instrucciones indicadas en [Configuración de certificados](#) de la Documentación de ayuda de Google Chrome Enterprise.
- Si utiliza Firefox, siga las instrucciones indicadas en [Carga del certificado en el navegador Mozilla o Firefox](#) de la Documentación de Oracle.

Cambie la versión de los paquetes de componentes de SiteWise Edge Gateway

Puede usar la AWS IoT SiteWise consola para cambiar la versión de los paquetes de componentes en sus gateways SiteWise Edge.

Para cambiar la versión de un paquete de componentes de una puerta de enlace SiteWise Edge

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación izquierdo, seleccione Puertas de enlace.
3. Seleccione la puerta de enlace SiteWise Edge cuyas versiones del paquete desee cambiar.
4. En Configuración de puerta de enlace, seleccione Ver las versiones de software.
5. En la página Editar versiones de software, para el paquete cuya versión desea actualizar, seleccione la versión que desea implementar y elija Implementar.
6. Seleccione Listo.

Enumere las pasarelas de SiteWise Edge

Console

Para enumerar las puertas de enlace de SiteWise Edge

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, seleccione Puertas de enlace Edge.
3. Vea la lista de todas sus puertas de enlace SiteWise Edge.

AWS CLI

Para enumerar las pasarelas que utiliza AWS CLI, siga estos pasos:

- Usa el comando `list-gateways` para ver todas tus pasarelas:

```
aws iotsitewise list-gateways
```

Este comando devuelve una lista de sus puertas de enlace con sus nombres y otra IDs información.

También puede especificar los parámetros de paginación:

```
aws iotsitewise list-gateways --max-results 10 --next-token your-token
```

Para obtener más información, consulte [list-gateways](#) en la Referencia de comandos. AWS CLI

Describa una puerta de enlace Edge SiteWise

Console

Para ver los detalles de la puerta de enlace

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, seleccione Puertas de enlace Edge.
3. Elija el nombre de la puerta de enlace de la que desea ver los detalles.

4. Vea los detalles de la puerta de enlace en la página de detalles de la puerta de enlace de Edge.

AWS CLI

Para obtener información detallada sobre el uso de una puerta de enlace específica AWS CLI, sigue estos pasos:

- Usa el comando `describe-gateway` con el ID de la puerta de enlace:

```
aws iotsitewise describe-gateway --gateway-id a1b2c3d4-5678-90ab-  
cdef-1a1a1EXAMPLE
```

Este comando devuelve información detallada sobre la puerta de enlace.

Para obtener más información, consulte [describe-gateway](#) en la AWS CLI Referencia de comandos.

Cree una puerta de enlace Edge SiteWise

Console

Para crear una puerta de enlace SiteWise Edge

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, seleccione Puertas de enlace Edge.
3. Seleccione Crear puerta de enlace.
4. Introduzca un nombre para la puerta de enlace.
5. Seleccione el grupo Greengrass para su puerta de enlace.
6. Si lo desea, añada etiquetas a su puerta de enlace.
7. Seleccione Crear.

AWS CLI

Para crear una nueva SiteWise puerta de enlace de IoT mediante AWS CLI, siga estos pasos:

- Utilice el comando `create-gateway` para crear una nueva puerta de enlace:

```
aws iotsitewise create-gateway \  
  --gateway-name "NewSiteWiseGateway" \  
  --gateway-platform '{  
    "greengrass": {  
      "groupArn": "arn:aws:greengrass:us-east-1:123456789012:group/  
a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE"  
    }  
  }' \  
  --tags '{  
    "Environment": "Production",  
    "Location": "Factory1"  
  }'
```

Este comando devuelve el ID y el ARN de la nueva puerta de enlace:

```
{  
  "gatewayId": "a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE",  
  "gatewayArn": "arn:aws:iotsitewise:us-east-1:123456789012:gateway/  
a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE"  
}
```

Para obtener más información, consulte [create-gateway](#) en la AWS CLI Referencia de comandos.

Actualice una puerta de enlace Edge SiteWise

Console

Para actualizar una puerta de enlace SiteWise Edge

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, seleccione Puertas de enlace Edge.
3. Seleccione la puerta de enlace que desee actualizar.
4. Elija Edit (Edición de).
5. Actualice el nombre de la puerta de enlace u otros ajustes según sea necesario.
6. Seleccione Save.

AWS CLI

Para actualizar una puerta de enlace existente mediante AWS CLI, siga estos pasos:

- Usa el comando `update-gateway` para actualizar el nombre de una puerta de enlace:

```
aws iotsitewise update-gateway \  
  --gateway-id a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE \  
  --gateway-name "UpdatedGatewayName"
```

Este comando no genera ninguna salida si se realiza correctamente.

Para obtener más información, consulte [update-gateway](#) en la Referencia de comandos. AWS CLI

Actualice la configuración de la capacidad del gateway

Console

Para actualizar la configuración de la capacidad de la pasarela

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, seleccione Puertas de enlace Edge.
3. Elija el nombre de la puerta de enlace que desea actualizar.
4. En la sección Fuentes de datos, elija Editar.
5. Actualice la configuración de la fuente de datos según sea necesario.
6. Seleccione Save.

AWS CLI

Para actualizar la configuración de capacidades de una puerta de enlace mediante AWS CLI, siga estos pasos:

- Utilice el `update-gateway-capability-configuration` comando para actualizar la configuración de la capacidad:

```
aws iotsitewise update-gateway-capability-configuration \  
  --gateway-id a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE \  
  --gateway-name "UpdatedGatewayName" \  
  --capacity-configuration { "Capacity": 100 }
```

```
--gateway-id a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE \  
--capability-namespace "iotsitewise:opcuacollector:1" \  
--capability-configuration '{  
  "sources": [  
    {  
      "name": "OPC-UA Server",  
      "endpoint": {  
        "certificateTrust": {  
          "type": "TrustAny"  
        },  
        "endpointUri": "opc.tcp://10.0.0.1:4840",  
        "securityPolicy": "NONE",  
        "messageSecurityMode": "NONE",  
        "identityProvider": {  
          "type": "Anonymous"  
        }  
      },  
      "measurementDataStreamPrefix": ""  
    }  
  ]  
}'
```

Este comando devuelve el espacio de nombres de la capacidad y el estado de sincronización:

```
{  
  "capabilityNamespace": "iotsitewise:opcuacollector:1",  
  "capabilitySyncStatus": "CONFIGURING"  
}
```

Para obtener más información, consulte la [update-gateway-capability-configuration](#) Referencia de AWS CLI comandos.

Etiquete los recursos de gateway

Console

Para etiquetar un recurso de puerta de enlace

1. Vaya a la [consola de AWS IoT SiteWise](#).

2. En el panel de navegación, seleccione Puertas de enlace Edge.
3. Elija el nombre de la puerta de enlace que desee etiquetar.
4. Elija la pestaña Etiquetas.
5. Elija Administrar etiquetas.
6. Elija Añadir nueva etiqueta e introduzca una clave y un valor para cada etiqueta.
7. Seleccione Save.

AWS CLI

Para añadir etiquetas a una pasarela mediante AWS CLI, sigue estos pasos:

- Usa el comando `tag-resource` para añadir etiquetas a una puerta de enlace:

```
aws iotsitewise tag-resource \  
  --resource-arn "arn:aws:iotsitewise:us-east-1:123456789012:gateway/  
a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE" \  
  --tags '{  
    "Department": "Operations",  
    "Project": "FactoryAutomation"  
  }'
```

Este comando no genera ninguna salida si se realiza correctamente.

Para obtener más información, consulte [tag-resource](#) en la Referencia de comandos. AWS CLI

Enumere las etiquetas de una puerta de enlace

Console

Para enumerar las etiquetas de una puerta de enlace

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, seleccione Puertas de enlace Edge.
3. Elija el nombre de la puerta de enlace cuyas etiquetas desee ver.
4. Elija la pestaña Etiquetas.
5. Vea la lista de etiquetas asociadas a la puerta de enlace.

AWS CLI

Para enumerar las etiquetas asociadas al uso de una puerta de enlace AWS CLI, sigue estos pasos:

- Utilice el `list-tags-for-resource` comando para enumerar las etiquetas de una puerta de enlace:

```
aws iotsitewise list-tags-for-resource \
  --resource-arn "arn:aws:iotsitewise:us-east-1:123456789012:gateway/
a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE"
```

Este comando devuelve las etiquetas asociadas a la puerta de enlace:

```
{
  "tags": {
    "Environment": "Production",
    "Location": "Factory1",
    "Department": "Operations",
    "Project": "FactoryAutomation"
  }
}
```

Para obtener más información, consulte [list-tags-for-resource](#) la Referencia de AWS CLI comandos.

Eliminar etiquetas de una puerta de enlace

Console

Para eliminar etiquetas de una puerta de enlace

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, seleccione Puertas de enlace Edge.
3. Elija el nombre de la puerta de enlace de la que desea eliminar las etiquetas.
4. Elija la pestaña Etiquetas.
5. Elija Administrar etiquetas.

6. Seleccione el icono de eliminación (X) situado junto a cada etiqueta que desee eliminar.
7. Seleccione Save.

AWS CLI

Para eliminar etiquetas de una pasarela mediante AWS CLI, sigue estos pasos:

- Usa el comando `untag-resource` para eliminar las etiquetas de una puerta de enlace:

```
aws iotsitewise untag-resource \  
  --resource-arn "arn:aws:iotsitewise:us-east-1:123456789012:gateway/  
a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE" \  
  --tag-keys '["Project", "Department"]'
```

Este comando no genera ninguna salida si se realiza correctamente.

Para obtener más información, consulte [untag-resource](#) en la Referencia de comandos. AWS CLI

Actualice la versión de un componente AWS IoT SiteWise

Actualice el componente de AWS IoT SiteWise puerta de enlace de su dispositivo AWS IoT Greengrass principal para garantizar el acceso a las funciones, las mejoras de rendimiento y los parches de seguridad más recientes.

Para actualizar un AWS IoT SiteWise componente en AWS IoT Greengrass

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación izquierdo, elija Edge Gateways.
3. Seleccione la puerta de enlace que desee editar y elija Editar.
4. En Edge Capabilities, en Versiones de software, selecciona Actualizaciones de software disponibles. Aparece la página Editar versiones de software.
5. Seleccione la versión del componente.

Note

Se recomienda seleccionar la última versión disponible. Mantener los componentes de la puerta de enlace le up-to-date ayuda a mantener una funcionalidad óptima para la recopilación y el procesamiento de datos industriales.

6. Elija Implementar. Esto inicia una AWS IoT Greengrass V2 implementación para actualizar el AWS IoT SiteWise componente en la puerta de enlace.

Eliminar una puerta de enlace SiteWise Edge

Console

Para eliminar la puerta de enlace de SiteWise Edge

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, seleccione Puertas de enlace Edge.
3. Elija la puerta de enlace que desea eliminar.
4. Elija Eliminar.
5. Para confirmar que desea eliminar la puerta de enlace, escriba «eliminar» y, a continuación, seleccione Eliminar en la ventana que aparece.

AWS CLI

Para eliminar una puerta de enlace mediante AWS CLI, sigue estos pasos:

1. Enumere sus puertas de enlace para identificar el ID de la puerta de enlace que desea eliminar.

```
aws iotsitewise list-gateways
```

Este comando devuelve una lista de sus puertas de enlace con sus IDs nombres y otra información:

```
{  
  "gatewaySummaries": [  


```

```
{
  "gatewayId": "a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE",
  "gatewayName": "ExampleCorpGateway",
  "gatewayCapabilitySummaries": [
    {
      "capabilityNamespace": "iotsitewise:opcuacollector:1",
      "capabilitySyncStatus": "IN_SYNC"
    }
  ],
  "creationDate": 1588369971.457,
  "lastUpdateDate": 1588369971.457
}
```

2. Elimine la puerta de enlace especificando su ID:

```
aws iotsitewise delete-gateway --gateway-id a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE
```

Este comando no genera ninguna salida si se realiza correctamente.

 Note

Al eliminar una puerta de enlace, algunos de los archivos de la puerta de enlace permanecen en el sistema de archivos de la puerta de enlace.

3. Para comprobar que la puerta de enlace se ha eliminado, puede volver a enumerarlas:

```
aws iotsitewise list-gateways
```

La puerta de enlace eliminada ya no debería aparecer en la lista.

Para obtener más información, consulte [delete-gateway](#) en la Referencia de AWS CLI comandos.

Realice copias de seguridad y restaure las puertas de enlace SiteWise Edge

En este tema se explica cómo restaurar las puertas de enlace de SiteWise Edge y hacer copias de seguridad de los datos métricos. Si tiene problemas con una puerta de enlace SiteWise Edge averiada en la misma máquina y necesita solucionar el problema, lea la AWS IoT SiteWise documentación [Solución de problemas con la puerta de enlace SiteWise Edge](#).

Note

La guía que se trata en este tema es para las puertas de enlace SiteWise Edge instaladas en la AWS IoT Greengrass V2 versión 2.1.0 o superior.

Copias de seguridad diarias de los datos de métricas

Crear una copia de seguridad es importante si desea transferir o restaurar los datos en una nueva máquina. Hacer una copia de seguridad de sus datos reduce en gran medida el riesgo de pérdida de datos operativos durante un proceso de transferencia o restauración.

Esta sección se aplica a las puertas de enlace que utilizan el paquete de procesamiento de datos. Para obtener más información sobre el paquete de procesamiento de datos, consulte [Configure un modelo de activos para el procesamiento de datos en SiteWise Edge](#).

La ruta de la carpeta influxdb es la siguiente:

Linux

```
/greengrass/v2/work/aws.iot.SiteWiseEdgeProcessor/influxdb
```

Windows

```
C:\greengrass\v2\work\aws.iot.SiteWiseEdgeProcessor\influxdb
```

Le recomendamos que haga una copia de seguridad de toda la carpeta con todo lo que contenga.

Le recomendamos que haga copias de seguridad periódicas de sus datos métricos desde la versión 1.0 SiteWise Edge en un disco duro externo o en la AWS nube.

Restaurar una puerta de enlace SiteWise Edge

Antes de intentar restaurar una puerta de enlace SiteWise Edge, asegúrese de que todos los dispositivos perimetrales conectados a la puerta de enlace estén detenidos o desconectados.

Utilice el siguiente procedimiento para restaurar una puerta de enlace SiteWise Edge:

1. Utilice el script de instalación descargado al crear la puerta de enlace SiteWise Edge para restaurar la puerta de enlace SiteWise Edge en la nueva máquina. Lea el procedimiento de [instalación del software de puerta de enlace SiteWise Edge en su dispositivo local](#) para configurar la puerta de enlace SiteWise Edge.

Si hubiera perdido o no pudiera encontrar el script de instalación, póngase en contacto con el [servicio de atención al cliente de AWS](#).

2. Una vez instalada la puerta de enlace SiteWise Edge, inicie sesión en la [AWS IoT Greengrass consola](#).
3. Para volver a implementar los componentes, vaya a Administrar y, a continuación, en Dispositivos AWS IoT Greengrass, seleccione Dispositivos centrales.
4. En la tabla de dispositivos AWS IoT Greengrass principales, seleccione el dispositivo principal correspondiente a su puerta de enlace SiteWise Edge.
5. Una vez en la página de dispositivos, abra la pestaña Implementaciones y seleccione su ID de implementación. Esto abre la página Implementaciones con su ID seleccionado.
6. Una vez en la página Implementaciones, en la parte superior derecha pulse el botón Acciones y seleccione la opción Revisar para iniciar una nueva implementación. Configure la implementación. Si desea mantener la implementación tal como está, pase a Revisar y Desplegar.
7. Espere a que el Estado de la implementación cambie a Completed.

Note

Además, todos los componentes del SiteWise Edge tardarán unos minutos en configurarse y funcionar por completo.

Restaurar AWS IoT SiteWise los datos

Utilice el siguiente procedimiento para restaurar los datos en una máquina nueva.

1. Copie la carpeta influxdb en la nueva máquina.
2. Detenga el SiteWise EdgeProcessor componente ejecutando el siguiente comando en su terminal:

Linux

```
sudo /greengrass/v2/bin/greengrass-cli component stop -n  
aws.iot.SiteWiseEdgeProcessor
```

Windows

```
C:\greengrass\v2\bin\greengrass-cli component stop -n  
aws.iot.SiteWiseEdgeProcesso
```

3. Localice la ruta en la que realizó la copia de seguridad de los datos y ejecute el siguiente comando:

Linux

```
sudo yes | sudo cp -rf <influxdb_backup_path> /greengrass/v2/work/  
aws.iot.SiteWiseEdgeProcessor/influxdb
```

PowerShell

```
Copy-Item -Recurse -Force <influxdb_backup_path>\* C:\greengrass  
\v2\work\aws.iot.SiteWiseEdgeProcessor\
```

Windows

```
robocopy <influxdb_backup_path> C:\greengrass\v2\work  
\aws.iot.SiteWiseEdgeProcessor\ /E
```

4. Reinicie el SiteWiseEdgeProcessor componente:

Linux

```
sudo /greengrass/v2/bin/greengrass-cli component restart -n  
aws.iot.SiteWiseEdgeProcessor
```

Windows

```
C:\greengrass\v2\bin\greengrass-cli component restart -n  
aws.iot.SiteWiseEdgeProcessor
```

Validación de copias de seguridad y restauraciones correctas

Utilice este procedimiento para validar los datos respaldados y las restauraciones de la puerta de enlace Edge. SiteWise

Note

Este procedimiento requiere que haya instalado para. AWS OpsHub AWS IoT SiteWise Para obtener más información, consulte [Administración de puertas de enlace SiteWise Edge mediante AWS OpsHub for AWS IoT SiteWise](#).

1. Abierto AWS OpsHub para. AWS IoT SiteWise
2. En la página de configuración de SiteWise Edge Gateway, compruebe el estado de cada componente que aparece en la tabla de componentes. Compruebe que el color del estado sea verde y que la lectura indique RUNNING.
3. Valide los datos pasados en el panel de control del portal para comprobar que tanto los datos pasados como los nuevos estén configurados correctamente. Habrá un tiempo de inactividad entre los datos pasados y nuevos. Debería esperar ver un periodo en el que no se recopilaran puntos de datos.

Si tiene problemas al realizar una copia de seguridad o restaurar una puerta de enlace SiteWise Edge, consulte los siguientes temas de solución de [problemas relacionados con una puerta de enlace AWS IoT SiteWise Edge](#).


Puertas de enlace antiguas (AWS IoT Greengrass Version 1)

Note

SiteWise Las puertas de enlace perimetrales que se ejecutan solo AWS IoT Greengrass V1 están disponibles si comenzó a usar esta función antes del 29 de julio de 2021. Para obtener más información sobre cómo ejecutar una AWS IoT SiteWise puerta de enlace mediante AWS IoT Greengrass V2, consulte [Aloje automáticamente una puerta de enlace AWS IoT SiteWise Edge con AWS IoT Greengrass V2](#).

SiteWise Las puertas de enlace perimetrales ahora se ejecutan exclusivamente AWS IoT Greengrass V2, lo que proporciona una funcionalidad y un rendimiento mejorados para sus aplicaciones de IoT industrial. Esta última versión AWS IoT Greengrass V2 representa una evolución arquitectónica, basada en un marco moderno basado en componentes que permite la implementación de software modular. Simplifica la instalación mediante un instalador unificado y, al mismo tiempo, ofrece a los desarrolladores una mayor flexibilidad a la hora de implementar componentes personalizados y realizar pruebas locales. El modelo basado en componentes permite una administración de recursos más eficiente e introduce un enfoque de configuración simplificado a través de recetas de componentes. Este diseño facilita un mejor manejo de la dependencia entre los componentes, admite prácticas de implementación continua y proporciona capacidades de CLI mejoradas para el desarrollo local. Además, AWS IoT Greengrass V2 centraliza la administración de la configuración AWS IoT Core y ofrece funciones mejoradas de registro y supervisión, todas ellas protegidas por un modelo de permisos de seguridad más detallado.

Para obtener más información sobre cómo empezar a utilizar las puertas de enlace SiteWise Edge mediante AWS IoT Greengrass V2, consulte [AWS IoT SiteWise Requisitos de la puerta de enlace autohospedada de Edge](#). Estos recursos proporcionan step-by-step instrucciones sobre la configuración de sus puertas de enlace, la configuración de las fuentes de datos y la administración de su infraestructura de IoT industrial.

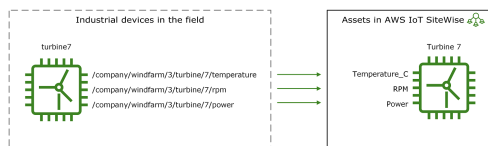
 Note

A medida que AWS continúa innovando y mejorando sus servicios de IoT, se recomienda mantenerse actualizado con las últimas funciones y mejoras. Consulte periódicamente la AWS IoT Greengrass documentación AWS IoT SiteWise y la documentación para conocer las nuevas capacidades que pueden optimizar aún más sus soluciones de IoT industrial.

Modelado de activos industriales

Descripción general de los activos

Puede crear representaciones virtuales de su operación industrial con AWS IoT SiteWise activos. Un activo representa un dispositivo, un equipo o un proceso que carga uno o más flujos de datos a la AWS nube. Por ejemplo, un dispositivo puede ser una turbina eólica que envía mediciones de la temperatura del aire, la velocidad de rotación de la hélice y series temporales de salida de potencia a las propiedades de activos en AWS IoT SiteWise.

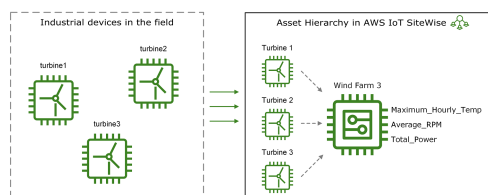


Los alias de propiedad identifican los flujos de datos de los equipos

Cada secuencia de datos corresponde a un alias de propiedad único. Por ejemplo, el alias `/company/windfarm/3/turbine/7/temperature` identifica de forma exclusiva el flujo de datos de temperatura procedente de la turbina número 7 en el parque eólico número 3. Puede configurar AWS IoT SiteWise los activos para transformar los datos de medición entrantes mediante expresiones matemáticas, como convertir los datos de temperatura de grados Celsius a Fahrenheit.

Las jerarquías de activos representan las relaciones entre los equipos

Un activo también puede representar una agrupación lógica de dispositivos, como un parque eólico completo. Puede asociar activos a otros activos para crear jerarquías de activos que representen operaciones industriales complejas. Los activos pueden acceder a los datos de sus activos secundarios asociados. De este modo, puede utilizar AWS IoT SiteWise expresiones para calcular métricas agregadas, como la producción neta de energía de un parque eólico.



Los modelos de activos estandarizan la representación de los equipos

Debe crear todos los activos a partir de un modelo de activos. Los modelos de activos son estructuras declarativas que normalizan el formato de los activos. Los modelos de activos aplican información coherente en varios activos del mismo tipo, de modo que puede procesar datos en activos que representan grupos de dispositivos. Por ejemplo, una planta de fabricación puede tener un modelo de activos para las máquinas CNC que defina propiedades como la temperatura, el tiempo de inactividad y la tasa de producción. En el diagrama anterior, se utiliza el mismo modelo de activos para las tres turbinas porque comparten un conjunto común de propiedades.

Opciones de modelado para equipos industriales

Al diseñar la representación de sus activos industriales, tenga en cuenta estas opciones:

- Los modelos de activos representan tipos específicos de equipos o procesos. Debe crear cada activo físico a partir de un modelo de activos. Por ejemplo, una planta de procesamiento químico puede tener modelos de activos independientes para reactores, mezcladores y tanques de almacenamiento.
- Los modelos de componentes definen subconjuntos reutilizables que se pueden incluir en los modelos de activos u otros modelos de componentes. Por ejemplo, podría incluir un modelo de componentes de sensores de temperatura en varios modelos de activos de equipo de una fábrica.
- Las interfaces de los modelos de activos aplican estándares a distintos modelos de activos. Por ejemplo, una interfaz de «equipo rotativo» podría definir las propiedades estándar de vibración, temperatura y RPM que se aplican a las bombas, turbinas y motores, a pesar de que cada uno tenga su propio modelo de activos único.

Crear y administrar activos

Después de definir los modelos de activos, puede crear los activos industriales. Para crear un activo, seleccione un modelo de activos ACTIVE para crear un activo a partir de ese modelo. A continuación, puede rellenar información específica del activo, como alias y atributos de flujo de datos. En el diagrama anterior, se crean tres activos de turbina a partir de un modelo de activos y, a continuación, se asocian alias de flujo de datos como `/company/windfarm/3/turbine/7/temperature` para cada turbina.

También puede actualizar y suprimir activos, modelos de activos y modelos de componentes existentes. Al actualizar un modelo de activos, todos los activos basados en ese modelo de activos reflejan los cambios que realice en el modelo subyacente. Al actualizar un modelo de componentes, se aplica a todos los activos en función de todos los modelos de activos que hacen referencia al modelo de componentes.

Gestión de modelos de activos complejos

Sus modelos de activos pueden ser muy complejos, por ejemplo, al modelar un equipo complicado que tiene muchos subcomponentes. Para ayudar a mantener estos modelos de activos organizados y fáciles de mantener, puede utilizar modelos compuestos personalizados que agrupen propiedades relacionadas o reutilicen componentes compartidos. Para obtener más información, consulte [Modelos compuestos personalizados \(componentes\)](#).

Estados de activos y modelos

Al crear, actualizar o eliminar un activo, un modelo de activo o un modelo de componentes, los cambios tardan en propagarse. AWS IoT SiteWise resuelve estas operaciones de forma asíncrona y actualiza el estado de cada recurso. Cada activo, modelo de activos y modelo de componentes tiene un campo de estado que contiene el estado del recurso y mensajes de error, si procede. El estado puede ser uno de los siguientes valores:

- **ACTIVE:** el recurso está activo. Este es el único estado en el que puede consultar e interactuar con activos, modelos de activos y modelos de componentes.
- **CREATING:** se está creando el recurso.
- **UPDATING:** se está actualizando el recurso.
- **DELETING:** se está eliminando el recurso.
- **PROPAGATING:** (solo modelos de activos y modelos de componentes) Los cambios se están propagando a todos los recursos dependientes (del modelo de activos a los activos o del modelo de componentes a los modelos de activos).
- **FAILED:** no se ha validado el recurso durante una operación de creación o actualización, posiblemente debido a una referencia circular en una expresión. Puede eliminar los recursos que están en estado FAILED.

Algunas de las operaciones de creación, actualización y eliminación AWS IoT SiteWise colocan un activo, un modelo de activo o un modelo de componente en un estado distinto ACTIVE al de

cuando se resuelve la operación. Para consultar o interactuar con un recurso después de realizar una de estas operaciones, debe esperar hasta que el estado cambie a ACTIVE. De lo contrario, sus solicitudes fallan.

Temas

- [Comprobación del estado de un activo](#)
- [Comprobación del estado de un modelo de activos o de componentes](#)

Comprobación del estado de un activo

Puede utilizar la AWS IoT SiteWise consola o la API para comprobar el estado de un activo.

Temas

- [Comprobación del estado de un activo \(consola\)](#)
- [Comprobación del estado de un activo \(AWS CLI\)](#)

Comprobación del estado de un activo (consola)

Utilice el procedimiento siguiente para comprobar el estado de un activo en la consola de AWS IoT SiteWise .

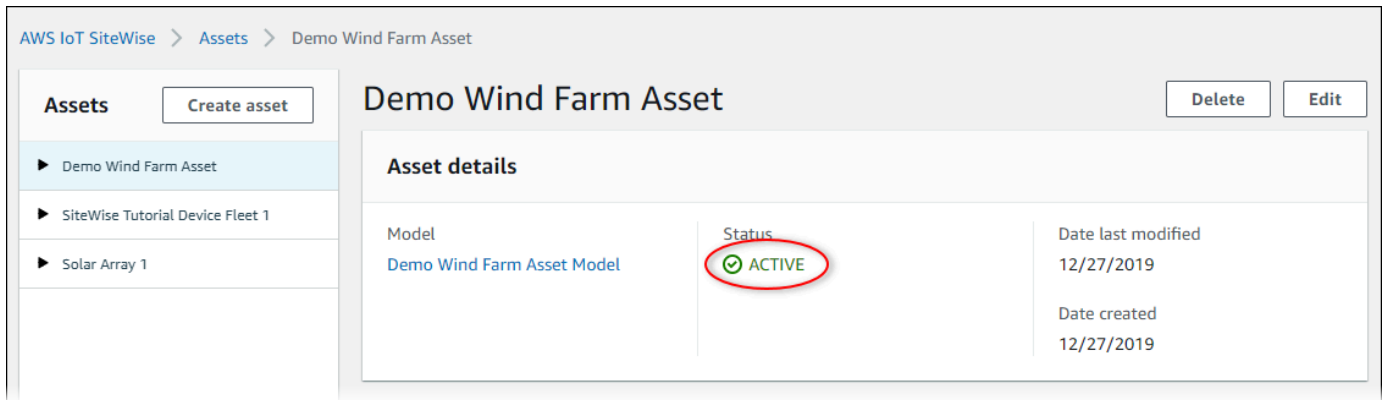
Para comprobar el estado de un activo (consola)

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, elija Activos.
3. Elija el activo que desea comprobar.

Tip

Puede elegir el icono de flecha para expandir una jerarquía de activos y encontrar su activo.

4. Busque Estado en el panel Detalles del activo.



Comprobación del estado de un activo (AWS CLI)

Puede utilizar el AWS Command Line Interface (AWS CLI) para comprobar el estado de un activo.

Para comprobar el estado de un activo, utilice la [DescribeAsset](#) operación con el `assetId` parámetro.

Para comprobar el estado de un activo (AWS CLI)

- Ejecute el siguiente comando para describir el activo. `asset-id` Sustitúyalo por el identificador del activo o el identificador externo. El ID externo es un ID definido por el usuario. Para obtener más información, consulte [Objetos de referencia con objetos externos IDs](#) en la Guía del usuario de AWS IoT SiteWise .

```
aws iotsitewise describe-asset --asset-id asset-id
```

La operación devuelve una respuesta que contiene los detalles del activo. La respuesta contiene un objeto `assetStatus` que tiene la siguiente estructura:

```
{
  ...
  "assetStatus": {
    "state": "String",
    "error": {
      "code": "String",
      "message": "String"
    }
  }
}
```

El estado del activo está en `assetStatus.state` en el objeto JSON.

Comprobación del estado de un modelo de activos o de componentes

Puede utilizar la AWS IoT SiteWise consola o la API para comprobar el estado de un modelo de activos o un modelo de componentes.

Temas

- [Comprobación del estado de un modelo de activos o un modelo de componentes \(consola\)](#)
- [Comprobación del estado de un modelo de activos o un modelo de componentes \(AWS CLI\)](#)

Comprobación del estado de un modelo de activos o un modelo de componentes (consola)

Utilice el siguiente procedimiento para comprobar el estado de un modelo de activos o un modelo de componentes en la AWS IoT SiteWise consola.

Tip

Tanto los modelos de activos como los modelos de componentes se muestran en Modelos en el panel de navegación. El panel Detalles del modelo de activos o del modelo de componentes seleccionado indica de qué tipo se trata.

Comprobación del estado de un modelo de activos o un modelo de componentes (consola)

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, elija Models (Modelos).
3. Elija el modelo que desea comprobar.
4. Busque Estado en el panel Detalles.

The screenshot shows the AWS IoT SiteWise console interface. On the left, there is a 'Models' sidebar with a 'Create model' button and a list of models including 'Demo Turbine Asset Model', 'Demo Wind Farm Asset Model' (highlighted), 'SiteWise Tutorial Device Fleet Model', 'SiteWise Tutorial Device Model', and 'Solar Array'. The main area displays the details for the 'Demo Wind Farm Asset Model'. It includes a 'Delete' and 'Edit' button. The 'Details' section contains a description, a status field showing 'ACTIVE' with a green checkmark icon (circled in red), and a table with 'Date last modified' (12/27/2019) and 'Date created' (12/27/2019).

Comprobación del estado de un modelo de activos o un modelo de componentes (AWS CLI)

Puede utilizarla AWS CLI para comprobar el estado de un modelo de activos o un modelo de componentes.

Para comprobar el estado de un modelo de activos o un modelo de componentes, utilice la [DescribeAssetModel](#) operación con el `assetModelId` parámetro.

Tip

AWS CLI Define los modelos de componentes como un tipo de modelo de activos. Por lo tanto, se utiliza la misma [DescribeAssetModel](#) operación para ambos tipos de modelo. El campo `assetModelType` de la respuesta indica si es un `ASSET_MODEL` o un `COMPONENT_MODEL`.

Comprobación del estado de un modelo de activos o un modelo de componentes (AWS CLI)

- Ejecute el siguiente comando para describir el modelo. `asset-model-id` Sustitúyalo por el ID o el ID externo del modelo de activos o del modelo de componentes. El ID externo es un ID definido por el usuario. Para obtener más información, consulte [Objetos de referencia con objetos externos IDs](#) en la Guía del usuario de AWS IoT SiteWise .

```
aws iotsitewise describe-asset-model --asset-model-id asset-model-id
```

La operación devuelve una respuesta que contiene los detalles del modelo. La respuesta contiene un objeto `assetModelStatus` que tiene la siguiente estructura.

```
{
  ...
  "assetModelStatus": {
    "state": "String",
    "error": {
      "code": "String",
      "message": "String"
    }
  }
}
```

El estado del modelo está en `assetModelStatus.state` en el objeto JSON.

Versiones de modelos de activos

AWS IoT SiteWise admite el procesamiento asíncrono de las operaciones de creación y actualización de modelos de activos y modelos de componentes. También actualiza el estado del modelo.

AWS IoT SiteWise propaga los cambios de un modelo válido en las solicitudes de creación y actualización a sus recursos dependientes (del modelo de activos a los activos o del modelo de componentes a los modelos de activos). A continuación, coloca el modelo en estado ACTIVE.

Si la definición del modelo proporcionada no es válida, AWS IoT SiteWise coloca el modelo en un FAILED estado. Los cambios no se propagan a los recursos dependientes. Los recursos dependientes hacen referencia a la última definición del modelo propagada cuando el modelo estaba en estado ACTIVE.

Según la información anterior, las definiciones de modelos tienen dos tipos de versiones de modelos:

1. Última versión: la definición más reciente aceptada como parte de una solicitud de creación o actualización.
2. Versión activa: la definición más reciente procesada correctamente y el estado del modelo es ACTIVE.

De forma predeterminada, los detalles de la última versión del modelo se muestran cuando APIs se solicita la descripción en un modelo de activos o un modelo de componentes. Hay escenarios en los que se necesita la versión activa del modelo de activos o del modelo de componentes. Consulte escenarios de ejemplo a continuación:

- Una operación de actualización con una definición no válida coloca el modelo de activos en estado FAILED. Debe revertir los cambios recuperando la versión activa del modelo de activos y creando otra solicitud de actualización que haga referencia a esta definición válida.
- AWS IoT SiteWise Existe una aplicación en la que los clientes pueden ver los activos y sus modelos de activos correspondientes. Cuando un usuario hace referencia a la definición del modelo de activos correspondiente a un activo concreto y el modelo de activos se encuentra en estado UPDATING, PROPAGATING o FAILED transitorio, la versión más reciente devuelve la definición del modelo de activos que aún no se ha propagado a sus activos. En este caso, debe recuperar la versión activa del modelo de activos a los clientes.

Temas

- [Recupere la versión activa de un modelo de activos o un modelo de componentes \(consola\).](#)
- [Recupere la versión activa de un modelo de activos o un modelo de componentes \(AWS CLI\).](#)

Recupere la versión activa de un modelo de activos o un modelo de componentes (consola).

Siga este procedimiento para recuperar la versión activa de un modelo de activos o un modelo de componentes en la AWS IoT SiteWise consola.

Tip

Tanto los modelos de activos como los modelos de componentes se muestran en Modelos en el panel de navegación. El panel Detalles del modelo de activos o del modelo de componentes seleccionado indica de qué tipo se trata.

Recuperación de la versión activa de un modelo de activos o un modelo de componentes (consola)

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, elija Models (Modelos).
3. Elija el modelo para el que desea recuperar su versión activa.
 - a. Si el modelo está en estado ACTIVE, está viendo su versión activa.
 - b. Si el modelo se encuentra en estado UPDATING, PROPAGATING o FAILED transitorio, busque la opción Ver la versión activa en Estado en el panel Detalles.

Recupere la versión activa de un modelo de activos o un modelo de componentes (AWS CLI).

Utilice el AWS CLI para recuperar la versión activa de un modelo de activos o un modelo de componentes.

Para recuperar la versión activa de un modelo de activos o un modelo de componentes, utilice la [DescribeAssetModel](#) operación con el `assetModelVersion` parámetro.

Tip

AWS CLI Define los modelos de componentes como un tipo de modelo de activos. Por lo tanto, se utiliza la misma [DescribeAssetModel](#) operación para ambos tipos de modelo. El campo `assetModelType` de la respuesta indica si es un `ASSET_MODEL` o un `COMPONENT_MODEL`.

Recuperación de la versión activa de un modelo de activos o un modelo de componentes (AWS CLI)

- Ejecute el siguiente comando para describir el modelo. `asset-model-id` Sustitúyalo por el ID o el ID externo del modelo de activos o del modelo de componentes. El ID externo es un ID definido por el usuario. Para obtener más información, consulte [Objetos de referencia con objetos externos IDs](#) en la Guía del usuario de AWS IoT SiteWise .

```
aws iotsitewise describe-asset-model --asset-model-id asset-model-id --asset-model-version ACTIVE
```

La operación devuelve una respuesta con los detalles del modelo. La respuesta contiene un objeto `assetModelStatus` con la siguiente estructura.

```
{
  ...
  "assetModelName": "string",
  "assetModelProperties": [ ... ],
  ...,
  "assetModelVersion": "string"
}
```

Modelos compuestos personalizados (componentes)

Al modelar un activo industrial especialmente complejo, como una pieza de maquinaria complicada que consta de muchas piezas, mantener los modelos de activos organizados y fáciles de mantener puede convertirse en un desafío.

En esos casos, puede añadir modelos compuestos personalizados, o componentes si utiliza la consola, a sus modelos de activos y componentes existentes. Esto le ayuda a mantenerse organizado agrupando propiedades relacionadas y reutilizando definiciones de subcomponentes.

Hay dos tipos de modelos compuestos personalizados:

- Los modelos compuestos personalizados en línea definen un conjunto de propiedades agrupadas que se aplican al modelo de activos o al modelo de componentes al que pertenece el modelo compuesto personalizado. Sirven para agrupar propiedades relacionadas. Constan de un nombre, una descripción y un conjunto de propiedades del modelo de activos. No son reutilizables.
- Los modelos compuestos component-model-based personalizados en C hacen referencia a un modelo de componentes que desee incluir en su modelo de activos o modelo de componentes. Se utilizan para incluir subconjuntos estándar en el modelo. Constan de un nombre, una descripción y el ID del modelo de componentes al que hacen referencia. No tienen propiedades propias; el modelo de componentes al que se hace referencia proporciona las propiedades asociadas a los activos creados.

Las siguientes secciones ilustran cómo utilizar modelos compuestos personalizados en sus diseños.

Temas

- [Modelos compuestos personalizados en línea](#)
- [Component-model-based modelos compuestos personalizados](#)
- [Uso de rutas para hacer referencia a propiedades de modelos compuestos personalizados](#)

Modelos compuestos personalizados en línea

Los modelos compuestos personalizados en línea proporcionan una forma de organizar el modelo de activos agrupando propiedades relacionadas.

Por ejemplo, supongamos que desea modelar un activo de robot. El robot incluye un servomotor, una fuente de alimentación y una batería. Cada una de esas partes constituyentes tiene sus propias

propiedades que conviene incluir en el modelo. Puede definir un modelo de activos denominado `robot_model` que tenga propiedades como las siguientes.

- `robot_model`
 - `servo_status` (entero)
 - `servo_position` (doble)
 - `powersupply_status` (entero)
 - `powersupply_temperature` (doble)
 - `battery_status` (entero)
 - `battery_charge` (doble)

Sin embargo, en algunos casos es posible que haya muchos subconjuntos o que los propios subconjuntos tengan muchas propiedades. En estos casos, es posible que haya tantas propiedades que resulte difícil hacer referencia a ellas y mantenerlas en una sola lista sin formato en la raíz del modelo, como en el ejemplo anterior.

Para hacer frente a estas situaciones puede utilizar un modelo compuesto personalizado en línea para agrupar propiedades. Un modelo compuesto personalizado en línea es un modelo compuesto personalizado que define sus propias propiedades. Por ejemplo, puede modelar su robot de la siguiente manera.

- `robot_model`
 - `servo`
 - `status` (entero)
 - `position` (doble)
 - `powersupply`
 - `status` (entero)
 - `temperature` (doble)
 - `battery`
 - `status` (entero)
 - `charge` (doble)

En el ejemplo anterior, `servo`, `powersupply` y `battery` son los nombres de los modelos compuestos personalizados en línea definidos en el modelo de activos de `robot_model`. A continuación, cada uno de estos modelos compuestos define sus propias propiedades.

Note

En este caso, cada modelo compuesto personalizado define sus propias propiedades, de modo que todas las propiedades forman parte del propio modelo de activos (`robot_model` en este caso). Estas propiedades no se comparten con ningún otro modelo de activos o de componentes. Por ejemplo, si ha creado algún otro modelo de activos que también tuviera un modelo compuesto personalizado en línea denominado `servo`, realizar un cambio en el `servo` dentro de `robot_model` no afectaría a la definición de `servo` del otro modelo de activos.

Si desea implementar este tipo de compartición (por ejemplo, tener solo una definición para un `servo` que puedan compartir todos sus modelos de activos), debería crear un modelo de componentes para él y, a continuación, crear modelos `component-model-based` compuestos que hagan referencia a él. Consulte la siguiente sección para obtener más información.

Para obtener información sobre cómo crear modelos compuestos personalizados en línea, consulte [Creación de modelos compuestos personalizados \(componentes\)](#).

Component-model-based modelos compuestos personalizados

Puede crear un modelo de componente AWS IoT SiteWise para definir un subensamblaje estándar y reutilizable. Una vez creado un modelo de componentes, puede añadirle referencias en otros modelos de activos y modelos de componentes. Para ello, añada un modelo compuesto `component-model-based` personalizado a cualquier modelo en el que desee hacer referencia al componente. Puede añadir referencias al componente desde numerosos modelos o varias veces dentro del mismo modelo.

De esta forma, puede evitar la duplicación de las mismas definiciones en todos los modelos. También se simplifica el mantenimiento de los modelos, ya que cualquier cambio que realice en un modelo de componentes se reflejará en todos los modelos de activos que lo utilicen.

Por ejemplo, supongamos que su instalación industrial tiene muchos tipos de equipos y todos utilizan el mismo tipo de servomotor. Algunos de ellos tienen muchos servomotores en un solo equipo. Cree un modelo de activos para cada tipo de equipo, pero no conviene duplicar la definición de `servo`

cada vez. Es mejor modelarlo solo una vez y usarlo en los distintos modelos de activos. Si más adelante cambia la definición de `servo`, se actualizará en todos los modelos y los activos.

Para modelar el robot del ejemplo anterior de esta manera puede definir servomotores, fuentes de alimentación y baterías como modelos de componentes, de este modo.

- `servo_component_model`
 - `status` (entero)
 - `position` (doble)

- `powersupply_component_model`
 - `status` (entero)
 - `temperature` (doble)

- `battery__component_model`
 - `status` (entero)
 - `charge` (doble)

A continuación, puede definir modelos de activos, por ejemplo `robot_model`, que hagan referencia a estos componentes. Varios modelos de activos pueden hacer referencia al mismo modelo de componentes. También puede hacer referencia al mismo modelo de componentes varias veces en un modelo de activos, por ejemplo, si el robot tiene varios servomotores.

- `robot_model`
 - `servo1` (referencia: `servo_component_model`)
 - `servo2` (referencia: `servo_component_model`)
 - `servo3` (referencia: `servo_component_model`)
 - `powersupply` (referencia: `powersupply_component_model`)
 - `battery` (referencia: `battery_component_model`)

Para obtener información sobre cómo crear modelos de componentes, consulte [Creación de modelos de componentes de](#).

Para obtener información sobre cómo hacer referencia a sus modelos de componentes en otros modelos, consulte [Creación de modelos compuestos personalizados \(componentes\)](#).

Uso de rutas para hacer referencia a propiedades de modelos compuestos personalizados

Al crear una propiedad en un modelo de activos, un modelo de componentes o un modelo compuesto personalizado, puede hacer referencia a ella desde otras propiedades que utilizan su valor, como [transformaciones](#) y [métricas](#).

AWS IoT SiteWise proporciona diferentes formas de hacer referencia a su propiedad. La forma más sencilla suele ser utilizar su ID de propiedad. Sin embargo, si la propiedad a la que desea hacer referencia está en un modelo compuesto personalizado, puede que le resulte más útil hacer referencia a ella mediante una ruta.

Una ruta es una secuencia ordenada de segmentos de ruta que especifica una propiedad en términos de su posición entre los modelos compuestos anidados dentro de un modelo de activos y un modelo compuesto.

Obtención de rutas de propiedades

Puede obtener la ruta de una propiedad desde el path campo de su propiedad [AssetModelProperty](#).

Por ejemplo, supongamos que tiene un modelo de activos `robot_model` que contiene un modelo compuesto personalizado `servo`, que tiene una propiedad `position`. Si [DescribeAssetModelCompositeModel](#) llama a `servo`, la `position` propiedad mostrará un path campo similar al siguiente:

```
"path": [  
  {  
    "id": "asset model ID",  
    "name": "robot_model"  
  },  
  {  
    "id": "composite model ID",  
    "name": "servo"  
  },  
  {
```

```
    "id": "property ID",
    "name": "position"
  }
]
```

Uso de rutas de propiedades

Puede usar una ruta de propiedad al definir una propiedad que haga referencia a otras propiedades, como una transformación o una métrica.

Una propiedad usa una variable para hacer referencia a otra propiedad. Para obtener más información sobre cómo trabajar con variables, consulte [Uso de variables en expresiones de fórmula](#).

Al definir una variable para hacer referencia a una propiedad, puede utilizar el ID de la propiedad o su ruta.

Para definir una variable que utilice la ruta de la propiedad a la que se hace referencia, especifique el campo `propertyPath` de su valor.

Por ejemplo, para definir un modelo de activos que tenga una métrica que haga referencia a una propiedad mediante una ruta, podrías pasar una carga útil como esta a [CreateAssetModel](#):

```
{
  ...
  "assetModelProperties": [
    {
      ...
      "type": {
        "metric": {
          ...
          "variables": [
            {
              "name": "variable name",
              "value": {
                "propertyPath": [
                  path segments
                ]
              }
            }
          ],
          ...
        }
      },
      ...
    }
  ],
  ...
}
```

```
    },  
    ...  
  ],  
  ...  
}
```

Interfaces del modelo de activos

AWS IoT SiteWise las interfaces establecen estándares en los diferentes modelos de activos. Definen una estructura común que garantiza la coherencia y, al mismo tiempo, permite variaciones en la implementación.

Las interfaces comparten la misma estructura que los modelos de activos (propiedades, modelos compuestos y jerarquías), pero no se pueden crear activos directamente a partir de ellas. En su lugar, las interfaces se aplican a los modelos de activos existentes para garantizar la estandarización. Los modelos de componentes no son compatibles con las interfaces.

El uso de interfaces ofrece varias ventajas:

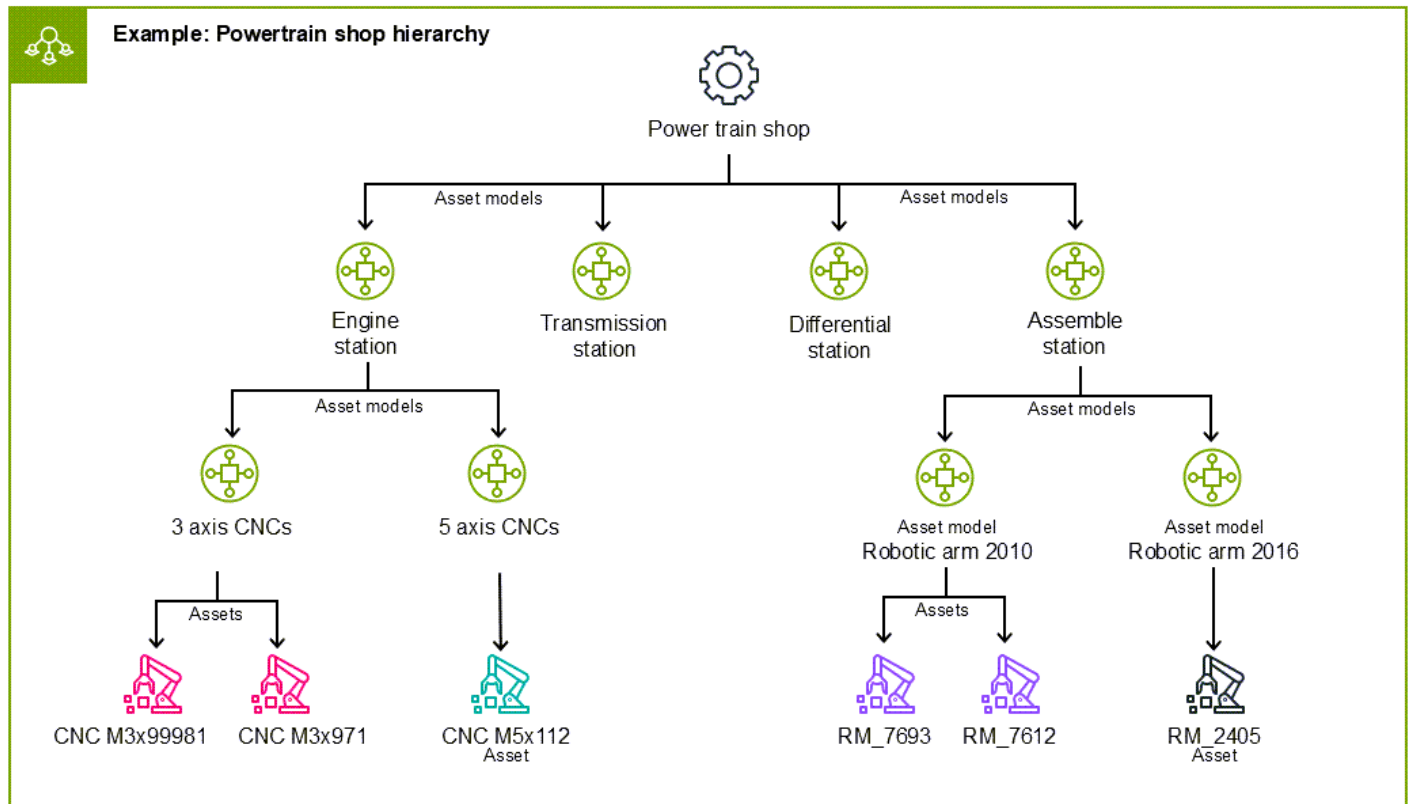
- Propiedades y métricas estandarizadas en diferentes variaciones del modelo de activos
- Definiciones de métricas simplificadas a nivel de interfaz
- Gestión más eficiente de jerarquías de activos complejas
- Gestión independiente del ciclo de vida de la propiedad para cada variación del modelo de activos
- Colaboración mejorada entre equipos, donde los equipos de operaciones se centran en la representación física de los activos, mientras que los equipos de datos establecen estándares para todos los equipos

Le recomendamos que cree primero sus modelos de activos para modelar su equipo industrial real. Cada tipo de equipo, con su propio conjunto de propiedades, se puede representar mediante sus propios modelos de activos.

Caso de uso de estandarización de modelos de activos

Las interfaces ayudan a estandarizar las propiedades en los diferentes modelos de activos, a la vez que conservan sus características únicas.

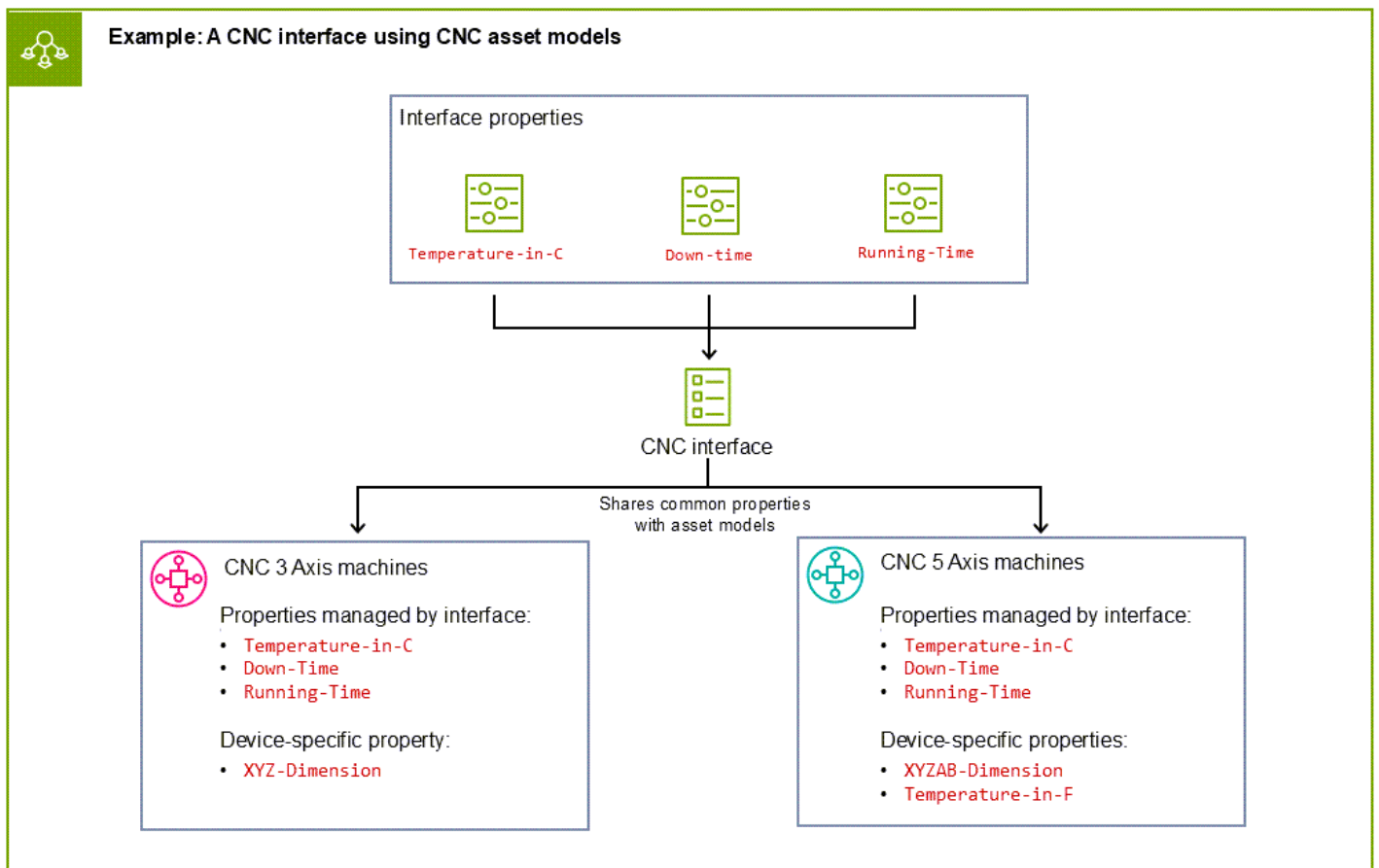
Por ejemplo, en un taller de sistemas de propulsión hay cuatro estaciones: motor, transmisión, diferencial y ensamblaje. Cada estación contiene varios tipos de equipos. Por ejemplo, la estación de máquinas incluye máquinas CNC, pero difieren en las especificaciones: algunas son de 3 ejes, mientras que otras son de 5 ejes.



Sin embargo, las interfaces permiten crear estándares para los puntos en común que se ven en las máquinas CNC. Puede utilizar las propiedades repetibles en una interfaz en lugar de crear modelos de activos para cada propiedad.

Por ejemplo, puede hacer lo siguiente:

1. Cree modelos de activos independientes para cada categoría de tipos de máquinas. En este ejemplo, se trata de las «máquinas CNC de 3 ejes» y las «máquinas CNC de 5 ejes».
2. Defina una interfaz estándar con propiedades y métricas comunes. En este ejemplo, `Temperature-in-CDown-time`, y `Running-time` son todas propiedades comunes que se aplican a ambas máquinas CNC.
3. Aplique esta interfaz a todos los modelos de activos de máquinas CNC y, al mismo tiempo, permita incluir propiedades específicas del dispositivo en los modelos de activos individuales.



También puede definir las métricas de disponibilidad a nivel de interfaz. Por ejemplo, $Avail = \text{avg}(\text{Down-time}, \text{Running-time})$ calcula la disponibilidad en función de los valores del tiempo de inactividad y el tiempo de ejecución.

El uso de interfaces simplifica la gestión del modelo de activos al garantizar la coherencia de las definiciones y métricas de las propiedades en todos los equipos aplicables y, al mismo tiempo, mantener las características únicas de cada tipo de máquina.

Estructura y componentes

Las interfaces incluyen los mismos tipos de propiedades que los modelos de activos: atributos, medidas, transformaciones y métricas. Cuando se superponen a un modelo de activos, se asignan las propiedades existentes a sus homólogas de la interfaz. Las propiedades de la interfaz no mapeadas se crean automáticamente en el modelo de activos.

Las jerarquías de la interfaz definen las métricas acumulativas, mientras que las jerarquías del modelo de activos permiten las asociaciones de activos. Cuando utilice una interfaz, el servicio

asignará automáticamente las jerarquías de los modelos de activos a las jerarquías de la interfaz al calcular las métricas acumulativas. Tras aplicar una interfaz, las métricas acumulativas se definen mediante la jerarquía de la interfaz y no mediante la propia jerarquía del modelo de activos.

Consideraciones

Al trabajar con interfaces, tenga en cuenta estas consideraciones:

- Las propiedades del modelo de activos y de la interfaz se pueden mapear automáticamente por nombre o se pueden mapear manualmente. El servicio mapea automáticamente las jerarquías al calcular las métricas acumulativas.
- No se pueden definir métricas adicionales en el modelo de activos vinculados que utilicen métricas de interfaz como entradas.
- Un modelo de activos solo se puede vincular a una interfaz. Sin embargo, puede aplicar varios modelos de activos a la misma interfaz.
- Las alarmas no se admiten en las interfaces.
- Los modelos de componentes no son compatibles con las interfaces.

Temas

- [Comprenda la relación entre la interfaz y el modelo de activos](#)
- [Cree una interfaz](#)
- [Aplicar una interfaz a un modelo de activos](#)
- [Administre las interfaces, los modelos de activos vinculados y las propiedades](#)
- [Ejemplos de interfaz adicionales](#)

Comprenda la relación entre la interfaz y el modelo de activos

Las interfaces y los modelos de activos funcionan juntos en una relación complementaria:

Interfaces frente a modelos de activos

Aspecto	Interfaces	Modelos de activos
Finalidad	Defina los estándares y aplique la coherencia	Represente activos físicos o lógicos

Aspecto	Interfaces	Modelos de activos
Creación de activos	No se pueden crear activos directamente	Se utiliza para crear activos
Propiedades	Defina las propiedades estándar que deben implementarse en los modelos	Puede tener propiedades únicas y aplicadas a la interfaz
Métricas	Defina los cálculos estándar	Implemente métricas de interfaz y puede tener métricas adicionales
Jerarquías	Defina las relaciones jerárquicas del cálculo de datos para las métricas acumulativas	Defina las relaciones jerárquicas físicas para las asociaciones de activos

Al aplicar una interfaz a un modelo de activos:

- El modelo de activos debe mapear todas las propiedades definidas en la interfaz.
- Las asignaciones de propiedades definen cómo las propiedades de la interfaz se corresponden con las propiedades del modelo de activos.
- Las propiedades del modelo de activos mapeado deben permanecer sincronizadas con sus propiedades de interfaz correspondientes y no se pueden modificar de forma que se produzca una incoherencia entre ambas.
- Las propiedades de interfaz no mapeadas se crean automáticamente en el modelo de activos.
- El modelo de activos puede tener propiedades adicionales además de las definidas en la interfaz.
- El modelo de activos implementa métricas de interfaz. Los cambios en las métricas de la interfaz se propagan a todos los modelos de activos que utilizan la interfaz.
- Las jerarquías de interfaz se utilizan para calcular las métricas acumulativas. Las jerarquías de los modelos de activos se pueden definir de forma independiente y el servicio las mapeará automáticamente al calcular las métricas acumulativas.

Esta relación garantiza la estandarización y, al mismo tiempo, permite la flexibilidad necesaria para representar diversos tipos de equipos.

Estandarice los modelos de activos existentes

Si bien las interfaces son valiosas a la hora de diseñar nuevos modelos de activos desde cero, son igualmente eficaces para estandarizar los modelos de activos existentes que pueden haber evolucionado de forma independiente con el tiempo.

Al trabajar con los modelos de activos existentes, puede aplicar interfaces para estandarizar las métricas y las propiedades:

1. Identifique las métricas y propiedades comunes en sus modelos de activos existentes
2. Cree una interfaz que defina estas propiedades y métricas estándar
3. Aplique la interfaz a sus modelos de activos existentes mediante el mapeo de propiedades
4. Utilice métricas acumulativas para agregar datos en toda su jerarquía de activos

Por ejemplo, si ya tiene modelos de activos de máquinas CNC con nombres de propiedades diferentes pero datos similares (por ejemplo `temp_celsius`, `temperature_c`, `machine_temp`), puede:

1. Cree un `CNC-INTERFACE` con una `Temperature-in-C` propiedad estandarizada
2. Aplique esta interfaz a cada modelo de activo CNC, mapeando las propiedades de temperatura existentes con las `Temperature-in-C` propiedades de la interfaz
3. Defina métricas acumulativas en la interfaz que calculen las estadísticas de todas las máquinas (por ejemplo, la temperatura media)

Este enfoque le permite mantener sus modelos de activos existentes y, al mismo tiempo, aprovechar las ventajas de la estandarización y el cálculo simplificado de las métricas.

Relaciones jerárquicas

Jerarquía de interfaz

Define las relaciones para calcular y agregar datos en diferentes interfaces. Por ejemplo, en un entorno de fábrica, una jerarquía de interfaces podría conectar las interfaces de monitoreo de temperatura en diferentes niveles para calcular las temperaturas promedio. Por ejemplo: máquina, línea de producción e instalación. Al definir una métrica acumulativa `AverageTemperature`, la jerarquía de la interfaz determina cómo esa métrica agrega los datos de los niveles inferiores a los niveles superiores.

Jerarquía del modelo de activos

Representa la estructura física o lógica real de sus activos. Por ejemplo, un modelo de activos de una máquina CNC puede ser parte de un modelo de activos de línea de producción, que a su vez pertenece a un modelo de activos de fábrica. Esta jerarquía refleja las relaciones del mundo real y ayuda a organizar los activos de manera que coincidan con su disposición física o estructura empresarial. Cuando se combinan con las jerarquías de interfaz, las jerarquías de los modelos de activos ayudan al sistema a entender qué activos deben incluirse en los cálculos acumulativos.

Estos dos tipos de jerarquías funcionan juntos: las jerarquías de interfaz definen cómo calcular las métricas agregadas, mientras que las jerarquías de los modelos de activos definen qué activos específicos deben incluirse en esos cálculos.

Métricas de interfaz y cálculos acumulativos

Las interfaces se destacan a la hora de definir métricas estandarizadas que se pueden aplicar a diferentes modelos de activos. Esto es particularmente valioso para las métricas acumulativas que agregan datos de varios activos.

Al definir las métricas en una interfaz, se aplican automáticamente a todos los modelos de activos que implementan la interfaz. Las métricas pueden hacer referencia a las propiedades definidas en la interfaz, utilizar funciones de agregación para calcular las estadísticas de los activos y garantizar la coherencia de los cálculos en todos los modelos de activos implementados. Por ejemplo, puede definir una métrica de disponibilidad en una interfaz que calcule la relación entre el tiempo de ejecución y el tiempo total:

```
{
  "name": "Availability",
  "dataType": "DOUBLE",
  "type": {
    "metric": {
      "expression": "Running-time / (Running-time + Down-time) * 100",
      "variables": [
        {
          "name": "Running-time",
          "value": {
            "propertyId": "${Running-time}"
          }
        }
      ],
    },
  }
}
```

```

        "name": "Down-time",
        "value": {
            "propertyId": "${Down-time}"
        }
    },
    "window": {
        "tumbling": {
            "interval": "1h"
        }
    }
},
"unit": "Percent"
}

```

Cuando esta interfaz se aplica a varios modelos de activos, la métrica de disponibilidad se calcula de forma coherente para todos ellos, incluso si los nombres de las propiedades subyacentes son diferentes (gracias al mapeo de propiedades).

Para obtener más información sobre la definición de métricas y el uso de las funciones de agregación, consulte [Agregación de datos desde propiedades y otros activos \(métricas\)](#).

Acumule métricas con interfaces

Las interfaces también pueden definir métricas acumulativas que agregan datos de todos los activos en una jerarquía. Al definir una jerarquía en una interfaz y aplicarla a un modelo de activos, puede crear métricas que agreguen datos de activos secundarios.

Por ejemplo, puede definir una métrica que calcule la temperatura media de todas las máquinas CNC de una fábrica:

```

{
  "name": "AverageTemperature",
  "dataType": "DOUBLE",
  "type": {
    "metric": {
      "expression": "avg(Temperature-in-C)",
      "variables": [
        {
          "name": "Temperature-in-C",
          "value": {

```

```
        "propertyId": "${Temperature-in-C}",
        "hierarchyId": "${CNC-machines}"
    }
  ],
  "window": {
    "tumbling": {
      "interval": "1h"
    }
  }
},
"unit": "Celsius"
}
```

Esta métrica utiliza la función de `avg()` agregación para calcular la temperatura media de todas las máquinas CNC de la jerarquía. El `hierarchyId` parámetro especifica qué jerarquía usar para la agregación.

Cuando esta interfaz se aplica a un modelo de activos, la métrica acumulativa agrega automáticamente los datos de todos los activos secundarios que coinciden con la asignación jerárquica.

Cree una interfaz

Puede crear interfaces mediante la AWS IoT SiteWise consola o el AWS CLI.

Console

1. Navegue hasta la [AWS IoT SiteWise consola](#) y elija Modelos en el panel de navegación.
2. Seleccione Crear interfaz.
3. Introduzca un nombre único y una descripción opcional para la interfaz. Si lo desea, también puede añadir un identificador externo de su elección.
4. Añada propiedades a su interfaz. Puede añadir atributos, medidas, transformaciones y métricas igual que con los modelos de activos. Para obtener más información, consulte [Creación de un modelo de activos \(consola\)](#).
5. Elija Crear interfaz para crear la interfaz.
6. Si dispone de jerarquías para definir las relaciones padre-hijo entre las interfaces, elija Añadir jerarquía e introduzca los detalles pertinentes.

AWS CLI

Para crear una interfaz, utilice la `CreateAssetModel` operación con el `assetModelType` parámetro establecido en: `INTERFACE`

```
aws iotsitewise create-asset-model \
  --asset-model-name "CNC-INTERFACE" \
  --asset-model-description "Standard interface for CNC machines" \
  --asset-model-type "INTERFACE" \
  --asset-model-properties '[
    {
      "name": "Temperature-in-C",
      "dataType": "DOUBLE",
      "type": {
        "measurement": {}
      },
      "unit": "Celsius"
    },
    {
      "name": "Down-time",
      "dataType": "DOUBLE",
      "type": {
        "measurement": {}
      },
      "unit": "Minutes"
    },
    {
      "name": "Running-time",
      "dataType": "DOUBLE",
      "type": {
        "measurement": {}
      },
      "unit": "Minutes"
    },
    {
      "name": "Availability",
      "dataType": "DOUBLE",
      "type": {
        "metric": {
          "expression": "Running-time / (Running-time + Down-time) * 100",
          "variables": [
            {
              "name": "Running-time",
              "value": {
```

```
        "propertyId": "${Running-time}"
      }
    },
    {
      "name": "Down-time",
      "value": {
        "propertyId": "${Down-time}"
      }
    }
  ],
  "window": {
    "tumbling": {
      "interval": "1h"
    }
  }
},
"unit": "Percent"
}
```

Aplicar una interfaz a un modelo de activos

Al aplicar una interfaz a un modelo de activos, se asignan las propiedades y jerarquías del modelo de activos a sus homólogos de la interfaz. En el caso de las propiedades de interfaz no mapeadas, las propiedades correspondientes se crean automáticamente en el modelo de activos. Tras la vinculación, el servicio evita cambios en el modelo de activos que puedan infringir los estándares de la interfaz.

Puede añadir un modelo de activos a una interfaz a la vez. Sin embargo, se pueden vincular varios modelos de activos a una sola interfaz.

Console

1. Navegue hasta la [AWS IoT SiteWise consola](#) y elija Modelos en el panel de navegación.
2. Seleccione el modelo de activo al que desee aplicar una interfaz.
3. Elija Vincular el modelo de activos en la sección Vincular modelos de activos. Aparece la página de la interfaz de enlace.
4. En la sección Modelos e interfaces de activos, seleccione un modelo de activos en el menú desplegable Seleccione un modelo para vincular.

5. En la sección Asignaciones de propiedades, asigne cada propiedad de la interfaz a una propiedad del modelo de activos existente o cree una nueva propiedad. AWS IoT SiteWise vincula automáticamente las propiedades con nombres coincidentes en el modelo de activos y la interfaz.
6. Revise las asignaciones de propiedades y elija Interfaz de enlace.

AWS CLI

Para aplicar una interfaz a un modelo de activos, utilice la `PutAssetModelInterfaceRelationship` operación:

```
aws iotsitewise put-asset-model-interface-relationship \  
  --asset-model-id "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE" \  
  --interface-asset-model-id "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE" \  
  --property-mapping-configuration '{  
    "createMissingProperty": true,  
    "matchByPropertyName": true,  
    "overrides": [  
      {  
        "assetModelPropertyId": "a1b2c3d4-5678-90ab-cdef-44444EXAMPLE",  
        "interfaceAssetModelPropertyId": "a1b2c3d4-5678-90ab-cdef-33333EXAMPLE"  
      }  
    ]  
  }'
```

Para recuperar información sobre una relación de interfaz, utilice la `DescribeAssetModelInterfaceRelationship` operación:

```
aws iotsitewise describe-asset-model-interface-relationship \  
  --asset-model-id "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE" \  
  --interface-asset-model-id "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE"
```

Para enumerar todos los modelos de activos a los que se les ha aplicado una interfaz específica, utilice la `ListInterfaceRelationships` operación:

```
aws iotsitewise list-interface-relationships \  
  --interface-asset-model-id "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE" \  
  --max-results 10
```

Para eliminar una relación de interfaz, utilice la `DeleteAssetModelInterfaceRelationship` operación:

```
aws iotsitewise delete-asset-model-interface-relationship \  
  --asset-model-id "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE" \  
  --interface-asset-model-id "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE"
```

Administre las interfaces, los modelos de activos vinculados y las propiedades

Tras crear las interfaces y vincularlas a los modelos de activos, puede gestionar las relaciones, editar y eliminar las interfaces a través de la consola o AWS CLI.

Modificar una relación entre la interfaz y el modelo de activos

Para cambiar la relación de una interfaz con un modelo de activos, haga lo siguiente en la AWS IoT SiteWise consola o mediante AWS CLI:

Console

1. Navegue hasta la [AWS IoT SiteWise consola](#) y elija Modelos en el panel de navegación.
2. Seleccione la interfaz que desee modificar.
3. Elija el modelo de activo que desee modificar y editar.

Puede seguir las [Aplicar una interfaz a un modelo de activos](#) instrucciones para vincular un modelo de activos diferente.

4. Seleccione Aplicar interfaz para guardar los cambios.

AWS CLI

Para editar una relación entre la interfaz y el modelo de activo, utilice la `PutAssetModelInterfaceRelationship` acción. Reemplace *your-asset-model-id* y *your-interface-asset-model-id* por sus propios valores. Para obtener más información, consulta [PutAssetModelInterfaceRelationship](#) en la AWS IoT SiteWise Referencia de la API de .

```
aws iotsitewise put-asset-model-interface-relationship \  
  --asset-model-id your-asset-model-id \  
  --interface-asset-model-id your-interface-asset-model-id
```

```
--interface-asset-model-id your-interface-asset-model-id
```

Modifique el mapeo de propiedades de una interfaz

Para cambiar una propiedad de la interfaz, haga lo siguiente en la AWS IoT SiteWise consola o mediante AWS CLI:

Console

1. Navegue hasta la [AWS IoT SiteWise consola](#) y elija Modelos en el panel de navegación.
2. Seleccione la interfaz para la que desee modificar las asignaciones de propiedades. Aparece la página Editar asignaciones de propiedades.
3. En la sección Asignaciones de propiedades, filtre la lista para encontrar las asignaciones de propiedades adecuadas.
4. Cambie las propiedades mediante la columna de propiedades del modelo.

AWS CLI

Para editar una relación entre la interfaz y el modelo de activo, utilice la `PutAssetModelInterfaceRelationship` acción. Reemplace *your-asset-model-id* y *your-interface-asset-model-id* por sus propios valores. Para obtener más información, consulta [PutAssetModelInterfaceRelationship](#) en la AWS IoT SiteWise Referencia de la API de .

```
aws iotsitewise put-asset-model-interface-relationship \  
  --asset-model-id your-asset-model-id \  
  --interface-asset-model-id your-interface-asset-model-id \  
  --
```

Enumere las interfaces vinculadas a un modelo de activos

Para obtener una lista de las interfaces aplicadas a un modelo de activos, haga lo siguiente en la AWS IoT SiteWise consola o mediante AWS CLI:

Console

1. Navegue hasta la [AWS IoT SiteWise consola](#) y elija Modelos en el panel de navegación.

2. En la sección Modelos, elija el modelo de activo o la interfaz adecuados. Puede ver una lista de interfaces aplicadas o modelos de activos vinculados en la página de detalles correspondiente del modelo.
 - Al ver una interfaz concreta, consulte la sección Modelos de activos vinculados.
 - Al ver un modelo de activo concreto, consulte la sección Interfaces aplicadas.

AWS CLI

Para enumerar las interfaces, puede utilizar la `ListInterfaceRelationships` operación. Reemplace *your-interface-asset-model-id* por su propio valor. Para obtener más información, consulta [ListInterfaceRelationships](#) en la AWS IoT SiteWise Referencia de la API de .

```
aws iotsitewise list-interface-relationships \
  --interface-asset-model-id your-interface-asset-model-id \
  [--next-token your-next-token] \
  [--max-results 20]
```

Vea los detalles de una relación entre la interfaz y el modelo de activo

Para ver los detalles de una interfaz aplicada a un modelo de activos, haga lo siguiente en la AWS IoT SiteWise consola o a través de AWS CLI:

Console

Vea los detalles de las interfaces aplicadas y los modelos de activos vinculados.

1. Navegue hasta la [AWS IoT SiteWise consola](#) y elija Modelos en el panel de navegación.
2. En la sección Modelos, busque el modelo de activo o la interfaz adecuados. Seleccione el nombre del modelo o la interfaz para abrir una página con más detalles.

AWS CLI

Para ver los detalles de la interfaz de una relación entre la interfaz y el modelo de activo, utilice la `DescribeAssetModelInterfaceRelationship` acción. Reemplace *your-asset-model-id* y *your-interface-asset-model-id* por sus propios valores. Para obtener más información, consulta [DescribeAssetModelInterfaceRelationship](#) en la AWS IoT SiteWise Referencia de la API de .

```
aws iotsitewise describe-asset-model-interface-relationship \  
  --asset-model-id your-asset-model-id \  
  --interface-asset-model-id your-interface-asset-model-id
```

Elimine una interfaz aplicada a un modelo de activos

Para eliminar una interfaz aplicada a un modelo de activos, haga lo siguiente en la AWS IoT SiteWise consola o a través de AWS CLI:

Console

Recomendamos eliminar una interfaz a través del modelo de activos. También puede eliminar una interfaz o desvincular una interfaz a través de la página de una interfaz concreta.

1. Navegue hasta la [AWS IoT SiteWise consola](#) y elija Modelos en el panel de navegación.
2. Seleccione el modelo de activo adecuado del que desee eliminar la relación de interfaz.
3. Elija Desvincular el modelo de activos.

AWS CLI

Para eliminar una relación de interfaz de un modelo de activos, puede utilizar la `DeleteAssetModelInterfaceRelationship` acción. Reemplace *your-interface-asset-model-id* por su propio valor. Para obtener más información, consulta [DeleteAssetModelInterfaceRelationship](#) en la AWS IoT SiteWise Referencia de la API de .

```
aws iotsitewise delete-asset-model-interface-relationship \  
  --asset-model-id your-asset-model-id \  
  --interface-asset-model-id your-interface-asset-model-id
```

Ejemplos de interfaz adicionales

A continuación, se muestran ejemplos adicionales de cómo se pueden utilizar las interfaces en diferentes escenarios industriales:

Equipos de generación de energía

Una empresa de generación de energía puede usar interfaces para estandarizar las métricas de los diferentes tipos de equipos de generación:

```

{
  "assetModelName": "GENERATOR-INTERFACE",
  "assetModelDescription": "Standard interface for power generators",
  "assetModelType": "INTERFACE",
  "assetModelProperties": [
    {
      "name": "ActivePower",
      "dataType": "DOUBLE",
      "type": { "measurement": {} },
      "unit": "MW"
    },
    {
      "name": "ReactivePower",
      "dataType": "DOUBLE",
      "type": { "measurement": {} },
      "unit": "MVAR"
    },
    {
      "name": "Frequency",
      "dataType": "DOUBLE",
      "type": { "measurement": {} },
      "unit": "Hz"
    },
    {
      "name": "PowerFactor",
      "dataType": "DOUBLE",
      "type": {
        "metric": {
          "expression": "cos(atan(ReactivePower / ActivePower))",
          "variables": [
            {
              "name": "ActivePower",
              "value": { "propertyId": "${ActivePower}" }
            },
            {
              "name": "ReactivePower",
              "value": { "propertyId": "${ReactivePower}" }
            }
          ]
        },
        "window": { "tumbling": { "interval": "5m" } }
      }
    },
    {
      "unit": "None"
    }
  ]
}

```

```

    }
  ]
}

```

Esta interfaz se puede aplicar a varios modelos de activos generadores (turbinas de gas, turbinas de vapor, turbinas eólicas) para garantizar métricas de potencia consistentes en toda la flota.

Instalaciones de tratamiento de agua

Una empresa de suministro de agua puede utilizar interfaces para estandarizar el monitoreo en todas las plantas de tratamiento:

```

{
  "assetModelName": "WATER-QUALITY-INTERFACE",
  "assetModelDescription": "Standard interface for water quality monitoring",
  "assetModelType": "INTERFACE",
  "assetModelProperties": [
    {
      "name": "pH",
      "dataType": "DOUBLE",
      "type": { "measurement": {} },
      "unit": "pH"
    },
    {
      "name": "Turbidity",
      "dataType": "DOUBLE",
      "type": { "measurement": {} },
      "unit": "NTU"
    },
    {
      "name": "DissolvedOxygen",
      "dataType": "DOUBLE",
      "type": { "measurement": {} },
      "unit": "mg/L"
    },
    {
      "name": "QualityIndex",
      "dataType": "DOUBLE",
      "type": {
        "metric": {
          "expression": "(pH >= 6.5 && pH <= 8.5 ? 100 : 50) * (Turbidity < 1 ? 1 : 0.8) * (DissolvedOxygen > 5 ? 1 : 0.7)",
          "variables": [

```

```

    {
      "name": "pH",
      "value": { "propertyId": "${pH}" }
    },
    {
      "name": "Turbidity",
      "value": { "propertyId": "${Turbidity}" }
    },
    {
      "name": "DissolvedOxygen",
      "value": { "propertyId": "${DissolvedOxygen}" }
    }
  ],
  "window": { "tumbling": { "interval": "1h" } }
}
"unit": "Score"
}
]
}

```

Esta interfaz garantiza que la calidad del agua se mida de manera uniforme en todas las instalaciones de tratamiento, independientemente de la configuración específica de sus equipos.

Interfaces jerárquicas

Las interfaces se pueden organizar jerárquicamente para admitir métricas agregadas en diferentes niveles de la operación:

1. Interfaz a nivel de equipo (por ejemplo,) PUMP-INTERFACE
 - Propiedades: caudal, presión, consumo de energía, vibración
 - Métricas: eficiencia, puntuación de salud
2. Interfaz a nivel de proceso (por ejemplo,) PUMPING-STATION-INTERFACE
 - Propiedades: flujo total, presión media, potencia total
 - Métricas: eficiencia de la estación, costo operativo por volumen
 - Jerarquía: contiene el PUMP-INTERFACE equipo
3. Interfaz a nivel de instalación (por ejemplo,) WATER-FACILITY-INTERFACE
 - Propiedades: rendimiento de la instalación, uso de energía, uso de productos químicos

- Métricas: eficiencia de las instalaciones, coste por unidad de volumen, huella de carbono
- Jerarquía: contiene PUMPING-STATION-INTERFACE procesos

Este enfoque jerárquico permite calcular las métricas en cada nivel y, al mismo tiempo, mantener la coherencia en toda la operación.

Configura el objeto AWS IoT SiteWise IDs

AWS IoT SiteWise define varios tipos de objetos persistentes, como activos, modelos de activos, propiedades y jerarquías. Todos estos objetos tienen identificadores únicos que puede utilizar para recuperarlos, actualizarlos y eliminarlos.

AWS IoT SiteWise tiene diferentes opciones para los clientes a la hora de crear identificaciones. AWS IoT SiteWise genera uno por defecto en el momento de la creación del objeto. Los usuarios también pueden proporcionar los suyos propios IDs a sus objetos.

Temas

- [Trabaja con un objeto UUIDs](#)
- [Usa un externo IDs](#)

Trabaja con un objeto UUIDs

Cada objeto persistente AWS IoT SiteWise tiene un [UUID](#) para identificarlo. Por ejemplo, los modelos de activos tienen un ID de modelo de activos, los activos tienen un ID de activo, etc. Este ID se asigna en el momento de crear el objeto y permanece inalterado durante la vida útil del objeto.

Cuando creas un objeto nuevo, AWS IoT SiteWise genera un identificador único para ti de forma predeterminada. También puede proporcionar su propio ID en el momento de la creación en formato UUID.

Note

UUIDs debe ser único a nivel mundial en la AWS región en la que se creó y para el mismo tipo de objeto. Cuando se AWS IoT SiteWise genera automáticamente un ID para ti, siempre es único. Si elige su propio ID, asegúrate de que sea único.

Por ejemplo, si creas un nuevo modelo de activos mediante una llamada [CreateAssetModel](#), puedes proporcionar tu propio UUID en el `assetModelId` campo opcional de la solicitud.

Por el contrario, si omite `assetModelId` en la solicitud, AWS IoT SiteWise genera un UUID para el nuevo modelo de activos.

Usa un externo IDs

Para definir su propio ID en un formato distinto de UUID, puede asignar un ID externo. Por ejemplo, puedes hacerlo si reutilizas un identificador que estás utilizando en un sistema que no lo está AWS, o para que sea más legible para las personas. IDs Los externos tienen un formato más flexible. Puede utilizarlos para hacer referencia a sus objetos en las operaciones de la API de AWS IoT SiteWise en las que, de otro modo, utilizaría el UUID.

Al igual que el UUIDs, cada identificador externo debe ser único dentro de su contexto. Por ejemplo, no se pueden tener dos modelos de activos con el mismo ID externo. Además, al igual que el UUIDs, un objeto solo puede tener un identificador externo durante su vida útil, lo cual no puede cambiar.

Diferencias entre externo IDs y UUIDs

IDs Las externas se UUIDs diferencian de las siguientes maneras:

- Cada objeto tiene un UUID, pero los externos IDs son opcionales.
- AWS IoT SiteWise nunca genera contenido externo IDs. Los proporciona usted mismo.
- Si el objeto aún no tiene uno, puede asignarle un ID externo en cualquier momento.

Formato de externo IDs

Un ID externo válido tiene las siguientes propiedades:

- Tiene entre 2 y 128 caracteres.
- El primer carácter y el último deben ser alfanuméricos (A-Z, a-z, 0-9).
- Los caracteres que no sean el primero y el último deben ser alfanuméricos o alguno de los siguientes: `_ - . :`

Por ejemplo, un ID externo debe ajustarse a la siguiente expresión regular:

```
[a-zA-Z0-9][a-zA-Z0-9_\-\. :]*[a-zA-Z0-9]+
```

Objetos de referencia con objetos externos IDs

En muchos lugares en los que puede hacer referencia a un objeto mediante su UUID, puede utilizar su ID externo en su lugar, si lo tiene. Para ello, añada el ID externo al `externalId`: de la cadena.

Por ejemplo, supongamos que tiene un modelo de activos cuyo UUID (ID del modelo de activos) es `a1b2c3d4-5678-90ab-cdef-11111EXAMPLE`, que también tiene el ID externo `myExternalId`. Llame [DescribeAssetModel](#) para obtener detalles al respecto. Puede utilizar cualquiera de los valores siguientes como valor de `assetModelId`:

- Con el propio ID del modelo de activos (UUID): `a1b2c3d4-5678-90ab-cdef-11111EXAMPLE`
- Con el ID externo: `externalId:myExternalId`

```
aws iotsitewise describe-asset-model --asset-model-id a1b2c3d4-5678-90ab-
cdef-11111EXAMPLE
aws iotsitewise describe-asset-model --asset-model-id externalId:myExternalId
```

Note

El prefijo `externalId:` no forma parte, en sí mismo, del ID externo. Solo necesitas proporcionar el prefijo cuando proporcionas un identificador externo a una operación de API que acepte una operación externa UUIDs o externa IDs. Por ejemplo, proporcione el prefijo cuando consulte o actualice un objeto existente.

Al definir un ID externo para un objeto, por ejemplo, al crear un modelo de activos, no incluya el prefijo.

Puedes usar el externo IDs en lugar de de de UUIDs esta manera para muchas operaciones de API AWS IoT SiteWise, pero no para todas. Por ejemplo, el [GetAssetPropertyValue](#), debe usarse UUIDs; no admite el uso de identificadores externos.

Para determinar si una operación de la API en particular admite este uso, consulte la [Referencia de la API](#).

Cree modelos de activos, modelos de componentes e interfaces para AWS IoT SiteWise

AWS IoT SiteWise Los modelos de activos, los modelos de componentes y las interfaces impulsan la estandarización de sus datos industriales. Los modelos de activos definen el activo general, como una turbina eólica o una línea de fabricación. Los modelos de componentes representan los componentes individuales que constituyen el activo, como palas, generadores o sensores. Las interfaces hacen cumplir los estándares en los diferentes modelos de activos. Al crear estos modelos, puede organizar y estructurar los datos de sus activos de manera que reflejen las relaciones y jerarquías reales de sus equipos industriales, lo que facilita la supervisión, el análisis y el mantenimiento.

Un modelo de activos o un modelo de componentes contiene un nombre, una descripción, propiedades de los activos y (opcionalmente) modelos compuestos personalizados que agrupan propiedades o que hacen referencia a modelos de componentes para subconjuntos.

En AWS IoT SiteWise, puede crear modelos de activos, modelos de componentes e interfaces para representar la estructura y las propiedades de sus activos industriales y sus componentes.

- Puede utilizar un modelo de activos para crear activos. Además de las características enumeradas anteriormente, un modelo de activos también puede contener definiciones jerárquicas que definan relaciones entre activos.
- Un modelo de componentes representa un subconjunto dentro de un modelo de activos u otro modelo de componentes. Una vez creado un modelo de componentes, puede añadirle referencias en otros modelos de activos y modelos de componentes. Sin embargo, no se pueden crear activos directamente a partir de modelos de componentes.
- Una interfaz hace cumplir los estándares en los diferentes modelos de activos. Las interfaces definen propiedades, métricas y jerarquías comunes que los modelos de activos deben implementar. No se pueden crear activos directamente a partir de las interfaces, pero ayudan a garantizar la coherencia entre tipos de activos similares.

Tras crear un modelo de activos o un modelo de componentes, puede crear modelos compuestos personalizados para agrupar propiedades o hacer referencia a los modelos de componentes existentes. También puede vincular las interfaces a los modelos de activos para reforzar la estandarización.

Para obtener información detallada sobre cómo crear modelos de activos, modelos de componentes e interfaces, consulte las siguientes secciones.

Temas

- [Cree modelos de activos en AWS IoT SiteWise](#)
- [Creación de modelos de componentes de](#)
- [Definición de propiedades de datos](#)
- [Creación de modelos compuestos personalizados \(componentes\)](#)

Cree modelos de activos en AWS IoT SiteWise

AWS IoT SiteWise los modelos de activos impulsan la estandarización de sus datos industriales. Un modelo de activos contiene el nombre, la descripción, las propiedades de activos y las definiciones de jerarquía de activos. Por ejemplo, puede definir un modelo de turbina eólica con propiedades de temperatura, rotaciones por minuto (RPM) y potencia. A continuación, puede definir un modelo de parque eólico con una propiedad de salida de potencia neta y una definición de jerarquía de turbina eólica.

Note

- Le recomendamos que modele su operación comenzando con los nodos de nivel más bajo. Por ejemplo, cree el modelo de turbina eólica antes de crear el modelo de parque eólico. Las definiciones de jerarquía de activos contienen referencias a modelos de activos existentes. Con este enfoque, puede definir jerarquías de activos a medida que crea sus modelos.
- Los modelos de activos no pueden contener otros modelos de activos. Si debe definir un modelo al que pueda hacer referencia como subconjunto dentro de otro modelo, debe crear un modelo de component--> en su lugar. Para obtener más información, consulte [Creación de modelos de componentes de](#).

En las siguientes secciones se describe cómo utilizar la AWS IoT SiteWise consola o la API para crear modelos de activos. En las secciones siguientes también se describen los diferentes tipos de propiedades y jerarquías de activos que se pueden utilizar para crear modelos.

Temas

- [Creación de un modelo de activos \(consola\)](#)
- [Creación de un modelo de activos \(AWS CLI\)](#)
- [Modelos de activos de ejemplo](#)
- [Definición de jerarquías de modelos de activos](#)

Creación de un modelo de activos (consola)


Puede usar la AWS IoT SiteWise consola para crear un modelo de activos. La AWS IoT SiteWise consola ofrece varias funciones, como el autocompletado de fórmulas, que pueden ayudarle a definir modelos de activos válidos.

Para crear un modelo de activos (consola)

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, elija Models (Modelos).
3. Seleccione Crear modelo de activos.
4. En la página Crear un modelo, haga lo siguiente:
 - a. Escriba un Nombre para el modelo de activos, como **Wind Turbine** o **Wind Turbine Model**. Este nombre debe ser único en todos los modelos de su cuenta en esta región.
 - b. (Opcional) Añada un ID externo para el modelo. Se trata de un ID definido por el usuario. Para obtener más información, consulte [Objetos de referencia con objetos externos IDs](#) en la Guía del usuario de AWS IoT SiteWise .
 - c. (Opcional) Agregue Definiciones de mediciones para el modelo. Las mediciones representan flujos de datos de su equipo. Para obtener más información, consulte [Definición de flujos de datos procedentes del equipo \(mediciones\)](#).
 - d. (Opcional) Agregue Definiciones de transformación para el modelo. Las transformaciones son fórmulas que asignan datos de un formulario a otro. Para obtener más información, consulte [Datos de transformación \(transformaciones\)](#).
 - e. (Opcional) Agregue Definiciones de métricas para el modelo. Las métricas son fórmulas que agregan datos a lo largo de intervalos de tiempo. Las métricas pueden agregar datos de entrada de activos asociados, de modo que puede calcular valores que representan la operación o un subconjunto de la operación. Para obtener más información, consulte [Agregación de datos desde propiedades y otros activos \(métricas\)](#).


- f. (Opcional) Agregue Definiciones de la jerarquía para el modelo. Las jerarquías son relaciones entre activos. Para obtener más información, consulte [Definición de jerarquías de modelos de activos](#).
- g. (Opcional) Agregue etiquetas para el modelo de activos. Para obtener más información, consulte [Etiquete sus AWS IoT SiteWise recursos](#).
- h. Seleccione Crear modelo.

Al crear un modelo de activos, la AWS IoT SiteWise consola navega a la página del nuevo modelo. En esta página, puede consultar el Estado del modelo, que inicialmente es CREANDO. Esta página se actualiza automáticamente, por lo que puede esperar a que se actualice el estado del modelo.

 Note

El proceso de creación de un modelo de activos puede tardar unos minutos para modelos complejos. Una vez que el estado del modelo de activos sea ACTIVO, puede utilizar el modelo de activos para crear activos. Para obtener más información, consulte [Estados de activos y modelos](#).


5. (Opcional) Tras crear el modelo de activo, puede configurarlo para la periferia. Para obtener más información sobre SiteWise Edge, consulte [Configure las capacidades perimetrales en AWS IoT SiteWise Edge](#).
 - a. En la página del modelo, elija Configurar para Edge.

 Note

La característica de paquete de procesamiento de datos (DPP) ya no está disponible para los nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte [Cambio en la disponibilidad del paquete de procesamiento de datos](#).

- b. En la página de configuración del modelo, elija la configuración de periferia para el modelo. Esto controla dónde se AWS IoT SiteWise pueden calcular y almacenar las propiedades asociadas a este modelo de activos. Para obtener más información acerca de la configuración del modelo para la periferia, consulte [Configure una fuente OPC UA en Edge SiteWise](#).

- c. Para la configuración perimetral personalizada, elija la ubicación en la que desee AWS IoT SiteWise calcular y almacenar cada una de las propiedades del modelo de activos.

 Note

Las transformaciones y las métricas asociadas deben configurarse para la misma ubicación. Para obtener más información acerca de la configuración del modelo para la periferia, consulte [Configure una fuente OPC UA en Edge SiteWise](#).

- d. Seleccione Save. En la página del modelo, su Configuración de periferia ahora debería estar Configurada.

Creación de un modelo de activos (AWS CLI)

Puede usar AWS Command Line Interface (AWS CLI) para crear un modelo de activos.

Utilice la operación [CreateAssetModel](#) para crear un modelo de activos con propiedades y jerarquías. Esta operación espera una carga con la siguiente estructura.

```
{
  "assetModelType": "ASSET_MODEL",
  "assetModelName": "String",
  "assetModelDescription": "String",
  "assetModelProperties": Array of AssetModelProperty,
  "assetModelHierarchies": Array of AssetModelHierarchyDefinition
}
```

Creación de un modelo de activos (AWS CLI)

1. Cree un archivo llamado `asset-model-payload.json` y, a continuación, copie el siguiente objeto JSON en el archivo.

```
{
  "assetModelType": "ASSET_MODEL",
  "assetModelName": "",
  "assetModelDescription": "",
  "assetModelProperties": [
  ],
  "assetModelHierarchies": [
```

```
  ],  
  "assetModelCompositeModels": [  
  
  ]  
}
```

2. Use su editor de texto JSON preferido para editar el archivo `asset-model-payload.json` para lo que se muestra a continuación:
 - a. Escriba un nombre (`assetModelName`) para el modelo de activos, como **Wind Turbine** o **Wind Turbine Model**. Este nombre debe ser único en todos los modelos de activos y modelos de componentes de su cuenta en esta Región de AWS.
 - b. (Opcional) Introduzca un ID externo (`assetModelExternalId`) para el modelo de activos. Se trata de un ID definido por el usuario. Para obtener más información, consulte [Objetos de referencia con objetos externos IDs](#) en la Guía del usuario de AWS IoT SiteWise .
 - c. (Opcional) Escriba una descripción (`assetModelDescription`) para el modelo de activos o elimine el par de clave-valor `assetModelDescription`.
 - d. (Opcional) Defina las propiedades del activo (`assetModelProperties`) para el modelo. Para obtener más información, consulte [Definición de propiedades de datos](#).
 - e. (Opcional) Defina jerarquías de activos (`assetModelHierarchies`) para el modelo. Para obtener más información, consulte [Definición de jerarquías de modelos de activos](#).
 - f. (Opcional) Defina las alarmas para el modelo. Las alarmas monitorean otras propiedades para que pueda identificar cuándo requieren atención los equipos o procesos. Cada definición de alarma es un modelo compuesto (`assetModelCompositeModels`) que estandariza el conjunto de propiedades que utiliza la alarma. Para obtener más información, consulte [Supervise los datos con las alarmas integradas AWS IoT SiteWise](#) y [Defina las alarmas en los modelos de activos en AWS IoT SiteWise](#).
 - g. (Opcional) Agregue etiquetas (`tags`) para el modelo de activos. Para obtener más información, consulte [Etiquete sus AWS IoT SiteWise recursos](#).
3. Ejecute el siguiente comando para crear un modelo de activos a partir de la definición en el archivo JSON.

```
aws iotsitewise create-asset-model --cli-input-json file://asset-model-payload.json
```

La operación devuelve una respuesta que contiene `assetModelId` al que hace referencia al crear un activo. La respuesta también contiene el estado del modelo

(`assetModelStatus.state`), que es inicialmente `CREATING`. El estado del modelo de activos es `CREATING` hasta que se propagan los cambios.

Note

El proceso de creación de un modelo de activos puede tardar unos minutos para modelos complejos. Para comprobar el estado actual de su modelo de activos, utilice la [DescribeAssetModel](#) operación especificando el `assetModelId`. Una vez que el estado del modelo de activos sea `ACTIVE`, puede utilizarlo para crear activos. Para obtener más información, consulte [Estados de activos y modelos](#).

4. (Opcional) Cree modelos compuestos personalizados para su modelo de activos. Con los modelos compuestos personalizados puede agrupar propiedades dentro del modelo o incluir un subconjunto haciendo referencia a un modelo de componentes. Para obtener más información, consulte [Creación de modelos compuestos personalizados \(componentes\)](#).

Modelos de activos de ejemplo

Esta sección contiene ejemplos de definiciones de modelos de activos que puede utilizar para crear modelos de activos con AWS CLI y AWS IoT SiteWise SDKs. Estos modelos de activos representan una turbina eólica y un parque eólico. Los activos de las turbinas eólicas ingieren datos sin procesar de los sensores y calculan valores como la potencia y la velocidad media del viento. Los activos del parque eólico calculan valores como la potencia total de todas las turbinas eólicas del parque eólico.

Temas

- [Modelo de activos de turbina eólica](#)
- [Modelo de activos de parque eólico](#)

Modelo de activos de turbina eólica

El siguiente modelo de activos representa una turbina en un parque eólico. La turbina eólica ingiere los datos de los sensores para calcular valores como la potencia y la velocidad media del viento.

Note

Este modelo de ejemplo se parece al modelo de aerogenerador de la AWS IoT SiteWise demostración. Para obtener más información, consulte [Usa la AWS IoT SiteWise demo](#).

```
{
  "assetModelType": "ASSET_MODEL",
  "assetModelName": "Wind Turbine Asset Model",
  "assetModelDescription": "Represents a turbine in a wind farm.",
  "assetModelProperties": [
    {
      "name": "Location",
      "dataType": "STRING",
      "type": {
        "attribute": {
          "defaultValue": "Renton"
        }
      }
    },
    {
      "name": "Make",
      "dataType": "STRING",
      "type": {
        "attribute": {
          "defaultValue": "Amazon"
        }
      }
    },
    {
      "name": "Model",
      "dataType": "INTEGER",
      "type": {
        "attribute": {
          "defaultValue": "500"
        }
      }
    },
    {
      "name": "Torque (KiloNewton Meter)",
      "dataType": "DOUBLE",
      "unit": "kNm",
      "type": {
        "measurement": {}
      }
    },
    {
      "name": "Wind Direction",
      "dataType": "DOUBLE",
```

```
    "unit": "Degrees",
    "type": {
      "measurement": {}
    }
  },
  {
    "name": "RotationsPerMinute",
    "dataType": "DOUBLE",
    "unit": "RPM",
    "type": {
      "measurement": {}
    }
  },
  {
    "name": "Wind Speed",
    "dataType": "DOUBLE",
    "unit": "m/s",
    "type": {
      "measurement": {}
    }
  },
  {
    "name": "RotationsPerSecond",
    "dataType": "DOUBLE",
    "unit": "RPS",
    "type": {
      "transform": {
        "expression": "rpm / 60",
        "variables": [
          {
            "name": "rpm",
            "value": {
              "propertyId": "RotationsPerMinute"
            }
          }
        ]
      }
    }
  },
  {
    "name": "Overdrive State",
    "dataType": "DOUBLE",
    "type": {
      "transform": {
```

```

    "expression": "gte(torque, 3)",
    "variables": [
      {
        "name": "torque",
        "value": {
          "propertyId": "Torque (KiloNewton Meter)"
        }
      }
    ]
  }
},
{
  "name": "Average Power",
  "dataType": "DOUBLE",
  "unit": "Watts",
  "type": {
    "metric": {
      "expression": "avg(torque) * avg(rps) * 2 * 3.14",
      "variables": [
        {
          "name": "torque",
          "value": {
            "propertyId": "Torque (Newton Meter)"
          }
        },
        {
          "name": "rps",
          "value": {
            "propertyId": "RotationsPerSecond"
          }
        }
      ],
      "window": {
        "tumbling": {
          "interval": "5m"
        }
      }
    }
  }
},
{
  "name": "Average Wind Speed",
  "dataType": "DOUBLE",

```

```

    "unit": "m/s",
    "type": {
      "metric": {
        "expression": "avg(windspeed)",
        "variables": [
          {
            "name": "windspeed",
            "value": {
              "propertyId": "Wind Speed"
            }
          }
        ],
        "window": {
          "tumbling": {
            "interval": "5m"
          }
        }
      }
    }
  },
  {
    "name": "Torque (Newton Meter)",
    "dataType": "DOUBLE",
    "unit": "Nm",
    "type": {
      "transform": {
        "expression": "knm * 1000",
        "variables": [
          {
            "name": "knm",
            "value": {
              "propertyId": "Torque (KiloNewton Meter)"
            }
          }
        ]
      }
    }
  }
],
{
  "name": "Overdrive State Time",
  "dataType": "DOUBLE",
  "unit": "Seconds",
  "type": {
    "metric": {

```

```

    "expression": "statetime(overdrive_state)",
    "variables": [
      {
        "name": "overdrive_state",
        "value": {
          "propertyId": "Overdrive State"
        }
      }
    ],
    "window": {
      "tumbling": {
        "interval": "5m"
      }
    }
  }
},
"assetModelHierarchies": []
}

```

Modelo de activos de parque eólico

El siguiente modelo de activos representa un parque eólico que comprende varias turbinas eólicas. Este modelo de activos define una [jerarquía](#) con respecto al modelo de turbinas eólicas. Esto permite al parque eólico calcular valores (como la potencia media) a partir de los datos de todas las turbinas eólicas del parque eólico.

Note

Este modelo de ejemplo se parece al modelo de parque eólico de la AWS IoT SiteWise demostración. Para obtener más información, consulte [Usa la AWS IoT SiteWise demo](#).

Este modelo de activos depende de [Modelo de activos de turbina eólica](#). Sustituya los valores `propertyId` y `childAssetModelId` por los de un modelo de activos de turbina eólica existente.

```

{
  "assetModelName": "Wind Farm Asset Model",
  "assetModelDescription": "Represents a wind farm.",
  "assetModelProperties": [
    {

```

```

    "name": "Code",
    "dataType": "INTEGER",
    "type": {
      "attribute": {
        "defaultValue": "300"
      }
    }
  },
  {
    "name": "Location",
    "dataType": "STRING",
    "type": {
      "attribute": {
        "defaultValue": "Renton"
      }
    }
  },
  {
    "name": "Reliability Manager",
    "dataType": "STRING",
    "type": {
      "attribute": {
        "defaultValue": "Mary Major"
      }
    }
  },
  {
    "name": "Total Overdrive State Time",
    "dataType": "DOUBLE",
    "unit": "seconds",
    "type": {
      "metric": {
        "expression": "sum(overdrive_state_time)",
        "variables": [
          {
            "name": "overdrive_state_time",
            "value": {
              "propertyId": "ID of Overdrive State Time property in Wind Turbine
Asset Model",
              "hierarchyId": "Turbine Asset Model"
            }
          }
        ]
      }
    }
  },
  "window": {

```

```

        "tumbling": {
            "interval": "5m"
        }
    }
},
{
    "name": "Total Average Power",
    "dataType": "DOUBLE",
    "unit": "Watts",
    "type": {
        "metric": {
            "expression": "sum(turbine_avg_power)",
            "variables": [
                {
                    "name": "turbine_avg_power",
                    "value": {
                        "propertyId": "ID of Average Power property in Wind Turbine Asset Model",
                        "hierarchyId": "Turbine Asset Model"
                    }
                }
            ],
            "window": {
                "tumbling": {
                    "interval": "5m"
                }
            }
        }
    }
},
"assetModelHierarchies": [
    {
        "name": "Turbine Asset Model",
        "childAssetModelId": "ID of Wind Turbine Asset Model"
    }
]
}

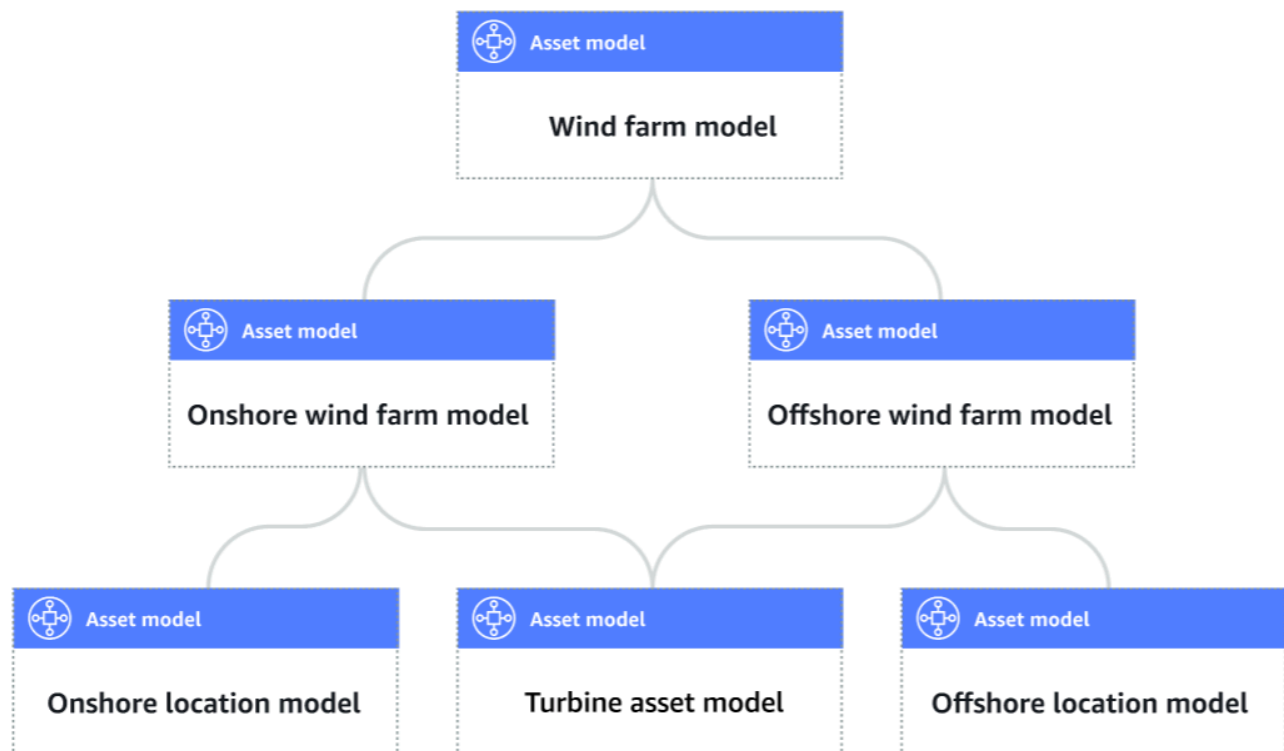
```

Definición de jerarquías de modelos de activos

Puede definir jerarquías de modelos de activos para crear asociaciones lógicas entre los modelos de activos de su operación industrial. Por ejemplo, puede definir un parque eólico compuesto por parques eólicos terrestres y marinos. Un parque eólico terrestre contiene una turbina y una ubicación terrestre. Un parque eólico marino contiene una turbina y una ubicación en alta mar.



Asset model hierarchy



Cuando asocia un modelo de activos secundario con un activo principal mediante una jerarquía, las métricas del modelo de activos principal pueden agregar datos de las métricas del modelo de activos secundario. Puede utilizar jerarquías y métricas de modelos de activos para calcular estadísticas que proporcionan información sobre la operación o un subconjunto de la operación. Para obtener más información, consulte [Agregación de datos desde propiedades y otros activos \(métricas\)](#).

Cada jerarquía define una relación entre un modelo de activo principal y un modelo de activo secundario. En un modelo de activo principal, puede definir varias jerarquías para el mismo modelo

de entidad secundaria. Por ejemplo, si tiene dos tipos diferentes de aerogeneradores en sus parques eólicos, donde todos los aerogeneradores están representados por el mismo modelo de activo, puede definir una jerarquía para cada tipo. A continuación, puede definir métricas en el modelo de parque eólico para calcular estadísticas independientes y combinadas para cada tipo de aerogenerador.

Un modelo de activo principal se puede asociar a varios modelos de entidades secundarias. Por ejemplo, si tiene un parque eólico terrestre y un parque eólico marino representados por dos modelos de activos diferentes, puede asociar estos modelos de activos al mismo modelo de activo del parque eólico principal.

También se puede asociar un modelo de entidades secundarias a varios modelos de activos principales. Por ejemplo, si tiene dos tipos diferentes de parques eólicos, en los que todos los aerogeneradores están representados por el mismo modelo de activos, puede asociar el modelo de activos de los aerogeneradores a distintos modelos de activos de parques eólicos.

Note

Al definir una jerarquía de modelos de activos, el modelo de activos secundario debe estar ACTIVE o tener una versión ACTIVE anterior. Para obtener más información, consulte [Estados de activos y modelos](#).

Después de definir modelos de activos jerárquicos y crear activos, puede asociar los activos para completar la relación principal-secundario. Para obtener más información, consulte [Cree activos para modelos de activos en AWS IoT SiteWise](#) y [Asociación y desasociación de activos](#).

Temas

- [Definición de jerarquías de modelos de activos \(consola\)](#)
- [Definición de jerarquías de activos \(AWS CLI\)](#)

Definición de jerarquías de modelos de activos (consola)

Al definir una jerarquía para un modelo de activos en la AWS IoT SiteWise consola, se especifican los siguientes parámetros:

- Nombre de la jerarquía: el nombre de la jerarquía, por ejemplo **Wind Turbines**.
- Modelo de la jerarquía: el modelo de entidad secundaria.

- Jerarquía ID externo (opcional): es un ID definido por el usuario. Para obtener más información, consulte [Objetos de referencia con objetos externos IDs](#) en la Guía del usuario de AWS IoT SiteWise .

Para obtener más información, consulte [Creación de un modelo de activos \(consola\)](#).

Definición de jerarquías de activos (AWS CLI)

Al definir una jerarquía para un modelo de activos con la AWS IoT SiteWise API, se especifican los siguientes parámetros:

- `name`: el nombre de la jerarquía, por ejemplo **Wind Turbines**.
- `childAssetModelId`: el ID o el ID externo del modelo de activos secundarios para la jerarquía. Puede utilizar la [ListAssetModels](#) operación para buscar el ID de un modelo de activos existente.

Example Definición de jerarquía de ejemplo

En el ejemplo siguiente, se muestra una jerarquía de modelos de activos que representa la relación de un parque eólico con las turbinas eólicas. Este objeto es un ejemplo de [AssetModelHierarchy](#). Para obtener más información, consulte [Creación de un modelo de activos \(AWS CLI\)](#).

```
{
  ...
  "assetModelHierarchies": [
    {
      "name": "Wind Turbines",
      "childAssetModelId": "a1b2c3d4-5678-90ab-cdef-1111EXAMPLE"
    },
  ],
}
```

Creación de modelos de componentes de

Utilice los modelos de AWS IoT SiteWise componentes para definir subconjuntos a los que pueda hacer referencia desde modelos de activos u otros modelos de componentes. De esta forma, puede reutilizar la definición del componente en varios modelos diferentes o varias veces dentro del mismo modelo.

El proceso de definición de un modelo de componentes es muy similar a la definición de un modelo de activos. Al igual que un modelo de activos, un modelo de componentes tiene un nombre, una

descripción y propiedades de activos. Sin embargo, los modelos de componentes no pueden incluir definiciones de jerarquías de activos, ya que en sí mismos no se pueden utilizar para crear activos directamente. Los modelos de componentes tampoco pueden definir alarmas.

Por ejemplo, puede definir un componente para un servomotor con las propiedades de temperatura del motor, temperatura del codificador y resistencia de aislamiento. A continuación, puede definir un modelo de activos para equipos que contienen servomotores, como una máquina CNC.

Note

- Le recomendamos que modele su operación comenzando con los nodos de nivel más bajo. Por ejemplo, cree el componente de servomotor antes de crear el modelo de activos de la máquina CNC. Los modelos de activos contienen referencias a modelos de componentes existentes.
- No se pueden crear activos directamente a partir de modelos de componentes. Para crear un activo que utilice su componente, debe crear un modelo de activos para su activo. A continuación, debe crear un modelo compuesto personalizado para este que haga referencia a su componente. Para obtener más información sobre la creación de modelos de activos, consulte [Cree modelos de activos en AWS IoT SiteWise](#). Para obtener más información sobre la creación de modelos compuestos personalizados, consulte [Creación de modelos compuestos personalizados \(componentes\)](#).

En las siguientes secciones se describe cómo utilizar la AWS IoT SiteWise API para crear modelos de componentes.

Temas

- [Creación de un modelo de componentes \(AWS CLI\)](#)
- [Modelo de componentes de ejemplo](#)

Creación de un modelo de componentes (AWS CLI)

Puede usar el AWS Command Line Interface (AWS CLI) para crear un modelo de componentes.

Utilice la [CreateAssetModel](#) operación para crear un modelo de componentes con propiedades. Esta operación espera una carga útil con la siguiente estructura:

```
{
```

```
"assetModelType": "COMPONENT_MODEL",
"assetModelName": "String",
"assetModelDescription": "String",
"assetModelProperties": Array of AssetModelProperty,
}
```

Creación de un modelo de componentes (AWS CLI)

1. Cree un archivo llamado `component-model-payload.json` y, a continuación, copie el siguiente objeto JSON en el archivo:

```
{
  "assetModelType": "COMPONENT_MODEL",
  "assetModelName": "",
  "assetModelDescription": "",
  "assetModelProperties": [
  ]
}
```

2. Use su editor de texto JSON preferido para editar el archivo `component-model-payload.json` para lo que se muestra a continuación:
 - a. Escriba un nombre (`assetModelName`) para el modelo de componentes, como **Servo Motor** o **Servo Motor Model**. Este nombre debe ser único en todos los modelos de activos y modelos de componentes de su cuenta en esta Región de AWS.
 - b. (Opcional) Introduzca un ID externo (`assetModelExternalId`) para el modelo de componentes. Se trata de un ID definido por el usuario. Para obtener más información, consulte [Objetos de referencia con objetos externos IDs](#) en la Guía del usuario de AWS IoT SiteWise .
 - c. (Opcional) Escriba una descripción (`assetModelDescription`) para el modelo de activos o elimine el par de clave-valor `assetModelDescription`.
 - d. (Opcional) Defina las propiedades del activo (`assetModelProperties`) para el modelo de componentes. Para obtener más información, consulte [Definición de propiedades de datos](#).
 - e. (Opcional) Agregue etiquetas (`tags`) para el modelo de activos. Para obtener más información, consulte [Etiquete sus AWS IoT SiteWise recursos](#).
3. Ejecute el siguiente comando para crear un modelo de componentes a partir de la definición en el archivo JSON.

```
aws iotsitewise create-asset-model --cli-input-json file://component-model-payload.json
```

La operación devuelve una respuesta que contiene el `assetModelId` al que se hace referencia al añadir una referencia al modelo de componentes en un modelo de activos o en otro modelo de componentes. La respuesta también contiene el estado del modelo (`assetModelStatus.state`), que es inicialmente `CREATING`. El estado del modelo de componentes es `CREATING` hasta que se propagan los cambios.

Note

El proceso de creación de un modelo de componentes puede tardar unos minutos para modelos complejos. Para comprobar el estado actual del modelo de componentes, utilice la [DescribeAssetModel](#) operación especificando el `assetModelId`. Cuando el estado del modelo de componentes es `ACTIVE`, puede añadir referencias a su modelo de componentes en modelos de activos u otros modelos de componentes. Para obtener más información, consulte [Estados de activos y modelos](#).

4. (Opcional) Cree modelos compuestos personalizados para su modelo de activos. Con los modelos compuestos personalizados, puede agrupar propiedades dentro del modelo o incluir un subconjunto haciendo referencia a otro modelo de componente. Para obtener más información, consulte [Creación de modelos compuestos personalizados \(componentes\)](#).

Modelo de componentes de ejemplo

Esta sección contiene un ejemplo de definición de modelo de componentes que puede utilizar para crear un modelo de componentes con AWS CLI y AWS IoT SiteWise SDKs. Este modelo de componentes representa un servomotor que se puede utilizar en otro equipo, como una máquina CNC.

Temas

- [Modelo de componentes de servomotor](#)

Modelo de componentes de servomotor

Este modelo de componentes representa un servomotor que se puede utilizar en otro equipo, como máquinas CNC. El servomotor proporciona diversas medidas, como la temperatura y la resistencia eléctrica. Estas medidas están disponibles como propiedades en los activos creados a partir de modelos de activos que hacen referencia al modelo de componentes del servomotor.

```
{
  "assetModelName": "ServoMotor",
  "assetModelType": "COMPONENT_MODEL",
  "assetModelProperties": [
    {
      "dataType": "DOUBLE",
      "name": "Servo Motor Temperature",
      "type": {
        "measurement": {}
      },
      "unit": "Celsius"
    },
    {
      "dataType": "DOUBLE",
      "name": "Spindle speed",
      "type": {
        "measurement": {}
      },
      "unit": "rpm"
    }
  ]
}
```

Definición de propiedades de datos

Las propiedades de activo son las estructuras dentro de cada activo que contienen datos del activo. Las propiedades de un activo pueden ser cualquiera de los siguientes tipos:

- **Atributos:** las propiedades generalmente estáticas de un activo, como el fabricante del dispositivo o la región geográfica. Para obtener más información, consulte [Definición de datos estáticos \(atributos\)](#).
- **Mediciones:** flujos de datos sin procesar de un activo procedentes del sensor de un dispositivo, como los valores de velocidad de rotación con marca temporal o los valores de temperatura en grados Celsius con marca temporal. Una medida se define mediante un alias de flujo de datos.

Para obtener más información, consulte [Definición de flujos de datos procedentes del equipo \(mediciones\)](#).

- Transformaciones: valores de serie temporal de un activo transformados, como los valores de temperatura en grados Fahrenheit con marca temporal. Una transformación se define por una expresión y las variables que se consumen con esa expresión. Para obtener más información, consulte [Datos de transformación \(transformaciones\)](#).
- Métricas: los datos de un activo agregados durante un intervalo de tiempo específico, como la temperatura media por hora. Una métrica se define mediante un intervalo de tiempo, una expresión y las variables que se consumen con esa expresión. Las expresiones métricas pueden introducir las propiedades métricas de los activos asociados, para que pueda calcular las métricas que representan su operación o un subconjunto de la misma. Para obtener más información, consulte [Agregación de datos desde propiedades y otros activos \(métricas\)](#).

Para obtener más información, consulte [Cree modelos de activos en AWS IoT SiteWise](#).

Para obtener un ejemplo de cómo utilizar mediciones, transformaciones y métricas para calcular la efectividad global del equipo (OEE), consulte [Calcule el OEE en AWS IoT SiteWise](#).

Temas

- [Definición de datos estáticos \(atributos\)](#)
- [Definición de flujos de datos procedentes del equipo \(mediciones\)](#)
- [Datos de transformación \(transformaciones\)](#)
- [Agregación de datos desde propiedades y otros activos \(métricas\)](#)
- [Uso de expresiones de fórmula](#)

Definición de datos estáticos (atributos)

Los atributos de activos representan información que generalmente es estática, como el fabricante del dispositivo o la ubicación geográfica. Cada activo creado a partir de un modelo de activos contiene los atributos de dicho modelo.

Temas

- [Definición de atributos \(consola\)](#)
- [Definición de atributos \(AWS CLI\)](#)

Definición de atributos (consola)

Al definir un atributo para un modelo de activos en la AWS IoT SiteWise consola, se especifican los siguientes parámetros:

- Nombre: el nombre de la propiedad.
- Valor predeterminado – (Opcional) El valor predeterminado de este atributo. Los activos creados a partir del modelo tienen este valor para el atributo. Para obtener más información acerca de cómo reemplazar el valor por defecto en un activo creado a partir de un modelo, consulte [Actualización de valores de atributos](#).
- Tipo de datos: el tipo de datos de la propiedad, que es uno de los siguientes:
 - Cadena – Una cadena con hasta 1024 bytes.
 - Entero – Un entero de 32 bits con signo con rango [-2.147.483.648, 2.147.483.647].
 - Doble – Un número de punto flotante con rango [-10¹⁰⁰, 10¹⁰⁰] e IEEE 754 doble precisión.
 - Booleano: true o false.
- ID externo: (opcional) es un ID definido por el usuario. Para obtener más información, consulte [Objetos de referencia con objetos externos IDs](#) en la Guía del usuario de AWS IoT SiteWise .

Para obtener más información, consulte [Creación de un modelo de activos \(consola\)](#).

Definición de atributos (AWS CLI)

Al definir un atributo para un modelo de activos con la AWS IoT SiteWise API, se especifican los siguientes parámetros:

- name: el nombre de la propiedad.
- defaultValue – (Opcional) El valor predeterminado de este atributo. Los activos creados a partir del modelo tienen este valor para el atributo. Para obtener más información acerca de cómo reemplazar el valor por defecto en un activo creado a partir de un modelo, consulte [Actualización de valores de atributos](#).
- dataType: el tipo de datos de la propiedad, que es uno de los siguientes:
 - STRING – Una cadena con hasta 1024 bytes.
 - INTEGER: un entero de 32 bits con signo con rango [-2.147.483.648, 2.147.483.647].
 - DOUBLE: un número de punto flotante con rango [-10¹⁰⁰, 10¹⁰⁰] e IEEE 754 doble precisión.
 - BOOLEAN – true o false.

- `externalId`: (opcional) es un ID definido por el usuario. Para obtener más información, consulte [Objetos de referencia con objetos externos IDs](#) en la Guía del usuario de AWS IoT SiteWise .

Example Definición de atributo de ejemplo

En el ejemplo siguiente se muestra un atributo que representa el número de modelo de un activo con un valor predeterminado. Este objeto es un ejemplo de un objeto [AssetModelProperty](#) que contiene un [atributo](#). Puede especificar este objeto como parte de la carga de solicitud de [CreateAssetModel](#) para crear una propiedad de atributo. Para obtener más información, consulte [Creación de un modelo de activos \(AWS CLI\)](#).

```
{
  ...
  "assetModelProperties": [
    {
      "name": "Model number",
      "dataType": "STRING",
      "type": {
        "attribute": {
          "defaultValue": "BLT123"
        }
      }
    }
  ],
  ...
}
```

Definición de flujos de datos procedentes del equipo (mediciones)

Una medición representa el flujo de datos sin formato del sensor de un dispositivo, como los valores de temperatura con marca de tiempo o los valores de revoluciones por minuto (RPM) con marca de tiempo.

Temas

- [Definición de mediciones \(consola\)](#)
- [Definición de mediciones \(AWS CLI\)](#)

Definición de mediciones (consola)

Al definir una medida para un modelo de activos en la AWS IoT SiteWise consola, se especifican los siguientes parámetros:

- Nombre: el nombre de la propiedad.
- Unidad: (opcional) la unidad científica de la propiedad, como mm o Celsius.
- Tipo de datos: el tipo de datos de la propiedad, que es uno de los siguientes:
 - Cadena – Una cadena con hasta 1024 bytes.
 - Entero – Un entero de 32 bits con signo con rango [-2.147.483.648, 2.147.483.647].
 - Doble – Un número de punto flotante con rango [-10¹⁰⁰, 10¹⁰⁰] e IEEE 754 doble precisión.
 - Booleano: true o false.
- ID externo: (opcional) es un ID definido por el usuario. Para obtener más información, consulte [Objetos de referencia con objetos externos IDs](#) en la Guía del usuario de AWS IoT SiteWise .

Para obtener más información, consulte [Creación de un modelo de activos \(consola\)](#).

Definición de mediciones (AWS CLI)

Al definir una medición para un modelo de activos con la AWS IoT SiteWise API, se especifican los siguientes parámetros:

- name: el nombre de la propiedad.
- dataType: el tipo de datos de la propiedad, que es uno de los siguientes:
 - STRING – Una cadena con hasta 1024 bytes.
 - INTEGER: un entero de 32 bits con signo con rango [-2.147.483.648, 2.147.483.647].
 - DOUBLE: un número de punto flotante con rango [-10¹⁰⁰, 10¹⁰⁰] e IEEE 754 doble precisión.
 - BOOLEAN – true o false.
- unit: (opcional) la unidad científica de la propiedad, como mm o Celsius.
- externalId: (opcional) es un ID definido por el usuario. Para obtener más información, consulte [Objetos de referencia con objetos externos IDs](#) en la Guía del usuario de AWS IoT SiteWise .

Example Definición de medida de ejemplo

En el ejemplo siguiente se muestra una medición que representa las lecturas del sensor de temperatura de un activo. Este objeto es un ejemplo de un objeto [AssetModelProperty](#) que contiene una [medición](#). Puede especificar este objeto como parte de la carga de la solicitud de [CreateAssetModel](#) para crear una propiedad de medición. Para obtener más información, consulte [Creación de un modelo de activos \(AWS CLI\)](#).

La estructura [Medida](#) es una estructura vacía cuando define un modelo de activos, ya que más adelante configurará cada activo para que utilice flujos de datos de dispositivos únicos. Para obtener más información acerca de cómo conectar la propiedad de medición de un activo al flujo de datos del sensor de un dispositivo, consulte [Gestione los flujos de datos para AWS IoT SiteWise](#).

```
{
  ...
  "assetModelProperty": [
    {
      "name": "Temperature C",
      "dataType": "DOUBLE",
      "type": {
        "measurement": {}
      },
      "unit": "Celsius"
    }
  ],
  ...
}
```

Datos de transformación (transformaciones)

Las transformaciones son expresiones matemáticas que asignan puntos de datos de la propiedad de un activo de un formulario a otro. Una expresión de transformación consta de variables de propiedad de activos, literales, operadores y funciones comunes. Los puntos de datos transformados mantienen una one-to-one relación con los puntos de datos de entrada. AWS IoT SiteWise calcula un nuevo punto de datos transformado cada vez que alguna de las propiedades de entrada recibe un nuevo punto de datos.

Note

En el caso de las actualizaciones de propiedades con la misma marca de tiempo, es posible que las actualizaciones de otras propiedades entrantes sobrescriban los valores de salida.

Por ejemplo, si su activo tiene un flujo de medición de temperatura denominado `Temperature_C` con unidades en Celsius, puede convertir cada punto de datos a Fahrenheit con la fórmula $Temperature_F = 9/5 * Temperature_C + 32$. Cada vez que AWS IoT SiteWise recibe un punto de datos en el flujo de `Temperature_C` medición, el `Temperature_F` valor correspondiente se calcula en unos segundos y está disponible como `Temperature_F` propiedad.

Si la transformación contiene más de una variable, el punto de datos que llegue antes inicia el cálculo inmediatamente. Considere un ejemplo en el que un fabricante de piezas utiliza una transformación para monitorear la calidad del producto. Aplicando estándares diferentes según el tipo de pieza, el fabricante utiliza las siguientes mediciones para representar el proceso:

- `Part_Number`: cadena que identifica el tipo de pieza.
- `Good_Count`: número entero que aumenta en uno si la pieza cumple el estándar.
- `Bad_Count`: número entero que aumenta en uno si la pieza no cumple el estándar.

El fabricante crea además una transformación `Quality_Monitor`, que equivale a `if(eq(Part_Number, "BLT123") and (Bad_Count / (Good_Count + Bad_Count) > 0.1), "Caution", "Normal")`.

Esta transformación monitorea el porcentaje de piezas defectuosas producidas para un tipo de pieza específico. Si el número de pieza es `BLT123` y el porcentaje de piezas defectuosas supera el 10 por ciento (0,1), la transformación vuelve `"Caution"`. De lo contrario, la transformación devuelve el mensaje `"Normal"`.

Note

- Si `Part_Number` recibe un nuevo punto de datos antes que otras mediciones, la transformación de `Quality_Monitor` utiliza el nuevo valor de `Part_Number` y los valores de `Good_Count` y `Bad_Count` más recientes. Para evitar errores, reinicie `Good_Count` y `Bad_Count` antes de la siguiente ronda de fabricación.

- Utilice [las métricas](#) si quiere evaluar las expresiones solo después de que todas las variables hayan recibido nuevos puntos de datos.

Temas

- [Definición de transformaciones \(consola\)](#)
- [Definición de transformaciones \(AWS CLI\)](#)

Definición de transformaciones (consola)

Al definir una transformación para un modelo de activos en la AWS IoT SiteWise consola, se especifican los siguientes parámetros:

- Nombre: el nombre de la propiedad.
- Unidad: (opcional) la unidad científica de la propiedad, como mm o Celsius.
- Tipo de datos: el tipo de datos de la transformación, que puede ser Doble o Cadena.
- ID externo: (opcional) es un ID definido por el usuario. Para obtener más información, consulte [Objetos de referencia con objetos externos IDs](#) en la Guía del usuario de AWS IoT SiteWise .
- Fórmula: la expresión de transformación. Las expresiones de transformación no pueden usar funciones de agregación ni funciones temporales. Para abrir la característica de autocompletar empiece a escribir o presione la tecla abajo. Para obtener más información, consulte [Uso de expresiones de fórmula](#).

Important

Las transformaciones pueden agregar propiedades de tipo entero, doble, booleano o cadena. Los valores booleanos se convierten en 0 (falso) y 1 (verdadero).

Las transformaciones deben especificar una o más propiedades que no sean atributos y cualquier número de propiedades de atributo. AWS IoT SiteWise calcula un nuevo punto de datos transformado cada vez que la propiedad de entrada de no atributo recibe un nuevo punto de datos. Los nuevos valores de atributo no inician actualizaciones de transformación. La misma tasa de solicitudes para las operaciones de la API de datos de propiedades de activos se aplica a los resultados del cálculo de transformaciones.

Las expresiones de fórmula solo pueden generar valores dobles o de cadena. Las expresiones anidadas pueden generar otros tipos de datos, como cadenas, pero la fórmula en su conjunto debe evaluarse como un número o una cadena. Puede usar la [función](#)

[jp](#) para convertir una cadena en un número. El valor booleano debe ser 1 (verdadero) o 0 (falso). Para obtener más información, consulte [Valores indefinidos, infinitos y de desbordamiento](#).

Para obtener más información, consulte [Creación de un modelo de activos \(consola\)](#).

Definición de transformaciones (AWS CLI)

Al definir una transformación para un modelo de activos con la AWS IoT SiteWise API, se especifican los siguientes parámetros:

- `name`: el nombre de la propiedad.
- `unit`: (opcional) la unidad científica de la propiedad, como mm o Celsius.
- `dataType`: el tipo de datos de la transformación, que debe ser `DOUBLE` o `STRING`.
- `externalId`: (opcional) es un ID definido por el usuario. Para obtener más información, consulte [Objetos de referencia con objetos externos IDs](#) en la Guía del usuario de AWS IoT SiteWise .
- `expression` – Una expresión de transformación. Las expresiones de transformación no pueden usar funciones de agregación ni funciones temporales. Para obtener más información, consulte [Uso de expresiones de fórmula](#).
- `variables` – Una lista de variables que define las otras propiedades del activo que se van a utilizar en la expresión. Cada estructura de variable contiene un nombre sencillo para utilizar en la expresión y una estructura de `value` que identifica qué propiedad vincular a esa variable. La estructura `value` contiene la siguiente información:
 - `propertyId`: el ID de la propiedad desde la que se van a introducir los valores. Puede usar el nombre de la propiedad en lugar de su ID.

Important

Las transformaciones pueden agregar propiedades de tipo entero, doble, booleano o cadena. Los valores booleanos se convierten en 0 (falso) y 1 (verdadero).

Las transformaciones deben especificar una o más propiedades que no sean atributos y cualquier número de propiedades de atributo. AWS IoT SiteWise calcula un nuevo punto de datos transformado cada vez que la propiedad de entrada de no atributo recibe un nuevo punto de datos. Los nuevos valores de atributo no inician actualizaciones de transformación. La misma tasa de solicitudes para las operaciones de la API de datos de propiedades de activos se aplica a los resultados del cálculo de transformaciones.

Las expresiones de fórmula solo pueden generar valores dobles o de cadena. Las expresiones anidadas pueden generar otros tipos de datos, como cadenas, pero la fórmula en su conjunto debe evaluarse como un número o una cadena. Puede usar la [función jp](#) para convertir una cadena en un número. El valor booleano debe ser 1 (verdadero) o 0 (falso). Para obtener más información, consulte [Valores indefinidos, infinitos y de desbordamiento](#).

Example definición de transformación

El siguiente ejemplo muestra una propiedad de transformación que convierte los datos de medición de temperatura de un activo de Celsius a Fahrenheit. Este objeto es un ejemplo de un objeto [AssetModelProperty](#) que contiene una [transformación](#). Puede especificar este objeto como parte de la carga de la solicitud de [CreateAssetModel](#) para crear una propiedad de transformación. Para obtener más información, consulte [Creación de un modelo de activos \(AWS CLI\)](#).

```
{
  ...
  "assetModelProperty": [
    ...
    {
      "name": "Temperature F",
      "dataType": "DOUBLE",
      "type": {
        "transform": {
          "expression": "9/5 * temp_c + 32",
          "variables": [
            {
              "name": "temp_c",
              "value": {
                "propertyId": "Temperature C"
              }
            }
          ]
        }
      },
      "unit": "Fahrenheit"
    }
  ],
  ...
}
```

Example definición de transformación que contiene tres variables

El siguiente ejemplo muestra una propiedad de transformación que devuelve un mensaje de advertencia ("Caution") si más del 10 por ciento de las BLT123 piezas no cumplen el estándar. De lo contrario, devuelve un mensaje informativo ("Normal").

```
{
  ...
  "assetModelProperties": [
    ...
    {
      "name": "Quality_Monitor",
      "dataType": "STRING",
      "type": {
        "transform": {
          "expression": "if(eq(Part_Number,\"BLT123\") and (Bad_Count / (Good_Count +
Bad_Count) > 0.1), \"Caution\", \"Normal\")",
          "variables": [
            {
              "name": "Part_Number",
              "value": {
                "propertyId": "Part Number"
              }
            },
            {
              "name": "Good_Count",
              "value": {
                "propertyId": "Good Count"
              }
            },
            {
              "name": "Bad_Count",
              "value": {
                "propertyId": "Bad Count"
              }
            }
          ]
        }
      }
    }
  ]
}
...
}
```

Agregación de datos desde propiedades y otros activos (métricas)

Las métricas son expresiones matemáticas que utilizan funciones agregadas para procesar todos los puntos de datos de entrada y generar un único punto de datos por intervalo de tiempo especificado. Por ejemplo, una métrica puede calcular la temperatura media por hora a partir de un flujo de datos de temperatura.

Las métricas pueden agregar datos de métricas de activos asociados, de modo que puede calcular estadísticas que proporcionan información sobre la operación o un subconjunto de la operación. Por ejemplo, una métrica puede calcular la temperatura media por hora en todas las turbinas eólicas de un parque eólico. Para obtener más información acerca de cómo definir asociaciones entre activos, consulte [Definición de jerarquías de modelos de activos](#).

Las métricas también pueden introducir datos de otras propiedades sin agregar datos en cada intervalo de tiempo. Si se especifica un [atributo](#) en una fórmula, AWS IoT SiteWise utiliza el valor [más reciente](#) de ese atributo al calcular la fórmula. Si especifica una métrica en una fórmula, AWS IoT SiteWise utiliza el [último](#) valor del intervalo de tiempo durante el que calcula la fórmula. Esto significa que se pueden definir métricas como $OEE = Availability * Quality * Performance$, donde Availability, Quality y Performance son todas las demás métricas en el mismo modelo de activos.

AWS IoT SiteWise también calcula automáticamente un conjunto de métricas de agregación básicas para todas las propiedades de los activos. Para reducir los costos de cálculo, puede utilizar estos agregados en lugar de definir métricas personalizadas para cálculos básicos. Para obtener más información, consulte [La propiedad del activo de consulta se agrega en AWS IoT SiteWise](#).

Temas

- [Definición de métricas \(consola\)](#)
- [Definición de métricas \(AWS CLI\)](#)

Definición de métricas (consola)

Al definir una métrica para un modelo de activos en la AWS IoT SiteWise consola, se especifican los siguientes parámetros:

- Nombre: el nombre de la propiedad.
- Tipo de datos: el tipo de datos de la transformación, que puede ser Doble o Cadena.

- ID externo: (opcional) es un ID definido por el usuario. Para obtener más información, consulte [Objetos de referencia con objetos externos IDs](#) en la Guía del usuario de AWS IoT SiteWise .
- Fórmula: la expresión métrica. Las expresiones métricas pueden utilizar [funciones de agregación](#) para introducir datos de una propiedad para todos los activos asociados en una jerarquía. Comience a escribir o presione la tecla de flecha hacia abajo para abrir la característica de autocompletar. Para obtener más información, consulte [Uso de expresiones de fórmula](#).

Important

Las métricas solo pueden agregar propiedades de tipo entero, doble, booleano o cadena. Los valores booleanos se convierten en 0 (falso) y 1 (verdadero).

Si define variables de entrada de métrica en la expresión de una métrica, esas entradas deben tener el mismo intervalo de tiempo que la métrica de salida.

Las expresiones de fórmula solo pueden generar valores dobles o de cadena. Las expresiones anidadas pueden generar otros tipos de datos, como cadenas, pero la fórmula en su conjunto debe evaluarse como un número o una cadena. Puede usar la [función jp](#) para convertir una cadena en un número. El valor booleano debe ser 1 (verdadero) o 0 (falso). Para obtener más información, consulte [Valores indefinidos, infinitos y de desbordamiento](#).

- Intervalo de tiempo – Un intervalo de tiempo de métrica. AWS IoT SiteWise admite los siguientes intervalos de tiempo con periodos de saltos, donde cada intervalo comienza cuando termina el anterior:
 - 1 minuto: 1 minuto, calculado al final de cada minuto (00:00:00 h, 00:01:00 h, 00:02:00 h, etc.).
 - 5 minutos: 5 minutos, calculados al final de cada cinco minutos a partir de la hora en punto (00:00:00 h., 00:05:00 h., 00:10:00 h, etc.).
 - 15 minutos – 15 minutos, calculados al final de cada quince minutos a partir de la hora en punto (00:00:00 h., 00:15:00 h., 00:30:00 h, etc.).
 - 1 hora: 1 hora (60 minutos), calculados al final de cada hora en UTC (00:00:00 h, 01:00:00 h, 02:00:00 h, etc.).
 - 1 día: 1 día (24 horas), calculado al final de cada día en UTC (00:00:00 h del lunes, 00:00:00 h del martes, etc.).
 - 1 semana: 1 semana (7 días), calculada al final de cada domingo en UTC (todos los lunes a las 00:00:00 h).

- Intervalo personalizado: puede escribir cualquier intervalo de tiempo entre un minuto y una semana.
- Desplazamiento de la fecha: (opcional) la fecha de referencia a partir de la cual se agregan los datos.
- Desplazamiento de la hora – (Opcional) la hora de referencia a partir de la cual se agregan los datos. El desplazamiento de la hora debe estar comprendido entre las 00:00:00 y las 23:59:59.
- Desplazamiento de la zona horaria: (opcional) la zona horaria de referencia. Si no se especifica, el desplazamiento predeterminado de la zona horaria es la hora universal coordinada (UTC). Consulte las siguientes zonas horarias compatibles.

Zonas horarias admitidas

- (UTC+00:00) Hora universal coordinada
- (UTC+01:00) Hora central europea
- (UTC+02:00) Europa del Este
- (UTC+03:00) Hora de África Oriental
- (UTC+04:00) Hora de Oriente Próximo
- (UTC+05:00) Hora de Lahore (Pakistán)
- (UTC+05:30) Hora estándar de India
- (UTC+06:00) Hora estándar de Bangladesh
- (UTC+07:00) Hora estándar de Vietnam
- (UTC+08:00) Hora de China Taiwán
- (UTC+09:00) Hora estándar de Japón
- (UTC+09:30) Hora central de Australia
- (UTC+10:00) Hora del Este de Australia
- (UTC+11:00) Hora estándar de Salomón
- (UTC+12:00) Hora estándar de Nueva Zelanda
- (UTC-11:00) Hora de las Islas Midway
- (UTC-10:00) Hora estándar de Hawái
- (UTC-09:00) Hora estándar de Alaska
- (UTC-08:00) Hora estándar del Pacífico


- (UTC-07:00) Hora estándar de Phoenix
- (UTC-06:00) Hora estándar central
- (UTC-05:00) Hora Estándar del Este
- (UTC-04:00) Hora de Puerto Rico y las Islas Vírgenes de los Estados Unidos
- (UTC-03:00) Hora estándar de Argentina
- (UTC-02:00) Hora de Georgia del Sur
- (UTC-01:00) Hora de África Central

Example intervalo de tiempo personalizado con un desplazamiento (consola)

El siguiente ejemplo muestra cómo definir un intervalo de tiempo de 12 horas con un desplazamiento el 20 de febrero de 2021 a las 18:30:30 h (Hora del Pacífico).

Para definir un intervalo personalizado con un desplazamiento

1. En Intervalo de tiempo, elija Intervalo personalizado.
2. Para Intervalo de tiempo, realice una de las siguientes acciones:
 - Escriba **12** y, a continuación, elija horas.
 - Escriba **720** y, a continuación, elija minutos.
 - Escriba **43200** y, a continuación, elija segundos.

 Important

El Intervalo de tiempo debe ser un número entero, independientemente de cuál sea la unidad.

3. En Desplazamiento de la fecha, elija 20/02/2021.
4. En Desplazamiento de la hora, escriba **18:30:30**.
5. Para el Desplazamiento de la zona horaria, elija (UTC-08:00), hora estándar del Pacífico.

Si crea la métrica el 1 de julio de 2021, a las 18:30:30 (Hora del Pacífico) o antes, obtendrá el primer resultado de agregación el 1 de julio de 2021 a las 18:30:30 (Hora del Pacífico). El segundo resultado de la agregación se produce el 2 de julio de 2021 a las 06:30:30 (Hora del Pacífico) y así sucesivamente.

Definición de métricas (AWS CLI)

Al definir una métrica para un modelo de activos con la AWS IoT SiteWise API, se especifican los siguientes parámetros:

- `name`: el nombre de la propiedad.
- `dataType`: el tipo de datos de la métrica, que puede ser `DOUBLE` o `STRING`.
- `externalId`: (opcional) es un ID definido por el usuario. Para obtener más información, consulte [Objetos de referencia con objetos externos IDs](#) en la Guía del usuario de AWS IoT SiteWise .
- `expression`: la expresión métrica. Las expresiones métricas pueden utilizar [funciones de agregación](#) para introducir datos de una propiedad para todos los activos asociados en una jerarquía. Para obtener más información, consulte [Uso de expresiones de fórmula](#).
- `window`: el intervalo de tiempo y el desplazamiento de la ventana de saltos de la métrica, donde cada intervalo comienza cuando termina el anterior:
 - `interval`: el intervalo de tiempo para la ventana de caída. El intervalo de tiempo debe estar comprendido entre un minuto y una semana.
 - `offsets`: el desvío de la ventana de caída.

Para obtener más información, consulta [TumblingWindow](#) en la AWS IoT SiteWise Referencia de la API de .

Example intervalo de tiempo personalizado con un desplazamiento (AWS CLI)

El siguiente ejemplo muestra cómo definir un intervalo de tiempo de 12 horas con un desplazamiento el 20 de febrero de 2021 a las 18:30:30 h (Hora del Pacífico).

```
{
  "window": {
    "tumbling": {
      "interval": "12h",
      "offset": " 2021-07-23T18:30:30-08"
    }
  }
}
```

Si crea la métrica el 1 de julio de 2021, a las 18:30:30 (Hora del Pacífico) o antes, obtendrá el primer resultado de agregación el 1 de julio de 2021 a las 18:30:30 (Hora del Pacífico). El segundo

resultado de la agregación se produce el 2 de julio de 2021 a las 06:30:30 (Hora del Pacífico) y así sucesivamente.

- **variables**: una lista de variables que define las otras propiedades del activo o los activos secundarios que se van a utilizar en la expresión. Cada estructura de variable contiene un nombre sencillo para su uso en la expresión y una estructura `value` que identifica qué propiedad vincular a esa variable. La estructura `value` contiene la siguiente información:
 - **propertyId**: el ID de la propiedad desde la cual se extraen los valores. Puede utilizar el nombre de la propiedad en lugar de su identificador si la propiedad está definida en el modelo actual (en lugar de definirse en un modelo de una jerarquía).
 - **hierarchyId**: (opcional) el ID de la jerarquía desde la que se consultan las entidades secundarias de la propiedad. Puede utilizar el nombre de la definición de jerarquía en lugar de su ID. Si omite este valor, AWS IoT SiteWise busca la propiedad en el modelo actual.

Important

Las métricas solo pueden agregar propiedades de tipo entero, doble, booleano o cadena. Los valores booleanos se convierten en 0 (falso) y 1 (verdadero).

Si define variables de entrada de métrica en la expresión de una métrica, esas entradas deben tener el mismo intervalo de tiempo que la métrica de salida.

Las expresiones de fórmula solo pueden generar valores dobles o de cadena. Las expresiones anidadas pueden generar otros tipos de datos, como cadenas, pero la fórmula en su conjunto debe evaluarse como un número o una cadena. Puede usar la [función jp](#) para convertir una cadena en un número. El valor booleano debe ser 1 (verdadero) o 0 (falso). Para obtener más información, consulte [Valores indefinidos, infinitos y de desbordamiento](#).

- **unit**: (opcional) la unidad científica de la propiedad, como mm o Celsius.

Example Definición de métrica de ejemplo

En el ejemplo siguiente se muestra una propiedad de métrica que agrega los datos de medición de temperatura de un activo para calcular la temperatura máxima por hora en Fahrenheit. Este objeto es un ejemplo de un objeto [AssetModelProperty](#) que contiene una [métrica](#). Puede especificar este objeto como parte de la carga de la solicitud de [CreateAssetModel](#) para crear una propiedad de métrica. Para obtener más información, consulte [Creación de un modelo de activos \(AWS CLI\)](#).

```

{
  ...
  "assetModelProperty": [
    ...
    {
      "name": "Max temperature",
      "dataType": "DOUBLE",
      "type": {
        "metric": {
          "expression": "max(temp_f)",
          "variables": [
            {
              "name": "temp_f",
              "value": {
                "propertyId": "Temperature F"
              }
            }
          ],
          "window": {
            "tumbling": {
              "interval": "1h"
            }
          }
        }
      },
      "unit": "Fahrenheit"
    }
  ],
  ...
}

```

Example Ejemplo de definición de métrica que introduce datos procedentes de los activos asociados

El siguiente ejemplo muestra una propiedad métrica que agrega los datos de potencia media de varias turbinas eólicas para calcular la potencia media total de un parque eólico. Este objeto es un ejemplo de un objeto [AssetModelProperty](#) que contiene una [métrica](#). Puede especificar este objeto como parte de la carga de la solicitud de [CreateAssetModel](#) para crear una propiedad de métrica.

```

{
  ...
  "assetModelProperty": [
    ...
    {

```

```
"name": "Total Average Power",
"dataType": "DOUBLE",
"type": {
  "metric": {
    "expression": "avg(power)",
    "variables": [
      {
        "name": "power",
        "value": {
          "propertyId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
          "hierarchyId": "Turbine Asset Model"
        }
      }
    ],
    "window": {
      "tumbling": {
        "interval": "5m"
      }
    }
  },
  "unit": "kWh"
}
],
...
}
```

Uso de expresiones de fórmula

Con las expresiones de fórmula, puede definir las funciones matemáticas para transformar y agregar los datos industriales sin procesar para obtener información sobre su operación. Las expresiones de fórmula combinan literales, operadores, funciones y variables para procesar datos. Para obtener más información acerca de cómo definir propiedades de activos que utilizan expresiones de fórmula, consulte [Datos de transformación \(transformaciones\)](#) y [Agregación de datos desde propiedades y otros activos \(métricas\)](#). Las transformaciones y las métricas son propiedades de la fórmula.

Temas

- [Uso de variables en expresiones de fórmula](#)
- [Uso de literales en expresiones de fórmula](#)
- [Uso de operadores en expresiones de fórmula](#)
- [Uso de constantes en expresiones de fórmula](#)

- [Uso de funciones en expresiones de fórmula](#)
- [Tutoriales de expresiones de fórmula](#)

Uso de variables en expresiones de fórmula

Las variables representan las propiedades AWS IoT SiteWise de los activos en las expresiones de fórmula. Utilice variables para introducir valores de otras propiedades de activos en sus expresiones, de modo que pueda procesar datos de propiedades constantes ([atributos](#)), flujos de datos sin procesar ([mediciones](#)) y otras propiedades de la fórmula.

Las variables pueden representar propiedades de activos del mismo modelo de activo o de modelos de entidades secundarias asociadas. Solo las fórmulas métricas pueden introducir variables procedentes de modelos de entidades secundarias.

Las variables se identifican con nombres diferentes en la consola y en la API.

- AWS IoT SiteWise consola: utilice los nombres de las propiedades de los activos como variables en sus expresiones.
- AWS IoT SiteWise API (AWS CLI, AWS SDKs): defina las variables con la [ExpressionVariable](#) estructura, que requiere un nombre de variable y una referencia a una propiedad de activo. El nombre puede contener letras en mayúsculas y minúsculas, números y guiones bajos. A continuación, utilice nombres de variables para hacer referencia a las propiedades de los activos en sus expresiones.

Los nombres de las variables distinguen mayúsculas de minúsculas.

Para obtener más información, consulte [Definición de transformaciones](#) y [Definición de métricas](#).

Uso de variables para hacer referencia a propiedades

El valor de una variable define la propiedad a la que hace referencia. AWS IoT SiteWise proporciona diferentes formas de hacer esto.

- Por ID de propiedad: puede especificar el ID único (UUID) de la propiedad para identificarla.
- Por nombre: si la propiedad está en el mismo modelo de activos, puede especificar su nombre en el campo de ID de propiedad.

- Por ruta: un valor de variable puede hacer referencia a una propiedad por su ruta. Para obtener más información, consulte [Uso de rutas para hacer referencia a propiedades de modelos compuestos personalizados](#).

Note

La AWS IoT SiteWise consola no admite variables. Las utiliza la AWS IoT SiteWise API, incluidas las AWS Command Line Interface (AWS CLI) y AWS SDKs.

Una variable de la que reciba una respuesta AWS IoT SiteWise incluye información completa sobre el valor, incluidos el ID y la ruta.

Sin embargo, cuando pasas una variable a AWS IoT SiteWise (por ejemplo, en una llamada de «creación» o «actualización»), solo necesitas especificar una de estas variables. Por ejemplo, si especifica la ruta, no necesita proporcionar el ID.

Uso de literales en expresiones de fórmula

AWS IoT SiteWise admite el uso de literales en expresiones y fórmulas. Los literales son valores fijos que representan un tipo de datos específico. En AWS IoT SiteWise, puede definir literales numéricos y de cadena en las expresiones de fórmula. Los literales se pueden usar en diversos contextos, como transformaciones de datos, condiciones de alarma y cálculos de visualización.

- #### Números

Utilice números y notación científica para definir números enteros y dobles. Puede usar la [notación E](#) para expresar números con notación científica.

Ejemplos: 1, 2.0, .9, -23.1, 7.89e3, 3.4E-5

- #### Strings

Utilice los caracteres ' (comillas) y " (comillas dobles) para definir las cadenas. El tipo de comilla para el inicio y el final deben coincidir. Para evitar que una comilla coincida con la que utiliza para declarar una cadena, incluya el carácter de esa comilla dos veces. Es el único carácter de escape de las AWS IoT SiteWise cadenas.

Ejemplos: 'active', "inactive", '{"temp": 52}', '{"temp": "high"}'

Uso de operadores en expresiones de fórmula

Puede utilizar los siguientes operadores comunes en sus expresiones de fórmulas.

Operador	Description (Descripción)
+	<p>Si ambos operandos son números, este operador suma los operandos izquierdo y derecho.</p> <p>Si alguno de los operandos es una cadena, este operador concatena los operandos izquierdo y derecho como cadenas. Por ejemplo, la expresión <code>1 + 2 + " is three"</code> se evalúa como <code>"3 is three"</code>. La cadena concatenada puede contener hasta 1024 caracteres. Si la cadena supera los 1024 caracteres, AWS IoT SiteWise no genera ningún punto de datos para ese cálculo.</p>
-	<p>Resta el operando derecho del operando izquierdo.</p> <p>Este operador solo se puede utilizar con operandos numéricos.</p>
/	<p>Divide el operando izquierdo por el operando derecho.</p> <p>Este operador solo se puede utilizar con operandos numéricos.</p>
*	<p>Multiplica los operandos izquierdo y derecho.</p> <p>Este operador solo se puede utilizar con operandos numéricos.</p>

Operador	Description (Descripción)
\wedge	<p>Eleva el operando izquierdo a la potencia del operando derecho (exponenciación).</p> <p>Este operador solo se puede utilizar con operandos numéricos.</p>
$\%$	<p>Devuelve el resto de la división del operando izquierdo por el operando derecho. El resultado tiene el mismo signo que el operando izquierdo . Este comportamiento difiere de la operación del módulo.</p> <p>Este operador solo se puede utilizar con operandos numéricos.</p>
$x < y$	<p>Devuelve 1 si x es menor que y, de lo contrario 0.</p>
$x > y$	<p>Devuelve 1 si x es mayor que y, de lo contrario 0.</p>
$x \leq y$	<p>Devuelve 1 si x es menor o igual que y, de lo contrario 0.</p>
$x \geq y$	<p>Devuelve 1 si x es mayor o igual que y, de lo contrario 0.</p>
$x == y$	<p>Devuelve 1 si x es igual a y, de lo contrario 0.</p>
$x \neq y$	<p>Devuelve 1 si x no es igual a y, de lo contrario 0.</p>

Operador	Description (Descripción)
! x	<p>Devuelve 1 si x se evalúa como 0 (falso); en caso contrario 0.</p> <p>x se evalúa como falso si:</p> <ul style="list-style-type: none"> • x es un operando numérico y se evalúa como 0. • x se evalúa como una cadena vacía. • x se evalúa como una matriz vacía. • x se evalúa como None.
x and y	<p>Devuelve 0 si x se evalúa como 0 (falso). De lo contrario, devuelve el resultado evaluado de y.</p> <p>x o y se evalúa como falso si:</p> <ul style="list-style-type: none"> • x o y es un operando numérico y se evalúa como 0. • x o y se evalúa como una cadena vacía. • x o y se evalúa como una matriz vacía. • x o y se evalúa como None.
x or y	<p>Devuelve 1 si x se evalúa como 1 (verdadero). De lo contrario, devuelve el resultado evaluado de y.</p> <p>x o y se evalúa como falso si:</p> <ul style="list-style-type: none"> • x o y es un operando numérico y se evalúa como 0. • x o y se evalúa como una cadena vacía. • x o y se evalúa como una matriz vacía. • x o y se evalúa como None.

Operador	Description (Descripción)
<code>not x</code>	<p>Devuelve 1 si x se evalúa como 0 (falso); en caso contrario 0.</p> <p>x se evalúa como falso si:</p> <ul style="list-style-type: none">• x es un operando numérico y se evalúa como 0.• x se evalúa como una cadena vacía.• x se evalúa como una matriz vacía.• x se evalúa como None.
<code>[]</code> <code>s[index]</code>	<p>Devuelve el carácter situado en un índice <code>index</code> de la cadena <code>s</code>. Esto equivale a la sintaxis de índices en Python.</p> <p>Example Ejemplos</p> <ul style="list-style-type: none">• <code>"Hello!"[1]</code> devuelve e.• <code>"Hello!"[-2]</code> devuelve o.

Operador	Description (Descripción)
<pre>[] s[start:end:step]</pre>	<p data-bbox="829 226 1507 359">Devuelve un sector de la cadena <code>s</code>. Esto equivale a la sintaxis de sectores en Python. Este operador tiene los siguientes argumentos:</p> <ul data-bbox="829 405 1507 932" style="list-style-type: none">• <code>start</code>: (opcional) el índice inicial inclusivo del sector. El valor predeterminado es <code>0</code>.• <code>end</code>: (opcional) el índice final exclusivo del sector. El valor predeterminado es la longitud de la cadena.• <code>step</code>: (opcional) el número que se debe incrementar por cada paso del sector. Por ejemplo, puede especificar <code>2</code> para devolver un sector cada dos caracteres o especificar <code>-1</code> para invertir el sector. El valor predeterminado es <code>1</code>. <p data-bbox="829 1014 1507 1146">Puede omitir el argumento <code>step</code> para usar su valor predeterminado. Por ejemplo, <code>s[1:4:1]</code> equivale a <code>s[1:4]</code>.</p> <p data-bbox="829 1192 1507 1367">Los argumentos deben ser enteros o la constante none. Si lo especifican <code>none</code>, AWS IoT SiteWise utiliza el valor por defecto para ese argumento.</p> <p data-bbox="829 1413 1094 1446">Example Ejemplos</p> <ul data-bbox="829 1493 1507 1814" style="list-style-type: none">• <code>"Hello!"[1:4]</code> devuelve <code>"ell"</code>.• <code>"Hello!"[:2]</code> devuelve <code>"He"</code>.• <code>"Hello!"[3:]</code> devuelve <code>"lo!"</code>.• <code>"Hello!"[:-4]</code> devuelve <code>"He"</code>.• <code>"Hello!"[::2]</code> devuelve <code>"Hlo"</code>.• <code>"Hello!"[::-1]</code> devuelve <code>"!olleH"</code>.

Uso de constantes en expresiones de fórmula

En AWS IoT SiteWise, puede utilizar constantes en sus expresiones y fórmulas para representar valores fijos o parámetros predefinidos. Las constantes se pueden usar en diversos contextos, como transformaciones de datos, condiciones de alarma o cálculos de visualización. Al usar constantes, puede simplificar las expresiones y hacerlas más legibles y fáciles de mantener.

Puede utilizar las siguientes constantes matemáticas comunes en sus expresiones. Todas las constantes son incapaces de distinguir mayúsculas y minúsculas.

Note


Si define una variable con el mismo nombre que una constante, la variable anula a la constante.

Constant	Description (Descripción)
pi	El número pi (π): 3.141592653589793
e	El número e: 2.718281828459045
true	Equivale al número 1. En AWS IoT SiteWise, los valores booleanos se convierten en sus equivalentes numéricos.
false	Equivale al número 0. En AWS IoT SiteWise, los valores booleanos se convierten a sus equivalentes numéricos.
none	Equivale a no tener ningún valor. Puede usar esta constante para no generar nada como resultado de una expresión condicional .

Uso de funciones en expresiones de fórmula

Puede utilizar las siguientes funciones para operar con datos de las expresiones de fórmula.

Las transformaciones y las métricas son compatibles con funciones diferentes. La siguiente tabla indica qué tipos de funciones son compatibles con cada tipo de propiedad de una fórmula.

 Note

Se puede incluir un máximo de 10 funciones en una expresión formulaica.

Tipo de función	Transformaciones	Métricas
Uso de funciones comunes en expresiones de fórmula	 Sí	 Sí
Uso de funciones de comparación en expresiones de fórmula	 Sí	 Sí
Uso de funciones condicionales en expresiones de fórmula	 Sí	 Sí
Uso de funciones de cadena en expresiones de fórmula	 Sí	 Sí
Uso de funciones de agregación en expresiones de fórmula	 No	 Sí

Tipo de función	Transformaciones	Métricas
Uso de funciones temporales en expresiones de fórmula	 Sí	 Sí
Uso de funciones de fecha y hora en expresiones de fórmula	 Sí	 Sí

Sintaxis de las funciones

Puede usar la siguiente sintaxis para crear funciones:

Sintaxis normal

Con la sintaxis normal, el nombre de la función va seguido de paréntesis con cero o más argumentos.

function_name(argument1, argument2, argument3, ...). Por ejemplo, las funciones con sintaxis normal podrían ser similares a `log(x)` y `contains(s, substring)`.

Sintaxis uniforme de llamada a funciones (UFCS)

La UFCS permite llamar a funciones mediante la sintaxis de las llamadas a métodos en la programación orientada a objetos. Con UFCS, el primer argumento va seguido de un punto (.) y del nombre de la función, seguidos del resto de los argumentos (si los hay) entre paréntesis.

argument1.function_name(argument2, argument3, ...). Por ejemplo, las funciones con UFCS podrían ser similares a `x.log()` y `s.contains(substring)`.

También puede utilizar el UFCS para encadenar funciones posteriores. AWS IoT SiteWise utiliza el resultado de la evaluación de la función actual como primer argumento de la siguiente función.

Por ejemplo, en lugar de utilizar `message.jp('$.status').lower().contains('fail')`, puede utilizar `contains(lower(jp(message, '$.status')), 'fail')`.

Para obtener más información, consulte el sitio web [Lenguaje de programación D](#).

Note

Puede utilizar el UFCS para todas las AWS IoT SiteWise funciones. AWS IoT SiteWise las funciones no distinguen entre mayúsculas y minúsculas. Por ejemplo, puede usar `lower(s)` y `Lower(s)` indistintamente.

Uso de funciones comunes en expresiones de fórmula

En las [transformaciones](#) y [las métricas](#), puede usar las siguientes funciones para calcular las funciones matemáticas comunes de las transformaciones y las métricas.

Función	Description (Descripción)
<code>abs(x)</code>	Devuelve el valor absoluto de x .
<code>acos(x)</code>	Devuelve el arco coseno de x .
<code>asin(x)</code>	Devuelve el arcoseno de x .
<code>atan(x)</code>	Devuelve el arco tangente de x .
<code>cbrt(x)</code>	Devuelve la raíz cúbica de x .
<code>ceil(x)</code>	Devuelve el entero más cercano mayor que x .
<code>cos(x)</code>	Devuelve el coseno de x .
<code>cosh(x)</code>	Devuelve el coseno hiperbólico de x .
<code>cot(x)</code>	Devuelve la tangente de x .
<code>exp(x)</code>	Devuelve e a la potencia de x .
<code>expm1(x)</code>	Devuelve $\exp(x) - 1$. Utilice esta función para calcular con mayor precisión $\exp(x) - 1$ para los valores pequeños de x .
<code>floor(x)</code>	Devuelve el entero más cercano menor que x .

Función	Description (Descripción)
$\log(x)$	Devuelve el \log_e (base e) de x.
$\log_{10}(x)$	Devuelve el \log_{10} (base 10) de x.
$\log_{1p}(x)$	Devuelve $\log(1 + x)$. Utilice esta función para calcular con mayor precisión $\log(1 + x)$ para los valores pequeños de x.
$\log_2(x)$	Devuelve el \log_2 (base 2) de x.
$\text{pow}(x, y)$	Devuelve x a la potencia de y. Esto equivale a $x ^ y$.
$\text{signum}(x)$	Devuelve el signo de x (-1 para entradas negativas, 0 para entradas cero, +1 para entradas positivas).
$\sin(x)$	Devuelve el seno de x.
$\sinh(x)$	Devuelve el seno hiperbólico de x.
$\text{sqrt}(x)$	Devuelve la raíz cuadrada de x.
$\tan(x)$	Devuelve la tangente de x.
$\tanh(x)$	Devuelve la tangente hiperbólica de x.

Uso de funciones de comparación en expresiones de fórmula

En las [transformaciones](#) y [las métricas](#), puede utilizar las siguientes funciones de comparación para comparar dos valores y obtener resultados 1 (verdadero) o 0 (falso). AWS IoT SiteWise compara las cadenas por orden [lexicográfico](#).

Función	Description (Descripción)
$\text{gt}(x, y)$	Devuelve 1 si x es mayor que y, de lo contrario 0 ($x > y$).

Función	Description (Descripción)
	<p>Esta función no devuelve un valor si x y y son tipos incompatibles, como un número y una cadena.</p>
<code>gte(x, y)</code>	<p>Devuelve 1 si x es mayor o igual que y, de lo contrario 0 ($x \geq y$).</p> <p>AWS IoT SiteWise considera que los argumentos son iguales si están dentro de una tolerancia relativa de $1E-9$. Esto se comporta de forma similar a la función isclose de Python.</p> <p>Esta función no devuelve un valor si x y y son tipos incompatibles, como un número y una cadena.</p>
<code>eq(x, y)</code>	<p>Devuelve 1 si x es igual a y, de lo contrario 0 ($x == y$).</p> <p>AWS IoT SiteWise considera que los argumentos son iguales si están dentro de una tolerancia relativa de $1E-9$. Esto se comporta de forma similar a la función isclose de Python.</p> <p>Esta función no devuelve un valor si x y y son tipos incompatibles, como un número y una cadena.</p>
<code>lt(x, y)</code>	<p>Devuelve 1 si x es menor que y, de lo contrario 0 ($x < y$).</p> <p>Esta función no devuelve un valor si x y y son tipos incompatibles, como un número y una cadena.</p>


Función	Description (Descripción)
<code>lte(x, y)</code>	<p>Devuelve 1 si x es menor o igual que y, de lo contrario 0 ($x \leq y$).</p> <p>AWS IoT SiteWise considera que los argumentos son iguales si están dentro de una tolerancia relativa de $1E-9$. Esto se comporta de forma similar a la función isclose de Python.</p> <p>Esta función no devuelve un valor si x y y son tipos incompatibles, como un número y una cadena.</p>
<code>isnan(x)</code>	<p>Devuelve 1 si x es igual a NaN, de lo contrario 0.</p> <p>Esta función no devuelve un valor si x es una cadena.</p>

Uso de funciones condicionales en expresiones de fórmula

En las [transformaciones](#) y las [métricas](#), puede usar la siguiente función para comprobar una condición y obtener resultados diferentes si la condición se evalúa como verdadera o falsa.

Función	Description (Descripción)
<code>if(condition, result_if_true, result_if_false)</code>	<p>Evalúa la <code>condition</code> y devuelve <code>result_if_true</code> si la condición se evalúa como verdadera o <code>result_if_false</code> si la condición se evalúa como <code>false</code>.</p> <p><code>condition</code> debe ser un número. Esta función considera <code>0</code> una cadena vacía como <code>false</code> y todo lo demás (incluido NaN) como <code>true</code>. Los valores booleanos se convierten en <code>0</code> (falso) y <code>1</code> (verdadero).</p>

Función	Description (Descripción)
	<p>Se puede devolver la constante none de esta función para descartar el resultado de una condición concreta. Esto significa que puede filtrar los puntos de datos que no cumplan una condición. Para obtener más información, consulte Filtrado de puntos de datos.</p> <p>Example Ejemplos</p> <ul style="list-style-type: none"> • <code>if(0, x, y)</code> devuelve la variable <code>y</code>. • <code>if(5, x, y)</code> devuelve la variable <code>x</code>. • <code>if(gt(temp, 300), x, y)</code> devuelve la variable <code>x</code> si la variable <code>temp</code> es mayor que <code>300</code>. • <code>if(gt(temp, 300), temp, none)</code> devuelve la variable <code>temp</code> si es mayor o igual que <code>300</code>, o <code>none</code> (sin valor) si <code>temp</code> es menor que <code>300</code>. <p>Se recomienda utilizar UFCS para las funciones condicionales anidadas en las que uno o más argumentos son funciones condicionales. Se puede utilizar <code>if(condition, result_if_true)</code> para evaluar una condición y <code>elif(condition, result_if_true, result_if_false)</code> para evaluar condiciones adicionales.</p> <p>Por ejemplo, en lugar de utilizar <code>if(condition1, result1_if_true).elif(condition2, result2_if_true, result2_if_false)</code>, puede utilizar <code>if(condition1, result1_if_true, if(condition2, result2_if_true, result2_if_false))</code>.</p>

Función	Description (Descripción)
	<p>También puede encadenar funciones condicionales intermedias adicionales. Por ejemplo, puede utilizar <code>if(condition1, result1_if_true).elif(condition2, result2_if_true).elif(condition3, result3_if_true, result3_if_false)</code> en lugar de anidar varias instrucciones <code>if</code>, como <code>if(condition1, result1_if_true, if(condition2, result2_if_true, if(condition3, result3_if_true, result3_if_false)))</code>.</p> <div data-bbox="829 814 1511 1087" style="border: 1px solid #f08080; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Important</p> <p>Es necesario usar <code>elif(condition, result_if_true, result_if_false)</code> con UFCS.</p> </div>

Uso de funciones de cadena en expresiones de fórmula

En las [transformaciones](#) y [las métricas](#), puede utilizar las siguientes funciones para operar con cadenas. Para obtener más información, consulte [Uso de cadenas en fórmulas](#).

Important

Las expresiones de fórmula solo pueden generar valores dobles o de cadena. Las expresiones anidadas pueden generar otros tipos de datos, como cadenas, pero la fórmula en su conjunto debe evaluarse como un número o una cadena. Puede usar la [función jp](#) para convertir una cadena en un número. El valor booleano debe ser 1 (verdadero) o 0 (falso). Para obtener más información, consulte [Valores indefinidos, infinitos y de desbordamiento](#).

Función	Description (Descripción)
<code>len(s)</code>	Devuelve la longitud de la cadena <code>s</code> .
<code>find(s, substring)</code>	Devuelve el índice de la cadena <code>substring</code> en la cadena <code>s</code> .
<code>contains(s, substring)</code>	Devuelve 1 si la cadena <code>s</code> contiene la cadena <code>substring</code> , en caso contrario 0.
<code>upper(s)</code>	Devuelve la cadena <code>s</code> en mayúsculas.
<code>lower(s)</code>	Devuelve la cadena <code>s</code> en minúsculas.
<code>jp(s, json_path)</code>	<p>Evalúa la cadena <code>s</code> con la JsonPath expresión <code>json_path</code> y devuelve el resultado.</p> <p>Utilice esta función para hacer lo siguiente:</p> <ul style="list-style-type: none">• Extraer un valor, una matriz o un objeto de una estructura JSON serializada.• Convertir una cadena en un número. Por ejemplo, la fórmula <code>jp('111', '\$')</code> devuelve 111 como un número. <p>Para extraer un valor de cadena de una estructura JSON y devolverlo como un número, debe utilizar varias funciones anidadas <code>jp</code>. La función externa <code>jp</code> extrae la cadena de la estructura JSON y la función interna <code>jp</code> convierte la cadena en un número.</p> <p>La cadena <code>json_path</code> debe contener un literal de cadena. Esto significa que <code>json_path</code> no puede ser una expresión que se evalúe como cadena.</p>

Función	Description (Descripción)
	<p>Example Ejemplos</p> <ul style="list-style-type: none"> • <code>jp('{"status":"active","value":15}', '\$.value')</code> devuelve 15. • <code>jp('{"measurement":{"reading":25,"confidence":0.95}}', '\$.measurement.reading')</code> devuelve 25. • <code>jp('[2,8,23]', '\$[2]')</code> devuelve 23. • <code>jp('{"values":[3,6,7]}', '\$.values[1]')</code> devuelve 6. • <code>jp('111', '\$')</code> devuelve 111. • <code>jp(jp('{"measurement":{"reading":25,"confidence":"0.95"}}', '\$.measurement.confidence'), '\$')</code> devuelve 0.95.
<p><code>join(s0, s1, s2, s3, ...)</code></p>	<p>Devuelve una cadena concatenada con un delimitador. Esta función usa la primera cadena de entrada como delimitador y une las cadenas de entrada restantes. Esto se comporta de forma similar a la función join (CharSequence delimiter, CharSequence... elements) de Java.</p> <p>Example Ejemplos</p> <ul style="list-style-type: none"> • <code>join("-", "aa", "bb", "cc")</code> devuelve aa-bb-cc

Función	Description (Descripción)
<code>format(expression: "format")</code> o <code>format("format", expression)</code>	<p>Devuelve una cadena en el formato especificado. Esta función evalúa <code>expression</code> como un valor y, a continuación, devuelve el valor en el formato especificado. Esto se comporta de forma similar a la función format(String format, Object... args) de Java. Para obtener más información sobre los formatos compatibles, consulte el apartado Conversions en la sección Class Formatter de la plataforma Java, edición estándar 7, especificación de API.</p> <p>Example Ejemplos</p> <ul style="list-style-type: none">• <code>format(100+1: "d")</code> devuelve una cadena, 101.• <code>format("The result is %d", 100+1)</code> devuelve una cadena, The result is 101.

Función	Description (Descripción)
f'expression'	<p>Devuelve una cadena concatenada. Con esta función formateada, puede utilizar una expresión sencilla para concatenar y formatear cadenas. Estas funciones pueden contener expresiones anidadas. Puede utilizar {} (corchetes) para interpolar expresiones. Esto se comporta de forma similar a los literales de cadena formateados de Python.</p> <p>Example Ejemplos</p> <ul style="list-style-type: none"> f'abc{1+2: "f"}d' devuelve abc3.000000d . Para evaluar esta expresión de ejemplo, haga lo siguiente: <ol style="list-style-type: none"> format(1+2: "f") devuelve un número de punto flotante, 3.000000. join(' ', "abc", 1+2, 'd') devuelve una cadena, abc3.000000d . <p>También puede escribir la expresión de la siguiente manera: join(' ', "abc", format(1+2: "f"), 'd') .</p>

Uso de funciones de agregación en expresiones de fórmula

Solo en [las métricas](#), puede utilizar las siguientes funciones que agregan valores de entrada en cada intervalo de tiempo y calculan un único valor de salida. Las funciones de agregación pueden agregar datos de activos asociados.

Los argumentos de las funciones de agregación pueden ser [variables](#), [literales numéricos](#), [funciones temporales](#), expresiones anidadas o funciones de agregación. La fórmula `max(latest(x), latest(y), latest(z))` utiliza una función de agregación como argumento y devuelve el valor actual más grande de las propiedades x, y y z.

Puede utilizar expresiones anidadas en las funciones de agregación. Cuando se utilizan expresiones anidadas, se aplican las reglas siguientes:

- Cada argumento solo puede tener una variable.

Example

Por ejemplo, $\text{avg}(x*(x-1))$ y $\text{sum}(x/2)/\text{avg}(y^2)$ son compatibles.

Por ejemplo, $\text{min}(x/y)$ no es compatible.

- Cada argumento puede tener expresiones anidadas de varios niveles.

Example

Por ejemplo, $\text{sum}(\text{avg}(x^2)/2)$ no se admite.

- Cada argumento puede tener variables diferentes.

Example

Por ejemplo, $\text{sum}(x/2, y*2)$ no se admite.

Note

- Si las expresiones contienen medidas, AWS IoT SiteWise utiliza los últimos valores del intervalo de tiempo actual para que las mediciones calculen los agregados.
- Si las expresiones contienen atributos, AWS IoT SiteWise utiliza los valores más recientes de los atributos para calcular los agregados.

Función	Description (Descripción)
$\text{avg}(x_0, \dots, x_n)$	<p>Devuelve la media de los valores de las variables dadas durante el intervalo de tiempo actual.</p> <p>Esta función genera un punto de datos solo si las variables dadas tienen al menos un punto de datos durante el intervalo de tiempo actual.</p>

Función	Description (Descripción)
$\text{sum}(x_0, \dots, x_n)$	<p>Devuelve la suma de los valores de las variables dadas durante el intervalo de tiempo actual.</p> <p>Esta función genera un punto de datos solo si las variables dadas tienen al menos un punto de datos durante el intervalo de tiempo actual.</p>
$\text{min}(x_0, \dots, x_n)$	<p>Devuelve el mínimo de los valores de las variables dadas durante el intervalo de tiempo actual.</p> <p>Esta función genera un punto de datos solo si las variables dadas tienen al menos un punto de datos durante el intervalo de tiempo actual.</p>
$\text{max}(x_0, \dots, x_n)$	<p>Devuelve el máximo de los valores de las variables dadas durante el intervalo de tiempo actual.</p> <p>Esta función genera un punto de datos solo si las variables dadas tienen al menos un punto de datos durante el intervalo de tiempo actual.</p>
$\text{count}(x_0, \dots, x_n)$	<p>Devuelve el número total de puntos de datos para las variables dadas durante el intervalo de tiempo actual. Para obtener más información acerca de cómo contar el número de puntos de datos que cumplen una condición, consulte Recuento de puntos de datos que coinciden con una condición.</p> <p>Esta función calcula un punto de datos para cada intervalo de tiempo.</p>

Función	Description (Descripción)
<code>stdev(x₀, ..., x_n)</code>	<p>Devuelve la desviación estándar de los valores de las variables dadas durante el intervalo de tiempo actual.</p> <p>Esta función genera un punto de datos solo si las variables dadas tienen al menos un punto de datos durante el intervalo de tiempo actual.</p>

Uso de funciones temporales en expresiones de fórmula

Use funciones temporales para devolver valores basados en marcas de tiempo de puntos de datos.

Uso de funciones temporales en las métricas

Solo en las [métricas](#), puede utilizar las siguientes funciones que devuelven valores basados en marcas temporales de puntos de datos.

Los argumentos de las funciones temporales deben ser propiedades del modelo de activo local o expresiones anidadas. Esto significa que no se pueden usar propiedades de modelos de entidades secundarias en las funciones temporales.

Puede usar expresiones anidadas en las funciones temporales. Cuando se utilizan expresiones anidadas, se aplican las reglas siguientes:

- Cada argumento solo puede tener una variable.
Por ejemplo, `latest(t*9/5 + 32)` no se admite.
- Los argumentos no pueden ser funciones de agregación.
Por ejemplo, `first(sum(x))` no es compatible.

Función	Description (Descripción)
<code>first(x)</code>	Devuelve el valor de la variable dada con la marca temporal más temprana durante el intervalo de tiempo especificado.

Función	Description (Descripción)
<code>last(x)</code>	Devuelve el valor de la variable dada con la última marca temporal durante el intervalo de tiempo especificado.
<code>earliest(x)</code>	<p>Devuelve el último valor de la variable dada antes del inicio del intervalo de tiempo actual.</p> <p>Esta función calcula un punto de datos para cada intervalo de tiempo, si la propiedad de entrada tiene al menos un punto de datos en su historial. Para obtener más información, consulte time-range-defintion.</p>
<code>latest(x)</code>	<p>Devuelve el último valor de la variable dada con la última marca de tiempo antes del fin del intervalo de tiempo especificado.</p> <p>Esta función calcula un punto de datos para cada intervalo de tiempo, si la propiedad de entrada tiene al menos un punto de datos en su historial. Para obtener más información, consulte time-range-defintion.</p>

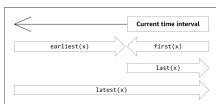
Función	Description (Descripción)
<code>statetime(x)</code>	<p>Devuelve la cantidad de tiempo en segundos que las variables dadas son positivas durante el intervalo de tiempo especificado. Puede usar las funciones de comparación para crear una propiedad de transformación para que la función <code>statetime</code> la consuma.</p> <p>Por ejemplo, si tiene una propiedad <code>Idle</code> que es <code>0</code> o <code>1</code>, puede calcular el tiempo de inactividad por intervalo de tiempo con esta expresión: <code>IdleTime = statetime(Idle)</code>. Para obtener más información, consulte el escenario de <code>statetime</code> de ejemplo.</p> <p>La función no admite propiedades de métricas como variables de entrada.</p> <p>Esta función calcula un punto de datos para cada intervalo de tiempo, si la propiedad de entrada tiene al menos un punto de datos en su historial.</p>

Función	Description (Descripción)
<code>TimeWeightedAvg(x, [interpolation])</code>	<p>Devuelve el promedio de los datos de entrada ponderados con los intervalos de tiempo entre puntos.</p> <p>Consulte los Parámetros de las funciones ponderadas por tiempo para obtener detalles sobre el cálculo y los intervalos.</p> <p>El argumento opcional <code>interpolation</code> debe ser una constante de cadena:</p> <ul style="list-style-type: none">• <code>locf</code> – Esta es la opción predeterminada. El cálculo utiliza el algoritmo de cálculo de última observación realizada para los intervalos entre puntos de datos. En este enfoque, el punto de datos se calcula como el último valor observado hasta la siguiente entrada de punto de datos con marca temporal. <p>El valor que sigue a un punto de datos válido se extrapola como su valor hasta la siguiente marca temporal de puntos de datos.</p> <ul style="list-style-type: none">• <code>linear</code>: el cálculo utiliza el algoritmo de cálculo de interpolación lineal para los intervalos entre puntos de datos. <p>El valor entre dos puntos de datos válidos se extrapola como una interpolación lineal entre los valores de esos puntos de datos.</p> <p>El valor entre puntos de datos válidos o nulos o el valor posterior al último punto de datos válido se extrapolará como un punto de datos válido.</p>

Función	Description (Descripción)
<code>TimeWeightedStDev(x, [algo])</code>	<p>Devuelve la desviación estándar de los datos de entrada ponderada con los intervalos de tiempo entre puntos.</p> <p>Consulte los Parámetros de las funciones ponderadas por tiempo para obtener detalles sobre el cálculo y los intervalos.</p> <p>El cálculo utiliza el algoritmo de cálculo de última observación realizada para los intervalos entre puntos de datos. En este enfoque, el punto de datos se calcula como el último valor observado hasta la siguiente entrada de punto de datos con marca temporal. El peso se calcula como el intervalo de tiempo en segundos entre los puntos de datos o los límites de las ventanas.</p> <p>El argumento opcional <code>algo</code> debe ser una constante de cadena:</p> <ul style="list-style-type: none">• <code>f</code> – Esta es la opción predeterminada. Devuelve una varianza muestral ponderada imparcial con ponderaciones de frecuencia, que <code>TimeWeight</code> se calcula en segundos. Por lo general, este algoritmo se asume dentro de la desviación estándar y se conoce como corrección de Bessel de desviación estándar para muestras ponderadas.• <code>p</code>: devuelve la varianza de las muestras ponderadas sesgada, también conocida como varianza de la población. <p>Para el cálculo se utilizan las siguientes fórmulas, donde:</p>

Función	Description (Descripción)
	<ul style="list-style-type: none"> • S_p = Desviación estándar de la población • S_f = desviación estándar de la frecuencia • X_i = Datos de entrada • ω_i = ponderación que equivale al intervalo de tiempo en segundos • μ^* = media ponderada de los datos entrantes <p>Ecuación para la desviación estándar de la población:</p> $S_p^2 = \frac{\sum_{i=1}^N \omega_i (x_i - \mu^*)^2}{\sum_{i=1}^N \omega_i}$ <p>Ecuación para la desviación estándar de la frecuencia:</p> $S_f^2 = \frac{\sum_{i=1}^N \omega_i (x_i - \mu^*)^2}{\sum_{i=1}^N \omega_i - 1}$

El siguiente diagrama muestra cómo se AWS IoT SiteWise calculan las funciones temporales `first`, `last`, `earliest` y `latest`, en relación con el intervalo de tiempo actual.



Note

- El intervalo de tiempo para `first(x)`, `last(x)` es (inicio de la ventana actual, fin de la ventana actual].
- El intervalo de tiempo para `latest(x)` es (inicio del tiempo, fin de la ventana actual].

- El intervalo de tiempo para `earliest(x)` es (inicio del tiempo, fin de la ventana anterior].

Parámetros de funciones con ponderación de tiempo

Las funciones con ponderación temporal calculadas para la ventana de agregación tienen en cuenta lo siguiente:

- Puntos de datos dentro de la ventana
- Intervalos de tiempo entre puntos de datos
- Último punto de datos antes de la ventana
- Primer punto de datos después de la ventana (para algunos algoritmos)

Términos:

- Punto de datos nulo: cualquier punto de datos con una calidad no buena o un valor que no sea numérico. Estos no se tienen en cuenta para el cálculo de los resultados de una ventana.
- Intervalo nulo: el intervalo que sigue a un punto de datos nulo. El intervalo anterior al primer punto de datos conocido también se considera un intervalo nulo.
- Punto de datos válido: cualquier punto de datos con buena calidad y valor numérico.

Note

- AWS IoT SiteWise solo consume datos GOOD de calidad cuando calcula las transformaciones y las métricas. Ignora los puntos de datos UNCERTAIN y BAD.
- El intervalo anterior al primer punto de datos conocido se considera un intervalo nulo. Para obtener más información, consulte [the section called “Tutoriales de expresiones de fórmula”](#).

El intervalo posterior al último punto de datos conocido continúa indefinidamente y afecta a todas las ventanas siguientes. Cuando llega un nuevo punto de datos, la función vuelve a calcular el intervalo.

Siguiendo las reglas anteriores, se calcula el resultado agregado de la ventana y se limita a los límites de la ventana. De forma predeterminada, la función solo envía el resultado de la ventana si toda la ventana es un intervalo válido.

Si el intervalo válido de la ventana es inferior a la longitud de la ventana, la función no envía la ventana.

Cuando cambian los puntos de datos que afectan al resultado de la ventana, la función vuelve a calcular la ventana, incluso si los puntos de datos están fuera de la ventana.

Si la propiedad de entrada tiene al menos un punto de datos en su historial y se ha iniciado un cálculo, la función calcula las funciones agregadas con ponderación temporal para cada intervalo de tiempo.

Example Escenario de statetime de ejemplo

Considere un ejemplo en el que tiene un activo con las siguientes propiedades:

- **Idle**: una medición que es 0 o 1. Cuando el valor es 1, la máquina está inactiva.
- **Idle Time**: una métrica que utiliza la fórmula `statetime(Idle)` para calcular la cantidad de tiempo en segundos en que la máquina está inactiva, por cada intervalo de 1 minuto.

La propiedad **Idle** tiene los siguientes puntos de datos.

Timestamp	14:00:00 h	14:00:30 h	14:01:15 h	14:02:45 h	14:04:00 h
Idle	0	1	1	0	0

AWS IoT SiteWise calcula la **Idle Time** propiedad cada minuto a partir de los valores de **Idle**. Una vez realizado este cálculo, la propiedad **Idle Time** tiene los siguientes puntos de datos.

Timestamp	14:00:00 h	14:01:00 h	14:02:00 h	14:03:00 h	14:04:00 h
Idle Time	N/A	30	60	45	0

AWS IoT SiteWise realiza los siguientes cálculos **Idle Time** al final de cada minuto.

- A las 14:00 h (de las 13:59 h a las 14:00 h)
 - No hay datos para **Idle** antes de las 14:00 h, por lo que no se calcula ningún punto de datos.
- A las 14:01 h (de las 14:00 h a las 14:01 h)
 - A las 14:00:00 h, la máquina está activa (**Idle** es 0).

- A las 14:00:30 h, la máquina está inactiva (Idle es 1).
- Idle no vuelve a cambiar antes del final del intervalo de las 14:01:00 h, por lo que el valor de Idle Time es de 30 segundos.
- A las 14:02 h (de las 14:01 h a las 14:02 h)
 - A las 14:01:00 h, el equipo está inactivo (según el último punto de datos de las 14:00:30 h).
 - A las 14:01:15 h, la máquina sigue inactiva.
 - Idle no vuelve a cambiar antes del final del intervalo de las 14:02:00 h, por lo que el valor de Idle Time es de 60 segundos.
- A las 14:03 h (de las 14:02 h a las 14:03 h)
 - A las 14:02:00 h, el equipo está inactivo (según el último punto de datos de las 14:01:15 h).
 - A las 14:02:45 h, la máquina está activa.
 - Idle no vuelve a cambiar antes del final del intervalo de las 14:03:00 h, por lo que el valor de Idle Time es de 45 segundos.
- A las 14:04 h (de las 14:03 h a las 14:04 h)
 - A las 14:03:00 h, el equipo está activo (según el último punto de datos de las 14:02:45 h).
 - Idle no vuelve a cambiar antes del final del intervalo de las 14:04:00 h, por lo que el valor de Idle Time es de 0 segundos.

Example Ejemplo TimeWeightedAvg y TimeWeightedStDev escenario

Las siguientes tablas proporcionan ejemplos de entradas y salidas para estas métricas de ventana de un minuto: `Avg(x)`, `TimeWeightedAvg(x)`, `TimeWeightedAvg(x, "linear")`, `stDev(x)`, `timeWeightedStDev(x)`, `timeWeightedStDev(x, 'p')`.

Ejemplo de entrada para una ventana agregada de un minuto:


Note

Todos estos puntos de datos tienen calidad GOOD.

03:00:00	4.0
03:01:00	2.0

03:01:10	8.0
03:01:50	20.0
03:02:00	14.0
03:02:05	10.0
03:02:10	3.0
03:02:30	20.0
03:03:30	0.0

Salida de resultados agregados:

 Note

Ninguno: no se produjo un resultado para esta ventana.

Time	Avg(x)	TimeWeightedAvg(x)	TimeWeightedAvg(X, "linear")	stDev(X)	timeWeightedStDev(x)	timeWeightedStDev(x, 'p')
3:00:00	4	Ninguno	Ninguno	0	Ninguno	Ninguno
3:01:00	2	4	3	0	0	0
3:02:00	14	9	13	6	5,4306100 41581775	5,3851648 07134504
3:03:00	11	13	12,875	8,5400374 531753	7,7240544 37220943	7,6594168 62050705
3:04:00	0	10	2,5	0	10,084389 681792215	10

Time	Avg(x)	TimeWeightedAvg(x)	TimeWeightedAvg(X, "linear")	stDev(X)	timeWeightedStDev(x)	timeWeightedStDev(x, 'p')
3:05:00	Ninguno	0	0	Ninguno	0	0

Uso de funciones temporales en las transformaciones

Solo en las [transformaciones](#), puede utilizar la función `pretrigger()` para recuperar el valor de calidad GOOD de una variable antes de actualizar la propiedad que inició el cálculo de la transformación actual.

Considere un ejemplo en el que un fabricante utiliza AWS IoT SiteWise para monitorizar el estado de una máquina. El fabricante utiliza las siguientes medidas y transformaciones para representar el proceso:

- Una medición, `current_state`, que puede ser 0 o 1.
 - Si la máquina está en estado de limpieza, `current_state` equivale a 1.
 - Si la máquina está en estado de fabricación, `current_state` equivale a 0.
- Una transformación, `cleaning_state_duration`, que equivale a `if(pretrigger(current_state) == 1, timestamp(current_state) - timestamp(pretrigger(current_state)), none)`. Esta transformación devuelve el tiempo que la máquina ha estado en estado de limpieza en segundos, en formato de tiempo Unix. Para obtener más información, consulte [Uso de funciones condicionales en expresiones de fórmula](#) y la función [marca temporal\(\)](#).

Si la máquina permanece en estado de limpieza más tiempo del esperado, el fabricante podría investigar la máquina.

También puede utilizar la función `pretrigger()` en transformaciones multivariantes. Por ejemplo, dispone de dos mediciones denominadas `x` y `y`, y una transformación `z`, que equivale a `x + y + pretrigger(y)`. En la siguiente tabla se muestran los valores de `x`, `y` y `z` desde las 9:00 h hasta las 9:15 h.

Note

- En este ejemplo se presupone que los valores de las mediciones llegan en orden cronológico. Por ejemplo, el valor de x para las 09:00 h llega antes que el valor de x para las 09:05 h.
- Si los puntos de datos de las 9:05 h llegan antes que los puntos de datos de las 9:00 h, no se calcula z a las 9:05 h.
- Si el valor de x para las 9:05 h llega antes que el valor de x para las 09:00 h y los valores de y llegan en orden cronológico, z equivale a $22 = 20 + 1 + 1$ a las las 9:05 h.

	09:00 h	09:05 h	09:10 h	09:15 h
x	10	20		30
y	1	2	3	
$z = x + y$ + pretrigge $r(y)$	y no recibe ningún punto de datos antes de las 09:00 h. Por lo tanto, z no se calcula a las 09:00 h.	$23 = 20 + 2 + 1$ pretrigge $r(y)$ equivale a 1.	$25 = 20 + 3 + 2$ x no recibe un nuevo punto de datos. pretrigge $r(y)$ equivale a 2.	$36 = 30 + 3 + 3$ y no recibe un nuevo punto de datos. Por lo tanto, pretrigge $r(y)$ equivale a 3 a las 09:15 h.

Uso de funciones de fecha y hora en expresiones de fórmula

En las [transformaciones](#) y las [métricas](#), puede utilizar las funciones de fecha y hora de las siguientes maneras:

- Para recuperar la marca temporal actual de un punto de datos en UTC o en la zona horaria local.
- Para construir marcas temporales con argumentos, como `year`, `month` y `day_of_month`.
- Para extraer un período de tiempo, como un año o un mes, con el argumento `unix_time`.

Función	Description (Descripción)
now()	Devuelve la fecha y hora actuales, en segundos, en formato de tiempo Unix.
timestamp()	<ul style="list-style-type: none"> En las transformaciones, la función devuelve la marca temporal, en segundos, del mensaje de entrada en formato de tiempo Unix. <p>Solo en las transformaciones, puede realizar una de las siguientes acciones:</p> <ul style="list-style-type: none"> Proporcionar una variable como argumento a la función. La función <code>timestamp(<i>variable-name</i>)</code> devuelve la marca temporal, en segundos, del valor más reciente de calidad GOOD de la variable, en formato de tiempo Unix. <p>Por ejemplo, si su activo tiene un nombre de propiedad de transformación <code>Temperature_F</code> que usa la fórmula $9/5 * \text{Temperature_C}$ para convertir cada punto de datos de temperatura de grados Celsius a Fahrenheit, puede usar la función <code>timestamp(Temperature_F)</code> para obtener la marca temporal valor de calidad GOOD más reciente para la propiedad <code>Temperature_F</code>.</p> <ul style="list-style-type: none"> Utilizar la función <code>pretrigger()</code> como argumento de la función. La función <code>timestamp(pretrigger(<i>variable-name</i>))</code> devuelve la marca temporal, en segundos, del valor más reciente de calidad GOOD de la variable especificada antes de actualizar la propiedad que inició el cálculo de la transformación actual, en

Función	Description (Descripción)
	<p>formato de tiempo Unix. Para obtener más información, consulte Uso de funciones temporales en las transformaciones.</p> <ul style="list-style-type: none">• En las métricas, la función devuelve la marca de tiempo recuperada al final de la ventana actual, en segundos, en formato de tiempo Unix.

Función	Description (Descripción)
<code>mktime(time_zone, year, month, day_of_month, hour, minute, second)</code>	<p>Devuelve el tiempo de entrada, en segundos, en formato de tiempo Unix.</p> <p>Los siguientes requisitos se aplican al uso de esta función:</p> <ul style="list-style-type: none">• El argumento de zona horaria debe ser una cadena entre comillas ('UTC'). Si no se especifica, la zona horaria predeterminada es UTC. <p>El argumento de zona horaria puede ser el primero o el último argumento.</p> <ul style="list-style-type: none">• Los argumentos año, mes, día del mes, hora, minuto y segundo deben estar en orden.• Los argumentos de año, mes y fecha son obligatorios. <p>Los siguientes límites se aplican al uso de esta función:</p> <ul style="list-style-type: none">• <code>year</code>: los valores válidos se encuentran entre 1970 y 2250.• <code>month</code>: los valores válidos se encuentran entre 1 y 12.• <code>day-of-month</code> : los valores válidos se encuentran entre 1 y 31.• <code>hour</code>: los valores válidos se encuentran entre 0 y 23.• <code>minute</code>: los valores válidos se encuentran entre 0 y 59.• <code>second</code>: los valores válidos se encuentran entre 0 y 60. Puede ser un número de punto flotante.

Función	Description (Descripción)
	<p>Ejemplos:</p> <ul style="list-style-type: none">• <code>mktime(2020, 2, 29)</code>• <code>mktime('UTC+3', 2021, 12, 31, 22)</code>• <code>mktime(2022, 10, 13, 2, 55, 13.68, 'PST')</code>

Función	Description (Descripción)
<code>localtime(unix_time, time_zone)</code>	<p>Devuelve el año, el día del mes, el día de la semana, el día del año, la hora, el minuto o el segundo de la zona horaria especificada a partir del tiempo Unix.</p> <p>Los siguientes requisitos se aplican al uso de esta función:</p> <ul style="list-style-type: none">• El argumento de zona horaria debe ser una cadena entre comillas ('UTC'). Si no se especifica, la zona horaria predeterminada es UTC.• El argumento tiempo Unix es el tiempo en segundos, en formato de tiempo Unix. El rango válido es de 1 a 31556889864403199. Puede ser un número de punto flotante. <p>Respuesta de ejemplo: 2007-12-03T10:15:30+01:00[Europe/Paris]</p> <p><code>localtime(unix_time, time_zone)</code> no es una función independiente. Las funciones <code>year()</code>, <code>mon()</code>, <code>mday</code>, <code>wday()</code>, <code>yday()</code>, <code>hour()</code>, <code>minute()</code> y <code>sec()</code> toman <code>localtime(unix_time, time_zone)</code> como argumento.</p> <p>Ejemplos:</p> <ul style="list-style-type: none">• <code>year(localtime('GMT', 1605898608.8113723))</code>• <code>now().localtime().year()</code>• <code>timestamp().localtime('PST').year()</code>

Función	Description (Descripción)
	<ul style="list-style-type: none"> <code>localtime(1605289736, 'Europe/London').year()</code>
<code>year(localtime(unix_time, time_zone))</code>	Devuelve el año desde <code>localtime(unix_time, time_zone)</code> .
<code>mon(localtime(unix_time, time_zone))</code>	Devuelve el mes desde <code>localtime(unix_time, time_zone)</code> .
<code>mday(localtime(unix_time, time_zone))</code>	Devuelve el día del mes desde <code>localtime(unix_time, time_zone)</code> .
<code>wday(localtime(unix_time, time_zone))</code>	Devuelve el día de la semana desde <code>localtime(unix_time, time_zone)</code> .
<code>yday(localtime(unix_time, time_zone))</code>	Devuelve el día del año desde <code>localtime(unix_time, time_zone)</code> .
<code>hour(localtime(unix_time, time_zone))</code>	Devuelve la hora desde <code>localtime(unix_time, time_zone)</code> .
<code>minute(localtime(unix_time, time_zone))</code>	Devuelve el minuto desde <code>localtime(unix_time, time_zone)</code> .
<code>sec(localtime(unix_time, time_zone))</code>	Devuelve el segundo desde <code>localtime(unix_time, time_zone)</code> .

Formatos de zona horaria admitidos

Puede especificar el argumento de zona horaria de las siguientes maneras:

- Desplazamiento de la zona horaria: especifique 'Z' si es UTC o un desplazamiento ('+2' o '-5').
- Desfase IDs : combina una abreviatura de zona horaria y un desfase. Por ejemplo, 'GMT+2' y 'UTC-01:00'. La abreviatura de la zona horaria debe contener solo tres letras.
- Basado en regiones IDs : por ejemplo, 'Etc/GMT+12' y 'Pacific/Pago_Pago'.

Se admiten abreviaturas de zona horaria

Las funciones de fecha y hora admiten las siguientes abreviaturas de zona horaria de tres letras:

- WEST: -05:00
- HOST: -10:00
- HST: -07:00
- ACT: Australia/Darwin
- AET: Australia/Sídney
- AGT - America/Argentina/Buenos_Aires
- ARTE: África/El Cairo
- AST: América/Anchorage
- BET: América/Sao_Paulo
- BST: Asia/Daca
- CAT: África/Harare
- CET: Europa/París
- CNT: América/St_Johns
- CST: América/Chicago
- CTT - Asia/Shanghai
- EAT: África/Adís_Abeba
- IET - America/Indiana/Indianapolis
- IST: Asia/Calcuta
- JST: Asia/Tokio
- MIT: Pacífico/Apia
- NET: Asia/Ereván
- NST: Pacífico/Auckland
- PLT: Asia/Karachi
- PRT: América/Puerto_Rico
- PST: America/Los_Ángeles
- SST: Pacífico/Guadalcanal

- VST: Asia/Ho_Chi_Minh

Compatible según la región IDs

Las funciones de fecha y hora son compatibles con las siguientes regiones IDs, organizadas según su relación con UTC+ 00:00:

- Etc/GMT+12 (UTC-12:00)
- Pacífico/Pago_Pago (UTC-11:00)
- Pacífico/Samoa (UTC-11:00)
- Pacífico/Niue (UTC-11:00)
- EE. UU./Samoa (UTC-11:00)
- Etc/GMT+11 (UTC-11:00)
- Pacífico/Midway (UTC-11:00)
- Pacífico/Honolulu (UTC-10:00)
- Pacífico/Rarotonga (UTC-10:00)
- Pacífico/Tahití (UTC-10:00)
- Pacífico/Johnston (UTC-10:00)
- EE. UU./Hawái (UTC-10:00)
- Sistema V/ HST10 (UTC- 10:00)
- Etc/GMT+10 (UTC-10:00)
- Pacífico/Marquesas (UTC-09:30)
- Etc/GMT+9 (UTC-09:00)
- Pacífico/Gambier (UTC-09:00)
- América/Atka (UTC-09:00)
- Sistema V/ YST9 (UTC- 09:00)
- América/Adak (UTC-09:00)
- EE. UU./Aleutianas (UTC-09:00)
- Etc/GMT+8 (UTC-08:00)
- EE. UU./Alaska (UTC-08:00)

- América/Juneau (UTC-08:00)
- América/Metlakatla (UTC-08:00)
- América/Yakutat (UTC-08:00)
- Pacífico/Pitcairn (UTC-08:00)
- América/Sitka (UTC-08:00)
- América/Anchorage (UTC-08:00)
- Sistema V/ PST8 (UTC - 08:00)
- América/Nome (UTC-08:00)
- Sistema V/ YST9 YDT (UTC- 08:00)
- Canadá/Yukon (UTC-07:00)
- EE. UU./Pacífico-Nuevo (UTC-07:00)
- Etc/GMT+7 (UTC-07:00)
- EE. UU./Arizona (UTC-07:00)
- América/Dawson_Creek (UTC-07:00)
- Canadá/Pacífico (UTC-07:00)
- PST8PDT (UTC- 07:00)
- Sistema V/ MST7 (UTC- 07:00)
- América/Dawson (UTC-07:00)
- México/ BajaNorte (UTC- 07:00)
- América/Tijuana (UTC-07:00)
- América/Creston (UTC-07:00)
- América/Hermosillo (UTC-07:00)
- América/Santa_Isabel (UTC-07:00)
- América/Vancouver (UTC-07:00)
- América/Ensenada (UTC-07:00)
- América/Phoenix (UTC-07:00)
- América/Whitehorse (UTC-07:00)
- América/Fort_Nelson (UTC-07:00)

- Sistema V/ PST8 PDT (UTC- 07:00)
- América/Los_Ángeles (UTC-07:00)
- EE. UU./Pacífico (UTC-07:00)
- América/El_Salvador (UTC-06:00)
- América/Guatemala (UTC-06:00)
- América/Belice (UTC-06:00)
- América/Managua (UTC-06:00)
- América/Tegucigalpa (UTC-06:00)
- Etc/GMT+6 (UTC-06:00)
- Pacífico/Pascua (UTC-06:00)
- México/ BajaSur (UTC- 06:00)
- América/Regina (UTC-06:00)
- América/Denver (UTC-06:00)
- Pacífico/Galápagos (UTC-06:00)
- América/Yellowknife (UTC-06:00)
- América/Swift_Current (UTC-06:00)
- América/Inuvik (UTC-06:00)
- América/Mazatlán (UTC-06:00)
- América/Boise (UTC-06:00)
- América/Costa_Rica (UTC-06:00)
- MST7MDT (UTC- 06:00)
- Sistema V/ CST6 (UTC- 06:00)
- América/Chihuahua (UTC-06:00)
- América/Ojinaga (UTC-06:00)
- Chile/ EasterIsland (UTC- 06:00)
- EE. UU./Montaña (UTC-06:00)
- América/Edmonton (UTC-06:00)
- Canadá/Montaña (UTC-06:00)

- América/Cambridge_Bay (UTC-06:00)
- Navajo (UTC-06:00)
- Sistema V/ MST7 MDT (UTC- 06:00)
- Canadá/Saskatchewan (UTC-06:00)
- América/Shiprock (UTC-06:00)
- América/Panamá (UTC-05:00)
- América/Chicago (UTC-05:00)
- América/Eirunepe (UTC-05:00)
- Etc/GMT+5 (UTC-05:00)
- México/General (UTC-05:00)
- América/Porto_Acre (UTC-05:00)
- América/Guayaquil (UTC-05:00)
- América/Rankin_Inlet (UTC-05:00)
- EE. UU./Central (UTC-05:00)
- América/Rainy_River (UTC-05:00)
- America/Indiana/Knox(UTC- 05:00)
- America/North_Dakota/Beulah(UTC- 05:00)
- América/Monterrey (UTC-05:00)
- América/Jamaica (UTC-05:00)
- América/Atikokan (UTC-05:00)
- América/Coral_Harbour (UTC-05:00)
- America/North_Dakota/Center(UTC- 05:00)
- América/Caimán (UTC-05:00)
- America/Indiana/Tell_Ciudad (UTC- 05:00)
- América/Ciudad_de_México (UTC-05:00)
- América/Matamoros (UTC-05:00)
- CST6CDT (UTC- 05:00)
- América/Knox_IN (UTC-05:00)

- América/Bogotá (UTC-05:00)
- América/Menominee (UTC-05:00)
- América/Resolute (UTC-05:00)
- Sistema V/ EST5 (UTC- 05:00)
- Canadá/Central (UTC-05:00)
- Brasil/Acre (UTC-05:00)
- América/Cancún (UTC-05:00)
- América/Lima (UTC-05:00)
- América/Bahía_Banderas (UTC-05:00)
- EE. UU./Indiana-Starke (UTC-05:00)
- América/Rio_Branco (UTC-05:00)
- Sistema V/ CST6 CDT (UTC- 05:00)
- Jamaica (UTC-05:00)
- América/Mérida (UTC-05:00)
- America/North_Dakota/New_Salem (UTC- 05:00)
- América/Winnipeg (UTC-05:00)
- América/Cuiabá (UTC-04:00)
- América/Marigot (UTC-04:00)
- America/Indiana/Petersburg(UTC- 04:00)
- Chile/Continental (UTC-04:00)
- América/Grand_Turk (UTC-04:00)
- Cuba (UTC-04:00)
- Etc/GMT+4 (UTC-04:00)
- América/Manaos (UTC-04:00)
- América/Fort_Wayne (UTC-04:00)
- América/St_Thomas (UTC-04:00)
- América/Anguila (UTC-04:00)
- América/Habana (UTC-04:00)
- EE. UU./Michigan (UTC-04:00)

- América/Barbados (UTC-04:00)
- América/Louisville (UTC-04:00)
- América/Curazao (UTC-04:00)
- América/Guyana (UTC-04:00)
- América/Martinica (UTC-04:00)
- América/Puerto_Rico (UTC-04:00)
- América/Puerto_España (UTC-04:00)
- Sistema V/ AST4 (UTC- 04:00)
- America/Indiana/Vevay(UTC- 04:00)
- America/Indiana/Vincennes(UTC- 04:00)
- América/Kralendijk (UTC-04:00)
- América/Antigua (UTC-04:00)
- América/Indianápolis (UTC-04:00)
- América/Iqaluit (UTC-04:00)
- América/St_Vincent (UTC-04:00)
- America/Kentucky/Louisville(UTC- 04:00)
- América/Dominica (UTC-04:00)
- América/Asunción (UTC-04:00)
- EST5EDT (UTC- 04:00)
- América/Nassau (UTC-04:00)
- America/Kentucky/Monticello(UTC- 04:00)
- Brasil/Oeste (UTC-04:00)
- América/Aruba (UTC-04:00)
- America/Indiana/Indianapolis(UTC- 04:00)
- América/Santiago (UTC-04:00)
- América/La_Paz (UTC-04:00)
- América/Thunder_Bay (UTC-04:00)
- America/Indiana/Marengo(UTC- 04:00)
- América/Blanc-Sablon (UTC-04:00)

- América/Santo_Domingo (UTC-04:00)
- EE. UU./Este (UTC-04:00)
- Canadá/Este (UTC-04:00)
- América/ Port-au-Prince (UTC- 04:00)
- América/San_Bartolomé (UTC-04:00)
- América/Nipigon (UTC-04:00)
- EE. UU./Indiana del Este (UTC-04:00)
- América/Santa_Lucía (UTC-04:00)
- América/Montserrat (UTC-04:00)
- América/Lower_Princes (UTC-04:00)
- América/Detroit (UTC-04:00)
- América/Tórtola (UTC-04:00)
- América/Porto_Velho (UTC-04:00)
- América/Campo_Grande (UTC-04:00)
- América/Virgin (UTC-04:00)
- América/Pangnirtung (UTC-04:00)
- América/Montreal (UTC-04:00)
- America/Indiana/Winamac(UTC- 04:00)
- América/Boa_Vista (UTC-04:00)
- América/Granada (UTC-04:00)
- América/Nueva_York (UTC-04:00)
- América/St_Kitts (UTC-04:00)
- América/Caracas (UTC-04:00)
- América/Guadalupe (UTC-04:00)
- América/Toronto (UTC-04:00)
- Sistema V/ EST5 EDT (UTC- 04:00)
- America/Argentina/Catamarca(UTC- 03:00)
- Canadá/Atlántico (UTC-03:00)
- America/Argentina/Cordoba(UTC- 03:00)

- América/Araguaina (UTC-03:00)
- America/Argentina/Salta(UTC- 03:00)
- Etc/GMT+3 (UTC-03:00)
- América/Montevideo (UTC-03:00)
- Brasil/Este (UTC-03:00)
- America/Argentina/Mendoza(UTC- 03:00)
- America/Argentina/Rio_Gallegos (UTC- 03:00)
- América/Catamarca (UTC-03:00)
- América/Córdoba (UTC-03:00)
- América/Sao_Paulo (UTC-03:00)
- America/Argentina/Jujuy(UTC- 03:00)
- América/Cayenne (UTC-03:00)
- América/Recife (UTC-03:00)
- América/Buenos_Aires (UTC-03:00)
- América/Paramaribo (UTC-03:00)
- América/Moncton (UTC-03:00)
- América/Mendoza (UTC-03:00)
- América/Santarén (UTC-03:00)
- Atlántico/Bermudas (UTC-03:00)
- América/Maceió (UTC-03:00)
- Atlántico/Stanley (UTC-03:00)
- América/Halifax (UTC-03:00)
- Antártica/Rothera (UTC-03:00)
- America/Argentina/San_Luis (UTC- 03:00)
- America/Argentina/Ushuaia(UTC- 03:00)
- Antártica/Palmer (UTC-03:00)
- América/Punta_Arenas (UTC-03:00)
- América/Glace_Bay (UTC-03:00)
- América/Fortaleza (UTC-03:00)

- América/Thule (UTC-03:00)
- America/Argentina/La_Rioja (UTC- 03:00)
- América/Belén (UTC-03:00)
- América/Jujuy (UTC-03:00)
- América/Bahía (UTC-03:00)
- América/Goose_Bay (UTC-03:00)
- America/Argentina/San_Juan (UTC- 03:00)
- America/Argentina/ComodRivadavia(UTC- 03:00)
- America/Argentina/Tucuman(UTC- 03:00)
- América/Rosario (UTC-03:00)
- Sistema V/ AST4 ADT (UTC- 03:00)
- America/Argentina/Buenos_Aires (UTC- 03:00)
- América/St_Johns (UTC-02:30)
- Canadá/Terranova (UTC-02:30)
- América/Miquelón (UTC-02:00)
- Etc/GMT+2 (UTC-02:00)
- América/Godthab (UTC-02:00)
- América/Noronha (UTC-02:00)
- Brasil/ DeNoronha (UTC- 02:00)
- Atlántico/Georgia_del_Sur (UTC-02:00)
- Etc/GMT+1 (UTC-01:00)
- Atlántico/Cabo_Verde (UTC-01:00)
- Pacífico/Kiritimati (UTC+14:00)
- Etc/GMT-14 (UTC+14:00)
- Pacífico/Fakaofu (UTC+13:00)
- Pacífico/Enderbury (UTC+13:00)
- Pacífico/Apia (UTC+13:00)
- Pacífico/Tongatapu (UTC+13:00)
- Etc/GMT-13 (UTC+13:00)

- NZ-CHAT (UTC+12:45)
- Pacífico/Chatham (UTC+12:45)
- Pacífico/Kwajalein (UTC+12:00)
- Antártida/ (UTC+ 12:00McMurdo)
- Pacífico/Wallis (UTC+12:00)
- Pacífico/Fiyi (UTC+12:00)
- Pacífico/Funafuti (UTC+12:00)
- Pacífico/Nauru (UTC+12:00)
- Kwajalein (UTC+12:00)
- NZ (UTC+12:00)
- Pacífico/Wake (UTC+12:00)
- Antártica/Polo_Sur (UTC+12:00)
- Pacífico/Tarawa (UTC+12:00)
- Pacífico/Auckland (UTC+12:00)
- Asia/Kamchatka (UTC+12:00)
- Etc/GMT-12 (UTC+12:00)
- Asia/Anádyr (UTC+12:00)
- Pacífico/Majuro (UTC+12:00)
- Pacífico/Ponapé (UTC+11:00)
- Pacífico/Bougainville (UTC+11:00)
- Antártica/Macquarie (UTC+11:00)
- Pacífico/Pohnpei (UTC+11:00)
- Pacífico/Efaté (UTC+11:00)
- Pacífico/Norfolk (UTC+11:00)
- Asia/Magadán (UTC+11:00)
- Pacífico/Kosrae (UTC+11:00)
- Asia/Sajalín (UTC+11:00)
- Pacífico/Numea (UTC+11:00)
- Etc/GMT-11 (UTC+11:00)

- Asia/Srednekolymk (UTC+11:00)
- Pacífico/Guadalcanal (UTC+11:00)
- Australia/Lord_Howe (UTC+10:30)
- Australia/LHI (UTC+10:30)
- Australia/Hobart (UTC+10:00)
- Pacífico/Yap (UTC+10:00)
- Australia/Tasmania (UTC+10:00)
- Pacífico/Puerto_Moresby (UTC+10:00)
- Australia/ACT (UTC+10:00)
- Australia/Victoria (UTC+10:00)
- Pacífico/Chuuk (UTC+10:00)
- Australia/Queensland (UTC+10:00)
- Australia/Canberra (UTC+10:00)
- Australia/Currie (UTC+10:00)
- Pacífico/Guam (UTC+10:00)
- Pacífico/Truk (UTC+10:00)
- Australia/Nueva_Gales_del_Sur (UTC+10:00)
- Asia/Vladivostok (UTC+10:00)
- Pacífico/Saipán (UTC+10:00)
- DUrville Antártica/Dumont (UTC+ 10:00)
- Australia/Sídney (UTC+10:00)
- Australia/Brisbane (UTC+10:00)
- Etc/GMT-10 (UTC+10:00)
- Asia/Ust-Nera (UTC+10:00)
- Australia/Melbourne (UTC+10:00)
- Australia/Lindeman (UTC+10:00)
- Australia/Norte (UTC+09:30)
- Australia/Yancowinna (UTC+09:30)
- Australia/Adelaida (UTC+09:30)

- Australia/Broken_Hill (UTC+09:30)
- Australia/Sur (UTC+09:30)
- Australia/Darwin (UTC+09:30)
- Etc/GMT-9 (UTC+09:00)
- Pacífico/Palaos (UTC+09:00)
- Asia/Chita (UTC+09:00)
- Asia/Dili (UTC+09:00)
- Asia/Jayapura (UTC+09:00)
- Asia/Yakutsk (UTC+09:00)
- Asia/Pyongyang (UTC+09:00)
- ROK (UTC+09:00)
- Asia/Seúl (UTC+09:00)
- Asia/Khandyga (UTC+09:00)
- Japón (UTC+09:00)
- Asia/Tokio (UTC+09:00)
- Australia/Eucla (UTC+08:45)
- Asia/Kuching (UTC+08:00)
- Asia/Chungking (UTC+08:00)
- Etc/GMT-8 (UTC+08:00)
- Australia/Perth (UTC+08:00)
- Asia/Macao (UTC+08:00)
- Asia/Macao (UTC+08:00)
- Asia/Choybalsan (UTC+08:00)
- Asia/Shanghái (UTC+08:00)
- Antártica/Casey (UTC+08:00)
- Asia/Ulán_Bator (UTC+08:00)
- Asia/Chongqing (UTC+08:00)
- Asia/Ulaanbaatar (UTC+08:00)
- Asia/Taipéi (UTC+08:00)

- Asia/Manila (UTC+08:00)
- RPC (UTC+08:00)
- Asia/Ujung_Pandang (UTC+08:00)
- Asia/Harbin (UTC+08:00)
- Singapur (UTC+08:00)
- Asia/Brunéi (UTC+08:00)
- Australia/Oeste (UTC+08:00)
- Asia/Hong_Kong (UTC+08:00)
- Asia/Macasar (UTC+08:00)
- HongKong (UTC+08:00)
- Asia/Kuala_Lumpur (UTC+08:00)
- Asia/Irkutsk (UTC+08:00)
- Asia/Singapur (UTC+08:00)
- Asia/Pontianak (UTC+07:00)
- Etc/GMT-7 (UTC+07:00)
- Asia/Phnom_Penh (UTC+07:00)
- Asia/Novosibirsk (UTC+07:00)
- Antártica/Davis (UTC+07:00)
- Asia/Tomsk (UTC+07:00)
- Asia/Yakarta (UTC+07:00)
- Asia/Barnaul (UTC+07:00)
- India/Navidad (UTC+07:00)
- Asia/Ho_Chi_Minh (UTC+07:00)
- Asia/Hovd (UTC+07:00)
- Asia/Bangkok (UTC+07:00)
- Asia/Vientián (UTC+07:00)
- Asia/Novokuznetsk (UTC+07:00)
- Asia/Krasnoyarsk (UTC+07:00)
- Asia/Saigón (UTC+07:00)

- Asia/Yangon (UTC+06:30)
- Asia/Rangún (UTC+06:30)
- India/Cocos (UTC+06:30)
- Asia/Kasgar (UTC+06:00)
- Etc/GMT-6 (UTC+06:00)
- Asia/Almatý (UTC+06:00)
- Asia/Dacca (UTC+06:00)
- Asia/Omsk (UTC+06:00)
- Asia/Dhaka (UTC+06:00)
- India/Chagos (UTC+06:00)
- Asia/Kyzylorda (UTC+06:00)
- Asia/Bishkek (UTC+06:00)
- Antártica/Vostok (UTC+06:00)
- Asia/Urumqi (UTC+06:00)
- Asia/Timbu (UTC+06:00)
- Asia/Thimphu (UTC+06:00)
- Asia/Katmandú (UTC+05:45)
- Asia/Katmandu (UTC+05:45)
- Asia/Calcuta (UTC+05:30)
- Asia/Colombo (UTC+05:30)
- Asia/Calcutta (UTC+05:30)
- Asia/Aktau (UTC+05:00)
- Etc/GMT-5 (UTC+05:00)
- Asia/Samarcanda (UTC+05:00)
- Asia/Karachi (UTC+05:00)
- Asia/Ekaterimburgo (UTC+05:00)
- Asia/Dusambé (UTC+05:00)
- India/Maldivas (UTC+05:00)
- Asia/Oral (UTC+05:00)

- Asia/Taskent (UTC+05:00)
- Antártica/Mawson (UTC+05:00)
- Asia/Aktobé (UTC+05:00)
- Asia/Asjabad (UTC+05:00)
- Asia/Ashgabat (UTC+05:00)
- Asia/Atirau (UTC+05:00)
- India/Kerguelen (UTC+05:00)
- Irán (UTC+04:30)
- Asia/Teherán (UTC+04:30)
- Asia/Kabul (UTC+04:30)
- Asia/Ereván (UTC+04:00)
- Etc/GMT-4 (UTC+04:00)
- Etc/GMT-4 (UTC+04:00)
- Asia/Dubái (UTC+04:00)
- India/Reunión (UTC+04:00)
- Europa/Sarátov (UTC+04:00)
- Europa/Samara (UTC+04:00)
- India/Mahé (UTC+04:00)
- Asia/Bakú (UTC+04:00)
- Asia/Mascate (UTC+04:00)
- Europa/Volgogrado (UTC+04:00)
- Europa/Astracán (UTC+04:00)
- Asia/Tiflis (UTC+04:00)
- Europa/Uliánovsk (UTC+04:00)
- Asia/Adén (UTC+03:00)
- África/Nairobi (UTC+03:00)
- Europa/Estambul (UTC+03:00)
- Etc/GMT-3 (UTC+03:00)
- Europa/Zaporiyia (UTC+03:00)

- Israel (UTC+03:00)
- India/Comoro (UTC+03:00)
- Antártica/Syowa (UTC+03:00)
- África/Mogadiscio (UTC+03:00)
- Europa/Bucarest (UTC+03:00)
- África/Asmera (UTC+03:00)
- Europa/Mariehamn (UTC+03:00)
- Asia/Estambul (UTC+03:00)
- Europa/Tiráspol (UTC+03:00)
- Europa/Moscú (UTC+03:00)
- Europa/Chisináu (UTC+03:00)
- Europa/Helsinki (UTC+03:00)
- Asia/Beirut (UTC+03:00)
- Asia/Tel_Aviv (UTC+03:00)
- África/Yibuti (UTC+03:00)
- Europa/Simferópol (UTC+03:00)
- Europa/Sofía (UTC+03:00)
- Asia/Gaza (UTC+03:00)
- África/Asmara (UTC+03:00)
- Europa/Riga (UTC+03:00)
- Asia/Bagdad (UTC+03:00)
- Asia/Damasco (UTC+03:00)
- África/Dar_es_Salaam (UTC+03:00)
- África/Addis_Ababa (UTC+03:00)
- Europa/Úzhgorod (UTC+03:00)
- Asia/Jerusalén (UTC+03:00)
- Asia/Riad (UTC+03:00)
- Asia/Kuwait (UTC+03:00)
- Europa/Kirov (UTC+03:00)

- África/Kampala (UTC+03:00)
- Europa/Minsk (UTC+03:00)
- Asia/Catar (UTC+03:00)
- Europa/Kiev (UTC+03:00)
- Asia/Bahréin (UTC+03:00)
- Europa/Vilna (UTC+03:00)
- India/Antananarivo (UTC+03:00)
- India/Mayotte (UTC+03:00)
- Europa/Tallin (UTC+03:00)
- Turquía (UTC+03:00)
- África/Juba (UTC+03:00)
- Asia/Nicosia (UTC+03:00)
- Asia/Famagusta (UTC+03:00)
- W-SU (UTC+03:00)
- EET (UTC+03:00)
- Asia/Hebrón (UTC+03:00)
- Asia/Ammán (UTC+03:00)
- Europa/Nicosia (UTC+03:00)
- Europa/Atenas (UTC+03:00)
- África/EI_Cairo (UTC+02:00)
- África/Babane (UTC+02:00)
- Europa/Bruselas (UTC+02:00)
- Europa/Varsovia (UTC+02:00)
- CET (UTC+02:00)
- Europa/Luxemburgo (UTC+02:00)
- Etc/GMT-2 (UTC+02:00)
- Libia (UTC+02:00)
- África/Kigali (UTC+02:00)
- África/Trípoli (UTC+02:00)

- Europa/Kaliningrado (UTC+02:00)
- África/Windhoek (UTC+02:00)
- Europa/Malta (UTC+02:00)
- Europa/Bosingen (UTC+02:00)
-
- Europa/Skopie (UTC+02:00)
- Europa/Sarajevo (UTC+02:00)
- Europa/Roma (UTC+02:00)
- Europa/Zúrich (UTC+02:00)
- Europa/Gibraltar (UTC+02:00)
- África/Lubumbashi (UTC+02:00)
- Europa/Vaduz (UTC+02:00)
- Europa/Liubliana (UTC+02:00)
- Europa/Berlín (UTC+02:00)
- Europa/Estocolmo (UTC+02:00)
- Europa/Budapest (UTC+02:00)
- Europa/Zagreb (UTC+02:00)
- Europa/París (UTC+02:00)
- África/Ceuta (UTC+02:00)
- Europa/Praga (UTC+02:00)
- Antártica/Troll (UTC+02:00)
- África/Gaborone (UTC+02:00)
- Europa/Copenhague (UTC+02:00)
- Europa/Viena (UTC+02:00)
- Europa/Tirana (UTC+02:00)
- MET (UTC+02:00)
- Europa/Ámsterdam (UTC+02:00)
- África/Maputo (UTC+02:00)
- Europa/San_Marino (UTC+02:00)

- Polonia (UTC+02:00)
- Europa/Andorra (UTC+02:00)
- Europa/Oslo (UTC+02:00)
- Europa/Podgorica (UTC+02:00)
- África/Buyumbura (UTC+02:00)
- Atlántico/Jan_Mayen (UTC+02:00)
- África/Maseru (UTC+02:00)
- Europa/Madrid (UTC+02:00)
- África/Blantire (UTC+02:00)
- África/Lusaka (UTC+02:00)
- África/Harare (UTC+02:00)
- África/Jartum (UTC+02:00)
- África/Johannesburgo (UTC+02:00)
- Europa/Belgrado (UTC+02:00)
- Europa/Bratislava (UTC+02:00)
- Ártico/Longyearbyen (UTC+02:00)
- Egipto (UTC+02:00)
- Europa/Vaticano (UTC+02:00)
- Europa/Mónaco (UTC+02:00)
- Europa/Londres (UTC+01:00)
- Etc/GMT-1 (UTC+01:00)
- Europa/Jersey (UTC+01:00)
- Europa/Guernsey (UTC+01:00)
- Europa/Isle_of_Man (UTC+01:00)
- África/Túnez (UTC+01:00)
- África/Malabo (UTC+01:00)
- GB-Eire (UTC+01:00)
- África/Lagos (UTC+01:00)
- África/Argel (UTC+01:00)

- GB (UTC+01:00)
- Portugal (UTC+01:00)
- África/Sao_Tome (UTC+01:00)
- África/Yamena (UTC+01:00)
- Atlántico/Islas Faeroe (UTC+01:00)
- Eire (UTC+01:00)
- Atlántico/Islas Feroe (UTC+01:00)
- Europa/Dublín (UTC+01:00)
- África/Libreville (UTC+01:00)
- África/EI_Aaiún (UTC+01:00)
- África/EI_Aaiún (UTC+01:00)
- África/Duala (UTC+01:00)
- África/Brazzaville (UTC+01:00)
- África/Porto-Novo (UTC+01:00)
- Atlántico/Madeira (UTC+01:00)
- Europa/Lisboa (UTC+01:00)
- Atlántico/Canarias (UTC+01:00)
- África/Casablanca (UTC+01:00)
- Europa/Belfast (UTC+01:00)
- África/Luanda (UTC+01:00)
- África/Kinsasa (UTC+01:00)
- África/Bangui (UTC+01:00)
- WET (UTC+01:00)
- África/Niamey (UTC+01:00)
- GMT (UTC+00:00)
- Etc/GMT-0 (UTC+00:00)
- Atlántico/St_Helena (UTC+00:00)
- Etc/GMT+0 (UTC+00:00)
- África/Banjul (UTC+00:00)

- Etc/GMT (UTC+00:00)
- África/Freetown (UTC+00:00)
- África/Bamako (UTC+00:00)
- África/Conakri (UTC+00:00)
- Universal (UTC+00:00)
- África/Nuakchot (UTC+00:00)
- UTC (UTC+00:00)
- Etc/Universal (UTC+00:00)
- Atlántico/Azores (UTC+00:00)
- África/Abiyán (UTC+00:00)
- África/Acra (UTC+00:00)
- Etc/UCT (UTC+00:00)
- GMT0 (UTC+ 00:00)
- Zulu (UTC+00:00) Zulu (UTC+00:00)
- África/Uagadugú (UTC+00:00)
- Atlántico/Reikiavik (UTC+00:00)
- Etc/Zulu (UTC+00:00)
- Islandia (UTC+00:00)
- África/Lomé (UTC+00:00)
- Greenwich (UTC+00:00)
- Etc/ (UTC+ 00:00) GMT0
- América/Danmarkshavn (UTC+00:00)
- África/Dakar (UTC+00:00)
- África/Bisáu (UTC+00:00)
- Etc/Greenwich (UTC+00:00)
- África/Tombuctú (UTC+00:00)
- UTC (UTC+00:00)
- África/Monrovia (UTC+00:00)
- Etc/UTC (UTC+00:00)

Tutoriales de expresiones de fórmula

Puede seguir estos tutoriales para utilizar expresiones de fórmula en AWS IoT SiteWise.

Temas

- [Uso de cadenas en fórmulas](#)
- [Filtrado de puntos de datos](#)
- [Recuento de puntos de datos que coinciden con una condición](#)
- [Datos antiguos en las fórmulas](#)
- [Calidad de los datos en las fórmulas](#)
- [Valores indefinidos, infinitos y de desbordamiento](#)

Uso de cadenas en fórmulas

Puede operar con cadenas en sus expresiones de fórmula. También puede introducir cadenas a partir de variables que hacen referencia a propiedades de atributos y medidas.

Important

Las expresiones de fórmula solo pueden generar valores dobles o de cadena. Las expresiones anidadas pueden generar otros tipos de datos, como cadenas, pero la fórmula en su conjunto debe evaluarse como un número o una cadena. Puede usar la [función jp](#) para convertir una cadena en un número. El valor booleano debe ser 1 (verdadero) o 0 (falso). Para obtener más información, consulte [Valores indefinidos, infinitos y de desbordamiento](#).

AWS IoT SiteWise proporciona las siguientes funciones de expresión de fórmulas que puede utilizar para operar con cadenas:

- [Literales de cadena](#)
- El [operador de índice](#) (s[index])
- El [operador de sector](#) (s[start:end:step])
- [Funciones de comparación](#), que puede utilizar para comparar cadenas en [orden lexicográfico](#)
- [Funciones de cadena](#), que incluyen la función jp que puede analizar objetos JSON serializados y convertir cadenas en números

Filtrado de puntos de datos

Puede utilizar la [función if](#) para filtrar los puntos de datos que no cumplan una condición. La función `if` evalúa una condición y devuelve valores diferentes para los resultados `true` y `false`. Puede utilizar la [constante none](#) como resultado para un caso de una función `if`, para descartar el punto de datos de ese caso.

Para filtrar los puntos de datos que coinciden con una condición

- Cree una transformación que utilice la función `if` para definir una condición que compruebe si se cumple una condición y que devuelva `none` como el valor `result_if_true` o `result_if_false`.

Example Ejemplo: filtrar los puntos de datos en los que el agua no esté hirviendo

Considere un escenario en el que usted tiene una medición, `temp_c`, que proporciona la temperatura (en Celsius) del agua de una máquina. Puede definir la siguiente transformación para filtrar los puntos de datos en los que el agua no esté hirviendo:

- Transformación `boiling_temps = if(gte(temp_c, 100), temp_c, none)`: devuelve la temperatura si es mayor o igual a 100 grados Celsius; de lo contrario, no devuelve ningún punto de datos.

Recuento de puntos de datos que coinciden con una condición

Puede utilizar [funciones de comparación](#) y [sum\(\)](#) para contar el número de puntos de datos para los que se cumple una condición.

Para contar los puntos de datos que coinciden con una condición

1. Cree una transformación que utilice una función de comparación para definir una condición de filtro en otra propiedad.
2. Cree una métrica que sume los puntos de datos donde se cumple esa condición.

Example Ejemplo: Contar el número de puntos de datos en los que el agua está hirviendo

Considere un escenario en el que usted tiene una medición, `temp_c`, que proporciona la temperatura (en Celsius) del agua de una máquina. Puede definir las siguientes propiedades de transformación y métrica para contar el número de puntos de datos en los que hierve el agua:

- Transformación `is_boiling = gte(temp_c, 100)`: devuelve 1 si la temperatura es mayor o igual a 100 grados Celsius; de lo contrario, devuelve 0.
- Métrica `boiling_count = sum(is_boiling)`: devuelve el número de puntos de datos en los que el agua está hirviendo.

Datos antiguos en las fórmulas

AWS IoT SiteWise admite la ingesta tardía de datos con una antigüedad de hasta 7 días. Cuando AWS IoT SiteWise recibe datos atrasados, recalcula los valores existentes para cualquier métrica que introduzca los datos atrasados en una ventana anterior. Estos nuevos cálculos dan lugar a cargos de procesamiento de datos.

Note

Cuando AWS IoT SiteWise calcula las propiedades que introducen datos atrasados, utiliza la expresión de fórmula actual de cada propiedad.

Después de AWS IoT SiteWise volver a calcular una ventana anterior para una métrica, reemplaza el valor anterior de esa ventana. Si ha activado las notificaciones para esa métrica, AWS IoT SiteWise también emite una notificación del valor de la propiedad. Esto significa que puede recibir una nueva notificación de actualización de valor de propiedad para la misma propiedad y marca temporal para la que recibió previamente una notificación. Si las aplicaciones o lagos de datos consumen notificaciones de valor de propiedad, debe actualizar el valor anterior con el nuevo valor para que sus datos sean precisos.

Calidad de los datos en las fórmulas

En AWS IoT SiteWise, cada punto de datos tiene un código de calidad, que puede ser uno de los siguientes:

- GOOD: los datos no se ven afectados por ningún problema.
- BAD: los datos se ven afectados por un problema, como un fallo del sensor.
- UNCERTAIN: los datos se ven afectados por un problema, como la falta de precisión de un sensor.

AWS IoT SiteWise consume solo datos GOOD de calidad cuando calcula las transformaciones y las métricas. AWS IoT SiteWise genera solo datos GOOD de calidad para que los cálculos se realicen

correctamente. Si un cálculo no tiene éxito, AWS IoT SiteWise no genera un punto de datos para ese cálculo. Esto puede ocurrir si un cálculo da como resultado un valor indefinido, infinito o de desbordamiento.

Para obtener más información acerca de cómo consultar datos y filtrar por calidad de datos, consulte [Consulta datos de AWS IoT SiteWise](#).

Valores indefinidos, infinitos y de desbordamiento

Algunas expresiones de fórmula (como $x / 0$, $\text{sqrt}(-1)$, o $\log(0)$) calculan valores indefinidos en un sistema de números reales, infinitos o que están fuera del rango admitido por él. AWS IoT SiteWise Cuando la expresión de una propiedad de un activo calcula un valor indefinido, infinito o desbordante, AWS IoT SiteWise no genera ningún punto de datos para ese cálculo.

AWS IoT SiteWise tampoco genera un punto de datos si calcula un valor no numérico como resultado de una expresión de fórmula. Esto significa que si se define una fórmula que calcula una cadena, una matriz o la [constante none](#), AWS IoT SiteWise no generará ningún punto de datos para ese cálculo.

Example Ejemplos

Cada una de las siguientes expresiones de fórmula da como resultado un valor que no AWS IoT SiteWise se puede representar como un número. AWS IoT SiteWise no genera un punto de datos cuando calcula estas expresiones de fórmula.

- $x / 0$ es indefinido.
- $\log(0)$ es indefinido.
- $\text{sqrt}(-1)$ es indefinido en un sistema numérico real.
- "hello" + " world" es una cadena.
- `jp({'values':[3,6,7]}, '$.values')` es una matriz.
- `if(gte(temp, 300), temp, none)` es none cuando temp es menos de 300.

Creación de modelos compuestos personalizados (componentes)

Los modelos compuestos personalizados, o los componentes si utiliza la consola, proporcionan otro nivel de organización para sus modelos de activos y modelos de componentes. Puede utilizarlos para estructurar sus modelos agrupando propiedades o haciendo referencia a otros modelos. Para

obtener más información acerca de cómo trabajar con modelos compuestos personalizados, consulte [Modelos compuestos personalizados \(componentes\)](#).

Puede crear un modelo compuesto personalizado dentro de un modelo de activos o modelo de componentes existente. Hay dos tipos de modelos compuestos personalizados. Para agrupar propiedades relacionadas dentro de un modelo, puede crear un modelo compuesto personalizado en línea. Para hacer referencia a un modelo de componentes dentro de su modelo de activos o modelo de componentes, puede crear un modelo compuesto component-model-based personalizado.

En las siguientes secciones se describe cómo utilizar la AWS IoT SiteWise API para crear modelos compuestos personalizados.

Temas

- [Creación de un componente en línea \(consola\)](#)
- [Creación de un modelo compuesto personalizado en línea \(AWS CLI\)](#)
- [Cree un component-model-based componente \(consola\)](#)
- [Crear un modelo compuesto component-model-based personalizado \(AWS CLI\)](#)

Creación de un componente en línea (consola)

Puede usar la AWS IoT SiteWise consola para crear un componente en línea que defina sus propias propiedades.

Note

Dado que se trata de un componente en línea, estas propiedades solo se aplican al modelo de activos actual y no se comparten en ningún otro lugar.

Si necesita crear un modelo reutilizable (por ejemplo, para compartirlo entre varios modelos de activos o para incluir varias instancias en un modelo de activos), debe crear un componente basado en un modelo de componentes. Consulte la siguiente sección para obtener más información.

Creación de un componente (consola)

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, elija Models (Modelos).

3. Elija el modelo de activos al que desea añadir un componente.
4. En la pestaña Propiedades, elija Componentes.
5. Seleccione Crear componente.
6. En la página Crear componente, haga lo siguiente:
 - a. Escriba un Nombre para el componente, por ejemplo, **ServoMotor** o **ServoMotor Model**. Este nombre debe ser único en todos los componentes de su cuenta en esta región.
 - b. (Opcional) Agregue Definiciones de atributos para el modelo. Los atributos representan información que rara vez cambia. Para obtener más información, consulte [Definición de datos estáticos \(atributos\)](#).
 - c. (Opcional) Agregue Definiciones de mediciones para el modelo. Las mediciones representan flujos de datos de su equipo. Para obtener más información, consulte [Definición de flujos de datos procedentes del equipo \(mediciones\)](#).
 - d. (Opcional) Agregue Definiciones de transformación para el modelo. Las transformaciones son fórmulas que asignan datos de un formulario a otro. Para obtener más información, consulte [Datos de transformación \(transformaciones\)](#).
 - e. (Opcional) Agregue Definiciones de métricas para el modelo. Las métricas son fórmulas que agregan datos a lo largo de intervalos de tiempo. Las métricas pueden agregar datos de entrada de activos asociados, de modo que puede calcular valores que representan la operación o un subconjunto de la operación. Para obtener más información, consulte [Agregación de datos desde propiedades y otros activos \(métricas\)](#).
 - f. Seleccione Crear componente.

Creación de un modelo compuesto personalizado en línea (AWS CLI)

Puede usar el AWS Command Line Interface (AWS CLI) para crear un modelo compuesto personalizado en línea que defina sus propias propiedades.

Utilice la [CreateAssetModelCompositeModel](#) operación para crear un modelo en línea con propiedades. Esta operación espera una carga con la siguiente estructura.

Note

Dado que se trata de un modelo compuesto en línea, estas propiedades solo se aplican al modelo de activos actual y no se comparten en ningún otro lugar. Lo que lo convierte en “integrado” es que no proporciona un valor para el campo `composedAssetModelId`.

Si necesita crear un modelo reutilizable (por ejemplo, para compartirlo entre varios modelos de activos o para incluir varias instancias en un modelo de activos), debe crear un modelo component-model-based compuesto en su lugar. Consulte la siguiente sección para obtener más información.

```
{
  "assetModelCompositeModelName": "CNCLathe_ServoMotorA",
  "assetModelCompositeModelType": "CUSTOM",
  "assetModelCompositeModelProperties": [
    {
      "dataType": "DOUBLE",
      "name": "Servo Motor Temperature",
      "type": {
        "measurement": {}
      },
      "unit": "Celsius"
    },
    {
      "dataType": "DOUBLE",
      "name": "Spindle speed",
      "type": {
        "measurement": {}
      },
      "unit": "rpm"
    }
  ]
}
```

Cree un component-model-based componente (consola)

Puede utilizar la AWS IoT SiteWise consola para crear un componente a partir de un modelo de componentes.

Para crear un component-model-based componente (consola)

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, elija Models (Modelos).
3. Elija el modelo de activos al que desea añadir un componente.
4. En la pestaña Propiedades, elija Componentes.

5. Seleccione Crear componente.
6. En la página Crear componente, haga lo siguiente:
 - a. Seleccione el modelo de componentes en el que desee basar el componente.
 - b. Escriba un Nombre para el componente, por ejemplo, **ServoMotor** o **ServoMotor Model**. Este nombre debe ser único en todos los componentes de su cuenta en esta región.
 - c. Seleccione Crear componente.

Crear un modelo compuesto component-model-based personalizado (AWS CLI)

Puede usarlo AWS CLI para crear un modelo compuesto component-model-based personalizado dentro de su modelo de activos. Un modelo compuesto component-model-based personalizado es una referencia a un modelo de componentes que ya ha definido en otro lugar.

Utilice la [CreateAssetModelCompositeModel](#) operación para crear un modelo compuesto component-model-based personalizado. Esta operación espera una carga con la siguiente estructura.

Note

En este ejemplo, el valor de `composedAssetModelId` es el ID del modelo de activos o el ID externo de un modelo de componentes existente. Para obtener más información, consulte [Objetos de referencia con objetos externos IDs](#) en la Guía del usuario de AWS IoT SiteWise. Para ver un ejemplo de cómo crear un modelo de componentes, consulte [Creación de un modelo de componentes \(AWS CLI\)](#).

```
{
  "assetModelCompositeModelName": "CNCLathe_ServoMotorA",
  "assetModelCompositeModelType": "CUSTOM",
  "composedAssetModelId": component model ID
}
```

Como es solo una referencia, un modelo compuesto component-model-based personalizado no tiene propiedades propias, salvo un nombre.

Si desea añadir varias instancias del mismo componente a su modelo de activos (por ejemplo, una máquina CNC con varios servomotores), puede añadir varios modelos compuestos component-

model-based personalizados, cada uno con su propio nombre, pero que hagan referencia al mismo `composedAssetModelId` nombre.

Puede anidar componentes dentro de otros componentes. Para ello, puede añadir un modelo `component-model-based` compuesto, como se muestra en este ejemplo, a uno de sus modelos de componentes.

Cree activos para modelos de activos en AWS IoT SiteWise

Puede crear un activo a partir de un modelo de activos. Debe tener un modelo de activos para poder crear un activo. Si no ha creado un modelo de activos, consulte [Cree modelos de activos en AWS IoT SiteWise](#).

Note

Solo se pueden crear activos a partir de modelos ACTIVE. Si el estado del modelo no es ACTIVE, es posible que tenga que esperar hasta unos minutos para poder crear activos a partir de ese modelo. Para obtener más información, consulte [Estados de activos y modelos](#).

Temas

- [Creación de un activo \(consola\)](#)
- [Creación de un activo \(AWS CLI\)](#)
- [Configuración de un nuevo activo](#)

Creación de un activo (consola)

Puede utilizar la AWS IoT SiteWise consola para crear un activo.

Para crear un activo (consola)

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, elija Activos.
3. Elija Create asset (Crear activo).
4. En la página Crear activo, haga lo siguiente:
 - a. En Modelo, elija el modelo de activos desde el que desea crear un activo.

Note

Si el modelo no está **ACTIVO**, debe esperar hasta que se active o resolver los problemas si está en **ERROR**.

- b. Escriba un Nombre para el activo.
- c. (Opcional) Agregue etiquetas para su activo. Para obtener más información, consulte [Etiquete sus AWS IoT SiteWise recursos](#).
- d. Elija Create asset (Crear activo).

Al crear un activo, la AWS IoT SiteWise consola navega hasta la página del nuevo activo. En esta página, puede consultar el Estado del activo, que inicialmente es **CREANDO**. Esta página se actualiza automáticamente, por lo que puede esperar a que se actualice el estado del activo.

Note

El proceso de creación de activos puede tardar hasta un minuto. Después de que el Estado esté **ACTIVO**, puede realizar operaciones de actualización en el activo. Para obtener más información, consulte [Estados de activos y modelos](#).

Después de crear un activo, consulte [Configuración de un nuevo activo](#).

Creación de un activo (AWS CLI)

Puedes usar AWS Command Line Interface (AWS CLI) para crear un activo a partir de un modelo de activos.

Debe tener un `assetModelId` para crear un activo. Si has creado un modelo de activos, pero no lo conoces `assetModelId`, usa la [ListAssetModelsAPI](#) para ver todos tus modelos de activos.

Para crear un activo a partir de un modelo de activos, usa la [CreateAssetAPI](#) con los siguientes parámetros:

- `assetName`: el nombre del nuevo activo. Asigne un nombre a su activo que le ayude a identificarlo.

- `assetModelId`: el ID del activo. Es el ID real en formato UUID o el `externalId:myExternalId` si lo tiene. Para obtener más información, consulte [Objetos de referencia con objetos externos IDs](#) en la Guía del usuario de AWS IoT SiteWise .

Creación de un activo (AWS CLI)

- Ejecute el siguiente comando para crear un activo. `asset-name` Sustitúyalo por un nombre para el activo y `asset-model-id` por el ID o el ID externo del modelo de activo.

```
aws iotsitewise create-asset \  
  --asset-name asset-name \  
  --asset-model-id asset-model-id
```

La operación devuelve una respuesta que contiene los detalles y el estado del nuevo activo en el siguiente formato.

```
{  
  "assetId": "String",  
  "assetArn": "String",  
  "assetStatus": {  
    "state": "String",  
    "error": {  
      "code": "String",  
      "message": "String"  
    }  
  }  
}
```

El state del activo es CREATING hasta que el activo crea.

Note

El proceso de creación de activos puede tardar hasta un minuto. Para comprobar el estado del activo, utilice la [DescribeAsset](#) operación con el ID del activo como `assetId` parámetro. Cuando el state del activo sea ACTIVE, puede realizar operaciones de actualización en su activo. Para obtener más información, consulte [Estados de activos y modelos](#).

Después de crear un activo, consulte [Configuración de un nuevo activo](#).

Configuración de un nuevo activo

Después de crear un activo AWS IoT SiteWise, hay varios pasos siguientes que puede seguir para utilizar al máximo el activo y sus datos. Estos pasos pueden incluir la configuración de los flujos de datos para incorporar los datos del activo, la configuración de alarmas y notificaciones para supervisar el rendimiento del activo, la creación de visualizaciones y paneles para mostrar los datos del activo y la integración del activo con otros AWS servicios o aplicaciones de terceros para su posterior análisis o automatización.

Termine de configurar el activo con las siguientes acciones opcionales:

- [Gestione los flujos de datos para AWS IoT SiteWise](#) si su activo tiene propiedades de medición.
- [Actualización de valores de atributos](#) si su activo tiene valores de atributo únicos.
- [Asociación y desasociación de activos](#) si su activo es un activo principal.

Busque activos en Consola de AWS IoT SiteWise

Utilice la función Consola de AWS IoT SiteWise de búsqueda para encontrar activos en función de los metadatos y los filtros del valor de las propiedades en tiempo real.

Requisitos previos

AWS IoT SiteWise requiere permisos para integrarse AWS IoT TwinMaker a fin de organizar y modelar mejor los datos industriales. Si ha concedido permisos para ello AWS IoT SiteWise, utilice la [ExecuteQuery](#) API. Si no ha concedido permisos y necesita ayuda para AWS IoT SiteWise empezar, consulte [Integrar AWS IoT SiteWise y AWS IoT TwinMaker](#).

Búsqueda avanzada en Consola de AWS IoT SiteWise

Búsqueda de metadatos

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, seleccione Búsqueda avanzada en Activos.
3. En Búsqueda avanzada, elija la opción Búsqueda de metadatos.

4. Rellene los parámetros. Rellene tantos campos como sea posible para realizar una búsqueda eficiente.
 - a. Nombre del activo: introduzca un nombre completo del activo o un nombre parcial para realizar una búsqueda más amplia.
 - b. Nombre de la propiedad: introduzca un nombre completo de la propiedad o un nombre parcial para realizar una búsqueda más amplia.
 - c. Operador: elija un operador entre:
 - =
 - <
 - >
 - <=
 - >=
 - d. Valor de la propiedad: este valor se compara con el último valor de la propiedad.
 - e. Tipo de valor de la propiedad: tipo de datos de la propiedad. Elija una de las siguientes opciones:
 - Doble
 - Entero
 - Cadena
 - Booleano
5. Elija Buscar.
6. En la tabla Resultados de la búsqueda, elija el activo en la columna Nombre. Esto le lleva a la página de detalles de ese activo.

Assets

Assets represent Industrial devices and processes that send data streams to SiteWise. Models are structures that enforce a specific model of properties and hierarchies for all instances of each asset. You must create every asset from a model.

Advanced search

Use advanced search to find assets based on specific metadata. In addition, you can enter SQL queries directly in the query builder.

Metadata search | Query builder

Asset name: Level-2 | Property name: power_max | Operator: > | Property value: 20 | Property value type: Double

Search results (2)

Name	Asset id	Description
Level-2-asset-1	d0e9019b-9c38-4316-b574-38317aa38143	
Level-2-asset-2	b9c0d2fc-1527-42ce-8ba2-d1a4e8ff43de	Example description

Búsqueda parcial

No es necesario proporcionar todos los parámetros para una búsqueda de activos. A continuación, se muestran algunos ejemplos de búsquedas parciales que utilizan la opción de búsqueda de metadatos:

- Busque activos por su nombre:
 - Introduzca un valor solo en el campo Nombre del activo.
 - Los campos Nombre de la propiedad y Valor de la propiedad están vacíos.
- Busque activos que contengan propiedades con un nombre específico:
 - Introduzca un valor solo en el campo Nombre del activo.
 - Los campos Nombre del activo y Valor de la propiedad están vacíos.
- Busque activos en función de los valores más recientes de sus propiedades:
 - Introduzca valores en los campos Nombre de la propiedad y Valor de la propiedad.
 - Seleccione un Operador y un Tipo de valor de la propiedad.

Búsqueda en el generador de consultas

1. Vaya a Consola de AWS IoT SiteWise.
2. En el panel de navegación, seleccione Búsqueda avanzada en Activos.
3. En Búsqueda avanzada, elija la opción Generador de consultas.
4. En el panel Generador de consultas, escriba su consulta de SQL para recuperar `asset_name`, `asset_id` y `asset_description`.
5. Elija Buscar.
6. En la tabla Resultados de la búsqueda, elija el activo en la columna Nombre. Esto le lleva a la página de detalles de ese activo.

Assets Refresh Create asset

Assets represent Industrial devices and processes that send data streams to SiteWise. Models are structures that enforce a specific model of properties and hierarchies for all instances of each asset. You must create every asset from a model.

Advanced search
Use advanced search to find assets based on specific metadata. In addition, you can enter SQL queries directly in the query builder.

Metadata search **Query builder**

Query builder

```
SELECT a.asset_id, a.asset_name, a.asset_description
FROM asset a, asset_property p, latest_value_time_series ts
WHERE a.asset_name LIKE '%asset-2%' AND a.property_name = 'temperature_f' AND ts.double_value > 50.0
```

Clear Search

Search results (2) < 1 >

Name	Asset id	Description
Level-2a-asset-2	4fed596d-e903-4338-86db-34ca9301233a	Generator #3
Level-2b-asset-2	b4ac2b24-4fce-4a72-9fea-ef6d0f741e8d	Generator #2

Note

- La cláusula SELECT de la consulta de SQL debe incluir los campos `asset_id` y `asset_name` para garantizar un activo válido en la tabla Resultados de la búsqueda.

- El Generador de consultas solo muestra el Nombre, el ID del activo y la Descripción en la tabla de resultados. Aunque añada más campos a la cláusula SELECT, no se añadirán más columnas a la tabla de resultados

Actualización de valores de atributos

Los activos heredan los atributos de su modelo de activos, incluido el valor predeterminado del atributo. En algunos casos, querrá conservar el atributo predeterminado del modelo de activos, por ejemplo para una propiedad de fabricante de activos. En otros casos, querrá actualizar el atributo heredado, como para la latitud y la longitud de un activo.

Updating an attribute value (console)

Puede utilizar la AWS IoT SiteWise consola para actualizar el valor de una propiedad de activo de atributo.

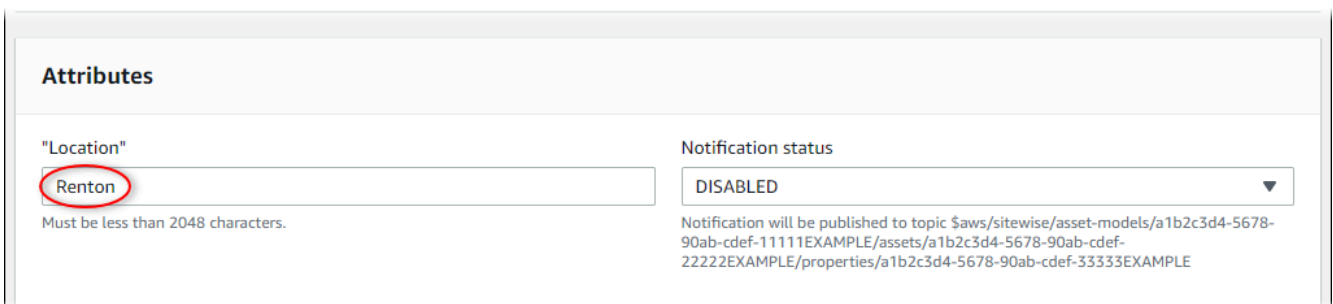
Para actualizar el valor de un atributo (consola)

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, elija Activos.
3. Elija el activo para el que desea actualizar un atributo.

Tip

Puede elegir el icono de flecha para expandir una jerarquía de activos y encontrar su activo.

4. Elija Edit (Edición de).
5. Encuentre el atributo que desea actualizar y, a continuación, escriba su nuevo valor.



Attributes

"Location" Notification status: DISABLED

Must be less than 2048 characters.

Notification will be published to topic \$aws/siteswise/asset-models/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE/assets/a1b2c3d4-5678-90ab-cdef-22222EXAMPLE/properties/a1b2c3d4-5678-90ab-cdef-33333EXAMPLE

6. Seleccione Save.

Updating an attribute value (AWS CLI)

Puedes usar AWS Command Line Interface (AWS CLI) para actualizar el valor de un atributo.

Debe conocer los `assetId` de sus activos y los `propertyId` de las propiedades para completar este procedimiento. También puede utilizar el ID externo. Si has creado un activo y no lo sabes `assetId`, usa la [ListAssets](#) API para enumerar todos los activos de un modelo específico. Utilice la [DescribeAsset](#) operación para ver las propiedades de su activo, incluida la propiedad IDs.

Utilice la operación [BatchPutAssetPropertyValue](#) para asignar valores de atributo a su activo. Puede utilizar esta operación para establecer varios atributos a la vez. La carga de esta operación contiene una lista de entradas y cada una contiene el ID de activo, el ID de propiedad y el valor de atributo.

Actualización del valor de un atributo (AWS CLI)

1. Cree un archivo llamado `batch-put-payload.json` y copie el siguiente objeto JSON en el archivo. En esta carga de ejemplo se muestra cómo establecer la latitud y la longitud de una turbina eólica. Actualice los IDs valores y las marcas de tiempo para modificar la carga útil según su caso de uso.

```
{
  "entries": [
    {
      "entryId": "windfarm3-turbine7-latitude",
      "assetId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
      "propertyId": "a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
      "propertyValues": [
        {
          "value": {
            "doubleValue": 47.6204
          },
          "timestamp": {
            "timeInSeconds": 1575691200
          }
        }
      ]
    },
    {
      "entryId": "windfarm3-turbine7-longitude",
      "assetId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
```

```
"propertyId": "a1b2c3d4-5678-90ab-cdef-55555EXAMPLE",
"propertyValues": [
  {
    "value": {
      "doubleValue": 122.3491
    },
    "timestamp": {
      "timeInSeconds": 1575691200
    }
  }
]
}
```

- Cada entrada de la carga contiene un `entryId` que puede definir como una única cadena. Si la entrada de la solicitud no se realiza correctamente, cada error contendrá el `entryId` de la solicitud correspondiente para que sepa qué solicitudes deben volver a intentarse.
- Para establecer un valor de atributo, puede incluir una estructura `timestamp-quality-value` (TQV) en la lista de propiedades de `propertyValues` cada atributo. Esta estructura debe contener el nuevo `value` y la `timestamp` actual.
 - `value`: una estructura que contiene uno de los siguientes campos, en función del tipo de propiedad que se establezca:
 - `booleanValue`
 - `doubleValue`
 - `integerValue`
 - `stringValue`
 - `nullValue`
 - `timestamp`— Una estructura que contiene el tiempo de época actual de Unix en segundos. `timeInSeconds` AWS IoT SiteWise rechaza todos los puntos de datos con marcas de tiempo que hayan existido durante más de 7 días o más de 5 minutos en el futuro.

Para obtener más información acerca de cómo preparar una carga para [BatchPutAssetPropertyValue](#), consulte [Ingiera datos con AWS IoT SiteWise APIs](#).

2. Ejecute el siguiente comando para enviar los valores del atributo a AWS IoT SiteWise:

```
aws iotsitewise batch-put-asset-property-value -\-cli-input-json file://batch-put-payload.json
```

Asociación y desasociación de activos

Si el modelo de activos define jerarquías de modelos de activos secundarios, puede asociar activos secundarios al activo. Los activos principales pueden acceder a los datos de los activos asociados y agregarlos. Para obtener más información acerca de los modelos de activos jerárquicos, consulte [Definición de jerarquías de modelos de activos](#). Si utiliza interfaces, las jerarquías definidas en la interfaz se aplican a los modelos de activos que implementan la interfaz. Para obtener más información sobre las interfaces, consulte [Interfaces del modelo de activos](#).

Temas

- [Asociación y desasociación de activos \(consola\)](#)
- [Asociación y desasociación de activos \(AWS CLI\)](#)

Asociación y desasociación de activos (consola)

Puede utilizar la AWS IoT SiteWise consola para asociar y desasociar activos.

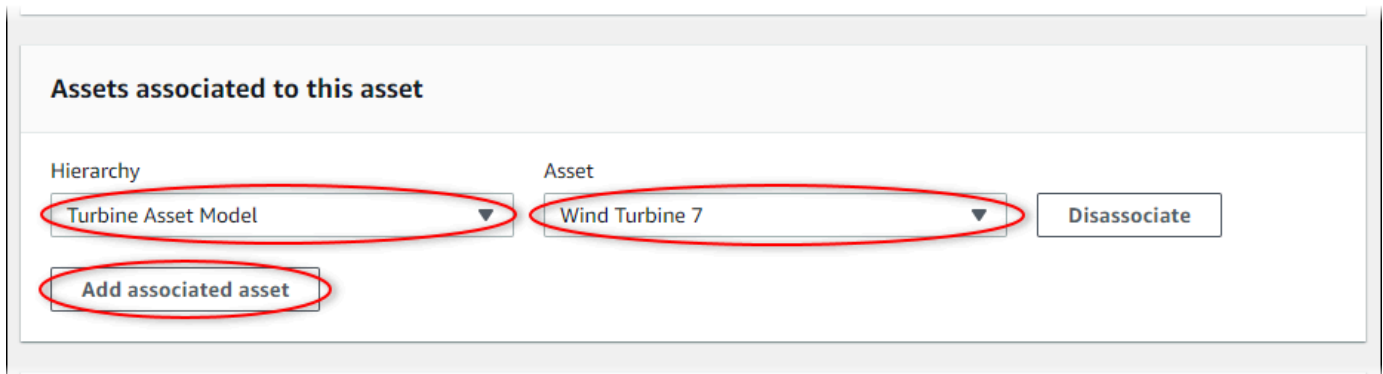
Para asociar un activo (consola)

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, elija Activos.
3. Elija el activo principal con el que desea asociar un activo secundario.

Tip

Puede elegir el icono de flecha para expandir una jerarquía de activos y encontrar su activo.

4. Elija Edit (Edición de).
5. En Activos asociados a este activo, elija Agregar un activo asociado.



Assets associated to this asset

Hierarchy: Turbine Asset Model ▼ Asset: Wind Turbine 7 ▼ Disassociate

Add associated asset

6. En Jerarquía, elija la jerarquía que defina la relación entre el activo principal y el activo secundario.
7. En Activo, elija el activo secundario que desea asociar.
8. Seleccione Save.

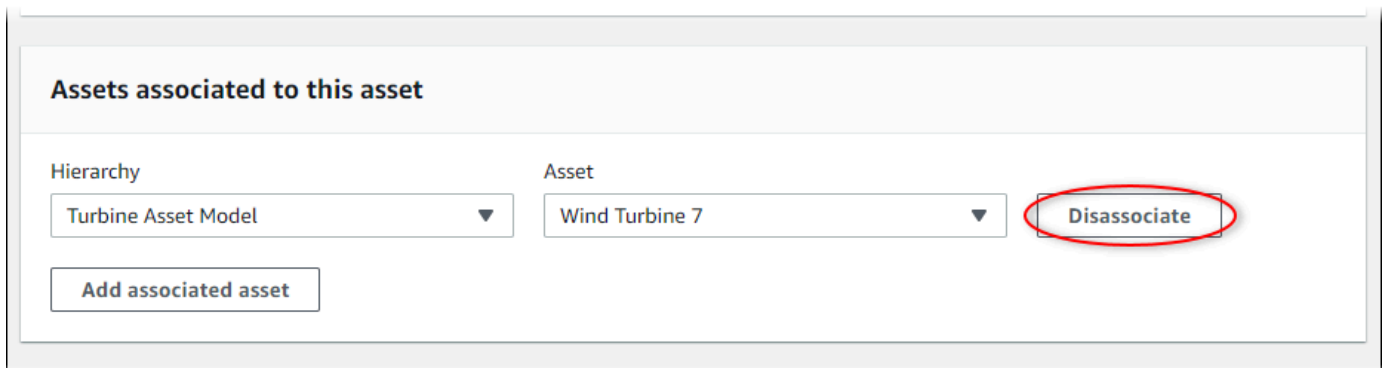
Para disociar un activo (consola)

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, elija Activos.
3. Elija el activo principal para el que desea disociar un activo secundario.

i Tip

Puede elegir el icono de flecha para expandir una jerarquía de activos y encontrar su activo.

4. Elija Edit (Edición de).
5. En Activos asociados a este activo, elija Desvincular para el activo.



Assets associated to this asset

Hierarchy: Turbine Asset Model ▼ Asset: Wind Turbine 7 ▼ Disassociate

Add associated asset

6. Seleccione Save.

Asociación y desasociación de activos (AWS CLI)

Puede usar AWS Command Line Interface (AWS CLI) para asociar y desasociar activos.

Para este procedimiento, debe conocer el ID de la jerarquía (`hierarchyId`) en el modelo de activos principal que define la relación con el modelo de activos secundario. Utilice la [DescribeAsset](#) operación para buscar el identificador de jerarquía en la respuesta.

Para encontrar un ID de jerarquía

- Ejecute el siguiente comando para describir el activo principal. `parent-asset-id` Sustitúyalo por el identificador del activo principal o el identificador externo.

```
aws iotsitewise describe-asset --asset-id parent-asset-id
```

La operación devuelve una respuesta que contiene los detalles del activo. La respuesta contiene una lista `assetHierarchies` que tiene la siguiente estructura:

```
{
  ...
  "assetHierarchies": [
    {
      "id": "String",
      "name": "String"
    }
  ],
  ...
}
```

El ID de jerarquía es el valor `id` para una jerarquía en la lista de jerarquías de activos.

Después de tener el ID de jerarquía, puede asociar o disociar un activo con esa jerarquía.

Para asociar un activo secundario a un activo principal, utilice la [AssociateAssets](#) operación. Para desasociar un activo secundario de un activo principal, utilice la [DisassociateAssets](#) operación. Especifique los siguientes parámetros, que son los mismos para ambas operaciones:

- `assetId`: ID del activo principal o ID externo.
- `hierarchyId`: ID de jerarquía o ID externo en el activo principal.

- `childAssetId`: ID del activo secundario o ID externo.

Asociación de un activo (AWS CLI)

- Ejecute el siguiente comando para asociar un activo secundario con un activo principal. Sustituya *parent-asset-id* *hierarchy-id*, y *child-asset-id* por lo respectivo IDs:

```
aws iotsitewise associate-assets \  
  --asset-id parent-asset-id \  
  --hierarchy-id hierarchy-id \  
  --child-asset-id child-asset-id
```

Disociación de un activo (AWS CLI)

- Ejecute el siguiente comando para disociar un activo secundario de un activo principal. Sustitúyase *parent-asset-id* *hierarchy-id*, y *child-asset-id* por lo respectivo IDs:

```
aws iotsitewise disassociate-assets \  
  --asset-id parent-asset-id \  
  --hierarchy-id hierarchy-id \  
  --child-asset-id child-asset-id
```

Actualización de activos y modelos

Puede actualizar sus activos, modelos de activos, modelos de componentes e interfaces AWS IoT SiteWise para modificar sus nombres y definiciones. Estas operaciones de actualización son asíncronas y tardan un tiempo en propagarse. AWS IoT SiteWise Compruebe el estado del activo o del modelo antes de realizar cambios adicionales. Debe esperar hasta que se propaguen los cambios para poder seguir utilizando el activo o modelo actualizado.

Temas

- [Actualice los activos en AWS IoT SiteWise](#)
- [Actualice los modelos de activos, los modelos de componentes y las interfaces](#)
- [Actualización de modelos compuestos personalizados \(componentes\)](#)
- [Bloqueo positivo de las escrituras del modelo de activos](#)

Actualice los activos en AWS IoT SiteWise

Puede usar la AWS IoT SiteWise consola o la API para actualizar el nombre de un activo.

Cuando se actualiza un activo, el estado del activo es UPDATING hasta que se propagan los cambios. Para obtener más información, consulte [Estados de activos y modelos](#).

Temas

- [Actualización de un activo \(consola\)](#)
- [Actualización de un activo \(AWS CLI\)](#)

Actualización de un activo (consola)

Puedes usar la AWS IoT SiteWise consola para actualizar los detalles del activo.

Para actualizar un activo (consola)

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, elija Activos.
3. Elija el activo que desea actualizar.

Tip

Puede elegir el icono de flecha para expandir una jerarquía de activos y encontrar su activo.

4. Elija Edit (Edición de).
5. Actualice el Nombre del activo.
6. (Opcional) En esta página, actualice otra información para el activo. Para obtener más información, consulte los siguientes temas:
 - [Gestione los flujos de datos para AWS IoT SiteWise](#)
 - [Actualización de valores de atributos](#)
 - [Interactúa con otros AWS servicios](#)
7. Seleccione Save.

Actualización de un activo (AWS CLI)

Puedes usar AWS Command Line Interface (AWS CLI) para actualizar el nombre de un activo.

Utilice la [UpdateAsset](#) operación para actualizar un activo. Especifique los siguientes parámetros:

- `assetId`: el ID del activo. Es el ID real en formato UUID o el `externalId:myExternalId` si lo tiene. Para obtener más información, consulte [Objetos de referencia con objetos externos IDs](#) en la Guía del usuario de AWS IoT SiteWise .
- `assetName`: el nuevo nombre del activo.

Actualización del nombre de un activo (AWS CLI)

- Ejecute el siguiente comando para actualizar el nombre de un activo. `asset-id` Sustitúyalo por el ID o el ID externo del activo. `asset-name` Actualízalo con el nuevo nombre del activo.

```
aws iotsitewise update-asset \  
  --asset-id asset-id \  
  --asset-name asset-name
```

Actualice los modelos de activos, los modelos de componentes y las interfaces

Utilice la AWS IoT SiteWise consola o la API para actualizar un modelo de activos, un modelo de componentes o una interfaz.

No se puede cambiar el tipo o el tipo de datos de una propiedad existente ni la ventana de una métrica existente. Tampoco puede cambiar el tipo de modelo, de modelo de activos a modelo de componentes o interfaz, o al revés.

Important

- Si elimina una propiedad de un modelo de activos o de un modelo de componentes, AWS IoT SiteWise elimina todos los datos anteriores de esa propiedad. En el caso de los modelos de componentes, esto afecta a todos los modelos de activos que utilizan ese modelo de componentes, por lo que hay que tener especial cuidado de entender hasta qué punto se puede aplicar el cambio.

- Si elimina una definición de jerarquía de un modelo de activos, AWS IoT SiteWise disocia todos los activos de esa jerarquía.

Al actualizar un modelo de activos, todos los activos basados en ese modelo reflejan los cambios que realice en el modelo subyacente. Hasta que los cambios se propaguen, cada activo tiene el estado UPDATING. Debe esperar hasta que esos activos vuelvan al estado ACTIVE antes de interactuar con ellos. Durante este tiempo, el estado del modelo de activos actualizado será PROPAGATING.

Al actualizar un modelo de componentes, todos los modelos de activos que incorporan ese modelo de componentes reflejan los cambios. Hasta que los cambios en el modelo de componentes se propaguen, cada modelo de activos afectado tiene el estado UPDATING, seguido de PROPAGATING cuando actualiza sus activos asociados, tal como se describe en el párrafo anterior. Debe esperar hasta que esos modelos de activos vuelvan al estado ACTIVE antes de interactuar con ellos. Durante este tiempo, el estado del modelo de componentes actualizado será PROPAGATING.

Para obtener más información, consulte [Estados de activos y modelos](#).

Temas

- [Actualización de un modelo de activos, un modelo de componentes o una interfaz \(consola\)](#)
- [Actualice un modelo de activo, un modelo de componentes o una interfaz \(AWS CLI\)](#)

Actualización de un modelo de activos, un modelo de componentes o una interfaz (consola)


Puede utilizar la AWS IoT SiteWise consola para actualizar un modelo de activos, un modelo de componentes o una interfaz.

Para actualizar un modelo de activos, un modelo de componentes o una interfaz (consola)

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, elija Models (Modelos).
3. Elija el modelo de activo, el modelo de componentes o la interfaz que desee actualizar.
4. Elija Edit (Edición de).
5. En la página Editar el modelo, realice alguna de las siguientes acciones:

- En Detalles del modelo, cambie el Nombre del modelo.
- Cambie cualquiera de las Definiciones de atributos. No se puede cambiar el Tipo de datos de los atributos existentes. Para obtener más información, consulte [Definición de datos estáticos \(atributos\)](#).
- Cambie cualquiera de las Definiciones de mediciones. No se puede cambiar el Tipo de datos de las mediciones existentes. Para obtener más información, consulte [Definición de flujos de datos procedentes del equipo \(mediciones\)](#).
- Cambie cualquiera de las Definiciones de transformación. Para obtener más información, consulte [Datos de transformación \(transformaciones\)](#).
- Cambie cualquiera de las Definiciones de métricas. No se puede cambiar el Intervalo de tiempo de las métricas existentes. Para obtener más información, consulte [Agregación de datos desde propiedades y otros activos \(métricas\)](#).
- (Solo modelos de activos) Cambie cualquiera de las Definiciones de la jerarquía. No se puede cambiar el Modelo de jerarquía de las jerarquías existentes. Para obtener más información, consulte [Definición de jerarquías de modelos de activos](#).

6. Seleccione Save.

 Note

Las solicitudes de actualización realizadas en la consola se rechazan si otro usuario actualiza correctamente el modelo de activos desde la última vez que usted abrió la página Editar modelo. La consola solicita al usuario que actualice la página Editar modelo para obtener el modelo actualizado. Debe volver a realizar las actualizaciones e intentar guardarlas. Consulte [Bloqueo positivo de las escrituras del modelo de activos](#) para obtener más detalles.

Actualice un modelo de activo, un modelo de componentes o una interfaz (AWS CLI)

Utilice AWS Command Line Interface (AWS CLI) para actualizar un modelo de activos, un modelo de componentes o una interfaz.

Utilice la [UpdateAssetModel](#) API para actualizar el nombre, la descripción y las propiedades de un modelo de activos, un modelo de componentes o una interfaz. Solo en el caso de los modelos de activos, puede actualizar las jerarquías. En el caso de las interfaces, puede actualizar las propiedades y jerarquías. Especifique los siguientes parámetros:

- `assetModelId`: el ID del activo. Es el ID real en formato UUID o el `externalId:myExternalId` si lo tiene. Para obtener más información, consulte [Objetos de referencia con objetos externos IDs](#) en la Guía del usuario de AWS IoT SiteWise .

Especifique el modelo actualizado en la carga útil. Para obtener información sobre el formato esperado de un modelo de activos o de componentes, consulte [Cree modelos de activos en AWS IoT SiteWise](#).

Warning

La [UpdateAssetModelAPI](#) sobrescribe el modelo existente con el modelo que usted proporciona en la carga útil. Para evitar eliminar las propiedades o jerarquías del modelo, debes incluir sus propiedades IDs y sus definiciones en la carga útil del modelo actualizado. Para obtener información sobre cómo consultar la estructura existente del modelo, consulte la [DescribeAssetModel](#) operación.

Note

El siguiente procedimiento solo puede actualizar modelos compuestos de tipo AWS/ALARM. Si desea actualizar los modelos CUSTOM compuestos, utilícelos [UpdateAssetModelCompositeModel](#) en su lugar. Para obtener más información, consulte [Actualización de modelos compuestos personalizados \(componentes\)](#).

Actualización de un modelo de activos o un modelo de componentes (AWS CLI)

1. Ejecute el siguiente comando para recuperar la definición del modelo existente. `asset-model-id` Sustitúyalo por el ID o el ID externo del modelo de activos o del modelo de componentes que desee actualizar.

```
aws iotsitewise describe-asset-model --asset-model-id asset-model-id
```

El comando anterior devuelve la definición del modelo correspondiente a la última versión del modelo.

Para un caso de uso en el que un modelo de activos esté en estado FAILED, recupere la definición del modelo válida correspondiente a su versión activa para crear su solicitud de

actualización. Para obtener más información, consulte [Versiones de modelos de activos](#). Ejecute el siguiente comando para recuperar la definición del modelo de activos activa:

```
aws iotsitewise describe-asset-model --asset-model-id asset-model-id --asset-model-version ACTIVE
```

La operación devuelve una respuesta que contiene los detalles del modelo. La respuesta tiene la siguiente estructura.

```
{
  "assetModelId": "String",
  "assetModelArn": "String",
  "assetModelName": "String",
  "assetModelDescription": "String",
  "assetModelProperties": Array of AssetModelProperty,
  "assetModelHierarchies": Array of AssetModelHierarchyDefinition,
  "assetModelCompositeModels": Array of AssetModelCompositeModel,
  "assetModelCompositeModelSummaries": Array of AssetModelCompositeModelSummary,
  "assetModelCreationDate": "String",
  "assetModelLastUpdateDate": "String",
  "assetModelStatus": {
    "state": "String",
    "error": {
      "code": "String",
      "message": "String"
    }
  },
  "assetModelType": "String",
  "assetModelVersion": "String",
  "eTag": "String"
}
```

Para obtener más información, consulte la operación [DescribeAssetModel](#).

2. Cree un archivo llamado `update-asset-model.json` y copie la respuesta del comando anterior en el archivo.
3. Elimine los siguientes pares de clave-valor del objeto JSON en `update-asset-model.json`:
 - `assetModelId`
 - `assetModelArn`
 - `assetModelCompositeModelSummaries`

- `assetModelCreationDate`
- `assetModelLastUpdateDate`
- `assetModelStatus`
- `assetModelType`
- `assetModelVersion`
- `eTag`

La [UpdateAssetModel](#) operación espera una carga útil con la siguiente estructura:

```
{
  "assetModelName": "String",
  "assetModelDescription": "String",
  "assetModelProperties": Array of AssetModelProperty,
  "assetModelHierarchies": Array of AssetModelHierarchyDefinition,
  "assetModelCompositeModels": Array of AssetModelCompositeModel
}
```

4. En `update-asset-model.json`, realice una de las siguientes acciones:
 - Cambie el nombre del modelo de activos (`assetModelName`).
 - Cambie, agregue o elimine la descripción del modelo de activos (`assetModelDescription`).
 - Cambie, agregue o elimine cualquiera de las propiedades del modelo de activos (`assetModelProperties`). No puede cambiar `dataType` de las propiedades existentes ni `window` de las métricas existentes. Para obtener más información, consulte [Definición de propiedades de datos](#).
 - Cambie, agregue o elimine cualquiera de las jerarquías del modelo de activos (`assetModelHierarchies`). No puede cambiar `childAssetModelId` de las jerarquías existentes. Para obtener más información, consulte [Definición de jerarquías de modelos de activos](#).
 - Cambie, añada o elimine cualquiera de los modelos compuestos del modelo de activos de tipo AWS/ALARM (`assetModelCompositeModels`). Las alarmas monitorean otras propiedades para que pueda identificar cuándo requieren atención los equipos o procesos. Cada definición de alarma es un modelo compuesto que estandariza el conjunto de propiedades que utiliza la alarma. Para obtener más información, consulte [Supervise los](#)

[datos con las alarmas integradas AWS IoT SiteWise](#) y [Defina las alarmas en los modelos de activos en AWS IoT SiteWise](#).

5. Ejecute el siguiente comando para actualizar el modelo de activos con la definición almacenada en `update-asset-model.json`. `asset-model-id` Sustitúyala por el ID del modelo de activos:

```
aws iotsitewise update-asset-model \  
  --asset-model-id asset-model-id \  
  --cli-input-json file://model-payload.json
```

Important

Cuando varios usuarios actualizan un modelo de activos al mismo tiempo, es posible que un usuario sobrescriba sin darse cuenta los cambios de otro usuario. Para evitarlo, debe definir una solicitud de actualización condicional. Consulte [Bloqueo positivo de las escrituras del modelo de activos](#).

Actualización de modelos compuestos personalizados (componentes)

Puede utilizar la AWS IoT SiteWise API para actualizar un modelo compuesto personalizado o la AWS IoT SiteWise consola para actualizar los componentes.

Temas

- [Actualización de un componente \(consola\)](#)
- [Actualización de un modelo compuesto personalizado \(AWS CLI\)](#)

Actualización de un componente (consola)

Puede utilizar la AWS IoT SiteWise consola para actualizar un componente.

Actualización de un componente (consola)

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, elija Models (Modelos).
3. Elija el modelo de activos en el que se encuentra el componente.

4. En la pestaña Propiedades, elija Componentes.
5. Elija el componente que desea actualizar.
6. Elija Edit (Edición de).
7. En la página Editar componente, lleve a cabo alguna de las siguientes operaciones:
 - En Detalles del modelo, cambie el Nombre del modelo.
 - Cambie cualquiera de las Definiciones de atributos. No se puede cambiar el Tipo de datos de los atributos existentes. Para obtener más información, consulte [Definición de datos estáticos \(atributos\)](#).
 - Cambie cualquiera de las Definiciones de mediciones. No se puede cambiar el Tipo de datos de las mediciones existentes. Para obtener más información, consulte [Definición de flujos de datos procedentes del equipo \(mediciones\)](#).
 - Cambie cualquiera de las Definiciones de transformación. Para obtener más información, consulte [Datos de transformación \(transformaciones\)](#).
 - Cambie cualquiera de las Definiciones de métricas. No se puede cambiar el Intervalo de tiempo de las métricas existentes. Para obtener más información, consulte [Agregación de datos desde propiedades y otros activos \(métricas\)](#).
8. Seleccione Save.

Actualización de un modelo compuesto personalizado (AWS CLI)

Utilice AWS Command Line Interface (AWS CLI) para actualizar un modelo compuesto personalizado.

Para actualizar el nombre o la descripción, utilice la [UpdateAssetModelCompositeModel](#) operación. Solo en el caso de los modelos compuestos personalizados en línea, también puede actualizar las propiedades. No puede actualizar las propiedades de un modelo compuesto component-model-based personalizado, ya que el modelo de componentes al que se hace referencia proporciona las propiedades asociadas.

Important

Si elimina una propiedad de un modelo compuesto personalizado, AWS IoT SiteWise elimina todos los datos anteriores de esa propiedad. No se puede cambiar el tipo o el tipo de datos de una propiedad existente.

Para reemplazar la propiedad de un modelo compuesto existente por otra nueva con el mismo name, haga lo siguiente:

1. Envíe una solicitud `UpdateAssetModelCompositeModel` con toda la propiedad existente eliminada.
2. Envía una segunda solicitud `UpdateAssetModelCompositeModel` que incluya la nueva propiedad. La nueva propiedad del activo tendrá la name misma que la anterior y AWS IoT SiteWise generará una nueva propiedad únicaid.

Actualización de un modelo compuesto personalizado (AWS CLI)

1. Para recuperar la definición del modelo compuesto existente, ejecute el siguiente comando.
`composite-model-id` Sustitúyalo por el ID o el ID externo del modelo compuesto personalizado que se va a actualizar y `asset-model-id` por el modelo de activos al que está asociado el modelo compuesto personalizado. Para obtener más información, consulte la Guía del usuario de AWS IoT SiteWise .

- a. Ejecute el siguiente comando:

```
aws iotsitewise describe-asset-model-composite-model \  
--asset-model-composite-model-id composite-model-id \  
--asset-model-id asset-model-id
```

- b. El comando anterior devuelve la definición del modelo compuesto correspondiente a la última versión del modelo asociado. Para un caso de uso en el que un modelo de activos esté en estado FAILED, recupere la definición del modelo válida correspondiente a su versión activa para crear su solicitud de actualización. Para obtener más información, consulte [Versiones de modelos de activos](#).
- c. Ejecute el siguiente comando para recuperar la definición del modelo de activos activa:

```
aws iotsitewise describe-asset-model-composite-model \  
--asset-model-composite-model-id composite-model-id \  
--asset-model-id asset-model-id \  
--asset-model-version ACTIVE
```

- d. Para obtener más información, consulte la operación [DescribeAssetModelCompositeModel](#).
2. Cree un archivo llamado `update-custom-composite-model.json` y, a continuación, copie la respuesta del comando anterior en el archivo.

3. Elimine todos los pares de clave-valor del objeto JSON en `update-custom-composite-model.json`, salvo los campos siguientes:
 - `assetModelCompositeModelName`
 - `assetModelCompositeModelDescription` (si está presente)
 - `assetModelCompositeModelProperties` (si está presente)
4. En `update-custom-composite-model.json`, realice una de las siguientes acciones:
 - Cambie el valor de `assetModelCompositeModelName`.
 - Añada o elimine `assetModelCompositeModelDescription`, o cambie su valor.
 - Solo para modelos compuestos personalizados en línea: cambie, añada o elimine cualquiera de las propiedades del modelo de activos en `assetModelCompositeModelProperties`.

Para obtener más información sobre el formato necesario para este archivo, consulte la sintaxis de la solicitud [UpdateAssetModelCompositeModel](#).

5. Ejecute el siguiente comando para actualizar el modelo compuesto personalizado con la definición almacenada en `update-custom-composite-model.json`. `composite-model-id` Sustitúyalo por el ID del modelo compuesto y `asset-model-id` por el ID del modelo de activos en el que se encuentra.

```
aws iotsitewise update-asset-model-composite-model \  
--asset-model-composite-model-id composite-model-id \  
--asset-model-id asset-model-id \  
--cli-input-json file://update-custom-composite-model.json
```

Important

Cuando varios usuarios actualizan un modelo de activos al mismo tiempo, es posible que un usuario sobrescriba sin darse cuenta los cambios de otro usuario. Para evitarlo, debe definir una solicitud de actualización condicional. Consulte [Bloqueo positivo de las escrituras del modelo de activos](#).

Bloqueo positivo de las escrituras del modelo de activos

Al actualizar un modelo de activos, el usuario hace lo siguiente:

1. Lee la definición del modelo de activos actual.
2. Edita la definición del modelo de activos con los cambios necesarios.
3. Actualiza el modelo de activos con la nueva definición.

En un escenario en el que dos usuarios actualizan un modelo, es posible hacer lo siguiente:

- El usuario A lee la definición X del modelo de activos.
- El usuario B lee la definición X del modelo de activos y confirma los cambios, modificando la definición de X.
- El usuario A confirma y sobrescribe el cambio realizado por el usuario B para el modelo de activos X, sin verificar ni incorporar los cambios del usuario B.

El bloqueo optimista es un mecanismo que se utiliza AWS IoT SiteWise para evitar sobrescrituras accidentales, como en el escenario anterior. El bloqueo positivo es una estrategia para garantizar que la versión actual del modelo de activos que se está actualizando o eliminando sea la misma que la versión actual de AWS IoT SiteWise. Esto evita que las actualizaciones accidentales sobrescriban las escrituras del modelo de activos.

Siga estos pasos para realizar escrituras del modelo de activos con bloqueo positivo:

Temas

- [Realización de escrituras del modelo de activos con bloqueo positivo \(consola\)](#)
- [Realización de escrituras del modelo de activos con bloqueo positivo \(AWS CLI\)](#)

Realización de escrituras del modelo de activos con bloqueo positivo (consola)

El siguiente procedimiento describe cómo realizar escrituras del modelo de activos con bloqueo positivo sobre la versión activa del modelo de activos de la consola.

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, elija Models (Modelos).
3. Elija el modelo de activos o el modelo de componentes que desee actualizar.
4. Elija Edit (Edición de).
5. Realice cambios en la página Editar modelo.
6. Seleccione Save.

Note

Algunas veces se han producido una o varias actualizaciones correctas del modelo entre el momento en que el usuario comienza a editar el modelo y el momento en que guarda las modificaciones realizadas en el modelo.

Para garantizar que el usuario no sobrescriba accidentalmente las nuevas actualizaciones realizadas correctamente, se rechaza la escritura del usuario. La consola deshabilita el botón Guardar y solicita al usuario que actualice la página Editar modelo. El usuario debe volver a actualizar la nueva versión activa del modelo. El usuario debe dar los siguientes pasos adicionales:

7. Elija Actualizar.
8. Siga los pasos 5 y 6 de nuevo.

Realización de escrituras del modelo de activos con bloqueo positivo (AWS CLI)

El siguiente procedimiento describe cómo realizar escrituras del modelo de activos con bloqueo positivo en la AWS CLI.

1. Obtenga la definición del modelo ETag asociada al modelo actual

ETag es un token único que se genera para cada nueva representación de un modelo de activos. Llame a la [DescribeAssetModel](#) API para obtener la definición del modelo de activos actual y la asociada ETag a la respuesta.

En actualizaciones simultáneas los usuarios realizan actualizaciones correctas (modelo en estado ACTIVE) o actualizaciones incorrectas (modelo en estado FAILED). Para garantizar que un usuario no sobrescriba accidentalmente una actualización correcta, debe recuperar la versión activa del modelo de activos de [Versiones de modelos de activos](#) y obtener el valor de ETag.

Use el siguiente comando:

```
aws iotsitewise describe-asset-model --asset-model-id asset-model-id \  
--asset-model-version ACTIVE
```

La respuesta devuelve la siguiente estructura:

```
{
  "assetModelId": "String",
  "assetModelArn": "String",
  "assetModelName": "String",
  ...
  "eTag": "String"
}
```

Note

Debe recuperar la última versión del modelo de activos y su ETag para no sobrescribir ninguna actualización.

2. Realice las operaciones UPDATE y DELETE con condiciones de escritura

El siguiente modelo de activos APIs apoya un bloqueo optimista:

- [UpdateAssetModel](#)
- [DeleteAssetModel](#)
- [CreateAssetModelCompositeModel](#)
- [UpdateAssetModelCompositeModel](#)
- [DeleteAssetModelCompositeModel](#)

Note

Los siguientes escenarios utilizan la API `UpdateAssetModel` como referencia. Las condiciones se aplican a todas las operaciones enumeradas anteriormente.

Los siguientes escenarios describen las diferentes condiciones de escritura en función de los requisitos de control de simultaneidad:

- Ejecute el siguiente comando para no sobrescribir actualizaciones correctas. No debe existir una nueva versión activa desde la última versión activa leída. Sustituya `e-tag` por la ETag devuelta en la operación de la API utilizada en la lectura de la versión activa.

```
aws iotsitewise update-asset-model \  
  --asset-model-id asset-model-id \  
  --if-match e-tag \  
  --match-for-version-type ACTIVE \  
  --cli-input-json file://model-payload.json
```

- Cuando se produce un error en la creación de un modelo, todavía no existe una versión activa para él, porque se encuentra en estado FAILED. Todavía sigue siendo posible sobrescribir una nueva versión activa que esté presente antes de que se confirmen los cambios. Ejecute el siguiente comando para no sobrescribir una nueva versión activa cuando no haya una versión activa durante la última lectura.

```
aws iotsitewise update-asset-model \  
  --asset-model-id asset-model-id \  
  --if-none-match "*" \  
  --match-for-version-type ACTIVE \  
  --cli-input-json file://model-payload.json
```

- Ejecute el siguiente comando para evitar sobrescribir actualizaciones correctas o fallidas. Este comando define una condición de escritura que garantiza que no se cree una versión más reciente desde la versión más reciente leída por última vez. Sustituya e-tag por la ETag devuelta en la operación de la API utilizada en la lectura de la versión activa.

```
aws iotsitewise update-asset-model \  
  --asset-model-id asset-model-id \  
  --if-match eTag \  
  --match-for-version-type LATEST \  
  --cli-input-json file://model-payload.json
```

Si la condición de escritura se evalúa como FALSE, la solicitud de escritura fallará con la `PreconditionFailedException`.

Elimine activos y modelos en AWS IoT SiteWise

Puede eliminar sus activos, modelos de activos, modelos de componentes e interfaces una AWS IoT SiteWise vez que haya terminado con ellos. Las operaciones de eliminación son asíncronas y tardan un tiempo en propagarse. AWS IoT SiteWise

Temas

- [Elimine activos en AWS IoT SiteWise](#)
- [Elimine los modelos de activos, los modelos de componentes y las interfaces en AWS IoT SiteWise](#)

Elimine activos en AWS IoT SiteWise

Puede usar la AWS IoT SiteWise consola o la API para eliminar un activo que ya no necesite en su entorno. Al eliminar un modelo de activos, también se eliminan todos los modelos de activos y componentes asociados. Sin embargo, es importante tener en cuenta que eliminar un activo o un modelo es una acción permanente, y que también se eliminarán todos los datos asociados a los recursos eliminados. Antes de eliminar activos o modelos, se recomienda revisar las dependencias o integraciones que puedan verse afectadas y asegurarse de tener una copia de seguridad de todos los datos importantes.

Antes de poder eliminar un activo, primero debe disociar sus activos secundarios y disociarlos de su activo principal. Para obtener más información, consulte [Asociación y desasociación de activos](#). Si usa AWS Command Line Interface (AWS CLI), puede usar la [ListAssociatedAssets](#) operación para enumerar los elementos secundarios de un activo.

Cuando se elimina un activo, su estado es DELETING hasta que se propagan los cambios. Para obtener más información, consulte [Estados de activos y modelos](#). Después de eliminar el activo, no puede consultar ese activo. Si lo hace, la API devuelve una respuesta HTTP 404.

Important

AWS IoT SiteWise elimina todos los datos de propiedad de los activos eliminados.

Temas

- [Eliminación de un activo \(consola\)](#)
- [Eliminación de un activo \(AWS CLI\)](#)

Eliminación de un activo (consola)

Puede utilizar la AWS IoT SiteWise consola para eliminar un activo.

Para eliminar un activo (consola)

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, elija Activos.
3. Elija el activo que desea eliminar.

Tip

Puede elegir el icono de flecha para expandir una jerarquía de activos y encontrar su activo.

4. Si el activo tiene Activos asociados, elimine cada activo. Puede elegir el nombre de un activo para navegar hasta su página, donde puede eliminarlo.
5. En la página del activo, elija Eliminar.
6. En el cuadro de diálogo Eliminar activo, haga lo siguiente:
 - a. Escriba **Delete** para confirmar la eliminación.
 - b. Elija Eliminar.

Eliminación de un activo (AWS CLI)

Puede usar el AWS Command Line Interface (AWS CLI) para eliminar un activo.

Utilice la [DeleteAsset](#) operación para eliminar un activo. Especifique el siguiente parámetro:

- `assetId`: el ID del activo. Es el ID real en formato UUID o el `externalId:myExternalId` si lo tiene. Para obtener más información, consulte [Objetos de referencia con objetos externos IDs](#) en la Guía del usuario de AWS IoT SiteWise .

Eliminación de un activo (AWS CLI)

1. Ejecute el siguiente comando para mostrar las jerarquías del activo. `asset-id` Sustitúyalo por el ID o el ID externo del activo:

```
aws iotsitewise describe-asset --asset-id asset-id
```

La operación devuelve una respuesta que contiene los detalles del activo. La respuesta contiene una lista `assetHierarchies` que tiene la siguiente estructura:

```
{
  ...
  "assetHierarchies": [
    {
      "id": "String",
      "name": "String"
    }
  ],
  ...
}
```

Para obtener más información, consulte la operación [DescribeAsset](#).

2. Para cada jerarquía, ejecute el siguiente comando para mostrar los secundarios del activo que están asociados con esa jerarquía. `asset-id` Sustitúyalo por el ID o el ID externo del activo y `hierarchy-id` por el ID o el ID externo de la jerarquía.

```
aws iotsitewise list-associated-assets \
  --asset-id asset-id \
  --hierarchy-id hierarchy-id
```

Para obtener más información, consulte la operación [ListAssociatedAssets](#).

3. Ejecute el siguiente comando para eliminar cada activo asociado y, a continuación, para eliminar el activo. `asset-id` Sustitúyalo por el ID o el ID externo del activo.

```
aws iotsitewise delete-asset --asset-id asset-id
```

Elimine los modelos de activos, los modelos de componentes y las interfaces en AWS IoT SiteWise

Puede utilizar la AWS IoT SiteWise consola o la API para eliminar un modelo de activos, un modelo de componentes o una interfaz.

Antes de poder eliminar un modelo de activos, primero debe eliminar todos los activos creados a partir del modelo de activos. Antes de poder eliminar una interfaz, primero debe desvincularla de todos los modelos de activos que la implementan.

Al eliminar un modelo de activo o una interfaz, su estado es DELETING hasta que se propaguen los cambios. Para obtener más información, consulte [Estados de activos y modelos](#). Una vez que se haya eliminado el modelo o la interfaz de activos, no podrá consultar ese modelo o interfaz de activos. Si lo hace, la API devuelve una respuesta HTTP 404.

Temas

- [Elimine un modelo de activos, un modelo de componentes o una interfaz \(consola\)](#)
- [Elimine un modelo de activo, un modelo de componentes o una interfaz \(AWS CLI\)](#)

Elimine un modelo de activos, un modelo de componentes o una interfaz (consola)

Puede utilizar la AWS IoT SiteWise consola para eliminar un modelo de activos, un modelo de componentes o una interfaz.

Para eliminar un modelo de activo, un modelo de componentes o una interfaz (consola)

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, elija Models (Modelos).
3. Elija el modelo de activo, el modelo de componentes o la interfaz que desee eliminar.
4. Si elimina un modelo de activo y tiene algún activo, elimine cada activo. Elija el nombre de un activo para navegar hasta su página, donde puede eliminarlo. Para obtener más información, consulte [Eliminación de un activo \(consola\)](#).
5. En la página del modelo, elija Eliminar.
6. En el cuadro de diálogo Eliminar modelo, haga lo siguiente:
 - a. Escriba **Delete** para confirmar la eliminación.
 - b. Elija Eliminar.

Elimine un modelo de activo, un modelo de componentes o una interfaz (AWS CLI)

Puede usar AWS Command Line Interface (AWS CLI) para eliminar un modelo de activos, un modelo de componentes o una interfaz.

Utilice la [DeleteAssetModel](#) operación para eliminar un modelo de activos, un modelo de componentes o una interfaz. Especifique el siguiente parámetro:

- `assetModelId`: el ID del activo. Es el ID real en formato UUID o el `externalId:myExternalId` si lo tiene. Para obtener más información, consulte [Objetos de referencia con objetos externos IDs](#) en la Guía del usuario de AWS IoT SiteWise .

Eliminación de un modelo de activos (AWS CLI)

1. Ejecute el siguiente comando para mostrar todos los activos creados a partir del modelo. `asset-model-id` Sustitúyalo por el ID o el ID externo del modelo de activos.

```
aws iotsitewise list-assets --asset-model-id asset-model-id
```

Para obtener más información, consulte la operación [ListAssets](#).

2. Si el comando anterior devuelve activos del modelo, elimine cada activo. Para obtener más información, consulte [Eliminación de un activo \(AWS CLI\)](#).
3. Ejecute el siguiente comando para eliminar el modelo de activos. `asset-model-id` Sustitúyalo por el ID o el ID externo del modelo de activos.

```
aws iotsitewise delete-asset-model --asset-model-id asset-model-id
```

Important

Para evitar eliminar un modelo de activos que se ha actualizado simultáneamente desde la última operación de lectura, debe definir una solicitud de eliminación condicional. Consulte [Bloqueo positivo de las escrituras del modelo de activos](#).

Operaciones masivas con activos y modelos

Para trabajar con una gran cantidad de activos o modelos de activos, utilice operaciones masivas para importar y exportar recursos de forma masiva a una ubicación diferente. Por ejemplo, puede crear un archivo de datos que defina los activos o modelos de activos en un bucket de Amazon S3 y utilizar la importación masiva para crearlos o actualizarlos en AWS IoT SiteWise. Como alternativa,

si tiene una gran cantidad de activos o modelos de activos AWS IoT SiteWise, puede exportarlos a Amazon S3.

Note

Las operaciones masivas se realizan AWS IoT SiteWise mediante llamadas a las operaciones de la AWS IoT TwinMaker API. Puede hacerlo sin configurar AWS IoT TwinMaker ni crear un AWS IoT TwinMaker espacio de trabajo. Lo único que necesita es un bucket de Amazon S3 en el que puede colocar su contenido de AWS IoT SiteWise .

Temas

- [Conceptos y terminología clave](#)
- [Funcionalidades compatibles](#)
- [Requisitos previos para operaciones masivas](#)
- [Ejecute un trabajo de importación masiva](#)
- [Ejecución de un trabajo de exportación masiva](#)
- [Seguimiento del progreso de los trabajos y gestión de errores](#)
- [Ejemplos de metadatos de importación](#)
- [Ejemplos de metadatos de exportación](#)
- [AWS IoT SiteWise esquema de tareas de transferencia de metadatos](#)

Conceptos y terminología clave

AWS IoT SiteWise Las funciones de importación y exportación masivas se basan en los siguientes conceptos y terminología:

- **Importar:** acción de mover activos o modelos de activos desde un archivo de un bucket de Amazon S3 hasta AWS IoT SiteWise.
- **Exportación:** acción de mover activos o modelos de activos desde AWS IoT SiteWise un bucket de Amazon S3.
- **Origen:** ubicación inicial desde la que desea mover el contenido.

Por ejemplo, un bucket de Amazon S3 es una fuente de importación y AWS IoT SiteWise una fuente de exportación.

- Destino: ubicación a la que desea mover el contenido.

Por ejemplo, un bucket de Amazon S3 es un destino de exportación y AWS IoT SiteWise un destino de importación.

- AWS IoT SiteWise Esquema: este esquema se utiliza para importar y exportar metadatos desde AWS IoT SiteWise.
- Recurso de nivel superior: un AWS IoT SiteWise recurso que puede crear o actualizar de forma individual, como un activo o un modelo de activos.
- Subrecurso: recurso anidado dentro de un AWS IoT SiteWise recurso de nivel superior. Algunos ejemplos son: propiedades, jerarquías y modelos compuestos.
- Metadatos: información clave necesaria para importar o exportar recursos correctamente. Algunos ejemplos de metadatos son las definiciones de activos y los modelos de activos.
- metadataTransferJob: el objeto que se crea cuando corres. `CreateMetadataTransferJob`

Funcionalidades compatibles

En este tema se explica lo que puede hacer cuando ejecuta una operación masiva. Las operaciones masivas admiten las siguientes funciones:

- Creación de recursos de nivel superior: al importar un activo o un modelo de activos que no define un ID o cuyo ID no coincide con el de uno existente, se creará como un recurso nuevo.
- Sustitución de recursos de nivel superior: cuando importa un activo o un modelo de activos cuyo ID coincide con uno que ya existe, reemplazará el recurso existente.
- Creación, sustitución o eliminación de subrecursos: cuando la importación sustituye un recurso de nivel superior, como un activo o un modelo de activos, la nueva definición sustituye todos los subrecursos, como propiedades, jerarquías o modelos compuestos.

Por ejemplo, si actualiza un modelo de activos durante una importación masiva y la versión actualizada define una propiedad que no estaba presente en el original, se crea una nueva propiedad. Si define una propiedad que ya existe, se actualizará la propiedad existente. Si el modelo de activos actualizado omite una propiedad que estaba presente en el original, se elimina la propiedad.

- No se eliminan los recursos de nivel superior: las operaciones masivas no eliminan un activo o un modelo de activos. Las operaciones masivas solo las crean o actualizan.

Requisitos previos para operaciones masivas

En esta sección se explican los requisitos previos de las operaciones masivas, incluidos los permisos AWS Identity and Access Management (de IAM) para intercambiar recursos entre los AWS servicios y la máquina local. Antes de iniciar una operación masiva cumpla los siguientes requisitos previos:

- Cree un bucket de Amazon S3 para almacenar recursos. Para obtener más información sobre el uso de Amazon S3, consulte [¿Qué es Amazon S3?](#)

Permisos de IAM

Para realizar operaciones masivas, debe crear una política AWS Identity and Access Management (IAM) con permisos que permitan el intercambio de AWS recursos entre Amazon S3 y su máquina local. AWS IoT SiteWise Para obtener más información acerca de la creación de políticas de IAM, consulte [Crear políticas de IAM](#).

Para realizar operaciones masivas necesita las siguientes políticas.

AWS IoT SiteWise política

Esta política permite el acceso a las acciones de la AWS IoT SiteWise API necesarias para las operaciones masivas:

```
{
  "Sid": "SiteWiseApiAccess",
  "Effect": "Allow",
  "Action": [
    "iotsitewise:CreateAsset",
    "iotsitewise:CreateAssetModel",
    "iotsitewise:UpdateAsset",
    "iotsitewise:UpdateAssetModel",
    "iotsitewise:UpdateAssetProperty",
    "iotsitewise:ListAssets",
    "iotsitewise:ListAssetModels",
    "iotsitewise:ListAssetProperties",
    "iotsitewise:ListAssetModelProperties",
    "iotsitewise:ListAssociatedAssets",
    "iotsitewise:DescribeAsset",
    "iotsitewise:DescribeAssetModel",
    "iotsitewise:DescribeAssetProperty",
    "iotsitewise:AssociateAssets",
```

```

    "iotsitewise:DisassociateAssets",
    "iotsitewise:AssociateTimeSeriesToAssetProperty",
    "iotsitewise:DisassociateTimeSeriesFromAssetProperty",
    "iotsitewise:BatchPutAssetPropertyValue",
    "iotsitewise:BatchGetAssetPropertyValue",
    "iotsitewise:TagResource",
    "iotsitewise:UntagResource",
    "iotsitewise:ListTagsForResource",
    "iotsitewise>CreateAssetModelCompositeModel",
    "iotsitewise:UpdateAssetModelCompositeModel",
    "iotsitewise:DescribeAssetModelCompositeModel",
    "iotsitewise>DeleteAssetModelCompositeModel",
    "iotsitewise>ListAssetModelCompositeModels",
    "iotsitewise>ListCompositionRelationships",
    "iotsitewise:DescribeAssetCompositeModel"
  ],
  "Resource": "*"
}

```

AWS IoT TwinMaker política

Esta política permite el acceso a las operaciones de la AWS IoT TwinMaker API que se utilizan para trabajar con operaciones masivas:

```

{
  "Sid": "MetadataTransferJobApiAccess",
  "Effect": "Allow",
  "Action": [
    "iottwinmaker:CreateMetadataTransferJob",
    "iottwinmaker:CancelMetadataTransferJob",
    "iottwinmaker:GetMetadataTransferJob",
    "iottwinmaker:ListMetadataTransferJobs"
  ],
  "Resource": "*"
}

```

Política de Amazon S3

Esta política brinda acceso a buckets de Amazon S3 para transferir metadatos para operaciones masivas.

For a specific Amazon S3 bucket

Si utiliza un bucket específico para trabajar con los metadatos de sus operaciones masivas, esta política le proporciona acceso a ese bucket:

```
{
  "Effect": "Allow",
  "Action": [
    "s3:PutObject",
    "s3:GetObject",
    "s3:GetBucketLocation",
    "s3:ListBucket",
    "s3:AbortMultipartUpload",
    "s3:ListBucketMultipartUploads",
    "s3:ListMultipartUploadParts"
  ],
  "Resource": [
    "arn:aws:s3:::bucket name",
    "arn:aws:s3:::bucket name/*"
  ]
}
```

To allow any Amazon S3 bucket

Si va a utilizar muchos buckets distintos para trabajar con los metadatos de sus operaciones masivas, esta política le proporciona acceso a ese bucket:

```
{
  "Effect": "Allow",
  "Action": [
    "s3:PutObject",
    "s3:GetObject",
    "s3:GetBucketLocation",
    "s3:ListBucket",
    "s3:AbortMultipartUpload",
    "s3:ListBucketMultipartUploads",
    "s3:ListMultipartUploadParts"
  ],
  "Resource": "*"
}
```

Para obtener información acerca de cómo solucionar problemas de importación y exportación, consulte [Solución de importación y exportación masivas](#).

Ejecute un trabajo de importación masiva

La importación masiva es la acción de mover los metadatos a un AWS IoT SiteWise espacio de trabajo. Por ejemplo, la importación masiva puede mover los metadatos de un archivo local o de un archivo de un bucket de Amazon S3 a un AWS IoT SiteWise espacio de trabajo.

Paso 1: preparación del archivo que desea importar

Descargue el archivo en formato AWS IoT SiteWise nativo para importar los activos y los modelos de activos. Consulte [AWS IoT SiteWise esquema de tareas de transferencia de metadatos](#) para obtener más detalles.

Paso 2: carga del archivo preparado en Amazon S3.

Cargue el archivo en Amazon S3. Para obtener más información, consulte [Uploading a file to Amazon S3](#) en la Guía del usuario de Amazon Simple Storage Service.

Importación de metadatos (consola)

Puede usarlo Consola de AWS IoT SiteWise para importar metadatos de forma masiva. Siga [Paso 1: preparación del archivo que desea importar](#) y [Paso 2: carga del archivo preparado en Amazon S3](#) para preparar un archivo que esté listo para su importación.

Importación de datos de Amazon S3 a Consola de AWS IoT SiteWise

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, elija Operaciones masivas Nueva.
3. Elija Nueva importación para iniciar el proceso de importación.
4. En la página Importar metadatos:
 - Seleccione Browse Amazon S3 para ver el bucket y los archivos de Amazon S3.
 - Navegue hasta el bucket de Amazon S3 que contiene el archivo de importación preparado.
 - Seleccione el archivo que desea importar.
 - Revise el archivo seleccionado y elija Importar.
5. La página Operaciones masivas en SiteWise metadatos Consola de AWS IoT SiteWise muestra el trabajo de importación recién creado en la tabla de progreso de los trabajos.

Importación de metadatos (AWS CLI)

Para realizar una acción de importación, lleve a cabo el siguiente procedimiento:

Importación de datos de Amazon S3 a AWS CLI

1. Cree un archivo de metadatos que especifique los recursos que desea importar, siguiendo el [AWS IoT SiteWise esquema de tareas de transferencia de metadatos](#). Almacene este archivo en su bucket de Amazon S3.

Para ver ejemplos de archivos de metadatos para importar, consulte [Ejemplos de metadatos de importación](#).

2. Ahora cree un archivo JSON con el cuerpo de la solicitud. El cuerpo de la solicitud especifica el origen y el destino del trabajo de transferencia. Este archivo es independiente del archivo del paso anterior. Asegúrese de especificar su bucket de Amazon S3 como origen y `iotsitewise` como destino.

El siguiente ejemplo muestra el cuerpo de la solicitud:

```
{
  "metadataTransferJobId": "your-transfer-job-Id",
  "sources": [{
    "type": "s3",
    "s3Configuration": {
      "location": "arn:aws:s3::amzn-s3-demo-bucket/
your_import_metadata.json"
    }
  ]},
  "destination": {
    "type": "iotsitewise"
  }
}
```

3. `CreateMetadataTransferJobPara` invocarlo, ejecute el siguiente AWS CLI comando. En este ejemplo, el archivo del cuerpo de la solicitud del paso anterior se llama `createMetadataTransferJobExport.json`.

```
aws iottwinmaker create-metadata-transfer-job --region us-east-1 \
  --cli-input-json file://createMetadataTransferJobImport.json
```

Se creará un trabajo de transferencia de metadatos y se iniciará el proceso de transferencia de los recursos seleccionados.

Ejecución de un trabajo de exportación masiva

La exportación masiva es la acción de mover los metadatos de un AWS IoT SiteWise espacio de trabajo a un bucket de Amazon S3.

Al realizar una exportación masiva de su AWS IoT SiteWise contenido a Amazon S3, puede especificar filtros para limitar los modelos de activos y activos específicos que desea exportar.

Los filtros deben especificarse en una sección `iotSiteWiseConfiguration` dentro de la sección de orígenes de su solicitud JSON.

Note

Puede incluir varios filtros en la solicitud. La operación masiva exportará los modelos de activos y los activos que coincidan con cualquiera de los filtros.

Si no proporciona ningún filtro, la operación masiva exporta todos los modelos de activos y los activos.

Example cuerpo de la solicitud con filtros

```
{
  "metadataTransferJobId": "your-transfer-job-id",
  "sources": [
    {
      "type": "iotsitewise",
      "iotSiteWiseConfiguration": {
        "filters": [
          {
            "filterByAssetModel": {
              "assetModelId": "asset model ID"
            }
          },
          {
            "filterByAssetModel": {
```

```

        "assetModelId": "asset model ID",
        "includeAssets": true
    }
},
{
    "filterByAssetModel": {
        "assetModelId": "asset model ID",
        "includeOffspring": true
    }
}
]
}
},
],
"destination": {
    "type": "s3",
    "s3Configuration": {
        "location": "arn:aws:s3:::amzn-s3-demo-bucket"
    }
}
}
}

```

Exportación de metadatos (consola)

El siguiente procedimiento explica la acción de exportación de la consola:

Cree un trabajo de exportación en el Consola de AWS IoT SiteWise

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, elija Operaciones masivas Nueva.
3. Seleccione Nueva exportación para iniciar el proceso de exportación.
4. En la página Exportar metadatos:
 - Introduzca un nombre para el trabajo de exportación. Es el nombre utilizado para el archivo exportado en el bucket de Amazon S3.
 - Elija los recursos que desea exportar, lo que establece los filtros para el trabajo:
 - Exporte todos los activos y modelos de activos. Utilice filtros en los activos y los modelos de activos.
 - Exporte activos. Filtre los activos.
 - Seleccione el activo que desee utilizar para el filtro de exportación.

- (Opcional) Añada la descendencia o el modelo de activos asociado.
 - Exporte modelos de activos. Filtre sus modelos de activos.
 - Seleccione el modelo de activos que desee utilizar para el filtro de exportación.
 - (Opcional) Añada la descendencia, el activo asociado o ambos.
 - Elija Siguiente.
 - Navegue hasta el bucket de Amazon S3:
 - Seleccione Browse Amazon S3 para ver el bucket y los archivos de Amazon S3.
 - Navegue hasta el bucket de Amazon S3 donde se debe colocar el archivo.
 - Elija Siguiente.
 - Revise el trabajo de exportación y seleccione Exportar.
5. La página Operaciones masivas en SiteWise metadatos Consola de AWS IoT SiteWise muestra el trabajo de importación recién creado en la tabla de progreso de los trabajos.

Para conocer las diferentes formas de utilizar los filtros al exportar metadatos, consulte [Ejemplos de metadatos de exportación](#).

Exportación de metadatos (AWS CLI)

El siguiente procedimiento explica la acción AWS CLI de exportación:

Exportación de datos desde AWS IoT SiteWise Amazon S3

1. Cree un archivo JSON con el cuerpo de la solicitud. El cuerpo de la solicitud especifica el origen y el destino del trabajo de transferencia. El siguiente ejemplo muestra un cuerpo de solicitud de ejemplo:

```
{
  "metadataTransferJobId": "your-transfer-job-Id",
  "sources": [{
    "type": "iotsitewise"
  }],
  "destination": {
    "type": "s3",
    "s3Configuration": {
      "location": "arn:aws:s3:::amzn-s3-demo-bucket"
    }
  }
}
```

```
}  
}
```

Asegúrese de especificar su bucket de Amazon S3 como destino del trabajo de transferencia de metadatos.

Note

En este ejemplo se exportarán todos sus modelos de activos y sus activos. Para limitar la exportación a modelos de activos o activos específicos, puede incluir filtros en el cuerpo de la solicitud. Para obtener más información sobre la aplicación de filtros de exportación, consulte [Ejemplos de metadatos de exportación](#).

2. Guarde el archivo del cuerpo de la solicitud para utilizarlo en el siguiente paso. En este ejemplo, el archivo se denomina `createMetadataTransferJobExport.json`.
3. `CreateMetadataTransferJob` Para invocarlo, ejecute el siguiente AWS CLI comando:

```
aws iottwinmaker create-metadata-transfer-job --region us-east-1 \  
--cli-input-json file://createMetadataTransferJobExport.json
```

Sustituya el archivo JSON de entrada `createMetadataTransferJobExport.json` por el nombre de su propio archivo de transferencia.

Seguimiento del progreso de los trabajos y gestión de errores

Procesar un trabajo de proceso masivo lleva tiempo. Cada trabajo se procesa en el orden en que se AWS IoT SiteWise recibió la solicitud. Se procesa one-at-a-time para cada cuenta. Cuando se completa un trabajo, el siguiente en la cola comienza a procesarse automáticamente. AWS IoT SiteWise resuelve los trabajos de forma asíncrona y actualiza el estado de cada uno a medida que avanza. Cada trabajo tiene un campo de estado que muestra el estado del recurso y un mensaje de error, si procede.

El estado puede ser uno de los siguientes valores:

- **VALIDATING**: validar el trabajo, incluido el formato de archivo enviado y su contenido.
- **PENDING**: el trabajo está en una cola. Puede cancelar los trabajos en este estado desde la consola de AWS IoT SiteWise , pero todos los demás estados continuarán hasta el final.

- **RUNNING:** procesando el trabajo. Se crean y actualizan los recursos según lo definido en el archivo de importación, o se exportan los recursos en función de los filtros elegidos para el trabajo de exportación. Si se cancela, no se eliminará ningún recurso importado por este trabajo. Para obtener más información, consulte [Revise el progreso y los detalles de los trabajos \(consola\)](#).
- **CANCELLING:** el trabajo se está cancelando activamente.
- **ERROR:** no se han podido procesar uno o más recursos. Consulte el informe detallado del trabajo para obtener más información. Para obtener más información, consulte [Inspección de detalles de errores \(consola\)](#).
- **COMPLETED:** trabajo realizado sin errores.
- **CANCELLED:** el trabajo está cancelado y no está en cola. Si ha cancelado un RUNNING trabajo, los recursos que ya había importado este trabajo en el momento de la cancelación no se eliminarán de él AWS IoT SiteWise.

Temas

- [Seguimiento del progreso de los trabajos](#)
- [Inspeccione los errores para AWS IoT SiteWise](#)

Seguimiento del progreso de los trabajos

Revise el progreso y los detalles de los trabajos (consola)

Consulte [Importación de metadatos \(consola\)](#) o [Exportación de metadatos \(consola\)](#) para iniciar un trabajo masivo.

Descripción general del progreso del trabajo en la AWS IoT SiteWise consola:

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, elija Operaciones masivas Nueva.
3. La tabla de progreso de los trabajos de la AWS IoT SiteWise consola muestra la lista de trabajos de operaciones masivas.
4. La columna Tipo de trabajo describe si se trata de un trabajo de exportación o de importación. Las columnas Fecha de importación muestran la fecha en que se inició el trabajo.
5. La columna Estado muestra el estado del trabajo. Puede seleccionar un trabajo para ver sus detalles.

6. El trabajo seleccionado muestra `Correcto` si se ha realizado correctamente o una lista de errores si ha fallado. También se muestra una descripción del error con cada tipo de recurso.

Descripción general de los detalles del trabajo en la AWS IoT SiteWise consola:

La tabla de progreso de los trabajos de la AWS IoT SiteWise consola muestra la lista de trabajos de operaciones masivas.

1. Elija un trabajo para ver más detalles.
2. Para un trabajo de importación, el `Data source ARN` representa la ubicación en Amazon S3 del archivo de importación.
3. Para un trabajo de exportación, el `Data destination ARN` representa la ubicación del archivo en Amazon S3 tras la exportación.
4. `Status` y `Status reason` proporcionan detalles adicionales sobre el trabajo actual. Consulte [Seguimiento del progreso de los trabajos y gestión de errores](#) para obtener más detalles.
5. `Queued position` representa la posición del trabajo en la cola de procesos. Los trabajos se procesan uno a uno. Una posición en cola igual a 1 indica que el trabajo se procesará a continuación.
6. La página de detalles de los trabajos también muestra los recuentos de progreso de los trabajos.
 - Los tipos de recuento de progreso de los trabajo son:
 - i. `Total resources`: indica el recuento total de activos en el proceso de transferencia.
 - ii. `Succeeded`: indica el recuento de activos transferidos correctamente durante el proceso.
 - iii. `Failed`: indica el recuento de activos que han fallado durante el proceso.
 - iv. `Skipped`: indica el recuento de activos que se han omitido durante el proceso.
7. Un estado de trabajo de `PENDING` o `VALIDATING` muestra todos los recuentos de progreso de los trabajos como `-`. Esto indica que se están evaluando los recuentos de progreso de los trabajos.
8. Un estado de trabajo de `RUNNING` muestra el recuento de `Total resources`, el trabajo enviado para su procesamiento. Los recuentos detallados (`Succeeded`, `Failed` y `Skipped`) se aplican a los recursos procesados. La suma de los recuentos detallados es menor que el recuento de `Total resources` hasta que el estado del trabajo es `COMPLETED` o `ERROR`.

9. Si el estado de un trabajo es COMPLETED o ERROR, el recuento de `Total resources` es igual a la suma de los recuentos detallados (`Succeeded`, `Failed` y `Skipped`).
10. Si el estado de un trabajo es ERROR, consulte la tabla Errores de trabajo para obtener detalles sobre errores y fallos específicos. Consulte [Inspección de detalles de errores \(consola\)](#) para obtener más detalles.

Revise el progreso y los detalles de los trabajos (AWS CLI)

Tras iniciar una operación masiva, puede comprobar o actualizar su estado mediante las siguientes acciones de la API:

- Para recuperar información sobre un trabajo específico, usa la acción de la [GetMetadataTransferJobAPI](#).

Recupere información con la API **GetMetadataTransferJob**:

1. Cree y ejecute un trabajo de transferencia. Llame a la API `GetMetadataTransferJob`.

Example AWS CLI comando:

```
aws iottwinmaker get-metadata-transfer-job \  
  --metadata-transfer-job-id your_metadata_transfer_job_id \  
  --region your_region
```

2. La API `GetMetadataTransferJob` devuelve un objeto `MetadataTransferJobProgress` con los siguientes parámetros:
 - `succeededCount`: indica el recuento de activos transferidos correctamente durante el proceso.
 - `failedCount`: indica el recuento de activos que han fallado durante el proceso.
 - `skippedCount`: indica el recuento de activos que se han omitido durante el proceso.
 - `totalCount`: indica el recuento total de activos en el proceso de transferencia.

Estos parámetros indican el estado del progreso del trabajo. Si el estado es `RUNNING`, ayudan a hacer un seguimiento de la cantidad de recursos que aún no se han procesado.

Si encuentra errores de validación del esquema o si `failedCount` es mayor o igual que 1, el estado de progreso de los trabajos pasa a ser `ERROR`. En el bucket de Amazon S3 se incluye

un informe de error completo del trabajo. Consulte [Inspeccione los errores para AWS IoT SiteWise](#) para obtener más detalles.

- Para enumerar los trabajos actuales, usa la acción [ListMetadataTransferJobs](#) de la API.

Use un archivo JSON para filtrar los trabajos devueltos en función de su estado actual. Consulte el procedimiento siguiente:

1. Para especificar los filtros que quieres usar, crea un archivo JSON AWS CLI de entrada. Quieres usar:

```
{
  "sourceType": "s3",
  "destinationType": "iottwinmaker",
  "filters": [{
    "state": "COMPLETED"
  }]
}
```

Para obtener una lista de state valores válidos, consulta la Guía [ListMetadataTransferJobsFilter](#) de referencia de la AWS IoT TwinMaker API.

2. Usa el archivo JSON como argumento en el siguiente comando de AWS CLI ejemplo:

```
aws iottwinmaker list-metadata-transfer-job --region your_region \
  --cli-input-json file://ListMetadataTransferJobsExample.json
```

- Para cancelar un trabajo, usa la acción de la [CancelMetadataTransferJob](#) API. Esta API cancela el trabajo de transferencia de metadatos específico sin que ello afecte a los recursos ya exportados o importados:

```
aws iottwinmaker cancel-metadata-transfer-job \
  --region your_region \
  --metadata-transfer-job-id job-to-cancel-id
```

Inspeccione los errores para AWS IoT SiteWise

Inspección de detalles de errores (consola)

Detalles del error en la AWS IoT SiteWise consola:

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. Consulte la tabla de progreso de los trabajos Consola de AWS IoT SiteWise para ver una lista de los trabajos de operaciones masivas.
3. Seleccione un trabajo para ver sus detalles.
4. Si el estado de un trabajo es COMPLETED o ERROR, el recuento de Total resources es igual a la suma de los recuentos detallados (Succeeded, Failed y Skipped).
5. Si el estado de un trabajo es ERROR, consulte la tabla Errores de trabajo para obtener detalles sobre errores y fallos específicos.
6. La tabla Errores de trabajo muestra el contenido del informe de trabajos. El campo Resource type indica la ubicación del error o los fallos, por ejemplo:
 - Por ejemplo, un error de validación en la Bulk operations template en el campo Resource type indica que la plantilla de importación y el formato del archivo del esquema de metadatos no coinciden. Para obtener más información, consulte [AWS IoT SiteWise esquema de tareas de transferencia de metadatos](#).
 - Un Asset con error en el campo Resource type indica que el activo no se ha creado debido a un conflicto con otro activo. Consulte [Errores comunes](#) para obtener información sobre los errores y conflictos de AWS IoT SiteWise recursos.

Inspección de detalles de errores (AWS CLI)

Para gestionar y diagnosticar los errores producidos durante un trabajo de transferencia, consulte el siguiente procedimiento sobre el uso de la acción de la API GetMetadataTransferJob:

1. Tras crear y ejecutar un trabajo de transferencia, llame a [GetMetadataTransferJob](#):

```
aws iottwinmaker get-metadata-transfer-job \  
  --metadata-transfer-job-id your_metadata_transfer_job_id \  
  --region us-east-1
```

2. Cuando vea que el estado del trabajo pasa a COMPLETED, puede empezar a verificar los resultados del trabajo.

3. Cuando llama a `GetMetadataTransferJob`, devuelve un objeto llamado [MetadataTransferJobProgress](#).

El `MetadataTransferJobProgress` objeto contiene los siguientes parámetros:

- `failedCount`: indica el recuento de activos que han fallado durante el proceso.
 - `skippedCount`: indica el recuento de activos que se han omitido durante el proceso de transferencia.
 - `succeededCount`: indica el recuento de activos transferidos correctamente durante el proceso.
 - `totalCount`: indica el recuento total de activos implicados en el proceso de transferencia.
4. Además, la llamada a la API devuelve un elemento `reportUrl`, que contiene una URL prefirmada. Si su trabajo de transferencia tiene algún problema que deba investigar más a fondo, visita esta URL.

Ejemplos de metadatos de importación

Esta sección muestra cómo crear archivos de metadatos para importar modelos de activos y activos con una sola operación de importación masiva.

Ejemplo de una importación masiva

Puede importar muchos modelos de activos y activos con una sola operación de importación masiva. En el ejemplo siguiente se muestra cómo crear un archivo de metadatos para hacerlo.

En este escenario de ejemplo, tiene varios sitios de trabajo que contienen robots industriales en células de trabajo.

El ejemplo define dos modelos de activos:

- `RobotModel1`: este modelo de activos representa un tipo concreto de robot que tiene en sus sitios de trabajo. El robot tiene una propiedad de medición, `Temperature`.
- `WorkCell`: este modelo de activos representa un conjunto de robots dentro de uno de sus sitios de trabajo. El modelo de activos define una jerarquía, `robotHierarchyOEM1`, para representar la relación que una célula de trabajo contiene robots.

El ejemplo también define algunos activos:

- `WorkCell1`: una célula de trabajo en su sitio de Boston

- RobotArm123456: un robot en esa célula de trabajo
- RobotArm987654: otro robot en esa célula de trabajo

El siguiente archivo de metadatos JSON define estos modelos de activos y activos. Al realizar una importación masiva con estos metadatos, se crean los modelos de activos y los activos que contienen AWS IoT SiteWise, incluidas sus relaciones jerárquicas.

Archivo de metadatos para importación

```
{
  "assetModels": [
    {
      "assetModelExternalId": "Robot.OEM1.3536",
      "assetModelName": "RobotModel1",
      "assetModelProperties": [
        {
          "dataType": "DOUBLE",
          "externalId": "Temperature",
          "name": "Temperature",
          "type": {
            "measurement": {
              "processingConfig": {
                "forwardingConfig": {
                  "state": "ENABLED"
                }
              }
            }
          }
        },
        {
          "unit": "fahrenheit"
        }
      ]
    },
    {
      "assetModelExternalId": "ISA95.WorkCell",
      "assetModelName": "WorkCell",
      "assetModelProperties": [],
      "assetModelHierarchies": [
        {
          "externalId": "workCellHierarchyWithOEM1Robot",
          "name": "robotHierarchyOEM1",
          "childAssetModelExternalId": "Robot.OEM1.3536"
        }
      ]
    }
  ]
}
```

```

    ]
  }
],
"assets": [
  {
    "assetExternalId": "Robot.OEM1.3536.123456",
    "assetName": "RobotArm123456",
    "assetModelExternalId": "Robot.OEM1.3536"
  },
  {
    "assetExternalId": "Robot.OEM1.3536.987654",
    "assetName": "RobotArm987654",
    "assetModelExternalId": "Robot.OEM1.3536"
  },
  {
    "assetExternalId": "BostonSite.Area1.Line1.WorkCell1",
    "assetName": "WorkCell1",
    "assetModelExternalId": "ISA95.WorkCell",
    "assetHierarchies": [
      {
        "externalId": "workCellHierarchyWithOEM1Robot",
        "childAssetExternalId": "Robot.OEM1.3536.123456"
      },
      {
        "externalId": "workCellHierarchyWithOEM1Robot",
        "childAssetExternalId": "Robot.OEM1.3536.987654"
      }
    ]
  }
]
}

```

Ejemplo de incorporación inicial de modelos y activos

En este escenario de ejemplo, tiene varios sitios de trabajo que contienen robots industriales en una empresa.

El ejemplo define varios modelos de activos:

- **Sample_Enterprise**: este modelo de activos representa la empresa de la que forman parte los sitios. El modelo de activos define una jerarquía, **Enterprise to Site**, para representar la relación de los sitios con la empresa.

- `Sample_Site`: este modelo de activos representa los sitios de fabricación de la empresa. El modelo de activos define una jerarquía, `Site` to `Line`, para representar la relación de las líneas con el sitio.
- `Sample_Welding Line`: este modelo de activos representa una línea de ensamblaje en los sitios de trabajo. El modelo de activos define una jerarquía, `Line` to `Robot`, para representar la relación de los robots con la línea.
- `Sample_Welding Robot`: este modelo de activos representa un tipo particular de robot en sus sitios de trabajo.

El ejemplo también define los activos en función de los modelos de activos.

- `Sample_AnyCompany Motor`: este activo se crea a partir de un modelo de activos de `Sample_Enterprise`.
- `Sample_Chicago`: este activo se crea a partir de un modelo de activos de `Sample_Site`.
- `Sample_Welding Line 1`: este activo se crea a partir de un modelo de activos de `Sample_Welding Line`.
- `Sample_Welding Robot 1`: este activo se crea a partir de un modelo de activos de `Sample_Welding Robot`.
- `Sample_Welding Robot 2`: este activo se crea a partir de un modelo de activos de `Sample_Welding Robot`.

El siguiente archivo de metadatos JSON define estos modelos de activos y activos. Al realizar una importación masiva con estos metadatos, se crean los modelos de activos y los activos que contienen AWS IoT SiteWise, incluidas sus relaciones jerárquicas.

Archivo JSON para incorporar activos y modelos para importación

```
{
  "assetModels": [
    {
      "assetModelExternalId": "External_Id_Welding_Robot",
      "assetModelName": "Sample_Welding Robot",
      "assetModelProperties": [
        {
          "dataType": "STRING",
          "externalId": "External_Id_Welding_Robot_Serial_Number",
```

```

        "name": "Serial Number",
        "type": {
            "attribute": {
                "defaultValue": "-"
            }
        },
        "unit": "-"
    },
    {
        "dataType": "DOUBLE",
        "externalId": "External_Id_Welding_Robot_Cycle_Count",
        "name": "CycleCount",
        "type": {
            "measurement": {}
        },
        "unit": "EA"
    },
    {
        "dataType": "DOUBLE",
        "externalId": "External_Id_Welding_Robot_Joint_1_Current",
        "name": "Joint 1 Current",
        "type": {
            "measurement": {}
        },
        "unit": "Amps"
    },
    {
        "dataType": "DOUBLE",
        "externalId": "External_Id_Welding_Robot_Joint_1_Max_Current",
        "name": "Max Joint 1 Current",
        "type": {
            "metric": {
                "expression": "max(joint1current)",
                "variables": [
                    {
                        "name": "joint1current",
                        "value": {
                            "propertyExternalId":
"External_Id_Welding_Robot_Joint_1_Current"
                        }
                    }
                ],
                "window": {
                    "tumbling": {

```

```

        "interval": "5m"
      }
    }
  },
  "unit": "Amps"
}
],
},
{
  "assetModelExternalId": "External_Id_Welding_Line",
  "assetModelName": "Sample_Welding Line",
  "assetModelProperties": [
    {
      "dataType": "DOUBLE",
      "externalId": "External_Id_Welding_Line_Availability",
      "name": "Availability",
      "type": {
        "measurement": {}
      },
      "unit": "%"
    }
  ],
  "assetModelHierarchies": [
    {
      "externalId": "External_Id_Welding_Line_T0_Robot",
      "name": "Line to Robot",
      "childAssetModelExternalId": "External_Id_Welding_Robot"
    }
  ]
},
{
  "assetModelExternalId": "External_Id_Site",
  "assetModelName": "Sample_Site",
  "assetModelProperties": [
    {
      "dataType": "STRING",
      "externalId": "External_Id_Site_Street_Address",
      "name": "Street Address",
      "type": {
        "attribute": {
          "defaultValue": "-"
        }
      }
    }
  ],

```

```

        "unit": "-"
      }
    ],
    "assetModelHierarchies": [
      {
        "externalId": "External_Id_Site_T0_Line",
        "name": "Site to Line",
        "childAssetModelExternalId": "External_Id_Welding_Line"
      }
    ]
  },
  {
    "assetModelExternalId": "External_Id_Enterprise",
    "assetModelName": "Sample_Enterprise",
    "assetModelProperties": [
      {
        "dataType": "STRING",
        "name": "Company Name",
        "externalId": "External_Id_Enterprise_Company_Name",
        "type": {
          "attribute": {
            "defaultValue": "-"
          }
        },
        "unit": "-"
      }
    ],
    "assetModelHierarchies": [
      {
        "externalId": "External_Id_Enterprise_T0_Site",
        "name": "Enterprise to Site",
        "childAssetModelExternalId": "External_Id_Site"
      }
    ]
  }
],
"assets": [
  {
    "assetExternalId": "External_Id_Welding_Robot_1",
    "assetName": "Sample_Welding Robot 1",
    "assetModelExternalId": "External_Id_Welding_Robot",
    "assetProperties": [
      {
        "externalId": "External_Id_Welding_Robot_Serial_Number",

```

```

        "attributeValue": "S1000"
    },
    {
        "externalId": "External_Id_Welding_Robot_Cycle_Count",
        "alias": "AnyCompany/Chicago/Welding Line/S1000/Count"
    },
    {
        "externalId": "External_Id_Welding_Robot_Joint_1_Current",
        "alias": "AnyCompany/Chicago/Welding Line/S1000/1/Current"
    }
]
},
{
    "assetExternalId": "External_Id_Welding_Robot_2",
    "assetName": "Sample_Welding Robot 2",
    "assetModelExternalId": "External_Id_Welding_Robot",
    "assetProperties": [
        {
            "externalId": "External_Id_Welding_Robot_Serial_Number",
            "attributeValue": "S2000"
        },
        {
            "externalId": "External_Id_Welding_Robot_Cycle_Count",
            "alias": "AnyCompany/Chicago/Welding Line/S2000/Count"
        },
        {
            "externalId": "External_Id_Welding_Robot_Joint_1_Current",
            "alias": "AnyCompany/Chicago/Welding Line/S2000/1/Current"
        }
    ]
},
{
    "assetExternalId": "External_Id_Welding_Line_1",
    "assetName": "Sample_Welding Line 1",
    "assetModelExternalId": "External_Id_Welding_Line",
    "assetProperties": [
        {
            "externalId": "External_Id_Welding_Line_Availability",
            "alias": "AnyCompany/Chicago/Welding Line/Availability"
        }
    ],
    "assetHierarchies": [
        {
            "externalId": "External_Id_Welding_Line_T0_Robot",

```

```

        "childAssetExternalId": "External_Id_Welding_Robot_1"
    },
    {
        "externalId": "External_Id_Welding_Line_T0_Robot",
        "childAssetExternalId": "External_Id_Welding_Robot_2"
    }
]
},
{
    "assetExternalId": "External_Id_Site_Chicago",
    "assetName": "Sample_Chicago",
    "assetModelExternalId": "External_Id_Site",
    "assetHierarchies": [
        {
            "externalId": "External_Id_Site_T0_Line",
            "childAssetExternalId": "External_Id_Welding_Line_1"
        }
    ]
},
{
    "assetExternalId": "External_Id_Enterprise_AnyCompany",
    "assetName": "Sample_AnyEnterprise Motor",
    "assetModelExternalId": "External_Id_Enterprise",
    "assetHierarchies": [
        {
            "externalId": "External_Id_Enterprise_T0_Site",
            "childAssetExternalId": "External_Id_Site_Chicago"
        }
    ]
}
]
}
}

```

La siguiente captura de pantalla muestra los modelos que se muestran en la Consola de AWS IoT SiteWise después de ejecutar el ejemplo de código anterior.

IoT SiteWise > Models

Models (4) Refresh Create component model Create asset model

Assets represent industrial devices and processes that send data streams to SiteWise. Models are structures that enforce a specific model of properties and hierarchies for all instances of each asset. You must create every asset from a model.

Filter instances

Name	Status	Model type	Date created	Date modified
Sample_Enterprise	ACTIVE	Asset model	November 10, 2023 at 11:22:13 (UT...)	November 10, 202...
Sample_Site	ACTIVE	Asset model	November 10, 2023 at 11:21:57 (UT...)	November 10, 202...
Sample_Welding Line	ACTIVE	Asset model	November 10, 2023 at 11:21:40 (UT...)	November 10, 202...
Sample_Welding Robot	ACTIVE	Asset model	November 10, 2023 at 11:21:24 (UT...)	November 10, 202...

La siguiente captura de pantalla muestra los modelos, activos y jerarquías que se muestran Consola de AWS IoT SiteWise después de ejecutar el ejemplo de código anterior.

IoT SiteWise > Assets

Assets (1) Refresh Create asset

Assets represent industrial devices and processes that send data streams to SiteWise. Models are structures that enforce a specific model of properties and hierarchies for all instances of each asset. You must create every asset from a model.

Filter top level assets

Name	Description	Status	Date created	Date modified
<input type="checkbox"/> Sample_AnyEnterprise Motor		ACTIVE	November 10, 2023 at 11:23:06 (UTC-5:00)	November 10, 2023 at 11:23:06 (UTC-...
<input type="checkbox"/> Sample_Chicago		ACTIVE	November 10, 2023 at 11:22:57 (UTC-5:00)	November 10, 2023 at 11:22:57 (UTC-...
<input type="checkbox"/> Sample_Welding Line 1		ACTIVE	November 10, 2023 at 11:22:48 (UTC-5:00)	November 10, 2023 at 11:22:48 (UTC-...
<input type="checkbox"/> Sample_Welding Robot 1		ACTIVE	November 10, 2023 at 11:22:39 (UTC-5:00)	November 10, 2023 at 11:22:39 (UTC-...
<input type="checkbox"/> Sample_Welding Robot 2		ACTIVE	November 10, 2023 at 11:22:30 (UTC-5:00)	November 10, 2023 at 11:22:30 (UTC-...

Ejemplo de incorporación de activos adicionales

En este ejemplo se definen activos adicionales para importarlos a un modelo de activos existente en su cuenta:

- **Sample_Welding Line 2**: este activo se crea a partir de un modelo de activos de **Sample_Welding Line**.
- **Sample_Welding Robot 3**: este activo se crea a partir de un modelo de activos de **Sample_Welding Robot**.
- **Sample_Welding Robot 4**: este activo se crea a partir de un modelo de activos de **Sample_Welding Robot**.

Para crear los activos iniciales de este ejemplo, consulte [Ejemplo de incorporación inicial de modelos y activos](#).

El siguiente archivo de metadatos JSON define estos modelos de activos y activos. Al realizar una importación masiva con estos metadatos, se crean los modelos de activos y los activos que contienen AWS IoT SiteWise, incluidas sus relaciones jerárquicas.

Archivo JSON para incorporar activos adicionales

```
{
  "assets": [
    {
      "assetExternalId": "External_Id_Welding_Robot_3",
      "assetName": "Sample_Welding Robot 3",
      "assetModelExternalId": "External_Id_Welding_Robot",
      "assetProperties": [
        {
          "externalId": "External_Id_Welding_Robot_Serial_Number",
          "attributeValue": "S3000"
        },
        {
          "externalId": "External_Id_Welding_Robot_Cycle_Count",
          "alias": "AnyCompany/Chicago/Welding Line/S3000/Count"
        },
        {
          "externalId": "External_Id_Welding_Robot_Joint_1_Current",
          "alias": "AnyCompany/Chicago/Welding Line/S3000/1/Current"
        }
      ]
    },
    {
      "assetExternalId": "External_Id_Welding_Robot_4",
      "assetName": "Sample_Welding Robot 4",
      "assetModelExternalId": "External_Id_Welding_Robot",
      "assetProperties": [
        {
          "externalId": "External_Id_Welding_Robot_Serial_Number",
          "attributeValue": "S4000"
        },
        {
          "externalId": "External_Id_Welding_Robot_Cycle_Count",
          "alias": "AnyCompany/Chicago/Welding Line/S4000/Count"
        }
      ]
    }
  ]
}
```

```

    },
    {
        "externalId": "External_Id_Welding_Robot_Joint_1_Current",
        "alias": "AnyCompany/Chicago/Welding Line/S4000/1/Current"
    }
]
},
{
    "assetExternalId": "External_Id_Welding_Line_1",
    "assetName": "Sample_Welding Line 1",
    "assetModelExternalId": "External_Id_Welding_Line",
    "assetHierarchies": [
        {
            "externalId": "External_Id_Welding_Line_T0_Robot",
            "childAssetExternalId": "External_Id_Welding_Robot_1"
        },
        {
            "externalId": "External_Id_Welding_Line_T0_Robot",
            "childAssetExternalId": "External_Id_Welding_Robot_2"
        },
        {
            "externalId": "External_Id_Welding_Line_T0_Robot",
            "childAssetExternalId": "External_Id_Welding_Robot_3"
        }
    ]
},
{
    "assetExternalId": "External_Id_Welding_Line_2",
    "assetName": "Sample_Welding Line 2",
    "assetModelExternalId": "External_Id_Welding_Line",
    "assetHierarchies": [
        {
            "externalId": "External_Id_Welding_Line_T0_Robot",
            "childAssetExternalId": "External_Id_Welding_Robot_4"
        }
    ]
},
{
    "assetExternalId": "External_Id_Site_Chicago",
    "assetName": "Sample_Chicago",
    "assetModelExternalId": "External_Id_Site",
    "assetHierarchies": [
        {
            "externalId": "External_Id_Site_T0_Line",

```

```

        "childAssetExternalId": "External_Id_Welding_Line_1"
      },
    {
      "externalId": "External_Id_Site_T0_Line",
      "childAssetExternalId": "External_Id_Welding_Line_2"
    }
  ]
}

```

La siguiente captura de pantalla muestra los modelos, activos y jerarquías que se muestran en la Consola de AWS IoT SiteWise después de ejecutar el ejemplo de código anterior.

The screenshot shows the AWS IoT SiteWise console interface for the 'Assets' section. At the top, there is a breadcrumb 'IoT SiteWise > Assets' and a 'Create asset' button. Below this, a search bar is labeled 'Filter top level assets'. The main content is a table with columns for Name, Description, Status, Date created, and Date modified. The table displays a hierarchical structure of assets:

- Sample_AnyCompany Motor (ACTIVE, created Nov 09, 2023 at 19:18:05)
- Sample_Chicago (ACTIVE, created Nov 09, 2023 at 19:17:56)
 - Sample_Welding Line 1 (ACTIVE, created Nov 09, 2023 at 19:17:48)
 - Sample_Welding Robot 2 (ACTIVE, created Nov 09, 2023 at 19:17:39)
 - Sample_Welding Robot 3 (ACTIVE, created Nov 09, 2023 at 20:40:02)
 - Sample_Welding Robot 1 (ACTIVE, created Nov 09, 2023 at 19:17:30)
 - Sample_Welding Line 2 (ACTIVE, created Nov 09, 2023 at 20:40:20)
 - Sample_Welding Robot 4 (ACTIVE, created Nov 09, 2023 at 20:40:11)

Ejemplo de incorporación de nuevas propiedades

En este ejemplo, se definen nuevas propiedades en modelos de activos existentes. Consulte [Ejemplo de incorporación de activos adicionales](#) para incorporar activos y modelos adicionales.

- **Joint 1 Temperature:** esta propiedad se añade al modelo de activos de Sample_Welding Robot. Esta nueva propiedad también se propagará a cada activo creado a partir del modelo de activos de Sample_Welding Robot.

Para añadir una nueva propiedad a un modelo de activos existente, consulte el siguiente archivo de metadatos JSON de ejemplo. Como se muestra en el JSON, se debe proporcionar la definición

completa del modelo de activos de `Sample_Welding Robot` existente junto con la nueva propiedad. Si no se proporciona la lista completa de propiedades de la definición existente, AWS IoT SiteWise elimina las propiedades omitidas.

Archivo JSON para incorporar nuevas propiedades

En este ejemplo se añade una nueva propiedad `Joint 1 Temperature` al modelo de activos.

```
{
  "assetModels": [
    {
      "assetModelExternalId": "External_Id_Welding_Robot",
      "assetModelName": "Sample_Welding Robot",
      "assetModelProperties": [
        {
          "dataType": "STRING",
          "externalId": "External_Id_Welding_Robot_Serial_Number",
          "name": "Serial Number",
          "type": {
            "attribute": {
              "defaultValue": "-"
            }
          },
          "unit": "-"
        },
        {
          "dataType": "DOUBLE",
          "externalId": "External_Id_Welding_Robot_Cycle_Count",
          "name": "CycleCount",
          "type": {
            "measurement": {}
          },
          "unit": "EA"
        },
        {
          "dataType": "DOUBLE",
          "externalId": "External_Id_Welding_Robot_Joint_1_Current",
          "name": "Joint 1 Current",
          "type": {
            "measurement": {}
          },
          "unit": "Amps"
        }
      ]
    }
  ]
}
```

```

    },
    {
      "dataType": "DOUBLE",
      "externalId": "External_Id_Welding_Robot_Joint_1_Max_Current",
      "name": "Max Joint 1 Current",
      "type": {
        "metric": {
          "expression": "max(joint1current)",
          "variables": [
            {
              "name": "joint1current",
              "value": {
                "propertyExternalId":
"External_Id_Welding_Robot_Joint_1_Current"
              }
            }
          ],
          "window": {
            "tumbling": {
              "interval": "5m"
            }
          }
        }
      },
      "unit": "Amps"
    },
    {
      "dataType": "DOUBLE",
      "externalId": "External_Id_Welding_Robot_Joint_1_Temperature",
      "name": "Joint 1 Temperature",
      "type": {
        "measurement": {}
      },
      "unit": "degC"
    }
  ]
}

```

Ejemplo de administración de flujos de datos

En este ejemplo se muestran dos formas de administrar flujos de datos asociados a una propiedad del activo. Al cambiar el nombre del alias de una propiedad del activo, hay dos opciones para los datos históricos almacenados actualmente en el flujo de datos de la propiedad del activo.

- Primera opción: conservar el flujo de datos actual y cambiar el nombre únicamente del alias, de forma que se pueda acceder a los datos históricos con el nuevo alias.

En el archivo de metadatos JSON de ejemplo, la propiedad del activo con el ID `External_Id_Welding_Robot_Cycle_Count` cambia su alias a `AnyCompany/Chicago/Welding Line/S3000/Count-Updated`. Los datos históricos de esta propiedad del activo siguen siendo los mismos después de este cambio.

- Opción dos: asignar un nuevo flujo de datos a la propiedad del activo al que se pueda acceder con el nuevo alias. El flujo de datos anterior, junto con sus datos históricos, sigue siendo accesible con el alias anterior, pero no está asociado a ninguna propiedad del activo.

En el archivo de metadatos JSON de ejemplo, la propiedad del activo con el ID `External_Id_Welding_Robot_Joint_1_Current` cambia su alias a `AnyCompany/Chicago/Welding Line/S4999/1/Current`. Esta vez, el valor adicional `retainDataOnAliasChange` está presente y establecido en `False`. Con esta configuración, el flujo de datos original se disocia de la propiedad del activo y se crea un flujo de datos nuevo que no contiene datos históricos.

Para acceder al flujo de datos antiguo con los datos históricos originales AWS Console Home, vaya a la página Flujos de datos y busque el alias anterior. `AnyCompany/Chicago/Welding Line/S3000/1/Current`

Archivo JSON para actualizar alias de propiedad

```
{
  "assetExternalId": "External_Id_Welding_Robot_3",
  "assetName": "Sample_Welding Robot 3",
  "assetModelExternalId": "External_Id_Welding_Robot",
  "assetProperties": [
    {
      "externalId": "External_Id_Welding_Robot_Serial_Number",
      "attributeValue": "S3000"
    }
  ]
}
```

```

    },
    {
      "externalId": "External_Id_Welding_Robot_Cycle_Count",
      "alias": "AnyCompany/Chicago/Welding Line/S3000/Count-Updated"
    },
    {
      "externalId": "External_Id_Welding_Robot_Joint_1_Current",
      "alias": "AnyCompany/Chicago/Welding Line/S4999/1/Current",
      "retainDataOnAliasChange": "FALSE"
    }
  ]
}

```

Ejemplos de metadatos de exportación

Al realizar una exportación masiva de su AWS IoT SiteWise contenido a Amazon S3, puede especificar filtros para limitar los modelos de activos y activos específicos que desea exportar.

Los filtros se especifican en una sección de `iotSiteWiseConfiguration` en la sección `sources` del cuerpo de la solicitud.

Note

También puede incluir varios filtros. La operación masiva exportará cualquier modelo de activos o activo que coincida con alguno de los filtros.

Si no proporciona ningún filtro, la operación exportará todos los modelos de activos y los activos.

```

{
  "metadataTransferJobId": "your-transfer-job-id",
  "sources": [{
    "type": "iotsitewise",
    "iotSiteWiseConfiguration": {
      "filters": [{
        list of filters
      }]
    }
  ]},
  "destination": {
    "type": "s3",

```

```
    "s3Configuration": {
      "location": "arn:aws:s3:::amzn-s3-demo-bucket"
    }
  }
}
```

Filtrado por modelo de activos

Puede filtrar un modelo de activos específico. También puede incluir todos los activos que utilizan ese modelo o todos los modelos de activos dentro de su jerarquía. No puede incluir a la vez activos y jerarquía.

Para obtener más información acerca de las jerarquías, consulte [Definición de jerarquías de modelos de activos](#).

Asset model

Este filtro incluye el modelo de activos especificado:

```
"filterByAssetModel": {
  "assetModelId": "asset model ID"
}
```

Asset model and its assets

Este filtro incluye el modelo de activos especificado, junto con todos los activos que utilizan ese modelo de activos:

```
"filterByAssetModel": {
  "assetModelId": "asset model ID",
  "includeAssets": true
}
```

Asset model and its hierarchy

Este filtro incluye el modelo de activos especificado, junto con todos los modelos de activos asociados en su jerarquía:

```
"filterByAssetModel": {
  "assetModelId": "asset model ID",
  "includeOffspring": true
}
```

```
}
```

Filtrado por activo

Puede filtrar un activo específico. También puede incluir su modelo de activos o todos los activos asociados dentro de su jerarquía. No puede incluir a la vez modelo de activos y jerarquía.

Para obtener más información acerca de las jerarquías, consulte [Definición de jerarquías de modelos de activos](#).

Asset

Este filtro incluye el activo especificado:

```
"filterByAsset": {  
  "assetId": "asset ID"  
}
```

Asset and its asset model

Este filtro incluye el activo especificado, junto con el modelo de activos que utiliza:

```
"filterByAsset": {  
  "assetId": "asset ID",  
  "includeAssetModel": true  
}
```

Asset and its hierarchy

Este filtro incluye el activo especificado, junto con todos los activos asociados en su jerarquía:

```
"filterByAsset": {  
  "assetId": "asset ID",  
  "includeOffspring": true  
}
```

AWS IoT SiteWise esquema de tareas de transferencia de metadatos

Utilice el esquema de tareas de transferencia de AWS IoT SiteWise metadatos como referencia cuando realice sus propias operaciones de importación y exportación masivas:

```

{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "title": "IoTSiteWise",
  "description": "Metadata transfer job resource schema for IoTSiteWise",
  "definitions": {
    "Name": {
      "type": "string",
      "minLength": 1,
      "maxLength": 256,
      "pattern": "[^\\u0000-\\u001F\\u007F]+"
    },
    "Description": {
      "type": "string",
      "minLength": 1,
      "maxLength": 2048,
      "pattern": "[^\\u0000-\\u001F\\u007F]+"
    },
    "ID": {
      "type": "string",
      "minLength": 36,
      "maxLength": 36,
      "pattern": "^[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}$"
    },
    "ExternalId": {
      "type": "string",
      "minLength": 2,
      "maxLength": 128,
      "pattern": "[a-zA-Z0-9_][a-zA-Z_\\-0-9.:]*[a-zA-Z0-9_]+"
    },
    "AttributeValue": {
      "description": "The value of the property attribute.",
      "type": "string",
      "pattern": "[^\\u0000-\\u001F\\u007F]+"
    },
    "PropertyUnit": {
      "description": "The unit of measure (such as Newtons or RPM) of the asset property.",
      "type": "string",
      "minLength": 1,
      "maxLength": 256,
      "pattern": "[^\\u0000-\\u001F\\u007F]+"
    },
    "PropertyAlias": {

```

```

    "description": "The property alias that identifies the property.",
    "type": "string",
    "minLength": 1,
    "maxLength": 1000,
    "pattern": "[^\\u0000-\\u001F\\u007F]+"
  },
  "AssetProperty": {
    "description": "The asset property's definition, alias, unit, and notification
state.",
    "type": "object",
    "additionalProperties": false,
    "anyOf": [
      {
        "required": [
          "id"
        ]
      },
      {
        "required": [
          "externalId"
        ]
      }
    ],
    "properties": {
      "id": {
        "description": "The ID of the asset property.",
        "$ref": "#/definitions/ID"
      },
      "externalId": {
        "description": "The ExternalID of the asset property.",
        "$ref": "#/definitions/ExternalId"
      },
      "alias": {
        "$ref": "#/definitions/PropertyAlias"
      },
      "unit": {
        "$ref": "#/definitions/PropertyUnit"
      },
      "attributeValue": {
        "$ref": "#/definitions/AttributeValue"
      },
      "retainDataOnAliasChange": {
        "type": "string",
        "default": "TRUE",

```

```

        "enum": [
            "TRUE",
            "FALSE"
        ]
    },
    "propertyNotificationState": {
        "description": "The MQTT notification state (ENABLED or DISABLED) for this
asset property.",
        "type": "string",
        "enum": [
            "ENABLED",
            "DISABLED"
        ]
    }
}
},
"AssetHierarchy": {
    "description": "A hierarchy specifies allowed parent/child asset relationships.",
    "type": "object",
    "additionalProperties": false,
    "anyOf": [
        {
            "required": [
                "id",
                "childAssetId"
            ]
        },
        {
            "required": [
                "externalId",
                "childAssetId"
            ]
        },
        {
            "required": [
                "id",
                "childAssetExternalId"
            ]
        },
        {
            "required": [
                "externalId",
                "childAssetExternalId"
            ]
        }
    ]
}

```

```
    }
  ],
  "properties": {
    "id": {
      "description": "The ID of a hierarchy in the parent asset's model.",
      "$ref": "#/definitions/ID"
    },
    "externalId": {
      "description": "The ExternalID of a hierarchy in the parent asset's model.",
      "$ref": "#/definitions/ExternalId"
    },
    "childAssetId": {
      "description": "The ID of the child asset to be associated.",
      "$ref": "#/definitions/ID"
    },
    "childAssetExternalId": {
      "description": "The ExternalID of the child asset to be associated.",
      "$ref": "#/definitions/ExternalId"
    }
  }
},
"Tag": {
  "type": "object",
  "additionalProperties": false,
  "required": [
    "key",
    "value"
  ],
  "properties": {
    "key": {
      "type": "string"
    },
    "value": {
      "type": "string"
    }
  }
},
"AssetModelType": {
  "type": "string",
  "default": null,
  "enum": [
    "ASSET_MODEL",
    "COMPONENT_MODEL"
  ]
}
```

```

    },
    "AssetModelCompositeModel": {
      "description": "Contains a composite model definition in an asset model. This composite model definition is applied to all assets created from the asset model.",
      "type": "object",
      "additionalProperties": false,
      "anyOf": [
        {
          "required": [
            "id"
          ]
        },
        {
          "required": [
            "externalId"
          ]
        }
      ],
      "required": [
        "name",
        "type"
      ],
      "properties": {
        "id": {
          "description": "The ID of the asset model composite model.",
          "$ref": "#/definitions/ID"
        },
        "externalId": {
          "description": "The ExternalID of the asset model composite model.",
          "$ref": "#/definitions/ExternalId"
        },
        "parentId": {
          "description": "The ID of the parent asset model composite model.",
          "$ref": "#/definitions/ID"
        },
        "parentExternalId": {
          "description": "The ExternalID of the parent asset model composite model.",
          "$ref": "#/definitions/ExternalId"
        },
        "composedAssetModelId": {
          "description": "The ID of the composed asset model.",
          "$ref": "#/definitions/ID"
        },
        "composedAssetModelExternalId": {

```

```

    "description": "The ExternalID of the composed asset model.",
    "$ref": "#/definitions/ExternalId"
  },
  "description": {
    "description": "A description for the asset composite model.",
    "$ref": "#/definitions/Description"
  },
  "name": {
    "description": "A unique, friendly name for the asset composite model.",
    "$ref": "#/definitions/Name"
  },
  "type": {
    "description": "The type of the composite model. For alarm composite models,
this type is AWS/ALARM.",
    "$ref": "#/definitions/Name"
  },
  "properties": {
    "description": "The property definitions of the asset model.",
    "type": "array",
    "items": {
      "$ref": "#/definitions/AssetModelProperty"
    }
  }
}
},
"AssetModelProperty": {
  "description": "Contains information about an asset model property.",
  "type": "object",
  "additionalProperties": false,
  "anyOf": [
    {
      "required": [
        "id"
      ]
    },
    {
      "required": [
        "externalId"
      ]
    }
  ],
  "required": [
    "name",
    "dataType",

```

```

    "type"
  ],
  "properties": {
    "id": {
      "description": "The ID of the asset model property.",
      "$ref": "#/definitions/ID"
    },
    "externalId": {
      "description": "The ExternalID of the asset model property.",
      "$ref": "#/definitions/ExternalId"
    },
    "name": {
      "description": "The name of the asset model property.",
      "$ref": "#/definitions/Name"
    },
    "dataType": {
      "description": "The data type of the asset model property.",
      "$ref": "#/definitions/DataType"
    },
    "dataTypeSpec": {
      "description": "The data type of the structure for this property.",
      "$ref": "#/definitions/Name"
    },
    "unit": {
      "description": "The unit of the asset model property, such as Newtons or
RPM.",
      "type": "string",
      "minLength": 1,
      "maxLength": 256,
      "pattern": "[^\\u0000-\\u001F\\u007F]+"
    },
    "type": {
      "description": "The property type",
      "$ref": "#/definitions/PropertyType"
    }
  }
},
"DataType": {
  "type": "string",
  "enum": [
    "STRING",
    "INTEGER",
    "DOUBLE",
    "BOOLEAN",

```

```

    "STRUCT"
  ]
},
"PropertyType": {
  "description": "Contains a property type, which can be one of attribute,
measurement, metric, or transform.",
  "type": "object",
  "additionalProperties": false,
  "properties": {
    "attribute": {
      "$ref": "#/definitions/Attribute"
    },
    "transform": {
      "$ref": "#/definitions/Transform"
    },
    "metric": {
      "$ref": "#/definitions/Metric"
    },
    "measurement": {
      "$ref": "#/definitions/Measurement"
    }
  }
},
"Attribute": {
  "type": "object",
  "additionalProperties": false,
  "properties": {
    "defaultValue": {
      "type": "string",
      "pattern": "[^\\u0000-\\u001F\\u007F]+"
    }
  }
},
"Transform": {
  "type": "object",
  "additionalProperties": false,
  "required": [
    "expression",
    "variables"
  ],
  "properties": {
    "expression": {
      "description": "The mathematical expression that defines the transformation
function.",

```

```
    "type": "string",
    "minLength": 1,
    "maxLength": 1024
  },
  "variables": {
    "description": "The list of variables used in the expression.",
    "type": "array",
    "items": {
      "$ref": "#/definitions/ExpressionVariable"
    }
  },
  "processingConfig": {
    "$ref": "#/definitions/TransformProcessingConfig"
  }
}
},
"TransformProcessingConfig": {
  "description": "The processing configuration for the given transform property.",
  "type": "object",
  "additionalProperties": false,
  "required": [
    "computeLocation"
  ],
  "properties": {
    "computeLocation": {
      "description": "The compute location for the given transform property.",
      "$ref": "#/definitions/ComputeLocation"
    },
    "forwardingConfig": {
      "description": "The forwarding configuration for a given property.",
      "$ref": "#/definitions/ForwardingConfig"
    }
  }
}
},
"Metric": {
  "type": "object",
  "additionalProperties": false,
  "required": [
    "expression",
    "variables",
    "window"
  ],
  "properties": {
    "expression": {
```

```
    "description": "The mathematical expression that defines the metric
aggregation function.",
    "type": "string",
    "minLength": 1,
    "maxLength": 1024
  },
  "variables": {
    "description": "The list of variables used in the expression.",
    "type": "array",
    "items": {
      "$ref": "#/definitions/ExpressionVariable"
    }
  },
  "window": {
    "description": "The window (time interval) over which AWS IoT SiteWise
computes the metric's aggregation expression",
    "$ref": "#/definitions/MetricWindow"
  },
  "processingConfig": {
    "$ref": "#/definitions/MetricProcessingConfig"
  }
}
},
"MetricProcessingConfig": {
  "description": "The processing configuration for the metric.",
  "type": "object",
  "additionalProperties": false,
  "required": [
    "computeLocation"
  ],
  "properties": {
    "computeLocation": {
      "description": "The compute location for the given metric property.",
      "$ref": "#/definitions/ComputeLocation"
    }
  }
},
"ComputeLocation": {
  "type": "string",
  "enum": [
    "EDGE",
    "CLOUD"
  ]
},
```

```
"ForwardingConfig": {
  "type": "object",
  "additionalProperties": false,
  "required": [
    "state"
  ],
  "properties": {
    "state": {
      "type": "string",
      "enum": [
        "ENABLED",
        "DISABLED"
      ]
    }
  }
},
"MetricWindow": {
  "description": "Contains a time interval window used for data aggregate
computations (for example, average, sum, count, and so on).",
  "type": "object",
  "additionalProperties": false,
  "properties": {
    "tumbling": {
      "description": "The tumbling time interval window.",
      "type": "object",
      "additionalProperties": false,
      "required": [
        "interval"
      ],
      "properties": {
        "interval": {
          "description": "The time interval for the tumbling window.",
          "type": "string",
          "minLength": 2,
          "maxLength": 23
        },
        "offset": {
          "description": "The offset for the tumbling window.",
          "type": "string",
          "minLength": 2,
          "maxLength": 25
        }
      }
    }
  }
}
```

```

    }
  },
  "ExpressionVariable": {
    "type": "object",
    "additionalProperties": false,
    "required": [
      "name",
      "value"
    ],
    "properties": {
      "name": {
        "description": "The friendly name of the variable to be used in the
expression.",
        "type": "string",
        "minLength": 1,
        "maxLength": 64,
        "pattern": "^[a-z][a-z0-9_]*$"
      },
      "value": {
        "description": "The variable that identifies an asset property from which to
use values.",
        "$ref": "#/definitions/VariableValue"
      }
    }
  },
  "VariableValue": {
    "type": "object",
    "additionalProperties": false,
    "anyOf": [
      {
        "required": [
          "propertyId"
        ]
      },
      {
        "required": [
          "propertyExternalId"
        ]
      }
    ],
    "properties": {
      "propertyId": {
        "$ref": "#/definitions/ID"
      }
    }
  },

```

```
    "propertyExternalId": {
      "$ref": "#/definitions/ExternalId"
    },
    "hierarchyId": {
      "$ref": "#/definitions/ID"
    },
    "hierarchyExternalId": {
      "$ref": "#/definitions/ExternalId"
    }
  }
},
"Measurement": {
  "type": "object",
  "additionalProperties": false,
  "properties": {
    "processingConfig": {
      "$ref": "#/definitions/MeasurementProcessingConfig"
    }
  }
},
"MeasurementProcessingConfig": {
  "type": "object",
  "additionalProperties": false,
  "required": [
    "forwardingConfig"
  ],
  "properties": {
    "forwardingConfig": {
      "description": "The forwarding configuration for the given measurement
property.",
      "$ref": "#/definitions/ForwardingConfig"
    }
  }
},
"AssetModelHierarchy": {
  "description": "Contains information about an asset model hierarchy.",
  "type": "object",
  "additionalProperties": false,
  "anyOf": [
    {
      "required": [
        "id",
        "childAssetModelId"
      ]
    }
  ]
}
```

```

    },
    {
      "required": [
        "id",
        "childAssetModelExternalId"
      ]
    },
    {
      "required": [
        "externalId",
        "childAssetModelId"
      ]
    },
    {
      "required": [
        "externalId",
        "childAssetModelExternalId"
      ]
    }
  ],
  "required": [
    "name"
  ],
  "properties": {
    "id": {
      "description": "The ID of the asset model hierarchy.",
      "$ref": "#/definitions/ID"
    },
    "externalId": {
      "description": "The ExternalID of the asset model hierarchy.",
      "$ref": "#/definitions/ExternalId"
    },
    "name": {
      "description": "The name of the asset model hierarchy.",
      "$ref": "#/definitions/Name"
    },
    "childAssetModelId": {
      "description": "The ID of the asset model. All assets in this hierarchy must be instances of the child AssetModelId asset model.",
      "$ref": "#/definitions/ID"
    },
    "childAssetModelExternalId": {
      "description": "The ExternalID of the asset model. All assets in this hierarchy must be instances of the child AssetModelId asset model.",

```

```

    "$ref": "#/definitions/ExternalId"
  }
}
},
"AssetModel": {
  "type": "object",
  "additionalProperties": false,
  "anyOf": [
    {
      "required": [
        "assetModelId"
      ]
    },
    {
      "required": [
        "assetModelExternalId"
      ]
    }
  ],
  "required": [
    "assetModelName"
  ],
  "properties": {
    "assetModelId": {
      "description": "The ID of the asset model.",
      "$ref": "#/definitions/ID"
    },
    "assetModelExternalId": {
      "description": "The ID of the asset model.",
      "$ref": "#/definitions/ExternalId"
    },
    "assetModelName": {
      "description": "A unique, friendly name for the asset model.",
      "$ref": "#/definitions/Name"
    },
    "assetModelDescription": {
      "description": "A description for the asset model.",
      "$ref": "#/definitions/Description"
    },
    "assetModelType": {
      "description": "The type of the asset model.",
      "$ref": "#/definitions/AssetModelType"
    },
    "assetModelProperties": {

```

```

    "description": "The property definitions of the asset model.",
    "type": "array",
    "items": {
      "$ref": "#/definitions/AssetModelProperty"
    }
  },
  "assetModelCompositeModels": {
    "description": "The composite asset models that are part of this asset model. Composite asset models are asset models that contain specific properties.",
    "type": "array",
    "items": {
      "$ref": "#/definitions/AssetModelCompositeModel"
    }
  },
  "assetModelHierarchies": {
    "description": "The hierarchy definitions of the asset model. Each hierarchy specifies an asset model whose assets can be children of any other assets created from this asset model.",
    "type": "array",
    "items": {
      "$ref": "#/definitions/AssetModelHierarchy"
    }
  },
  "tags": {
    "description": "A list of key-value pairs that contain metadata for the asset model.",
    "type": "array",
    "items": {
      "$ref": "#/definitions/Tag"
    }
  }
},
"Asset": {
  "type": "object",
  "additionalProperties": false,
  "anyOf": [
    {
      "required": [
        "assetId",
        "assetModelId"
      ]
    }
  ],
}
{

```

```

    "required": [
      "assetExternalId",
      "assetModelId"
    ]
  },
  {
    "required": [
      "assetId",
      "assetModelExternalId"
    ]
  },
  {
    "required": [
      "assetExternalId",
      "assetModelExternalId"
    ]
  }
],
"required": [
  "assetName"
],
"properties": {
  "assetId": {
    "description": "The ID of the asset",
    "$ref": "#/definitions/ID"
  },
  "assetExternalId": {
    "description": "The external ID of the asset",
    "$ref": "#/definitions/ExternalId"
  },
  "assetModelId": {
    "description": "The ID of the asset model from which to create the asset.",
    "$ref": "#/definitions/ID"
  },
  "assetModelExternalId": {
    "description": "The ExternalID of the asset model from which to create the
asset.",
    "$ref": "#/definitions/ExternalId"
  },
  "assetName": {
    "description": "A unique, friendly name for the asset.",
    "$ref": "#/definitions/Name"
  },
  "assetDescription": {

```

```
    "description": "A description for the asset",
    "$ref": "#/definitions/Description"
  },
  "assetProperties": {
    "type": "array",
    "items": {
      "$ref": "#/definitions/AssetProperty"
    }
  },
  "assetHierarchies": {
    "type": "array",
    "items": {
      "$ref": "#/definitions/AssetHierarchy"
    }
  },
  "tags": {
    "description": "A list of key-value pairs that contain metadata for the
asset.",
    "type": "array",
    "uniqueItems": false,
    "items": {
      "$ref": "#/definitions/Tag"
    }
  }
}
},
"additionalProperties": false,
"properties": {
  "assetModels": {
    "type": "array",
    "uniqueItems": false,
    "items": {
      "$ref": "#/definitions/AssetModel"
    }
  },
  "assets": {
    "type": "array",
    "uniqueItems": false,
    "items": {
      "$ref": "#/definitions/Asset"
    }
  }
}
}
```

}

Supervise los datos con las alarmas integradas AWS IoT SiteWise

Puede configurar alarmas para que sus datos le alerten a usted y a sus colaboradores cuando su equipo o los procesos no funcionen de manera óptima. Un rendimiento óptimo de una máquina o de un proceso significa que los valores de determinadas métricas deben estar dentro de un rango de límites altos y bajos. Cuando estas métricas están fuera de su rango de operación, se debe notificar a los operadores de los equipos para que solucionen el problema. Use alarmas para identificar rápidamente los problemas y notificar a los operadores para que maximicen el rendimiento del equipo y de los procesos.

Temas

- [Tipos de Alarmas](#)
- [Estados de alarma](#)
- [Propiedades del estado de alarma](#)
- [Defina las alarmas en los modelos de activos en AWS IoT SiteWise](#)
- [Configure las alarmas de los activos en AWS IoT SiteWise](#)
- [Responda a las alarmas en AWS IoT SiteWise](#)
- [Introduzca un estado de alarma externo en AWS IoT SiteWise](#)

Tipos de Alarmas

Puede definir las alarmas que detecte en la AWS nube y las alarmas que detecte con procesos externos. AWS IoT SiteWise admite los siguientes tipos de alarmas:

- AWS IoT Events alarmas

Note


Aviso de fin de soporte: se AWS ha finalizado el soporte para AWS IoT Events. Para obtener más información, consulte [AWS IoT Events el fin del soporte](#).

AWS IoT Events las alarmas son alarmas que detectan la entrada AWS IoT Events. AWS IoT SiteWise envía los valores de las propiedades de los activos a un modelo de alarma en AWS IoT Events. A continuación, AWS IoT Events envía el estado de alarma a AWS IoT SiteWise. Puede configurar opciones, como cuándo detecta algo la alarma y a quién notificar cuando cambie el estado de alarma. También puede definir las [acciones de AWS IoT Events](#) que se deben producirse cuando cambie el estado de alarma.

Las alarmas AWS IoT Events son instancias de modelos de alarma. El modelo de alarma especifica el umbral y la gravedad de la alarma, qué hacer cuando cambie el estado de alarma y más. Al configurar cada característica del modelo de alarma, se especifica una propiedad de atributo a partir del modelo de activo que monitorea la alarma. Todos los activos basados en el modelo de activos utilizan el valor del atributo al AWS IoT Events evaluar esa característica de la alarma. Para obtener más información, consulte [Uso de alarmas](#) en la Guía para desarrolladores de AWS IoT Events .

Puede responder a una AWS IoT Events alarma cuando cambia de estado. Por ejemplo, puede reconocer o posponer una alarma cuando se activa. También puede habilitar, deshabilitar y restablecer las alarmas.

SiteWise Los usuarios de Monitor pueden visualizar, configurar y responder a AWS IoT Events las alarmas en los portales de SiteWise Monitor. Para obtener más información, consulte [Supervisión con alarmas](#) en la Guía de la aplicación AWS IoT SiteWise Monitor .

 Note

AWS IoT Events se aplican cargos para evaluar estas alarmas y transferir datos entre AWS IoT SiteWise y AWS IoT Events. Para más información, consulte [Precios de AWS IoT Events](#).

- Alarmas externas

Las alarmas externas son alarmas que se evalúan fuera de ellas AWS IoT SiteWise. Utilice alarmas externas si tiene un origen de datos que notifique estados de alarma. La alarma externa contiene una propiedad de medición a la que se ingieren los datos del estado de alarma.

No se puede reconocer ni posponer una alarma externa cuando cambia de estado.

SiteWise Los usuarios de Monitor pueden ver el estado de las alarmas externas en los portales de SiteWise Monitor, pero no pueden configurarlas ni responder a ellas.

AWS IoT SiteWise no evalúa el estado de las alarmas externas.

Estados de alarma

Las alarmas industriales incluyen información sobre el estado del equipo o proceso que monitorean, e información (opcional) sobre la respuesta del operador al estado de alarma.

Al definir una AWS IoT Events alarma, se especifica si se habilita o no el flujo de confirmación. El flujo de reconocimiento está habilitado de forma predeterminada. Al habilitar esta opción, los operadores pueden reconocer la alarma y dejar una nota con detalles sobre la alarma o las acciones que han tomado para solucionarla. Si un operador no confirma una alarma activa antes de que se inactive, la alarma se queda bloqueada. El estado bloqueado indica que la alarma se ha activado pero no se ha confirmado, por lo que el operador debe comprobar el equipo o el proceso y reconocer la alarma bloqueada.

Las alarmas tienen los siguientes estados:

- Normal (Normal): la alarma está habilitada pero inactiva. El proceso o equipo industrial funciona según lo esperado.
- Activa (Active): la alarma está activa. El proceso o equipo industrial está fuera de su rango de operación y requiere atención.
- Confirmado (Acknowledged): un operador ha confirmado el estado de la alarma.

Este estado solo se aplica a las alarmas en las que se habilita el flujo de reconocimiento.

- Bloqueado (Latched): la alarma ha vuelto a la normalidad, pero estaba activa y ningún operador la ha confirmado. El proceso o equipo industrial requiere la atención de un operador para restablecer la alarma a su estado normal.

Este estado solo se aplica a las alarmas en las que se habilita el flujo de reconocimiento.

- Silenciado (SnoozeDisabled): la alarma está desactivada porque un operador la ha pospuesto. El operador define el tiempo durante el que se pospone la alarma. Transcurrido ese tiempo, la alarma vuelve al estado normal.
- Deshabilitada (Disabled): la alarma está deshabilitada y no es capaz de detectar nada.

Propiedades del estado de alarma

AWS IoT SiteWise almacena los datos del estado de la alarma como un objeto JSON serializado en una cadena. Este objeto contiene el estado e información adicional sobre la alarma, como las acciones de respuesta del operador y la regla que evalúa la alarma.

La propiedad del estado de alarma se identifica por su nombre y tipo de estructura, `AWS/ALARM_STATE`. Para obtener más información, consulte [Defina las alarmas en los modelos de activos en AWS IoT SiteWise](#).

El objeto datos de estado de la alarma contiene la siguiente información:

`stateName`

El estado de la alarma. Para obtener más información, consulte [Estados de alarma](#).

Tipo de datos: STRING

`customerAction`

(opcional) objeto que contiene información sobre la respuesta de un operador a la alarma. Los operadores pueden habilitar, deshabilitar, confirmar y posponer las alarmas. Cuando lo hacen, los datos del estado de alarma incluyen su respuesta y la nota que pueden dejar al responder. Este objeto contiene la siguiente información:

`actionName`

El nombre de la acción que realiza el operador para responder a la alarma. Este valor contiene uno de los siguientes strings:

- ENABLE
- DISABLE
- SNOOZE
- ACKNOWLEDGE
- RESET

Tipo de datos: STRING

`enable`

(opcional) objeto que está presente en `customerAction` cuando el operador habilita la alarma. Cuando un operador habilita la alarma, el estado de alarma cambia a `Normal`. Este objeto contiene la siguiente información:

note

(opcional) la nota que deja el cliente cuando habilita la alarma.

Tipo de datos: STRING

Longitud máxima: 128 caracteres

disable

(opcional) objeto que está presente en `customerAction` cuando el operador deshabilita la alarma. Cuando un operador habilita la alarma, el estado de alarma cambia a `Disabled`. Este objeto contiene la siguiente información:

note

(opcional) la nota que deja el cliente cuando deshabilita la alarma.

Tipo de datos: STRING

Longitud máxima: 128 caracteres

acknowledge

(opcional) objeto que está presente en `customerAction` cuando el operador confirma la alarma. Cuando un operador habilita la alarma, el estado de alarma cambia a `Acknowledged`. Este objeto contiene la siguiente información:

note

(opcional) la nota que deja el cliente cuando confirma la alarma.

Tipo de datos: STRING

Longitud máxima: 128 caracteres

snooze

(opcional) objeto que está presente en `customerAction` cuando el operador pospone la alarma. Cuando un operador habilita la alarma, el estado de alarma cambia a `SnoozeDisabled`. Este objeto contiene la siguiente información:

snoozeDuration

El tiempo en segundos durante el que el operador pospone la alarma. La alarma cambia al estado `Normal` pasado este tiempo.

Tipo de datos: INTEGER

`note`

(opcional) la nota que deja el cliente al posponer la alarma.

Tipo de datos: STRING

Longitud máxima: 128 caracteres

`ruleEvaluation`

(opcional) objeto que contiene información sobre la regla que evalúa la alarma. Este objeto contiene la siguiente información:

`simpleRule`

objeto que contiene información acerca de una regla simple, que compara el valor de la propiedad frente a un valor de umbral con un operador de comparación. Este objeto contiene la siguiente información:

`inputProperty`

El valor de la propiedad que evalúa esta alarma.

Tipo de datos: DOUBLE

`operator`

El operador de comparación que utiliza esta alarma para comparar la propiedad con el umbral. Este valor contiene uno de los siguientes strings:

- `<`: menor que
- `<=`: menor que o igual a
- `==`: igual
- `!=`: distinto de
- `>=`: mayor que o igual a
- `>`: mayor que

Tipo de datos: STRING

`threshold`

El valor de umbral frente al que esta alarma compara el valor de la propiedad.

Tipo de datos: DOUBLE

Defina las alarmas en los modelos de activos en AWS IoT SiteWise

Los modelos de activos impulsan la normalización de sus datos y alarmas industriales. Puede establecer definiciones de alarmas en los modelos de activos, para estandarizar las alarmas de todos los activos basados en un modelo de activo.

Puede utilizar modelos de activos compuestos para definir alarmas en este modelo de activos. Los modelos de activos compuestos son modelos de activos que estandarizan un conjunto específico de propiedades en otro modelo de activo. Los modelos de activos compuestos garantizan la inclusión de determinadas propiedades en un modelo de activo. Las alarmas tienen propiedades de tipo, estado y origen (opcional), por lo que el modelo compuesto de alarmas exige que existan estas propiedades.

Cada modelo de activo compuesto tiene un tipo que define las propiedades que admite el modelo compuesto. Los modelos compuestos de alarmas definen las propiedades de tipo de alarma, estado de alarma y origen de alarma (opcional). Al crear un activo a partir de un modelo de activo con modelos compuestos, el activo incluye las propiedades del modelo compuesto junto con las propiedades que especifique en el modelo de activo.

Cada propiedad de un modelo compuesto debe tener el nombre que la identifique para su tipo de modelo compuesto. Las propiedades del modelo compuesto admiten propiedades con tipos de datos complejos. Estas propiedades tienen el tipo de datos STRUCT y una característica `dataTypeSpec` que especifica el tipo de datos complejo de la propiedad. Las propiedades de los tipos de datos complejos contienen datos JSON serializados en cadenas.

Los modelos compuestos de alarmas tienen las siguientes propiedades. Cada propiedad debe tener el nombre que la identifique para su tipo de modelo compuesto.

Tipo de alarma

El tipo de alarma. Especifique uno de los siguientes valores:

- `IOT_EVENTS`— Una AWS IoT Events alarma. AWS IoT SiteWise envía datos AWS IoT Events para evaluar el estado de esta alarma. Debe especificar la propiedad de la fuente de la alarma para definir el modelo de AWS IoT Events alarma para esta definición de alarma.

Note

Aviso de fin de soporte: soporte AWS finalizado para AWS IoT Events. Para obtener más información, consulte [AWS IoT Events el fin del soporte](#).

- **EXTERNAL**: una alarma externa. El estado de la alarma se ingiere en forma de medida.

Nombre de la propiedad: AWS/ALARM_TYPE

Tipo de propiedad: [atributo](#)

Tipo de datos: STRING

Estado de alarma

Los datos de serie temporal del estado de la alarma. Se trata de un objeto serializado en una cadena que contiene el estado y otra información sobre la alarma. Para obtener más información, consulte [Propiedades del estado de alarma](#).

Nombre de la propiedad: AWS/ALARM_STATE

Tipo de propiedad: [medida](#)

Tipo de datos: STRUCT

Tipo de estructura de datos: AWS/ALARM_STATE

Origen de alarma

(opcional) el nombre de recurso de Amazon (ARN) del recurso que evalúa el estado de la alarma. En el AWS IoT Events caso de las alarmas, este es el ARN del modelo de alarma.

Nombre de la propiedad: AWS/ALARM_SOURCE

Tipo de propiedad: [atributo](#)

Tipo de datos: STRING

Example Ejemplo de modelo compuesto de alarma

El siguiente modelo de activos representa una caldera que tiene una alarma para controlar su temperatura. AWS IoT SiteWise envía los datos de temperatura AWS IoT Events para detectar la alarma.

```
{
  "assetModelName": "Boiler",
  "assetModelDescription": "A boiler that alarms when its temperature exceeds its
limit.",
  "assetModelProperties": [
    {
```

```

    "name": "Temperature",
    "dataType": "DOUBLE",
    "unit": "Celsius",
    "type": {
      "measurement": {}
    }
  },
  {
    "name": "High Temperature",
    "dataType": "DOUBLE",
    "unit": "Celsius",
    "type": {
      "attribute": {
        "defaultValue": "105.0"
      }
    }
  }
],
"assetModelCompositeModels": [
  {
    "name": "BoilerTemperatureHighAlarm",
    "type": "AWS/ALARM",
    "properties": [
      {
        "name": "AWS/ALARM_TYPE",
        "dataType": "STRING",
        "type": {
          "attribute": {
            "defaultValue": "IOT_EVENTS"
          }
        }
      },
      {
        "name": "AWS/ALARM_STATE",
        "dataType": "STRUCT",
        "dataTypeSpec": "AWS/ALARM_STATE",
        "type": {
          "measurement": {}
        }
      },
      {
        "name": "AWS/ALARM_SOURCE",
        "dataType": "STRING",
        "type": {

```

```
        "attribute": {}
      }
    ]
  }
]
```

Temas

- [Requisitos para las notificaciones de alarma en AWS IoT SiteWise](#)
- [Defina AWS IoT Events alarmas para AWS IoT SiteWise](#)
- [Defina las alarmas externas en AWS IoT SiteWise](#)

Requisitos para las notificaciones de alarma en AWS IoT SiteWise

AWS IoT Events utiliza una AWS Lambda función de su AWS cuenta para enviar notificaciones de alarma. Debe crear esta función Lambda en la misma AWS región que sus alarmas para activar las notificaciones de alarma. Esta función de Lambda utiliza [Amazon Simple Notification Service \(Amazon SNS\)](#) para enviar notificaciones de texto y [Amazon Simple Email Service \(Amazon SES\)](#) para enviar notificaciones por correo electrónico. Al crear la AWS IoT Events alarma, se configuran los protocolos y los ajustes que utiliza la alarma para enviar las notificaciones.

Note

Aviso de fin de soporte: se AWS ha finalizado el soporte para AWS IoT Events. Para obtener más información, consulte [AWS IoT Events el fin del soporte](#).

AWS IoT Events proporciona una plantilla de AWS CloudFormation pila que puede utilizar para crear esta función Lambda en su cuenta. Para obtener más información, consulte [Notificación de alarma de la función de Lambda](#) en la Guía para desarrolladores de AWS IoT Events .

Defina AWS IoT Events alarmas para AWS IoT SiteWise

Note

Aviso de fin de soporte: se AWS ha finalizado el soporte para AWS IoT Events. Para obtener más información, consulte [AWS IoT Events el fin del soporte](#).

Al crear una AWS IoT Events alarma, AWS IoT SiteWise envía los valores de las propiedades de los activos AWS IoT Events para evaluar el estado de la alarma. AWS IoT Events las definiciones de alarma dependen del modelo de alarma en el que se defina AWS IoT Events. Para definir una AWS IoT Events alarma en un modelo de activos, defina un modelo compuesto de alarmas que especifique el modelo de AWS IoT Events alarma como su propiedad de origen de alarma.

AWS IoT Events las alarmas dependen de entradas como los umbrales de alarma y la configuración de las notificaciones de alarmas. Estas entradas se definen como atributos en el modelo de activo. A continuación, puede personalizar estas entradas en cada activo basado en el modelo. La AWS IoT SiteWise consola puede crear estos atributos por usted. Si define las alarmas con la API AWS CLI o la API, debe definir estos atributos manualmente en el modelo de activos.

También puede definir otras acciones, como las acciones de notificación de alarma personalizadas, que deben producirse cuando la alarma detecte algo. Por ejemplo, puede configurar una acción que envíe una notificación push a un tema de Amazon SNS. Para obtener más información sobre las acciones que puede definir, consulte [Trabajar con otros AWS servicios](#) en la Guía para AWS IoT Events desarrolladores.

Al actualizar o eliminar un modelo de activos, AWS IoT SiteWise puede comprobar si un modelo de alarma AWS IoT Events monitorea una propiedad de activo asociada a este modelo de activos. Esto le impide eliminar una propiedad de activo que una AWS IoT Events alarma esté utilizando actualmente. Para activar esta función AWS IoT SiteWise, debe tener el `iotevents:ListInputRoutings` permiso. Este permiso permite AWS IoT SiteWise realizar llamadas a la operación de [ListInputRoutings](#) API admitida por AWS IoT Events. Para obtener más información, consulte [ListInputRoutings Permiso \(opcional\)](#).

Note

La característica de notificaciones de alarma no está disponible en la región de China (Pekín).

Temas

- [Defina una AWS IoT Events alarma \(AWS IoT SiteWise consola\)](#)
- [Defina una AWS IoT Events alarma \(AWS IoT Events consola\)](#)
- [Defina una AWS IoT Events alarma \(AWS CLI\)](#)

Defina una AWS IoT Events alarma (AWS IoT SiteWise consola)

Puede utilizar la AWS IoT SiteWise consola para definir una AWS IoT Events alarma en un modelo de activos existente. Para definir una AWS IoT Events alarma en un nuevo modelo de activos, cree el modelo de activos y, a continuación, complete estos pasos. Para obtener más información, consulte [Cree modelos de activos en AWS IoT SiteWise](#).

Important


Cada alarma requiere un atributo que especifique el valor de umbral con el que comparar la alarma. Debe definir el atributo de valor de umbral en el modelo de activo antes de poder definir una alarma.

Considere un ejemplo en el que desee definir una alarma que detecte cuando una turbina eólica supere su índice de velocidad máxima del viento de 80 km/h. Antes de definir la alarma, debe definir un atributo (Velocidad máxima del viento) con un valor predeterminado de 50.

Para definir una AWS IoT Events alarma en un modelo de activos


1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, elija Models (Modelos).
3. Elija el modelo de activo para el que desee definir una alarma.
4. Seleccione la pestaña Alarma.
5. Seleccione Añadir alarma.
6. En la sección Opciones de tipos de alarma, seleccione Alarma de AWS IoT Events .
7. En la sección Detalles de regla haga lo siguiente:
 - a. Escriba un nombre para la alarma.
 - b. (Opcional) Ingrese una descripción para su alarma.

8. En la sección Definiciones de umbral se define cuándo debe detectar algo la alarma y la gravedad de la alarma. Haga lo siguiente:
 - a. Seleccione la Propiedad sobre la que debe detectar la alarma. Cada vez que esta propiedad recibe un nuevo valor, AWS IoT SiteWise envía el valor AWS IoT Events a para evaluar el estado de la alarma.
 - b. Seleccione el Operador que se utilizará para comparar la propiedad con el valor de umbral. Puede elegir entre las siguientes opciones:
 - < Menor que
 - <= Menor que o igual a
 - == (igual)
 - != Distinto de
 - >= Mayor que o igual a
 - > Mayor que
 - c. En Valor, seleccione la propiedad del atributo que desee utilizar como valor de umbral. AWS IoT Events compara el valor de la propiedad con el valor de este atributo.
 - d. Introduzca la Gravedad de la alarma. Use un número que pueda comprender su equipo para reflejar la gravedad de esta alarma.
9. (Opcional) En la sección Configuración de notificaciones: opcional, haga lo siguiente:
 - a. Elija Activar.

 Note

Si elige Inactivo, ni usted ni su equipo recibirán ninguna notificación de alarma.


- b. En Destinatario, elija el destinatario.

 Important

Puede enviar notificaciones de alarma a AWS IAM Identity Center los usuarios. Para utilizar esta característica, debe habilitar IAM Identity Center. Solo puede activar el IAM Identity Center en una AWS región a la vez. Esto significa que solo puede definir las notificaciones de alarma en la región en la que habilite IAM Identity


Center. Para obtener más información, consulte la [Introducción](#) de la Guía del usuario de AWS IAM Identity Center .

- c. En Protocolo elija una de las siguientes opciones:
- Correo electrónico y texto: la alarma notifica a los usuarios del IAM Identity Center con un mensaje SMS y un mensaje de correo electrónico.
 - Correo electrónico: la alarma notifica a los usuarios del IAM Identity Center con un mensaje de correo electrónico.
 - Texto: la alarma notifica a los usuarios del IAM Identity Center con un mensaje SMS.
- d. En Remitente, elija el remitente.

 Important


Debe verificar la dirección de correo electrónico del remitente en Amazon Simple Email Service (Amazon SES). Para obtener más información, consulte [Verificación de una identidad de dirección de correo electrónico](#) en la Guía para desarrolladores de Amazon Simple Email Service.

10. En la sección Estado predeterminado del activo, puede establecer el estado predeterminado para las alarmas creadas a partir de este modelo de activos.

 Note

Podrá activar o desactivar esta alarma para los activos que cree a partir de este modelo de activo en un paso posterior.

11. En la sección de configuración avanzada, puede configurar los permisos, los ajustes de notificación adicionales, las acciones del estado de alarma, el modelo de alarma en SiteWise Monitor y el flujo de confirmación.

 Note

AWS IoT Events las alarmas requieren las siguientes funciones de servicio:

- Una función que se AWS IoT Events supone que debe enviar los valores del estado de alarma a AWS IoT SiteWise.

- Una función que AWS IoT Events asume el envío de datos a Lambda. Solo necesita esta función si su alarma envía notificaciones.

En la sección Permisos, haga lo siguiente:

- a. En Rol de AWS IoT Events , utilice un rol existente o cree uno con los permisos necesarios. Este rol requiere el permiso `iotsitewise:BatchPutAssetPropertyValue` y una relación de confianza que permita a `iotevents.amazonaws.com` asumir ese rol.
- b. Para el AWS IoT Events rol de Lambda, utilice un rol existente o cree un rol con los permisos necesarios. Este rol requiere los permisos `lambda:InvokeFunction` y `sso-directory:DescribeUser` y una relación de confianza que permita a `iotevents.amazonaws.com` asumir ese rol.

12. (Opcional) En la sección Configuración de notificación adicional, haga lo siguiente:

- a. En Atributo del destinatario, defina un atributo cuyo valor especifique el destinatario de la notificación. Puede elegir a los usuarios del IAM Identity Center como destinatarios.

Puede crear un atributo o utilizar uno existente en el modelo de activo.

- Si elige Crear un nuevo atributo de destinatario, especifique el Nombre del atributo de destinatario y el Valor predeterminado del destinatario: opcional para el atributo.
- Si elige Usar un atributo de destinatario existente, elija el atributo en Nombre del atributo del destinatario. La alarma utiliza el valor predeterminado del atributo que elija.

Puede anular el valor predeterminado de cada activo que cree desde este modelo de activo.

- b. En Atributo del mensaje personalizado, defina un atributo cuyo valor especifique el mensaje personalizado que se debe enviar además del mensaje de cambio de estado predeterminado. Por ejemplo, puede especificar un mensaje que ayude a su equipo a entender cómo abordar esta alarma.

Puede optar por crear un atributo o utilizar uno existente en el modelo de activo.

- Si opta por Crear un nuevo atributo de mensaje personalizado, especifique el Nombre de atributo del mensaje personalizado y el Valor predeterminado del mensaje personalizado: opcional para el atributo.

- Si opta por Usar un atributo de mensaje personalizado existente, elija el atributo en Nombre de atributo del mensaje personalizado. La alarma utiliza el valor predeterminado del atributo que elija.

Puede anular el valor predeterminado de cada activo que cree desde este modelo de activo.

c. En Gestionar su función de Lambda, realice alguna de las siguientes operaciones:

- Para AWS IoT SiteWise crear una nueva función de Lambda, elija Crear una nueva lambda a partir de una plantilla gestionada por AWS.
- Para usar una función de Lambda existente, elija Usar una Lambda existente y elija el nombre de la función.

Para obtener más información, consulte [Administración de notificaciones de alarma](#) en la Guía para desarrolladores de AWS IoT Events .

13. (Opcional) En la sección Configurar acción de estado, haga lo siguiente:

- a. Elija Editar acción.
- b. En Agregar acciones del estado de la alarma, añada acciones y, a continuación, seleccione Guardar.

Puede agregar hasta 10 acciones.

AWS IoT Events puede realizar acciones cuando la alarma está activa. Puede definir acciones integradas para usar un temporizador o establecer una variable, o enviar datos a otros AWS recursos. Para obtener más información, consulte [Acciones admitidas](#) en la Guía para desarrolladores de AWS IoT Events .

14. (Opcional) En Administrar el modelo de alarma en el SiteWise monitor (opcional), seleccione Activo o Inactivo.

Utilice esta opción para actualizar el modelo de alarma en SiteWise Monitorss. Esta opción está habilitada de forma predeterminada.

15. En Flujo de reconocimiento, seleccione Activo o Inactivo. Para obtener más información sobre el flujo de reconocimiento, consulte [Estados de alarma](#).


16. Seleccione Añadir alarma.

 Note

La AWS IoT SiteWise consola realiza varias solicitudes de API para añadir la alarma al modelo de activos. Al elegir Agregar alarma, la consola abre un cuadro de diálogo que muestra el progreso de estas solicitudes de API. Permanezca en esta página hasta que cada solicitud de API se realice correctamente o hasta que falle una solicitud de API. Si se produce un error en una solicitud, cierre el cuadro de diálogo, corrija el problema y seleccione Agregar alarma para volver a intentarlo.

Defina una AWS IoT Events alarma (AWS IoT Events consola)

Puede utilizar la AWS IoT Events consola para definir una AWS IoT Events alarma en un modelo de activos existente. Para definir una AWS IoT Events alarma en un nuevo modelo de activos, cree el modelo de activos y, a continuación, complete estos pasos. Para obtener más información, consulte [Cree modelos de activos en AWS IoT SiteWise](#).

 Important

Cada alarma requiere un atributo que especifique el valor de umbral con el que comparar la alarma. Debe definir el atributo de valor de umbral en el modelo de activo antes de poder definir una alarma.

Considere un ejemplo en el que desee definir una alarma que detecte cuando una turbina eólica supere su índice de velocidad máxima del viento de 80 km/h. Antes de definir la alarma, debe definir un atributo (Velocidad máxima del viento) con un valor predeterminado de 50.

Para definir una AWS IoT Events alarma en un modelo de activos

1. Vaya a la [consola de AWS IoT Events](#).
2. En el panel de navegación, elija Modelos de alarmas.
3. Elija Crear modelo de alarma.
4. Escriba un nombre para la alarma.
5. (Opcional) Ingrese una descripción para su alarma.
6. En la sección Destino de la alarma, haga lo siguiente:

- a. En Opciones de destino, elija la propiedad del activo de AWS IoT SiteWise .
 - b. Elija el ,modelo de activo para el que desea añadir una alarma.
7. En la sección Definiciones de umbral se define cuándo debe detectar algo la alarma y la gravedad de la alarma. Haga lo siguiente:
- a. Seleccione la Propiedad sobre la que debe detectar la alarma. Cada vez que esta propiedad recibe un nuevo valor, AWS IoT SiteWise envía el valor AWS IoT Events a para evaluar el estado de la alarma.
 - b. Seleccione el Operador que se utilizará para comparar la propiedad con el valor de umbral. Puede elegir entre las siguientes opciones:
 - < Menor que
 - <= Menor que o igual a
 - == (igual)
 - != Distinto de
 - >= Mayor que o igual a
 - > Mayor que
 - c. En Valor, seleccione la propiedad del atributo que desee utilizar como valor de umbral. AWS IoT Events compara el valor de la propiedad con el valor de este atributo.
 - d. Introduzca la Gravedad de la alarma. Use un número que pueda comprender su equipo para reflejar la gravedad de esta alarma.
8. (Opcional) En la sección Configuración de notificaciones: opcional, haga lo siguiente:
- a. En Protocolo elija una de las siguientes opciones:
 - Correo electrónico y texto: la alarma notifica a los usuarios del IAM Identity Center con un mensaje SMS y un mensaje de correo electrónico.
 - Correo electrónico: la alarma notifica a los usuarios del IAM Identity Center con un mensaje de correo electrónico.
 - Texto: la alarma notifica a los usuarios del IAM Identity Center con un mensaje SMS.
 - b. En Remitente, elija el remitente.

⚠ Important

Debe verificar la dirección de correo electrónico del remitente en Amazon Simple Email Service (Amazon SES). Para obtener más información, consulte [Verificación de direcciones de correo electrónico en Amazon SES](#), en la Guía para desarrolladores de Amazon Simple Email Service.

- c. Elija el atributo en Atributo del destinatario: (opcional). La alarma utiliza el valor predeterminado del atributo que elija.
 - d. Elija el atributo en Atributo del mensaje personalizado: (opcional). La alarma utiliza el valor predeterminado del atributo que elija.
9. En la sección Instancia, especifique el Estado predeterminado de esta alarma. Podrá activar o desactivar esta alarma para todos los activos que cree a partir de este modelo de activo en un paso posterior.
10. En los ajustes avanzados, puede configurar los permisos, los ajustes de notificación adicionales, las acciones del estado de alarma, el modelo de alarma en SiteWise Monitor y el flujo de confirmación.

ℹ Note

AWS IoT Events las alarmas requieren las siguientes funciones de servicio:

- Una función que se AWS IoT Events supone que debe enviar los valores del estado de alarma a AWS IoT SiteWise.
- Una función que AWS IoT Events asume el envío de datos a Lambda. Solo necesita esta función si su alarma envía notificaciones.

- a. En la sección Flujo de reconocimiento, elija Habilitado o Deshabilitado. Para obtener más información sobre el flujo de reconocimiento, consulte [Estados de alarma](#).
- b. En la sección Permisos, haga lo siguiente:
 - i. En Rol de AWS IoT Events , utilice un rol existente o cree uno con los permisos necesarios. Este rol requiere el permiso `iotsitewise:BatchPutAssetPropertyValue` y una relación de confianza que permita a `iotevents.amazonaws.com` asumir ese rol.

- ii. Para el rol de Lambda, utilice un rol existente o cree un rol con los permisos necesarios. Este rol requiere los permisos `lambda:InvokeFunction` y `sso-directory:DescribeUser` y una relación de confianza que permita a `iotevents.amazonaws.com` asumir ese rol.
- c. (Opcional) En el panel de Configuración de notificación adicional, haga lo siguiente:
 - En Gestionar su Función de Lambda, realice alguna de las siguientes operaciones:
 - Para AWS IoT Events crear una nueva función Lambda, elija Crear una nueva función Lambda.
 - Para usar una función de Lambda existente, elija Usar una función de Lambda existente y elija el nombre de la función.

Para obtener más información, consulte [Administración de notificaciones de alarma](#) en la Guía para desarrolladores de AWS IoT Events .

- d. (Opcional) En la sección Establecer acción de estado: opcional, haga lo siguiente:
 - En Acciones de estado de alarma, añada acciones y, a continuación, seleccione Guardar.

Puede agregar hasta 10 acciones.

AWS IoT Events puede realizar acciones cuando la alarma está activa. Puede definir acciones integradas para usar un temporizador o establecer una variable, o enviar datos a otros AWS recursos. Para obtener más información, consulte [Acciones admitidas](#) en la Guía para desarrolladores de AWS IoT Events .

11. Seleccione Crear.

Note

La AWS IoT Events consola realiza varias solicitudes de API para añadir la alarma al modelo de activos. Al elegir Agregar alarma, la consola abre un cuadro de diálogo que muestra el progreso de estas solicitudes de API. Permanezca en esta página hasta que cada solicitud de API se realice correctamente o hasta que falle una solicitud de API. Si se produce un error en una solicitud, cierre el cuadro de diálogo, corrija el problema y seleccione Agregar alarma para volver a intentarlo.

Defina una AWS IoT Events alarma (AWS CLI)

Puede usar el AWS Command Line Interface (AWS CLI) para definir una AWS IoT Events alarma que supervise la propiedad de un activo. Puede definir una alarma en un modelo de activo nuevo o existente. Después de definir la alarma en el modelo de activos, se crea una alarma AWS IoT Events y se conecta al modelo de activos. En este proceso, hará lo siguiente:

Steps

- [Paso 1: definición de una alarma en un modelo de activos](#)
- [Paso 2: Defina un modelo AWS IoT Events de alarma](#)
- [Paso 3: Habilite el flujo de datos entre y AWS IoT SiteWise AWS IoT Events](#)

Paso 1: definición de una alarma en un modelo de activos

Añada una definición de alarma y las propiedades asociadas a un modelo de activos nuevo o existente.

Para definir una alarma en un modelo de activo (CLI)

1. Cree un archivo denominado `asset-model-payload.json`. Siga los pasos de estas otras secciones para añadir los detalles de su modelo de activo al archivo, pero no envíe la solicitud para crear o actualizar el modelo de activo. En esta sección, debe añadir una definición de alarma a los detalles del modelo de activo del archivo `asset-model-payload.json`.
 - Para obtener más información acerca de cómo crear un modelo de activos, consulte [Creación de un modelo de activos \(AWS CLI\)](#).
 - Para obtener más información acerca de cómo actualizar un modelo de activo existente, consulte [Actualice un modelo de activo, un modelo de componentes o una interfaz \(AWS CLI\)](#).

Note

Su modelo de activo debe definir al menos una propiedad de activo, incluida la propiedad del activo que se va a monitorear con la alarma.

2. Añada un modelo compuesto de alarma (`assetModelCompositeModels`) al modelo de activo. Un modelo compuesto de AWS IoT Events alarmas especifica el `IOT_EVENTS` tipo y especifica


una propiedad de la fuente de la alarma. La propiedad de la fuente de alarma se añade después de crear el modelo de alarma en AWS IoT Events.

⚠ Important

El modelo compuesto de alarma debe tener el mismo nombre que el modelo de AWS IoT Events alarma que cree más adelante. Los nombres de modelo de la alarma pueden contener únicamente caracteres alfanuméricos. Especifique un nombre alfanumérico único para poder usar el mismo nombre para el modelo de alarma.

```
{
  ...
  "assetModelCompositeModels": [
    {
      "name": "BoilerTemperatureHighAlarm",
      "type": "AWS/ALARM",
      "properties": [
        {
          "name": "AWS/ALARM_TYPE",
          "dataType": "STRING",
          "type": {
            "attribute": {
              "defaultValue": "IOT_EVENTS"
            }
          }
        },
        {
          "name": "AWS/ALARM_STATE",
          "dataType": "STRUCT",
          "dataTypeSpec": "AWS/ALARM_STATE",
          "type": {
            "measurement": {}
          }
        }
      ]
    }
  ]
}
```


3. Añada un atributo de umbral de alarma al modelo de activo. Especifique el valor predeterminado que se utilizará para este umbral. Puede anular este valor predeterminado de cada activo basado en este modelo.

 Note

El atributo de umbral de alarma debe ser un INTEGER o un DOUBLE.

```
{
  ...
  "assetModelProperties": [
    ...
    {
      "name": "Temperature Max Threshold",
      "dataType": "DOUBLE",
      "type": {
        "attribute": {
          "defaultValue": "105.0"
        }
      }
    }
  ]
}
```

4. (Opcional) Añada atributos de notificaciones de alarma al modelo de activo. Estos atributos especifican el destinatario del IAM Identity Center y otras entradas que se AWS IoT Events utilizan para enviar notificaciones cuando la alarma cambia de estado. Puede anular estos valores predeterminados en cada activo basado en este modelo.

 Important

Puede enviar notificaciones de alarma a AWS IAM Identity Center los usuarios. Para utilizar esta característica, debe habilitar IAM Identity Center. Solo puede activar el IAM Identity Center en una AWS región a la vez. Esto significa que solo puede definir las notificaciones de alarma en la región en la que habilite IAM Identity Center. Para obtener más información, consulte la [Introducción](#) de la Guía del usuario de AWS IAM Identity Center .

Haga lo siguiente:

- a. Añada un atributo que especifique el ID de su almacén de identidades de IAM Identity Center. Puede utilizar la operación de la [ListInstances](#) API del IAM Identity Center para enumerar sus almacenes de identidades. Esta operación solo funciona en la región en la que se habilite IAM Identity Center.

```
aws sso-admin list-instances
```

A continuación, especifique el ID del almacén de identidades (por ejemplo, d-123EXAMPLE) como valor predeterminado para el atributo.

```
{
  ...
  "assetModelProperties": [
    ...
    {
      "name": "identityStoreId",
      "dataType": "STRING",
      "type": {
        "attribute": {
          "defaultValue": "d-123EXAMPLE"
        }
      }
    }
  ]
}
```

- b. Añada un atributo que especifique el ID del usuario de IAM Identity Center que recibe notificaciones. Para definir un destinatario de notificación predeterminado, añada un ID de usuario de IAM Identity Center como valor predeterminado. Realice una de las siguientes acciones para obtener un ID de usuario de IAM Identity Center:
 - i. Puede utilizar la [ListUsers](#) API del IAM Identity Center para obtener el ID de un usuario cuyo nombre de usuario conozca. *d-123EXAMPLE* Sustitúyalo por el ID de tu almacén de identidades y *Name* sustitúyelo por el nombre de usuario del usuario.

```
aws identitystore list-users \
  --identity-store-id d-123EXAMPLE \
```

```
--filters AttributePath=UserName,AttributeValue=Name
```

- ii. Utilice la [consola de IAM Identity Center](#) para explorar los usuarios y buscar un ID de usuario.

A continuación, especifique el ID de usuario (por ejemplo, 123EXAMPLE-a1b2c3d4-5678-90ab-cdef-33333EXAMPLE) como valor predeterminado para el atributo o defina el atributo sin un valor predeterminado.

```
{
  ...
  "assetModelProperties": [
    ...
    {
      "name": "userId",
      "dataType": "STRING",
      "type": {
        "attribute": {
          "defaultValue": "123EXAMPLE-a1b2c3d4-5678-90ab-cdef-33333EXAMPLE"
        }
      }
    }
  ]
}
```

- c. (Opcional) Añada un atributo que especifique el ID de remitente predeterminado para las notificaciones de mensajes SMS (de texto). El ID del remitente aparece como el remitente del mensaje en los mensajes que envía Amazon Simple Notification Service (Amazon SNS). Para obtener más información, consulte [Solicitar un identificador de remitente AWS End User Messaging SMS en](#) la Guía del AWS End User Messaging SMS usuario.

```
{
  ...
  "assetModelProperties": [
    ...
    {
      "name": "senderId",
      "dataType": "STRING",
      "type": {
        "attribute": {
          "defaultValue": "MyFactory"
        }
      }
    }
  ]
}
```

```

    }
  }
}
]
}

```

- d. (Opcional) Añada un atributo que especifique la dirección de correo electrónico predeterminada que se debe utilizar como dirección de origen de en las notificaciones por correo electrónico.

```

{
  ...
  "assetModelProperties": [
    ...
    {
      "name": "fromAddress",
      "dataType": "STRING",
      "type": {
        "attribute": {
          "defaultValue": "my.factory@example.com"
        }
      }
    }
  ]
}

```

- e. (Opcional) Añada un atributo que especifique el asunto predeterminado que se debe utilizar en las notificaciones por correo electrónico.

```

{
  ...
  "assetModelProperties": [
    ...
    {
      "name": "emailSubject",
      "dataType": "STRING",
      "type": {
        "attribute": {
          "defaultValue": "[ALERT] High boiler temperature"
        }
      }
    }
  ]
}

```

```
}

```

- f. (Opcional) Añada un atributo que especifique un mensaje adicional para incluirlo en las notificaciones. De forma predeterminada, los mensajes de la notificación incluyen información sobre la alarma. También puede incluir un mensaje adicional que proporcione al usuario más información.

```
{
  ...
  "assetModelProperties": [
    ...
    {
      "name": "additionalMessage",
      "dataType": "STRING",
      "type": {
        "attribute": {
          "defaultValue": "Turn off the power before you check the alarm."
        }
      }
    }
  ]
}
```

5. Cree el modelo de activo o actualice el modelo de activo existente. Realice una de las siguientes acciones:

- Ejecute el siguiente comando para crear el modelo de activo.

```
aws iotsitewise create-asset-model --cli-input-json file://asset-model-payload.json
```

- Ejecute el siguiente comando para actualizar el modelo de activo existente. *asset-model-id* Sustitúyalo por el ID del modelo de activo.

```
aws iotsitewise update-asset-model \
  --asset-model-id asset-model-id \
  --cli-input-json file://asset-model-payload.json
```

Después de ejecutar el comando, anote `assetModelId` en la respuesta.

Ejemplo: modelo de activo de caldera

El siguiente modelo de activo representa una caldera que informa de los datos de temperatura. Este modelo de activos define una alarma que detecta cuando la caldera se sobrecalienta.

```
{
  "assetModelName": "Boiler Model",
  "assetModelDescription": "Represents a boiler.",
  "assetModelProperties": [
    {
      "name": "Temperature",
      "dataType": "DOUBLE",
      "unit": "C",
      "type": {
        "measurement": {}
      }
    },
    {
      "name": "Temperature Max Threshold",
      "dataType": "DOUBLE",
      "type": {
        "attribute": {
          "defaultValue": "105.0"
        }
      }
    },
    {
      "name": "identityStoreId",
      "dataType": "STRING",
      "type": {
        "attribute": {
          "defaultValue": "d-123EXAMPLE"
        }
      }
    },
    {
      "name": "userId",
      "dataType": "STRING",
      "type": {
        "attribute": {
          "defaultValue": "123EXAMPLE-a1b2c3d4-5678-90ab-cdef-33333EXAMPLE"
        }
      }
    }
  ],
}
```

```
{
  "name": "senderId",
  "dataType": "STRING",
  "type": {
    "attribute": {
      "defaultValue": "MyFactory"
    }
  }
},
{
  "name": "fromAddress",
  "dataType": "STRING",
  "type": {
    "attribute": {
      "defaultValue": "my.factory@example.com"
    }
  }
},
{
  "name": "emailSubject",
  "dataType": "STRING",
  "type": {
    "attribute": {
      "defaultValue": "[ALERT] High boiler temperature"
    }
  }
},
{
  "name": "additionalMessage",
  "dataType": "STRING",
  "type": {
    "attribute": {
      "defaultValue": "Turn off the power before you check the alarm."
    }
  }
}
],
"assetModelHierarchies": [

],
"assetModelCompositeModels": [
  {
    "name": "BoilerTemperatureHighAlarm",
    "type": "AWS/ALARM",
```

```

    "properties": [
      {
        "name": "AWS/ALARM_TYPE",
        "dataType": "STRING",
        "type": {
          "attribute": {
            "defaultValue": "IOT_EVENTS"
          }
        }
      },
      {
        "name": "AWS/ALARM_STATE",
        "dataType": "STRUCT",
        "dataTypeSpec": "AWS/ALARM_STATE",
        "type": {
          "measurement": {}
        }
      }
    ]
  }
}

```

Paso 2: Defina un modelo AWS IoT Events de alarma

Cree el modelo de alarma en AWS IoT Events. En AWS IoT Events, se utilizan expresiones para especificar valores en los modelos de alarma. Puede utilizar expresiones para especificar los valores desde los AWS IoT SiteWise que evaluar y utilizarlos como entradas para la alarma. Cuando AWS IoT SiteWise envía los valores de las propiedades del activo al modelo de alarma, AWS IoT Events evalúa la expresión para obtener el valor de la propiedad o el identificador del activo. Puede utilizar las siguientes expresiones en el modelo de alarma:

- Valores de propiedades de activos

Para obtener el valor de la propiedad de un activo, utilice la siguiente expresión.

assetModelId Sustitúyala por la ID del modelo de activos y *propertyId* sustitúyala por la ID de la propiedad.

```
$sitewise.assetModel.`assetModelId`.`propertyId` .propertyValue.value
```

- Activo IDs

Para obtener el ID del activo, utilice la siguiente expresión. *assetModelId* sustitúyalo por el ID del modelo de activo y *propertyId* sustitúyalo por el ID de la propiedad.

```
$sitewise.assetModel.`assetModelId`.`propertyId`.assetId
```

Note

Al crear el modelo de alarma, puede definir literales en lugar de expresiones que se evalúan como AWS IoT SiteWise valores. Esto puede reducir el número de atributos que define en su modelo de activo. Sin embargo, si define un valor como literal, no podrá personalizar ese valor en los activos basados en el modelo de activo. AWS IoT SiteWise Monitor Los usuarios tampoco pueden personalizar la alarma, ya que solo pueden configurar los ajustes de alarma en los activos.

Para crear un modelo AWS IoT Events de alarma (CLI)


1. Al crear el modelo de alarma en AWS IoT Events, debe especificar el ID de cada propiedad que utilice la alarma, que incluye lo siguiente:
 - La propiedad del estado de alarma en el modelo de activo compuesto
 - La propiedad que monitorea la alarma
 - El atributo de umbral
 - (Opcional) El atributo de ID del almacén de identidades de IAM Identity Center
 - (Opcional) El atributo de ID de usuario de IAM Identity Center
 - (Opcional) El atributo de ID del remitente del SMS
 - (Opcional) El atributo de dirección de origen de del correo electrónico
 - (Opcional) El atributo de asunto del correo electrónico
 - (Opcional) El atributo de mensaje adicional

Ejecute el siguiente comando para recuperar estas propiedades en el modelo de activos. IDs *asset-model-id* sustitúyalo por el ID del modelo de activos del paso anterior.

```
aws iotsitewise describe-asset-model --asset-model-id asset-model-id
```

La operación devuelve una respuesta que contiene los detalles del modelo de activo. Anote el ID de cada propiedad que utiliza la alarma. Los usará IDs cuando cree el modelo de AWS IoT Events alarma en el siguiente paso.

2. Cree el modelo de alarma en AWS IoT Events. Haga lo siguiente:
 - a. Cree un archivo denominado `alarm-model-payload.json`.
 - b. Copie el objeto JSON siguiente en el archivo.
 - c. Escriba un nombre (`alarmModelName`), la descripción (`alarmModelDescription`) y gravedad (`severity`) de su alarma. Para la gravedad, especifique un número entero que refleje los niveles de gravedad de su empresa.

 Important

El modelo de alarma debe tener el mismo nombre que el modelo compuesto de alarma que definió anteriormente en su modelo de activo.

Los nombres de modelo de la alarma pueden contener únicamente caracteres alfanuméricos.

```
{
  "alarmModelName": "BoilerTemperatureHighAlarm",
  "alarmModelDescription": "Detects when the boiler temperature is high.",
  "severity": 3
}
```

- d. Añada la regla de comparación (`alarmRule`) a la alarma. Esta regla define la propiedad que se debe monitorear (`inputProperty`), el valor de umbral que se va a comparar (`threshold`) y el operador de comparación que se debe usar (`comparisonOperator`).
 - `assetModelId` Sustitúyalo por el ID del modelo de activos.
 - `alarmPropertyId` Sustitúyalo por el ID de la propiedad que monitorea la alarma.
 - `thresholdAttributeId` Sustitúyalo por el ID de la propiedad del atributo de umbral.
 - `GREATER` Sustitúyalo por el operador que se utilizará para comparar los valores de las propiedades con el umbral. Puede elegir entre las siguientes opciones:
 - LESS
 - LESS_OR_EQUAL

- EQUAL
- NOT_EQUAL
- GREATER_OR_EQUAL
- GREATER

```
{
  "alarmModelName": "BoilerTemperatureHighAlarm",
  "alarmModelDescription": "Detects when the boiler temperature is high.",
  "severity": 3,
  "alarmRule": {
    "simpleRule": {
      "inputProperty":
"$sitewise.assetModel.`assetModelId`.`alarmPropertyId` .propertyValue.value",
      "comparisonOperator": "GREATER",
      "threshold":
"$sitewise.assetModel.`assetModelId`.`thresholdAttributeId` .propertyValue.value"
    }
  }
}
```

- e. Agregar una acción (alarmEventActions) para enviar el estado de alarma al AWS IoT SiteWise cuando la alarma cambie de estado.

Note

Para una configuración avanzada, puede definir acciones adicionales que deben realizarse cuando la alarma cambie de estado. Por ejemplo, puede llamar a una función de AWS Lambda o publicar en un tema de MQTT. Para obtener más información, consulte [Trabajar con otros AWS servicios](#) en la Guía para AWS IoT Events desarrolladores.

- *assetModelId* Sustitúyalo por el ID del modelo de activos.
- *alarmPropertyId* Sustitúyalo por el ID de la propiedad que monitorea la alarma.
- *alarmStatePropertyId* Sustitúyalo por el ID de la propiedad del estado de alarma en el modelo compuesto de alarmas.

```
{
  "alarmModelName": "BoilerTemperatureHighAlarm",
  "alarmModelDescription": "Detects when the boiler temperature is high.",
  "severity": 3,
  "alarmRule": {
    "simpleRule": {
      "inputProperty":
"$sitewise.assetModel.`assetModelId`.`alarmPropertyId`.propertyValue.value",
      "comparisonOperator": "GREATER",
      "threshold":
"$sitewise.assetModel.`assetModelId`.`thresholdAttributeId`.propertyValue.value"
    }
  },
  "alarmEventActions": {
    "alarmActions": [
      {
        "iotSiteWise": {
          "assetId":
"$sitewise.assetModel.`assetModelId`.`alarmPropertyId`.assetId",
          "propertyId": "`alarmStatePropertyId`"
        }
      }
    ]
  }
}
```

- f. (Opcional) Configure las opciones de notificación de alarmas. La acción de notificación de alarmas utiliza una función de Lambda en su cuenta para enviar notificaciones de alarma. Para obtener más información, consulte [Requisitos para las notificaciones de alarma en AWS IoT SiteWise](#). En los ajustes de notificación de alarmas, puede configurar las notificaciones por SMS y correo electrónico para enviarlas a los usuarios del IAM Identity Center. Haga lo siguiente:
- i. Añada la configuración de notificaciones de alarma (`alarmNotification`) a la carga en `alarm-model-payload.json`.
 - `alarmNotificationFunctionArn` Sustitúyalo por el ARN de la función Lambda que gestiona las notificaciones de alarma.

```

{
  "alarmModelName": "BoilerTemperatureHighAlarm",
  "alarmModelDescription": "Detects when the boiler temperature is high.",
  "severity": 3,
  "alarmRule": {
    "simpleRule": {
      "inputProperty":
"$sitewise.assetModel.`assetModelId`.`alarmPropertyId` .propertyValue.value",
      "comparisonOperator": "GREATER",
      "threshold":
"$sitewise.assetModel.`assetModelId`.`thresholdAttributeId` .propertyValue.value"
    }
  },
  "alarmEventActions": {
    "alarmActions": [
      {
        "iotSiteWise": {
          "assetId":
"$sitewise.assetModel.`assetModelId`.`alarmPropertyId` .assetId",
          "propertyId": "'alarmStatePropertyId'"
        }
      }
    ]
  },
  "alarmNotification": {
    "notificationActions": [
      {
        "action": {
          "lambdaAction": {
            "functionArn": "alarmNotificationFunctionArn"
          }
        }
      }
    ]
  }
}

```

- ii. (Opcional) Configure las notificaciones por SMS (`smsConfigurations`) para enviarlas a un usuario de IAM Identity Center cuando la alarma cambie de estado.

- *identityStoreIdAttributeId* Sustitúyalo por el ID del atributo que contiene el ID del almacén de identidades del IAM Identity Center.
- *userIdAttributeId* Sustitúyalo por el ID del atributo que contiene el ID del usuario del Centro de Identidad de IAM.
- *senderIdAttributeId* Sustitúyalo por el ID del atributo que contiene el ID del remitente de Amazon SNS o elimínalo `senderId` de la carga útil.
- *additionalMessageAttributeId* Sustitúyalo por el ID del atributo que contiene el mensaje adicional o elimínalo `additionalMessage` de la carga útil.

```
{
  "alarmModelName": "BoilerTemperatureHighAlarm",
  "alarmModelDescription": "Detects when the boiler temperature is high.",
  "severity": 3,
  "alarmRule": {
    "simpleRule": {
      "inputProperty":
"$sitewise.assetModel.`assetModelId`.`alarmPropertyId`.propertyValue.value",
      "comparisonOperator": "GREATER",
      "threshold":
"$sitewise.assetModel.`assetModelId`.`thresholdAttributeId`.propertyValue.value"
    }
  },
  "alarmEventActions": {
    "alarmActions": [
      {
        "iotSiteWise": {
          "assetId":
"$sitewise.assetModel.`assetModelId`.`alarmPropertyId`.assetId",
          "propertyId": "'alarmStatePropertyId'"
        }
      }
    ]
  },
  "alarmNotification": {
    "notificationActions": [
      {
        "action": {
          "lambdaAction": {
            "functionArn": "alarmNotificationFunctionArn"
          }
        }
      }
    ]
  }
}
```

```

    },
    "smsConfigurations": [
      {
        "recipients": [
          {
            "ssoIdentity": {
              "identityStoreId":
"$sitewise.assetModel.`assetModelId`.`identityStoreIdAttributeId`.propertyValue.value",
              "userId":
"$sitewise.assetModel.`assetModelId`.`userIdAttributeId`.propertyValue.value"
            }
          }
        ],
        "senderId":
"$sitewise.assetModel.`assetModelId`.`senderIdAttributeId`.propertyValue.value",
        "additionalMessage":
"$sitewise.assetModel.`assetModelId`.`additionalMessageAttributeId`.propertyValue.value"
      }
    ]
  }
}
}
}
}

```

- iii. (Opcional) Configure las notificaciones por correo electrónico (emailConfigurations) para enviarlas a un usuario de IAM Identity Center cuando la alarma cambie de estado.
 - *identityStoreIdAttributeId* Sustitúyalo por el ID de la propiedad del atributo ID del almacén de identidades del IAM Identity Center.
 - *userIdAttributeId* Sustitúyalo por el ID de la propiedad del atributo de ID de usuario del IAM Identity Center.
 - *fromAddressAttributeId* Sustitúyalo por el ID de la propiedad del atributo de dirección «de» o elimínalo from de la carga útil.
 - *emailSubjectAttributeId* Sustitúyalo por el ID de la propiedad del atributo del asunto del correo electrónico o elimínalo subject de la carga útil.
 - *additionalMessageAttributeId* Sustitúyala por el ID de la propiedad de atributo de mensaje adicional o elimínala additionalMessage de la carga útil.

```

{
  "alarmModelName": "BoilerTemperatureHighAlarm",
  "alarmModelDescription": "Detects when the boiler temperature is high.",
  "severity": 3,
  "alarmRule": {
    "simpleRule": {
      "inputProperty":
"$sitewise.assetModel.`assetModelId`.`alarmPropertyId` .propertyValue.value",
      "comparisonOperator": "GREATER",
      "threshold":
"$sitewise.assetModel.`assetModelId`.`thresholdAttributeId` .propertyValue.value"
    }
  },
  "alarmEventActions": {
    "alarmActions": [
      {
        "iotSiteWise": {
          "assetId":
"$sitewise.assetModel.`assetModelId`.`alarmPropertyId` .assetId",
          "propertyId": "'alarmStatePropertyId'"
        }
      }
    ]
  },
  "alarmNotification": {
    "notificationActions": [
      {
        "action": {
          "lambdaAction": {
            "functionArn": "alarmNotificationFunctionArn"
          }
        },
        "smsConfigurations": [
          {
            "recipients": [
              {
                "ssoIdentity": {
                  "identityStoreId":
"$sitewise.assetModel.`assetModelId`.`identityStoreIdAttributeId` .propertyValue.va",
                  "userId":
"$sitewise.assetModel.`assetModelId`.`userIdAttributeId` .propertyValue.value"
                }
              }
            ]
          }
        ]
      }
    ]
  }
}

```

```
    }
    ],
    "senderId":
"$sitewise.assetModel.`assetModelId`.`senderIdAttributeId`.propertyValue.value",
    "additionalMessage":
"$sitewise.assetModel.`assetModelId`.`additionalMessageAttributeId`.propertyValue.
    }
  ],
  "emailConfigurations": [
    {
      "from":
"$sitewise.assetModel.`assetModelId`.`fromAddressAttributeId`.propertyValue.value"
      "recipients": {
        "to": [
          {
            "ssoIdentity": {
              "identityStoreId":
"$sitewise.assetModel.`assetModelId`.`identityStoreIdAttributeId`.propertyValue.va
              "userId":
"$sitewise.assetModel.`assetModelId`.`userIdAttributeId`.propertyValue.value"
            }
          }
        ]
      },
      "content": {
        "subject":
"$sitewise.assetModel.`assetModelId`.`emailSubjectAttributeId`.propertyValue.value
        "additionalMessage":
"$sitewise.assetModel.`assetModelId`.`additionalMessageAttributeId`.propertyValue.
      }
    }
  ]
}
}
```

- g. (Opcional) Añada las funciones de alarma (`alarmCapabilities`) a la carga en `alarm-model-payload.json`. En este objeto, puede especificar si se habilita el flujo de reconocimiento y el estado de activación predeterminado de los activos basados en el modelo de activo. Para obtener más información sobre el flujo de reconocimiento, consulte [Estados de alarma](#).

```

{
  "alarmModelName": "BoilerTemperatureHighAlarm",
  "alarmModelDescription": "Detects when the boiler temperature is high.",
  "severity": 3,
  "alarmRule": {
    "simpleRule": {
      "inputProperty":
"$sitewise.assetModel.`assetModelId`.`alarmPropertyId` .propertyValue.value",
      "comparisonOperator": "GREATER",
      "threshold":
"$sitewise.assetModel.`assetModelId`.`thresholdAttributeId` .propertyValue.value"
    }
  },
  "alarmEventActions": {
    "alarmActions": [
      {
        "iotSiteWise": {
          "assetId":
"$sitewise.assetModel.`assetModelId`.`alarmPropertyId` .assetId",
          "propertyId": "'alarmStatePropertyId'"
        }
      }
    ]
  },
  "alarmNotification": {
    "notificationActions": [
      {
        "action": {
          "lambdaAction": {
            "functionArn": "alarmNotificationFunctionArn"
          }
        },
        "smsConfigurations": [
          {
            "recipients": [
              {
                "ssoIdentity": {
                  "identityStoreId":
"$sitewise.assetModel.`assetModelId`.`identityStoreIdAttributeId` .propertyValue.value",
                  "userId":
"$sitewise.assetModel.`assetModelId`.`userIdAttributeId` .propertyValue.value"
                }
              }
            ]
          }
        ]
      }
    ]
  }
}

```

```

    ],
    "senderId":
"$sitewise.assetModel.`assetModelId`.`senderIdAttributeId`.propertyValue.value",
    "additionalMessage":
"$sitewise.assetModel.`assetModelId`.`additionalMessageAttributeId`.propertyValue.value"
  }
],
"emailConfigurations": [
  {
    "from":
"$sitewise.assetModel.`assetModelId`.`fromAddressAttributeId`.propertyValue.value",
    "recipients": {
      "to": [
        {
          "ssoIdentity": {
            "identityStoreId":
"$sitewise.assetModel.`assetModelId`.`identityStoreIdAttributeId`.propertyValue.value"
            "userId":
"$sitewise.assetModel.`assetModelId`.`userIdAttributeId`.propertyValue.value"
          }
        }
      ]
    },
    "content": {
      "subject":
"$sitewise.assetModel.`assetModelId`.`emailSubjectAttributeId`.propertyValue.value",
      "additionalMessage":
"$sitewise.assetModel.`assetModelId`.`additionalMessageAttributeId`.propertyValue.value"
    }
  }
]
},
"alarmCapabilities": {
  "initializationConfiguration": {
    "disabledOnInitialization": false
  },
  "acknowledgeFlow": {
    "enabled": true
  }
}
}

```

- h. Añada la función de servicio de IAM (`roleArn`) a la que AWS IoT Events se pueden enviar los datos. AWS IoT SiteWise Este rol requiere el permiso `iotsitewise:BatchPutAssetPropertyValue` y una relación de confianza que permita a `iotevents.amazonaws.com` asumir el rol. Para enviar notificaciones, este rol también requiere los permisos `lambda:InvokeFunction` y `sso-directory:DescribeUser`. Para obtener más información, consulte [Rol de servicio de alarmas](#) en la Guía para desarrolladores de AWS IoT Events .
- Sustituya el `roleArn` por el ARN del rol que AWS IoT Events puede asumir para realizar estas acciones.

```
{
  "alarmModelName": "BoilerTemperatureHighAlarm",
  "alarmModelDescription": "Detects when the boiler temperature is high.",
  "severity": 3,
  "alarmRule": {
    "simpleRule": {
      "inputProperty":
"$sitewise.assetModel.`assetModelId`.`alarmPropertyId`.propertyValue.value",
      "comparisonOperator": "GREATER",
      "threshold":
"$sitewise.assetModel.`assetModelId`.`thresholdAttributeId`.propertyValue.value"
    }
  },
  "alarmEventActions": {
    "alarmActions": [
      {
        "iotSiteWise": {
          "assetId":
"$sitewise.assetModel.`assetModelId`.`alarmPropertyId`.assetId",
          "propertyId": "'alarmStatePropertyId'"
        }
      }
    ]
  },
  "alarmNotification": {
    "notificationActions": [
      {
        "action": {
          "lambdaAction": {
            "functionArn": "alarmNotificationFunctionArn"
          }
        }
      }
    ]
  }
}
```

```

    }
  },
  "smsConfigurations": [
    {
      "recipients": [
        {
          "ssoIdentity": {
            "identityStoreId":
"$sitewise.assetModel.`assetModelId`.`identityStoreIdAttributeId`.propertyValue.value"
            "userId":
"$sitewise.assetModel.`assetModelId`.`userIdAttributeId`.propertyValue.value"
          }
        }
      ],
      "senderId":
"$sitewise.assetModel.`assetModelId`.`senderIdAttributeId`.propertyValue.value",
      "additionalMessage":
"$sitewise.assetModel.`assetModelId`.`additionalMessageAttributeId`.propertyValue.value"
    }
  ],
  "emailConfigurations": [
    {
      "from":
"$sitewise.assetModel.`assetModelId`.`fromAddressAttributeId`.propertyValue.value",
      "recipients": {
        "to": [
          {
            "ssoIdentity": {
              "identityStoreId":
"$sitewise.assetModel.`assetModelId`.`identityStoreIdAttributeId`.propertyValue.value"
              "userId":
"$sitewise.assetModel.`assetModelId`.`userIdAttributeId`.propertyValue.value"
            }
          }
        ]
      },
      "content": {
        "subject":
"$sitewise.assetModel.`assetModelId`.`emailSubjectAttributeId`.propertyValue.value",
        "additionalMessage":
"$sitewise.assetModel.`assetModelId`.`additionalMessageAttributeId`.propertyValue.value"
      }
    }
  ]
]

```

```
    }
  ]
},
"alarmCapabilities": {
  "initializationConfiguration": {
    "disabledOnInitialization": false
  },
  "acknowledgeFlow": {
    "enabled": false
  }
},
"roleArn": "arn:aws:iam::123456789012:role/MyIoTEventsAlarmRole"
}
```

- i. Ejecute el siguiente comando para crear el modelo de AWS IoT Events alarma a partir de la carga útil entrante. `alarm-model-payload.json`

```
aws iotevents create-alarm-model --cli-input-json file://alarm-model-payload.json
```

- j. La operación devuelve una respuesta que incluye el ARN del modelo de alarma, `alarmModelArn`. Copie este ARN para configurar en el siguiente paso la definición de la alarma en su modelo de activos.

Paso 3: Habilite el flujo de datos entre y AWS IoT SiteWise AWS IoT Events

Después de crear los recursos necesarios en AWS IoT SiteWise y AWS IoT Events, puede habilitar el flujo de datos entre los recursos para activar la alarma. En esta sección, usted actualiza la definición de alarma en el modelo de activo para usar el modelo de alarma que ha creado en el paso anterior.

Para habilitar el flujo de datos entre AWS IoT SiteWise y AWS IoT Events (CLI)

- Configure el modelo de alarma como origen de la alarma en el modelo de activo. Haga lo siguiente:
 - a. Ejecute el siguiente comando para recuperar la definición del modelo de activo existente. `asset-model-id`Sustitúyalo por el ID del modelo de activos.

```
aws iotsitewise describe-asset-model --asset-model-id asset-model-id
```

La operación devuelve una respuesta que contiene los detalles del modelo de activo.

- b. Cree un archivo llamado `update-asset-model-payload.json` y copie la respuesta del comando anterior en el archivo.
- c. Elimine los siguientes pares de clave-valor del archivo `update-asset-model-payload.json`:
 - `assetModelId`
 - `assetModelArn`
 - `assetModelCreationDate`
 - `assetModelLastUpdateDate`
 - `assetModelStatus`
- d. Añada la propiedad de origen de alarma (`AWS/ALARM_SOURCE`) al modelo compuesto de alarmas que definió anteriormente. `alarmModelArn` Sustitúyalo por el ARN del modelo de alarma, que establece el valor de la propiedad de la fuente de alarma.

```
{
  ...
  "assetModelCompositeModels": [
    ...
    {
      "name": "BoilerTemperatureHighAlarm",
      "type": "AWS/ALARM",
      "properties": [
        {
          "id": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
          "name": "AWS/ALARM_TYPE",
          "dataType": "STRING",
          "type": {
            "attribute": {
              "defaultValue": "IOT_EVENTS"
            }
          }
        },
        {
          "id": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
          "name": "AWS/ALARM_STATE",
          "dataType": "STRUCT",
          "dataTypeSpec": "AWS/ALARM_STATE",
          "type": {
```

```

        "measurement": {}
      }
    },
    {
      "name": "AWS/ALARM_SOURCE",
      "dataType": "STRING",
      "type": {
        "attribute": {
          "defaultValue": "alarmModelArn"
        }
      }
    }
  ]
}
]
}

```

- e. Ejecute el siguiente comando para actualizar el modelo de activo con la definición almacenada en el archivo `update-asset-model-payload.json`. *asset-model-id* Sustitúyalo por el ID del modelo de activos.

```

aws iotsitewise update-asset-model \
  --asset-model-id asset-model-id \
  --cli-input-json file://update-asset-model-payload.json

```

Su modelo de activo ahora define una alarma capaz de detectar en AWS IoT Events. La alarma monitorea la propiedad de destino en todos los activos basados en este modelo de activo. Puede configurar la alarma de cada activo para personalizar propiedades como el umbral o el destinatario de IAM Identity Center para cada activo. Para obtener más información, consulte [Configure las alarmas de los activos en AWS IoT SiteWise](#).

Defina las alarmas externas en AWS IoT SiteWise

Las alarmas externas contienen el estado de las alarmas que se detectan fuera de AWS IoT SiteWise.

Definición de una alarma externa (consola)

Puede utilizar la AWS IoT SiteWise consola para definir una alarma externa en un modelo de activos existente. Para definir una alarma externa en un nuevo modelo de activo, cree el modelo de activo y,

a continuación, siga estos pasos. Para obtener más información, consulte [Cree modelos de activos en AWS IoT SiteWise](#).

Para definir una alarma en un modelo de activo

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, elija Models (Modelos).
3. Elija el modelo de activo para el que desee definir una alarma.
4. Seleccione la pestaña Definiciones de alarma.
5. Seleccione Añadir alarma.
6. En Opciones de tipos de alarmas, seleccione Alarma externa.
7. Escriba un nombre para la alarma.
8. (Opcional) Ingrese una descripción para su alarma.
9. Seleccione Añadir alarma.

Definición de una alarma externa (CLI)

Puede utilizarla AWS CLI para definir una alarma externa en un modelo de activos nuevo o existente.

Para añadir una alarma externa a un modelo de activo, añada un modelo compuesto de alarma al modelo de activo. Un modelo compuesto de alarmas externas especifica el tipo de EXTERNAL, pero no especifica la propiedad de origen de alarma. El siguiente ejemplo de alarma compuesta define una alarma externa de temperatura.

```
{
  ...
  "assetModelCompositeModels": [
    {
      "name": "BoilerTemperatureHighAlarm",
      "type": "AWS/ALARM",
      "properties": [
        {
          "name": "AWS/ALARM_TYPE",
          "dataType": "STRING",
          "type": {
            "attribute": {
              "defaultValue": "EXTERNAL"
            }
          }
        }
      ]
    }
  ]
}
```

```
    },
    {
      "name": "AWS/ALARM_STATE",
      "dataType": "STRUCT",
      "dataTypeSpec": "AWS/ALARM_STATE",
      "type": {
        "measurement": {}
      }
    }
  ]
}
]
```

Para obtener más información sobre cómo añadir un modelo compuesto a un modelo de activo nuevo o existente, consulte lo siguiente:

- [Creación de un modelo de activos \(AWS CLI\)](#)
- [Actualice un modelo de activo, un modelo de componentes o una interfaz \(AWS CLI\)](#)

Después de definir la alarma externa, puede ingerir el estado de alarma a los activos basados en el modelo de activo. Para obtener más información, consulte [Introduzca un estado de alarma externo en AWS IoT SiteWise](#).

Configure las alarmas de los activos en AWS IoT SiteWise

Tras definir una AWS IoT Events alarma en un modelo de activo, puede configurar la alarma en cada activo en función del modelo de activo. Puede editar el valor de umbral y los ajustes de notificación de la alarma. Cada uno de estos valores es un atributo del activo, por lo que puede actualizar el valor predeterminado del atributo para configurar estos valores.

Note

Puede configurar estos valores para AWS IoT Events las alarmas, pero no para las alarmas externas.

Temas

- [Configuración de un valor de umbral \(consola\)](#)

- [Configuración de un valor de umbral \(AWS CLI\)](#)
- [Configure los ajustes de notificación en AWS IoT SiteWise](#)

Configuración de un valor de umbral (consola)

Puede utilizar la AWS IoT SiteWise consola para actualizar el valor del atributo que especifica el valor umbral de una alarma.

Para actualizar el valor del umbral de una alarma (consola)

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, elija Activos.
3. Elija el activo para el que desea actualizar un valor del umbral de la alarma.

Tip

Puede elegir el icono de flecha para expandir una jerarquía de activos y encontrar su activo.

4. Elija Edit (Edición de).
5. Busque el atributo que utiliza la alarma para su valor de umbral y, a continuación, introduzca su nuevo valor.
6. Seleccione Save.

Configuración de un valor de umbral (AWS CLI)

Puede usar el AWS Command Line Interface (AWS CLI) para actualizar el valor del atributo que especifica el valor umbral de una alarma.

Debe conocer los `assetId` de sus activos y los `propertyId` de las propiedades para completar este procedimiento. También puede utilizar el ID externo. Si has creado un activo y no lo sabes `assetId`, usa la [ListAssets](#) API para enumerar todos los activos de un modelo específico. Utilice la [DescribeAsset](#) operación para ver las propiedades de su activo, incluida la propiedad IDs.

Utilice la operación [BatchPutAssetPropertyValue](#) para asignar valores de atributo a su activo. Puede utilizar esta operación para establecer varios atributos a la vez. La carga de esta operación contiene una lista de entradas y cada una contiene el ID de activo, el ID de propiedad y el valor de atributo.

Actualización del valor de un atributo (AWS CLI)

1. Cree un archivo llamado `batch-put-payload.json` y copie el siguiente objeto JSON en el archivo. En esta carga de ejemplo se muestra cómo establecer la latitud y la longitud de una turbina eólica. Actualice los IDs valores y las marcas de tiempo para modificar la carga útil según su caso de uso.

```
{
  "entries": [
    {
      "entryId": "windfarm3-turbine7-latitude",
      "assetId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
      "propertyId": "a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
      "propertyValues": [
        {
          "value": {
            "doubleValue": 47.6204
          },
          "timestamp": {
            "timeInSeconds": 1575691200
          }
        }
      ]
    },
    {
      "entryId": "windfarm3-turbine7-longitude",
      "assetId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
      "propertyId": "a1b2c3d4-5678-90ab-cdef-55555EXAMPLE",
      "propertyValues": [
        {
          "value": {
            "doubleValue": 122.3491
          },
          "timestamp": {
            "timeInSeconds": 1575691200
          }
        }
      ]
    }
  ]
}
```

- Cada entrada de la carga contiene un `entryId` que puede definir como una única cadena. Si la entrada de la solicitud no se realiza correctamente, cada error contendrá el `entryId` de la solicitud correspondiente para que sepa qué solicitudes deben volver a intentarse.
- Para establecer un valor de atributo, puede incluir una estructura `timestamp-quality-value` (TQV) en la lista de propiedades de `propertyValues` cada atributo. Esta estructura debe contener el nuevo `value` y la `timestamp` actual.
 - `value`: una estructura que contiene uno de los siguientes campos, en función del tipo de propiedad que se establezca:
 - `booleanValue`
 - `doubleValue`
 - `integerValue`
 - `stringValue`
 - `nullValue`
 - `timestamp`— Una estructura que contiene el tiempo de época actual de Unix en segundos. `timeInSeconds` AWS IoT SiteWise rechaza todos los puntos de datos con marcas de tiempo que hayan existido durante más de 7 días o más de 5 minutos en el futuro.

Para obtener más información acerca de cómo preparar una carga para [BatchPutAssetPropertyValue](#), consulte [Ingiera datos con AWS IoT SiteWise APIs](#).

2. Ejecute el siguiente comando para enviar los valores del atributo a AWS IoT SiteWise:

```
aws iotsitewise batch-put-asset-property-value -\-cli-input-json file://batch-put-payload.json
```

Configure los ajustes de notificación en AWS IoT SiteWise

Puede configurar los ajustes de notificación de alarmas mediante la AWS IoT SiteWise consola o el AWS Command Line Interface (AWS CLI).

Configuración de ajustes de notificación (consola)

Puede utilizar la AWS IoT SiteWise consola para actualizar el valor de los atributos que especifican la configuración de notificación de una alarma.

Para actualizar los ajustes de notificación de una alarma (consola)

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, elija Activos.
3. Elija el activo para el que desea actualizar la configuración de la alarma.
4. Elija Edit (Edición de).
5. Busque el atributo que utiliza la alarma utiliza para el ajuste de notificación que desee cambiar y, a continuación, introduzca su nuevo valor.
6. Seleccione Save.

Configuración de ajustes de notificaciones (CLI)

Puede usar el AWS Command Line Interface (AWS CLI) para actualizar el valor del atributo que especifica la configuración de notificación de una alarma.

Debe conocer los `assetId` de sus activos y los `propertyId` de las propiedades para completar este procedimiento. También puede utilizar el ID externo. Si has creado un activo y no lo sabes `assetId`, usa la [ListAssets](#) API para enumerar todos los activos de un modelo específico. Utilice la [DescribeAsset](#) operación para ver las propiedades de su activo, incluida la propiedad IDs.

Utilice la operación [BatchPutAssetPropertyValue](#) para asignar valores de atributo a su activo. Puede utilizar esta operación para establecer varios atributos a la vez. La carga de esta operación contiene una lista de entradas y cada una contiene el ID de activo, el ID de propiedad y el valor de atributo.

Actualización del valor de un atributo (AWS CLI)

1. Cree un archivo llamado `batch-put-payload.json` y copie el siguiente objeto JSON en el archivo. En esta carga de ejemplo se muestra cómo establecer la latitud y la longitud de una turbina eólica. Actualice los IDs valores y las marcas de tiempo para modificar la carga útil según su caso de uso.

```
{
  "entries": [
    {
      "entryId": "windfarm3-turbine7-latitude",
      "assetId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
      "propertyId": "a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
      "propertyValues": [
        {
```

```
    "value": {
      "doubleValue": 47.6204
    },
    "timestamp": {
      "timeInSeconds": 1575691200
    }
  }
],
{
  "entryId": "windfarm3-turbine7-longitude",
  "assetId": "a1b2c3d4-5678-90ab-cdef-2222EXAMPLE",
  "propertyId": "a1b2c3d4-5678-90ab-cdef-5555EXAMPLE",
  "propertyValues": [
    {
      "value": {
        "doubleValue": 122.3491
      },
      "timestamp": {
        "timeInSeconds": 1575691200
      }
    }
  ]
}
]
```

- Cada entrada de la carga contiene un `entryId` que puede definir como una única cadena. Si la entrada de la solicitud no se realiza correctamente, cada error contendrá el `entryId` de la solicitud correspondiente para que sepa qué solicitudes deben volver a intentarse.
- Para establecer un valor de atributo, puede incluir una estructura `timestamp-quality-value` (TQV) en la lista de propiedades de `propertyValues` cada atributo. Esta estructura debe contener el nuevo `value` y la `timestamp` actual.
- `value`: una estructura que contiene uno de los siguientes campos, en función del tipo de propiedad que se establezca:
 - `booleanValue`
 - `doubleValue`
 - `integerValue`
 - `stringValue`

- `nullValue`
- `timestamp`— Una estructura que contiene el tiempo de época actual de Unix en segundos. `timeInSeconds` AWS IoT SiteWise rechaza todos los puntos de datos con marcas de tiempo que hayan existido durante más de 7 días o más de 5 minutos en el futuro.

Para obtener más información acerca de cómo preparar una carga para [BatchPutAssetPropertyValue](#), consulte [Ingiera datos con AWS IoT SiteWise APIs](#).

2. Ejecute el siguiente comando para enviar los valores del atributo a AWS IoT SiteWise:

```
aws iotsitewise batch-put-asset-property-value -\cli-input-json file://batch-put-payload.json
```

Responda a las alarmas en AWS IoT SiteWise

Cuando una AWS IoT Events alarma cambia de estado, puedes hacer lo siguiente para responder a la alarma:

- Confirmar la alarma para indicar que está gestionando el problema.
- Posponer la alarma para desactivarla temporalmente.
- Deshabilitar la alarma para desactivarla permanentemente hasta que la vuelva a habilitar.
- Habilitar una alarma deshabilitada para detectar el estado de alarma.
- Restablecer la alarma para borrar su estado y su último valor.

Puedes usar la AWS IoT SiteWise consola o la AWS IoT Events API para responder a una alarma.

Note

Puede responder a AWS IoT Events las alarmas, pero no a las externas.

Temas

- [Respuesta a una alarma \(consola\)](#)
- [Respuesta a una alarma \(API\)](#)

Respuesta a una alarma (consola)

Puede usar la AWS IoT SiteWise consola para reconocer, posponer, deshabilitar o activar una alarma.

Temas

- [Confirmación de una alarma \(consola\)](#)
- [Posponer una alarma \(consola\)](#)
- [Deshabilitación de una alarma \(consola\)](#)
- [Habilitar una alarma \(consola\)](#)
- [Restablecer una alarma \(consola\)](#)

Confirmación de una alarma (consola)

Se puede confirmar la alarma para indicar que se está gestionando el problema.

Note

Es necesario habilitar el flujo de reconocimiento en la alarma para poder confirmarla. Esta opción está habilitada de forma predeterminada si define la alarma desde la consola de AWS IoT SiteWise .

Para confirmar una alarma (consola)

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, elija Activos.
3. Elija el activo para el que desea confirmar una alarma.

Tip

Puede elegir el icono de flecha para expandir una jerarquía de activos y encontrar su activo.

4. Seleccione la pestaña Alarmas.
5. Seleccione la alarma que desee reconocer y, a continuación, elija Acciones para abrir el menú de acciones de respuesta.

6. Seleccione Confirmar. El estado de la alarma cambia a Confirmado.

Posponer una alarma (consola)

Puede posponer una alarma para deshabilitarla temporalmente. Especifique el tiempo durante el que se debe posponer la alarma.

Para posponer una alarma (consola)

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, elija Activos.
3. Elija el activo para el que desea posponer una alarma.

Tip

Puede elegir el icono de flecha para expandir una jerarquía de activos y encontrar su activo.

4. Seleccione la pestaña Alarmas.
5. Seleccione la alarma que desee posponer y, a continuación, elija Acciones para abrir el menú de acciones de respuesta.
6. Seleccione Posponer. Se abre un modelo en el que se especifica la duración de la pausa.
7. Elija la duración de la acción de posponer o introduzca una duración personalizada de la acción de posponer.
8. Seleccione Save. El estado de la alarma cambia a Silenciado.

Deshabilitación de una alarma (consola)

Se puede deshabilitar una alarma para que no detecte nada más. Después de deshabilitar la alarma, debe volver a habilitarla si desea que la alarma detecte algo.

Para deshabilitar una alarma (consola)

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, elija Activos.
3. Elija el activo para el que desea deshabilitar una alarma.

 Tip

Puede elegir el icono de flecha para expandir una jerarquía de activos y encontrar su activo.

4. Seleccione la pestaña Alarmas.
5. Seleccione la alarma que desee deshabilitar y, a continuación, elija Acciones para abrir el menú de acciones de respuesta.
6. Elija Deshabilitar. El estado de la alarma cambia a Deshabilitado.

Habilitar una alarma (consola)

Puede habilitar una alarma para que se vuelva a detectar después de deshabilitarla o posponerla.

Para habilitar una alarma (consola)

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, elija Activos.
3. Elija el activo para el que desea habilitar una alarma.

 Tip

Puede elegir el icono de flecha para expandir una jerarquía de activos y encontrar su activo.

4. Seleccione la pestaña Alarmas.
5. Seleccione la alarma que desee habilitar y, a continuación, elija Acciones para abrir el menú de acciones de respuesta.
6. Seleccione Habilitar. El estado de la alarma cambia a Normal.

Restablecer una alarma (consola)

Puede restablecer una alarma para borrar su estado y su último valor.

Para restablecer una alarma (consola)

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, elija Activos.
3. Elija el activo para el que desea restablecer una alarma.

Tip

Puede elegir el icono de flecha para expandir una jerarquía de activos y encontrar su activo.

4. Seleccione la pestaña Alarmas.
5. Seleccione la alarma que desee habilitar y, a continuación, elija Acciones para abrir el menú de acciones de respuesta.
6. Elija Restablecer. El estado de la alarma cambia a Normal.

Respuesta a una alarma (API)

Puedes usar la AWS IoT Events API para reconocer, posponer, deshabilitar, activar o restablecer una alarma. Para obtener más información, consulte las siguientes operaciones en la Referencia de la API de AWS IoT Events :

- [BatchAcknowledgeAlarm](#)
- [BatchSnoozeAlarm](#)
- [BatchDisableAlarm](#)
- [BatchEnableAlarm](#)
- [BatchResetAlarm](#)

Para obtener más información, consulte [Respuesta a las alarmas](#) en la Guía para desarrolladores de AWS IoT Events .

Introduzca un estado de alarma externo en AWS IoT SiteWise

Las alarmas externas son alarmas que se evalúan fuera de ellas AWS IoT SiteWise. Puede utilizar alarmas externas si tiene un origen de datos que notifique el estado de alarma que desea ingerir a AWS IoT SiteWise.

Las propiedades de los estados de alarma requieren un formato específico para los valores de los datos de los estados de alarma. Cada valor de datos debe ser un objeto JSON serializado en una cadena. A continuación, ingiera la cadena serializada como valor de cadena. Para obtener más información, consulte [Propiedades del estado de alarma](#).

Example Ejemplo de valor de los datos de estado de alarma (no serializado)

```
{
  "stateName": "Active"
}
```

Example Ejemplo de valor de los datos de estado de alarma (serializado)

```
{"stateName\":\"Active\"}
```

Note

Si su origen de datos no puede generar datos en este formato o no puede convertirlos a este formato antes de ingerirlos, puede optar por no utilizar una propiedad de alarma. En su lugar, puede ingerir los datos como una propiedad de medición con el tipo de datos de cadena, por ejemplo. Para obtener más información, consulte [Definición de flujos de datos procedentes del equipo \(mediciones\)](#) y [Ingiera datos para AWS IoT SiteWise](#).

Mapee los flujos de estados de alarma externos AWS IoT SiteWise

Puede definir alias de propiedades para asignar sus flujos de datos a las propiedades de su estado de alarma. Esto le ayuda a identificar fácilmente una propiedad de estado de alarma cuando ingiera o recupera datos. Para obtener más información acerca de los alias de propiedad, consulte [Gestione los flujos de datos para AWS IoT SiteWise](#).

Temas

- [Asignación de flujos de estados de alarmas externas \(consola\)](#)
- [Asignación de flujos de estados de alarmas externas \(AWS CLI\)](#)

Asignación de flujos de estados de alarmas externas (consola)

Puede definir alias de propiedades para asignar sus flujos de datos a las propiedades de su estado de alarma. Esto le ayuda a identificar fácilmente una propiedad de estado de alarma cuando ingiera o recupera datos. Para obtener más información acerca de los alias de propiedad, consulte [Gestione los flujos de datos para AWS IoT SiteWise](#).

Puede utilizar la AWS IoT SiteWise consola para configurar un alias para una propiedad de estado de alarma.

Para configurar un alias de propiedad para una propiedad de estado de alarma (consola)

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, elija Activos.
3. Elija el activo para el que desea configurar un alias de propiedad.

Tip

Puede elegir el icono de flecha para expandir una jerarquía de activos y encontrar su activo.

4. Elija Edit (Edición de).
5. Desplácese hasta Alarmas y expanda la sección.
6. En Alarmas externas, introduce el alias en Alias de propiedad – opcional.
7. Seleccione Save.

Asignación de flujos de estados de alarmas externas (AWS CLI)

Puede definir alias de propiedades para asignar sus flujos de datos a las propiedades de su estado de alarma. Esto le ayuda a identificar fácilmente una propiedad de estado de alarma cuando ingiera o recupera datos. Para obtener más información acerca de los alias de propiedad, consulte [Gestione los flujos de datos para AWS IoT SiteWise](#).

Puedes usar el AWS Command Line Interface (AWS CLI) para configurar un alias para una propiedad de estado de alarma.

Debe conocer los `assetId` de sus activos y los `propertyId` de las propiedades para completar este procedimiento. También puede utilizar el ID externo. Si has creado un activo y no lo

conocesassetId, usa la [ListAssets](#) API para enumerar todos los activos de un modelo específico. Utilice la [DescribeAsset](#) operación para ver las propiedades de su activo, incluida la propiedad IDs.

Note

La [DescribeAsset](#) respuesta incluye la lista de modelos de activos compuestos para el activo. Cada alarma es un modelo compuesto. Para encontrar el `propertyId`, busque el modelo compuesto de la alarma y, a continuación, busque la propiedad de `AWS/ALARM_STATE` en ese modelo compuesto.

Para obtener información acerca de cómo establecer el alias de la propiedad, consulte [Actualice el alias de una propiedad de un activo](#).

Ingiera los datos del estado de alarma en AWS IoT SiteWise

Las propiedades del estado de alarma esperan que el estado de la alarma sea una cadena JSON serializada. Para transferir el estado de alarma a una alarma externa AWS IoT SiteWise, ingiera esta cadena serializada como un valor de cadena con una marca de tiempo. En el ejemplo siguiente se muestra un valor de datos de estado para una alarma activa.

```
{\"stateName\": \"Active\"}
```

Para identificar una propiedad del estado de alarma, puede especificar uno de los elementos siguientes:

- El `assetId` y el `propertyId` de la propiedad de alarma a la que se envían los datos.
- El `propertyAlias`, que es un alias de flujo de datos (por ejemplo, `/company/windfarm/3/turbine/7/temperature/high`). Para utilizar esta opción, primero debe establecer el alias de la propiedad de la alarma. Para obtener información sobre cómo configurar los alias de propiedad para las propiedades de los estados de alarma, consulte [Mapee los flujos de estados de alarma externos AWS IoT SiteWise](#).

En el siguiente ejemplo de carga útil de la [BatchPutAssetPropertyValue](#) API, se muestra cómo formatear el estado de una alarma externa. Esta alarma externa informa cuando la lectura de rotaciones por minuto (RPM) de una turbina eólica es demasiado alta.

Example Ejemplo de BatchPutAssetPropertyValue carga útil para datos de estado de alarma

```
{
  "entries": [
    {
      "entryId": "unique entry ID",
      "propertyAlias": "/company/windfarm/3/turbine/7/temperature/high",
      "propertyValues": [
        {
          "value": {
            "stringValue": "{\"stateName\":\"Active\"}"
          },
          "timestamp": {
            "timeInSeconds": 1607550262
          }
        }
      ]
    }
  ]
}
```

Para obtener más información sobre cómo usar la API BatchPutAssetPropertyValue para ingerir datos, consulte [Ingiera datos con AWS IoT SiteWise APIs](#).

Para obtener más información acerca de otras formas de ingerir datos, consulte [Ingiera datos para AWS IoT SiteWise](#).

AWS IoT SiteWise Asistente

Note

La función SiteWise Supervisar ya no está disponible para los nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte [SiteWise Supervisar el cambio de disponibilidad](#).

El AWS IoT SiteWise Asistente es un asistente generativo impulsado por IA. Permite a los usuarios, como los gerentes de planta, los ingenieros de calidad y los técnicos de mantenimiento, obtener información, resolver problemas y tomar medidas directamente a partir de sus datos operativos y empresariales.

El AWS IoT SiteWise Asistente consolida la información de AWS IoT los datos, los modelos de activos, los manuales y la documentación en resúmenes comprensibles de eventos críticos. También permite sesiones interactivas de preguntas y respuestas exhaustivas para facilitar el diagnóstico, la exploración de las causas fundamentales y las recomendaciones guiadas.

Temas

- [Configura el asistente AWS IoT SiteWise](#)
- [Crear un conjunto de datos](#)
- [Edita un conjunto de datos](#)
- [Eliminar un conjunto de datos.](#)
- [AWS IoT SiteWise Preguntas de asistente](#)

Configura el asistente AWS IoT SiteWise

Note

La función SiteWise Supervisar ya no está disponible para los nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte [SiteWise Supervisar el cambio de disponibilidad](#).

AWS IoT SiteWise Configuración del asistente

1. Inicie sesión en la [consola de AWS IoT SiteWise](#).

Note

Otorgue permisos para permitir la integración con el AWS IoT TwinMaker servicio. Esto es necesario para que el AWS IoT SiteWise Asistente y el panel ejecuten consultas SQL en AWS IoT SiteWise los recursos. Consulte [Integrar AWS IoT SiteWise y AWS IoT TwinMaker](#).

Grant permission

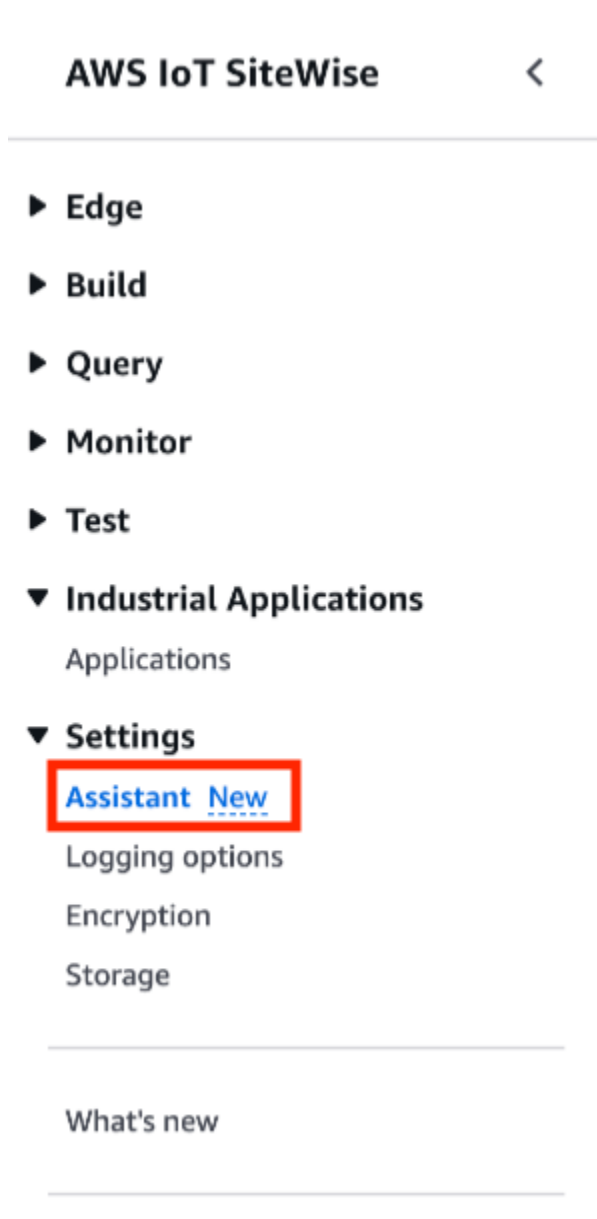


In order to better organize and model industrial data, AWS IoT SiteWise requires permission to integrate with AWS IoT TwinMaker. Choose **Grant permission** to enable this integration. This integration could incur costs. For more information, see [pricing](#).

Cancel

Grant permission

2. Seleccione Asistente en el panel de navegación izquierdo.



Crear un conjunto de datos

Note

La función SiteWise Supervisar ya no está disponible para los nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte [SiteWise Supervisar el cambio de disponibilidad](#).

Note

El AWS IoT SiteWise Asistente debe usar un conjunto de datos con un índice de [Amazon Kendra](#) para obtener conocimientos y orientación a nivel empresarial. Si no tiene un índice de Amazon Kendra, consulte [Creación de un índice](#) para crear uno. Añadir un [conjunto de datos](#) mejora la calidad de la respuesta del Asistente y minimiza las alucinaciones.

Console

Cree un conjunto de datos en la consola AWS IoT SiteWise

1. Los conjuntos de datos se muestran en la sección Conjuntos de datos de la página del AWS IoT SiteWise Asistente.
2. Si no existe ningún conjunto de datos, elija Crear conjunto de datos.
3. En la página de detalles del conjunto de datos, elija un índice de Kendra en el menú desplegable para asociarlo al conjunto de datos.
4. El nombre del conjunto de datos se rellena con el índice de Kendra seleccionado en el paso 3. Edite el nombre si es necesario.
5. (Opcional) La descripción del conjunto de datos se completa con el índice de Kendra seleccionado en el paso 3. Edite la descripción si es necesario.
6. En la sección Permisos, elige una de las siguientes opciones:
 - a. Elija Crear y utilizar un nuevo rol de servicio. De forma predeterminada, crea AWS IoT SiteWise automáticamente un rol de servicio. Esta función permite al AWS IoT SiteWise Asistente acceder a los índices de Kendra.
 - b. Elija Utilizar un rol de servicio existente y, a continuación, el rol deseado.
7. Seleccione Crear.

[AWS IoT SiteWise](#) > [Assistant](#) > Create dataset

Create dataset

Create a dataset for the Assistant.

Dataset details [Info](#)

Kendra index

Select a Kendra index for the Assistant dataset.

test-index

[Amazon Kendra](#)



Dataset name

Dataset name is pre-populated from the Kendra index selected. You can modify the dataset name.

Dataset1

Dataset name must be 1-256 characters. Valid characters are a-z, A-Z, 0-9, _ (underscore) and - (hyphen).

Dataset description - optional

Dataset description is pre-populated from the Kendra index selected. You can modify the dataset description.

Dataset for AWS IoT SiteWise Assistant

Dataset description must be 1-2048 characters.

Permissions

Assistant must have permissions to access the data. To create a custom role, visit the [IAM console](#)

Choose a method to authorize Assistant

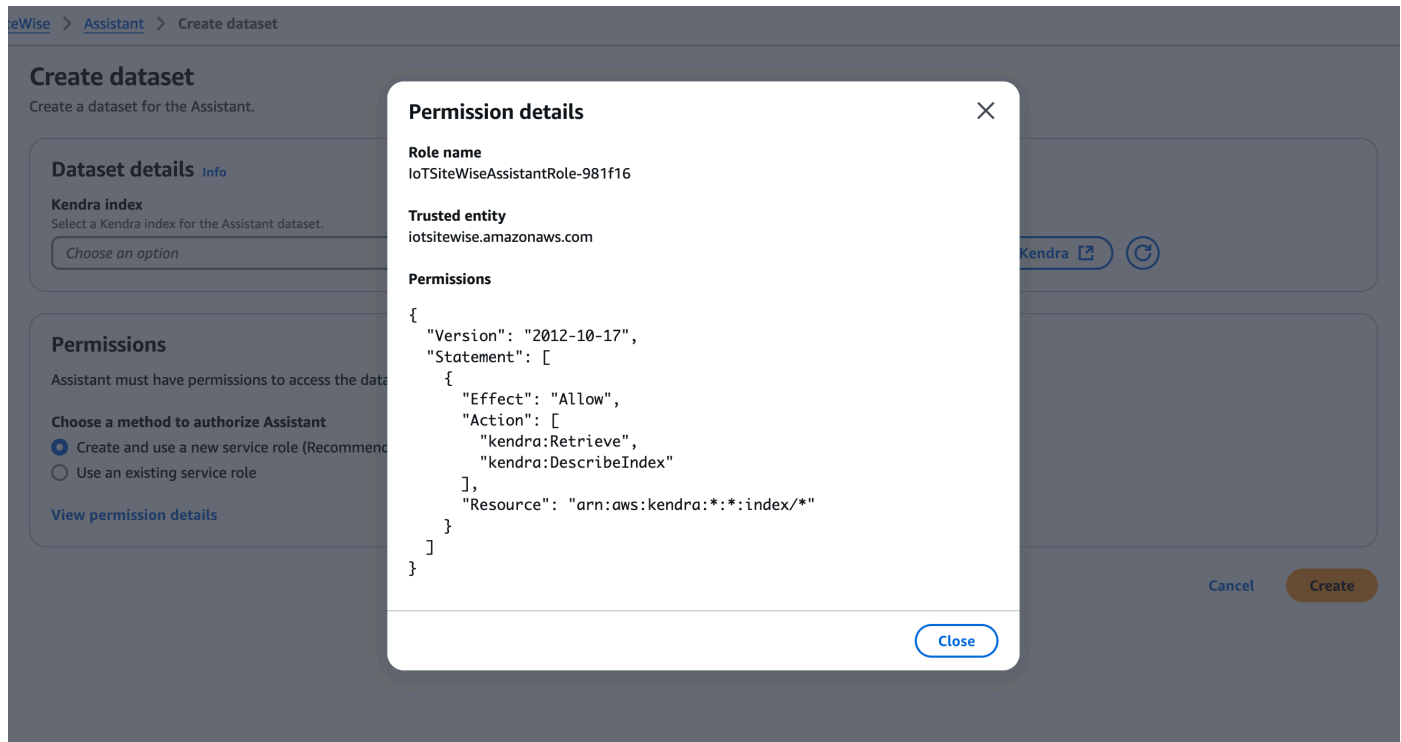
- Create and use a new service role - *recommended*
- Use an existing service role

[View permission details](#)

[Cancel](#)

[Create](#)

El rol de servicio creado AWS IoT SiteWise por el usuario, si el usuario eligió crear y usar un nuevo rol de servicio.



AWS CLI

Cree un conjunto de datos en AWS CLI

1. Cree un rol de IAM que se utilice para crear un conjunto de datos. Utilice la siguiente política de permisos:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kendra:Retrieve"
      ],
      "Resource": "arn:aws:kendra:*:*:index/*"
    }
  ]
}
```

Utilice la siguiente relación de confianza:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "iotsitewise.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

2. Cree un archivo create-dataset.json con la plantilla proporcionada en el ejemplo. Rellene y conecte con este datasetId conjunto de kendra knowledgeBaseArn datos. roleArn

```
{
  "datasetId": "<UUID>",
  "datasetName": "DatasetForAssistant",
  "datasetSource": {
    "sourceType": "KENDRA",
    "sourceFormat": "KNOWLEDGE_BASE",
    "sourceDetail": {
      "kendra": {
        "knowledgeBaseArn": "arn:aws:kendra::%s:index/index",
        "roleArn": "arn:aws:iam::%s:role/role"
      }
    }
  }
}
```

3. Cree el conjunto de datos con el siguiente comando:

```
aws iotsitewise create-dataset --cli-input-json file://create-dataset.json --  
region us-east-1
```

Edita un conjunto de datos

Note

La función SiteWise Monitor ya no está disponible para los nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte [SiteWise Supervisar el cambio de disponibilidad](#).

Console

Editar un conjunto de datos

1. Los conjuntos de datos se muestran en la sección Conjuntos de datos de la página del Asistente. Elija un conjunto de datos para editarlo. Selecciona Editar para empezar a editar.
2. En la página de detalles del conjunto de datos, elija un índice de Kendra en el menú desplegable para asociarlo al conjunto de datos.
3. El nombre del conjunto de datos se rellena con el índice de Kendra seleccionado en el paso 2. Edite el nombre si es necesario.
4. (Opcional) La descripción del conjunto de datos se completa con el índice de Kendra seleccionado en el paso 2. Edite la descripción si es necesario.
5. En la sección Permisos, elige una de las siguientes opciones:
 - a. Elija Crear y utilizar un nuevo rol de servicio. De forma predeterminada, crea AWS IoT SiteWise automáticamente un rol de servicio. Esta función permite al AWS IoT SiteWise Asistente acceder a los índices de Kendra.
 - b. Elija Utilizar un rol de servicio existente y, a continuación, el rol deseado.
6. Seleccione Guardar cambios para guardar la selección.

[AWS IoT SiteWise](#) > [Assistant](#) > Edit dataset

Edit dataset

Edit a dataset for the Assistant.

Dataset details [Info](#)

Kendra index

Select a Kendra index for the Assistant dataset.

test-index ▼ Amazon Kendra [↗](#) 🔄

Dataset name

Dataset name is pre-populated from the Kendra index selected. You can modify the dataset name.

Dataset1

Dataset name must be 1-256 characters. Valid characters are a-z, A-Z, 0-9, _ (underscore) and - (hyphen).

Dataset description - optional

Dataset description is pre-populated from the Kendra index selected. You can modify the dataset description.

Dataset for AWS IoT SiteWise Assistant

Dataset description must be 1-2048 characters.

Permissions

Assistant must have permissions to access the data. To create a custom role, visit the [IAM console](#)

Choose a method to authorize Assistant

- Create and use a new service role - *recommended*
- Use an existing service role

Existing role

IoTSiteWiseAssistantRole-40f994 ▼

[View the IoTSiteWiseAssistantRole-40f994 role in the IAM console](#)

Cancel

Save changes

AWS CLI

Edite un conjunto de datos en AWS CLI

1. Cree un archivo update-dataset.json con la plantilla proporcionada en el ejemplo. Rellene y conecte con este datasetId conjunto de kendra knowledgeBaseArn datos. roleArn

```
{
  "datasetId": "<UUID>",
  "datasetName": "DatasetForAssistant",
  "datasetSource": {
    "sourceType": "KENDRA",
    "sourceFormat": "KNOWLEDGE_BASE",
    "sourceDetail": {
      "kendra": {
        "knowledgeBaseArn": "arn:aws:kendra::%s:index/index",
        "roleArn": "arn:aws:iam::%s:role/role"
      }
    }
  }
}
```

```
}  
  }  
} }
```

2. Actualice el conjunto de datos con el siguiente comando:

```
aws iotsitewise update-dataset --cli-input-json file://update-dataset.json --  
region us-east-1
```

Eliminar un conjunto de datos.

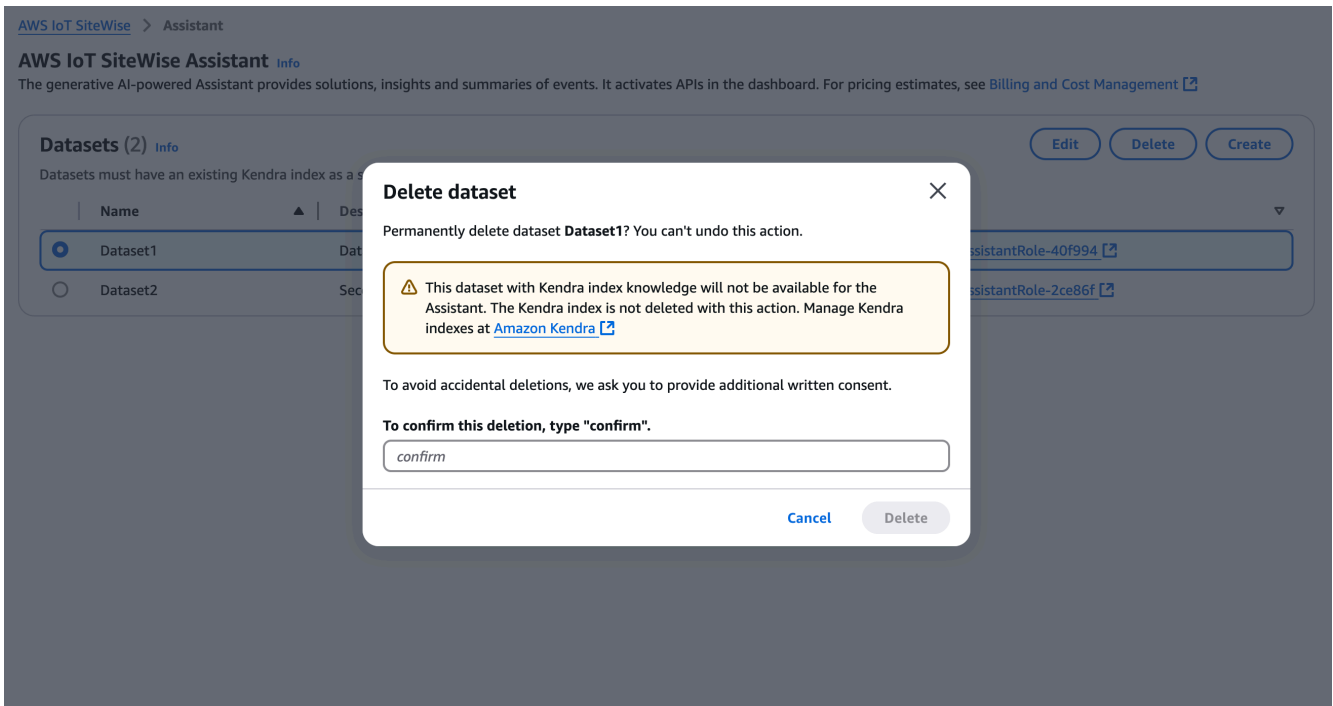
Note

La función SiteWise Monitor ya no está disponible para los nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte [SiteWise Supervisar el cambio de disponibilidad](#).

Console

Eliminar un conjunto de datos.

1. Los conjuntos de datos se muestran en la sección Conjuntos de datos de la página del Asistente. Elige un conjunto de datos. Elija Eliminar.
2. Escriba confirm en la ventana emergente para confirmar la eliminación.



3. Elija Eliminar.

AWS CLI

Eliminar un conjunto de datos.

- Elimine el conjunto de datos con `datasetId`.

```
aws iotsitewise delete-dataset --region us-east-1 --dataset-id <UUID>
```

AWS IoT SiteWise Preguntas de asistente

Note

La función SiteWise Monitor ya no está disponible para los nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte [SiteWise Supervisar el cambio de disponibilidad](#).

Consulte [Ejemplos de preguntas para hacerle al Asistente AWS IoT SiteWise](#) para obtener más información sobre el AWS IoT SiteWise Asistente de consultas.

Supervise los datos con AWS IoT SiteWise Monitor

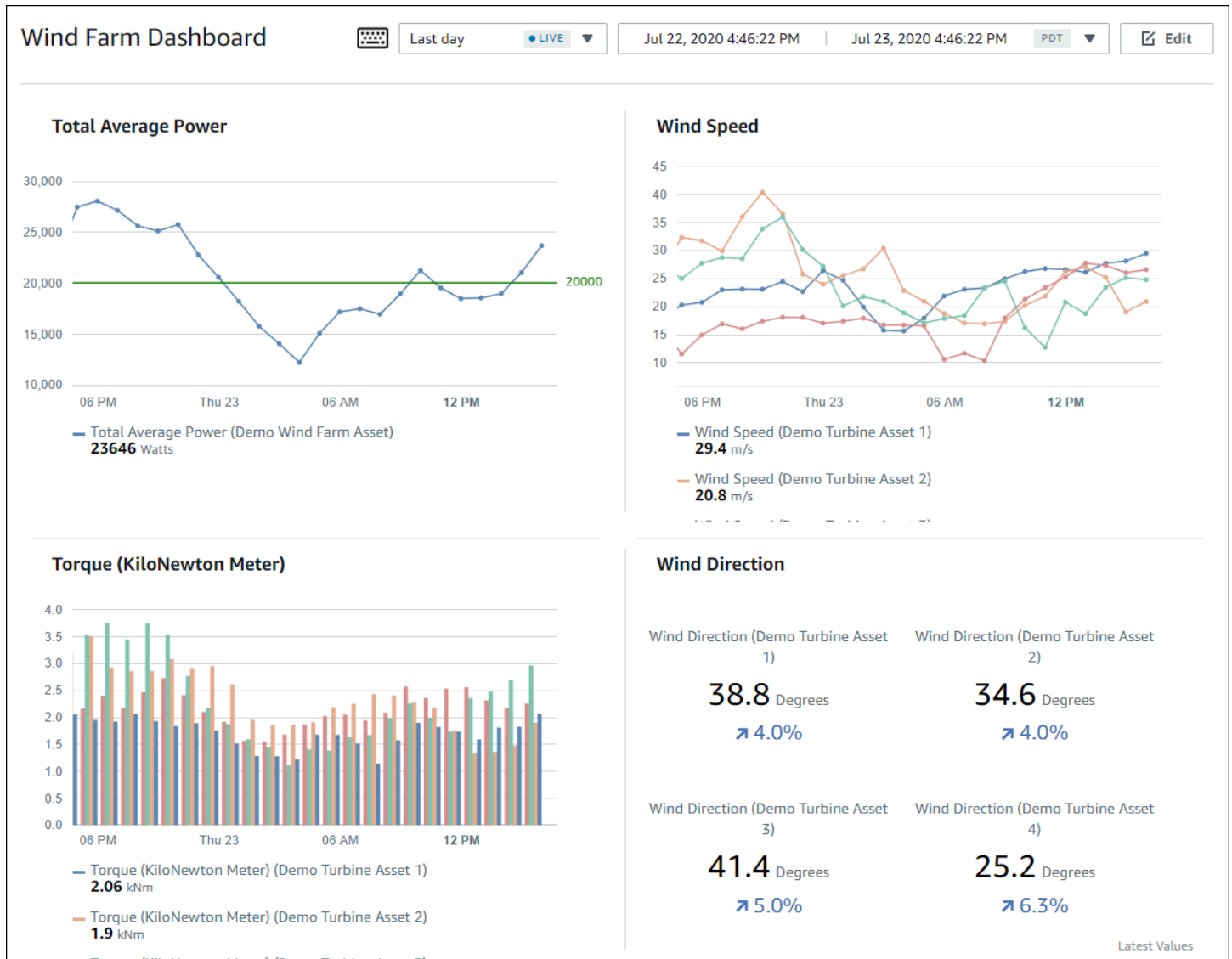
Note

La función SiteWise Monitor ya no está disponible para los nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte [SiteWise Supervisar el cambio de disponibilidad](#).

Puede utilizarlos AWS IoT SiteWise para supervisar los datos de sus procesos, dispositivos y equipos mediante la creación de portales web de SiteWise Monitor. SiteWise Monitor es una función AWS IoT SiteWise que puede utilizar para crear portales en forma de una aplicación web gestionada. A continuación, puede utilizar estos portales para ver y compartir sus datos operativos. Puede crear proyectos con paneles para visualizar datos de los procesos, dispositivos y equipos a los que están conectados AWS IoT.

Los expertos de dominio, como los ingenieros de procesos, pueden utilizar estos portales para obtener rápidamente información sobre sus datos operativos y comprender el comportamiento de los dispositivos y equipos.

A continuación se muestra un tablero de instrumentos de ejemplo que muestra los datos de un conjunto de molinos eólicos.



Como AWS IoT SiteWise captura datos a lo largo del tiempo, puede usar SiteWise Monitor para ver los datos operativos a lo largo del tiempo o los últimos valores informados en momentos específicos. Esto le permite descubrir ideas que de otro modo podrían ser difíciles de encontrar.

SiteWise Supervise las funciones

Cuatro funciones interactúan con SiteWise Monitor:

AWS administrador

El AWS administrador usa la AWS IoT SiteWise consola para crear portales. El administrador de AWS también puede asignar administradores del portal y agregar usuarios del portal.

Posteriormente, los administradores del portal asignan usuarios del portal a proyectos como propietarios o lectores. El AWS administrador trabaja exclusivamente en la AWS consola.

Administrador del portal

Cada portal de SiteWise Monitor tiene uno o más administradores de portal. Los administradores del portal utilizan el portal para crear proyectos que contengan recopilaciones de activos y paneles. A continuación, el administrador del portal asigna activos y propietarios a cada proyecto. Al controlar el acceso al proyecto, los administradores del portal especifican los activos que los propietarios y observadores de proyectos pueden ver.

Propietario del proyecto

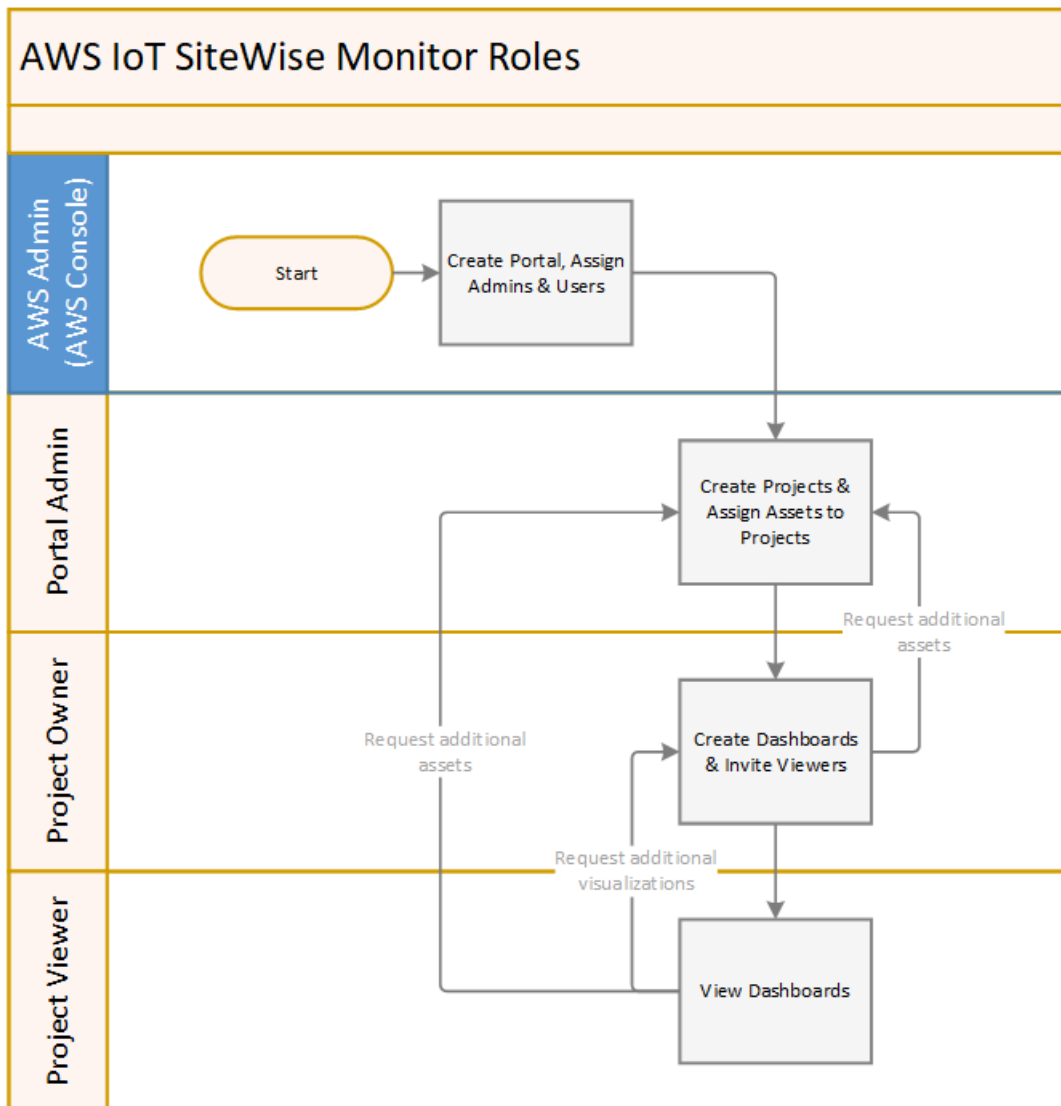
Cada proyecto de SiteWise Monitor tiene propietarios. Los propietarios de proyectos crean visualizaciones en forma de paneles para representar los datos operativos de manera coherente. Cuando los paneles están listos para compartirse, el propietario del proyecto puede invitar a lectores al proyecto. Los propietarios de proyectos también pueden asignar otros propietarios al proyecto. Los propietarios del proyecto pueden configurar los umbrales y los ajustes de notificación de las alarmas.

Observador de proyectos

Cada proyecto de SiteWise Monitor tiene visores. Los observadores de proyectos pueden conectarse al portal para ver los paneles creados por los propietarios de proyectos. En cada panel de control, los observadores de proyectos pueden ajustar el intervalo de tiempo para comprender mejor los datos operativos. Los observadores de proyectos solo pueden ver los paneles de los proyectos a los que tienen acceso. Los observadores de proyectos pueden confirmar y posponer alarmas.

En función de la organización, la misma persona puede desempeñar varios roles.

La siguiente imagen ilustra cómo interactúan estas cuatro funciones en el portal SiteWise Monitor.



Puede administrar quién tiene acceso a sus datos utilizando AWS IAM Identity Center o IAM. Sus usuarios de datos pueden iniciar sesión en SiteWise Monitor desde un navegador de escritorio o móvil con sus credenciales de IAM Identity Center o de IAM.

Federación de SAML

El Centro de identidades de IAM y IAM admiten la federación de identidades con [SAML \(lenguaje de marcado de aserción de seguridad\) 2.0](#). SAML 2.0 es un estándar abierto que utilizan muchos proveedores de identidad externos (IdPs) para autenticar a los usuarios y transmitir su información de identidad y seguridad a los proveedores de servicios (SPs). SPs suelen ser aplicaciones o servicios. La federación SAML permite a los administradores y usuarios del portal SiteWise Monitor iniciar sesión en los portales asignados con credenciales externas, como sus nombres de usuario y contraseñas corporativas.

Puede configurar IAM Identity Center e IAM para que utilicen una federación basada en SAML para acceder a sus portales de Monitor. SiteWise

IAM Identity Center

Los administradores y usuarios del portal pueden iniciar sesión en el portal de AWS acceso con sus nombres de usuario y contraseñas corporativos. A continuación, pueden navegar hasta los portales de SiteWise Monitor que tengan asignados. El Centro de Identidad de IAM utiliza certificados para establecer una relación de confianza SAML entre su proveedor de identidad y. AWS Para obtener más información, consulte [Implementación del perfil SCIM y de SAML 2.0](#) en la Guía del usuario de AWS IAM Identity Center .

IAM

Los administradores y usuarios del portal pueden solicitar credenciales de seguridad temporales para acceder a los portales de SiteWise Monitor asignados. Debe crear una identidad de proveedor de identidades SAML en IAM para establecer una relación de confianza entre su proveedor de identidades y. AWS Para obtener más información, consulte [Uso de la federación basada en SAML para el acceso mediante API AWS](#), en la Guía del usuario de IAM.

Los administradores y usuarios del portal pueden iniciar sesión en el portal de su empresa y seleccionar la opción de ir a la AWS consola de administración. A continuación, pueden navegar hasta los portales de SiteWise Monitor que tengan asignados. El portal de su empresa gestiona el intercambio de confianza entre su proveedor de identidad y AWS. Para obtener más información, consulte [Permitir que los usuarios federados de SAML 2.0 accedan a la consola de AWS administración en la](#) Guía del usuario de IAM.

Note

Al añadir usuarios o administradores al portal, evite crear políticas de IAM que restrinjan los permisos de los usuarios, como IP limitada. Las políticas adjuntas con permisos restringidos no podrán conectarse al portal. AWS IoT SiteWise

SiteWise Conceptos de monitoreo

Para utilizar SiteWise Monitor, debe estar familiarizado con los siguientes conceptos:

Portal

Un AWS IoT SiteWise Monitor portal es una aplicación web que puede utilizar para visualizar y compartir sus AWS IoT SiteWise datos. Un portal tiene uno o varios administradores y contiene cero o más proyectos.

Proyecto

Cada portal de SiteWise Monitor contiene un conjunto de proyectos. Cada proyecto tiene un subconjunto de sus activos de AWS IoT SiteWise asociados al mismo. Los propietarios de proyectos crean uno o varios paneles para proporcionar una forma coherente de ver los datos asociados a esos activos. Los propietarios del proyecto pueden invitar a los lectores al proyecto para permitirles ver los activos y paneles del proyecto. El proyecto es la unidad básica para compartir en SiteWise Monitor. Los propietarios del proyecto pueden invitar a los usuarios a los que el AWS administrador les dio acceso al portal. Un usuario debe tener acceso a un portal antes de que un proyecto de ese portal pueda compartirse con ese usuario.

Activo

Cuando se ingieren datos AWS IoT SiteWise de su equipo industrial, cada uno de sus dispositivos, equipos y procesos se representa como activos. Cada activo tiene propiedades y alarmas asociadas al mismo. El administrador del portal asigna conjuntos de activos a cada proyecto.

Propiedad

Las propiedades son datos de serie temporal asociados a los activos. Por ejemplo, un equipo podría tener un número de serie, una ubicación, una marca y un modelo y una fecha de instalación. También puede tener valores de series temporales para disponibilidad, rendimiento, calidad, temperatura, presión, etc.

Alarma

Las alarmas monitorean las propiedades para identificar cuando el equipo está fuera de su rango de operación. Cada alarma define un umbral y una propiedad por monitorear. Cuando la propiedad supera el umbral, la alarma se activa e indica que usted o alguien de su equipo debe ocuparse del problema. Los propietarios de proyectos pueden personalizar los umbrales y los ajustes de notificación de las alarmas. Los observadores de proyectos pueden confirmar y posponer las alarmas y pueden dejar un mensaje con detalles sobre la alarma o la acción que realizaron para solucionarla.

Panel de control

Cada proyecto contiene un conjunto de paneles. Los paneles proporcionan un conjunto de visualizaciones para los valores de un conjunto de activos. Los propietarios de proyectos crean los paneles y las visualizaciones que contiene. Cuando un propietario de proyecto está listo para compartir el conjunto de paneles, el propietario puede invitar a observadores al proyecto, lo que les da acceso a todos los paneles del mismo. Si desea un conjunto distinto de observadores para distintos paneles, debe dividir los paneles entre proyectos. Cuando los observadores consultan paneles, pueden personalizar el intervalo de tiempo para buscar datos específicos.

Visualización

En cada panel de control, los propietarios de proyectos deciden cómo mostrar las propiedades y alarmas de los activos asociados al proyecto. La disponibilidad puede representarse como un gráfico de líneas, mientras que otros valores pueden mostrarse como gráficos de barras o indicadores clave de rendimiento (KPIs). Las alarmas se visualizan mejor como cuadrículas de estado y líneas temporales de estado. Los propietarios de proyectos personalizan cada visualización para ofrecer la mejor comprensión de los datos de ese activo.

Comience con AWS IoT SiteWise Monitor (Classic)

Note

La función SiteWise Supervisar ya no está disponible para los nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte [SiteWise Supervisar el cambio de disponibilidad](#).

Si es el AWS administrador de su organización, puede crear portales desde la AWS IoT SiteWise consola. Complete los siguientes pasos para crear un portal para que los miembros de su organización puedan ver sus AWS IoT SiteWise datos:

1. Configure y cree un portal
2. Agregar administradores del portal y enviar correos electrónicos de invitación
3. Agregar usuarios al portal

Tras crear un portal, el administrador del portal puede ver sus AWS IoT SiteWise activos y asignarlos a los proyectos del portal. Los propietarios de proyectos pueden crear paneles para visualizar

las propiedades de los activos que ayudan a los observadores de proyectos a comprender el rendimiento de sus dispositivos, procesos y equipos.

Note

Al añadir usuarios o administradores al portal, evite crear políticas AWS Identity and Access Management (de IAM) que restrinjan los permisos de los usuarios, como una IP limitada. Las políticas adjuntas con permisos restringidos no podrán conectarse al AWS IoT SiteWise portal.

Puede seguir un tutorial que recorra los pasos necesarios para configurar un portal con un proyecto, paneles y varios usuarios para un escenario específico utilizando datos de parques eólicos. Para obtener más información, consulte [Visualice y comparta datos de parques eólicos en SiteWise Monitor](#).

Temas

- [Cree un portal en SiteWise Monitor](#)
- [Configure su portal en SiteWise Monitor](#)
- [Invite a los administradores a SiteWise Monitor](#)
- [Añada usuarios del portal en Monitor SiteWise](#)
- [Crear AWS IoT SiteWise paneles \(AWS CLI\)](#)
- [Active las alarmas de sus portales en AWS IoT SiteWise](#)
- [Habilite su AWS IoT SiteWise portal en la periferia](#)
- [Administre sus portales de SiteWise Monitor](#)

Cree un portal en SiteWise Monitor

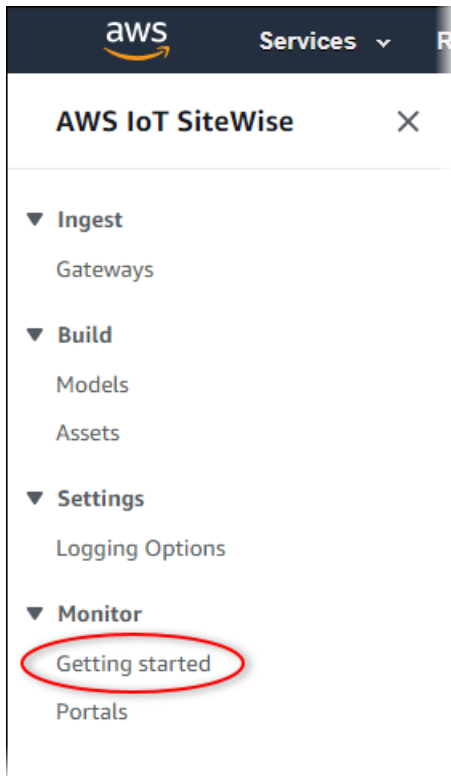
Note

La función SiteWise Supervisar ya no está disponible para los nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte [SiteWise Supervisar el cambio de disponibilidad](#).

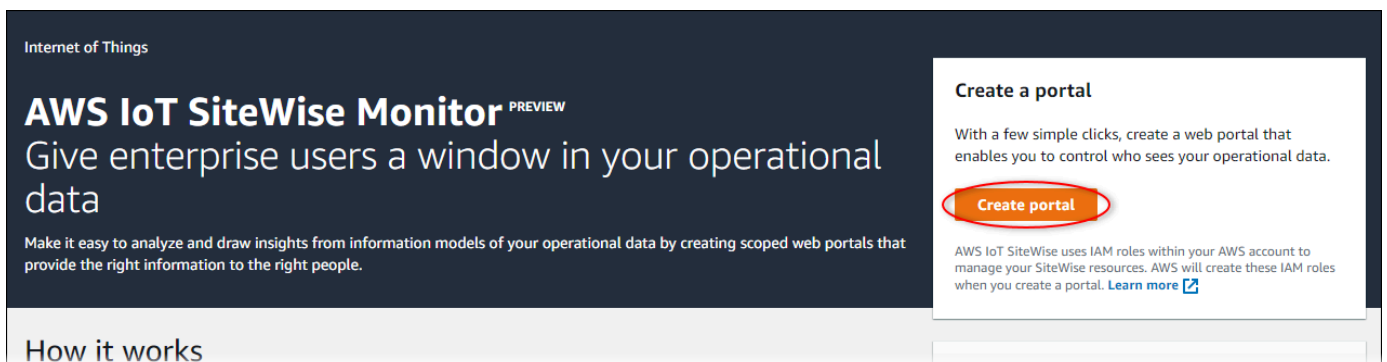
Puede crear un portal de SiteWise monitoreo en la AWS IoT SiteWise consola.

Para crear un portal

1. Inicie sesión en la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, elija Monitor, luego Introducción.



3. Elija Crear portal.



A continuación, debe proporcionar información básica para configurar el portal.

Configure su portal en SiteWise Monitor

Note

La función SiteWise Supervisar ya no está disponible para los nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte [SiteWise Supervisar el cambio de disponibilidad](#).

Sus usuarios utilizan los portales para ver sus datos. Puede personalizar el nombre, la descripción, la marca, la autenticación de usuario, el correo electrónico de contacto de asistencia y los permisos de un portal.

AWS IoT SiteWise > Monitor > Portals > Create portal

Step 1
Portal configurationStep 2 - optional
Additional featuresStep 3
Invite administratorsStep 4
Assign users

Portal configuration

Each web portal provides enterprise users with access to your IoT SiteWise assets. [Learn more](#)

Portal details

Portal name

Choose a portal name to identify the web portal to your users. Company name is recommended.

example-factory-1

Name should be 1-128 characters and only contain A-Z a-z 0-9 _ and -.

Description - optional

Create a description of your portal

Example Corp Factory #1 in Renton, WA

Description should contain a maximum of 2048 characters.

Portal branding

You can provide your logo image to display your brand in this web portal.

Logo image


Upload a square, high-resolution .png file. The image is displayed on a dark background.

Choose file

The file size must be less than 1 MB.

User authentication

Your users can sign in to this portal with their AWS Single Sign-On (AWS SSO) or AWS Identity and Access Management (IAM) credentials. If you choose AWS SSO, you must enable the service for your AWS account.

 You haven't enabled AWS SSO in your account yet. When you create your first portal user, this automatically enables AWS SSO in your AWS account.

[Create user](#)

AWS SSO

Your users can sign in to the portal with their corporate usernames and passwords.

IAM

Your users can sign in to the portal with their IAM credentials.

Support contact email

You can provide an email address for cases where there's a problem or issue with this portal and your users need to contact support to resolve.

Email

support@example.com

Tags

This resource doesn't have any tags.

[Add tag](#)

You can add up to 50 more tags.

Permissions

SiteWise Monitor assumes this role to give permissions to your federated users to access AWS IoT SiteWise resources. [Learn](#)

Para configurar un portal

1. Escriba un nombre para el portal.
2. (Opcional) Escriba una descripción para el portal. Si tiene varios portales, utilice descripciones significativas para ayudarlo a realizar un seguimiento de lo que contiene cada portal.
3. (Opcional) Suba una imagen para mostrar su marca en el portal. Elija una imagen PNG cuadrada. Si carga una imagen no cuadrada, el portal escala la imagen a un cuadrado.
4. Seleccione una de las siguientes opciones:
 - Elija Centro de identidades de IAM si los usuarios de su portal inician sesión en este portal con sus nombres de usuario y contraseñas corporativos.

Si no ha habilitado el Centro de identidades de IAM en su cuenta, haga lo siguiente:

- a. Seleccione la opción Crear usuario.
- b. En la página Crear usuario, para crear el primer portal, introduzca la dirección de correo electrónico, nombre y apellido del usuario y, a continuación, seleccione Crear usuario.

Create user [X]

When you create your first portal user, this automatically enables AWS SSO in your AWS account.

Email address
janedoe@example.com


First name: Jane Last name: Doe

Cancel **Create user**

Note


- AWS habilita automáticamente el Centro de Identidad de IAM en su cuenta al crear el primer usuario del portal.
- Solo puede configurar el Centro de identidades de IAM en una región a la vez. SiteWise El monitor se conecta a la región que configuró para el IAM Identity Center. Esto significa que utiliza una región para el acceso al Centro de identidades de IAM, pero puede crear portales en cualquier región.

- Elija IAM si los usuarios de su portal inician sesión en este portal con sus credenciales de IAM.

 Important

Los usuarios o roles deben tener el permiso de `iotsitewise:DescribePortal` para iniciar sesión en el portal.

5. Introduzca una dirección de correo electrónico con la que los usuarios del portal puedan ponerse en contacto cuando tengan un problema con el portal y necesiten ayuda para resolverlo.
6. (Opcional) Agregue etiquetas para su portal. Para obtener más información, consulte [Etiquete sus AWS IoT SiteWise recursos](#).
7. Seleccione una de las siguientes opciones:
 - Elija Crear y utilizar un nuevo rol de servicio. De forma predeterminada, SiteWise Monitor crea automáticamente un rol de servicio para cada portal. Esta función permite a los usuarios del portal acceder a sus AWS IoT SiteWise recursos. Para obtener más información, consulte [Utilice los roles de servicio para AWS IoT SiteWise Monitor](#).
 - Elija Utilizar un rol de servicio existente y, a continuación, el rol deseado.
8. Seleccione Siguiente
9. (Opcional) Habilite alarmas para su portal. Para obtener más información, consulte [Active las alarmas de sus portales en AWS IoT SiteWise](#).
10. Elija Crear. AWS IoT SiteWise creará tu portal.

 Note

Si cierra la consola, puede finalizar el proceso de instalación agregando administradores y usuarios. Para obtener más información, consulte [Agregue o elimine administradores del portal en AWS IoT SiteWise](#). Si no desea conservar este portal, elimínelo para que no consuma recursos. Para obtener más información, consulte [Eliminar un portal en AWS IoT SiteWise](#).

La columna Estado puede adoptar uno de los siguientes valores.

- **CREANDO:** AWS IoT SiteWise está procesando su solicitud para crear el portal. Este proceso puede tardar varios minutos en completarse.
- **ACTUALIZACIÓN:** AWS IoT SiteWise está procesando su solicitud de actualización del portal. Este proceso puede tardar varios minutos en completarse.
- **PENDIENTE:** AWS IoT SiteWise está esperando a que finalice la propagación del registro DNS. Este proceso puede tardar varios minutos en completarse. Durante el estado PENDIENTE puede eliminar el portal.
- **ELIMINAR:** AWS IoT SiteWise está procesando su solicitud para eliminar el portal. Este proceso puede tardar varios minutos en completarse.
- **ACTIVO:** cuando el portal se activa, los usuarios de su portal tienen acceso al mismo.
- **ERROR:** no se ha AWS IoT SiteWise podido procesar tu solicitud para crear, actualizar o eliminar el portal. Si has habilitado AWS IoT SiteWise el envío de registros a Amazon CloudWatch Logs, puedes usar estos registros para solucionar problemas. Para obtener más información, consulte [Supervisión AWS IoT SiteWise con CloudWatch registros](#).

Aparece un mensaje cuando se crea el portal.



Successfully created portal URL at <https://a1b2c3d4-5678-90ab-cdef-11111EXAMPLE.app.iotsitewise.aws>

A continuación, debe invitar a uno o más administradores del portal al portal. Hasta ahora, ha creado un portal pero nadie puede acceder a él.

Invite a los administradores a SiteWise Monitor

Para comenzar en el nuevo portal, debe asignar un administrador del portal. El administrador del portal crea proyectos, elige los propietarios de proyectos y asigna activos a los proyectos. Los administradores del portal pueden ver todos sus AWS IoT SiteWise activos.

En función del servicio de autenticación de usuarios, elija una de las siguientes opciones:

IAM Identity Center

Si es la primera vez que utiliza SiteWise Monitor, puede elegir al usuario que creó anteriormente como administrador del portal. Si desea añadir otro usuario como administrador del portal, puede crear un usuario del Centro de identidades de IAM desde esta página. Si lo desea, puede conectar un proveedor de identidades externo al Centro de identidades de IAM. Para obtener más información, consulte la [Guía del usuario de AWS IAM Identity Center](#).

Para invitar a administradores

1. Active las casillas de verificación de los usuarios que quiera que sean administradores del portal. Esto añade los usuarios a la lista de Administradores del portal.

Note

Si utiliza el Centro de identidades de IAM como almacén de identidades y ha iniciado sesión en su cuenta de administración de AWS Organizations , puede elegir Crear usuario para crear un usuario del Centro de identidades de IAM. El Centro de identidades de IAM enviará al nuevo usuario un correo electrónico para que establezca su contraseña. A continuación, puede asignar el usuario al portal como administrador. Para obtener más información, consulte [Administrar identidades en el IAM Identity Center](#).

2. (Opcional) Elija Enviar invitación a los usuarios seleccionados. Se abrirá el cliente de correo electrónico y se rellenará una invitación en el cuerpo del mensaje.

Puede personalizar el correo electrónico antes de enviarlo a los administradores del portal. También puede enviar el correo electrónico a los administradores de su portal más tarde. Si es la primera vez que prueba SiteWise Monitor y añade su nuevo usuario o rol del Centro de Identidad de IAM o de IAM como administrador del portal, no necesita enviarse un correo electrónico.

3. Si añade un usuario que no desea como administrador, desactive la casilla de verificación correspondiente a ese usuario.
4. Cuando haya terminado de invitar a los administradores del portal, seleccione Siguiente.


IAM

Puede elegir un usuario o rol para que sea el administrador del portal. Si desea añadir otro usuario o rol como administrador del portal, puede crear un usuario o rol en la consola de IAM. Para obtener más información, consulte [Creación de un usuario de IAM en su cuenta de AWS](#) y [Creación de roles de IAM](#) en la Guía del usuario de IAM.


Para invitar a administradores

1. Haga lo siguiente:

- Elija Usuarios de IAM para añadir un usuario de IAM como administrador de su portal.
 - Elija Roles de IAM para añadir un rol de IAM como administrador de su portal.
2. Marque las casillas de verificación de los usuarios o roles que desee como administradores de su portal. Esto añade los usuarios o roles a la lista de Administradores del portal.
 3. Si añade un usuario o rol que no desea como administrador, desmarque la casilla de verificación correspondiente a ese usuario o rol.
 4. Cuando haya terminado de invitar a los administradores del portal, seleccione Siguiente.


 Important

Los usuarios o roles deben tener el permiso de `iotsitewise:DescribePortal` para iniciar sesión en el portal.

 Note

Si utiliza el Centro de identidades de IAM como almacén de identidades y ha iniciado sesión en su cuenta de administración de AWS Organizations , puede elegir Crear usuario para crear un usuario del Centro de identidades de IAM. El Centro de identidades de IAM enviará al nuevo usuario un correo electrónico para que establezca su contraseña. A continuación, puede asignar el usuario al portal como administrador. Para obtener más información, consulte [Administrar identidades en el IAM Identity Center](#).

Puede cambiar la lista de administradores del portal más adelante. Para obtener más información, consulte [Agregue o elimine administradores del portal en AWS IoT SiteWise](#).

 Note

Dado que solo un administrador del portal puede crear proyectos y asignarles activos, debe especificar al menos un único administrador del portal.

Como último paso, añada usuarios que pueden acceder al nuevo portal.

Añada usuarios del portal en Monitor SiteWise

Note

La función SiteWise Supervisar ya no está disponible para los nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte [SiteWise Supervisar el cambio de disponibilidad](#).

Puede controlar qué usuarios tienen acceso a sus portales. En cada portal, los administradores del portal crean uno o varios proyectos y asignan usuarios del portal como propietarios o lectores para cada proyecto. Cada propietario del proyecto puede invitar a usuarios adicionales del portal para que sean propietarios del proyecto o lo vean.

En función del servicio de autenticación de usuarios, elija una de las siguientes opciones:

IAM Identity Center

Si desea añadir un usuario a la lista Usuarios, realice los pasos siguientes.

Para añadir usuarios del portal

1. Elija usuarios en la lista Usuarios para añadirlos al portal. Esto añade los usuarios a la lista Usuarios del portal. Si es la primera vez que usa SiteWise Monitor, no necesita agregar al administrador del portal como usuario del portal.

Note

Si utiliza el Centro de identidades de IAM como almacén de identidades y ha iniciado sesión en su cuenta de administración de AWS Organizations, puede elegir Crear usuario para crear un usuario del Centro de identidades de IAM. El Centro de identidades de IAM enviará al nuevo usuario un correo electrónico para que establezca su contraseña. A continuación, puede asignar el usuario al portal como usuario. Para obtener más información, consulte [Administrar identidades en el IAM Identity Center](#).

2. Si añade un usuario al que no desea tener acceso al portal, desactive la casilla de verificación correspondiente.
3. Cuando haya terminado de seleccionar los usuarios, elija Asignar usuarios.

AWS IoT SiteWise > Monitor > Portals > Create portal

Step 1
Portal configuration

Step 2
Invite administrators

Step 3
Assign users

Assign users

Select the users you want to be able to access and view this portal. Portal administrators will send invitations to these users at a later date. [Learn more](#)

Users (2) Create user

< 1 > ⚙

	Display name	Email
<input type="checkbox"/>	Jane Doe	janedoe@example.com
<input checked="" type="checkbox"/>	John Doe	johndoe@example.com

▶ Selected users (1)

Cancel Previous Assign users

IAM

Si ve al usuario o rol que desea añadir en la lista Usuarios de IAM o Roles de IAM, complete los pasos siguientes.

Para añadir usuarios del portal

1. Realice las siguientes opciones:
 - Elija Usuarios de IAM para añadir un usuario de IAM como usuario del portal.
 - Elija Roles de IAM para añadir un rol de IAM como usuario de portal.

Si es la primera vez que usa SiteWise Monitor, no necesita agregar al administrador del portal como usuario del portal.

2. Marque las casillas de verificación de los usuarios o roles que desee como usuarios del portal. Esto añade los usuarios o roles a la lista Usuarios del portal.
3. Si añade un usuario al que no desea tener acceso al portal, desactive la casilla de verificación correspondiente.
4. Cuando haya terminado de seleccionar los usuarios, elija Asignar usuarios.

⚠ Important

Los usuarios o roles deben tener el permiso de `iotsitewise:DescribePortal` para iniciar sesión en el portal.

AWS IoT SiteWise > Monitor > Portals > Create portal

Step 1
Portal configuration

Step 2
Invite administrators

Step 3
Assign users

Assign users

Select the users you want to be able to access and view this portal. Portal administrators will send invitations to these users at a later date. [Learn more](#)

Users Roles

IAM users (1) [Manage users in IAM console](#)

Find user name

<input checked="" type="checkbox"/>	Name	Date created
<input checked="" type="checkbox"/>	raspberrypi-testing	11-08-2019

▶ **Portal users (1)** [Remove](#)

Cancel Previous **Assign users**

AWS IoT SiteWise > Monitor > Portals > Create portal

Step 1
Portal configuration

Step 2
[Invite administrators](#)

Step 3
Assign users

Assign users

Select the users you want to be able to access and view this portal. Portal administrators will send invitations to these users at a later date. [Learn more](#)

Users **Roles**

IAM roles (66) [Manage roles in IAM console](#)

Find role name

<input type="checkbox"/>	Name	Date created
<input type="checkbox"/>	[REDACTED]	
<input type="checkbox"/>	[REDACTED]	
<input checked="" type="checkbox"/>	AWSIoTSiteWiseMonitorServiceRole_4wZigNpA1	03-16-2021
<input type="checkbox"/>	AWSIoTSiteWiseMonitorServiceRole_EcKT-2Oar	03-11-2021
<input type="checkbox"/>	AWSIoTSiteWiseMonitorServiceRole_GTnd004Wr	03-16-2021
<input type="checkbox"/>	AWSIoTSiteWiseMonitorServiceRole_rHINLNCs-	03-11-2021
<input type="checkbox"/>	AWSIoTSiteWiseMonitorServiceRole_XB330QUIO	03-10-2021
<input type="checkbox"/>	[REDACTED]	
<input type="checkbox"/>	[REDACTED]	
<input type="checkbox"/>	[REDACTED]	

► Portal users (2) [Remove](#)

Cancel Previous **Assign users**

¡Enhorabuena! Ha creado correctamente un portal y ha asignado administradores de portal y usuarios que pueden utilizar ese portal cuando se les invite a hacerlo. Los administradores del portal ahora pueden crear proyectos y agregar activos a esos proyectos. A continuación, los propietarios del proyecto pueden crear paneles para visualizar los datos de los activos de cada proyecto.

Puede cambiar la lista de usuarios del portal más adelante. Para obtener más información, consulte [Agregue o elimine usuarios del portal en AWS IoT SiteWise](#).

Si necesita realizar cambios en el portal, consulte [Administre sus portales de SiteWise Monitor](#).

Para empezar a usar el portal, consulte [Primeros pasos](#) en la Guía de aplicaciones de SiteWise Monitor.

Crear AWS IoT SiteWise paneles (AWS CLI)

Note

La función SiteWise Monitor ya no está disponible para los nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte [SiteWise Supervisar el cambio de disponibilidad](#).

Al definir visualizaciones (o widgets) en los paneles mediante el AWS CLI, debe especificar la siguiente información en el documento `dashboardDefinition` JSON. Esta definición es un parámetro de las operaciones [CreateDashboard](#) y [UpdateDashboard](#).

widgets

Lista de estructuras de definición de widgets que contiene cada una la siguiente información:

type

El tipo de widget. AWS IoT SiteWise proporciona los siguientes tipos de widgets:

- `sc-line-chart`: un gráfico de líneas. Para obtener más información, consulte [Gráficos de líneas](#) en la Guía de la aplicación AWS IoT SiteWise Monitor .
- `sc-scatter-chart`: un gráfico de dispersión. Para obtener más información, consulte [Gráficos de dispersión](#) en la Guía de la aplicación AWS IoT SiteWise Monitor .
- `sc-bar-chart`: un gráfico de barras. Para obtener más información, consulte [Gráficos de barras](#) en la Guía de la aplicación AWS IoT SiteWise Monitor .
- `sc-status-grid`: un widget de estado que muestra el último valor de las propiedades de activos en forma de cuadrícula. Para obtener más información, consulte [Widgets de estado](#) en la Guía de la aplicación AWS IoT SiteWise Monitor .
- `sc-status-timeline`: un widget de estado que muestra los valores históricos de las propiedades de activos como línea temporal. Para obtener más información, consulte [Widgets de estado](#) en la Guía de la aplicación AWS IoT SiteWise Monitor .

- `sc-kpi`: una visualización de indicadores clave de rendimiento (KPI). Para obtener más información, consulte [Widgets de KP](#) en la Guía de la aplicación AWS IoT SiteWise Monitor .
- `sc-table`: un widget de tabla. Para obtener más información, consulte [Widgets de tabla](#) en la Guía de la aplicación AWS IoT SiteWise Monitor .

`title`

Título del widget.

`x`

Posición horizontal del widget, comenzando desde la izquierda de la cuadrícula. Este valor se refiere a la posición del widget en la cuadrícula del panel.

`y`

Posición vertical del widget, comenzando desde la parte superior de la cuadrícula. Este valor se refiere a la posición del widget en la cuadrícula del panel.

`width`

Anchura del widget, expresada en número de espacios en la cuadrícula del panel.

`height`

Altura del widget, expresada en número de espacios en la cuadrícula del panel.

`metrics`

Lista de estructuras métricas que definan cada una un flujo de datos para este widget. Cada estructura de la lista debe contener la siguiente información:

`label`

Etiqueta que se mostrará para esta métrica.

`type`

El tipo de fuente de datos de esta métrica. AWS IoT SiteWise proporciona los siguientes tipos de métricas:

- `iotsitewise`: el panel de control obtiene datos de una propiedad de activo en AWS IoT SiteWise. Si elige esta opción, deberá definir `assetId` y `propertyId` para esta métrica.

assetId

(Opcional) ID de un activo en AWS IoT SiteWise.

Este campo es obligatorio si elige `iotsitewise` para `type` en esta métrica.

propertyId

(Opcional) ID de una propiedad de activo en AWS IoT SiteWise.

Este campo es obligatorio si elige `iotsitewise` para `type` en esta métrica.

analysis

(Opcional) Una estructura que define el análisis, como líneas de tendencia, que se mostrará para el widget. Para obtener más información, consulte [Configuración de líneas de tendencia](#) en la Guía de la aplicación AWS IoT SiteWise Monitor . Puede añadir una línea de tendencia de cada tipo por propiedad en el widget. La estructura de análisis contiene la siguiente información:

trends

(Opcional) Lista de estructuras de tendencia que definan cada una un análisis de tendencia para este widget. Cada estructura de la lista contiene la siguiente información:

type

El tipo de línea de tendencia. Elija la opción siguiente:

- `linear-regression`— Muestra una línea de regresión lineal. SiteWise El monitor utiliza el método de [mínimos cuadrados](#) para calcular la regresión lineal.

annotations

(Opcional) Una estructura de anotaciones que define umbrales para el widget. Para obtener más información, consulte [Configuración de umbrales](#) en la Guía de la aplicación AWS IoT SiteWise Monitor . Puede añadir hasta seis anotaciones por widget. La estructura de anotación contiene la siguiente información:

y

(Opcional) Lista de estructuras de anotación que definan cada una un umbral horizontal para este widget. Cada estructura de la lista contiene la siguiente información:

`comparisonOperator`

El operador de comparación para el umbral. Seleccione una de las siguientes opciones:

- `LT`: resalta las propiedades que tienen al menos un punto de datos menor que `value`.
- `GT`: resalta las propiedades que tienen al menos un punto de datos mayor que `value`.
- `LTE`: resalta las propiedades que tienen al menos un punto de datos menor o igual que `value`.
- `GTE`: resalta las propiedades que tienen al menos un punto de datos mayor o igual que `value`.
- `EQ`: resalta las propiedades que tienen al menos un punto de datos igual que `value`.

`value`

El valor de umbral para comparar los puntos de datos con `comparisonOperator`.

`color`

(Opcional) El código hexadecimal de 6 dígitos del color del umbral. La visualización muestra leyendas de propiedades en este color para las propiedades con al menos un punto de datos que cumpla la regla de umbral. De forma predeterminada es negro (`#000000`).

`showValue`

(Opcional) Mostrar o no el valor del umbral en los márgenes del widget. El valor predeterminado es `true`.

`properties`

(Opcional) Un diccionario plano de propiedades para el widget. Los miembros de esta estructura dependen del contexto. AWS IoT SiteWise proporciona los siguientes widgets que utilizan: `properties`

- Los [Gráficos de líneas](#), [Gráficos de dispersión](#), y [Gráficos de barras](#) tienen la siguiente propiedad:

`colorDataAcrossThresholds`

(Opcional) Cambiar o no el color de los datos que cruzan los umbrales en este widget. Al habilitar esta opción, los datos que cruzan un umbral aparecen en el color que usted elija. El valor predeterminado es `true`.

- Las [cuadrículas de estado](#) tienen la siguiente propiedad:

labels

(Opcional) Una estructura que defina las etiquetas que se mostrarán en la cuadrícula de estado. La estructura de etiquetas contiene la siguiente información:

showValue

(Opcional) Mostrar o no la unidad y el valor de cada propiedad de activo en este widget. El valor predeterminado es `true`.

Example Ejemplo de definición de panel

En el ejemplo siguiente se define un panel a partir de una carga almacenada en un archivo JSON.

```
aws iotsitewise create-dashboard \
  --project-id a1b2c3d4-5678-90ab-cdef-eeeeEXAMPLE \
  --dashboard-name "Wind Farm Dashboard" \
  --dashboard-definition file://dashboard-definition.json
```

El siguiente ejemplo JSON para `dashboard-definition.json` define un panel con los siguientes widgets de visualización:

- Un gráfico de líneas que muestra la energía eólica total en la parte superior izquierda del panel. Este gráfico de líneas incluye un umbral que indica cuándo el parque eólico produce menos energía que su producción mínima prevista. Este gráfico de líneas también incluye una línea de tendencia de regresión lineal.
- Un gráfico de barras que muestra la velocidad del viento de cuatro turbinas en la parte superior derecha del panel.

Note

Este ejemplo representa visualizaciones de gráficos de líneas y barras en un panel de control. Este panel es similar al [ejemplo de panel de energía eólica](#).

```
{
  "widgets": [
    {
```

```
"type": "sc-line-chart",
"title": "Total Average Power",
"x": 0,
"y": 0,
"height": 3,
"width": 3,
"metrics": [
  {
    "label": "Power",
    "type": "iotsitewise",
    "assetId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
    "propertyId": "a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
    "analysis": {
      "trends": [
        {
          "type": "linear-regression"
        }
      ]
    }
  }
],
"annotations": {
  "y": [
    {
      "comparisonOperator": "LT",
      "value": 20000,
      "color": "#D13212",
      "showValue": true
    }
  ]
}
},
{
  "type": "sc-bar-chart",
  "title": "Wind Speed",
  "x": 3,
  "y": 3,
  "height": 3,
  "width": 3,
  "metrics": [
    {
      "label": "Turbine 1",
      "type": "iotsitewise",
      "assetId": "a1b2c3d4-5678-90ab-cdef-2a2a2EXAMPLE",
```

```
    "propertyId": "a1b2c3d4-5678-90ab-cdef-55555EXAMPLE"
  },
  {
    "label": "Turbine 2",
    "type": "iotsitewise",
    "assetId": "a1b2c3d4-5678-90ab-cdef-2b2b2EXAMPLE",
    "propertyId": "a1b2c3d4-5678-90ab-cdef-55555EXAMPLE"
  },
  {
    "label": "Turbine 3",
    "type": "iotsitewise",
    "assetId": "a1b2c3d4-5678-90ab-cdef-2c2c2EXAMPLE",
    "propertyId": "a1b2c3d4-5678-90ab-cdef-55555EXAMPLE"
  },
  {
    "label": "Turbine 4",
    "type": "iotsitewise",
    "assetId": "a1b2c3d4-5678-90ab-cdef-2d2d2EXAMPLE",
    "propertyId": "a1b2c3d4-5678-90ab-cdef-55555EXAMPLE"
  }
]
}
```

Active las alarmas de sus portales en AWS IoT SiteWise


Note

La función SiteWise Monitor ya no está disponible para los nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte [SiteWise Supervisar el cambio de disponibilidad](#).

Note

Aviso de fin del soporte: AWS finalizó el soporte para AWS IoT Events. Para obtener más información, consulte [AWS IoT Events el fin del soporte](#).

Puede habilitar la función de alarmas compatible con sus portales AWS IoT Events para que los administradores del portal puedan crear, editar y eliminar modelos de AWS IoT Events alarmas en sus portales de SiteWise Monitor. Los propietarios de proyectos pueden configurar alarmas. Los observadores de proyectos pueden ver los detalles de las alarmas. En esta sección se explica cómo puede utilizar la AWS IoT SiteWise consola para habilitar la función de alarmas en sus portales.

 Important

- No puede crear alarmas externas en sus portales.
- Si desea enviar notificaciones de alarma, debe elegir el Centro de identidades de IAM para el servicio de autenticación de usuarios.
- La función de notificaciones de alarma no está disponible en China (Pekín) Región de AWS.

Al configurar y crear un portal, puede habilitar las alarmas y las notificaciones de alarma en Paso 2 Características adicionales. En función del servicio de autenticación de usuarios, elija una de las siguientes opciones:

IAM Identity Center

AWS IoT SiteWise > Monitor > Portals > Create portal

Step 1
Portal configuration

Step 2- optional
Additional features

Step 3
Invite administrators

Step 4
Assign users

Additional features - *optional*

Alarms

Your portal users can create alarms in the portal to monitor equipment or processes. They can also get notified when the equipment or processes perform outside specified range.

Enable alarms
If enabled, your portal users can define AWS IoT Events alarms in SiteWise Monitor.

AWS IoT SiteWise access role
Choose an IAM role that allows AWS IoT Events to send data to AWS IoT SiteWise. To edit the role, go to the [IAM console](#).

Create a role from an AWS managed template

Use an existing role

Enable alarm notifications
If enabled, alarms can send email or SMS notifications.

Sender
Specify the email address that sends alarm notifications. To edit or add a sender, go to the [Amazon SES console](#).

AWS Lambda role
Choose an IAM role that allows AWS Lambda to send data to Amazon SES and Amazon SNS. To edit the role, go to the [IAM console](#).

Create a role from an AWS managed template

Use an existing role

AWS Lambda function
Choose an AWS Lambda function to manage alarm notifications. To edit the function, go to the [AWS Lambda console](#).

Create a lambda from an AWS managed template

Use an existing lambda

Previous **Create**

Para habilitar alarmas para un portal

1. (Opcional) Seleccione Habilitar alarmas.
 - En Rol de acceso de AWS IoT SiteWise , utilice un rol existente o cree uno con los permisos necesarios. Este rol requiere el permiso `iotevents:BatchPutMessage` y una relación de confianza que permita a `iot.amazonaws.com` y `iotevents.amazonaws.com` asumir el rol.
2. (Opcional) Seleccione Habilitar notificaciones de alarma.
 - a. En Remitente, elija el remitente.

⚠ Important

Debe verificar la dirección de correo electrónico del remitente en Amazon SES. Para obtener más información, consulte [Verificación de direcciones de correo electrónico en Amazon SES](#), en la Guía para desarrolladores de Amazon Simple Email Service.

- b. En Rol de AWS Lambda , utilice un rol existente o cree uno con los permisos necesarios. Este rol requiere los permisos `lambda:InvokeFunction` y `sso-directory:DescribeUser` y una relación de confianza que permita a `iotevents.amazonaws.com` y `lambda.amazonaws.com` asumir el rol.
- c. En Funciones de AWS Lambda , elija una función de Lambda existente o cree una que administre las notificaciones de alarma. Para obtener más información, consulte [Administración de notificaciones de alarma](#) en la Guía para desarrolladores de AWS IoT Events .

IAM

AWS IoT SiteWise > Monitor > Portals > Create portal

Step 1
Portal configuration

Step 2- optional
Additional features

Step 3
Invite administrators

Step 4
Assign users

Additional features - optional

Alarms

Your portal users can create alarms in the portal to monitor equipment or processes. They can also get notified when the equipment or processes perform outside specified range.

Enable alarms
If enabled, your portal users can define AWS IoT Events alarms in SiteWise Monitor.

AWS IoT SiteWise access role
Choose an IAM role that allows AWS IoT Events to send data to AWS IoT SiteWise. To edit the role, go to the [IAM console](#).

Create a role from an AWS managed template
 Use an existing role

ⓘ Alarms created in the portal can't send notifications. If you want to send alarm notifications, choose **Previous**. Then, on the **Portal configuration** page, choose **AWS SSO for User authentication**.

Previous **Create**

Para habilitar alarmas para un portal

- (Opcional) Seleccione Habilitar alarmas.

- En Rol de acceso de AWS IoT SiteWise , utilice un rol existente o cree uno con los permisos necesarios. Este rol requiere el permiso `iotevents:BatchPutMessage` y una relación de confianza que permita a `iot.amazonaws.com` y `iotevents.amazonaws.com` asumir el rol.

Para obtener más información sobre las alarmas en SiteWise Monitor, consulte [Supervisión con alarmas](#) en la Guía de AWS IoT SiteWise aplicaciones.

Habilitar su AWS IoT SiteWise portal en la periferia

Note

La función SiteWise Monitor ya no está disponible para los nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte [SiteWise Supervisar el cambio de disponibilidad](#).

Después de activar su portal en Edge, este portal estará disponible en todas las puertas de enlace SiteWise Edge con el paquete de procesamiento de datos activado en su cuenta.

Para habilitar el portal en la periferia

1. En la sección Configuración de la periferia, active Habilitar este portal en la periferia.
2. Seleccione Crear.

Administre sus portales de SiteWise Monitor

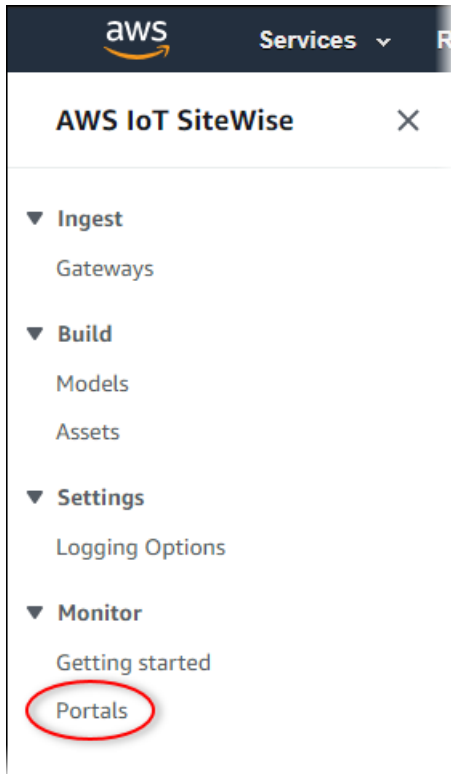
Note

La función SiteWise Supervisar ya no está disponible para los nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte [SiteWise Supervisar el cambio de disponibilidad](#).

Usted tiene capacidad para administrar y configurar varios aspectos del portal. Esto incluye agregar y eliminar usuarios o administradores, configurar permisos y roles de los usuarios, personalizar la

URL y el nombre del portal, configurar la información de contacto de soporte y enviar invitaciones por correo electrónico a los administradores del portal.

1. Inicie sesión en la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, elija Monitor (Monitorizar), Portals (Portales).



3. Elija un portal y, a continuación, elija View details (Ver detalles) (o elija el Name (Nombre) del portal).
4. Puede realizar cualquiera de las siguientes tareas administrativas:
 - [Cambie los detalles del portal en AWS IoT SiteWise](#)
 - [Agregue o elimine administradores del portal en AWS IoT SiteWise](#)
 - [Envíe invitaciones por correo electrónico a administradores del portal](#)
 - [Agregue o elimine usuarios del portal en AWS IoT SiteWise](#)
 - [Eliminar un portal en AWS IoT SiteWise](#)

Para obtener información sobre cómo crear un volumen, consulte [Comience con AWS IoT SiteWise Monitor \(Classic\)](#).

Temas

- [Cambie los detalles del portal en AWS IoT SiteWise](#)
- [Agregue o elimine administradores del portal en AWS IoT SiteWise](#)
- [Envíe invitaciones por correo electrónico a administradores del portal](#)
- [Agregue o elimine usuarios del portal en AWS IoT SiteWise](#)
- [Eliminar un portal en AWS IoT SiteWise](#)

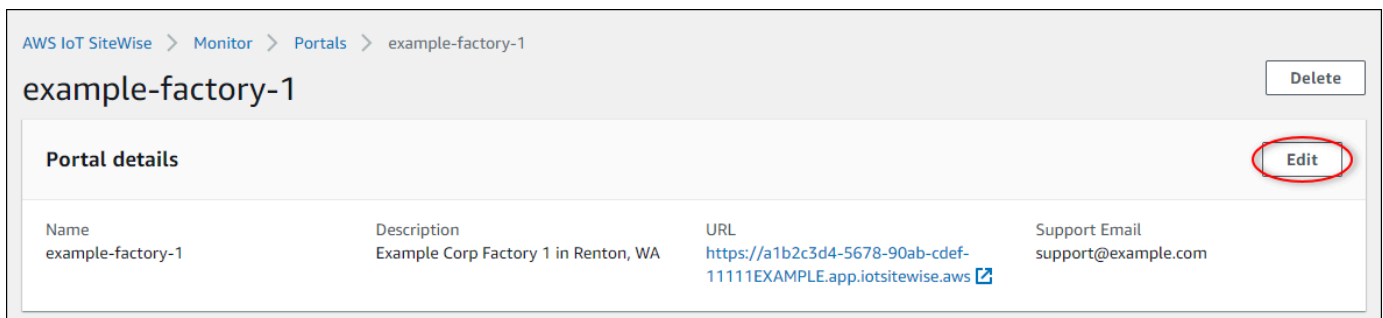
Cambie los detalles del portal en AWS IoT SiteWise

Note

La función SiteWise Supervisar ya no está disponible para los nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte [SiteWise Supervisar el cambio de disponibilidad](#).

Puede cambiar el nombre, la descripción, la marca, el email de soporte y los permisos de un portal.

1. En la página de detalles del portal, en la sección Portal details (Detalles del portal) elija Edit (Editar).



AWS IoT SiteWise > Monitor > Portals > example-factory-1

example-factory-1 Delete

Portal details Edit

Name	Description	URL	Support Email
example-factory-1	Example Corp Factory 1 in Renton, WA	https://a1b2c3d4-5678-90ab-cdef-11111EXAMPLE.app.iotsitewise.aws	support@example.com

2. Actualice el nombre, la descripción, la marca del portal, el email de contacto de soporte o los permisos.
3. Cuando haya terminado, elija Save.

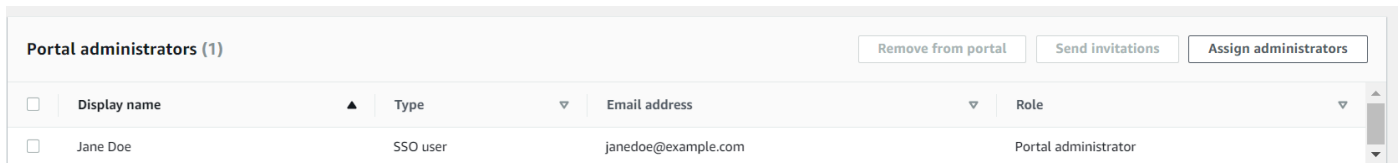
Agregue o elimine administradores del portal en AWS IoT SiteWise

Note

La función SiteWise Supervisar ya no está disponible para los nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte [SiteWise Supervisar el cambio de disponibilidad](#).

En unos pocos pasos, puede añadir o quitar usuarios como administradores de un portal. En función del servicio de autenticación de usuarios, elija una de las siguientes opciones.

IAM Identity Center



<input type="checkbox"/>	Display name	Type	Email address	Role
<input type="checkbox"/>	Jane Doe	SSO user	janedoe@example.com	Portal administrator

Para añadir administradores de portal

1. En la página de detalles del portal, en la sección Administradores del portal, seleccione Asignar administradores.
2. En la página Asignar administradores, seleccione las casillas de verificación de los usuarios que quiera añadir al portal como administradores.

Note

Si utiliza el Centro de identidades de IAM como almacén de identidades y ha iniciado sesión en su cuenta de administración de AWS Organizations, puede elegir Crear usuario para crear un usuario del Centro de identidades de IAM. El Centro de identidades de IAM enviará al nuevo usuario un correo electrónico para que establezca su contraseña. A continuación, puede asignar el usuario al portal como administrador. Para obtener más información, consulte [Administrar identidades en el IAM Identity Center](#).

3. Seleccione Asignar administradores.

AWS IoT SiteWise > Monitor > Portals > example-factory-1 > Assign administrators

Assign administrators

Choose the users that you want to be portal administrators. Portal administrators can grant users access to specific industrial equipment data. [Learn more](#)

Users (2) Create user

Find resources

<input type="checkbox"/>	Display name	Email
<input type="checkbox"/>	Jane Doe	janedoe@example.com
<input checked="" type="checkbox"/>	John Doe	johndoe@example.com

Selected users (1)

Cancel Assign administrators

Para eliminar administradores del portal

- En la página de detalles del portal, en la sección Administradores del portal active la casilla de verificación de cada usuario que quiera eliminar y, a continuación, seleccione Eliminar del portal.

Note

Le recomendamos que seleccione al menos un administrador del portal.

IAM

Portal administrators (1)

Remove from portal Send invitations Assign administrators

<input checked="" type="checkbox"/>	Display name	Type	Email address	Role
<input checked="" type="checkbox"/>	██████████	IAM user	-	Portal administrator

Para añadir administradores de portal

- En la página de detalles del portal, en la sección Administradores del portal, seleccione Asignar administradores.
- En la página Asignación de administradores, haga lo siguiente:
 - Elija Usuarios de IAM si lo que desea es añadir un usuario de IAM como administrador de su portal.
 - Elija Roles de IAM si lo que desea es añadir un rol de IAM como administrador de su portal.

3. Marque las casillas de verificación de los usuarios o roles que desee como administradores de su portal. Esto añade los usuarios o roles a la lista de Administradores del portal.
4. Seleccione Asignar administradores.


Important

Los usuarios o roles deben tener el permiso de `iotsitewise:DescribePortal` para iniciar sesión en el portal.

AWS IoT SiteWise > Monitor > Portals > example-factory-2 > Assign administrators

Assign administrators

Choose the users that you want to be portal administrators. Portal administrators can grant users access to specific industrial equipment data. [Learn more](#)

 IAM users or roles must have the `iotsitewise:DescribePortal` permission to sign in to the portal.

Users | Roles

IAM users (1) [Manage users in IAM console](#)

<input checked="" type="checkbox"/>	Name	Date created
<input checked="" type="checkbox"/>	raspberrypi-testing	11-08-2019


► **Portal administrators (1)** [Remove](#)

Cancel **Assign administrators**

AWS IoT SiteWise > Monitor > Portals > example-factory-2 > Assign administrators

Assign administrators

Choose the users that you want to be portal administrators. Portal administrators can grant users access to specific industrial equipment data. [Learn more](#)

 IAM users or roles must have the `iotsitewise:DescribePortal` permission to sign in to the portal.

Users | **Roles**

IAM roles (66) [Manage roles in IAM console](#)

<input type="checkbox"/>	Name	Date created
<input type="checkbox"/>	[REDACTED]	
<input checked="" type="checkbox"/>	AWSIoTSiteWiseMonitorServiceRole_4wZigNpA1	03-16-2021
<input type="checkbox"/>	AWSIoTSiteWiseMonitorServiceRole_EcKT-2Oar	03-11-2021
<input type="checkbox"/>	AWSIoTSiteWiseMonitorServiceRole_GTnd004Wr	03-16-2021
<input type="checkbox"/>	AWSIoTSiteWiseMonitorServiceRole_rHINLNC5-	03-11-2021
<input type="checkbox"/>	AWSIoTSiteWiseMonitorServiceRole_XB330QUIO	03-10-2021
<input type="checkbox"/>	[REDACTED]	
<input type="checkbox"/>	[REDACTED]	
<input type="checkbox"/>	[REDACTED]	

► **Portal administrators (2)** [Remove](#)

Cancel **Assign administrators**

Para eliminar administradores del portal

- En la página de detalles del portal, en la sección Administradores del portal active la casilla de verificación de cada usuario que quiera eliminar y, a continuación, seleccione Eliminar del portal.

Note

No se recomienda dejar un portal sin un administrador del portal.

Envíe invitaciones por correo electrónico a administradores del portal

Note

La función SiteWise Supervisar ya no está disponible para los nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte [SiteWise Supervisar el cambio de disponibilidad](#).

Puede enviar invitaciones por correo electrónico a administradores del portal.

- En la página de detalles del portal, en la sección Portal administrators (Administradores del portal), seleccione las casillas de verificación de los administradores del portal.

Portal administrators (1)				Remove from portal	Send invitations	Assign users
<input checked="" type="checkbox"/>	Display name	Email address	Role			
<input checked="" type="checkbox"/>	John Doe	john.doe@example.com	Portal administrator			

- Seleccione Send invitations (Enviar invitaciones). Se abrirá el cliente de correo electrónico y se rellenará una invitación en el cuerpo del mensaje.

Puede personalizar el correo electrónico antes de enviarlo a los administradores del portal.

Agregue o elimine usuarios del portal en AWS IoT SiteWise

Note

La función SiteWise Supervisar ya no está disponible para los nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte [SiteWise Supervisar el cambio de disponibilidad](#).

Elija los usuarios que tienen acceso a sus portales. Los usuarios del portal aparecen en la lista de usuarios de un portal de SiteWise Monitor. En esta lista, los administradores del portal pueden añadir propietarios de proyectos y los propietarios de proyectos pueden añadir observadores de proyectos.

Note

Los administradores del portal y los usuarios del portal pueden ponerse en contacto con usted a través del correo electrónico de soporte de un portal si tienen que agregar o quitar un usuario.

En función del servicio de autenticación de usuarios, elija una de las siguientes opciones.

IAM Identity Center

Portal users (1)				Remove from portal	Assign users
<input type="checkbox"/>	Display name	Type	Email address	Role	
<input type="checkbox"/>	John Doe	SSO user	johndoe@example.com	Portal viewer	

Para añadir usuarios del portal

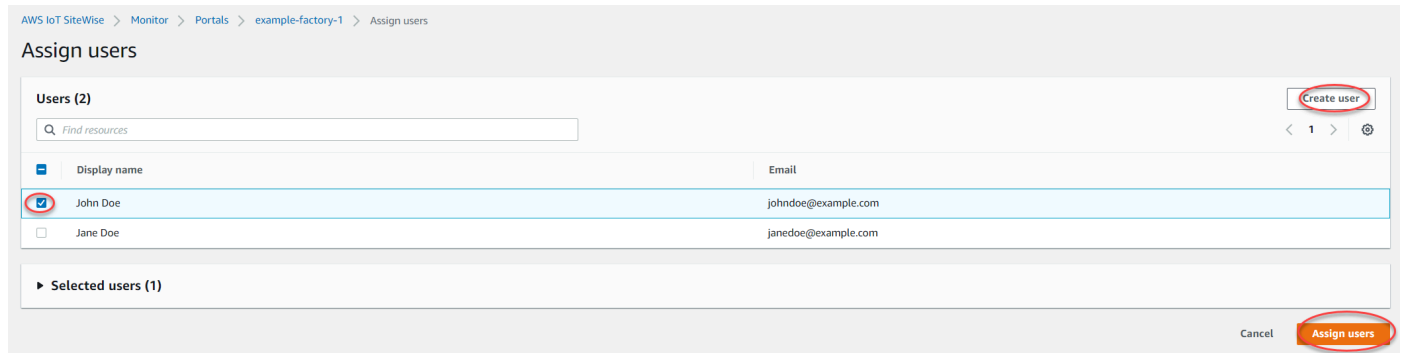
1. En la página de detalles del portal, en la sección Portal users (Usuarios del portal), elija Assign users (Asignar usuarios).
2. En la página Asignar usuarios, seleccione la casilla de verificación de los usuarios que va a añadir al portal.

Note

Si utiliza el Centro de identidades de IAM como almacén de identidades y ha iniciado sesión en su cuenta de administración de AWS Organizations , puede elegir

Crear usuario para crear un usuario del Centro de identidades de IAM. El Centro de identidades de IAM enviará al nuevo usuario un correo electrónico para que establezca su contraseña. A continuación, puede asignar el usuario al portal como usuario. Para obtener más información, consulte [Administrar identidades en el IAM Identity Center](#).

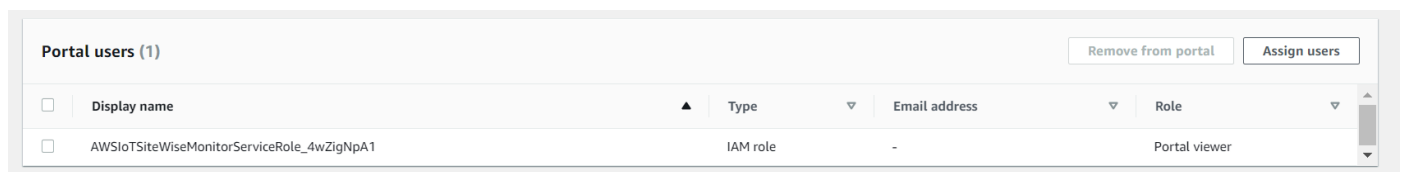
3. Elija Assign users (Asignar usuarios).



Para quitar usuarios del portal

- En la página de detalles del portal, en la sección Usuarios del portal, seleccione las casillas de verificación de los usuarios que va a eliminar del portal y, a continuación, Eliminar del portal.

IAM



Para añadir usuarios del portal

1. En la página de detalles del portal, en la sección Portal users (Usuarios del portal), elija Assign users (Asignar usuarios).
2. En la página Asignación de usuarios, haga lo siguiente:
 - Elija Usuarios de IAM para añadir un usuario de IAM como usuario de su portal.
 - Elija Roles de IAM para añadir un rol de IAM como usuario de su portal.

3. Marque las casillas de verificación de los usuarios o roles que desee como usuarios de su portal. Esto añade los usuarios o roles a la lista Usuarios del portal.
4. Elija Assign users (Asignar usuarios).

AWS IoT SiteWise > Monitor > Portals > example-factory-2 > Assign users

Assign users

Users Roles

IAM users (1) [Manage users in IAM console](#)

Find user name

< 1 >

<input checked="" type="checkbox"/>	Name	Date created
<input checked="" type="checkbox"/>	[REDACTED]	11-08-2019

Portal users (1) [Remove](#)

Cancel [Assign users](#)

AWS IoT SiteWise > Monitor > Portals > example-factory-2 > Assign users

Assign users

Users Roles

IAM roles (66) [Manage roles in IAM console](#)

Find role name

< 1 2 3 4 5 6 7 >

<input type="checkbox"/>	Name	Date created
<input type="checkbox"/>	[REDACTED]	
<input checked="" type="checkbox"/>	AWSIoTSiteWiseMonitorServiceRole_4wZigNpA1	03-16-2021
<input type="checkbox"/>	AWSIoTSiteWiseMonitorServiceRole_ECKT-2Oar	03-11-2021
<input type="checkbox"/>	AWSIoTSiteWiseMonitorServiceRole_GTnd0O4Wr	03-16-2021
<input type="checkbox"/>	AWSIoTSiteWiseMonitorServiceRole_rHINLNCS-	03-11-2021
<input type="checkbox"/>	AWSIoTSiteWiseMonitorServiceRole_XB330QUIO	03-10-2021
<input type="checkbox"/>	[REDACTED]	
<input type="checkbox"/>	[REDACTED]	
<input type="checkbox"/>	[REDACTED]	

Portal users (2) [Remove](#)

Cancel [Assign users](#)

Para quitar usuarios del portal

- En la página de detalles del portal, en la sección Usuarios del portal, seleccione las casillas de verificación de los usuarios que va a eliminar del portal y, a continuación, Eliminar del portal.

⚠ Important

Los usuarios o roles deben tener el permiso de `iotsitewise:DescribePortal` para iniciar sesión en el portal.

Eliminar un portal en AWS IoT SiteWise

ℹ Note

La función SiteWise Supervisar ya no está disponible para los nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte [SiteWise Supervisar el cambio de disponibilidad](#).

Puede eliminar un portal si lo ha creado con fines de prueba o si ha creado un duplicado de un portal que ya existe.

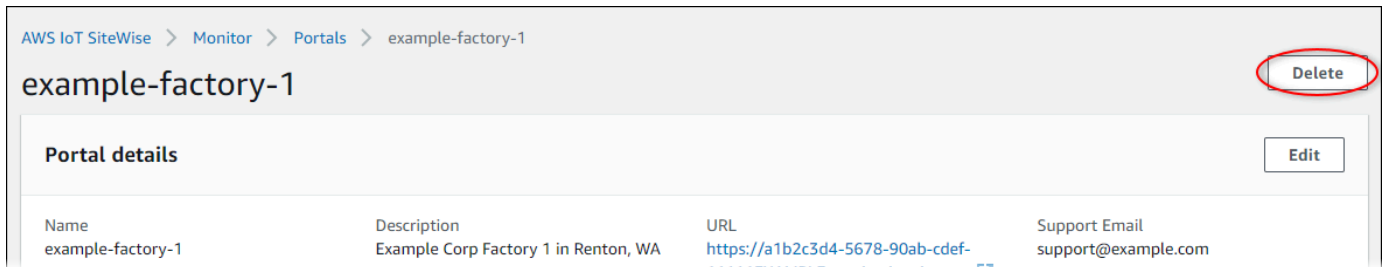
ℹ Note

Primero debe eliminar manualmente todos los paneles y proyectos del portal antes de poder eliminarlo. Para obtener más información, consulte [Eliminar proyectos](#) y [Eliminar paneles](#) en la Guía de aplicaciones de SiteWise monitoreo.

1. En la página de detalles del portal, elija Eliminar.

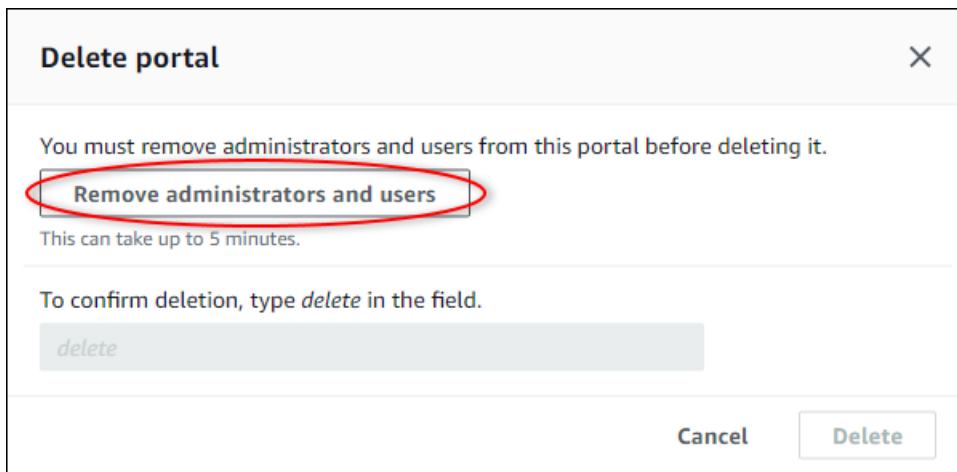
⚠ Important

Cuando se elimina un portal, se pierden todos los proyectos que contiene el portal y todos los paneles de cada proyecto. Esta acción no se puede deshacer. Los datos de activos no se ven afectados.

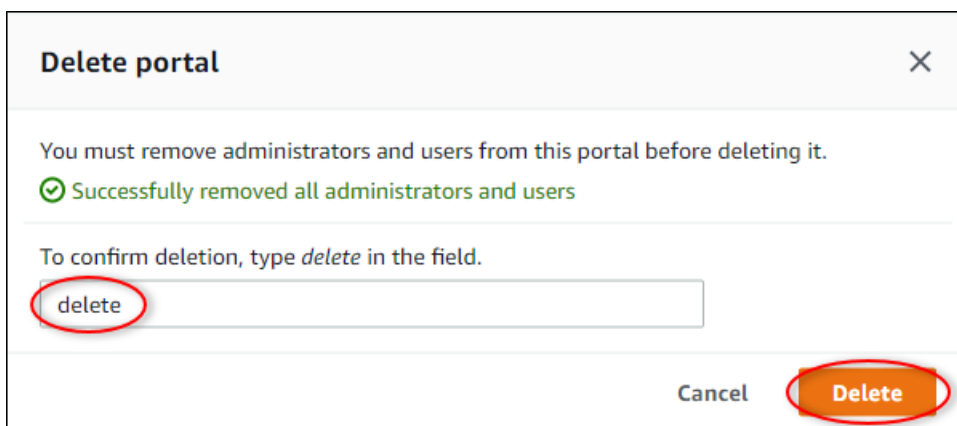


2. En el cuadro de diálogo Eliminar portales, seleccione Eliminar administradores y usuarios.

Debe quitar los administradores y usuarios del portal antes de poder eliminarlo. Si el portal no tiene administradores o usuarios, el botón no aparece y puede pasar al siguiente paso.



3. Si está seguro de que desea eliminar todo el portal, escriba **delete** en el campo para confirmar la eliminación.



4. Elija Eliminar.

Comience con AWS IoT SiteWise Monitor (compatible con la IA): vista previa

Note

La función SiteWise Supervisar ya no está disponible para los nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte [SiteWise Supervisar el cambio de disponibilidad](#).

Como AWS administrador de su organización, puede crear portales desde la AWS IoT SiteWise consola para que los miembros de la organización puedan ver sus AWS IoT SiteWise datos. Complete los siguientes pasos para empezar.

1. Configure y cree un portal.
2. Agregue administradores del portal y envíe correos electrónicos de invitación.
3. Agregue usuarios del portal.

Tras crear un portal, el administrador del portal puede crear proyectos y añadir un usuario al proyecto. A continuación, los miembros del proyecto crean paneles de control para visualizar los datos conectados AWS IoT SiteWise, lo que les permite supervisar el rendimiento de sus dispositivos, procesos y equipos conectados.

Note

Al añadir usuarios o administradores al portal, evite crear políticas AWS Identity and Access Management (de IAM) que restrinjan los permisos de los usuarios, como la IP limitada. Las políticas adjuntas con permisos restringidos no podrán conectarse al AWS IoT SiteWise portal.

Crea proyectos para compartirlos con tus equipos. Luego, los propietarios de los proyectos pueden crear paneles para visualizar las propiedades de los activos, lo que ayuda a los espectadores del proyecto a comprender el rendimiento de los dispositivos, los procesos y los equipos. También proporciona a sus equipos una visión coherente de las operaciones.

Los paneles ayudan a visualizar y comprender los datos del proyecto. Ayuda a las empresas y a los usuarios de aplicaciones a realizar un seguimiento de sus AWS IoT dispositivos y datos. Elija el tipo de visualización que mejor muestre sus datos según sus necesidades. Reorganice y modifique el tamaño de las visualizaciones para crear un diseño que se adapte a su equipo. Explore los activos y los datos de sus dispositivos, procesos y equipos, e identifique rápidamente los problemas y mejore la eficiencia operativa.

Temas

- [Creación de un portal](#)
- [Configure su portal](#)
- [Administración de los portales](#)
- [Eliminación de un portal](#)
- [Cree paneles con AWS CLI](#)
- [Iniciar sesión en el portal](#)
- [Crear un proyecto](#)
- [Actualiza un proyecto.](#)
- [Eliminación de un proyecto](#)
- [Creación de un panel](#)
- [Actualización de un panel](#)
- [Eliminación de un panel](#)
- [Configurar panel](#)

Creación de un portal

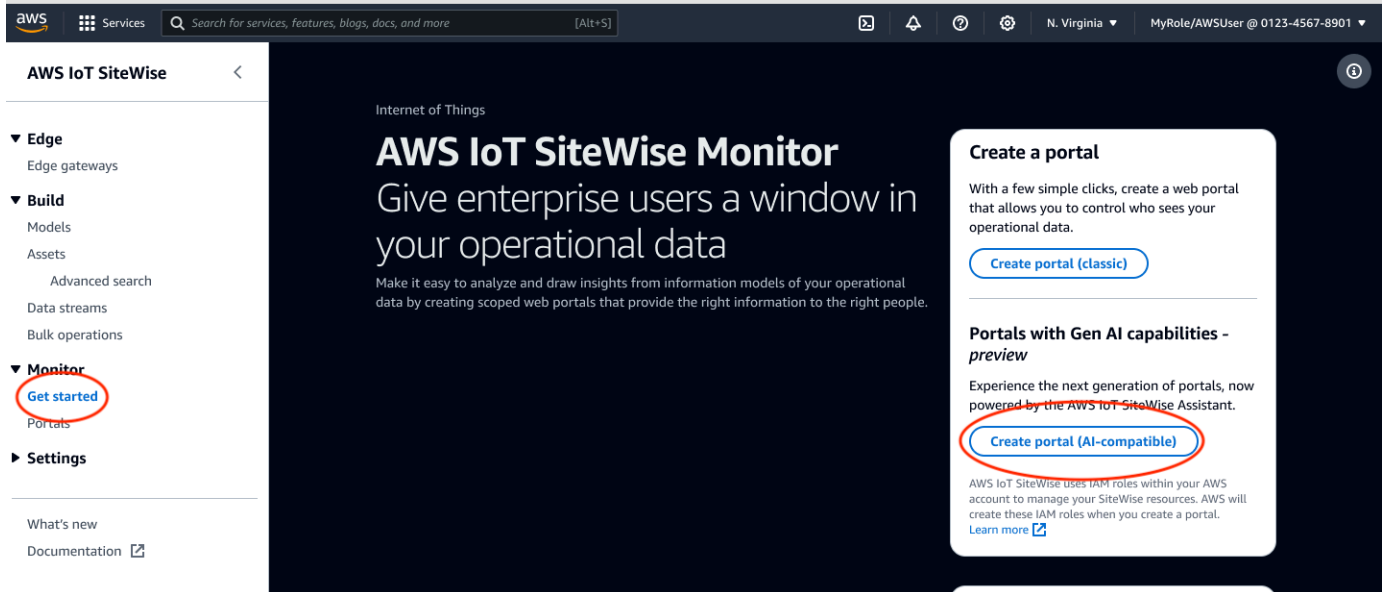
Note

La función SiteWise Supervisar ya no está disponible para los nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte [SiteWise Supervisar el cambio de disponibilidad](#).

Puede crear un portal de SiteWise monitoreo en la AWS IoT SiteWise consola.

Para crear un portal

1. Inicie sesión en la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, elija Supervisar y empezar.
3. Seleccione Crear portal (compatible con IA).



A continuación, debe proporcionar información básica para configurar el portal.

Configure su portal

Note

La función SiteWise Supervisar ya no está disponible para los nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte [SiteWise Supervisar el cambio de disponibilidad](#).

Sus usuarios utilizan los portales para ver sus datos. Puede personalizar el nombre, la descripción, la marca, la autenticación de usuario, el correo electrónico de contacto de asistencia y los permisos de un portal.


Pasos para configurar un portal:

1. Escriba un nombre para el portal.

2. (Opcional) Escriba una descripción para el portal. Si tiene varios portales, utilice descripciones significativas para ayudarlo a realizar un seguimiento de lo que contiene cada portal.
3. (Opcional) Suba una imagen para mostrar su marca en el portal. Elija una imagen PNG cuadrada. Si carga una imagen no cuadrada, el portal escala la imagen a un cuadrado.
4. Introduzca una dirección de correo electrónico en la casilla de correo electrónico de contacto de Support para problemas de soporte.
5. En el cuadro Autenticación de usuario, selecciona la siguiente opción:
 - Elija Centro de identidades de IAM si los usuarios de su portal inician sesión en este portal con sus nombres de usuario y contraseñas corporativos.


Si no ha habilitado el Centro de identidades de IAM en su cuenta, haga lo siguiente:

- a. Seleccione la opción Crear usuario.
- b. En la página Crear usuario, para crear el primer portal, introduzca la dirección de correo electrónico, nombre y apellido del usuario y, a continuación, seleccione Crear usuario.

 Note

El soporte para las credenciales de IAM estará disponible próximamente.

6. Elija una de las siguientes opciones en la sección de acceso al servicio:
 - Elija Crear y utilizar un nuevo rol de servicio. De forma predeterminada, SiteWise Monitor crea automáticamente un rol de servicio para cada portal. Esta función permite a los usuarios del portal acceder a sus AWS IoT SiteWise recursos. Para obtener más información, consulte [Utilice los roles de servicio para AWS IoT SiteWise Monitor](#).
 - Elija Utilizar un rol de servicio existente y, a continuación, el rol deseado.
7. Seleccione habilitar el AWS IoT SiteWise Asistente para este portal. El AWS IoT SiteWise Asistente proporciona un análisis de datos rápido, información en tiempo real y recomendaciones guiadas.

 Note

Habilitar el AWS IoT SiteWise Asistente generará cargos. Para utilizar las soluciones de conocimiento y la orientación a nivel empresarial, debe tener un conjunto de datos asociado al índice Amazon Kendra.

8. (Opcional) Agregue etiquetas para su portal. Para obtener más información, consulte [Etiquete sus AWS IoT SiteWise recursos](#).
9. Elija Crear portal. AWS IoT SiteWise creará su portal.

Note


Si cierra la consola, puede finalizar el proceso de instalación agregando administradores y usuarios. Para obtener más información, consulte [Adición o eliminación de administradores del portal](#). Si no desea conservar este portal, elimínelo para que no consuma recursos. Para obtener más información, consulte [Eliminación de un portal](#).

Aparece un mensaje cuando se crea el portal.


☑ Successfully created portal "example portal".
✕

AWS IoT SiteWise > Monitor > Portals > example portal

example portal

Edit
Delete
Open portal 

Portal details

Name example portal	Status Active	URL  https://p-jfnlf2d8.gamma.iotsitewise.aws
Type AI-compatible	AWS IoT SiteWise Assistant Disabled	Portal branding -
Description -	Last updated November 1, 2024, 15:37 (UTC-07:00)	Support contact email myemail@mycompany.com
ID f5fc93a1-011c-4c5a-81a6-e001b50d2547	Date created November 1, 2024, 15:37 (UTC-07:00)	

Una vez creado un portal, aparece en la sección Portales. La sección de detalles del portal muestra el nombre, la descripción, el ID, la URL, el estado, las fechas de la última actualización y creación, la marca del portal y el correo electrónico de soporte de cada portal.

La columna Estado puede adoptar uno de los siguientes valores.

- CREATING - AWS IoT SiteWise está procesando su solicitud para crear el portal. Este proceso puede tardar varios minutos en completarse.
- ACTUALIZAR - AWS IoT SiteWise está procesando su solicitud de actualización del portal. Este proceso puede tardar varios minutos en completarse.

- **PENDIENTE** - AWS IoT SiteWise está esperando a que finalice la propagación del registro DNS. Este proceso puede tardar varios minutos en completarse. Durante el estado PENDIENTE puede eliminar el portal.
- **ELIMINAR** - AWS IoT SiteWise está procesando su solicitud de eliminación del portal. Este proceso puede tardar varios minutos en completarse.
- **ACTIVO**: cuando el portal se activa, los usuarios de su portal tienen acceso al mismo.
- **ERROR**: no se ha podido procesar su solicitud de creación, actualización o eliminación del portal. Si has habilitado AWS IoT SiteWise el envío de registros a Amazon CloudWatch Logs, puedes usar estos registros para solucionar problemas. Para obtener más información, consulte [Supervisión AWS IoT SiteWise con CloudWatch registros](#).

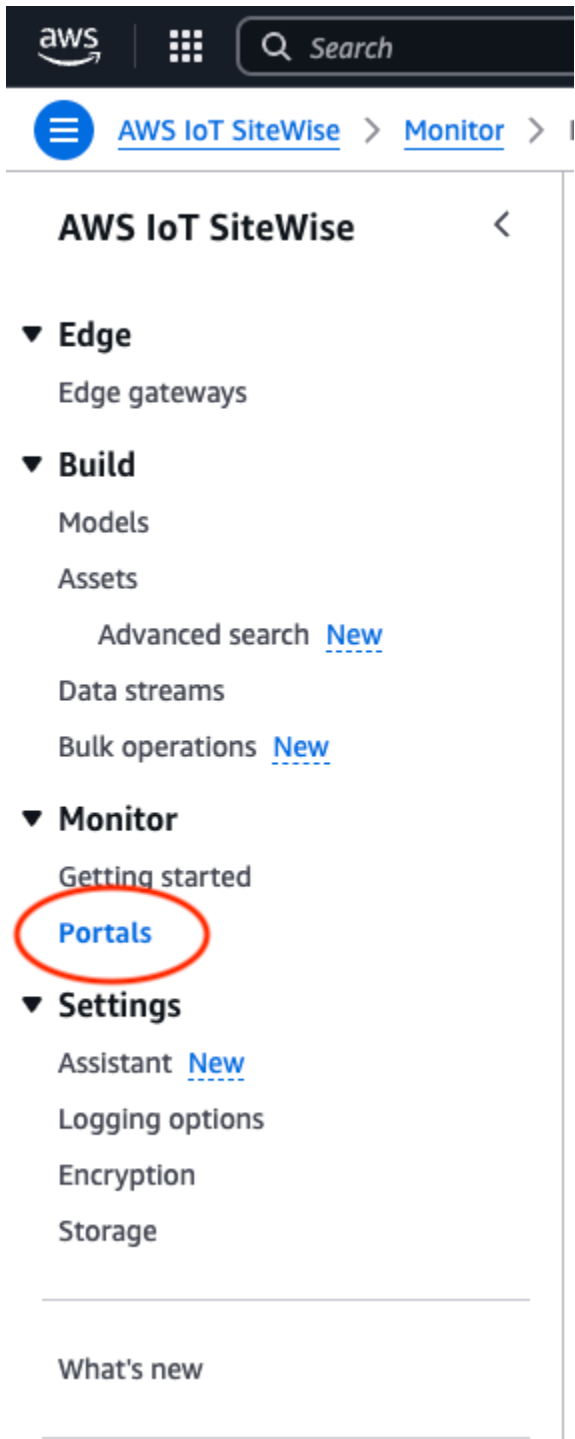
Administración de los portales

Note

La función SiteWise Supervisar ya no está disponible para los nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte [SiteWise Supervisar el cambio de disponibilidad](#).

Usted tiene capacidad para administrar y configurar varios aspectos del portal. Esto incluye añadir y eliminar administradores, establecer permisos y funciones, personalizar el nombre y la descripción, configurar el correo electrónico de asistencia e invitar a los administradores del portal.

1. Inicie sesión en la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, elija Monitor (Monitorizar), Portals (Portales).



3. Elija un portal y, a continuación, elija Abrir portal (o elija el nombre del portal).
4. Puede realizar cualquiera de las siguientes tareas administrativas:
 - [Edite los atributos del portal](#)
 - [Adición o eliminación de administradores del portal](#)

- [Envíe invitaciones por correo electrónico a administradores del portal](#)
- [Eliminar un portal en AWS IoT SiteWise](#)

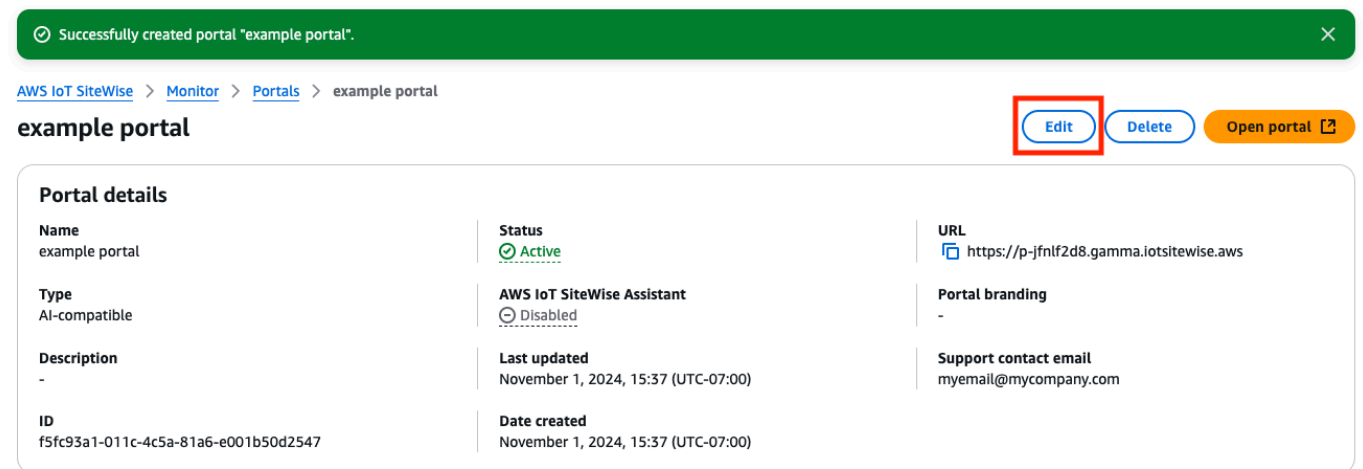
Edite los atributos del portal

Note

La función SiteWise Supervisar ya no está disponible para los nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte [SiteWise Supervisar el cambio de disponibilidad](#).

Puede cambiar el nombre, la descripción, la marca, el correo electrónico de asistencia y el acceso al servicio de un portal.

1. En la página de detalles del portal, en la sección Portal details (Detalles del portal) elija Edit (Editar).



Successfully created portal "example portal".

[AWS IoT SiteWise](#) > [Monitor](#) > [Portals](#) > example portal

example portal Edit Delete Open portal

Portal details		
Name example portal	Status Active	URL https://p-jfnlf2d8.gamma.iotsitewise.aws
Type AI-compatible	AWS IoT SiteWise Assistant Disabled	Portal branding -
Description -	Last updated November 1, 2024, 15:37 (UTC-07:00)	Support contact email myemail@mycompany.com
ID f5fc93a1-011c-4c5a-81a6-e001b50d2547	Date created November 1, 2024, 15:37 (UTC-07:00)	

2. Actualice el nombre, la descripción, la marca del portal, el correo electrónico de contacto de Support, el AWS IoT SiteWise asistente o el acceso al servicio.
3. Cuando haya finalizado, elija Save changes (Guardar cambios).

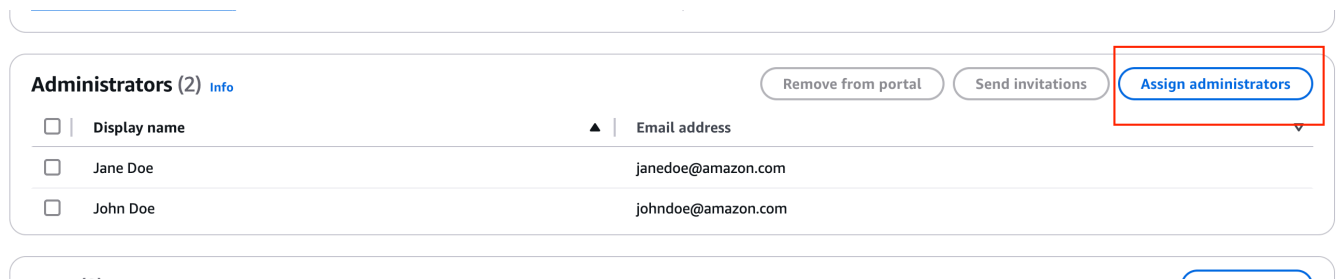
Adición o eliminación de administradores del portal

Note

La función SiteWise Supervisar ya no está disponible para los nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte [SiteWise Supervisar el cambio de disponibilidad](#).

En unos pocos pasos, puede añadir o quitar usuarios como administradores de un portal. En función del servicio de autenticación de usuarios, elija una de las siguientes opciones.

IAM Identity Center



Para añadir administradores de portal

1. En la página de detalles del portal, en la sección Administradores, elija Asignar administradores.
2. En la página Asignar administradores, seleccione los usuarios que desee añadir al portal como administradores.

Note

Si utiliza el Centro de identidades de IAM como almacén de identidades y ha iniciado sesión en su cuenta de administración de AWS Organizations, puede elegir Crear usuario para crear un usuario del Centro de identidades de IAM. El Centro de identidades de IAM enviará al nuevo usuario un correo electrónico para que establezca su contraseña. A continuación, puede asignar el usuario al portal como administrador. Para obtener más información, consulte [Administrar identidades en el IAM Identity Center](#).

3. Seleccione Asignar administradores.

Assign administrators [Info](#)

Choose portal administrators from the users list. Portal administrators grant access to specific industrial equipment data. [Learn more](#)

Users (11) Create user

Search: 2 matches < 1 > ⚙

<input type="checkbox"/>	Name	Email
<input checked="" type="checkbox"/>	johndoe@amazon.com	johndoe@amazon.com
<input type="checkbox"/>	janedoe@amazon.com	janedoe@amazon.com

[Cancel](#) [Assign administrators](#)

Para eliminar administradores del portal

- En la página de detalles del portal, en la sección Administradores del portal active la casilla de verificación de cada usuario que quiera eliminar y, a continuación, seleccione Eliminar del portal.

Note

Los administradores (#) muestran el número de administradores del portal. Puede añadir varios administradores del portal para gestionar los proyectos y trabajar en ellos.

Envíe invitaciones por correo electrónico a administradores del portal

Note

La función SiteWise Supervisar ya no está disponible para los nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte [SiteWise Supervisar el cambio de disponibilidad](#).

Puede enviar invitaciones por correo electrónico a administradores del portal.

- En la página de detalles del portal, en la sección Administradores, active las casillas de verificación de los administradores del portal.
- Seleccione Send invitations (Enviar invitaciones). Se abrirá el cliente de correo electrónico y se rellenará una invitación en el cuerpo del mensaje.

Puede personalizar el correo electrónico antes de enviarlo a los administradores del portal.

Eliminación de un portal

Puede eliminar un portal si lo ha creado con fines de prueba o si ha creado un duplicado de un portal que ya existe.

Note

Primero debe eliminar manualmente todos los paneles y proyectos del portal antes de poder eliminarlo.

1. En la página de detalles del portal, elija Eliminar.

Important

Cuando se elimina un portal, se pierden todos los proyectos que contiene el portal y todos los paneles de cada proyecto. Esta acción no se puede deshacer. Los datos de activos no se ven afectados.

The screenshot shows the AWS IoT SiteWise Mezzanine interface. The breadcrumb navigation is 'AWS IoT SiteWise > Monitor > Portals'. The page title is 'Portals (3)'. There are buttons for 'Open portal', 'Edit', 'Delete' (highlighted with a red box), and 'Create portal'. Below the buttons is a search bar 'Find portals' and a table of portals.

Name	Status	Link	Type	Assistant
Portal1	Updating	https://p-mrr5m4bb.app.iotsitewise.aws	Classic	-
Portal2	Failed	https://p-w6zwmryz.app.iotsitewise.aws	Classic	-
BikeFactory	Active	https://p-f5r29mr5.app.iotsitewise.aws	AI-compatible	Enabled

2. En el cuadro de diálogo Eliminar portal, elija Eliminar administradores y usuarios.

Debe quitar los administradores y usuarios del portal antes de poder eliminarlo. Si el portal no tiene administradores o usuarios, el botón no aparece y puede pasar al siguiente paso.

3. Si está seguro de que desea eliminar todo el portal, escriba **confirm** en el campo para confirmar la eliminación.

4. Elija Eliminar.

Cree paneles con AWS CLI

Note

La función SiteWise Supervisar ya no está disponible para los nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte [SiteWise Supervisar el cambio de disponibilidad](#).

Al definir visualizaciones (o widgets) en los paneles mediante el AWS CLI, debe especificar la siguiente información en el documento `dashboardDefinition` JSON. Esta definición es un parámetro de las operaciones [CreateDashboard](#) y [UpdateDashboard](#).

`displaySettings`

La configuración de pantalla con los siguientes parámetros:

- `numRows`— Número de filas en el diseño del cuadro de mandos. El ancho de cada fila es de tamaño de celda.
- `numColumns`— Número de columnas en el diseño del panel de control. Cada columna mide el tamaño de una celda de ancho.
- `cellSize`— (Opcional) El tamaño de una celda del diseño en píxeles. Debe ser un número positivo. El valor predeterminado es 10.
- `significantDigits`— (Opcional) Número de dígitos significativos que se mostrarán en el panel de control. El valor predeterminado es 4.

`querySettings`

La información de la consulta con el siguiente parámetro:

- `refreshRate`— (Opcional) La velocidad a la que se actualizan los datos en milisegundos. Acepta los siguientes valores: 1000, 5000, 10000, 60000, 300000.

`defaultViewport`

Si no se proporciona, el valor predeterminado es de los últimos cinco minutos. Contiene los siguientes parámetros:

- `duration`— (Opcional) Determina qué tan atrás se deben consultar los datos a partir del presente.
- `start`— (Opcional) Es de tipo Fecha. El intervalo de tiempo de inicio de la consulta de datos. Necesita especificar una `end` fecha.
- `end`— (Opcional) Es de tipo Fecha. El intervalo de tiempo de finalización de la consulta de datos. Necesita especificar una `start` fecha.

widgets

Una lista de estructuras de definición de widgets que contiene la siguiente información:

`type`

El tipo de widget. AWS IoT SiteWise proporciona los siguientes tipos de widgets:

- `xy-plot`— Un gráfico de líneas o un diagrama de dispersión según la configuración.
- `bar-chart`: un gráfico de barras.
- `kpi-chart`— Un gráfico de indicadores clave de rendimiento.
- `status-timeline`— Un widget de estado que visualiza y navega por los datos de series temporales de una o más fuentes de datos.
- `text`— Un widget de texto.
- `table`: un widget de tabla.

`id`

Un identificador único para el widget.

`x`

La posición horizontal del widget, empezando por la izquierda del panel de control. Este valor se refiere a la posición del widget en la cuadrícula del panel.

`y`

La posición vertical del widget, empezando por la parte superior del tablero. Este valor se refiere a la posición del widget en la cuadrícula del panel.

Z

El orden relativo de los widgets. Si se superponen, se muestra un widget con un valor Z más grande delante del widget con un valor Z más bajo.

width

El ancho del widget, expresado en el número de celdas del panel de control.

height

La altura del widget, expresada en el número de celdas del tablero.

properties

Una lista de propiedades del widget. Varía según el tipo de widget. Consulte el [kit de aplicaciones de IoT](#) para obtener más información.

Example Ejemplo de definición de panel

En el ejemplo siguiente se define un panel a partir de una carga almacenada en un archivo JSON.

```
aws iotsitewise create-dashboard \  
  --project-id a1b2c3d4-5678-90ab-cdef-eeeeeeEXAMPLE \  
  --dashboard-name "Example Dashboard" \  
  --dashboard-definition file://dashboard-definition.json
```

El siguiente ejemplo JSON para `dashboard-definition.json` define un panel con los siguientes widgets de visualización:

```
{  
  "displaySettings": {  
    "numColumns": 200,  
    "numRows": 1000,  
    "cellSize": 20,  
    "significantDigits": 4  
  },  
  "widgets": [{  
    "id": "0t73JcxUoc6oEXAMPLE",  
    "type": "xy-plot",  
    "width": 33,  
    "height": 20,  
    "x": 0,
```

```
"y": 0,
"z": 0,
"properties": {
  "aggregationType": "AVERAGE",
  "queryConfig": {
    "source": "iotsitewise",
    "query": {
      "assets": [{
        "assetId": "97c97abf-e883-47bb-a3f4-EXAMPLE",
        "properties": [{
          "propertyId": "97cc61f4-57a4-4c5f-a82c-EXAMPLE",
          "refId": "692ce941-f3d9-4074-a297-EXAMPLE",
          "aggregationType": "AVERAGE",
          "color": "#7d2105",
          "resolution": "1m"
        }]
      }],
      "properties": [],
      "assetModels": [],
      "alarms": [],
      "alarmModels": []
    }
  },
  "line": {
    "connectionStyle": "linear",
    "style": "solid"
  },
  "symbol": {
    "style": "filled-circle"
  },
  "axis": {
    "yVisible": true,
    "xVisible": true
  },
  "legend": {
    "visible": true,
    "position": "right",
    "width": "30%",
    "height": "30%",
    "visibleContent": {
      "unit": true,
      "asset": true,
      "latestValue": true,
      "latestAlarmStateValue": true,

```

```

        "maxValue": false,
        "minValue": false
    }
}
}, {
  "id": "fto7rF40Ny1EXAMPLE-G",
  "type": "bar-chart",
  "width": 33,
  "height": 20,
  "x": 0,
  "y": 20,
  "z": 0,
  "properties": {
    "aggregationType": "AVERAGE",
    "queryConfig": {
      "source": "iotsitewise",
      "query": {
        "assets": [{
          "assetId": "97c97abf-e883-47bb-a3f4-EXAMPLE",
          "properties": [{
            "propertyId": "c84ca8f3-3dea-478a-afec-EXAMPLE",
            "aggregationType": "AVERAGE",
            "refId": "2960b958-2034-4d6e-bcc2-EXAMPLE"
          }]
        }]
      },
      "properties": [],
      "assetModels": [],
      "alarms": [],
      "alarmModels": [],
      "requestSettings": {
        "aggregation": "AVERAGE"
      }
    }
  },
  "axis": {
    "showX": true,
    "showY": true
  },
  "styleSettings": {
    "2960b958-2034-4d6e-bcc2-360f1f02e505": {
      "color": "#7d2105"
    }
  }
}
}

```

```
    }  
  }],  
  "querySettings": {  
    "refreshRate": 5000  
  }  
}
```

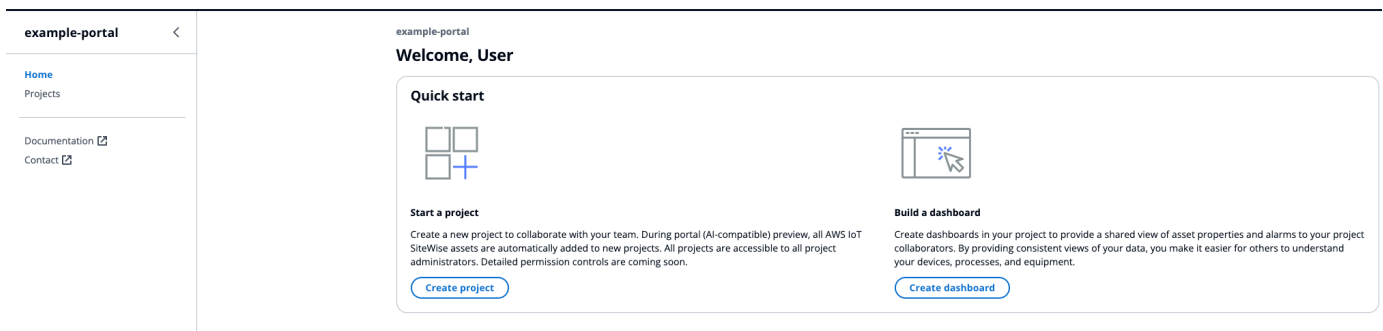
Iniciar sesión en el portal

Note

La función SiteWise Supervisar ya no está disponible para los nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte [SiteWise Supervisar el cambio de disponibilidad](#).

Inicio de sesión de usuario

1. En su navegador, introduzca la URL de la aplicación.
2. Introduzca su nombre de usuario y contraseña y haga clic en el botón Iniciar sesión.
3. Ahora ha iniciado sesión en la aplicación.



Crear un proyecto

Note

La función SiteWise Supervisar ya no está disponible para los nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte [SiteWise Supervisar el cambio de disponibilidad](#).

Creación de proyecto

1. Un proyecto se crea de dos maneras:
 - a. En la página de inicio, en la sección de bienvenida de Inicio rápido, selecciona Crear proyecto.
 - b. En el panel de navegación izquierdo, selecciona Proyectos. Selecciona Crear en la esquina superior derecha para crear un proyecto.
2. En la sección Crear proyectos, introduce un nombre de proyecto y proporciona una descripción opcional.
3. Seleccione Crear.

example-portal

example-portal <

Home
Projects
Documentation [?](#)
Contact [?](#)

Home > Projects > Create

Create project

Create a project to host your dashboards.

Project name
Example project
The project name must contain from 1 and 256 characters. Character count: 15/256.

Project description - optional
An example project.
The project description must contain from 1 and 2048 characters. Character count: 19/2048.

Cancel **Create**

Actualiza un proyecto.

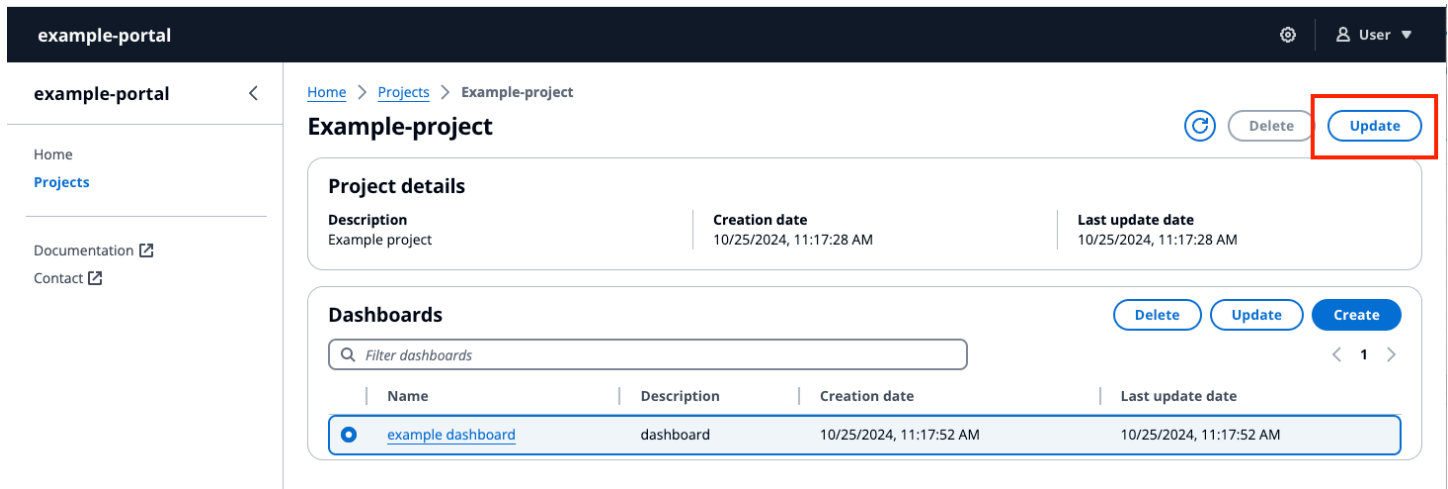
Note

La función SiteWise Supervisar ya no está disponible para los nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte [SiteWise Supervisar el cambio de disponibilidad](#).

Edición de un proyecto

1. Pulse el botón Actualizar en la esquina superior derecha de la página del proyecto para editar los detalles del proyecto.
2. Cambie el nombre del proyecto editando el nombre del proyecto.
3. Cambie la descripción del proyecto editando los detalles de la descripción.

4. Seleccione Actualizar para guardar los cambios.



The screenshot shows the 'Example-project' page in the AWS IoT SiteWise portal. The page has a dark header with 'example-portal' and a user profile. A left sidebar contains navigation links: Home, Projects, Documentation, and Contact. The main content area shows the project details and a list of dashboards. The 'Update' button in the top right corner of the project details section is highlighted with a red box.

Project details

Description Example project	Creation date 10/25/2024, 11:17:28 AM	Last update date 10/25/2024, 11:17:28 AM
---------------------------------------	---	--

Dashboards

Filter dashboards

Name	Description	Creation date	Last update date
example dashboard	dashboard	10/25/2024, 11:17:52 AM	10/25/2024, 11:17:52 AM

Eliminación de un proyecto

Note

La función SiteWise Supervisor ya no está disponible para los nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte [SiteWise Supervisor el cambio de disponibilidad](#).

Eliminación de proyecto

1. Solo puede eliminar el proyecto después de eliminar todos los paneles del proyecto.
2. Seleccione el botón Eliminar en la parte superior derecha de la página del proyecto.
3. Vuelva a confirmar que desea eliminar el proyecto.
4. Seleccione Eliminar para eliminar el proyecto.

The screenshot shows the AWS IoT SiteWise portal interface. At the top, there is a navigation bar with 'example-portal' on the left and a user profile 'User' on the right. Below the navigation bar, the main content area is titled 'Example-project'. On the left side of this area, there is a sidebar with 'example-portal' and a list of links: 'Home', 'Projects', 'Documentation', and 'Contact'. The main content area is divided into two sections. The first section is 'Project details', which contains three columns: 'Description' (Example project), 'Creation date' (10/25/2024, 11:17:28 AM), and 'Last update date' (10/25/2024, 11:17:28 AM). The second section is 'Dashboards', which features a search bar labeled 'Filter dashboards' and three buttons: 'Delete', 'Update', and 'Create'. Below the search bar is a table with columns for 'Name', 'Description', 'Creation date', and 'Last update date'. The table is currently empty, and the text 'No dashboards' is displayed in the center of the table area. A red box highlights the 'Delete' button in the top right corner of the 'Project details' section, and another red box highlights the 'No dashboards' text in the 'Dashboards' section.

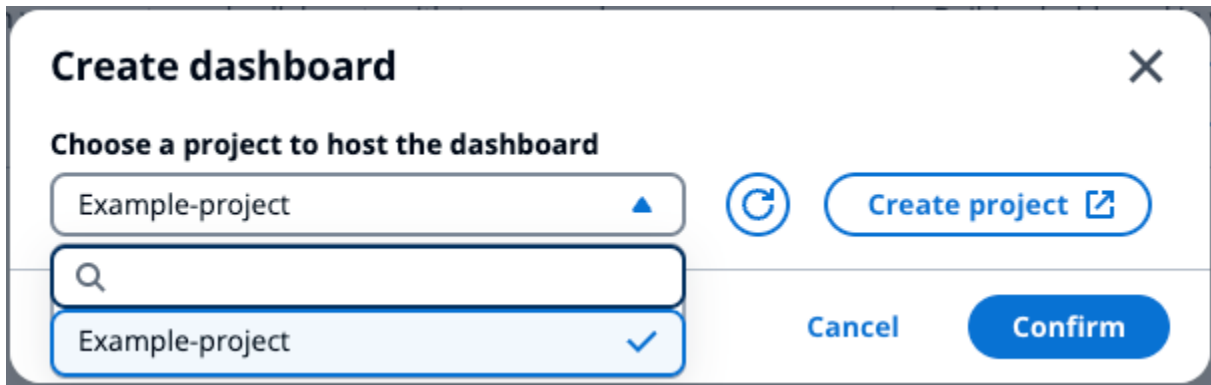
Creación de un panel

Note

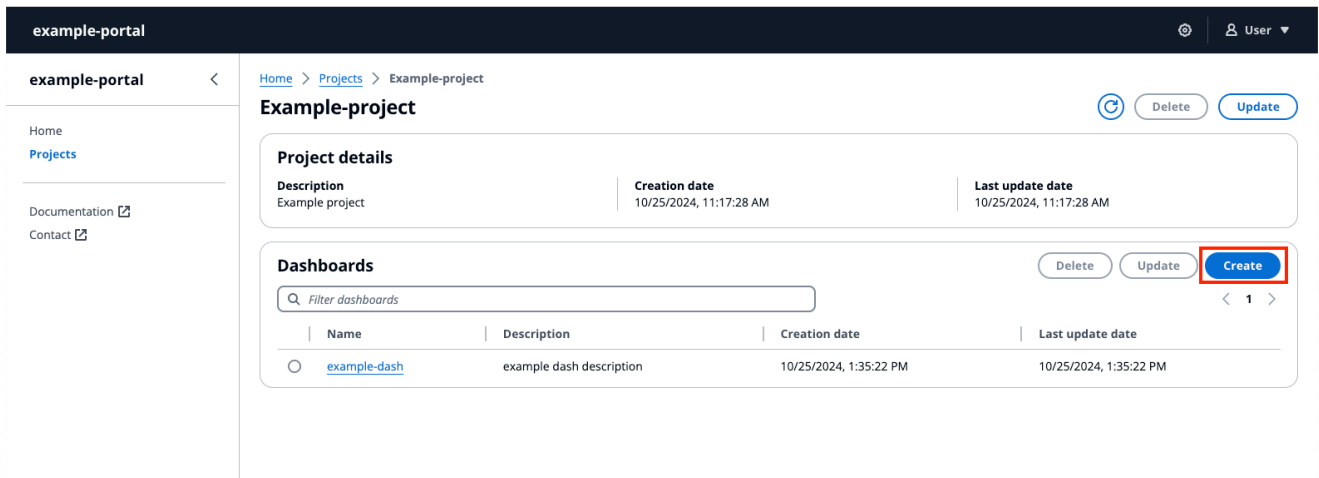
La función SiteWise Supervisar ya no está disponible para los nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte [SiteWise Supervisar el cambio de disponibilidad](#).

Creación de un panel

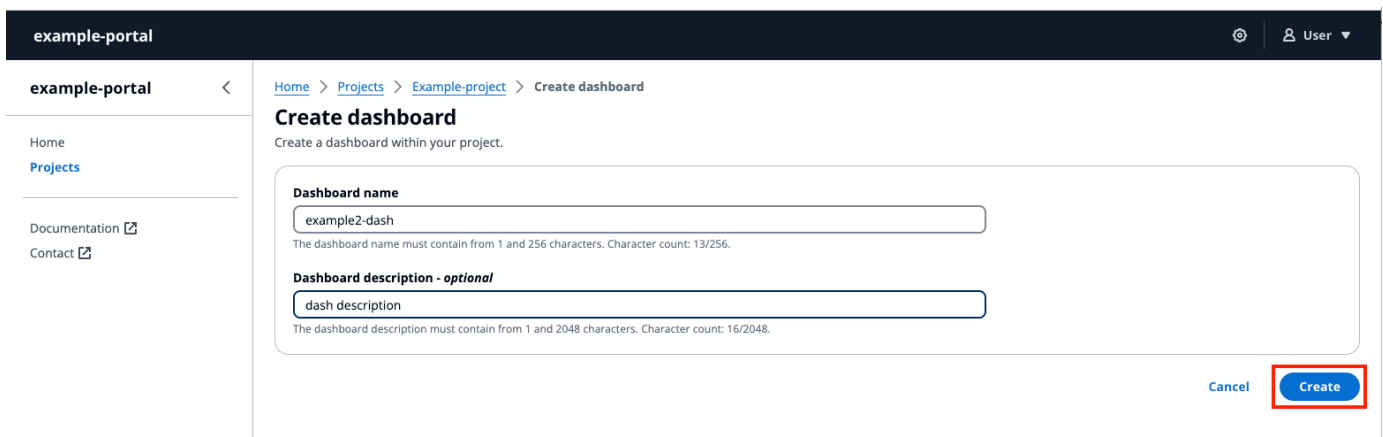
1. Cree un panel de dos maneras:
 - a. Cree un panel desde Crear un panel en la página de inicio.
 - i. Para crear el panel en un proyecto existente, elija un nombre de proyecto en el menú desplegable de la sección Elija un proyecto para alojar el panel.
 - ii. Si no tiene un proyecto, elija Crear proyecto y, a continuación, seleccione Confirmar.



- b. Cree un panel a partir de un proyecto en la sección Proyectos, en Paneles de control.



2. Selecciona Crear en la esquina superior derecha.
3. Introduzca un nombre para el panel y, si lo desea, proporcione una descripción del panel.
4. Seleccione Crear.



5. Configura tu panel de control recién creado.

Actualización de un panel

Note

La función SiteWise Supervisar ya no está disponible para los nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte [SiteWise Supervisar el cambio de disponibilidad](#).

En la sección Cuadros de mando se enumeran los cuadros de mando del proyecto. Seleccione un panel de la lista.

Actualización de un panel

1. Seleccione el panel de control que desee actualizar.

The screenshot shows the 'Example-project' dashboard management interface. At the top, there are navigation links for 'Home', 'Projects', and 'Example-project'. Below this, the 'Example-project' title is followed by 'Delete' and 'Update' buttons. The 'Project details' section shows 'Description: Example project', 'Creation date: 10/25/2024, 11:17:28 AM', and 'Last update date: 10/25/2024, 11:17:28 AM'. The 'Dashboards' section has a search bar and 'Delete', 'Update', and 'Create' buttons. A table lists the dashboard 'example-dash' with its description and dates. The 'Update' button for this dashboard is highlighted with a red box.

2. Actualice el nombre del panel y, si lo desea, la descripción del panel. Seleccione Actualizar para guardar los cambios.

The screenshot shows the 'Update dashboard: example-dash' form. It has a breadcrumb trail: 'Home > Projects > example-dash > Update'. The form title is 'Update dashboard: example-dash'. There are two input fields: 'Dashboard name' with the value 'example-update-dash-name' and a character count of 24/256; and 'Dashboard description - optional' with the value 'update dash description' and a character count of 23/2048. At the bottom right, there are 'Cancel' and 'Update' buttons, with the 'Update' button highlighted by a red box.

Eliminación de un panel

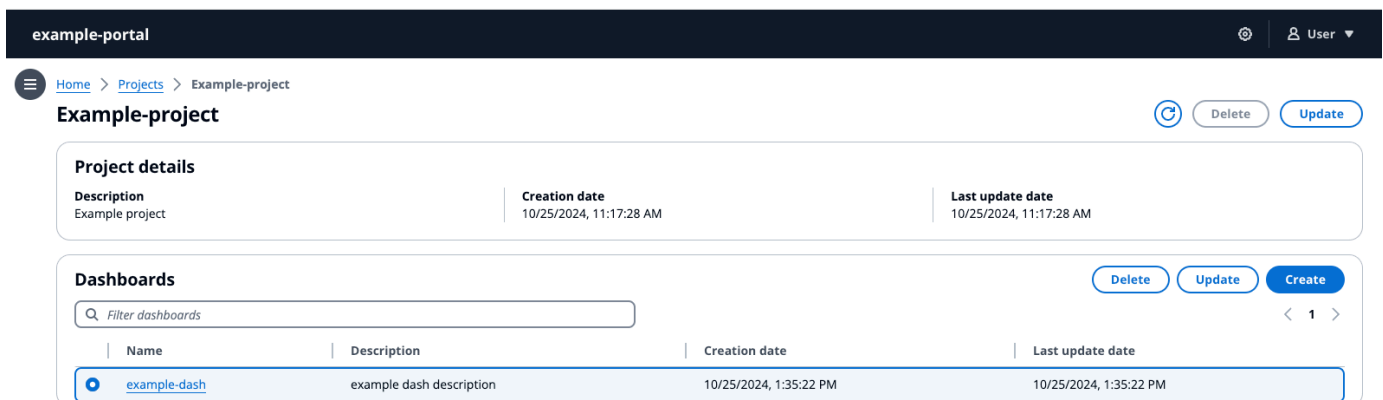
Note

La función SiteWise Supervisar ya no está disponible para los nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte [SiteWise Supervisar el cambio de disponibilidad](#).

En la sección Cuadros de mando se enumeran los cuadros de mando del proyecto. Seleccione un panel de la lista.

Eliminación de un panel

1. Seleccione el panel de control que desee eliminar.



The screenshot shows the AWS IoT SiteWise interface for an 'Example-project'. The page title is 'Example-project' and it includes navigation links for 'Home' and 'Projects'. The 'Project details' section shows the description 'Example project', creation date '10/25/2024, 11:17:28 AM', and last update date '10/25/2024, 11:17:28 AM'. The 'Dashboards' section has a search bar and a table with the following data:

Name	Description	Creation date	Last update date
example-dash	example dash description	10/25/2024, 1:35:22 PM	10/25/2024, 1:35:22 PM

2. Seleccione Eliminar para eliminar el panel. Esto no se puede deshacer.

Configurar panel

Note

La función SiteWise Monitor ya no está disponible para los nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte [SiteWise Supervisar el cambio de disponibilidad](#).

En la sección Cuadros de mando se enumeran los cuadros de mando del proyecto. Seleccione un panel de la lista. El modo de edición le permite configurar su panel añadiendo widgets y configurándolos. El botón Vista previa le permite visualizar los cambios.



Pasos para configurar el panel de control:

- Arrastra y suelta distintos tipos de widgets de datos en el lienzo del panel para visualizar los datos.
- Añada datos a los widgets que desee desde el explorador de recursos de la izquierda. El explorador de recursos consta de las secciones de activos modelados, no modelados y dinámicos. Busque por nombre de activo o nombre de propiedad. Seleccione la propiedad que desee añadir y pulse Añadir.
- Ajuste el diseño y el estilo cambiando las configuraciones de los widgets. Configure los componentes, incluidos el título, los umbrales y otros detalles de configuración.
- Configure el intervalo de tiempo en el que se muestran los datos.
 - Seleccione el intervalo de tiempo en el que se muestran los datos. Elige un rango de tiempo y una frecuencia de actualización en la esquina superior derecha y personaliza el rango. Seleccione la velocidad a la que se van a actualizar los datos en el menú.
 - Seleccione el intervalo de tiempo en un widget utilizando el trackball, la rueda de desplazamiento del ratón o haciendo clic con el botón derecho del ratón. Esto mueve el rango de tiempo de la pantalla.
- Seleccione Save.

Temas

- [Explorador de recursos](#)
- [Widgets](#)
- [Configura los widgets](#)
- [Usa widgets](#)
- [Alarmas en los widgets](#)

- [AWS IoT SiteWise Uso del asistente en los widgets](#)
- [Ejemplos de preguntas para hacerle al Asistente AWS IoT SiteWise](#)

Explorador de recursos

Note

La función SiteWise Monitor ya no está disponible para los nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte [SiteWise Supervisar el cambio de disponibilidad](#).

En esta sección se describen los activos modelados, no modelados y dinámicos. Elija activos de cualquiera de los tres, agréguelos a sus widgets y visualícelos.

Temas

- [Modelado](#)
- [Sin modelar](#)
- [Activos dinámicos](#)

Modelado

Note

La función SiteWise Supervisar ya no está disponible para los nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte [SiteWise Supervisar el cambio de disponibilidad](#).

En esta sección se describe el proceso de selección y visualización de los activos modelados.

Selección de activos

Los activos se pueden consultar de la siguiente manera:

- Busque el nombre de un activo. Usa un comodín. * Por ejemplo, Wind* devuelve los nombres de los activos que comienzan por el textoWind. Debe [integrarse con AWS IoT TwinMaker](#) para poder utilizar esta función.

- Todos los activos se muestran de forma predeterminada.

De los activos de la lista, filtre por nombre, descripción, ID o ID de modelo de activo. Seleccione un activo para enumerar sus propiedades (flujos de datos) y sus alarmas.

Selección de flujo de datos

Los flujos de datos se muestran debajo del menú Flujos de datos. Filtre los flujos de datos listados por metadatos de [propiedad](#) en la versión <https://docs.aws.amazon.com/iot-sitewise/más reciente de//APIReference>. Seleccione uno o más flujos de datos en función del widget seleccionado.

- KPI y Gauge solo admiten un único flujo de datos.
- Los widgets restantes admiten múltiples flujos de datos con selección múltiple.

Selección de alarmas


AWS IoT SiteWise las alarmas se muestran debajo del menú de flujos de datos de alarmas. Filtre los flujos de datos de alarma enumerados por metadatos de alarma. El nombre, la propiedad de entrada y el ID del modelo compuesto son algunos de los metadatos que se utilizan para filtrar. Seleccione uno o más flujos de datos en función del widget seleccionado.

- KPI y Gauge solo admiten una sola alarma.
- Los widgets restantes admiten varias alarmas con selección múltiple.

Visualización de activos modelada

1. Arrastra el widget al lienzo. Seleccione las propiedades de cada panel de widgets para crear un cuadro de mando.
2. La opción Filtrar filtra los activos para elegir el activo que se va a visualizar. El filtrado se realiza por texto, propiedad o valor. El filtrado es para los activos cargados en el navegador y no para el filtrado de fondo.
3. Busca para incluir un activo en una lista y añadirlo a tu widget.
4. Agrega el activo al widget en el lienzo.
5. Elija Restablecer para seleccionar otro activo o realizar modificaciones en el activo elegido.

6. Guarde el panel de control. En el modo de vista previa, seleccione distintos activos en el menú desplegable para supervisar las propiedades de cada activo sin tener que reconstruir los paneles de datos.

 Note

La rueda de ajustes de la parte derecha muestra las preferencias que el usuario puede elegir, como el tamaño de la página, las primeras columnas adhesivas, las últimas columnas adhesivas y las preferencias de columnas. Personalice sus preferencias y seleccione Confirmar para aplicar los cambios.

Modeled | Unmodeled | Dynamic assets

Assets (1)

Browse through your asset hierarchy and select an asset to view its associated data streams.

Root

Search Search

Filter < 1 >

Name	Description
<input checked="" type="radio"/> Demo Wind Farm Asset	

Asset properties (8)

Select a modeled datastream to add to a selected widget

Filter < 1 >

<input type="checkbox"/>	Name	Unit	Data ty...	Latest ...	Latest value time
<input type="checkbox"/>	Total Aver...		DOUBLE	37478.2303	2024-10-01 09:50:00 p.m.
<input type="checkbox"/>	Total Aver...		DOUBLE	6.0000	2022-10-26 03:42:43 p.m.
<input type="checkbox"/>	Total Aver...		DOUBLE	555.0000	2022-10-13 11:59:49 p.m.
<input type="checkbox"/>	Code		INTEGER	300.0000	2022-10-13 10:59:28 p.m.
<input type="checkbox"/>	Reliability ...		STRING	Mary Major	2022-10-13 10:59:28 p.m.
<input type="checkbox"/>	Location		STRING	Renton	2022-10-13 10:59:28 p.m.
<input type="checkbox"/>	Total Over...		DOUBLE	900.0000	2024-10-01 09:50:00 p.m.
<input type="checkbox"/>	recipient a...		STRING	54a88418-...	2022-10-26 03:42:43 p.m.

Sin modelar

Note

La función SiteWise Supervisar ya no está disponible para los nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte [SiteWise Supervisar el cambio de disponibilidad](#).

En esta sección se describe la búsqueda de flujos de datos no modelados y su adición a los widgets para su visualización.

Visualización de flujos de datos sin modelar

1. Arrastra el widget al lienzo. Seleccione las propiedades de cada panel de widgets para crear un cuadro de mando.
2. Los flujos de datos no modelados se enumeran en la sección Series temporales. Tienen propiedades que se pueden personalizar.
3. La opción Filtrar filtra los flujos de datos para visualizarlos. El filtrado es para los flujos de datos cargados en el navegador y no para el filtrado final.
4. Añada el flujo de datos al widget del lienzo.
5. Elija Restablecer para deseleccionar el flujo de datos.
6. Guarde el panel de control. En el modo de vista previa, seleccione distintos activos en el menú desplegable para supervisar las propiedades de cada activo sin tener que reconstruir los paneles de datos.

Note

La rueda de ajustes de la parte derecha muestra las preferencias que el usuario puede elegir, como el tamaño de la página, las primeras columnas adhesivas, las últimas columnas adhesivas y las preferencias de columnas. Personalice sus preferencias y seleccione Confirmar para aplicar los cambios.

Modeled
Unmodeled
Dynamic assets

Time series (1)

Select a unmodeled datastream to add to a selected widget

Filter
< 1 > ⚙️

<input checked="" type="checkbox"/>	Alias	ID	Data type	Latest val...	Latest valu...
<input checked="" type="checkbox"/>	DemoDisass...	33e78bb9-39...	DOUBLE	3.0283	2024-10-01 0...

Activos dinámicos

i Note

La función SiteWise Supervisar ya no está disponible para los nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte [SiteWise Supervisar el cambio de disponibilidad](#).

El nuevo SiteWise Monitor permite a los clientes cambiar los activos de forma dinámica para un modelo de activos seleccionado. Puede visualizar las propiedades de diferentes activos seleccionándolos en un menú desplegable.

Visualización dinámica de activos

1. Seleccione la pestaña Activos dinámicos en el explorador de recursos.
2. Seleccione un modelo de activos para enumerar los activos en el menú desplegable.
3. Seleccione el activo predeterminado en el menú desplegable.
4. Elija Definir modelo de activo para seleccionar el modelo de activo.
5. Guarde el panel de control. En el modo de vista previa, elija diferentes activos en el menú desplegable para supervisar las propiedades de cada activo, sin tener que reconstruir los paneles de datos.

Resource explorer | <

Modeled | **Unmodeled** | **Dynamic assets**

Dynamic asset visualizations allow you to build one visualization to represent any asset of a specified asset model. [Learn more](#)

Asset model
Select an asset model to add the associated properties into your dynamic display.

Demo Wind Farm Asset Model 2022-Oct-14 05:59:28 ▲

🔍 *Filter asset models*

Demo Turbine Asset Model 2022-Oct-14 05:59:12
This is an asset model used in the IoT SiteWise Demo for representing a turbine in a wind farm. It will be deleted at the end of the demo.

Demo Wind Farm A Demo Turbine Asset Model 2022-Oct-14 05:59:12
This is an asset model used in the IoT SiteWise Demo for representing a wind farm. It will be deleted at the end of the demo. ✓

Finished loading asset models.

Set asset model

Widgets

Note

La función SiteWise Monitor ya no está disponible para los nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte [SiteWise Supervisar el cambio de disponibilidad](#).

Los widgets admiten una amplia gama de funciones, que incluyen alarmas, transmisión en vivo de alto rendimiento y una sincronización fluida con otros componentes del IoT App Kit. El panel de control admite los siguientes widgets:

- **Línea:** el widget Line es un widget de visualización que muestra las tendencias y los cambios a lo largo del tiempo. Consiste en una serie de puntos de datos, cada uno representado por un punto

o marcador, conectados por segmentos de línea recta para crear un gráfico lineal. Es compatible con una amplia gama de funciones, que incluyen alarmas, umbrales, transmisión en vivo de alto rendimiento y una sincronización fluida con otros componentes del IoT App Kit. Este widget se puede personalizar para comunicar datos complejos de forma clara y concisa.

- **Gráfico de barras:** el gráfico de barras es una poderosa herramienta de visualización que muestra datos de series temporales. Es compatible con una amplia gama de funciones, que incluyen alarmas, transmisión en vivo de alto rendimiento y una sincronización fluida con otros componentes del IoT App Kit.
- **Cronología:** el widget Cronología proporciona una forma de visualizar y navegar por los datos de series temporales procedentes de fuentes de datos. Es exclusivo para mostrar los valores del flujo de datos en colores distintos en la línea de tiempo. Admite un amplio conjunto de funciones que incluyen alarmas, transmisión en vivo de alto rendimiento y una sincronización fluida entre otros componentes del IoT App Kit. Se utiliza mejor para mostrar tipos de datos no numéricos/
- **KPI:** el componente Indicador clave de rendimiento (KPI) proporciona una representación compacta de una visión general de las propiedades de sus activos. Soporta alarmas y umbrales. Esta descripción general proporciona información fundamental sobre el rendimiento general de sus dispositivos, equipos y procesos. El KPI solo admite un único flujo de datos o alarma, y no varios flujos de datos.
- **Medidor:** el componente Gauge proporciona una representación compacta de una visión general de las propiedades de sus activos. Se utiliza para visualizar información fundamental sobre el rendimiento general de sus dispositivos, equipos o procesos. Es funcionalmente igual que el KPI, pero visualmente diferente. El indicador muestra el valor, el umbral y el rango de valores del flujo de datos. Puede interactuar con AWS IoT los datos de una o más fuentes de datos con Gauge.
- **Tabla:** el componente Table proporciona un formato compacto para ver uno o más flujos de datos de una o más fuentes de datos de series temporales. Muestra los activos con la propiedad, el valor más reciente y la unidad en forma de tabla. Soporta AWS IoT SiteWise alarmas.
- **Texto:** el widget de texto ayuda a escribir texto con varios colores y fuentes. Puedes crear un enlace asociando un texto a una URL. Los campos Propiedades y Umbrales no están habilitados para este widget.



Configura los widgets

Note

La función SiteWise Supervisar ya no está disponible para los nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte [SiteWise Supervisar el cambio de disponibilidad](#).

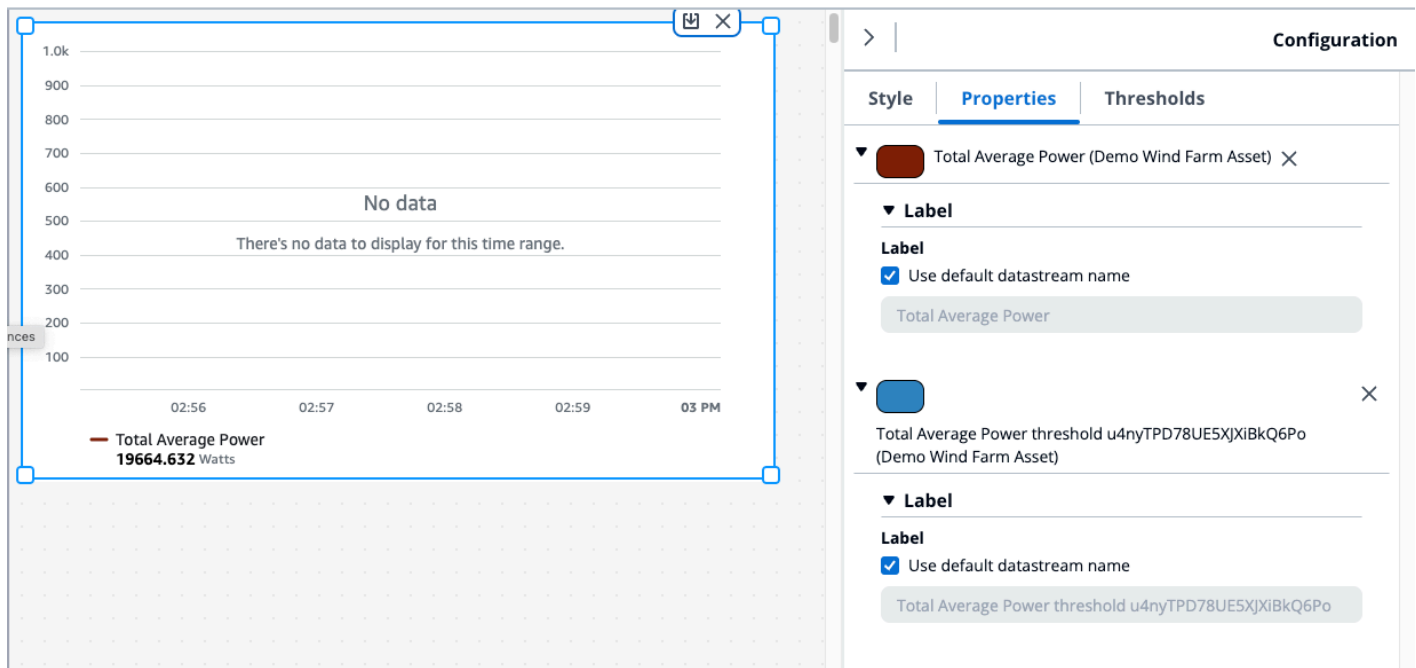
Una vez agregado el widget al panel de control, puede configurarlo seleccionando el icono de configuración en el panel derecho.

- Estilo: agrega un título al título del widget. Los distintos widgets tienen diferentes configuraciones. A continuación se muestran algunos ejemplos.
 - Widget de barra:
 - Resolución y agregación: defina aquí los valores de resolución y agregación.
 - Dar formato a los datos: establece los decimales en función del número de decimales que se van a mostrar.
 - Estilo de visualización: seleccione los valores que desee mostrar.
 - Eje: elija mostrar el eje.
 - Widget de línea:
 - Resolución y agregación: defina aquí los valores de resolución y agregación.
 - Dar formato a los datos: establece los decimales en función del número de decimales que se van a mostrar.
 - Eje Y: agrega una etiqueta y valores mínimo y máximo.
 - Estilo del widget: seleccione el tipo de línea, el estilo de línea, el grosor de la línea y los valores de la forma de los puntos de datos.
 - Leyenda: elija la alineación y la visualización.
 - Widget de indicador:
 - Resolución y agregación: defina aquí los valores de resolución y agregación.
 - Dar formato a los datos: establece los decimales en función del número de decimales que se van a mostrar.
 - Estilo de visualización: seleccione los valores que desee mostrar.
 - Eje Y: añade una etiqueta y valores mínimo y máximo.

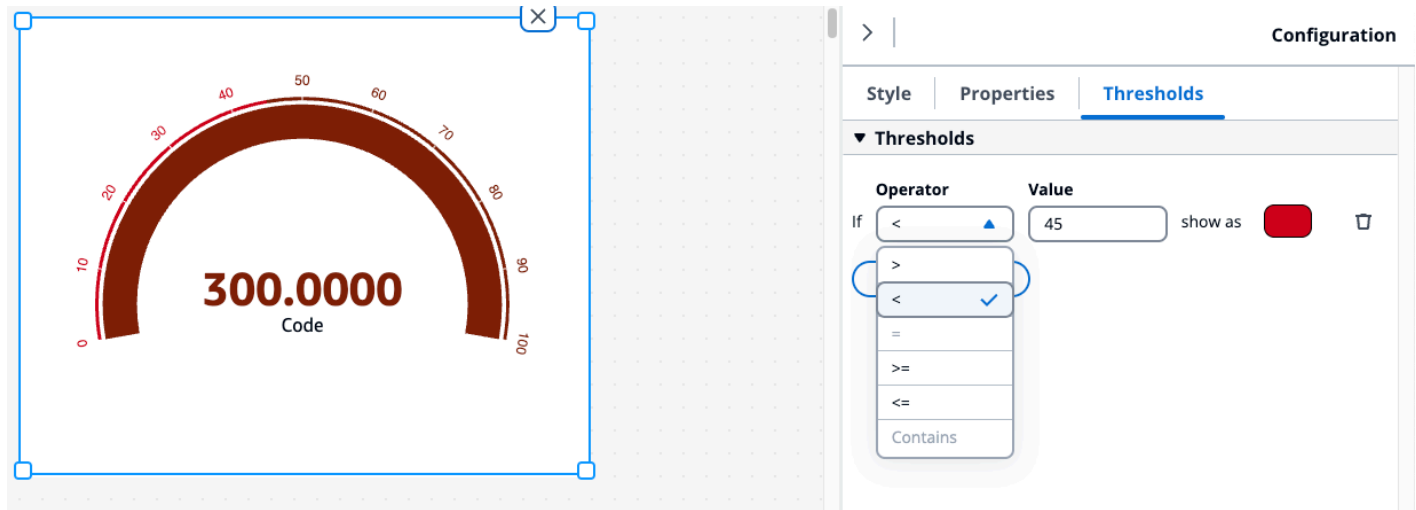
- Fuentes: seleccione los valores de tamaño de fuente, tamaño de fuente de unidad y tamaño de fuente de etiqueta.

Configuration for a Line widget

- Propiedades: todas las propiedades de los widgets se muestran en esta sección. Los distintos widgets tienen propiedades diferentes. A continuación se muestran algunos ejemplos.
 - Widget de línea:
 - Etiqueta: elija usar el nombre predeterminado del flujo de datos o asigne un nombre nuevo.
 - Estilo: defina el tipo de línea y el estilo de línea en el número de decimales que desea mostrar.
 - Eje Y: seleccione los valores con el estilo predeterminado, muestre los controles del eje Y y establezca los valores mínimo y máximo.
 - Widget de tabla:
 - Etiqueta: elija usar el nombre predeterminado del flujo de datos o asigne un nombre nuevo.
 - Widget de tabla:
 - Etiqueta: elija usar el nombre predeterminado del flujo de datos o asigne un nombre nuevo.



- Umbrales: agrega un umbral para un widget. Los distintos widgets tienen diferentes configuraciones. A continuación se muestran algunos ejemplos.
 - Widget de gráfico de barras:
 - Selecciona Añadir un umbral para añadirlo al widget.
 - Elija Operador y asigne un valor al umbral. Personalice el umbral con un color de la paleta de colores.
 - Puede optar por aplicar el umbral a todos los datos.
 - Widget de línea:
 - Selecciona Añadir un umbral para añadirlo al widget.
 - Elija Operador y asigne un valor al umbral. Personalice el umbral con un color de la paleta de colores.
 - Seleccione cómo mostrar los umbrales en el menú desplegable.
 - Widget de indicador:
 - Selecciona Añadir un umbral para añadirlo al widget.
 - Elija Operador y asigne un valor al umbral. Personalice el umbral con un color de la paleta de colores.



Usa widgets

Note

La función SiteWise Supervisar ya no está disponible para los nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte [SiteWise Supervisar el cambio de disponibilidad](#).

Puede utilizar los widgets del panel de forma individual o seleccionándolos de forma múltiple.

Edita los widgets en el panel

Elige un único widget y edítalo. Para editar varios widgets en el panel, pulse Mayús y haga clic con el botón izquierdo del ratón y seleccione todos los widgets del panel. Una vez seleccionados, los usuarios pueden añadir nuevos flujos de datos y modificar el título del widget en los ajustes de configuración del estilo. Se cambia el título de todos los widgets del panel de control.

Haga clic con el botón derecho en el lienzo y haga lo siguiente:

- Copiar: añade una copia del widget al lienzo.
- Eliminar: elimina el widget.
- Poner al frente: coloca el widget seleccionado en la parte delantera del lienzo.
- Enviar al reverso: envía el widget seleccionado a la parte posterior del lienzo.

Cambiar el tamaño de los widgets

Cambie el tamaño de los widgets de forma individual o en grupo mediante una selección múltiple de los widgets en el panel de control.

Para cambiar el tamaño de los widgets:

- Para cambiar el tamaño de un único widget, selecciónelo y arrástrelo por una esquina para cambiar su tamaño.
- Para cambiar el tamaño de varios widgets, selecciona varios widgets haciendo clic con el botón izquierdo del ratón y arrástralos por una esquina para cambiar su tamaño.

Elimine los widgets del panel

Elimine los widgets de forma individual o en grupo mediante una selección múltiple de los widgets en el panel de control.

Para eliminar los widgets:

- Para eliminar un único widget, selecciónelo, haga clic con el botón derecho del ratón y elija Eliminar. También puedes seleccionar y hacer clic en la X situada en la esquina superior derecha para eliminar el widget.
- Para eliminar varios widgets, seleccione varios widgets con la tecla Mayús y, a continuación, haga clic con el botón izquierdo del ratón y, a continuación, haga clic con el botón derecho y seleccione Eliminar.

Alarmas en los widgets

Note

La función SiteWise Monitor ya no está disponible para los nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte [SiteWise Supervisar el cambio de disponibilidad](#).

Las alarmas le avisan a usted y a su equipo cuando los equipos o los procesos no funcionan de manera óptima. Un rendimiento óptimo de una máquina o de un proceso significa que los valores

de determinadas métricas deben estar dentro de un rango de límites altos y bajos. Cuando estas métricas están fuera de su rango de operación, se debe notificar a los operadores de los equipos para que solucionen el problema. Las alarmas le ayudan a identificar rápidamente los problemas y notificar a los operadores para que maximicen el rendimiento del equipo y de los procesos.

Puede encontrar una alarma asociada a un activo en la pestaña Modelado del explorador de recursos.

- Busque y seleccione un activo.
- Desplácese hacia abajo más allá de la tabla de flujos de datos hasta la sección Flujos de datos de alarmas y amplíela.
- Seleccione una alarma en la tabla Alarmas y pulse Añadir.

Temas

- [Alarmas en distintos widgets](#)

Alarmas en distintos widgets

Note

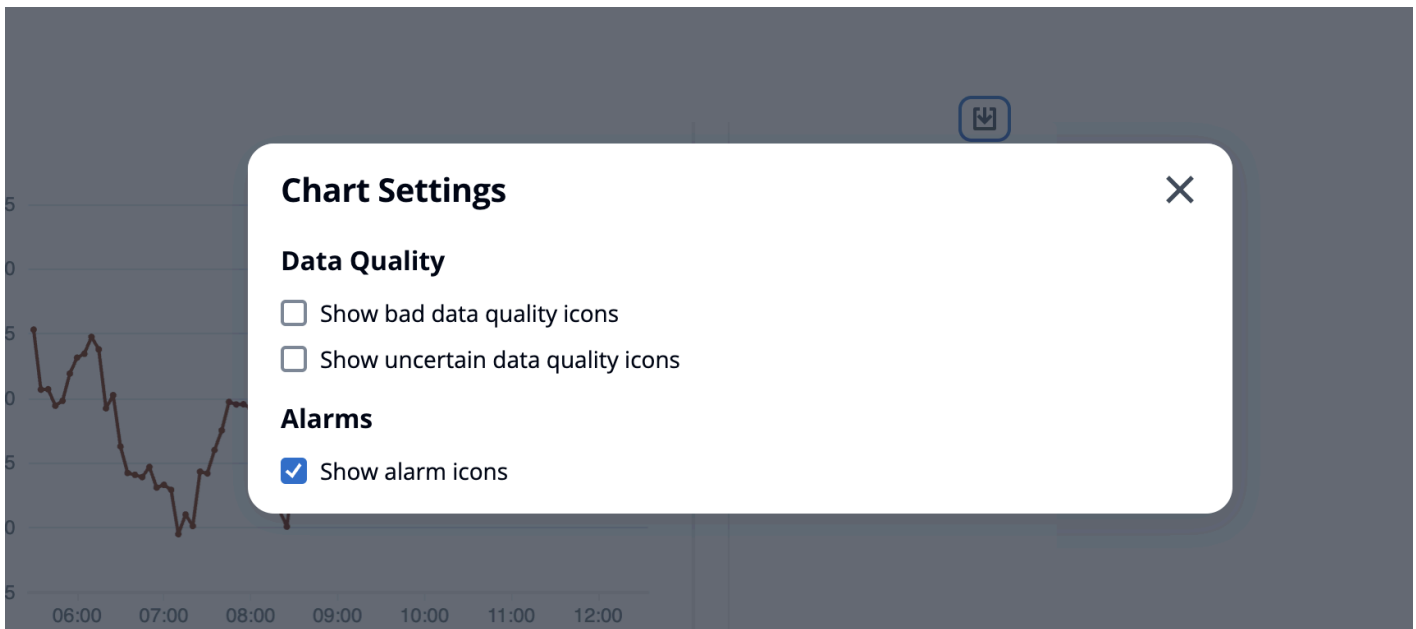
La función SiteWise Monitor ya no está disponible para los nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte [SiteWise Supervisar el cambio de disponibilidad](#).

Para todos los widgets:

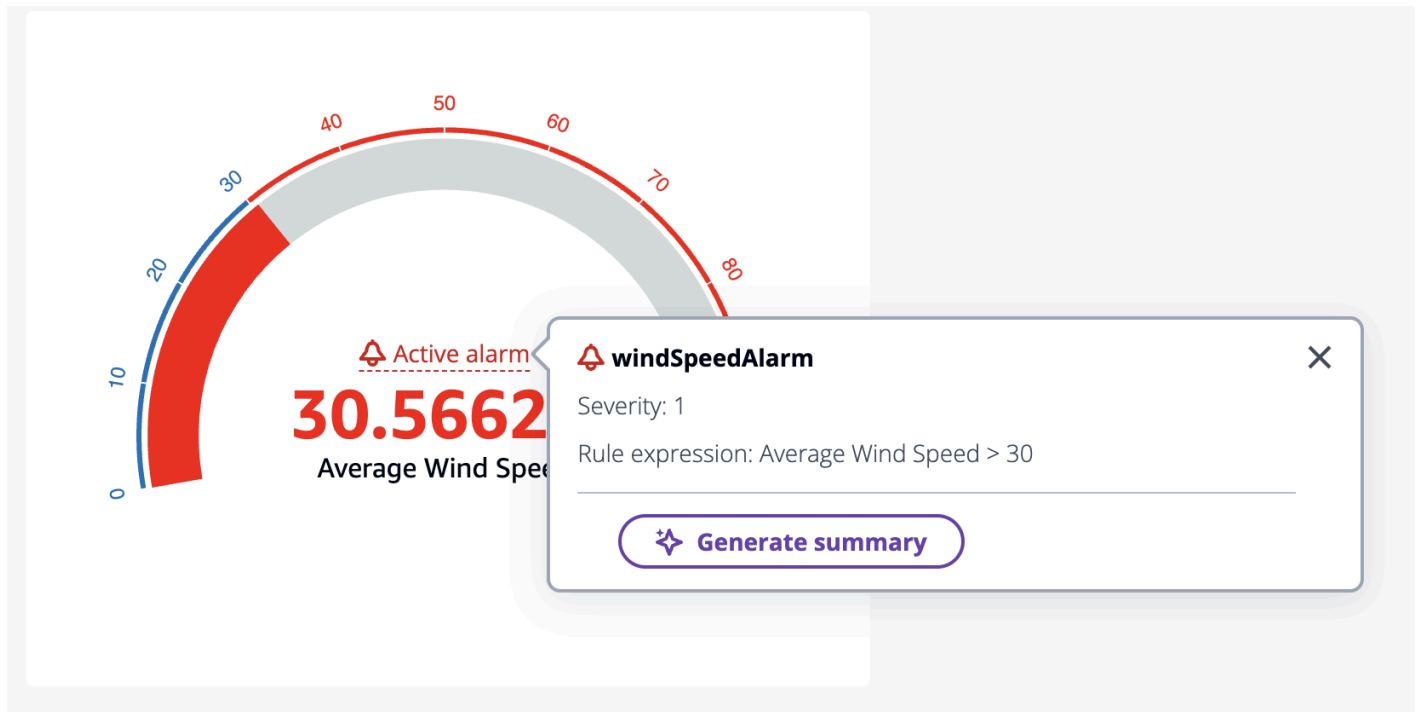
- La configuración de las propiedades del flujo de datos depende del tipo de propiedad que se añada a un widget. Las propiedades del flujo de datos son totalmente compatibles con la configuración de propiedades, mientras que las propiedades de alarma actualmente no permiten la configuración de propiedades.
- Si agrega un flujo de datos de alarma, el flujo de datos de propiedades de entrada asociado también se agrega al gráfico. Si elimina el flujo de datos de la alarma, también se elimina su propiedad de entrada.
- Para controlar individualmente el flujo de datos de propiedades de entrada de una alarma, debe añadirlos por separado.

Los siguientes ejemplos indican cómo algunos widgets utilizan las alarmas.

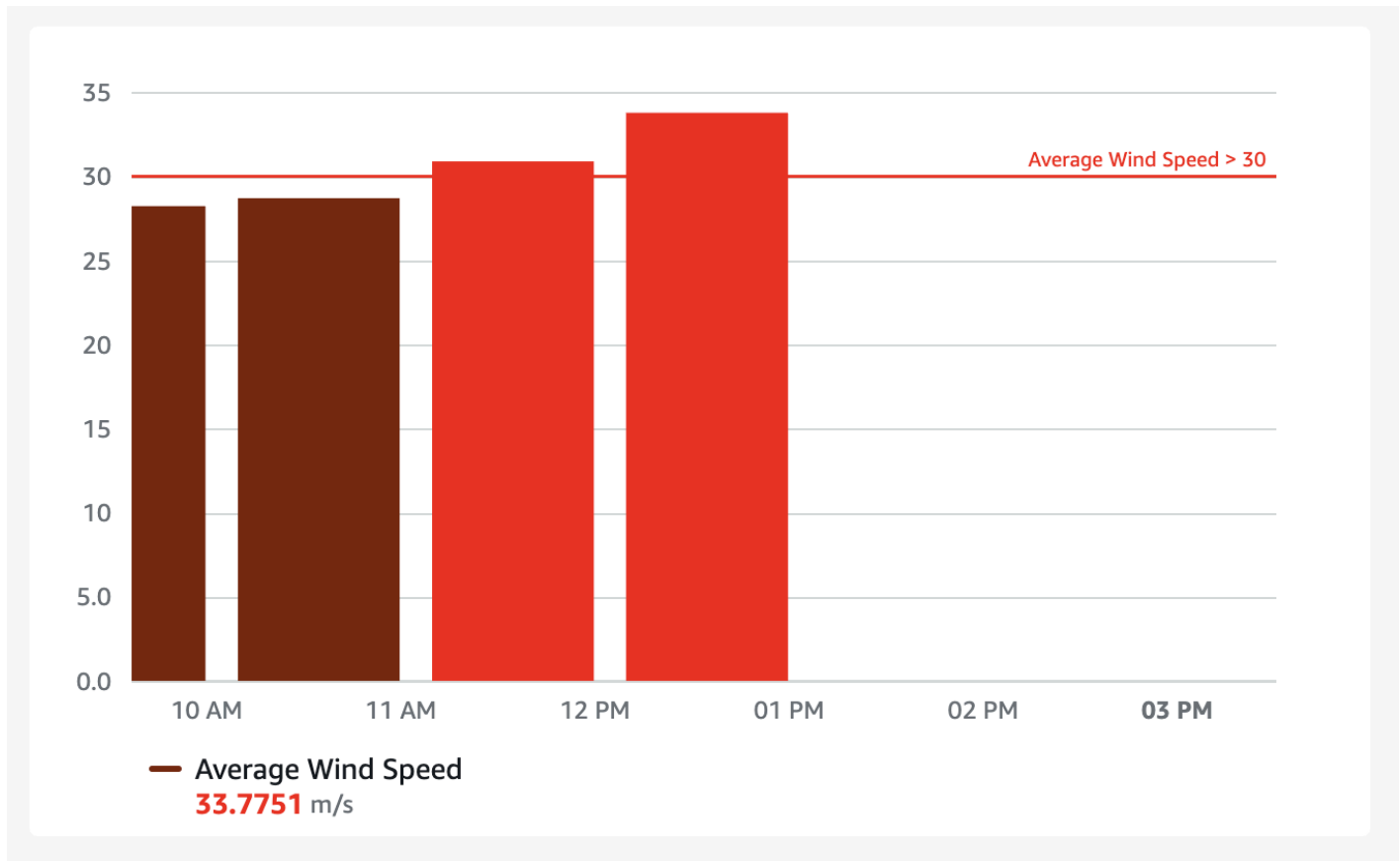
- Gráfico de líneas
 - La alarma y el flujo de datos de sus propiedades de entrada se añaden al gráfico.
 - Puede ver el estado de alarma en la leyenda del gráfico y en forma de iconos que aparecen sobre el flujo de datos cuando la alarma cambia de estado.
 - Puede desactivar los íconos de alarma desde la configuración del gráfico.



- KPI y indicador
 - La alarma y el flujo de datos de sus propiedades de entrada se añaden al widget elegido.
 - El umbral de alarma se añade al widget, que cambia de color en función de su configuración.
 - Puede seleccionar el estado de la alarma en el widget, ver los detalles de la alarma y hacer clic en Generar resumen para llamar AWS IoT SiteWise al y obtener un resumen de la alarma.



- Tabla
 - La alarma y su propiedad de entrada se añaden como una fila en la tabla.
- Gráfico de barras
 - La alarma se agrega como un umbral al gráfico, lo que cambia el color de cualquier flujo de datos que supere el umbral.
 - Puede añadir cualquier flujo de datos asociado por separado.
 - No puede interactuar con el AWS IoT SiteWise Asistente desde el widget.



- Cronología del estado
 - La alarma se añade como umbral a la línea temporal.
 - Se está trabajando en añadir el estado de alarma y sus datos de propiedades de entrada a la línea temporal.
 - No puedes interactuar con el AWS IoT SiteWise Asistente desde el widget.

AWS IoT SiteWise Uso del asistente en los widgets

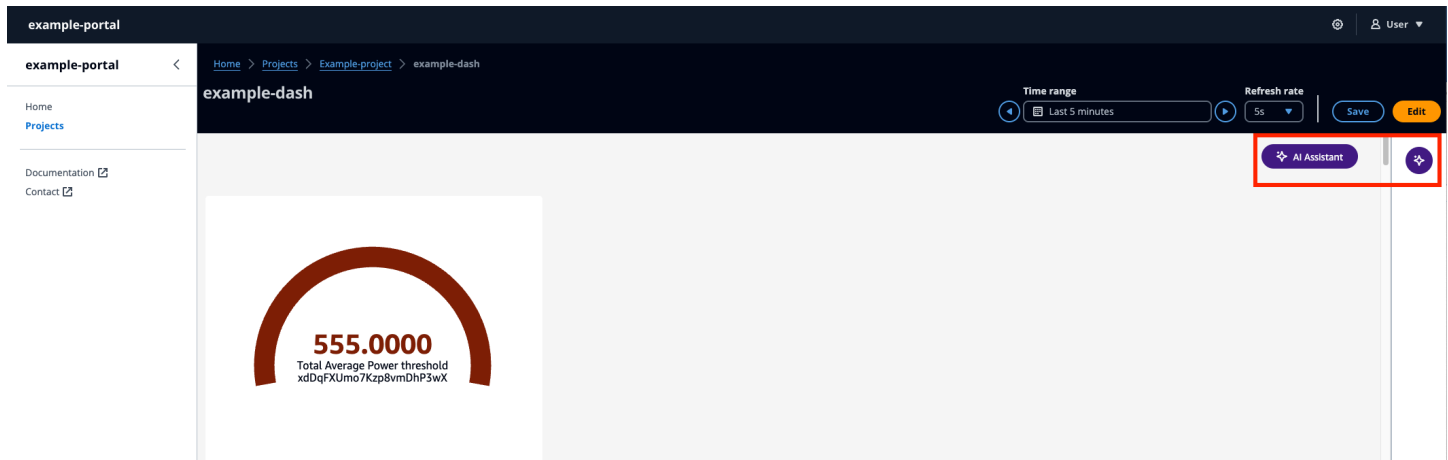
Note

La función SiteWise Monitor ya no está disponible para los nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte [SiteWise Supervisar el cambio de disponibilidad](#).

El AWS IoT SiteWise Asistente es un asistente generativo impulsado por IA. Permite a los usuarios, como los gerentes de planta, los ingenieros de calidad y los técnicos de mantenimiento, obtener

información, resolver problemas y tomar medidas directamente a partir de sus datos operativos y empresariales. El AWS IoT SiteWise Asistente consolida la información de AWS IoT los datos, los modelos de activos, los manuales y la documentación en resúmenes comprensibles de eventos críticos. También permite sesiones interactivas de preguntas y respuestas exhaustivas para facilitar el diagnóstico, la exploración de las causas fundamentales y las recomendaciones guiadas.

El botón del AWS IoT SiteWise Asistente se encuentra en la esquina superior derecha del panel de control. Haga clic en él para activar el Asistente. Solo se puede usar con el modo de vista previa del panel de control.



Utilice el AWS IoT SiteWise Asistente en los siguientes escenarios:

Temas

- [Caso de uso: resúmenes de alarmas](#)
- [Caso de uso: resúmenes situacionales](#)
- [Caso de uso: resúmenes detallados](#)

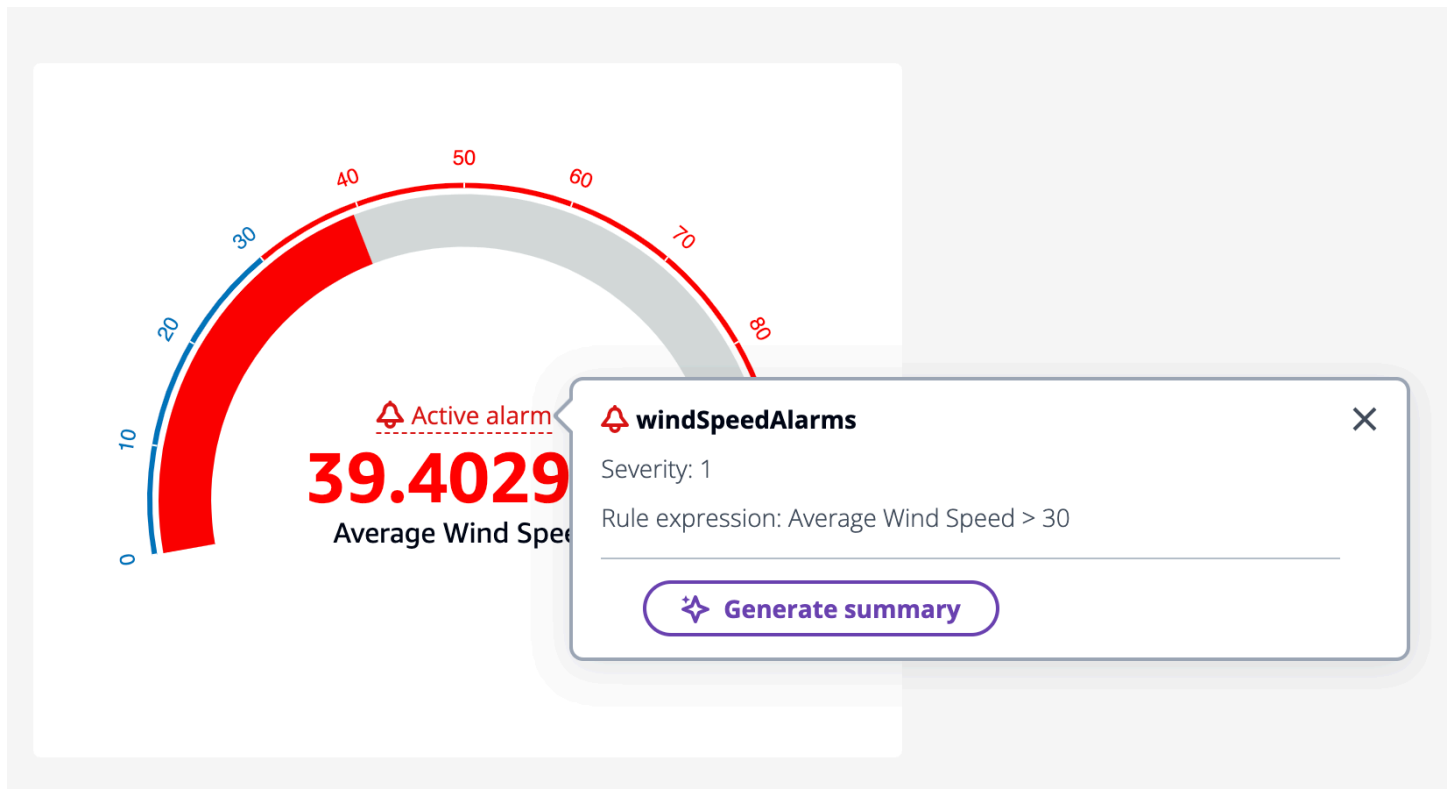
Caso de uso: resúmenes de alarmas

Note

La función SiteWise Monitor ya no está disponible para los nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte [SiteWise Supervisar el cambio de disponibilidad](#).

Resuma la alarma actual de un panel seleccionado en el panel de control. Las alarmas son compatibles con los widgets Line, KPI, Gauge y Table. Elige un widget con una alarma y resúmirlo.

- Selecciona Alarma activa en el widget.
- Se muestra la expresión Gravedad y regla de la alarma.
- Seleccione Generar resumen para generar un resumen.



Caso de uso: resúmenes situacionales

Note

La función SiteWise Monitor ya no está disponible para los nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte [SiteWise Supervisar el cambio de disponibilidad](#).

Seleccione hasta tres widgets para resumirlos. Pueden ser una combinación de widgets y propiedades. Si se seleccionan más de tres, el Asistente devuelve un error.

Genere un resumen de la situación con AWS IoT SiteWise Assistant

1. Haz clic en AI Assistant. Muestra un menú con tres opciones.
 - a. Elementos seleccionados: seleccione solo tres. No puede seleccionar más de tres.
 - b. Borrar todo: borra la selección.
 - c. Generar resumen: genere un resumen de los elementos seleccionados.
2. Seleccione Generar resumen para generar el resumen de los elementos seleccionados.

La imagen de abajo tiene un widget seleccionado y un resumen del AWS IoT SiteWise Asistente.

The screenshot displays the AWS IoT SiteWise Assistant interface. At the top, there are controls for 'Time range' (Last 10 minutes) and 'Refresh rate' (5s). Below this, a toolbar shows '1/3 items selected', 'Clear all', 'Generate summary', and 'AI Assistant'. The main area features a selected widget for 'Average Wind Speed (m/s)' with a value of 38.3260. The widget includes a gauge chart showing a decreasing trend from 39.40 m/s to 38.33 m/s, with an 'Active alarm' indicator. Below the gauge is a table of properties:

Property	Latest value	Unit
<input type="checkbox"/> Torque (Newton Meter) (Demo Turbine	3691.8240	Nm
<input checked="" type="checkbox"/> Average Wind Speed (Demo Turbine As	38.3260	m/s
<input type="checkbox"/> RotationsPerMinute (Demo Turbine As	29.6543	RPM

To the right, the 'AI Assistant' panel provides 'Property Insights' for the selected 'Average Wind Speed' property:

- Historical Trend Analysis:**
 - The historical analysis of the "Average Wind Speed" property for the "Demo Turbine Asset 1" shows a decreasing trend over the last 5 minutes, from 39.40 m/s at 21:30:00 to 38.33 m/s at 21:35:00 on October 2, 2024.
 - This indicates that the wind speed at the turbine location has been gradually declining during this time period.
- Latest Value Comparison:**
 - The latest value of 38.33 m/s recorded at 21:35:00 on October 2, 2024 is slightly lower than the previous 5-minute reading of 39.40 m/s.
 - This suggests that the wind conditions at the turbine site have become less favorable in the latest resolution window.
- Cross-Property Evaluation:**
 - Since there is only one property, "Average Wind Speed," there are no other properties to evaluate and compare.

A summary at the bottom of the AI Assistant panel states: "In summary, the analysis of the "Average Wind Speed" property indicates a decreasing trend in wind speed over the last 5 minutes, with the latest value being slightly lower than the historical value, suggesting less favorable wind conditions at the turbine site." A search bar at the bottom of the AI Assistant panel contains the text "what is SOP for this property ?".

Caso de uso: resúmenes detallados

Note

La función SiteWise Monitor ya no está disponible para los nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte [SiteWise Supervisar el cambio de disponibilidad](#).

Este es el caso de uso en el que el usuario puede hacer un análisis profundo y acceder SOPs (procedimiento operativo estándar), a los manuales y a la documentación, y considerar los próximos pasos a seguir. Para el ejemplo de la sección anterior, si el usuario decide obtener más información sobre el SOP de esta propiedad, pregunte al Asistente sobre el SOP de esta propiedad. Esto muestra al usuario información detallada sobre el SOP.


En el siguiente ejemplo, se muestra la respuesta a la pregunta «¿Existe algún SOP para la alarma? windSpeedAlarm »

The screenshot displays the AWS IoT SiteWise interface. At the top, there are controls for 'Time range' (Last 5 minutes) and 'Refresh rate' (5s), along with 'Save' and 'Edit' buttons. Below this, a navigation bar shows '2/3 items selected', 'Clear all', 'Generate summary', and 'AI Assistant'. The main content area features a gauge for 'Wind Direction' showing a value of 36.9589 Degrees. To the right, an 'AI Assistant' chat window is open, displaying the question 'is there any SOP for the alarm windSpeedAlarm?' and a detailed response titled 'SOP for "windSpeedAlarm"'. The response includes three sections: '1. Alarm Identification', '2. Data Collection', and '3. Root Cause Analysis', each with a list of bullet points. At the bottom of the chat window, there is a text input field with the placeholder 'Ask me anything about your IoT data' and a send button.

Ejemplos de preguntas para hacerle al Asistente AWS IoT SiteWise

Note

La función SiteWise Monitor ya no está disponible para los nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte [SiteWise Supervisar el cambio de disponibilidad](#).

 Note

- El AWS IoT SiteWise Asistente debe usar un conjunto de datos con un índice de [Amazon Kendra](#) para obtener conocimientos y orientación a nivel empresarial. Si no tiene un índice de Amazon Kendra, consulte [Creación de un índice](#) para crear uno. Añadir un [conjunto de datos](#) mejora la calidad de la respuesta del Asistente. Consulte [Crear un conjunto de datos](#) para obtener más información.
- Algunas preguntas requieren AWS IoT TwinMaker integración. Consulte [Integrar AWS IoT TwinMaker y AWS IoT SiteWise](#) para obtener más información.

Algunas preguntas complementarias para hacerle al Asistente después de ver un resumen de la alarma en el panel de control, como parte de la misma conversación.

- ¿Mostrar los detalles del activo del resumen anterior?
- ¿Cuál es la ruta jerárquica desde la raíz hasta el activo mencionado?
- ¿Cuáles son los activos descendientes dependientes del activo mencionado?
- ¿Cuáles son los activos dependientes del activo mencionado que tienen alarmas activas?
- Encuentre todos los activos que tienen alarmas activas.

Algunas son preguntas de seguimiento para hacerle al Asistente después de ver un resumen de la propiedad en el panel de control, como parte de la misma conversación.

- Realice el mismo análisis durante las últimas 24 horas.
- Encuentre la documentación relacionada con las propiedades mencionadas anteriormente.
- Proporcione los detalles del identificador del activo 1da67d28-14f8-4f71-a06a-386f0425a21d/
nombre del activo Demo Turbine Asset 1.

AWS IoT SiteWise Invoca al Asistente desde la API.

- Genera un resumen de la alarma para el nombre de la alarma windSpeedAlarmen el identificador del activo. d591e153-e5cf-4206-96bb-ce3c119d9d2d
- Genera un resumen de las alarmas de las últimas 12 horas/2 días/1 semana para incluir el nombre windSpeedAlarmde la alarma en el identificador del activo. d591e153-e5cf-4206-96bb-ce3c119d9d2d

- Genere un resumen de la propiedad para la identificación de la propiedad en la identificación del activo ab187fb7-d74b-44d9-bd9b-f2f19a9137cc d591e153-e5cf-4206-96bb-ce3c119d9d2d
- Genera un resumen de la propiedad de las últimas 12 horas/2 días/1 semana para incluir el identificador de la propiedad en el identificador del ab187fb7-d74b-44d9-bd9b-f2f19a9137cc activo. d591e153-e5cf-4206-96bb-ce3c119d9d2d
- Busque los activos con el nombre Turbine.
- Indíqueme los valores de propiedad actuales de la identificación de la propiedad 5356168c-3390-456f-802c-9f6e047810d4 en la identificación del activo d591e153-e5cf-4206-96bb-ce3c119d9d2d,3cbb084e-1ded-4b08-9f21-1b47b2fb86fd.
- ¿Cuál es la relación entre el identificador del activo d591e153-e5cf-4206-96bb-ce3c119d9d2d y el identificador del activo 3cbb084e-1ded-4b08-9f21-1b47b2fb86fd?
- Encuentre documentación sobre cómo solucionar el problema de las bajas RPM de los aerogeneradores.
- Genere un resumen de la propiedad para el alias de la propiedad WindSpeed.
- ¿Cuáles son las comprobaciones previas a la operación según mi base de conocimientos?

Consulta datos de AWS IoT SiteWise

Puede utilizar las operaciones de la AWS IoT SiteWise API para consultar los valores actuales, históricos y agregados de sus propiedades de activos en intervalos de tiempo específicos. AWS IoT SiteWise proporciona múltiples interfaces de consulta para satisfacer diferentes necesidades de integración:

- Operaciones de API directas: llamadas de API sencillas y específicas para necesidades específicas de recuperación de datos
- Lenguaje de consultas similar a SQL: consultas potentes y flexibles para análisis de datos complejos
- Controlador ODBC: integración con herramientas y aplicaciones de inteligencia empresarial

Utilice estas funciones de consulta para:

- Obtenga información en tiempo real sobre los datos operativos
- Analice las tendencias y los patrones históricos
- Calcule las métricas de rendimiento de sus activos industriales
- Integre los datos de IoT con sistemas y paneles empresariales
- Cree aplicaciones personalizadas que aprovechen los datos industriales

Por ejemplo, puede descubrir todos los activos con valores de propiedad específicos, crear representaciones personalizadas de sus datos o desarrollar soluciones de software que se integren con los datos industriales almacenados en sus AWS IoT SiteWise activos. También puede explorar los datos de sus activos en tiempo real en AWS IoT SiteWise Monitor. Para obtener información sobre cómo configurar SiteWise Monitor, consulte [Supervise los datos con AWS IoT SiteWise Monitor](#).

Las operaciones descritas en esta sección devuelven objetos de valor de propiedad que contienen estructuras de marca de tiempo, calidad y valor (TQV):

- El `timestamp` contiene el tiempo actual en formato de tiempo Unix en segundos con desplazamiento en nanosegundos.
- `quality` contiene una de las siguientes cadenas, que indican la calidad del punto de datos:
 - `GOOD`: los datos no se ven afectados por ningún problema.

- BAD: los datos se ven afectados por un problema, como un fallo del sensor.
- UNCERTAIN: los datos se ven afectados por un problema, como la falta de precisión de un sensor.
- El `value` contiene uno de los siguientes campos, en función del tipo de propiedad:
 - `booleanValue`
 - `doubleValue`
 - `integerValue`
 - `stringValue`
 - `nullValue`

Temas

- [Consulte los valores actuales de las propiedades de los activos en AWS IoT SiteWise](#)
- [Consulte los valores históricos de las propiedades de los activos en AWS IoT SiteWise](#)
- [La propiedad del activo de consulta se agrega en AWS IoT SiteWise](#)
- [AWS IoT SiteWise idioma de consulta](#)
- [Optimización de las consultas](#)
- [ODBC](#)

Consulte los valores actuales de las propiedades de los activos en AWS IoT SiteWise

En este tutorial se muestran dos formas de obtener el valor actual de la propiedad de un activo. Puede usar la AWS IoT SiteWise consola o la API en AWS Command Line Interface (AWS CLI).

Temas

- [Consulta del valor actual de la propiedad de un activo \(consola\)](#)
- [Consulta del valor actual de la propiedad de un activo \(AWS CLI\)](#)

Consulta del valor actual de la propiedad de un activo (consola)

Puede usar la AWS IoT SiteWise consola para ver el valor actual de una propiedad de activo.

Para obtener el valor actual de la propiedad de un activo (consola)

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, elija activos.
3. Elija el activo con la propiedad que desea consultar.
4. Puede elegir el icono de flecha para expandir una jerarquía de activos y buscar su activo.
5. Elija la pestaña para el tipo de propiedad. Por ejemplo, elija Medidas para consultar el valor actual de una propiedad de medida.
6. Encuentre la propiedad que desea consultar. El valor actual aparece en la columna Valor más reciente.

Consulta del valor actual de la propiedad de un activo (AWS CLI)

Puede usar el AWS Command Line Interface (AWS CLI) para consultar el valor actual de una propiedad de activo.

Utilice la [GetAssetPropertyValue](#) operación para consultar el valor actual de una propiedad de un activo.

Para identificar la propiedad de un activo, especifique una de las siguientes opciones:

- `assetId` y `propertyId` de la propiedad del activo a la que está enviando datos.
- El `propertyAlias`, que es un alias de flujo de datos (por ejemplo, `/company/windfarm/3/turbine/7/temperature`). Para utilizar esta opción, primero debe establecer el alias de la propiedad del activo. Para establecer alias de propiedades, consulte [Gestione los flujos de datos para AWS IoT SiteWise](#).

Obtención del valor actual de la propiedad de un activo (AWS CLI)

- Ejecute el siguiente comando para obtener el valor actual de la propiedad del activo. `asset-id` Sustitúyalo por el identificador del activo y `property-id` por el identificador de la propiedad.

```
aws iotsitewise get-asset-property-value \  
  --asset-id asset-id \  
  --property-id property-id
```

La operación devuelve una respuesta que contiene el TQV actual de la propiedad en el siguiente formato.

```
{
  "propertyValue": {
    "value": {
      "booleanValue": Boolean,
      "doubleValue": Number,
      "integerValue": Number,
      "stringValue": "String",
      "nullValue": {
        "valueType": "String"
      }
    }
  },
  "timestamp": {
    "timeInSeconds": Number,
    "offsetInNanos": Number
  },
  "quality": "String"
}
```

Consulte los valores históricos de las propiedades de los activos en AWS IoT SiteWise

Puede utilizar la [GetAssetPropertyValueHistory](#) operación de la AWS IoT SiteWise API para consultar los valores históricos de una propiedad de un activo.

Para identificar la propiedad de un activo, especifique una de las siguientes opciones:

- `assetId` y `propertyId` de la propiedad del activo a la que está enviando datos.
- El `propertyAlias`, que es un alias de flujo de datos (por ejemplo, `/company/windfarm/3/turbine/7/temperature`). Para utilizar esta opción, primero debe establecer el alias de la propiedad del activo. Para establecer alias de propiedades, consulte [Gestione los flujos de datos para AWS IoT SiteWise](#).

Pase los siguientes parámetros para refinar los resultados:

- `startDate`: el inicio inclusivo del rango del cual se consultan los datos históricos, expresado en segundos en tiempo epoch de Unix.
- `endDate`: el final inclusivo del rango del cual se consultan los datos históricos, expresado en segundos en tiempo epoch de Unix.
- `maxResults`: el número máximo de resultados por devolver en una petición. Predeterminado a 20 resultados.
- `nextToken`: un token de paginación devuelto por una llamada anterior de esta operación.
- `timeOrdering`: el orden por aplicar a los valores devueltos: ASCENDING o DESCENDING.
- `qualities`: calidad para filtrar los resultados: GOOD, BAD, o UNCERTAIN.

Para consultar el historial de valores de la propiedad de un activo (AWS CLI)

1. Ejecute el siguiente comando para obtener el historial de valores de la propiedad del activo. Este comando consulta el historial de la propiedad durante un intervalo específico de 10 minutos. `asset-id` Sustitúyalo por el ID del activo y `property-id` por el ID de la propiedad. Reemplace los parámetros de fecha por el intervalo que desea consultar.

```
aws iotsitewise get-asset-property-value-history \  
  --asset-id asset-id \  
  --property-id property-id \  
  --start-date 1575216000 \  
  --end-date 1575216600
```

La operación devuelve una respuesta que contiene el historial TQVs de la propiedad en el siguiente formato:

```
{  
  "assetPropertyValueHistory": [  
    {  
      "value": {  
        "booleanValue": Boolean,  
        "doubleValue": Number,  
        "integerValue": Number,  
        "stringValue": "String",  
        "nullValue": {  
          "valueType": "String"  
        }  
      }  
    },  
  ],  
}
```

```
    "timestamp": {
      "timeInSeconds": Number,
      "offsetInNanos": Number
    },
    "quality": "String"
  }
],
"nextToken": "String"
}
```

2. Si existen más entradas de valores, puede pasar el token de paginación del `nextToken` campo a una llamada posterior a la [GetAssetPropertyValueHistory](#) operación.

La propiedad del activo de consulta se agrega en AWS IoT SiteWise

AWS IoT SiteWise calcula automáticamente los valores agregados de las propiedades de los activos, que son un conjunto de métricas básicas calculadas en varios intervalos de tiempo. AWS IoT SiteWise calcula los siguientes agregados cada minuto, hora y día para las propiedades de sus activos:

- promedio: el promedio (media) de los valores de una propiedad en un intervalo de tiempo.
- recuento: el número de puntos de datos de una propiedad a lo largo de un intervalo de tiempo.
- máximo: el máximo de los valores de una propiedad en un intervalo de tiempo.
- mínimo: el mínimo de los valores de una propiedad en un intervalo de tiempo.
- desviación estándar: la desviación estándar de los valores de una propiedad en un intervalo de tiempo.
- suma: la suma de los valores de una propiedad en un intervalo de tiempo.

Para las propiedades no numéricas, como cadenas y valores booleanos, AWS IoT SiteWise calcula solo el recuento agregado.

También puede calcular métricas personalizadas para los datos de activos. Con las propiedades de métricas puede definir agregaciones específicas para la operación. Las propiedades métricas ofrecen funciones de agregación e intervalos de tiempo adicionales que no están precalculados para la API. AWS IoT SiteWise Para obtener más información, consulte [Agregación de datos desde propiedades y otros activos \(métricas\)](#).

Temas

- [Agregaciones para la propiedad de un activo \(API\)](#)
- [Agregaciones para la propiedad de un activo \(AWS CLI\)](#)

Agregaciones para la propiedad de un activo (API)

Usa la AWS IoT SiteWise API para obtener los agregados de una propiedad de activo.

Utilice la [GetAssetPropertyAggregates](#) operación para consultar los agregados de una propiedad de activo.

Para identificar la propiedad de un activo, especifique una de las siguientes opciones:

- `assetId` y `propertyId` de la propiedad del activo a la que está enviando datos.
- El `propertyAlias`, que es un alias de flujo de datos (por ejemplo, `/company/windfarm/3/turbine/7/temperature`). Para utilizar esta opción, primero debe establecer el alias de la propiedad del activo. Para establecer alias de propiedades, consulte [Gestione los flujos de datos para AWS IoT SiteWise](#).

Debe superar los siguientes parámetros obligatorios:

- `aggregateTypes`: la lista de agregados que se va a recuperar. Puede especificar uno de estos: AVERAGE, COUNT, MAXIMUM, MINIMUM, STANDARD_DEVIATION y SUM.
- `resolution`— El intervalo de tiempo durante el que se va a recuperar la métrica: 1m (1 minuto), 15m (15 minutos), 1h (1 hora) o 1d (1 día).
- `startDate`: el inicio inclusivo del rango del cual se consultan los datos históricos, expresado en segundos en tiempo epoch de Unix.
- `endDate`: el final inclusivo del rango del cual se consultan los datos históricos, expresado en segundos en tiempo epoch de Unix.

También puede pasar cualquiera de los siguientes parámetros para refinar los resultados:

- `maxResults`: el número máximo de resultados por devolver en una petición. Predeterminado a 20 resultados.
- `nextToken`: un token de paginación devuelto por una llamada anterior de esta operación.

- `timeOrdering`: el orden por aplicar a los valores devueltos: `ASCENDING` o `DESCENDING`.
- `qualities`: calidad para filtrar los resultados: `GOOD`, `BAD`, o `UNCERTAIN`.

Note

La [GetAssetPropertyAggregates](#) operación devuelve un TQV con un formato diferente al de otras operaciones descritas en esta sección. La estructura del `value` contiene un campo para cada uno de los `aggregateTypes` de la solicitud. En `timestamp` se incluye el tiempo en que se produjo la agregación, en segundos en formato Unix.

Agregaciones para la propiedad de un activo (AWS CLI)

Consulta de agregados para la propiedad de un activo (AWS CLI)

1. Ejecute el siguiente comando para obtener agregados para la propiedad del activo. Este comando consulta la media y la suma con una resolución de 1 hora para un intervalo de 1 hora específico. `asset-id` Sustitúyalo por el identificador del activo y `property-id` por el identificador de la propiedad. Reemplace los parámetros por los agregados y el intervalo a consultar.

```
aws iotsitewise get-asset-property-aggregates \
  --asset-id asset-id \
  --property-id property-id \
  --start-date 1575216000 \
  --end-date 1575219600 \
  --aggregate-types AVERAGE SUM \
  --resolution 1h
```

La operación devuelve una respuesta que contiene el historial TQVs de la propiedad en el siguiente formato. La respuesta incluye solo los agregados solicitados.

```
{
  "aggregatedValues": [
    {
      "timestamp": Number,
      "quality": "String",
      "value": {
        "average": Number,
```

```
    "count": Number,
    "maximum": Number,
    "minimum": Number,
    "standardDeviation": Number,
    "sum": Number
  }
}
],
"nextToken": "String"
}
```

2. Si existen más entradas de valores, puede pasar el token de paginación del `nextToken` campo a una llamada posterior a la [GetAssetPropertyAggregates](#) operación.

Note

Si el rango de consultas contiene un `null` valor TQVs, consulta la [AssetPropertyValueAPI](#). Todas las estadísticas, excepto el recuento, dan como resultado una `null` respuesta, similar a las estadísticas de `String` TQVs. Si el rango de consulta contiene `Double.NaN` texto doble TQVs, todos los cálculos, excepto el recuento, darán como resultado un `Double.NaN`.

AWS IoT SiteWise idioma de consulta

Con la operación de la [ExecuteQuery](#) API de recuperación de AWS IoT SiteWise datos, puede recuperar información sobre las definiciones estructurales declarativas y los datos de series temporales asociadas a ellas a partir de lo siguiente:

- Modelos de
- recursos
- mediciones
- métricas
- transformaciones
- agregaciones

Esto se puede hacer con instrucciones de consulta tipo SQL, en una sola solicitud de API.

Note

Esta función está disponible en todas las regiones en las que AWS IoT SiteWise está disponible, excepto AWS GovCloud (EE. UU. oeste), Canadá (centro), China (Beijing) y EE. UU. Este (Ohio).

Temas

- [Consulte la referencia de idioma para AWS IoT SiteWise](#)

Consulte la referencia de idioma para AWS IoT SiteWise

AWS IoT SiteWise admite un lenguaje de consulta enriquecido para trabajar con los datos. En los siguientes temas se describen los tipos de datos, operadores, funciones y constructos disponibles.

Consulte [Consultas de ejemplo](#) para escribir consultas con el lenguaje de AWS IoT SiteWise consultas.

Temas

- [Consulta vistas de referencia](#)
- [Tipos de datos compatibles](#)
- [Cláusulas admitidas](#)
- [Logical operators \(Operadores lógicos\)](#)
- [Operadores de comparación](#)
- [Funciones SQL](#)
- [Consultas de ejemplo](#)

Consulta vistas de referencia

En esta sección se proporciona información que le ayudará a entender las vistas AWS IoT SiteWise, como los metadatos del proceso y los datos de telemetría.

En las tablas siguientes se proporcionan los nombres y las descripciones de las vistas:

Modelo de datos

Nombre de la vista	Descripción de vista
asset	Contiene información sobre la derivación del activo y el modelo.
asset_property	Contiene información sobre la estructura de la propiedad del activo.
raw_time_series	Contiene los datos históricos de la serie temporal.
latest_value_time_series	Contiene el valor más reciente de la serie temporal.
precomputed_aggregates	Contiene los valores agregados de la propiedad del activo calculados automáticamente. Son un conjunto de métricas básicas calculadas a lo largo de varios intervalos de tiempo.

Las siguientes vistas muestran los nombres de las columnas y los tipos de datos de cada vista.

View:asset

nombre de columna	datatype (tipo_de_datos)
asset_id	cadena
asset_name	cadena
asset_description	cadena
asset_model_id	cadena
parent_asset_id	cadena
asset_external_id	cadena

nombre de columna	datatype (tipo_de_datos)
asset_model_external_id	cadena
hierarchy_id	cadena

View:asset_property

nombre de columna	datatype (tipo_de_datos)			
asset_id	cadena			
property_id	cadena			
property_name	cadena			
property_alias	cadena			
property_external_id	cadena			
asset_composite_model_id	cadena			
tipo_propiedad	cadena			
property_data_type	cadena			
int_attribute_value	entero			
valor_attribute_doble	double			
valor_attribute_booleano	booleano			

nombre de columna	datatype (tipo_de_datos)			
valor_atributo_cadena	cadena			

View:raw_time_series

nombre de columna	datatype (tipo_de_datos)
asset_id	cadena
property_id	cadena
property_alias	cadena
event_timestamp	timestamp
quality	cadena
boolean_value	booleano
int_value	entero
double_value	double
string_value	cadena

View:latest_value_time_series

nombre de columna	datatype (tipo_de_datos)
asset_id	cadena
property_id	cadena
property_alias	cadena
event_timestamp	timestamp

nombre de columna	datatype (tipo_de_datos)
quality	cadena
boolean_value	booleano
int_value	entero
double_value	double
string_value	cadena

View:precomputed_aggregates

nombre de columna	datatype (tipo_de_datos)
asset_id	cadena
property_id	cadena
property_alias	cadena
event_timestamp	timestamp
quality	cadena
resolution	cadena
sum_value	double
count_value	entero
average_value	double
maximum_value	double
minimum_value	double
stdev_value	double

Tipos de datos compatibles

AWS IoT SiteWise el lenguaje de consulta admite los siguientes tipos de datos.

Valor escalar

Tipo de datos:	Descripción
STRING	Una cadena con una longitud máxima de 1024 bytes.
INTEGER	Un entero de 32 bits firmado con un intervalo de $-2,147,483,648$ to $2,147,483,647$.
DOUBLE	Un número de coma flotante con un rango -10^{100} to 10^{100} de o Nan con IEEE 754 doble precisión.
BOOLEAN	true o bien false.
TIMESTAMP	Marcas de tiempo que cumplen con la norma ISO-8601: <ul style="list-style-type: none"> 'yyyy-MM-dd HH:mm:ss[.SSS]' TIMESTAMP 'yyyy-MM-dd[\s T]HH:mm:ss[.SSS][+HH:mm 'Z']' 'yyyy-MM-dd'T'HH:mm:ss[.SSS][+HH:mm 'Z']' 'yyyy-MM-dd'T'HH:mm:ss+[hh:mm]'

Note

Null: Un valor booleano que true indica la falta de datos definidos.

Example

TIMESTAMPejemplos de valores:

```
TIMESTAMP '2025-12-21 23:59:59.999Z'  
TIMESTAMP '2025-12-21 23:59:59+23:59'  
'2025-12-21 23:59:59'  
'2025-12-21T23:59:59.123+11:11'
```

Note

Los datos de precisión doble no son exactos. Algunos valores no se convierten exactamente y no representan todos los números reales debido a la precisión limitada. Es posible que los datos de coma flotante de la consulta no tengan el mismo valor representado internamente. El valor se redondea si la precisión de un número de entrada es demasiado alta.

Cláusulas admitidas

La SELECT sentencia se utiliza para recuperar datos de una o más vistas. AWS IoT SiteWise admite las INNER JOIN operaciones JOIN y.

Las vistas se unen con una JOIN sintaxis explícita o con anotaciones separadas por comas en la cláusula. FROM

Example

Una declaración general: SELECT

```
SELECT expression [, ...]  
  [ FROM table_name AS alias [, ...] ]  
  [ WHERE condition ]  
  [ GROUP BY expression [, ...] ]  
  [ HAVING condition ]  
  [ ORDER BY expression [ ASC | DESC ] [ NULLS FIRST | NULLS LAST ] [, ...] ]  
  [ LIMIT expression ]
```

Example

Una declaración SELECT con las diferentes cláusulas:

```
SELECT  
  a.asset_name,  
  a.asset_id,
```

```

p.property_type,
p.property_data_type,
p.string_attribute_value,
p.property_name
FROM asset a, asset_property p
WHERE a.asset_description LIKE '%description%'
AND p.property_type IN ('attribute', 'metric')
OR p.property_id IN (
  SELECT property_id
  FROM raw_time_series
  WHERE event_timestamp BETWEEN TIMESTAMP '2025-01-01 00:00:00' AND TIMESTAMP
'2025-01-02 00:00:00'
  GROUP BY asset_id, property_id
  HAVING COUNT(*) > 100
)
GROUP BY p.property_type
HAVING COUNT(*) > 5
ORDER BY a.asset_name ASC
LIMIT 20;

```

Note

Una implícita JOIN combina dos o más tablas diferentes sin usar la JOIN palabra clave según AWS IoT SiteWise su esquema interno. Esto equivale a realizar una JOIN en los property_id campos asset_id y entre las tablas de metadatos y de datos sin procesar. Este patrón SiteWise permite aprovechar cualquier filtro de metadatos de la consulta cuando se extraen datos de tablas de datos sin procesar de forma que se escaneen menos datos generales.

Example de una consulta:

```

SELECT a.asset_name, p.property_name, r.event_timestamp
FROM asset a, asset_property p, raw_time_series r
WHERE a.asset_name='my_asset' AND p.property_name='my_property'

```

El ejemplo anterior solo escanea los datos de la propiedad del activo que pertenece a los nombres de metadatos especificados.

Example de un equivalente menos optimizado de la consulta anterior:

```

SELECT a.asset_name, p.property_name, r.event_timestamp

```

```
FROM asset a
JOIN asset_property p ON a.asset_id=p.asset_id
JOIN raw_time_series r ON p.asset_id=r.asset_id AND p.property_id=r.property_id
WHERE a.asset_name='my_asset' AND p.property_name='my_property'
```

A continuación se detalla una explicación de cada cláusula y su descripción:

cláusula	Signature	Descripción
LIMIT	<pre>LIMIT { count }</pre>	<p>Esta cláusula limita el conjunto de resultados al número de filas especificado. Puede utilizarla LIMIT con o sin OFFSET cláusulas ORDER BY y.</p> <p>LIMIT solo funciona con números enteros no negativos de [0,2147483647].</p>
ORDER BY	<pre>ORDER BY expression [ASC DESC] [NULLS FIRST NULLS LAST]</pre>	<p>La ORDER BY cláusula ordena el conjunto de resultados de una consulta.</p> <div data-bbox="1088 1318 1219 1356" data-label="Section-Header"> <p>Note</p> </div> <p>Al hacer referenci a a las columnas seleccionadas en una agregación de la ORDER BY cláusula, utilice el índice ordinal de la columna en lugar del nombre o el alias.</p> <pre>SELECT AVG(t.dou ble_value)</pre>

cláusula	Signature	Descripción
		<pre>FROM latest_value_time_series t GROUP BY t.asset_id ORDER BY 1</pre>
GROUP BY	<pre>GROUP BY expression [, ...]</pre>	La GROUP BY cláusula identifica las columnas de agrupación de la consulta. Se usa junto con una expresión agregada.
HAVING	<pre>HAVING boolean-expression</pre>	La HAVING cláusula filtra las filas del grupo creadas por la cláusula GROUP BY.
SUB SELECT	<pre>SELECT column1, column2 FROM table1 WHERE column3 IN (SELECT column4 FROM table2);</pre>	Una SELECT declaración incrustada dentro de otra SELECT declaración.
JOIN	<pre>SELECT column1, column2 FROM table1 JOIN table2 ON table1.column1 = table2.column1;</pre>	
INNER JOIN	<pre>SELECT columns FROM table1 INNER JOIN table2 ON table1.column = table2.column;</pre>	An INNER JOIN devuelve todas las filas de ambas tablas que coincidan con la condición de unión.

cláusula	Signature	Descripción
UNION	<pre>query { UNION [ALL] } another_query</pre>	El UNION operador calcula la unión de conjuntos de sus dos argumentos y elimina automáticamente los registros duplicados del conjunto de resultados.

Logical operators (Operadores lógicos)

AWS IoT SiteWise admite los siguientes operadores lógicos.

Operador	Signature	Descripción
AND	a AND b	TRUE si ambos valores son verdaderos
OR	a OR b	TRUE si un valor es verdadero
NOT	expresión NOT	TRUE si una expresión es falsa y FALSE si una expresión es verdadera
IN	INexpresión x	TRUE si el valor está en la expresión
BETWEEN	BETWEENa AND b	TRUE si el valor está entre el límite superior e inferior, e incluye ambos límites
LIKE	Patrón LIKE	<p>TRUE si el valor sigue un patrón</p> <p>LIKE admite caracteres comodín. A continuación se muestran algunos ejemplos:</p>

Operador	Signature	Descripción
		<ul style="list-style-type: none"> • %sustituye uno o más caracteres en una cadena. • _sustituye un carácter de una cadena. • ESCAPEse usa con un carácter para designar un carácter de escape en el patrón LIKE.

Ejemplos de todos los operadores lógicos:

Función	Ejemplo
AND	<pre>SELECT a.asset_name FROM asset AS a, latest_value_time_ series AS t WHERE t.int_value > 30 AND t.event_timestamp > TIMESTAMP '2025-05-15 00:00:01'</pre>
OR	<pre>SELECT a.asset_name FROM asset AS a WHERE a.asset_name like 'abc' OR a.asset_name like 'pqr'</pre>
NOT	<pre>SELECT ma.asset_id AS a_id FROM asset AS ma WHERE (ma.asset_id NOT LIKE 'some %patterna%' escape 'a') AND ma.asset_ id='abc'</pre>
IN	<pre>SELECT a.asset_name FROM asset AS a</pre>

Función	Ejemplo
	<pre>WHERE a.asset_name IN ('abc', 'pqr')</pre>
BETWEEN	<pre>SELECT asset_id, int_value, event_timestamp AS i_v FROM raw_time_series WHERE event_timestamp BETWEEN TIMESTAMP '2025-04-15 00:00:01' and TIMESTAMP '2025-05-15 00:00:01'</pre>
LIKE	<ul style="list-style-type: none"> • %patrón: <pre>SELECT POWER(rw.int_value, 5) AS raised_value FROM raw_time_series AS rw WHERE rw.asset_id LIKE 'some%pattern%' AND rw.int_value > 30</pre> • _patrón: <pre>SELECT asset_id, property_id FROM asset_property WHERE string_attribute_value LIKE 'Floor_'</pre> • ESCAPEpatrón: <pre>SELECT asset_id FROM asset WHERE asset_name LIKE 'MyAsset/_%' ESCAPE '/'</pre>

Operadores de comparación

AWS IoT SiteWise admite los siguientes operadores de comparación. Todas las operaciones de comparación están disponibles para los tipos de datos integrados y se evalúan con un valor booleano.

Logical operators (Operadores lógicos)

Operador	Descripción
<	Menor que
>	Mayor que
<=	Menor o igual que
>=	Mayor o igual que
=	Igual a
!=	Desigualdad

Tabla de verdad de operaciones de comparación para valores no numéricos

Tipo	Escriba >= x	Escriba <= x	Escriba > x	Escriba < x	Tipo = x	¡Escribe! = x
NULL	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE

Algunos predicados se comportan como operadores pero tienen una sintaxis especial. Consulte a continuación:

Predicados de comparación

Operador	Descripción
IS NULL	Comprueba si un valor es. NULL
IS NOT NULL	Comprueba si un valor no lo esNULL.

Operadores NaN

NaN, o «No es un número», es un valor especial en aritmética de punto flotante. Esta es una lista de comparaciones y cómo funcionan. NaN

- NaN los valores deben estar entre comillas simples. Por ejemplo, 'NaN'.
- NaN los valores se consideran iguales entre sí.
- NaN es mayor que otros valores numéricos.
- En funciones agregadas como AVG(), STDDEV(), y SUM(), si hay algún valor NaN, el resultado es NaN.
- En las funciones agregadas como MAX() y MIN(), NaN los valores se incluyen en los cálculos.

Comparaciones de valores de NaN

Comparación	Resultado
'NaN' ≥ x	True
'NaN' ≤ x	Verdadero si x es igual a NaN, falso en caso contrario
'NaN' > x	Falso si x es igual a NaN, verdadero en caso contrario
'NaN' < x	False
'NaN' = x	Verdadero si x es igual a NaN, falso en caso contrario
'NaN' != x	Falso si x es igual a NaN, verdadero en caso contrario

Funciones SQL

Los grupos de funciones compatibles son:

Temas

- [Funciones escalares](#)

- [Funciones de agregación](#)

Funciones escalares

Las funciones escalares toman uno o más valores de entrada y devuelven un único valor de salida. Se utilizan ampliamente en SQL (lenguaje de consulta estructurado) para la manipulación y recuperación de datos, lo que mejora la eficiencia de las tareas de procesamiento de datos.

Temas

- [Funciones de datos nulos](#)
- [Funciones de cadena](#)
- [Funciones matemáticas](#)
- [Funciones de fecha y hora](#)
- [Funciones de conversión de tipos](#)

Funciones de datos nulos

Las funciones de datos nulos gestionan o manipulan valores NULOS, que representan la ausencia de un valor. Las funciones permiten sustituirlos por NULLs otros valores, comprobar si un valor es NULO o realizar operaciones que se NULLs gestionen de una forma específica.

Función	Signature	Descripción
COALESCE	COALESCE (expresión1, expresión2,..., expresiónN)	Si todas las expresiones se evalúan como nulas, COALESCE devuelve nulo. Las expresiones deben ser del mismo tipo.

Example de una función COALESCE

```
SELECT COALESCE (1.double_value, 100) AS non_double_value FROM latest_value_time_series AS 1 LIMIT 1
```

Funciones de cadena

Las funciones de cadena son herramientas integradas que se utilizan para manipular y procesar datos de texto. Permiten tareas como la concatenación, la extracción, el formateo y la búsqueda dentro de cadenas. Estas funciones son esenciales para limpiar, transformar y analizar los datos basados en texto de una base de datos.

Funciones de cadena

Función	Signature	Descripción
LENGTH	LONGITUD (CADENA)	Devuelve la longitud de la cadena.
CONCAT	CONCAT (cadena, cadena)	Concatena los argumentos de una cadena.
SUBSTR	<ul style="list-style-type: none"> • SUBSTR (cadena, inicio) • SUBSTR (cadena, inicio, longitud) • SUBSTR (cadena, expresión regular) 	<p>Devuelve uno de los siguientes valores:</p> <ul style="list-style-type: none"> • Devuelve la subcadena de la cadena de entrada que comienza en la ubicación especificada y, opcionalmente, tiene la longitud especificada. • Devuelve la primera subcadena de la cadena de entrada que coincide con la expresión regular especificada. <p>Utiliza la indexación basada en 1 para el parámetro de inicio.</p>

Función	Signature	Descripción
UPPER	UPPER (cadena)	Convierte los caracteres de la cadena de entrada a mayúsculas.
LOWER	INFERIOR (cadena)	Convierte los caracteres de la cadena de entrada a minúsculas.
TRIM	TRIM (cadena)	Elimina los caracteres de espacio del principio, del final o de ambos lados de la cadena.
LTRIM	LTRIM (cadena)	Elimina los caracteres de espacio del principio de la cadena.
RTRIM	RTRIM (cadena)	Elimina los caracteres de espacio del final de la cadena.
STR_REPLACE	STR_REPLACE (cadena, desde, hasta)	Sustituye todas las apariciones de la subcadena especificada por otra subcadena especificada.

Ejemplos de todas las funciones:

Función	Ejemplo
LENGTH	<pre>SELECT LENGTH(a.asset_id) AS asset_id_length FROM asset AS a</pre>
CONCAT	<pre>SELECT CONCAT(p.property_id, p.property_name) FROM asset_pro perty AS p</pre>

Función	Ejemplo
SUBSTR	<ul style="list-style-type: none"> • <code>SELECT SUBSTR(a.asset_name, 1, 3) AS substr_val FROM asset AS a</code> • <code>SELECT SUBSTR(p.property_name, 3) AS substr_val1 FROM asset_property AS p</code> • <code>SELECT SUBSTR(p.property_name, '@[^\.]*') AS substr_val2 FROM asset_property AS p</code>
UPPER	<code>SELECT UPPER(d.string_value) AS up_string FROM raw_time_series AS d</code>
LOWER	<code>SELECT LOWER(d.string_value) AS low_string FROM raw_time_series AS d</code>
TRIM	<code>SELECT TRIM(d.string_value) AS tm_string FROM raw_time_series AS d</code>
LTRIM	<code>SELECT LTRIM(d.string_value) AS ltrim_string FROM raw_time_series AS d</code>
RTRIM	<code>SELECT RTRIM(d.string_value) AS rtrim_string FROM raw_time_series AS d</code>
STR_REPLACE	<code>SELECT STR_REPLACE(d.string_value, 'abc', 'def') AS replaced_string FROM raw_time_series AS d</code>

Operador de concatenación

El operador de concatenación `||`, u operador de tubería, une dos cadenas. Proporciona una alternativa a la `CONCAT` función y es más legible cuando se combinan varias cadenas.

Example del operador de concatenación

```
SELECT a.asset_name || ' - ' || p.property_name
       AS full_name
FROM asset a, asset_property p
```

Funciones matemáticas

Las funciones matemáticas son operaciones matemáticas predefinidas que se utilizan en las consultas SQL para realizar cálculos con datos numéricos. Proporcionan formas de manipular y transformar los datos sin necesidad de extraerlos de la base de datos y procesarlos por separado.

Funciones matemáticas

Función	Signature	Descripción
POWER	POTENCIA (int double, int double)	Devuelve el valor del primer argumento elevado a la potencia del segundo argumento.
ROUND	<ul style="list-style-type: none"> REDONDEAR (int double, decimal_places_int) REDONDO (int double) 	Redondea al entero más cercano.
FLOOR	PISO (int double)	Devuelve el entero más grande que no sea mayor que el valor dado.

Ejemplos de todas las funciones:

Función	Ejemplo
POWER	<ul style="list-style-type: none"> POWER (3, 77)

Función	Ejemplo
	<ul style="list-style-type: none"> POWER (2.3, 3.9) POWER (1.0, 4.2)
ROUND	ROUND (32.12435, 3)
FLOOR	FLOOR (21.2)

Funciones de fecha y hora

Las funciones de fecha y hora funcionan con fechas y horas. Estas funciones permiten extraer componentes específicos de una fecha, realizar cálculos y manipular los valores de la fecha.

Los identificadores permitidos en estas funciones son:

- YEAR
- MONTH
- DAY
- HOUR
- MINUTE
- SECOND

Función	Signature	Descripción
NOW	AHORA ()	Devuelve la marca de tiempo actual con una precisión de milisegundos. Proporciona la hora exacta en el que se ejecuta en una consulta.
DATE_ADD	DATE_ADD (identificador, intervalo de duración, columna)	Devuelve la suma de un intervalo date/time y un número. days/hours, or of a date/time and date/time

Función	Signature	Descripción
DATE_SUB	DATE_SUB (identificador, duración del intervalo, columna)	Devuelve la diferencia entre un intervalo date/time y un número. days/hours, or between a date/time and date/time
TIMESTAMP_ADD	TIMESTAMP_ADD (identificador, duración del intervalo, columna)	Añade un intervalo de tiempo, en las unidades de tiempo dadas, a una expresión de fecha y hora.
TIMESTAMP_SUB	TIMESTAMP_SUB (identificador, intervalo de duración, columna)	Resta un intervalo de tiempo, en las unidades de tiempo dadas, de una expresión de fecha y hora.
CAST	CAST (expresión como patrón de formato de marca de tiempo)	Convierte una expresión de cadena en una marca de tiempo utilizando el patrón de formato especificado. Los patrones más comunes incluyen el formato 'yyyy-MM-dd HH:mm:ss' de fecha y hora estándar. Por ejemplo, SELECT CAST('2023-12-25 14:30:00' AS TIMESTAMP) AS converted_timestamp

Example de una consulta SQL mediante las funciones enumeradas:

```
SELECT r.asset_id, r.int_value,
       date_add(DAY, 7, r.event_timestamp) AS date_in_future,
       date_sub(YEAR, 2, r.event_timestamp) AS date_in_past,
       timestamp_add(DAY, 2, r.event_timestamp) AS timestamp_in_future,
       timestamp_sub(DAY, 2, r.event_timestamp) AS timestamp_in_past,
```

```
now() AS time_now
FROM raw_time_series AS r
```

Funciones de conversión de tipos

Las funciones de conversión de tipos se utilizan para cambiar el tipo de datos de un valor a otro. Son esenciales para garantizar la compatibilidad de los datos y realizar operaciones que requieren datos en un formato específico.

Función	Signature	Descripción
TO_DATE	<ul style="list-style-type: none"> TO_DATE (entero) TO_DATE (expresión, formato) 	<ul style="list-style-type: none"> Convierte un entero de milisegundos de época en un valor de fecha. Convierte una representación de cadena de caracteres en un valor de fecha.
TO_TIMESTAMP	<ul style="list-style-type: none"> TO_TIMESTAMP (doble) TO_TIMESTAMP (cadena, formato) 	<ul style="list-style-type: none"> Convierte un segundo entero de época en un tipo de datos de marca de tiempo. Convierte una representación en cadena de una fecha y hora en un tipo de datos de marca de tiempo.
TO_TIME	<ul style="list-style-type: none"> TO_TIME (int) TO_TIME (cadena, formato) 	<ul style="list-style-type: none"> Convierte un entero de milisegundos de época en un valor de tiempo. Convierte una representación de cadena de caracteres en un valor temporal.

Función	Signature	Descripción
CAST	CAST (<expression>AS<data type>)	<p>Convierte una entidad o expresión que se evalúa en un único valor de un tipo a otro.</p> <p>Los tipos de datos admitidos son:</p> <ul style="list-style-type: none"> • BOOLEANO • INTEGER • INT • TIMESTAMP • DATE • CHAR • CHARACTER • STRING

Example de una consulta SQL mediante las funciones enumeradas:

```
SELECT TO_TIMESTAMP (100) AS timestamp_value,
       TO_DATE(r.event_timestamp) AS date_value,
       TO_TIME(r.event_timestamp) AS time_value
FROM raw_time_series AS r
```

Funciones de agregación

Las funciones agregadas son operaciones de bases de datos que realizan cálculos en varias filas de datos para producir un único resultado resumido. Estas funciones analizan los conjuntos de datos para devolver valores calculados, como sumas, promedios, recuentos u otras medidas estadísticas.

Función	Signature	Descripción
AVG	AVG (expresión)	Devuelve el promedio de una expresión numérica.

Función	Signature	Descripción
COUNT	CONTAR (expresión)	Devuelve el número de filas que coinciden con los criterios dados.
MAX	MAX (expresión)	Devuelve el valor más alto de las expresiones seleccionadas.
MIN	MIN (expresión)	Devuelve el valor más pequeño de las expresiones seleccionadas.
SUM	SUM (expresión)	Devuelve la suma de una expresión numérica.
STDDEV	STDDEV (expresión)	Devuelve la desviación estándar de la muestra.
GROUP BY	GROUP BY expresión	Devuelve una fila creada por la agrupación de columnas.
HAVING	CON una expresión booleana	Devuelve las filas del grupo filtradas por cláusula. GROUP BY

Ejemplos de todas las funciones:

Función	Ejemplo
AVG	<pre>SELECT d.asset_id, d.property_id, AVG(d.int_value) FROM raw_time_ series AS d</pre>
COUNT	<pre>SELECT COUNT(d.int_value) FROM raw_time_series AS d</pre>

Función	Ejemplo
MAX	<pre>SELECT MAX(d.int_value) FROM raw_time_series AS d</pre>
MIN	<pre>SELECT MIN(d.int_value) FROM raw_time_series AS d</pre>
SUM	<pre>SELECT SUM(d.int_value) FROM raw_time_series AS d</pre>
STDDEV	<pre>SELECT STDDEV(d.int_value) FROM raw_time_series AS d</pre>
<ul style="list-style-type: none"> • GROUP BY • HAVING 	<pre>SELECT MAX(d.int_value) AS max_int_value, d.asset_id FROM raw_time_series AS d GROUP BY d.asset_id HAVING MAX(d.int_value) > 5</pre>

Consultas de ejemplo

Filtrado de metadatos

El siguiente ejemplo es para filtrar metadatos con una SELECT sentencia con el lenguaje de AWS IoT SiteWise consulta:

```
SELECT a.asset_name, p.property_name
FROM asset a, asset_property p
WHERE a.asset_name LIKE 'Windmill%'
```

Filtrado de valores

A continuación se muestra un ejemplo de filtrado de valores mediante una SELECT sentencia con el lenguaje de AWS IoT SiteWise consulta:

```
SELECT a.asset_name, r.int_value
FROM asset a, raw_time_series r
WHERE r.int_value > 30
AND r.event_timestamp > TIMESTAMP '2022-01-05 12:15:00'
AND r.event_timestamp < TIMESTAMP '2022-01-05 12:20:00'
```

Optimización de las consultas

Filtros de metadatos

Cuando consulte metadatos o datos sin procesar, utilice la `WHERE` cláusula para filtrar por campos de metadatos a fin de reducir la cantidad de datos escaneados. Utilice los siguientes operadores para limitar el escaneo de metadatos:

- Igual (=)
- Distinto de (!=)
- LIKE
- IN
- AND
- OR

Para las propiedades de los atributos, utilice los siguientes campos para filtrar los resultados. :

- `double_attribute_value`
- `int_attribute_value`
- `boolean_attribute_value`
- `string_attribute_value`

Estos campos ofrecen un mejor rendimiento que la tabla `latest_value_time_series` para las propiedades de los activos del tipo de atributo.

Note

Utilice literales en el lado derecho de los operadores para limitar adecuadamente el escaneo de datos. Por ejemplo, la siguiente consulta tiene un rendimiento peor que el uso de un literal de cadena estricto:

```
SELECT property_id FROM asset_property WHERE property_name = CONCAT('my',  
'property')
```

Example para filtros de metadatos:

```
SELECT p.property_name FROM asset_property p  
WHERE p.property_type = 'attribute' AND p.string_attribute_value LIKE 'my-property-%'
```

Filtros de datos sin procesar

Todas las tablas de datos sin procesar (`raw_time_series`, `latest_value_time_series`, `precomputed_aggregates`) tienen marcas de tiempo asociadas a sus filas. Además de los filtros de metadatos, utilice filtros de cláusulas en el campo para reducir la cantidad de datos escaneados. `WHERE event_timestamp` Utilice las siguientes operaciones para limitar el escaneo de datos sin procesar:

- Igual (=)
- Mayor que (>)
- Menor que (<)
- Mayor que o igual a (>=)
- Menor que o igual a (<=)
- BETWEEN
- AND

Ejemplos de filtros:

- Al consultar la tabla `precomputed_aggregates`, especifique siempre un filtro de calidad en la cláusula. `WHERE` Esto reduce la cantidad de datos que escanea la consulta, especialmente si está buscando datos. `BAD UNCERTAIN`

También recomendamos encarecidamente utilizar un filtro de resolución (1 m, 15 m, 1 h o 1 d) al consultar la tabla `precomputed_aggregates`. Si no especificas un filtro de resolución, se AWS IoT SiteWise escaneará de forma predeterminada la tabla completa en todas las resoluciones, lo que resulta ineficiente.

- Al consultar datos sin procesar, las funciones de marca de tiempo también se pueden usar en la WHERE cláusula para filtrar la cantidad de datos escaneados. Por ejemplo, la siguiente consulta solo escanea los datos de los últimos 30 minutos de la tabla raw_time_series:

```
SELECT r.event_timestamp, r.double_value
FROM raw_time_series r
WHERE r.event_timestamp > TIMESTAMP_SUB(MINUTE, 30, NOW())
```

Note

No es igual (!=) y, por lo general, OR los operadores no aplican filtros significativos al escaneo de datos sin procesar. Los filtros de los valores de los datos sin procesar (string_value, double_value, etc.) tampoco limitan el escaneo de datos sin procesar.

Únete a la optimización

AWS IoT SiteWise SQL admite la JOIN palabra clave para combinar dos tablas. Solo JOIN se admiten los filtros activos en un campo (mediante la ON palabra clave). Están prohibidas las uniones cartesianas completas.

AWS IoT SiteWise también admite la JOIN s implícita sin utilizar la palabra clave. JOIN Se permiten entre diferentes tablas de metadatos y entre una tabla de metadatos y una tabla sin procesar. Por ejemplo, esta consulta:

```
SELECT a.asset_name, p.property_name FROM asset a, asset_property p
```

Funciona mejor que esta consulta equivalente:

```
SELECT a.asset_name, p.property_name FROM asset a
JOIN asset_property p ON a.asset_id = p.asset_id
```

Se permiten las siguientes uniones implícitas (O está permitida, X está prohibida):

	asset	asset_pro perty	latest_va lue_time_ series	raw_time_ series	precomput ed_aggreg ates	subquery
asset	X	O	O	O	O	X
asset_pro perty	O	X	O	O	O	X
latest_va lue_time_ series	O	O	X	X	X	X
raw_time_ series	O	O	X	X	X	X
precomput ed_aggreg ates	O	O	X	X	X	X
subquery	X	X	X	X	X	X

Utilice JOIN s implícitos siempre que sea posible. Si debe usar la JOIN palabra clave, aplique filtros en las tablas editoriales individuales JOIN para minimizar los datos escaneados. Por ejemplo, en lugar de esta consulta:

```
SELECT level1.asset_id, level2.asset_id, level3.asset_id
FROM asset AS level1
JOIN asset AS level2 ON level2.parent_asset_id = level1.asset_id
JOIN asset AS level3 ON level3.parent_asset_id = level2.asset_id
WHERE level1.asset_name LIKE 'level1%'
AND level2.asset_name LIKE 'level2%'
AND level3.asset_name LIKE 'level3%'
```

Usa esta consulta más eficiente:

```
SELECT level1.asset_id, level2.asset_id, level3.asset_id
FROM asset AS level1
```

```
JOIN (SELECT asset_id, parent_asset_id FROM asset WHERE asset_name LIKE 'level2%') AS
level2 ON level2.parent_asset_id = level1.asset_id
JOIN (SELECT asset_id, parent_asset_id FROM asset WHERE asset_name LIKE 'level3%') AS
level3 ON level3.parent_asset_id = level2.asset_id
WHERE level1.asset_name LIKE 'level1%'
```

Al insertar filtros de metadatos en las subconsultas, se asegura de que las tablas individuales de la JOIN s se filtren durante el proceso de digitalización. También puedes usar la LIMIT palabra clave en las subconsultas para obtener el mismo efecto.

Consultas grandes

Para las consultas que producen más filas que las predeterminadas, establece el tamaño de página de la [ExecuteQuery](#) API en un valor máximo de 20 000. Esto mejora el rendimiento general de las consultas.

Utilice la LIMIT cláusula para reducir la cantidad de datos escaneados en algunas consultas. Tenga en cuenta que las funciones de agregado y ciertas cláusulas que abarcan toda la tabla (GROUP BYORDER BY,,JOIN) requieren un análisis completo antes de aplicar la LIMIT cláusula.

Note

AWS IoT SiteWise pueden escanear una cantidad mínima de datos incluso con la LIMIT cláusula aplicada, especialmente en el caso de consultas de datos sin procesar que escanean varias propiedades.

ODBC

El [controlador ODBC](#) de código abierto AWS IoT SiteWise proporciona una interfaz relacional SQL AWS IoT SiteWise para los desarrolladores y permite la conectividad desde herramientas de inteligencia empresarial (BI) como Power BI Desktop y Microsoft Excel. El controlador AWS IoT SiteWise ODBC está disponible actualmente en [Windows](#) y admite el inicio de sesión único con Okta y Microsoft Azure Active Directory (AD).

Para obtener más información, consulte la documentación del controlador [AWS IoT SiteWise ODBC](#) en. GitHub

Temas

- [Sintaxis de la cadena de conexión y opciones del controlador de ODBC](#)
- [Ejemplos de cadenas de conexión para el controlador AWS IoT SiteWise ODBC](#)
- [Solución de problemas de conexión con el controlador de ODBC](#)

Sintaxis de la cadena de conexión y opciones del controlador de ODBC

La sintaxis para especificar las opciones de cadena de conexión para el controlador de ODBC es la siguiente:

```
Driver={AWS IoT SiteWise ODBC Driver};(option)=(value);
```

Las opciones disponibles son las siguientes:

Opciones de conexión del controlador

- **Driver** (obligatorio): el controlador que se utiliza con ODBC.

El valor predeterminado es AWS IoT SiteWise.

- **DSN** – El nombre del origen de datos (DSN) que se utilizará para configurar la conexión.

El valor predeterminado es NONE.

- **Auth** – Modo de autenticación. Este debe ser uno de los siguientes:
 - **AWS_PROFILE**: utilice la cadena de credenciales predeterminada.
 - **IAM**— Utilice las credenciales AWS de IAM.
 - **AAD**: utilice el proveedor de identidades de Azure Active Directory (AD).
 - **OKTA**: utilice el proveedor de identidades Okta.

El valor predeterminado es AWS_PROFILE.

Opciones de configuración de puntos de conexión

- **EndpointOverride**— La anulación del punto final del AWS IoT SiteWise servicio. Se trata de una opción avanzada que anula la región. Por ejemplo:

```
iotsitewise.us-east-1.amazonaws.com
```

- **Region**— La región de firma del punto final del AWS IoT SiteWise servicio.

El valor predeterminado es `us-east-1`.

Opción de proveedores de credenciales

- **ProfileName**— El nombre del perfil en el archivo de AWS configuración.

El valor predeterminado es `NONE`.

AWS Opciones de autenticación de IAM

- **UIDo AccessKeyId**: el identificador de la clave AWS de acceso del usuario. Si `UID` y `AccessKeyId` se proporcionan en la cadena de conexión, se utilizará el valor `UID` a menos que esté vacío.

El valor predeterminado es `NONE`.

- **PWD o SecretKey** – La clave de acceso secreta del usuario de AWS. Si `PWD` y `SecretKey` se proporcionan en la cadena de conexión, se utilizará el valor `PWD` a menos que esté vacío.

El valor predeterminado es `NONE`.

- **SessionToken** – El token de sesión temporal necesario para acceder a una base de datos con la autenticación multifactor (MFA) habilitada. No incluya un `=` final en la entrada.

El valor predeterminado es `NONE`.

Opciones de autenticación basadas en SAML para Okta

- **IdPHost** – El nombre de host del IdP especificado.

El valor predeterminado es `NONE`.

- **UID o IdPUserName** – El nombre de usuario de la cuenta de IdP especificada. Si `UID` y `IdPUserName` se proporcionan en la cadena de conexión, se utilizará el valor `UID` a menos que esté vacío.

El valor predeterminado es `NONE`.

- **PWD o IdPPassword** – La contraseña de la cuenta de IdP especificada. Si `PWD` y `IdPPassword` se proporcionan en la cadena de conexión, se utilizará el valor `PWD` a menos que esté vacío.

El valor predeterminado es NONE.

- **OktaApplicationID**— El identificador único proporcionado por Okta y asociado a la AWS IoT SiteWise aplicación. El identificador de la aplicación (AppId) se encuentra en el `entityID` campo proporcionado en los metadatos de la aplicación. Un ejemplo es:

```
entityID="http://www.okta.com//(IdPAppID)
```

El valor predeterminado es NONE.

- **RoleARN** – El nombre de recurso de Amazon (ARN) del rol que la persona que llama va a tomar.

El valor predeterminado es NONE.

- **IdPARN** – El nombre de recurso de Amazon (ARN) del proveedor de SAML en IAM que describe el IdP.

El valor predeterminado es NONE.

Opciones de autenticación basadas en SAML para Azure Active Directory

- **UID** o **IdPUserName** – El nombre de usuario de la cuenta de IdP especificada.

El valor predeterminado es NONE.

- **PWD** o **IdPPassword** – La contraseña de la cuenta de IdP especificada.

El valor predeterminado es NONE.

- **AADApplicationID** – El identificador único de la aplicación registrada en Azure AD.

El valor predeterminado es NONE.

- **AADClientSecret** – El secreto del cliente asociado a la aplicación registrada en Azure AD que se utiliza para autorizar la obtención de los tokens.

El valor predeterminado es NONE.

- **AADTenant** – El ID de inquilino de Azure AD.

El valor predeterminado es NONE.

- **RoleARN** – El nombre de recurso de Amazon (ARN) del rol que la persona que llama va a tomar.

El valor predeterminado es NONE.

- **IdPARN** – El nombre de recurso de Amazon (ARN) del proveedor de SAML en IAM que describe el IdP.

El valor predeterminado es NONE.

AWS Opciones del SDK (avanzadas)

- **RequestTimeout**— El tiempo en milisegundos que el AWS SDK espera una solicitud de consulta antes de que se agote el tiempo de espera. Cualquier valor no positivo desactiva el tiempo de espera de la solicitud.

El valor predeterminado es 3000.

- **ConnectionTimeout**— El tiempo en milisegundos que el AWS SDK espera a que los datos se transfieran a través de una conexión abierta antes de que se agote el tiempo de espera. Un valor de 0 deshabilita el tiempo de espera de conexión. Este valor no debe ser negativo.

El valor predeterminado es 1000.

- **MaxRetryCountClient** – El número máximo de intentos de reintento en el caso de errores reintentables con códigos de error de 5xx en el SDK. El valor no debe ser negativo.

El valor predeterminado es 0.

- **MaxConnections**— El número máximo de conexiones HTTP abiertas simultáneamente con el AWS IoT SiteWise servicio. El valor debe ser un número positivo.

El valor predeterminado es 25.

Opciones de registro del controlador de ODBC

- **LogLevel** – El nivel de registro para el registro de los controladores. Debe ser uno de los siguientes:
 - 0 (APAGADO).
 - 1 (ERROR).
 - 2 (ADVERTENCIA).

- 3 (INFORMACIÓN).
- 4 (DEPURACIÓN).

El valor predeterminado es 1 (ERROR).

Advertencia: El controlador puede registrar información personal al utilizar el modo de registro de DEPURACIÓN.

- **LogOutput** – Carpeta en la que almacenar el archivo de registro.

El valor predeterminado es:

- Windows: %USERPROFILE% o si no está disponible, %HOMEDRIVE%%HOMEPATH%.
- macOS y Linux: \$HOME o si no está disponible, el campo pw_dir del valor devuelto `getpwuid(getuid())` por la función.

Opciones de registro de SDK

El nivel de registro del AWS SDK es independiente del nivel de registro del controlador AWS IoT SiteWise ODBC. Configurar uno no afecta al otro.

El nivel de registro del SDK se establece mediante la variable de entorno `SW_AWS_LOG_LEVEL`. Los valores válidos son:

- OFF
- ERROR
- WARN
- INFO
- DEBUG
- TRACE
- FATAL

Si `SW_AWS_LOG_LEVEL` no está establecido, el nivel de registro del SDK se establece en el valor predeterminado, que es WARN.

Conexión a través de un proxy

El controlador ODBC admite la conexión a AWS IoT SiteWise a través de un proxy. Para utilizar esta característica, configure las siguientes variables de entorno en función de la configuración del proxy:

- **SW_PROXY_HOST** – el host del proxy.
- **SW_PROXY_PORT** – Número de puerto del proxy.
- **SW_PROXY_SCHEME** – El esquema de proxy, ya sea http o https.
- **SW_PROXY_USER** – Nombre de usuario que se utilizará para la autenticación del proxy.
- **SW_PROXY_PASSWORD** – Contraseña de usuario que se utilizará para la autenticación del proxy.
- **SW_PROXY_SSL_CERT_PATH** – El archivo de certificado SSL que se utilizará para conectarse a un proxy HTTPS.
- **SW_PROXY_SSL_CERT_TYPE** – El tipo de certificado SSL del cliente proxy.
- **SW_PROXY_SSL_KEY_PATH** – El archivo de clave privada que se utilizará para conectarse a un proxy HTTPS.
- **SW_PROXY_SSL_KEY_TYPE** – El tipo de archivo de clave privada que se utilizará para conectarse a un proxy HTTPS.
- **SW_PROXY_SSL_KEY_PASSWORD** – La frase de contraseña para el archivo de clave privada que se utilizará para conectarse a un proxy HTTPS.

Ejemplos de cadenas de conexión para el controlador AWS IoT SiteWise ODBC

Ejemplo de conexión al controlador de ODBC con credenciales de IAM

```
Driver={AWS IoT SiteWise ODBC Driver};Auth=IAM;AccessKeyId=(your access key ID);SecretKey=(your secret key);SessionToken=(your session token);Region=us-east-1;
```

Ejemplo de conexión al controlador de ODBC con un perfil

```
Driver={AWS IoT SiteWise ODBC Driver};ProfileName=(the profile name);region=us-east-1;
```

El controlador intentará conectarse con las credenciales proporcionadas en `~/.aws/credentials`, o si se especifica un archivo en la variable de entorno `AWS_SHARED_CREDENTIALS_FILE`, con las credenciales de ese archivo.

Ejemplo de conexión al controlador de ODBC con Okta

```
Driver={AWS IoT SiteWise ODBC Driver};Auth=OKTA;region=us-east-1;idPHost=(your host at Okta);idPUsername=(your user name);idPPassword=(your password);OktaApplicationID=(your Okta AppId);roleARN=(your role ARN);idPARN=(your Idp ARN);
```

Ejemplo de conexión al controlador ODBC con Azure Active Directory (AAD)

```
Driver={AWS IoT SiteWise ODBC Driver};Auth=AAD;region=us-east-1;idPUsername=(your user name);idPPassword=(your password);aadApplicationID=(your AAD AppId);aadClientSecret=(your AAD client secret);aadTenant=(your AAD tenant);roleARN=(your role ARN);idPARN=(your idP ARN);
```

Ejemplo de conexión al controlador ODBC con un punto de conexión específico y un nivel de registro de 2 (ADVERTENCIA)

```
Driver={AWS IoT SiteWise ODBC Driver};Auth=IAM;AccessKeyId=(your access key ID);SecretKey=(your secret key);EndpointOverride=iotsitewise.us-east-1.amazonaws.com;Region=us-east-1;LogLevel=2;
```

Solución de problemas de conexión con el controlador de ODBC

Note

Si el nombre de usuario y la contraseña ya están especificados en el DSN, no los vuelva a especificar cuando el administrador de controladores ODBC los solicite.

Se produce un código de error de 01S02 con un mensaje, Re-writing (*connection string option*) (have you specified it several times?) ocurre cuando se pasa una opción de cadena de conexión más de una vez en la cadena de conexión. Si se especifica una opción más de una vez, se produce un error. Al realizar una conexión con un DSN y una cadena de conexión, si una opción de conexión ya está especificada en el DSN, no la vuelva a especificar en la cadena de conexión.

Interactúa con otros AWS servicios

AWS IoT SiteWise puede publicar datos de activos en el intermediario de mensajes de publicación y suscripción de AWS IoT MQTT, de modo que pueda interactuar con los datos de sus activos desde otros servicios. AWS IoT SiteWise asigna a cada propiedad de activo un tema de MQTT único que puede utilizar para dirigir los datos de sus activos a otros AWS servicios mediante las reglas básicas. Por ejemplo, puede configurar las reglas AWS IoT principales para realizar las siguientes tareas:

- Identificar fallos en los equipos y notificar al personal idóneo mediante el envío de datos a [AWS IoT Events](#).
- Crear un historial de los datos de activos seleccionados para utilizarlos en soluciones de software externas mediante el envío de datos a [Amazon DynamoDB](#).
- Generar informes semanales mediante la activación de una función de [AWS Lambda](#).

Puede seguir un tutorial que recorre los pasos necesarios para configurar una regla que almacene valores de propiedad en DynamoDB. Para obtener más información, consulte [Publicar actualizaciones de valor de propiedad en Amazon DynamoDB](#).

Para obtener más información sobre cómo configurar una regla, consulte [Reglas](#) en la Guía para desarrolladores de AWS IoT .

También puede volver a consumir datos de otros AWS servicios AWS IoT SiteWise. Para ingerir datos mediante la acción de la AWS IoT SiteWise regla, consulte [Ingiera datos para AWS IoT SiteWise usar reglas AWS IoT Core](#).

Temas

- [Descripción de propiedades de activos en temas de MQTT](#)
- [Activa las notificaciones de propiedades de activos en AWS IoT SiteWise](#)
- [Consulte las notificaciones de propiedades de activos en AWS IoT SiteWise](#)
- [Exportación de datos a Amazon S3 con notificaciones sobre propiedades de activos](#)
- [Integre AWS IoT SiteWise con Grafana](#)
- [Integrar AWS IoT SiteWise y AWS IoT TwinMaker](#)
- [Detección de anomalías con Lookout for Equipment](#)

Descripción de propiedades de activos en temas de MQTT

Cada propiedad de activo tiene una ruta de tema MQTT única en el siguiente formato.

```
$aws/sitewise/asset-models/assetModelId/assets/assetId/properties/propertyId
```

Note

AWS IoT SiteWise no admite el comodín de filtro de temas # (de varios niveles) del motor de reglas AWS IoT principales. Puede utilizar el comodín + (de un solo nivel). Por ejemplo, puede utilizar el siguiente filtro de temas para que coincida con todas las actualizaciones de un modelo de activos concreto.

```
$aws/sitewise/asset-models/assetModelId/assets/+/properties/+
```

Para obtener más información sobre los comodines del filtro de temas, consulte [Temas](#) en la Guía para desarrolladores de AWS IoT .

Activa las notificaciones de propiedades de activos en AWS IoT SiteWise

Puede activar las notificaciones de propiedades para publicar actualizaciones de datos de activos y AWS IoT Core, a continuación, ejecutar consultas sobre sus datos. Con las notificaciones de propiedades de los activos, AWS IoT SiteWise proporciona una CloudFormation plantilla que puede utilizar para exportar AWS IoT SiteWise datos a Amazon S3.

Note

Los datos de los activos se envían AWS IoT Core cada vez que los reciben AWS IoT SiteWise, independientemente de si el valor ha cambiado.

Temas

- [Activación de notificaciones sobre propiedades de activos \(consola\)](#)
- [Activación de notificaciones sobre propiedades de activos \(AWS CLI\)](#)

Activación de notificaciones sobre propiedades de activos (consola)

De forma predeterminada, AWS IoT SiteWise no publica las actualizaciones del valor de las propiedades. Puede utilizar la AWS IoT SiteWise consola para activar las notificaciones de una propiedad de un activo.

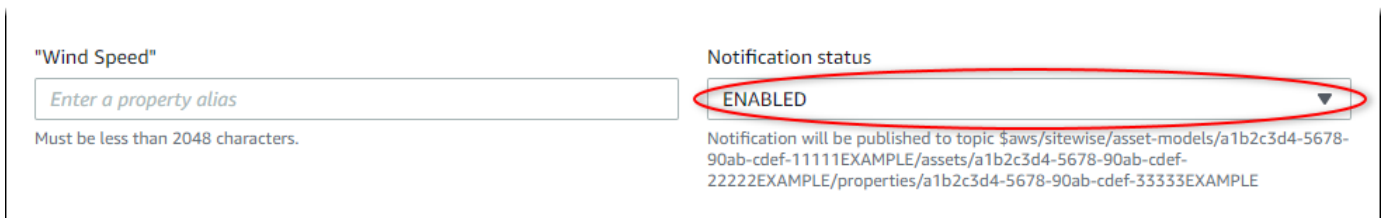
Para habilitar o desactivar las notificaciones de la propiedad de un activo (consola)

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, elija activos.
3. Elija el activo para habilitar las notificaciones de una propiedad.

Tip

Puede elegir el icono de flecha para expandir una jerarquía de activos y encontrar su activo.

4. Seleccione Editar.
5. Para el Estado de la notificación de la propiedad del activo, elija HABILITADO.



The screenshot shows a form for editing a property. On the left, there is a text input field labeled "Wind Speed" with a placeholder "Enter a property alias" and a note "Must be less than 2048 characters." On the right, there is a dropdown menu labeled "Notification status" with "ENABLED" selected. Below the dropdown, there is a long alphanumeric string representing the notification topic: "Notification will be published to topic \$aws/sitewise/asset-models/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE/assets/a1b2c3d4-5678-90ab-cdef-22222EXAMPLE/properties/a1b2c3d4-5678-90ab-cdef-33333EXAMPLE". A red oval highlights the "ENABLED" option in the dropdown menu.

También puede elegir DESHABILITADO para desactivar las notificaciones de la propiedad del activo.

6. Seleccione Save.

Activación de notificaciones sobre propiedades de activos (AWS CLI)

De forma predeterminada, AWS IoT SiteWise no publica las actualizaciones del valor de las propiedades. Puede utilizar el AWS Command Line Interface (AWS CLI) para activar o desactivar las notificaciones de una propiedad de un activo.

Debe conocer los `assetId` de sus activos y los `propertyId` de las propiedades para completar este procedimiento. También puede utilizar el ID externo. Si has creado un activo y no lo

sabesassetId, usa la [ListAssets](#) API para enumerar todos los activos de un modelo específico. Utilice la [DescribeAsset](#) operación para ver las propiedades de su activo, incluida la propiedad IDs.

Utilice la [UpdateAssetProperty](#) operación para activar o desactivar las notificaciones de una propiedad de un activo. Especifique los siguientes parámetros:

- `assetId`: el ID del activo.
- `propertyId`: el ID de la propiedad del activo.
- `propertyNotificationState`: el estado de notificación del valor de la propiedad, `ENABLED` o `DISABLED`.
- `propertyAlias`: el alias de la propiedad. Especifique el alias existente de la propiedad cuando actualice el estado de notificación. Si omite este parámetro, se elimina el alias existente de la propiedad.

Para habilitar o desactivar las notificaciones de la propiedad de un activo (CLI)

1. Ejecute el siguiente comando para recuperar el alias de la propiedad del activo. `asset-id` Sustitúyalo por el identificador del activo y `property-id` por el identificador de la propiedad.

```
aws iotsitewise describe-asset-property \  
  --asset-id asset-id \  
  --property-id property-id
```

La operación devuelve una respuesta que contiene detalles de la propiedad del activo en el siguiente formato. El alias de propiedad se encuentra en `assetProperty.alias` en el objeto JSON.

```
{  
  "assetId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",  
  "assetName": "Wind Turbine 7",  
  "assetModelId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",  
  "assetProperty": {  
    "id": "a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",  
    "name": "Wind Speed",  
    "alias": "/company/windfarm/3/turbine/7/windspeed",  
    "notification": {  
      "topic": "$aws/sitewise/asset-models/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE/  
assets/a1b2c3d4-5678-90ab-cdef-22222EXAMPLE/properties/a1b2c3d4-5678-90ab-  
cdef-33333EXAMPLE",
```

```

    "state": "DISABLED"
  },
  "dataType": "DOUBLE",
  "unit": "m/s",
  "type": {
    "measurement": {}
  }
}
}

```

2. Ejecute el siguiente comando para habilitar las notificaciones de la propiedad del activo. *property-alias* Sustitúyalo por el alias de la propiedad de la respuesta del comando anterior u `--property-alias` omite actualizar la propiedad sin un alias.

```

aws iotsitewise update-asset-property \
  --asset-id asset-id \
  --property-id property-id \
  --property-notification-state ENABLED \
  --property-alias property-alias

```

También puede pasar `--property-notification-state DISABLED` para desactivar las notificaciones de la propiedad del activo.

Consulte las notificaciones de propiedades de activos en AWS IoT SiteWise

Para consultar las notificaciones de propiedades de los activos, cree AWS IoT Core reglas compuestas por sentencias SQL.

AWS IoT SiteWise publica las actualizaciones de los datos de propiedades de los activos en AWS IoT Core en el siguiente formato.

```

{
  "type": "PropertyValueUpdate",
  "payload": {
    "assetId": "String",
    "propertyId": "String",
    "values": [
      {
        "timestamp": {

```

```

    "timeInSeconds": Number,
    "offsetInNanos": Number
  },
  "quality": "String",
  "value": {
    "booleanValue": Boolean,
    "doubleValue": Number,
    "integerValue": Number,
    "stringValue": "String",
    "nullValue": {
      "valueType": "String"
    }
  }
}
]
}
}

```

Cada estructura de la values lista es una estructura timestamp-quality-value (TQV).

- El `timestamp` contiene el tiempo actual en formato de tiempo Unix en segundos con desplazamiento en nanosegundos.
- `quality` contiene una de las siguientes cadenas, que indican la calidad del punto de datos:
 - `GOOD`: los datos no se ven afectados por ningún problema.
 - `BAD`: los datos se ven afectados por un problema, como un fallo del sensor.
 - `UNCERTAIN`: los datos se ven afectados por un problema, como la falta de precisión de un sensor.
- El `value` contiene uno de los siguientes campos, en función del tipo de propiedad:
 - `booleanValue`
 - `doubleValue`
 - `integerValue`
 - `stringValue`
 - `nullValue`

`nullValue`— Una estructura con el siguiente campo que indica el tipo de valor de la propiedad con un valor nulo y una calidad de `BAD` o `UNCERTAIN`

- `valueType`— Enuma de {"B», «D», «S», «I"}

Para analizar valores fuera de la matriz `values`, debe usar consultas complejas de objetos anidados en las instrucciones SQL de sus reglas. Para obtener más información, consulte [Consultas de objetos anidados](#) en la Guía para desarrolladores de AWS IoT , o consulte el tutorial de [Publicar actualizaciones de valor de propiedad en Amazon DynamoDB](#) para ver un ejemplo concreto de análisis sintáctico de mensajes de notificación de propiedades de activo.

Example Consulta de ejemplo para extraer la matriz de valores

La siguiente instrucción demuestra cómo consultar la matriz de valores de propiedad actualizados para una propiedad de tipo doble específica en todos los activos con esa propiedad.

```
SELECT
  (SELECT VALUE (value.doubleValue) FROM payload.values) AS windspeed
FROM
  '$aws/sitewise/asset-models/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE/assets/+/
properties/a1b2c3d4-5678-90ab-cdef-33333EXAMPLE'
WHERE
  type = 'PropertyValueUpdate'
```

La instrucción de consulta de regla anterior genera los datos con el siguiente formato.

```
{
  "windspeed": [
    26.32020195042838,
    26.282584572975477,
    26.352566977372508,
    26.283084346171442,
    26.571883739599322,
    26.60684140743005,
    26.628738636715045,
    26.273486932802125,
    26.436379105473964,
    26.600590095377303
  ]
}
```

Example Consulta de ejemplo para extraer un solo valor

La siguiente instrucción demuestra cómo consultar el primer valor de la matriz de valores de propiedad para una propiedad de tipo doble específica en todos los activos con esa propiedad.

```
SELECT
```

```
get((SELECT VALUE (value.doubleValue) FROM payload.values), 0) AS windspeed
FROM
'$aws/sitewise/asset-models/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE/assets/+
properties/a1b2c3d4-5678-90ab-cdef-33333EXAMPLE'
WHERE
type = 'PropertyValueUpdate'
```

La instrucción de consulta de regla anterior genera los datos con el siguiente formato.

```
{
  "windspeed": 26.32020195042838
}
```

Important

Esta instrucción de consulta de regla ignora las actualizaciones de valores distintas de la primera en cada lote. Cada lote puede contener hasta 10 valores. Si necesita incluir los valores restantes, debe configurar una solución más compleja para generar valores de propiedad de activos a otros servicios. Por ejemplo, puede configurar una regla con una AWS Lambda acción para volver a publicar cada valor de la matriz en otro tema y configurar otra regla para consultar ese tema y publicar cada valor en la acción de regla deseada.

Exportación de datos a Amazon S3 con notificaciones sobre propiedades de activos

Puede exportar los datos entrantes AWS IoT SiteWise a un bucket de Amazon S3 de su cuenta. Puede realizar una copia de seguridad de los datos en un formato que le permita crear informes históricos o para analizar los datos con métodos complejos.

Para exportar datos de series temporales AWS IoT SiteWise, habilite la función de capa fría para almacenar los datos en un bucket de Amazon S3. Consulte [Administrar el almacenamiento de datos en AWS IoT SiteWise](#) para obtener más información.

Para exportar el modelo de activos y los metadatos de los activos AWS IoT SiteWise, utilice la función de operaciones masivas para exportar los metadatos a un bucket de Amazon S3. Consulte [Operaciones masivas con activos y modelos](#) para obtener más información.

Integre AWS IoT SiteWise con Grafana

Grafana es una plataforma de visualización de datos que se utiliza para visualizar y monitorear los datos en los paneles de control. En la versión 10.4.0 y posteriores de Grafana, utilice el AWS IoT SiteWise complemento para visualizar los datos de sus AWS IoT SiteWise activos en los paneles de Grafana. Los usuarios pueden visualizar datos de múltiples AWS fuentes (como AWS IoT SiteWise Amazon Timestream y CloudWatch Amazon) y otras fuentes de datos con un único panel de Grafana.

Tienes dos opciones para usar el complemento: AWS IoT SiteWise

- Servidores locales de Grafana

Puede configurar el AWS IoT SiteWise complemento en un servidor Grafana que administre. Para obtener más información sobre cómo añadir y usar el complemento, consulta el archivo [README de la AWS IoT SiteWise fuente](#) de datos en el sitio web. GitHub

- AWS Managed Service for Grafana

Puede utilizar el AWS IoT SiteWise complemento en AWS Managed Service for Grafana (AMG). AMG administra los servidores de Grafana por usted para que pueda visualizar sus datos sin tener que construir, empaquetar ni implementar ningún hardware ni cualquier otra infraestructura de Grafana. Para obtener más información, consulte los siguientes temas en la Guía del usuario de AWS Managed Service for Grafana:

- [¿Qué es Amazon Managed Service for Grafana \(AMG\)?](#)
- [Uso de la fuente de datos AWS IoT SiteWise](#)

Example Ejemplo de panel de control de Grafana

El siguiente panel de control de Grafana visualiza el [parque eólico de demostración](#). Puede acceder a este panel de control de demostración en el sitio web [Grafana Play](#).



Integrar AWS IoT SiteWise y AWS IoT TwinMaker

La integración con AWS IoT TwinMaker ella permite acceder a funciones sólidas AWS IoT SiteWise, como la ExecuteQuery API de recuperación de AWS IoT SiteWise datos y la búsqueda avanzada de activos en la AWS IoT SiteWise consola. Para integrar los servicios y utilizar estas características, primero debe habilitar la integración.

Temas

- [Habilitación de la integración](#)
- [Integrar y AWS IoT SiteWise AWS IoT TwinMaker](#)

Habilitación de la integración

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones. El elemento `Action` de una política JSON describe las acciones que puede utilizar para conceder o denegar el acceso en una política. Para obtener más información sobre las acciones AWS IoT SiteWise admitidas, consulte [las acciones definidas AWS IoT SiteWise](#) en la Referencia de autorización de servicios.

Para obtener más información sobre las funciones AWS IoT TwinMaker vinculadas a un servicio, consulte [Funciones vinculadas a servicios AWS IoT TwinMaker en la Guía del usuario](#).AWS IoT TwinMaker

Antes de poder realizar la integración AWS IoT SiteWise AWS IoT TwinMaker, debe conceder los siguientes permisos que le permitan integrarse en un espacio de trabajo AWS IoT SiteWise vinculado: AWS IoT TwinMaker

- `iotsitewise:EnableSiteWiseIntegration`— Permite AWS IoT SiteWise integrarse con un espacio de AWS IoT TwinMaker trabajo vinculado. Esta integración permite AWS IoT TwinMaker leer toda la información de modelado AWS IoT SiteWise a través de una función AWS IoT TwinMaker vinculada al servicio. Para habilitar este permiso, añada la siguiente política a su rol de IAM:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iotsitewise:EnableSiteWiseIntegration"
      ],
      "Resource": "*"
    }
  ]
}
```

Integrar y AWS IoT SiteWise AWS IoT TwinMaker

Para integrar AWS IoT SiteWise y AWS IoT TwinMaker, debe tener lo siguiente:

- AWS IoT SiteWise función vinculada al servicio configurada en su cuenta
- AWS IoT TwinMaker función vinculada a un servicio configurada en tu cuenta
- AWS IoT TwinMaker espacio de trabajo con un ID `IoTSiteWiseDefaultWorkspace` en tu cuenta en la región.

Para integrarlo mediante la AWS IoT SiteWise consola

Cuando vea el banner Integración con AWS IoT TwinMaker en la consola, seleccione Conceder permiso. Los requisitos previos se crean en su cuenta.

Para realizar la integración mediante AWS CLI

Para integrar AWS IoT SiteWise y AWS IoT TwinMaker utilizar el AWS CLI, introduzca los siguientes comandos:

1. Llame a `CreateServiceLinkedRole` con un `AWSServiceName` de `iotsitewise.amazonaws.com`.

```
aws iam create-service-linked-role --aws-service-name iotsitewise.amazonaws.com
```

2. Llame a `CreateServiceLinkedRole` con un `AWSServiceName` de `iottwinmaker.amazonaws.com`.

```
aws iam create-service-linked-role --aws-service-name iottwinmaker.amazonaws.com
```

3. Llame a `CreateWorkspace` con un ID de `IoTSiteWiseDefaultWorkspace`.

```
aws iottwinmaker create-workspace --workspace-id IoTSiteWiseDefaultWorkspace
```

Detección de anomalías con Lookout for Equipment

Note

La detección de anomalías solo se encuentra disponible en las regiones donde está disponible Amazon Lookout for Equipment.

Puede realizar la integración AWS IoT SiteWise con Amazon Lookout for Equipment para obtener información sobre sus equipos industriales mediante la detección de anomalías y el mantenimiento predictivo de los equipos industriales. Lookout for Equipment es un servicio de machine learning (ML) para supervisar equipos industriales que detecta un comportamiento anormal de los equipos e identifica errores potenciales. Con Lookout for Equipment, puede implementar programas de mantenimiento predictivo e identificar los procesos poco óptimos en los equipos. Para obtener más información sobre Lookout for Equipment, consulte [What is Amazon Lookout for Equipment?](#) en la Guía del usuario de Amazon Lookout for Equipment.

Al crear una predicción para entrenar un modelo de aprendizaje automático a fin de detectar el comportamiento anómalo del equipo, AWS IoT SiteWise envía los valores de las propiedades de los activos a Lookout for Equipment para entrenar un modelo de aprendizaje automático a fin de detectar el comportamiento anómalo del equipo. Para crear una definición de predicción en un modelo de activos, debe especificar los roles de IAM necesarios para que Lookout for Equipment acceda a los datos y las propiedades que se deben enviar a Lookout for Equipment y enviar datos procesados a Amazon S3. Para obtener más información, consulte [Cree modelos de activos en AWS IoT SiteWise](#).

Para integrar AWS IoT SiteWise Lookout for Equipment, realizarás los siguientes pasos de alto nivel:

- Añada una definición de predicción a un modelo de activos que describa las propiedades de las que desea hacer un seguimiento. La definición de predicción es un conjunto reutilizable de mediciones, transformaciones y métricas que se utiliza para crear predicciones sobre los activos que se basan en ese modelo de activos.
- Entrene la predicción en función de los datos históricos que proporcione.
- Programa la inferencia, que indica la AWS IoT SiteWise frecuencia con la que se debe ejecutar una predicción específica.

Una vez programada la inferencia, el modelo de Lookout for Equipment supervisa los datos que recibe de su equipo y busca anomalías en el comportamiento de los equipos. Puede ver y analizar

los resultados en SiteWise Monitor, mediante las operaciones de la API AWS IoT SiteWise GET o la consola Lookout for Equipment. También puede crear alarmas utilizando detectores de alarmas desde el modelo de activos para recibir alertas sobre el comportamiento anormal de los equipos.

Temas

- [Adición de una definición de predicción \(consola\)](#)
- [Entrenamiento de una predicción \(consola\)](#)
- [Inicie o detenga la inferencia sobre una predicción \(consola\)](#)
- [Adición de una definición de predicción \(CLI\)](#)
- [Entrenamiento de una predicción e inicio de la inferencia \(CLI\)](#)
- [Entrenamiento de una predicción \(CLI\)](#)
- [Inicie o detenga la inferencia sobre una predicción \(CLI\)](#)

Adición de una definición de predicción (consola)

Para empezar a enviar los datos recopilados por AWS IoT SiteWise Lookout for Equipment, debes añadir AWS IoT SiteWise una definición de predicción a un modelo de activos.

Para añadir una definición de predicción a un modelo de AWS IoT SiteWise activos

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, elija Modelos y seleccione el modelo de activos al que desea añadir la definición de predicción.
3. Elija Predicciones.
4. Seleccione Agregar definición de predicciones.
5. Defina los detalles sobre la definición de predicción.
 - a. Introduzca un Nombre único y una Descripción para la definición de predicción. Elija el nombre con cuidado porque, una vez que haya creado la definición de predicción, no podrá modificarlo.
 - b. Cree o seleccione un rol de permisos de IAM que permita a AWS IoT SiteWise compartir los datos de sus activos con Amazon Lookout for Equipment. El rol debe tener las siguientes políticas de IAM y de confianza. Para obtener ayuda en la creación de roles, consulte [Crear un rol mediante políticas de confianza personalizadas](#).

Política de IAM

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "L4EPermissions",
      "Effect": "Allow",
      "Action": [
        "lookoutequipment:CreateDataset",
        "lookoutequipment:CreateModel",
        "lookoutequipment:CreateInferenceScheduler",
        "lookoutequipment:DescribeDataset",
        "lookoutequipment:DescribeModel",
        "lookoutequipment:DescribeInferenceScheduler",
        "lookoutequipment:ListInferenceExecutions",
        "lookoutequipment:StartDataIngestionJob",
        "lookoutequipment:StartInferenceScheduler",
        "lookoutequipment:UpdateInferenceScheduler",
        "lookoutequipment:StopInferenceScheduler"
      ],
      "Resource": [
        "arn:aws:lookoutequipment:us-  
east-1:123456789012:inference-scheduler/IoTSiteWise_*",
        "arn:aws:lookoutequipment:us-east-1:123456789012:model/  
IoTSiteWise_*",
        "arn:aws:lookoutequipment:us-east-1:123456789012:dataset/  
IoTSiteWise_*"
      ]
    },
    {
      "Sid": "L4EPermissions2",
      "Effect": "Allow",
      "Action": [
        "lookoutequipment:DescribeDataIngestionJob"
      ],
      "Resource": "*"
    },
    {
      "Sid": "S3Permissions",
      "Effect": "Allow",
      "Action": [

```

```

        "s3:CreateBucket",
        "s3:ListBucket",
        "s3:PutObject",
        "s3:GetObject"
    ],
    "Resource": [
        "arn:aws:s3:::iotsitewise-*"
    ]
},
{
    "Sid": "IAMPermissions",
    "Effect": "Allow",
    "Action": [
        "iam:GetRole",
        "iam:PassRole"
    ],
    "Resource": "arn:aws:iam::111122223333:role/Role_name"
}
]
}

```

Política de confianza

JSON

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "iotsitewise.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "123456789012"
                },
                "ArnEquals": {
                    "aws:SourceArn": "arn:aws:iotsitewise:us-east-1:123456789012:asset/*"
                }
            }
        }
    ]
}

```

```
    }
  },
  {
    "Effect": "Allow",
    "Principal": {
      "Service": "lookoutequipment.amazonaws.com"
    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "StringEquals": {
        "aws:SourceAccount": "123456789012"
      },
      "ArnEquals": {
        "aws:SourceArn": "arn:aws:lookoutequipment:us-
east-1:123456789012:*"
      }
    }
  }
]
```

- c. Elija Siguiente.
6. Seleccione los atributos de datos (mediciones, transformaciones y métricas) que quiera enviar a Lookout for Equipment.
 - a. (Opcional) Seleccione las mediciones.
 - b. (Opcional) Seleccione las transformaciones.
 - c. (Opcional) Seleccione las métricas.
 - d. Elija Siguiente.
 7. Revise sus selecciones. Para añadir la definición de predicción al modelo de activos, en la página de resumen, seleccione Agregar definición de predicciones.

También puede Editar o Eliminar una definición de predicción existente que tenga predicciones activas asociadas.

Entrenamiento de una predicción (consola)

Después de añadir una definición de predicción a un modelo de activos, puede entrenar las predicciones que están en sus activos.

Para entrenar una predicción en AWS IoT SiteWise

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, seleccione Activos y el activo que desea supervisar.
3. Elija Predicciones.
4. Seleccione las predicciones que desea entrenar.
5. En Acciones, seleccione Comenzar entrenamiento y haga lo siguiente:
 - a. En Detalles de la predicción, selecciona un rol de permisos de IAM que te permita AWS IoT SiteWise compartir los datos de tus activos con Lookout for Equipment. Si necesita crear un nuevo rol, elija Crear un nuevo rol.
 - b. En Configuración de los datos de entrenamiento, introduzca un Intervalo de tiempo de los datos de entrenamiento para seleccionar qué datos desea usar para entrenar la predicción.
 - c. (Opcional) Seleccione la frecuencia de muestreo de los datos después del procesamiento posterior.
 - d. (Opcional) En Etiquetas de datos, proporcione un bucket y un prefijo de Amazon S3 que contenga los datos de etiquetado. Para obtener más información sobre el etiquetado de datos, consulte [Labeling your data](#) en la Guía del usuario de Amazon Lookout for Equipment.
 - e. Elija Siguiente.
6. (Opcional) Si quiere que la predicción esté activa en cuanto termine el entrenamiento, en Configuración avanzada, seleccione Activar automáticamente la predicción después del entrenamiento y, a continuación, haga lo siguiente:
 - a. En Datos de entrada, en Frecuencia de carga de datos, defina la frecuencia con la que se cargan los datos y, en Tiempo de retraso de desplazamiento, defina la cantidad de búfer que desea utilizar.
 - b. Elija Siguiente.
7. Revise los detalles de la predicción y seleccione Guardar e iniciar.

Inicie o detenga la inferencia sobre una predicción (consola)

Note

Los cargos de Lookout for Equipment se aplican a las inferencias programadas con los datos transferidos AWS IoT SiteWise entre Lookout for Equipment y Lookout for Equipment. Para obtener más información, consulte [Amazon Lookout for Equipment pricing](#).

Si ha añadido la predicción `lookoutequipment:CreateDataset`, pero no ha decidido activarla después del entrenamiento, debe activarla para iniciar la supervisión de sus activos.

Inicio de la inferencia de una predicción

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, elija Activos y seleccione el activo al que se añade la predicción.
3. Elija Predicciones.
4. Seleccione las predicciones que desea activar.
5. En Acciones, seleccione Iniciar inferencia y haga lo siguiente:
 - a. En Datos de entrada, en Frecuencia de carga de datos, defina la frecuencia con la que se cargan los datos y, en Tiempo de retraso de desplazamiento, defina la cantidad de búfer que desea utilizar.
 - b. Elija Guardar e iniciar.

Detención de la inferencia de una predicción

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, elija Activos y seleccione el activo al que se añade la predicción.
3. Elija Predicciones.
4. Seleccione las predicciones que desea detener.
5. En Acciones, elija Detener inferencia.

Adición de una definición de predicción (CLI)

Para definir una definición de predicción en un modelo de activos nuevo o existente, puede usar `aws iot-sitewise create-prediction-definition`. AWS Command Line Interface (AWS CLI). Tras definir la definición de predicción en el modelo de activos, entrena y programa la inferencia de una predicción sobre un activo para detectar anomalías con Lookout for Equipment. AWS IoT SiteWise

Requisitos previos

Para completar estos pasos debe tener un modelo de activos y haber creado al menos un activo. Para obtener más información, consulte [Creación de un modelo de activos \(AWS CLI\)](#) y [Creación de un activo \(AWS CLI\)](#).

Si es la primera vez que lo utiliza AWS IoT SiteWise, debe llamar a la operación de la `CreateBulkImportJob` API para importar los valores de las propiedades de los activos AWS IoT SiteWise, que se utilizarán para entrenar el modelo. Para obtener más información, consulte [Crea un trabajo de importación AWS IoT SiteWise masiva \(AWS CLI\)](#).

Adición de una definición de predicción

1. Cree un archivo denominado `asset-model-payload.json`. Siga los pasos de estas otras secciones para añadir los detalles de su modelo de activos al archivo, pero no envíe la solicitud para crear o actualizar el modelo de activos.
 - Para obtener más información acerca de cómo crear un modelo de activos, consulte [Creación de un modelo de activos \(AWS CLI\)](#)
 - Para obtener más información acerca de cómo actualizar un modelo de activos existente, consulte [Actualice un modelo de activo, un modelo de componentes o una interfaz \(AWS CLI\)](#)
2. Añada un modelo compuesto de Lookout for Equipment (`assetModelCompositeModels`) al modelo de activos añadiendo el siguiente código.
 - Reemplace *Property* por el ID de las propiedades que desea incluir. Para obtenerlos IDs, llama [DescribeAssetModel](#).
 - *RoleARN* Sustitúyalo por el ARN de un rol de IAM que permita a Lookout for Equipment acceder a tus datos. AWS IoT SiteWise

```
{  
  ...  
}
```

```

"assetModelCompositeModels": [
  {
    "name": "L4Epredictiondefinition",
    "type": "AWS/L4E_ANOMALY",
    "properties": [
      {
        "name": "AWS/L4E_ANOMALY_RESULT",
        "dataType": "STRUCT",
        "dataTypeSpec": "AWS/L4E_ANOMALY_RESULT",
        "unit": "none",
        "type": {
          "measurement": {}
        }
      },
      {
        "name": "AWS/L4E_ANOMALY_INPUT",
        "dataType": "STRUCT",
        "dataTypeSpec": "AWS/L4E_ANOMALY_INPUT",
        "type": {
          "attribute": {
            "defaultValue": "{\"properties\": [\"Property1\", \"Property2\"]}"
          }
        }
      },
      {
        "name": "AWS/L4E_ANOMALY_PERMISSIONS",
        "dataType": "STRUCT",
        "dataTypeSpec": "AWS/L4E_ANOMALY_PERMISSIONS",
        "type": {
          "attribute": {
            "defaultValue": "{\"roleArn\": \"RoleARN\"}"
          }
        }
      },
      {
        "name": "AWS/L4E_ANOMALY_DATASET",
        "dataType": "STRUCT",
        "dataTypeSpec": "AWS/L4E_ANOMALY_DATASET",
        "type": {
          "attribute": {}
        }
      },
      {
        "name": "AWS/L4E_ANOMALY_MODEL",

```

```

    "dataType": "STRUCT",
    "dataTypeSpec": "AWS/L4E_ANOMALY_MODEL",
    "type": {
      "attribute": {}
    }
  },
  {
    "name": "AWS/L4E_ANOMALY_INFERENCE",
    "dataType": "STRUCT",
    "dataTypeSpec": "AWS/L4E_ANOMALY_INFERENCE",
    "type": {
      "attribute": {}
    }
  },
  {
    "name": "AWS/L4E_ANOMALY_TRAINING_STATUS",
    "dataType": "STRUCT",
    "dataTypeSpec": "AWS/L4E_ANOMALY_TRAINING_STATUS",
    "type": {
      "attribute": {
        "defaultValue": "{}"
      }
    }
  },
  {
    "name": "AWS/L4E_ANOMALY_INFERENCE_STATUS",
    "dataType": "STRUCT",
    "dataTypeSpec": "AWS/L4E_ANOMALY_INFERENCE_STATUS",
    "type": {
      "attribute": {
        "defaultValue": "{}"
      }
    }
  }
]
}

```

3. Cree el modelo de activos o actualice el modelo de activos existente. Realice una de las siguientes acciones:
 - Ejecute el siguiente comando para crear el modelo de activos:

```
aws iotsitewise create-asset-model --cli-input-json file://asset-model-payload.json
```

- Ejecute el siguiente comando para actualizar el modelo de activos existente. Reemplace *asset-model-id* por el ID del modelo de activos que desea actualizar.

```
aws iotsitewise update-asset-model \  
  --asset-model-id asset-model-id \  
  --cli-input-json file://asset-model-payload.json
```

Después de ejecutar el comando, anote `assetModelId` en la respuesta.

Entrenamiento de una predicción e inicio de la inferencia (CLI)

Ahora que la definición de predicción está establecida, puede entrenar activos en función de ella e iniciar la inferencia. Si desea entrenar su predicción pero no iniciar la inferencia, pase a [Entrenamiento de una predicción \(CLI\)](#). Para entrenar la predicción e iniciar la inferencia sobre el activo, necesitará el `assetId` del recurso de destino.

Entrenamiento e inicio de la inferencia de la predicción

1. Ejecute el comando siguiente para buscar el `assetModelCompositeModelId` en `assetModelCompositeModelSummaries`. Sustituya *asset-model-id* por el ID del modelo de activos que ha creado en [Actualice un modelo de activo, un modelo de componentes o una interfaz \(AWS CLI\)](#).

```
aws iotsitewise describe-asset-model \  
  --asset-model-id asset-model-id \  
  --cli-input-json file://asset-model-payload.json
```

2. Ejecute el siguiente comando para buscar el `actionDefinitionId` de la acción `TrainingWithInference`. Sustituya *asset-model-id* por el ID utilizado en el paso anterior y sustituya *asset-model-composite-model-id* por el ID devuelto en el paso anterior.

```
aws iotsitewise describe-asset-model-composite-model \  
  --asset-model-id asset-model-id \  
  --asset-model-composite-model-id asset-model-composite-model-id \  
  --cli-input-json file://asset-model-composite-model-payload.json
```

3. Cree un archivo con el nombre `train-start-inference-prediction.json` y agregue el siguiente código, sustituyendo lo siguiente:

- *asset-id* por el ID del activo de destino
- *action-definition-id* con el ID de la acción TrainingWithInference
- *StartTime* por el inicio de los datos de entrenamiento, proporcionados en segundos epoch
- *EndTime* por el fin de los datos de entrenamiento, proporcionados en segundos epoch
- *TargetSamplingRate* por la frecuencia de muestreo de los datos tras el procesamiento posterior por parte de Lookout for Equipment. Los valores permitidos son: PT1S | PT5S | PT10S | PT15S | PT30S | PT1M | PT5M | PT10M | PT15M | PT30M | PT1H.

```
{
  "targetResource": {
    "assetId": "asset-id"
  },
  "actionDefinitionId": "action-definition-Id",
  "actionPayload": {
    "stringValue": "{\"l4ETrainingWithInference\":{\"trainingWithInferenceMode\": \"START\", \"trainingPayload\": {\"exportDataStartTime\": StartTime, \"exportDataEndTime\": EndTime}, \"targetSamplingRate\": \"TargetSamplingRate\"}, \"inferencePayload\": {\"dataDelayOffsetInMinutes\": 0, \"dataUploadFrequency\": \"PT5M\"}}}"
  }
}
```

4. Ejecute el siguiente comando para iniciar el entrenamiento y la inferencia:

```
aws iotsitewise execute-action --cli-input-json file://train-start-inference-prediction.json
```

Entrenamiento de una predicción (CLI)

Ahora que la definición de predicción está establecida, puede entrenar activos en función de ella. Para entrenar la predicción sobre el activo, necesitará el `assetId` del recurso de destino.

Entrenamiento de la predicción

1. Ejecute el comando siguiente para buscar el `assetModelCompositeModelId` en `assetModelCompositeModelSummaries`. Sustituya *asset-model-id* por el ID del modelo


de activos que ha creado en [Actualice un modelo de activo, un modelo de componentes o una interfaz \(AWS CLI\)](#).

```
aws iotsitewise describe-asset-model \  
  --asset-model-id asset-model-id \  
  \
```

2. Ejecute el siguiente comando para buscar el `actionDefinitionId` de la acción Training. Sustituya *asset-model-id* por el ID utilizado en el paso anterior y sustituya *asset-model-composite-model-id* por el ID devuelto en el paso anterior.

```
aws iotsitewise describe-asset-model-composite-model \  
  --asset-model-id asset-model-id \  
  --asset-model-composite-model-id asset-model-composite-model-id \  
  \
```

3. Cree un archivo con el nombre `train-prediction.json` y agregue el siguiente código, sustituyendo lo siguiente:
 - *asset-id* por el ID del activo de destino
 - *action-definition-id* por el ID de la acción de entrenamiento
 - *StartTime* por el inicio de los datos de entrenamiento, proporcionados en segundos epoch
 - *EndTime* por el fin de los datos de entrenamiento, proporcionados en segundos epoch
 - (Opcional) *BucketName* por el nombre del bucket de Amazon S3 que contiene los datos de su etiqueta
 - (Opcional) *Prefix* por el prefijo asociado con el bucket de Amazon S3.
 - *TargetSamplingRate* por la frecuencia de muestreo de los datos tras el procesamiento posterior por parte de Lookout for Equipment. Los valores permitidos son: PT1S | PT5S | PT10S | PT15S | PT30S | PT1M | PT5M | PT10M | PT15M | PT30M | PT1H.

 Note

Incluya el nombre y el prefijo del bucket o ninguno de ellos.

```
{  
  "targetResource": {  
    "assetId": "asset-id"  
  },  
  "actionDefinitionId": "action-definition-Id",  
}
```

```
"actionPayload":{ "stringValue": "{\"l4ETraining\": {\"trainingMode\":
\\\"START\\\",\\\"exportDataStartTime\\\": StartTime, \\\"exportDataEndTime\\\": EndTime,
\\\"targetSamplingRate\\\":\\\"TargetSamplingRate\\\"}, \\\"labelInputConfiguration\\\":
{\\\"bucketName\\\": \\\"BucketName\\\", \\\"prefix\\\": \\\"Prefix\\\"}}}"
}
}
```

4. Ejecute el siguiente comando para iniciar el entrenamiento:

```
aws iotsitewise execute-action --cli-input-json file://train-prediction.json
```

Antes de poder iniciar la inferencia se debe completar el entrenamiento. Para comprobar el estado del entrenamiento, realice una de las siguientes acciones:

- Desde la consola, navegue hasta el activo en el que se encuentra la predicción.
- Desde el AWS CLI, llame BatchGetAssetPropertyValue usando el propertyId de la trainingStatus propiedad.

Inicie o detenga la inferencia sobre una predicción (CLI)

Una vez entrenada la predicción, puede iniciar la inferencia para indicar a Lookout for Equipment que comience a supervisar sus activos. Para iniciar o detener la inferencia, necesitará el assetId del recurso de destino.

Inicio de la inferencia

1. Ejecute el comando siguiente para buscar el assetModelCompositeModelId en assetModelCompositeModelSummaries. Sustituya *asset-model-id* por el ID del modelo de activos que ha creado en [Actualice un modelo de activo, un modelo de componentes o una interfaz \(AWS CLI\)](#).

```
aws iotsitewise describe-asset-model \
  --asset-model-id asset-model-id \
```

2. Ejecute el siguiente comando para buscar el actionDefinitionId de la acción Inference. Sustituya *asset-model-id* por el ID utilizado en el paso anterior y sustituya *asset-model-composite-model-id* por el ID devuelto en el paso anterior.

```
aws iotsitewise describe-asset-model-composite-model \
```

```
--asset-model-id asset-model-id \  
--asset-model-composite-model-id asset-model-composite-model-id \  

```

3. Cree un archivo con el nombre `start-inference.json` y agregue el siguiente código, sustituyendo lo siguiente:

- *asset-id* por el ID del activo de destino
- *action-definition-id* por el ID de la acción de inferencia inicial
- *Offset* por la cantidad de búfer que desea utilizar
- *Frequency* por la frecuencia con la que se cargan los datos

```
{  
  "targetResource": {  
    "assetId": "asset-id"  
  },  
  "actionDefinitionId": "action-definition-Id",  
  "actionPayload": { "stringValue": "{\\"l4EInference\\": {\\"inferenceMode\\":\\"START  
\\",\\"dataDelayOffsetInMinutes\\": Offset, \\"dataUploadFrequency\\": \\"Frequency\\"}}"  
  }  
}
```

4. Ejecute el siguiente comando para iniciar la inferencia.

```
aws iotsitewise execute-action --cli-input-json file://start-inference.json
```

Detención de la inferencia

1. Ejecute el comando siguiente para buscar el `assetModelCompositeModelId` en `assetModelCompositeModelSummaries`. Sustituya *asset-model-id* por el ID del modelo de activos que ha creado en [Actualice un modelo de activo, un modelo de componentes o una interfaz \(AWS CLI\)](#).

```
aws iotsitewise describe-asset-model \  
--asset-model-id asset-model-id \  

```

2. Ejecute el siguiente comando para buscar el `actionDefinitionId` de la acción Inference. Sustituya *asset-model-id* por el ID utilizado en el paso anterior y sustituya *asset-model-composite-model-id* por el ID devuelto en el paso anterior.

```
aws iotsitewise describe-asset-model-composite-model \  
  --asset-model-id asset-model-id \  
  --asset-model-composite-model-id asset-model-composite-model-id \  
  --cli-input-json file://stop-inference.json
```

3. Cree un archivo con el nombre `stop-inference.json` y agregue el siguiente código, sustituyendo lo siguiente:

- *asset-id* por el ID del activo de destino
- *action-definition-id* por el ID de la acción de inferencia inicial

```
{  
  "targetResource": {  
    "assetId": "asset-id"  
  },  
  "actionDefinitionId": "action-definition-Id",  
  "actionPayload": { "stringValue": "{\"Inference\":{\"inferenceMode\":\"STOP\"}}"}  
}
```

4. Ejecute el siguiente comando para detener la inferencia:

```
aws iotsitewise execute-action --cli-input-json file://stop-inference.json
```

Detección nativa de anomalías

AWS IoT SiteWise La detección nativa de anomalías es una función de aprendizaje automático (ML) para monitorear equipos industriales que detecta el comportamiento anormal de los equipos e identifica posibles fallas. Con la detección de anomalías nativa, puede implementar programas de mantenimiento predictivo e identificar los procesos de los equipos que no son óptimos.

AWS IoT SiteWise La detección nativa de anomalías no requiere amplios conocimientos ni experiencia en aprendizaje automático. Solo tiene que seleccionar las propiedades para preparar un modelo de aprendizaje automático personalizado que detecte los posibles fallos por usted. AWS IoT SiteWise La detección nativa de anomalías crea automáticamente el mejor modelo para conocer las condiciones normales de funcionamiento del equipo. El modelo está optimizado para detectar el comportamiento anormal del equipo que se haya producido en los datos históricos. Con la AWS IoT SiteWise consola o el SDK, puede ejecutar el modelo para procesar nuevos datos de series temporales de acuerdo con el cronograma deseado.

Para utilizar la detección de anomalías AWS IoT SiteWise nativa, haga lo siguiente:

- Selecciona las propiedades y el período de tiempo con el que te gustaría entrenar.
- Agregue los períodos de fallas históricas que se muestran en los datos (datos de etiqueta), si existen.
- Entrene su modelo de aprendizaje automático mediante la detección de anomalías AWS IoT SiteWise nativa. Si lo desea, puede configurar el reentrenamiento automático para mantener su modelo actualizado a lo largo del tiempo.
- Configure su programa de inferencias para probar sus flujos de datos en vivo con su modelo entrenado.

La detección de anomalías nativa monitorea los equipos industriales fijos y estacionarios que funcionan con una variabilidad limitada en las condiciones de operación. Los equipos compatibles incluyen maquinaria rotativa, como bombas, compresores, motores, máquinas de control numérico computarizado (CNC) y turbinas. Las aplicaciones de la industria de procesos incluyen intercambiadores de calor, calderas e inversores. La detección nativa de anomalías es un servicio de análisis interno que se integra en AWS IoT SiteWise los sistemas de mantenimiento existentes y los complementa.

Temas

- [Funciones nativas de detección de anomalías](#)
- [Requisitos previos](#)
- [Habilite la detección de anomalías en los sensores de un activo](#)
- [Habilite la detección de anomalías en los sensores de todos los activos](#)
- [Configuraciones de entrenamiento avanzadas](#)
- [Configuraciones de inferencia avanzadas](#)
- [Revise los resultados de la inferencia](#)
- [Activa acciones personalizadas en caso de comportamiento anómalo \(\)Consola de administración de AWS](#)
- [Prácticas recomendadas](#)

Funciones nativas de detección de anomalías

La AWS IoT SiteWise plataforma ofrece una gama de potentes funciones que pueden transformar sus operaciones industriales:

- **Mantenimiento predictivo:** detecte los fallos a tiempo y se integre con los sistemas de órdenes de trabajo para actuar en función de la información recopilada.
- **Estrecha integración con AWS IoT SiteWise:** aproveche la estrecha integración en su infraestructura existente y elimine la necesidad de trasladar sus datos a un servicio externo.
- **Selección automática de modelos y formación:** benefíciense de AWS IoT SiteWise este apoyo sin necesidad de conocimientos de aprendizaje automático.
- **Personalice su programación de inferencias:** programe los trabajos de inferencia para adaptarlos a sus necesidades operativas y a los horarios de los turnos.
- **Datos etiquetados:** mejore la precisión con intervalos de error conocidos durante el entrenamiento del modelo.
- **Evaluación del modelo con diagnósticos puntuales:** evalúe el rendimiento del modelo a nivel de evento.
- **Escalable en todos los activos:** cree modelos de cálculo para activos individuales o para varios activos para lograr la escalabilidad.
- **Información a nivel de sensores:** obtenga diagnósticos detallados que identifiquen los sensores específicos que contribuyen a una anomalía.

- Rentabilidad más rápida: pase del modelado de sensores a la detección de anomalías en cuestión de horas, no de semanas.

Requisitos previos

Para completar estos pasos debe tener un modelo de activos y haber creado al menos un activo. Para obtener más información, consulte [Crear un modelo de activo \(AWS CLI\)](#) y [Crear un activo \(AWS CLI\)](#). No admitimos productos externos IDs en este momento.

Si eres nuevo AWS IoT SiteWise (y no tienes datos históricos), debes llamar a la [CreateBulkImportJob](#) API para importar los valores de las propiedades de los activos AWS IoT SiteWise. Esto se usa para entrenar el modelo. Para obtener más información, consulte [Crear un trabajo de importación AWS IoT SiteWise masiva \(AWS CLI\)](#).

Configuración AWS CLI del modelo de cálculo APIs

Siga estos pasos para actualizar la AWS CLI configuración y acceder al modelo de cálculo: APIs

- Instale la última versión de awscli. `aws-cli`
- Verifique la instalación comprobando la nueva: APIs

```
aws iotsitewise help
```

El resultado del comando muestra la lista completa de las operaciones del modelo de cálculo recién agregadas AWS IoT SiteWise APIs, incluidas las que se acaban de añadir.

Requisitos de propiedad

Para configurar la detección de anomalías, debe cumplir los siguientes requisitos y los [UpdateAssetModel \(AWS CLI\)](#):

- Al menos una propiedad de entrada que sea de uno DOUBLE o varios tipos de INTEGER datos. Se trata de una propiedad de medición o de transformación y se utiliza para entrenar el modelo.
- Una propiedad de resultado del tipo de STRING datos. Debe ser una propiedad de medición y almacena los resultados de la detección de anomalías.

Requisitos previos de etiquetado

- Sube tus etiquetas de datos a un bucket de Amazon S3.
- Actualice la política de este depósito AWS IoT SiteWise para que pueda leer sus etiquetas.

En la consola, ve a Permisos -> Política de buckets. Sustituya el ARN de la cubeta por el ARN de la cubeta.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "SiteWiseS3ReadAccess",
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucketbucket-name",
        "arn:aws:s3:::amzn-s3-demo-bucketbucket-name/*"
      ]
    }
  ]
}
```

Requisitos previos para la evaluación del modelo

- La evaluación del modelo genera diagnósticos puntuales del modelo en la ubicación del bucket de Amazon S3 proporcionada por usted.
- Para que los resultados del diagnóstico puntual se escriban en su bucket de Amazon S3, actualice la política de bucket de este bucket AWS IoT SiteWise para permitir la escritura de los resultados.

En la consola, vaya a. Permissions -> Bucket policy Sustituya el ARN de la cubeta por el ARN de la cubeta.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "SiteWiseS3Access",
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:ListBucket",
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket",
        "arn:aws:s3:::amzn-s3-demo-bucket/*"
      ]
    }
  ]
}
```

Requisitos previos de las métricas del modelo

- Las métricas del modelo proporcionan información exhaustiva sobre el rendimiento, incluida la evaluación de la calidad del modelo, las estadísticas de detección de eventos y las métricas de comparación entre las versiones del modelo en la ubicación del bucket de Amazon S3 proporcionada.
- Para que el archivo json de métricas del modelo se escriba en su bucket de Amazon S3, actualice la política del bucket AWS IoT SiteWise para poder escribir los resultados.

En la consola, vaya a `Permissions -> Bucket policy`. Sustituya el ARN de la cubeta por el ARN de la cubeta.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "SiteWiseS3Access",
```

```

        "Effect": "Allow",
        "Action": [
            "s3:GetObject",
            "s3:ListBucket",
            "s3:PutObject"
        ],
        "Resource": [
            "arn:aws:s3:::amzn-s3-demo-bucket",
            "arn:aws:s3:::amzn-s3-demo-bucket-s3-bucket;/*"
        ]
    }
}
]
}

```

Habilite la detección de anomalías en los sensores de un activo

Crear un modelo de cálculo ()AWS CLI

Para crear un modelo de cálculo, utilice la interfaz de línea de AWS comandos ()AWS CLI. Tras definir el modelo de cálculo, entrene el modelo y programe la inferencia para detectar anomalías en un activo. AWS IoT SiteWise

- Cree un archivo `anomaly-detection-computation-model-payload.json` con el siguiente contenido:

```

{
  "computationModelName": "anomaly-detection-computation-model-name",
  "computationModelConfiguration": {
    "anomalyDetection": {
      "inputProperties": "${input_properties}",
      "resultProperty": "${result_property}"
    }
  },
  "computationModelDataBinding": {
    "input_properties": {
      "list": [{
        "assetModelProperty": {
          "assetModelId": "asset-model-id",
          "propertyId": "input-property-id-1"
        }
      ]
    }
  },

```

```

    {
      "assetModelProperty": {
        "assetModelId": "asset-model-id",
        "propertyId": "input-property-id-2"
      }
    }
  ],
  "result_property": {
    "assetModelProperty": {
      "assetModelId": "asset-model-id",
      "propertyId": "results-property-id"
    }
  }
}

```

- Ejecute el siguiente comando para crear un modelo de cálculo:

```

aws iotsitewise create-computation-model \
  --cli-input-json file://anomaly-detection-computation-model-payload.json

```

ExecuteAction Preparación de la carga útil de la API

Los siguientes pasos para ejecutar la formación y la inferencia se realizan con la [ExecuteActionAPI](#). Tanto el entrenamiento como la inferencia se configuran con una configuración de carga útil de acciones de JSON. Al invocar la [ExecuteActionAPI](#), la carga útil de la acción debe proporcionarse como un valor junto con una carga útil. `stringValue`

La carga útil debe cumplir estrictamente los requisitos de la API. En concreto, el valor debe ser una cadena plana, sin caracteres de control (por ejemplo, líneas nuevas, tabulaciones o devoluciones).

Las siguientes opciones proporcionan dos formas fiables de proporcionar una carga útil de acción válida:

Opción 1: utilice un archivo de carga limpio

El siguiente procedimiento describe los pasos para limpiar un archivo de carga útil:

1. Limpie el archivo para eliminar los caracteres de control.

```
tr -d '\n\r\t' < original-action-payload.json > training-or-inference-action-payload.json
```

2. Ejecute la acción con el archivo@=file://....

```
aws iotsitewise execute-action \
  --target-resource computationModelId=<MODEL_ID> \
  --action-definition-id <ACTION_DEFINITION_ID> \
  --resolve-to assetId=<ASSET_ID> \
  --action-payload stringValue=file://training-or-inference-action-payload.json
```

Opción 2: cadena en línea con comillas de escape

En los siguientes pasos se describen los pasos para suministrar la carga útil en línea y evitar los archivos intermediarios:

- Usa comillas dobles escapadas (\") dentro de la cadena JSON.
- Envuelva la stringValue=.. expresión completa entre comillas dobles.

Example de una carga útil de acción que se ha escapado:

```
aws iotsitewise execute-action \
  --target-resource computationModelId=<MODEL_ID> \
  --action-definition-id <ACTION_DEFINITION_ID> \
  --resolve-to assetId=<ASSET_ID> \
  --action-payload "stringValue={\"exportDataStartTime\":1717225200,
  \"exportDataEndTime\":1722789360,\"targetSamplingRate\": \"PT1M\"}"
```

Entrena al modelo ()AWS CLI


Una vez creado un modelo de cálculo, puede entrenar un modelo en función de los activos. Siga los pasos que se indican a continuación para entrenar un modelo para un activo:

1. Ejecute el siguiente comando para buscar el actionDefinitionId de la acción AWS/ANOMALY_DETECTION_TRAINING. computation-model-idSustitúyalo por el identificador devuelto en el paso anterior.

```
aws iotsitewise describe-computation-model \
```

```
--computation-model-id computation-model-id
```

2. Cree un archivo llamado `anomaly-detection-training-payload.json` y añada los siguientes valores:

 Note

La carga útil debe ajustarse a [Opción 1: utilice un archivo de carga limpio](#).

- a. `StartTime` con el inicio de los datos de entrenamiento, proporcionados en segundos por época.
- b. `EndTime` con el final de los datos de entrenamiento, proporcionados en segundos de época.
- c. Opcionalmente [Configuraciones de entrenamiento avanzadas](#), puede configurarlo para mejorar el rendimiento del modelo.
 - i. (Opcional) `TargetSamplingRate` con la frecuencia de muestreo de los datos.
 - ii. (Opcional) `LabelInputConfiguration` para especificar los períodos de tiempo en los que se produjo un comportamiento anómalo para mejorar el entrenamiento del modelo.
 - iii. (Opcional) `ModelEvaluationConfiguration` para evaluar el rendimiento del modelo mediante la ejecución de la inferencia en un intervalo de tiempo específico una vez finalizado el entrenamiento.
 - iv. (Opcional) `ModelMetricsDestination` para recopilar datos de rendimiento completos (precisión, recuperación, área bajo la curva).

```
{  
  "trainingMode": "TRAIN_MODEL",  
  "exportDataStartTime": StartTime,  
  "exportDataEndTime": EndTime  
}
```

Example de un ejemplo de carga útil de entrenamiento:

```
{  
  "trainingMode": "TRAIN_MODEL",  
  "exportDataStartTime": 1717225200,  
}
```

```
"exportDataEndTime": 1722789360
}
```

3. Ejecute el siguiente comando para iniciar el entrenamiento. Sustituya los siguientes parámetros del comando:
 - a. `computation-model-id` con el ID del modelo de cálculo objetivo.
 - b. `asset-id` con el ID del activo con el que entrenará el modelo.
 - c. `training-action-definition-id` con el ID de la AWS/ANOMALY_DETECTION_TRAINING acción del paso 1.

```
aws iotsitewise execute-action \
  --target-resource computationModelId=computation-model-id \
  --resolve-to assetId=asset-id \
  --action-definition-id training-action-definition-id \
  --action-payload stringValue@=file://anomaly-detection-training-payload.json
```

Example de una acción de ejecución:

```
aws iotsitewise execute-action --target-resource computationModelId=27cb824c-
fd84-45b0-946b-0a5b0466d890 --resolve-to assetId=cefd4b68-481b-4735-
b466-6a4220cd19ee --action-definition-id e54cea94-5d1c-4230-a59e-4f54dcbc972d --
action-payload stringValue@=file://anomaly-detection-training-payload.json
```

4. Ejecute el siguiente comando para comprobar el estado del proceso de entrenamiento del modelo. El resumen de ejecución más reciente muestra el estado de la ejecución (RUNNINGCOMPLETED//FAILED).

```
aws iotsitewise list-executions \
  --target-resource-type COMPUTATION_MODEL \
  --target-resource-id computation-model-id \
  --resolve-to-resource-type ASSET \
  --resolve-to-resource-id asset-id
```

5. Ejecute el siguiente comando para comprobar la configuración del último modelo entrenado. Este comando produce un resultado solo si al menos un modelo se entrenó correctamente.

```
aws iotsitewise describe-computation-model-execution-summary \
  --computation-model-id computation-model-id \
  --resolve-to-resource-type ASSET \
```

```
--resolve-to-resource-id asset-id
```

6. Cuando a `ComputationModel` esté en uso `AssetModelProperty`, utilice la [ListComputationModelResolveToResources](#) API para identificar los activos con las acciones ejecutadas.

```
aws iotsitewise list-computation-model-resolve-to-resources \  
  --computation-model-id computation-model-id
```

Inicie y detenga el reentrenamiento del modelo ()AWS CLI

Tras el entrenamiento inicial del modelo, puede configurar el reentrenamiento automático para abordar la desviación de los datos y mantener la precisión del modelo a lo largo del tiempo. El programador de reentrenamiento le permite configurar actualizaciones periódicas del modelo con modos de promoción configurables.

Inicie el programador de reentrenamiento

1. Ejecute el siguiente comando para buscar el `actionDefinitionId` de la acción AWS/ANOMALY_DETECTION_TRAINING. `computation-model-id` Sustitúyalo por el ID devuelto al crear el modelo de cálculo.

```
aws iotsitewise describe-computation-model \  
  --computation-model-id computation-model-id
```

2. Cree un archivo llamado `anomaly-detection-start-retraining-payload.json` y añada el siguiente código. Sustituya los parámetros por valores tal y como se describe.

Note

La carga útil debe ajustarse a [Opción 1: utilice un archivo de carga limpio](#).

- a. `lookbackWindow` con la ventana de datos históricos para utilizarla en el reentrenamiento (P180D//P360DP540D/P720D).
- b. `retrainingFrequency` con qué frecuencia se debe volver a entrenar el modelo (mínima P30D, máxima P1Y).

- c. (Opcional) `promotion` con el modo de promoción del modelo (`SERVICE_MANAGED`o`CUSTOMER_MANAGED`). El valor predeterminado es `SERVICE_MANAGED`.
- d. (Opcional) `retrainingStartDate` con la fecha de inicio del programa de reentrenamiento, indicada en segundos por época. Trunca la hora hasta el día UTC más cercano. Opcional, el valor predeterminado es la fecha actual.
- e. Si lo desea, puede configurarlo [Configuraciones de entrenamiento avanzadas](#) para mejorar el rendimiento del modelo.
 - (Opcional) `ModelMetricsDestination` para obtener datos de rendimiento completos (precisión, recuperación, área bajo la curva).

```
{
  "trainingMode": "START_RETRAINING_SCHEDULER",
  "retrainingConfiguration": {
    "lookbackWindow": "P180D",
    "promotion": "SERVICE_MANAGED",
    "retrainingFrequency": "P30D",
    "retrainingStartDate": "StartDate"
  }
}
```

3. Ejecute el siguiente comando para iniciar el programador de reentrenamiento. Sustituya los siguientes parámetros del comando:
 - a. `computation-model-id` con el ID del modelo de cálculo objetivo.
 - b. `asset-id` con el ID del activo con el que entrenará el modelo.
 - c. `training-action-definition-id` con el ID de la AWS/`ANOMALY_DETECTION_TRAINING` acción del paso 1.

```
aws iotsitewise execute-action \
  --target-resource computationModelId=computation-model-id \
  --resolve-to assetId=asset-id \
  --action-definition-id training-action-definition-id \
  --action-payload stringValue@=file://anomaly-detection-start-retraining-
payload.json
```

Example Ejemplo del comando de ejecución de la acción

```
aws iotsitewise execute-action --target-resource computationModelId=27cb824c-  
fd84-45b0-946b-0a5b0466d890 --resolve-to assetId=cefd4b68-481b-4735-  
b466-6a4220cd19ee --action-definition-id e54cea94-5d1c-4230-a59e-4f54dcbc972d --  
action-payload stringValue=@file://anomaly-detection-start-retraining-payload.json
```

Detenga el programador de reentrenamiento

1. Ejecute el siguiente comando para buscar el `actionDefinitionId` de la acción AWS/ANOMALY_DETECTION_TRAINING. `computation-model-id` Sustitúyalo por el ID real del modelo de cálculo creado anteriormente.

```
aws iotsitewise describe-computation-model \  
--computation-model-id computation-model-id
```

2. Cree un archivo `anomaly-detection-stop-retraining-payload.json` y añada lo siguiente:

Note

La carga útil debe ajustarse a [Opción 1: utilice un archivo de carga limpio](#).

```
{  
  "trainingMode": "STOP_RETRAINING_SCHEDULER"  
}
```

3. Ejecute el siguiente comando para detener el programador de reentrenamiento. Sustituya los siguientes parámetros del comando:
 - a. `computation-model-id` con el ID del modelo de cálculo objetivo.
 - b. `asset-id` con el ID del activo con el que entrenará el modelo.
 - c. `training-action-definition-id` con el ID de la AWS/ANOMALY_DETECTION_TRAINING acción del paso 1.

```
aws iotsitewise execute-action \  
  --target-resource computationModelId=computation-model-id \  
  --resolve-to assetId=asset-id \  
  --action-definition-id training-action-definition-id \  
  --action-payload stringValue@=file://anomaly-detection-stop-retraining-  
payload.json
```

Iniciar y detener la inferencia ()AWS CLI

Después de entrenar el modelo, inicie la inferencia. Esto indica AWS IoT SiteWise que debe monitorear activamente sus activos industriales para detectar anomalías.

Comience la inferencia

1. Ejecute el siguiente comando para buscar el `actionDefinitionId` de la acción AWS/ANOMALY_DETECTION_INFERENCE. `computation-model-id` Sustitúyalo por el identificador real del modelo de cálculo creado anteriormente.

```
aws iotsitewise describe-computation-model \  
  --computation-model-id computation-model-id
```

2. Cree un archivo `anomaly-detection-start-inference-payload.json` y añada los siguientes valores:

Note

La carga útil debe ajustarse a [Opción 1: utilice un archivo de carga limpio](#).

```
"inferenceMode": "START",  
"dataUploadFrequency": "DataUploadFrequency"
```

- a. `DataUploadFrequency`: Configure la frecuencia con la que se ejecuta el programa de inferencias para realizar la detección de anomalías. Los valores permitidos son: PT5M, PT10M, PT15M, PT30M, PT1H, PT2H..PT12H, PT1D.

- b. (Opcional) `DataDelayOffsetInMinutes` con el retardo compensado en minutos. Establezca este valor entre 0 y 60 minutos.
 - c. (Opcional) `TargetModelVersion` con la versión del modelo que se va a activar.
 - d. (Opcional) Configure la configuración `weeklyOperatingWindow` con una configuración de turnos.
 - e. Opcionalmente, puede configurarlo [Configuraciones de inferencia avanzadas](#).
 - i. [Inferencia de alta frecuencia \(de 5 minutos a 1 hora\)](#).
 - ii. [Inferencia de baja frecuencia \(2 horas — 1 día\)](#).
 - iii. [Programación flexible](#).
3. Ejecute el siguiente comando para iniciar la inferencia. Sustituya los siguientes parámetros en el archivo de carga útil.
- a. `computation-model-id` con el ID del modelo de cálculo objetivo.
 - b. `asset-id` con el ID del activo con el que se entrenó el modelo.
 - c. `inference-action-definition-id` con el ID de la AWS/ANOMALY_DETECTION_INFERENCE acción del paso 1.

```
aws iotsitewise execute-action \
  --target-resource computationModelId=computation-model-id \
  --resolve-to assetId=asset-id \
  --action-definition-id inference-action-definition-id \
  --action-payload stringValue@=file://anomaly-detection-inference-payload.json
```

4. Ejecute el siguiente comando para comprobar si la inferencia sigue ejecutándose. El `inferenceTimerActive` campo se establece TRUE cuando la inferencia está activa.

```
aws iotsitewise describe-computation-model-execution-summary \
  --computation-model-id computation-model-id \
  --resolve-to-resource-type ASSET \
  --resolve-to-resource-id asset-id
```

5. El siguiente comando muestra todas las ejecuciones de inferencias:

```
aws iotsitewise list-executions \
  --target-resource-type COMPUTATION_MODEL \
  --target-resource-id computation-model-id \
```

```
--resolve-to-resource-type ASSET \  
--resolve-to-resource-id asset-id
```

6. Ejecute el siguiente comando para describir una ejecución individual. `execution-id` Sustitúyalo por el identificador del paso 5 anterior.

```
aws iotsitewise describe-execution \  
--execution-id execution-id
```

Detenga la inferencia

1. Ejecute el siguiente comando para buscar el `actionDefinitionId` de la acción AWS/ANOMALY_DETECTION_INFERENCE. `computation-model-id` Sustitúyalo por el identificador real del modelo de cálculo creado anteriormente.

```
aws iotsitewise describe-computation-model \  
--computation-model-id computation-model-id
```

2. Cree un archivo `anomaly-detection-stop-inference-payload.json` y añada el siguiente código.

```
{  
  "inferenceMode": "STOP"  
}
```

Note

La carga útil debe ajustarse a [Opción 1: utilice un archivo de carga limpio](#).

3. Ejecute el siguiente comando para detener la inferencia. Sustituya el siguiente parámetro en el archivo de carga útil:
 - a. `computation-model-id` con el ID del modelo de cálculo de destino.
 - b. `asset-id` con el ID del activo con el que se entrenó el modelo.
 - c. `inference-action-definition-id` con el ID de la AWS/ANOMALY_DETECTION_INFERENCE acción del paso 1.

Example del comando stop inference:

```
aws iotsitewise execute-action \  
  --target-resource computationModelId=computation-model-id \  
  --resolve-to assetId=asset-id \  
  --action-definition-id inference-action-definition-id \  
  --action-payload stringValue@=file://anomaly-detection-stop-inference-  
payload.json
```

Encuentre modelos de cálculo que utilicen un recurso determinado en el enlace de datos

Para enumerar los modelos de cálculo que están vinculados a un recurso determinado:

- modelo de activos (busque todos los modelos de cálculo en los que esté vinculada alguna de las propiedades de este modelo de activos).
- activo (busque todos los modelos de cálculo en los que esté vinculada alguna de las propiedades de este activo)
- propiedad del modelo de activos (busca todos los modelos de cálculo en los que está vinculada esta propiedad)
- propiedad de activo (busca todos los modelos de cálculo a los que está vinculada esta propiedad). Esto puede ser con fines informativos o ser necesario cuando el usuario intenta vincular esta propiedad a otro modelo de cálculo (pero ya está enlazada a otro lugar)

Utilice la [ListComputationModelDataBindingUsagesAPI](#) para obtener una lista de `ComputationModelId`s que incluyan el activo (propiedad) o el modelo de activo (propiedad) como enlace de datos.

Prepare una `request.json` con la siguiente información:

```
{  
  "dataBindingValueFilter": {  
    "asset": {  
      "assetId": "<string>"  
    }  
  }  
  // OR
```

```
"assetModel": {
  "assetModelId": "<string>"
}
// OR
"assetProperty": {
  "assetId": "<string>",
  "propertyId": "<string>"
}
// OR
"assetModelProperty": {
  "assetModelId": "<string>",
  "propertyId": "<string>"
}
},
"nextToken": "<string>",
"maxResults": "<number>"
}
```

Utilice el `list-computation-model-data-binding-usages` comando para recuperar los modelos con activos o los modelos de activos como enlaces de datos.

```
aws iotsitewise list-computation-model-data-binding-usages \
--cli-input-json file://request.json
```

Habilite la detección de anomalías en los sensores de todos los activos


Crear un modelo de cálculo (AWS CLI)

Para crear un modelo de cálculo, utilice la interfaz de línea de AWS comandos (AWS CLI). Tras definir el modelo de cálculo, entrénelo y programe la inferencia para detectar anomalías en todos los activos. AWS IoT SiteWise

Los siguientes pasos explican este proceso:

1. Para configurar la detección de anomalías, utilice [UpdateAssetModel\(AWS CLI\)](#) y cumpla los siguientes requisitos:

- a. Al menos una propiedad de entrada que sea de uno DOUBLE o varios tipos de INTEGER datos. Se trata de una propiedad de medición o de transformación y se utiliza para entrenar el modelo.
 - b. Una propiedad de resultado del tipo de STRING datos. Debe ser una propiedad de medición y almacena los resultados de la detección de anomalías.
2. Cree un archivo `anomaly-detection-computation-model-payload.json` con el siguiente contenido:

 Note

Cree un modelo de cálculo proporcionándolo directamente `assetProperty` como fuente de datos.

```
{
  "computationModelName": "name of ComputationModel",
  "computationModelConfiguration": {
    "anomalyDetection": {
      "inputProperties": "${properties}",
      "resultProperty": "${p3}"
    }
  },
  "computationModelDataBinding": {
    "properties": {
      "list": [
        {
          "assetProperty": {
            "assetId": "asset-id",
            "propertyId": "input-property-id-1"
          }
        },
        {
          "assetProperty": {
            "assetId": "asset-id",
            "propertyId": "input-property-id-2"
          }
        }
      ]
    }
  },
  "p3": {
```

```

        "assetProperty": {
            "assetId": "asset-id",
            "propertyId": "results-property-id"
        }
    }
}

```

3. Ejecute el siguiente comando para crear un modelo de cálculo:

```

aws iotsitewise create-computation-model \
    --cli-input-json file://anomaly-detection-computation-model-payload.json

```

ExecuteAction Preparación de la carga útil de la API

Los siguientes pasos para ejecutar la formación y la inferencia se realizan con la [ExecuteAction](#) API. Tanto el entrenamiento como la inferencia se configuran con una configuración de carga útil de acciones de JSON. Al invocar la [ExecuteAction](#) API, la carga útil de la acción debe proporcionarse como un valor junto con una carga útil. `stringValue`

La carga útil debe cumplir estrictamente los requisitos de la API. En concreto, el valor debe ser una cadena plana sin caracteres de control (por ejemplo, líneas nuevas, tabulaciones o devoluciones). Las siguientes opciones proporcionan dos formas fiables de proporcionar una carga útil de acción válida.

Opción 1: utilice un archivo de carga limpio

El siguiente procedimiento describe los pasos para limpiar un archivo de carga útil:

1. Limpie el archivo para eliminar los caracteres de control.

```

tr -d '\n\r\t' < original-action-payload.json > training-or-inference-action-payload.json

```

2. Ejecute la acción con el archivo `@=file://...`

```

aws iotsitewise execute-action \
    --target-resource computationModelId=<MODEL_ID> \
    --action-definition-id <ACTION_DEFINITION_ID> \
    --action-payload stringValue@=file://training-or-inference-action-payload.json

```

Opción 2: cadena en línea con comillas de escape

En los siguientes pasos se describen los pasos para suministrar la carga útil en línea y evitar los archivos intermediarios:

- Usa comillas dobles escapadas (\") dentro de la cadena JSON.
- Envuelva la `StringValue=..` expresión completa entre comillas dobles.

Example de una carga útil de acción que se ha escapado:

```
aws iotsitewise execute-action \  
  --target-resource computationModelId=<MODEL_ID> \  
  --action-definition-id <ACTION_DEFINITION_ID> \  
  --action-payload "stringValue={\"exportDataStartTime\":1717225200,  
\"exportDataEndTime\":1722789360,\"targetSamplingRate\": \"PT1M\"}"
```

Entrena al modelo ()AWS CLI

1. Ejecute el siguiente comando para buscar el `actionDefinitionId` de la acción AWS/ANOMALY_DETECTION_TRAINING. `computation-model-id` Sustitúyalo por el identificador devuelto en el paso anterior.

```
aws iotsitewise describe-computation-model \  
  --computation-model-id computation-model-id
```

2. Cree un archivo llamado `anomaly-detection-training-payload.json` y añada los siguientes valores:

Note

La carga útil debe ajustarse a [Opción 1: utilice un archivo de carga limpio](#).

- a. `StartTime` con el inicio de los datos de entrenamiento, proporcionados en segundos por época.
- b. `EndTime` con el final de los datos de entrenamiento, proporcionados en segundos de época.
- c. Puede [Configuraciones de inferencia avanzadas](#) configurarlo de forma opcional.

- i. (Opcional) `TargetSamplingRate` con la frecuencia de muestreo de los datos.
- ii. (Opcional) `LabelInputConfiguration` para especificar los períodos de tiempo en los que se produjo un comportamiento anómalo para mejorar el entrenamiento del modelo.
- iii. (Opcional) `ModelEvaluationConfiguration` para evaluar el rendimiento del modelo mediante la ejecución de la inferencia en un intervalo de tiempo específico una vez finalizado el entrenamiento.

```
{
  "exportDataStartTime": StartTime,
  "exportDataEndTime": EndTime
}
```

Example de un ejemplo de carga útil de entrenamiento:

```
{
  "exportDataStartTime": 1717225200,
  "exportDataEndTime": 1722789360
}
```

3. Ejecute el siguiente comando para iniciar el entrenamiento (sin proporcionar un activo como recurso de destino). Sustituya los siguientes parámetros del comando:

```
aws iotsitewise execute-action \
  --target-resource computationModelId=computation-model-id \
  --action-definition-id training-action-definition-id \
  --action-payload stringValue@=file://anomaly-detection-training-payload.json
```

4. Ejecute el siguiente comando para comprobar el estado del proceso de entrenamiento del modelo. El resumen de ejecución más reciente muestra el estado de la ejecución (RUNNINGCOMPLETED//FAILED).

```
aws iotsitewise list-executions \
  --target-resource-type COMPUTATION_MODEL \
  --target-resource-id computation-model-id
```

5. Ejecute el siguiente comando para comprobar la configuración del último modelo entrenado. Este comando produce un resultado solo si al menos un modelo ha completado el entrenamiento correctamente.

```
aws iotsitewise describe-computation-model-execution-summary \  
  --computation-model-id computation-model-id
```

Inicie y detenga el reentrenamiento del modelo ()AWS CLI

Tras el entrenamiento inicial del modelo, puede configurar el reentrenamiento automático para abordar la desviación de los datos y mantener la precisión del modelo a lo largo del tiempo. El programador de reentrenamiento le permite configurar actualizaciones periódicas del modelo con modos de promoción configurables.

Inicie el programador de reentrenamiento

1. Prepare la misma carga útil que se menciona en [Inicie el programador de reentrenamiento](#)
2. Ejecute la acción de entrenamiento (sin proporcionar un activo como recurso objetivo). Sustituya los siguientes parámetros del comando:
 - a. `computation-model-id` con el ID del modelo de cálculo objetivo.
 - b. `training-action-definition-id` con el ID de la AWS/ANOMALY_DETECTION_TRAINING acción.

```
aws iotsitewise execute-action \  
  --target-resource computationModelId=computation-model-id \  
  --action-definition-id training-action-definition-id \  
  --action-payload stringValue@=file://anomaly-detection-start-retraining-  
  payload.json
```

3. Ejecute el siguiente comando para comprobar el estado del proceso del programador de inicio de reentrenamiento. El resumen de ejecución más reciente muestra el estado de la ejecución (RUNNING/COMPLETED/FAILED).

```
aws iotsitewise list-executions \  
  --target-resource-type COMPUTATION_MODEL \  
  --target-resource-id computation-model-id
```

Detenga el programador de reentrenamiento

1. Prepare la misma carga útil que se menciona en [Detenga el programador de reentrenamiento](#)
2. Ejecute la acción de entrenamiento (sin proporcionar un activo como recurso objetivo). Sustituya los siguientes parámetros del comando:
 - a. `computation-model-id` con el ID del modelo de cálculo objetivo.
 - b. `training-action-definition-id` con el ID de la AWS/ANOMALY_DETECTION_TRAINING acción.

```
aws iotsitewise execute-action \  
  --target-resource computationModelId=computation-model-id \  
  --action-definition-id training-action-definition-id \  
  --action-payload stringValue@=file://anomaly-detection-stop-retraining-  
  payload.json
```

3. Ejecute el siguiente comando para comprobar el estado del proceso de detener el programador de reentrenamiento. El resumen de ejecución más reciente muestra el estado de la ejecución (RUNNING/COMPLETED/FAILED).

```
aws iotsitewise list-executions \  
  --target-resource-type COMPUTATION_MODEL \  
  --target-resource-id computation-model-id
```

Iniciar y detener la inferencia (AWS CLI)


Después de entrenar el modelo, inicie la inferencia, que indica que debe comenzar AWS IoT SiteWise a monitorear sus activos industriales para detectar anomalías.

Inicie la inferencia

1. Ejecute el siguiente comando para buscar el `actionDefinitionId` de la acción AWS/ANOMALY_DETECTION_INFERENCE. `computation-model-id` Sustitúyalo por el identificador real del modelo de cálculo creado anteriormente.

```
aws iotsitewise describe-computation-model \  
  --computation-model-id computation-model-id
```

2. Cree un archivo `anomaly-detection-start-inference-payload.json` y añada el siguiente código. Sustituya los siguientes parámetros tal y como se describe:

 Note

La carga útil debe ajustarse a [Opción 1: utilice un archivo de carga limpio](#).

- a. `DataUploadFrequency`: Configure la frecuencia con la que se ejecuta el programa de inferencias para realizar la detección de anomalías. Los valores permitidos son: PT5M, PT10M, PT15M, PT30M, PT1H, PT2H..PT12H, PT1D.

```
"inferenceMode": "START",
"dataUploadFrequency": "DataUploadFrequency"
```

- b. (Opcional) `DataDelayOffsetInMinutes` con el retardo compensado en minutos. Establezca este valor entre 0 y 60 minutos.
 - c. (Opcional) `TargetModelVersion` con la versión del modelo que se va a activar.
 - d. (Opcional) Configure la configuración `weeklyOperatingWindow` con una configuración de turnos.
 - e. Opcionalmente, puede configurarlo [Configuraciones de inferencia avanzadas](#).
 - i. [Inferencia de alta frecuencia \(de 5 minutos a 1 hora\)](#).
 - ii. [Inferencia de baja frecuencia \(2 horas — 1 día\)](#).
 - iii. [Programación flexible](#).
3. Ejecute el siguiente comando para iniciar la inferencia. Sustituya los siguientes parámetros en el archivo de carga útil.
 - a. `computation-model-id` con el ID del modelo de cálculo objetivo.
 - b. `inference-action-definition-id` con el ID de la AWS/ANOMALY_DETECTION_INFERENCE acción del paso 1.

```
aws iotsitewise execute-action \
  --target-resource computationModelId=computation-model-id \
  --action-definition-id inference-action-definition-id \
  --action-payload stringValue@=file://anomaly-detection-inference-payload.json
```

4. Ejecute el siguiente comando para comprobar si la inferencia sigue ejecutándose. El `inferenceTimerActive` campo se establece `TRUE` cuando la inferencia está activa.

```
aws iotsitewise describe-computation-model-execution-summary \  
  --computation-model-id computation-model-id
```

5. El siguiente comando muestra todas las ejecuciones de inferencias:

```
aws iotsitewise list-executions \  
  --target-resource-type COMPUTATION_MODEL \  
  --target-resource-id computation-model-id
```

6. Ejecute el siguiente comando para describir una ejecución individual. `execution-id` Sustitúyalo por el identificador del paso 5 anterior.

```
aws iotsitewise describe-execution \  
  --execution-id execution-id
```

Detenga la inferencia

1. Ejecute el siguiente comando para buscar el `actionDefinitionId` de la acción `AWS/ANOMALY_DETECTION_INFERENCE`. `computation-model-id` Sustitúyalo por el identificador real del modelo de cálculo creado anteriormente.

```
aws iotsitewise describe-computation-model \  
  --computation-model-id computation-model-id
```

2. Cree un archivo `anomaly-detection-stop-inference-payload.json` y añada el siguiente código.

```
{  
  "inferenceMode": "STOP"  
}
```

Note

La carga útil debe ajustarse a [Opción 1: utilice un archivo de carga limpio](#).

3. Ejecute el siguiente comando para detener la inferencia. Sustituya el siguiente parámetro en el archivo de carga útil:
 - a. `computation-model-id` con el ID del modelo de cálculo de destino.
 - b. `inference-action-definition-id` con el ID de la AWS/ANOMALY_DETECTION_INFERENCE acción del paso 1.

Example del comando stop inference:

```
aws iotsitewise execute-action \  
--target-resource computationModelId=computation-model-id \  
--action-definition-id inference-action-definition-id \  
--action-payload stringValue@=file://anomaly-detection-stop-inference-payload.json
```

Configuraciones de entrenamiento avanzadas

configuración de frecuencia de muestreo

La frecuencia de muestreo define la frecuencia con la que se registran las lecturas del sensor (por ejemplo, una vez cada segundo o una vez cada minuto). Esta configuración afecta directamente a la granularidad de los datos de entrenamiento e influye en la capacidad del modelo para capturar las variaciones a corto plazo en el comportamiento de los sensores.

[Toma de muestras para obtener datos de alta frecuencia y garantizar la coherencia entre el entrenamiento y la inferencia](#) Visítelo para obtener información sobre las mejores prácticas.

Configure la frecuencia de muestreo objetivo

Si lo desea, puede especificar una `TargetSamplingRate` en su configuración de entrenamiento para controlar la frecuencia con la que se muestrean los datos. Los valores admitidos son:

```
PT1S | PT5S | PT10S | PT15S | PT30S | PT1M | PT5M | PT10M | PT15M | PT30M | PT1H
```

Estos son los formatos de duración ISO 8601, que representan los siguientes formatos de tiempo:

- PT1S= 1 segundo
- PT1M= 1 minuto

- PT1H= 1 hora

Elija una frecuencia de muestreo que logre el equilibrio adecuado entre la resolución de los datos y la eficiencia del entrenamiento. Están disponibles las siguientes tasas:

- Las frecuencias de muestreo más altas (PT1S) ofrecen detalles más precisos, pero pueden aumentar el volumen de datos y el tiempo de entrenamiento.
- Las frecuencias de muestreo más bajas (PT10M,PT1H) reducen el tamaño y el costo de los datos, pero pueden pasar por alto anomalías efímeras.

Manejar la desalineación de las marcas de tiempo

AWS IoT SiteWise compensa automáticamente la desalineación de las marcas de tiempo en varios flujos de datos durante el entrenamiento. Esto garantiza un comportamiento uniforme del modelo, incluso si las señales de entrada no están perfectamente alineadas en el tiempo.

[Toma de muestras para obtener datos de alta frecuencia y garantizar la coherencia entre el entrenamiento y la inferencia](#) Visítelo para obtener información sobre las mejores prácticas.

Habilite el muestreo

Añada el siguiente código `anomaly-detection-training-payload.json`.

Configure el muestreo añadiendo `TargetSamplingRate` la carga útil de la acción de entrenamiento, con la frecuencia de muestreo de los datos. Los valores permitidos son:PT1S | PT5S | PT10S | PT15S | PT30S | PT1M | PT5M | PT10M | PT15M | PT30M | PT1H.

```
{
  "exportDataStartTime": StartTime,
  "exportDataEndTime": EndTime,
  "targetSamplingRate": "TargetSamplingRate"
}
```

Example de una configuración de frecuencia de muestreo:

```
{
  "exportDataStartTime": 1717225200,
  "exportDataEndTime": 1722789360,
  "targetSamplingRate": "PT1M"
}
```

}

Etiquete sus datos

Al etiquetar los datos, debe definir intervalos de tiempo que representen períodos de comportamiento anormal del equipo. Esta información de etiquetado se proporciona como un CSV archivo, en el que cada fila especifica un intervalo de tiempo durante el cual el equipo no funcionó correctamente.

Cada fila contiene dos marcas de tiempo:

- La hora de inicio, que indica cuándo se cree que ha comenzado un comportamiento anormal.
- La hora de finalización, que representa el momento en que se observó por primera vez la falla o el problema.

Este archivo CSV se almacena en un bucket de Amazon S3 y se utiliza durante el entrenamiento de modelos para ayudar al sistema a aprender de ejemplos conocidos de comportamiento anormal. En el siguiente ejemplo, se muestra cómo deben aparecer los datos de la etiqueta como un .csv archivo. El archivo no tiene encabezado.

Example de un archivo CSV:

```
2024-06-21T00:00:00.000000,2024-06-21T12:00:00.000000
2024-07-11T00:00:00.000000,2024-07-11T12:00:00.000000
2024-07-31T00:00:00.000000,2024-07-31T12:00:00.000000
```

La fila 1 representa un evento de mantenimiento ocurrido el 21 de junio de 2024, con un período de 12 horas (de 2024-06-21T00:00:00.000000Z a2024-06-21T12:00:00.000000Z) AWS IoT SiteWise para detectar un comportamiento anormal.

La fila 2 representa un evento de mantenimiento ocurrido el 11 de julio de 2024, con un intervalo de 12 horas (de 2024-07-11T00:00:00.000000Z a2024-07-11T12:00:00.000000Z) AWS IoT SiteWise para detectar un comportamiento anormal.

La fila 3 representa un evento de mantenimiento ocurrido el 31 de julio de 2024, con un intervalo de 12 horas (de 2024-07-31T00:00:00.000000Z a2024-07-31T12:00:00.000000Z) AWS IoT SiteWise para detectar un comportamiento anormal.

AWS IoT SiteWise utiliza todos estos intervalos de tiempo para entrenar y evaluar modelos que puedan identificar un comportamiento anormal en torno a estos eventos. Tenga en cuenta que no

todos los eventos son detectables y que los resultados dependen en gran medida de la calidad y las características de los datos subyacentes.

Para obtener más información sobre las mejores prácticas de muestreo, consulte [Prácticas recomendadas](#).

Pasos para el etiquetado de datos

- Configure su depósito de Amazon S3 de acuerdo con los requisitos previos de etiquetado en Requisitos previos [de etiquetado de datos](#).
- Cargue el archivo en su balde de etiquetado.
- Añada lo siguiente `aanomaly-detection-training-payload.json`.
 - Proporcione las ubicaciones en la `labelInputConfiguration` sección del archivo. `label-bucket` Sustitúyalo por el nombre del depósito y `files-prefix` por la ruta del archivo o cualquier parte del prefijo. Todos los archivos de la ubicación se analizan y (en caso de éxito) se utilizan como archivos de etiquetas.

```
{
  "exportDataStartTime": StartTime,
  "exportDataEndTime": EndTime,
  "labelInputConfiguration":
    {
      "bucketName": "label-bucket",
      "prefix": "files-prefix"
    }
}
```

Example de una configuración de etiquetas:

```
{
  "exportDataStartTime": 1717225200,
  "exportDataEndTime": 1722789360,
  "labelInputConfiguration": {
    "bucketName": "anomaly-detection-customer-data-278129555252-iad",
    "prefix": "Labels/model=b2d8ab3e-73af-48d8-9b8f-a290bef931b4/asset[d3347728-4796-4c5c-afdb-ea2f551ffe7a]/Lables.csv"
  }
}
```

Evaluar el modelo

El diagnóstico puntual de un modelo de AWS IoT SiteWise entrenamiento es una evaluación del rendimiento del modelo en los eventos individuales. Durante el entrenamiento, AWS IoT SiteWise genera una puntuación de anomalías y un diagnóstico de la contribución del sensor para cada fila del conjunto de datos de entrada. Una puntuación de anomalía más alta indica una mayor probabilidad de que se produzca un evento anormal.

Los diagnósticos puntuales están disponibles cuando se entrena un modelo con la [ExecuteAction](#) API y el tipo de AWS/ANOMALY_DETECTION_TRAINING acción.

Para configurar la evaluación del modelo,

- Configure su depósito de Amazon S3 de acuerdo con los requisitos previos de etiquetado en [Requisitos previos de etiquetado de datos](#).
- Añada lo siguiente a `anomaly-detection-training-payload.json`
 - Proporcione los datos `evaluationStartTime` y `evaluationEndTime` (ambos en segundos de época) en la ventana utilizada para evaluar el rendimiento del modelo.
 - Proporcione la ubicación del depósito de Amazon S3 (`resultDestination`) para que se escriban los diagnósticos de evaluación.

Note

El intervalo de evaluación del modelo (`dataStartTime` a `dataEndTime`) debe superponerse o ser contiguo al intervalo de entrenamiento. No se permiten espacios vacíos.

```
{
  "exportDataStartTime": StartTime,
  "exportDataEndTime": EndTime,
  "modelEvaluationConfiguration": {
    "dataStartTime": evaluationStartTime,
    "dataEndTime": evaluationEndTime
    "resultDestination": {
      "bucketName": "s3BucketName",
      "prefix": "bucketPrefix"
    }
  }
}
```

```
}
```

Example de una configuración de evaluación de modelos:

```
{
  "exportDataStartTime": 1717225200,
  "exportDataEndTime": 1722789360,
  "modelEvaluationConfiguration": {
    "dataStartTime": 1722789360,
    "dataEndTime": 1725174000,
    "resultDestination": {
      "bucketName": "anomaly-detection-customer-data-278129555252-iad",
      "prefix": "Evaluation/asset[d3347728-4796-4c5c-afdb-ea2f551ffe7a]/1747681026-
evaluation_results.jsonl"
    }
  }
}
```

Genere métricas del modelo

Las métricas de los modelos proporcionan información exhaustiva sobre el rendimiento y la calidad de los modelos de detección de anomalías entrenados. El proceso de formación genera automáticamente estas métricas y las publica en el bucket de Amazon S3 especificado, lo que permite acceder a ellas fácilmente para su análisis, comparación de modelos y decisiones de promoción en los flujos de trabajo de reentrenamiento.

Comprender las métricas de los modelos

El proceso de formación genera automáticamente métricas del modelo y proporciona información detallada sobre:

- Rendimiento del modelo: medidas cuantitativas como la precisión, la recuperación y el AUC cuando hay datos etiquetados disponibles
- Calidad de los datos: información sobre los datos de entrenamiento utilizados y los períodos de tiempo cubiertos
- Detección de eventos: estadísticas sobre las anomalías identificadas y los eventos etiquetados
- Comparación de modelos: métricas de comparación entre diferentes versiones del modelo durante el reentrenamiento

Configure el destino de las métricas del modelo

Para habilitar la generación de métricas del modelo, configure un destino de Amazon S3 donde se publiquen las métricas.

1. Configure su bucket de Amazon S3 según la [Requisitos previos para la evaluación del modelo](#).
2. Añada lo siguiente a la carga útil de sus acciones de entrenamiento para especificar dónde deben almacenarse las métricas del modelo:

```
{
  "trainingMode": "TRAIN_MODEL",
  "exportDataStartTime": StartTime,
  "exportDataEndTime": EndTime,
  "modelMetricsDestination": {
    "bucketName": "bucket-name",
    "prefix": "prefix"
  }
}
```

Example Ejemplo de configuración de métricas del modelo

```
{
  "exportDataStartTime": 1717225200,
  "exportDataEndTime": 1722789360,
  "modelMetricsDestination": {
    "bucketName": "anomaly-detection-metrics-bucket-123456789012-iad",
    "prefix": "ModelMetrics/computation-model-id/asset-id/training-
metrics.json"
  }
}
```

Configure las métricas del modelo para el reentrenamiento

Al configurar los programas de reentrenamiento, se requiere el destino de las métricas del modelo para permitir un seguimiento y una comparación exhaustivos del rendimiento del modelo:

```
{
  "trainingMode": "START_RETRAINING_SCHEDULER",
  "modelMetricsDestination": {
    "bucketName": "bucket-name",
```

```
    "prefix": "prefix"
  },
  "retrainingConfiguration": {
    "lookbackWindow": "P180D",
    "promotion": "SERVICE_MANAGED",
    "retrainingFrequency": "P30D",
    "retrainingStartDate": "StartDate"
  }
}
```

Parameters

bucketName

Depósito de Amazon S3 donde se almacenarán las métricas del modelo

prefix

Amazon S3 prefix/path para organizar archivos de métricas de modelos

Estructura de métricas del modelo

Las métricas del modelo se almacenan como archivos JSON en su bucket de Amazon S3 con la siguiente estructura:

```
{
  "labeled_ranges": [],
  "labeled_event_metrics": {
    "num_labeled": 0,
    "num_identified": 0,
    "total_warning_time_in_seconds": 0
  },
  "predicted_ranges": [],
  "unknown_event_metrics": {
    "num_identified": 0,
    "total_duration_in_seconds": 0
  },
  "data_start_time": "2023-11-01",
  "data_end_time": "2023-12-31",
  "labels_present": false,
  "model_version_metrics": {
    "precision": 1.0,
    "recall": 1.0,

```

```
    "mean_fractional_lead_time": 0.7760964912280702,  
    "auc": 0.5971207364893062  
  }  
}
```

Métricas clave

labeled_ranges

Intervalos de tiempo en los que se detectaron anomalías etiquetadas durante el entrenamiento

labeled_event_metrics

Estadísticas sobre qué tan bien el modelo identificó los eventos etiquetados conocidos

num_labeled

Número total de eventos etiquetados en los datos de entrenamiento

num_identified

Número de eventos etiquetados que el modelo identificó correctamente

total_warning_time_in_seconds

Tiempo total que el modelo pasó en estado de advertencia para los eventos etiquetados

Intervalos de tiempo en los que el modelo predijo anomalías durante la evaluación

Estadísticas sobre las anomalías detectadas en datos no etiquetados

data_start_time / data_end_time

Ventana de tiempo cubierta por los datos de entrenamiento

labels_present

Un valor booleano que indica si se usaron datos etiquetados durante el entrenamiento

model_version_metrics

Métricas adicionales específicas de la versión para comparar modelos

Métricas avanzadas para modelos etiquetados

Cuando proporciona datos etiquetados durante el entrenamiento, se incluyen métricas de rendimiento adicionales en los archivos de Amazon S3:

- **Recuerde:** la proporción de eventos que se identificaron AWS IoT SiteWise correctamente con respecto a los eventos que etiquetó durante el mismo período. Por ejemplo, es posible que haya etiquetado 10 eventos, pero AWS IoT SiteWise solo haya identificado 9 de ellos. En este caso, la retirada es del 90%.
- **Precisión:** la proporción de positivos verdaderos con respecto al total de eventos identificados. Por ejemplo, si AWS IoT SiteWise identifica 10 eventos, pero solo 7 de esos eventos corresponden a los eventos que ha etiquetado, la precisión es del 70%.
- **MeanFractionalLeadTime:** una medida de la rapidez con la que, en promedio, se AWS IoT SiteWise detecta cada evento (en relación con la duración del evento). Por ejemplo, un evento típico en sus instalaciones puede durar 10 horas. En promedio, el modelo puede tardar 3 horas en identificar el evento. En este caso, el tiempo medio de entrega fraccional es de 0,7.
- **AUC:** Area Under the Curve (AUC) mide la capacidad de un modelo de aprendizaje automático para predecir una puntuación más alta en los ejemplos positivos en comparación con los negativos. Un valor entre 0 y 1 que indica qué tan bien el modelo es capaz de separar las categorías del conjunto de datos. Un valor de 1 indica que fue capaz de separar las categorías a la perfección.

Promoción y métricas del modelo

Durante los flujos de trabajo de reentrenamiento, las métricas almacenadas en Amazon S3 permiten tomar decisiones informadas sobre la promoción de modelos:

Modo gestionado (promoción automática)

- El sistema compara automáticamente las métricas entre las versiones antiguas y nuevas del modelo utilizando los datos almacenados en Amazon S3
- Los modelos se promocionan en función de indicadores de rendimiento mejorados
- Las decisiones de promoción incluyen códigos de motivo específicos que se almacenan junto con las métricas:
 - **AUTO_PROMOTION_SUCCESSFUL:** Las métricas del nuevo modelo son mejores que las de la versión actual

- `MODEL_METRICS_DIDNT_IMPROVE`: El rendimiento del nuevo modelo no mejoró
- `POOR_MODEL_QUALITY_DETECTED`: El nuevo modelo tiene una evaluación de calidad deficiente

Modo manual (promoción controlada por el cliente)

- Puede descargar y analizar métricas detalladas de Amazon S3 para tomar decisiones de promoción
- Todas las versiones históricas del modelo y sus métricas permanecen accesibles en Amazon S3
- Puede crear paneles y herramientas de análisis personalizados con las métricas almacenadas en Amazon S3

Configuraciones de inferencia avanzadas

AWS IoT SiteWise permite a los clientes configurar programas de inferencia de modelos adaptados a sus necesidades operativas.

La programación de inferencias se clasifica en términos generales en tres modos:

- [Inferencia de alta frecuencia \(de 5 minutos a 1 hora\)](#)
- [Inferencia de baja frecuencia \(2 horas — 1 día\)](#)
- [Programación flexible](#)

Inferencia de alta frecuencia (de 5 minutos a 1 hora)

Este modo es ideal para procesos que funcionan de forma continua o que tienen una alta tasa de cambio en los valores de los sensores. En esta configuración, la inferencia se ejecuta con frecuencia, hasta cada 5 minutos.

Casos de uso:

- Se utiliza para monitorear equipos que cambian rápidamente, como compresores o transportadores.
- Es útil para detectar anomalías de corta duración que requieren una respuesta inmediata.
- Se trata de una operación permanente en la que los datos fluyen de forma constante.

Soporte de compensación condicional:

Puede definir un desplazamiento condicional (de 0 a 60 minutos) para retrasar la inferencia tras la ingesta de datos. Esto garantiza que los datos que llegan tarde se sigan incluyendo en la ventana de análisis.

Para configurar la inferencia de alta frecuencia:

- Configure el valor de la carga útil de la `AWS/ANOMALY_DETECTION_INFERENCE` acción `DataUploadFrequency` con valores: `PT5M`, `PT10M`, `PT15M`, `PT30M`, `PT1H` al iniciar la inferencia.
- (Opcional) Configure `DataDelayOffsetInMinutes` con la compensación de retardo en minutos. Establezca este valor entre 0 y 60 minutos.

```
{
  "inferenceMode": "START",
  "dataDelayOffsetInMinutes": "DataDelayOffsetInMinutes",
  "dataUploadFrequency": "DataUploadFrequency"
}
```

Example de configuración de inferencia de alta frecuencia:

```
{
  "inferenceMode": "START",
  "dataDelayOffsetInMinutes": "2",
  "dataUploadFrequency": "PT5M"
}
```

Inferencia de baja frecuencia (2 horas — 1 día)

Este modo es adecuado para procesos que se mueven más lentamente o para casos de uso en los que las evaluaciones diarias son suficientes. Los clientes configuran la inferencia para que se ejecute cada hora o una vez al día.

Soporte de hora de inicio para un intervalo de 1 día:

Para la inferencia diaria, si lo desea, especifique una **startTime**(a las 8 de la mañana todos los días), junto con el conocimiento de la zona horaria.

Soporte de zona horaria:

Cuando se proporciona `unstartTime`, AWS IoT SiteWise utiliza la [base de datos de zonas horarias](#), mantenida por la Autoridad de Números Asignados de Internet (IANA). Esto garantiza que su inferencia se ajuste a las horas de trabajo locales, incluso en todas las regiones.

Soporte de compensación condicional:

Al igual que con otros modos, se configura un desplazamiento condicional de 0 a 60 minutos.

Casos de uso:

- Controles de estado diarios para procesos por lotes u operaciones por turnos.
- Evita las inferencias durante el mantenimiento o el tiempo de inactividad.
- Resulta útil en entornos con recursos limitados, en los que se debe minimizar el uso informático.

Para configurar la inferencia de baja frecuencia:

- Configure el valor de la carga útil de la `AWS/ANOMALY_DETECTION_INFERENCE` acción `DataUploadFrequency` con los valores: `PT2H..PT12H`
 - En el caso de 1 día, `DataUploadFrequency` es `P1D`.
- (Opcional) Configúrelo `DataDelayOffsetInMinutes` con la compensación de retardo en minutos. Establezca este valor entre 0 y 60 minutos.

Example de configuración de inferencia de baja frecuencia:

```
{
  "inferenceMode": "START",
  "dataUploadFrequency": "P1D",
  "inferenceStartTime": "13:00",
  "inferenceTimeZone": "America/Chicago"
}
```

Programación flexible

La programación flexible permite a los clientes definir días y rangos de tiempo específicos durante los cuales se realiza la inferencia. Esto les da a los clientes un control total sobre la programación en función de las horas de producción, los horarios de los turnos y los tiempos de inactividad planificados.

Esto `weeklyOperatingWindow` ayuda cuando:

- El equipo funciona solo durante horas específicas (de 8 a.m. a 4 p.m.).
- No hay producción los fines de semana.
- El mantenimiento diario se programa durante los intervalos de tiempo conocidos.

Soporte de zona horaria:

Cuando se proporciona `unstartTime`, AWS IoT SiteWise utiliza la [base de datos de zonas horarias](#), mantenida por la Autoridad de Números Asignados de Internet (IANA). Esto garantiza que la inferencia se alinee con los horarios de trabajo locales, incluso en todas las regiones.

Soporte de compensación condicional:

Al igual que con otros modos, se puede configurar un desplazamiento condicional de 0 a 60 minutos.

Ventajas de `weeklyOperatingWindow`:

- Evita las inferencias durante los períodos de inactividad o mantenimiento, lo que reduce los falsos positivos.
- Alinea la detección de anomalías con las prioridades operativas y los flujos de trabajo basados en turnos.

Para configurar una programación flexible:

- Configure el valor de carga útil de la `AWS/ANOMALY_DETECTION_INFERENCE` acción con `DataUploadFrequency`.
- (Opcional) `DataDelayOffsetInMinutes` con el retardo compensado en minutos. Establezca este valor entre 0 y 60 minutos.
- Configure `weeklyOperatingWindow` con una configuración de turnos:
 - Las claves `weeklyOperatingWindow` son los días de la semana: `monday` | `tuesday` | `wednesday` | `thursday` | `friday` | `saturday` | `sunday`.
 - Cada intervalo de tiempo debe estar en formato de 24 horas como `"HH:MM-HH:MM"` (`"08:00-16:00"`).
 - Se pueden especificar varios intervalos por día.

Example de configuración de programación flexible:

```
{
```

```
"inferenceMode": "START",
"dataUploadFrequency": "PT5M",
"weeklyOperatingWindow": {
  "tuesday": ["11:00-13:00"],
  "monday": ["10:00-11:00", "13:00-15:00"]
}
}
```

Activación de la versión del modelo

Al iniciar la inferencia, puede activar opcionalmente una versión específica del modelo para utilizarla en la detección de anomalías. Esta función le permite seleccionar una versión específica del modelo entrenado, volver a versiones anteriores o anular las decisiones automáticas de promoción del modelo.

Casos de uso

- Reversión de la producción: vuelva rápidamente a una versión de modelo estable cuando la versión actual muestre un rendimiento inferior o un comportamiento inesperado.
- Pruebas A/B: compare el rendimiento entre diferentes versiones del modelo cambiando de una a otra durante períodos de tiempo controlados.
- Selección manual del modelo: anule las decisiones de promoción automáticas y seleccione manualmente la versión de modelo que prefiera en función de los requisitos empresariales.
- Implementación por etapas: pruebe las versiones de modelos más recientes en períodos de tiempo no críticos antes de promocionarlos para su uso en plena producción.
- Optimización del rendimiento: seleccione las versiones del modelo que tengan un mejor rendimiento para condiciones operativas específicas o patrones estacionales.
- Reversión durante el mantenimiento: utilice versiones de modelos más antiguas y probadas durante el mantenimiento o las actualizaciones del sistema para garantizar la estabilidad.

Comportamiento de selección de versiones del modelo

Cuándo `targetModelVersion` se especifica:

- El sistema activa la versión del modelo solicitada para realizar inferencias.
- Valida la existencia de la versión del modelo especificada.
- Anula cualquier configuración de promoción automática.

Cuando no `targetModelVersion` se especifica:

- Activa la última versión del modelo activo si la inferencia se inició anteriormente.
- Si la inferencia nunca se activó, utiliza la última versión del modelo entrenado.

Para activar una versión de modelo específica:

- Configure la carga útil de la acción de inferencia con el `targetModelVersion` número de versión de modelo que desee.
- La versión del modelo especificada se valida y activa si existe.

Example de la activación de la versión del modelo:

```
{
  "inferenceMode": "START",
  "dataUploadFrequency": "PT15M",
  "targetModelVersion": 2
}
```

Comprobación de las versiones del modelo

Para comprobar la versión del modelo actualmente activa:

- Utilice la [DescribeComputationModelExecutionSummary](#) API, que incluye la versión del modelo activo en la respuesta.

Para ver todas las versiones de modelos disponibles:

- Utilice la [ListExecutions](#) API para obtener una lista completa de las versiones históricas de los modelos.
- Utilice la [DescribeExecution](#) API para recuperar información específica del modelo, incluido el intervalo de tiempo de los datos de exportación, la versión del modelo de cálculo y la duración facturable en minutos.

Características de la versión del modelo

- Los números de versión del modelo se asignan secuencialmente empezando por 1.

- Puede activar cualquier versión del modelo previamente entrenada.
- La versión del modelo activada se conserva hasta que se modifique de forma explícita.
- La activación de la versión del modelo funciona con todos los modos de programación de inferencias (alta, baja frecuencia y flexible).
- Si la versión del modelo especificada no existe, la acción de inferencia falla y se produce un error.

Revise los resultados de la inferencia

Recupere los resultados de la inferencia

Resultados de inferencia más recientes

Ejecute el siguiente comando para obtener el resultado de inferencia más reciente de una propiedad de activo. Para obtener más información, consulte la [get-asset-property-value](#) Guía de referencia de AWS CLI comandos.

```
aws iotsitewise get-asset-property-value \  
  -asset-id asset-id \  
  -property-id result-property-id
```

Historial de resultados de inferencias

Ejecute el siguiente comando para obtener el historial de los resultados de la inferencia para un período de tiempo específico. Para obtener más información, consulte [get-asset-property-value-history en la Guía](#) de referencia de AWS CLI comandos.

```
aws iotsitewise get-asset-property-value-history \  
  -asset-id asset-id \  
  -property-id result-property-id \  
  -start-date start-time \  
  -end-date end-time
```

Ejemplo de respuesta

Example de una respuesta al resultado de una inferencia:

```
{  
  "value": {
```

```

    "stringValue": "{\"timestamp\": \"2025-02-10T22:42:00.000000\", \"prediction\": 0,
    \"prediction_reason\": \"NO_ANOMALY_DETECTED\", \"diagnostics\": [{\"name\": \"asset-
    id\\\\\\\\property-id\", \"value\": 0.53528}]}"
  },
  "timestamp": {
    "timeInSeconds": 1739227320,
    "offsetInNanos": 0
  },
  "quality": "GOOD"
}

```

Campos de respuesta

- `value.stringValue`: cadena JSON que contiene el resultado de la inferencia con los siguientes campos:
 - `timestamp`: marca temporal del TQV con el que se realiza la inferencia.
 - `predicción`: el resultado de la predicción (0 para que no se detecte ninguna anomalía, 1 para la anomalía detectada).
 - `prediction_reason`: el motivo de la predicción (o). `NO_ANOMALY_DETECTED`
`ANOMALY_DETECTED`
 - `diagnóstico`: conjunto de información de diagnóstico que muestra los factores que contribuyen a ello.
- `marca temporal`: marca temporal en la que se registra el resultado. AWS IoT SiteWise
- `calidad`: la calidad del punto de datos (normalmente). `GOOD`

Comprenda los resultados de las inferencias

El resultado de una inferencia devuelto por la detección de AWS IoT SiteWise anomalías incluye información clave sobre la predicción del modelo en un momento específico, como si se detectó una anomalía y qué sensores contribuyeron a la anomalía.

Example de un resultado de inferencia detallado:

```

{
  "timestamp": "2021-03-11T22:25:00.000000",
  "prediction": 1,
  "prediction_reason": "ANOMALY_DETECTED",
  "anomaly_score": 0.72385,

```

```
"diagnostics": [  
  { "name": "asset_id_1\\\\\\\\property_id_1", "value": 0.02346 },  
  { "name": "asset_id_2\\\\\\\\property_id_2", "value": 0.10011 },  
  { "name": "asset_id_3\\\\\\\\property_id_3", "value": 0.11162 }  
]  
}
```

El `diagnostics` campo es útil para interpretar por qué el modelo hace una predicción determinada. Cada entrada de la lista incluye:

- `name`: El sensor que contribuyó a la predicción (formato: `asset_id\\\\\\\\property_id`).
- `value`: Un número de punto flotante entre 0 y 1, que representa el peso o la importancia relativos de ese sensor en ese momento.

Ventajas para el usuario:

- Comprenda qué sensores tuvieron el mayor impacto en una anomalía.
- Correlaciona los sensores de alto peso con el comportamiento del equipo físico.
- Infórmese sobre el análisis de la causa raíz.

Note

Incluso cuando `prediction = 0` (comportamiento normal), se devuelve la lista de diagnósticos. Esto ayuda a evaluar qué sensores influyen actualmente en las decisiones del modelo, incluso en estados saludables.

Activa acciones personalizadas en caso de comportamiento anómalo ()Consola de administración de AWS

Puede activar acciones personalizadas en respuesta a un comportamiento anómalo mediante el uso de notificaciones AWS IoT SiteWise MQTT en combinación con. AWS IoT Core

Siga estos pasos para configurar las notificaciones de MQTT y activar una acción personalizada en AWS IoT SiteWise AWS IoT Core función de los resultados de la inferencia:

- Localice el activo en el lugar donde AWS IoT SiteWise se ejecuta la inferencia.

- Identifique la propiedad que utilizó `resultProperty` al crear el modelo de cálculo. Habilite la notificación MQTT para esta propiedad.
- Una vez que active la notificación MQTT, copie el tema de notificación que se genera en AWS IoT SiteWise.
- Navegue hasta **AWS IoT Core** En el cliente de prueba de MQTT, suscríbese al tema de notificación copiado para supervisar los mensajes entrantes.
- Cree una **AWS IoT Core** regla para procesar los resultados de las inferencias. (Esto garantiza que las acciones se activen solo si el sistema detecta una anomalía). `notification-topic` Sustitúyalo por el tema de notificación que se genera en AWS IoT SiteWise.

```
SELECT * FROM "notification-topic"  
WHERE indexof(get(get(payload.values, 0).value, 'stringValue'),  
"NO_ANOMALY_DETECTED") < 0
```

Configure la regla para activar cualquiera de las acciones **AWS IoT Core** compatibles. Obtenga más información sobre las acciones compatibles con [AWS IoT Core](#).

Prácticas recomendadas

Comprenda el intervalo de fechas mínimo

Utilice un mínimo de 14 días para la duración de los datos de entrenamiento. Sin embargo, le recomendamos que incluya un período de datos más largo en muchos casos.

Asegúrese de que su conjunto de datos de entrenamiento abarque un período de tiempo durante el cual el activo funcionó en todos sus modos de funcionamiento normales. Este enfoque ayuda a distinguir **AWS IoT SiteWise** con precisión entre el comportamiento esperado y las anomalías reales.

Si los datos de entrenamiento no representan todos los modos de funcionamiento típicos, es posible que **AWS IoT SiteWise** marque incorrectamente como anomalías los patrones no conocidos pero normales, lo que aumenta los falsos positivos.

Toma de muestras para obtener datos de alta frecuencia y garantizar la coherencia entre el entrenamiento y la inferencia

Si sus sensores generan datos a una frecuencia superior a 1 Hz (más de una lectura por segundo), aplique el muestreo durante el entrenamiento. El muestreo reduce el volumen de datos y, al mismo

tiempo, preserva las tendencias esenciales, lo que permite un procesamiento eficiente y mejora la generalización de los modelos al minimizar el impacto del ruido o las fluctuaciones transitorias.

AWS IoT SiteWise La detección nativa de anomalías actualmente no admite la ingesta de datos a velocidades inferiores a 1 Hz. Compruebe que sus datos cumplen este requisito de frecuencia mínima antes de configurar la detección de anomalías.

Además, también AWS IoT SiteWise utiliza la frecuencia de muestreo que configuró durante el entrenamiento para realizar inferencias. Para mantener la coherencia y garantizar unos resultados de detección de anomalías precisos, elija una frecuencia de muestreo que se adapte tanto a sus necesidades operativas como al comportamiento de los datos de los sensores.

Obtenga más información sobre cómo establecer la frecuencia de muestreo en [configuración de frecuencia de muestreo](#)

Recomendaciones de etiquetado

El etiquetado preciso y coherente de las anomalías es esencial para una evaluación eficaz del modelo y una mejora continua. Tenga en cuenta las siguientes prácticas recomendadas al etiquetar las anomalías:

- **Consolide las anomalías relacionadas:** no etiquete las anomalías que se producen de forma cercana como eventos separados si forman parte del mismo problema subyacente. Por ejemplo, si las anomalías se producen con una diferencia de uno a dos días y su origen es la misma causa, trátelas como si se tratara de un único intervalo de anomalías. Este enfoque ayuda al modelo a conocer mejor el patrón de comportamiento anormal y reduce el ruido en los datos de la evaluación.
- **Etiquete las ventanas de anomalías, no solo los puntos:** en lugar de marcar puntos de datos individuales como anómalos, etiquete toda la ventana que refleja el comportamiento anormal desde el inicio de la desviación hasta la recuperación. Este enfoque proporciona límites más claros y mejora la alineación del modelo con los problemas operativos reales.
- **Excluya los períodos inciertos:** si no está seguro de si un período es anómalo, déjelo sin etiquetar. Las etiquetas ambiguas pueden confundir el modelo y reducir su precisión con el paso del tiempo.

Encontrará más información sobre cómo añadir etiquetas en [Etiquete sus datos](#).

Gestione el almacenamiento de datos en AWS IoT SiteWise

Puede configurarlo AWS IoT SiteWise para guardar sus datos en los siguientes niveles de almacenamiento:

Nivel de acceso frecuente

El nivel de almacenamiento activo es un almacenamiento de series temporales AWS IoT SiteWise gestionado. El nivel activo es más eficaz para los datos a los que se accede con frecuencia, con baja write-to-read latencia. Los datos almacenados en el nivel de acceso frecuente son utilizados por aplicaciones industriales que necesitan acceso rápido a los valores más recientes de las mediciones de sus equipos. Esto incluye aplicaciones que visualizan métricas en tiempo real con un panel interactivo o aplicaciones que supervisan las operaciones y activan alarmas para identificar problemas de rendimiento.

De forma predeterminada, los datos ingresados AWS IoT SiteWise se almacenan en la capa activa. Puede definir un período de retención para el nivel activo, tras el cual AWS IoT SiteWise se transfieren los datos del nivel activo a un almacenamiento de nivel caliente o frío, según su configuración. Para obtener el mejor rendimiento y rentabilidad, configure el período de retención del nivel de acceso frecuente para que sea más largo que el tiempo que se tarda en recuperar los datos con frecuencia. Esto se usa para métricas, alarmas y escenarios de supervisión en tiempo real. Si no se establece un período de retención, sus datos se almacenarán indefinidamente en el nivel de acceso frecuente.

Nivel de acceso semiactivo

El nivel de almacenamiento en caliente es un nivel AWS IoT SiteWise administrado que resulta eficaz para el almacenamiento rentable de datos históricos. Se utiliza mejor para recuperar grandes volúmenes de datos con características de write-to-read latencia media. Utilice el nivel de acceso semiactivo para almacenar datos históricos que se necesitan para grandes cargas de trabajo. Por ejemplo, se utiliza para la recuperación de datos para análisis, aplicaciones de inteligencia empresarial (BI), herramientas de elaboración de informes y entrenamiento de modelos de machine learning (ML). Si habilita el nivel de almacenamiento de acceso inactivo, puede definir un período de retención del nivel de acceso semiactivo. Una vez finalizado el período de retención, AWS IoT SiteWise elimina los datos del nivel cálido.

Nivel inactivo

El nivel de almacenamiento de acceso inactivo utiliza un bucket de Amazon S3 para almacenar datos que se utilizan con poca frecuencia. Con el nivel de acceso inactivo, AWS IoT SiteWise

replica las series temporales, incluidas mediciones, transformaciones y agregados, así como las definiciones del modelo de activos, cada 6 horas. El nivel de acceso inactivo se utiliza para almacenar datos que toleran una latencia de alta lectura para informes históricos y copias de seguridad.

Temas

- [Configure los ajustes de almacenamiento en AWS IoT SiteWise](#)
- [Solucionar problemas de configuración de almacenamiento para AWS IoT SiteWise](#)
- [Rutas de archivos y esquemas de datos guardados en el nivel inactivo](#)

Configure los ajustes de almacenamiento en AWS IoT SiteWise

Puede configurar ajustes de almacenamiento para habilitar el almacenamiento de nivel de acceso semiactivo administrado por el servicio y también para replicar los datos en el nivel de acceso inactivo. Para obtener más información sobre el período de retención de los niveles de acceso semiactivo y frecuente, consulte [Impacto en la retención de datos](#). Al configurar los ajustes de almacenamiento, haga lo siguiente:

- **Retención en la capa activa:** establece un período de retención durante el cual tus datos se almacenan en la capa activa antes de que se eliminen y se transfieran al almacenamiento en capa caliente o almacenamiento en capa fría gestionado por el servicio en función de tu configuración de almacenamiento. AWS IoT SiteWise eliminará todos los datos de la capa activa que existieran antes de que finalice el período de retención. Si no establece un período de retención, sus datos se almacenan indefinidamente en el nivel de acceso frecuente.
- **Retención en la capa cálida:** establece un período de retención para el tiempo que tus datos permanecerán almacenados en la capa cálida antes de que se eliminen del AWS IoT SiteWise almacenamiento y se trasladen al almacenamiento en la capa fría gestionada por el cliente. AWS IoT SiteWise elimina todos los datos de la capa cálida que existían antes de que finalizara el período de retención. Si no se establece un período de retención, sus datos se almacenan indefinidamente en el nivel de acceso semiactivo.

Note

Para mejorar el rendimiento de las consultas, establezca un período de retención de nivel de acceso activo con almacenamiento de nivel de acceso semiactivo.

Impacto de la retención de datos en el almacenamiento de nivel de acceso activo y semiactivo

- Al reducir el período de retención del almacenamiento de nivel de acceso activo, los datos se transfieren permanentemente del nivel de acceso activo al nivel de acceso semiactivo o inactivo. Al reducir el período de retención del almacenamiento de nivel de acceso semiactivo, los datos se transfieren permanentemente del nivel de acceso inactivo y se eliminan permanentemente del nivel de acceso semiactivo.
- Al aumentar el período de retención del almacenamiento de nivel caliente o caliente, el cambio afecta a los datos que se envíen a AWS IoT SiteWise partir de ese momento. AWS IoT SiteWise no recupera los datos del almacenamiento caliente o frío para poblarlos en el nivel activo. Por ejemplo, si el período de retención del almacenamiento de nivel de acceso activo se establece inicialmente en 30 días y más tarde se aumenta a 60 días, el almacenamiento de nivel de acceso activo tarda 30 días en contener datos de 60 días.

Temas

- [Configuración de ajustes de almacenamiento para el nivel de acceso semiactivo \(consola\)](#)
- [Configuración de ajustes de almacenamiento para el nivel de acceso semiactivo \(AWS CLI\)](#)
- [Configuración de ajustes de almacenamiento para el nivel de acceso inactivo \(consola\)](#)
- [Configuración de ajustes de almacenamiento para el nivel de acceso inactivo \(AWS CLI\)](#)

Configuración de ajustes de almacenamiento para el nivel de acceso semiactivo (consola)

El siguiente procedimiento muestra cómo configurar los ajustes de almacenamiento para replicar los datos en el nivel caliente de la AWS IoT SiteWise consola.


Para configurar los parámetros de disponibilidad en la consola

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, en Configuración, seleccione Listas.
3. En la esquina superior derecha, elija Edit (Editar).
4. En la página Editar acción, haga lo siguiente:
5. En Configuración del nivel caliente, haga lo siguiente:
 - Si quiere establecer un período de retención durante el cual se almacenen sus datos en el nivel de acceso frecuente antes de que se eliminen y transfieran al almacenamiento de nivel de acceso semiactivo administrado por el servicio, elija Habilitar período de retención.
 - Para configurar un período de retención, introduzca un número entero y elija una unidad. El periodo de retención debe ser mayor o igual a 30 días.

AWS IoT SiteWise elimina todos los datos del nivel activo que sean anteriores al período de retención. Si no establece un período de retención, sus datos se almacenarán indefinidamente.

6. (Recomendado) Para Configuración de nivel cálido, haga lo siguiente:
 - Para suscribirse al almacenamiento en el nivel de acceso frecuente, seleccione Confirmando la suscripción al almacenamiento de nivel cálido.
 - (Opcional) Para configurar un período de retención, introduzca un número entero y elija una unidad. El periodo de retención debe ser mayor o igual a 365 días.

AWS IoT SiteWise elimina los datos del nivel cálido que existían antes del período de retención. Si no establece un período de retención, sus datos se almacenarán indefinidamente.

 Note

- Si se suscribe al nivel de acceso semiactivo, la configuración se muestra solo una vez.
- Para configurar la retención de nivel de acceso frecuente debe tener almacenamiento de nivel de acceso semiactivo o inactivo. Para lograr una mayor rentabilidad y recuperar datos históricos, se AWS IoT SiteWise recomienda almacenar los datos a largo plazo en el nivel cálido.

- Para configurar la retención de nivel de acceso semiactivo debe tener almacenamiento de nivel de acceso inactivo.

7. Seleccione Guardar para guardar la configuración de almacenamiento.

En la sección Almacenamiento de AWS IoT SiteWise , el Almacenamiento en el nivel cálido se encuentra en uno de los siguientes estados:

- **Habilitado:** si los datos existían antes del período de retención del nivel de acceso activo, AWS IoT SiteWise los mueve al nivel de acceso semiactivo.
- **Deshabilitado:** el almacenamiento de nivel semiactivo está deshabilitado.

Configuración de ajustes de almacenamiento para el nivel de acceso semiactivo (AWS CLI)

Puede configurar los ajustes de almacenamiento para mover los datos al nivel cálido mediante AWS CLI los siguientes comandos.

Para evitar anular la configuración existente, recupere la información de configuración de almacenamiento actual ejecutando el siguiente comando:

```
aws iotsitewise describe-storage-configuration
```

Example respuesta sin configuración de nivel de acceso inactivo existente

```
{
  "storageType": "SITEWISE_DEFAULT_STORAGE",
  "disassociatedDataStorage": "ENABLED",
  "configurationStatus": {
    "state": "ACTIVE"
  },
  "lastUpdateDate": "2021-10-14T15:53:35-07:00",
  "warmTier": "DISABLED"
}
```

Example respuesta con configuración de nivel de acceso inactivo existente

```
{
```

```

"storageType": "MULTI_LAYER_STORAGE",
  "multiLayerStorage": {
    "customerManagedS3Storage": {
      "s3ResourceArn": "arn:aws:s3:::amzn-s3-demo-bucket/prefix/",
      "roleArn": "arn:aws:iam::aws-account-id:role/role-name"
    }
  },
  "disassociatedDataStorage": "ENABLED",
  "retentionPeriod": {
    "numberOfDays": retention-in-days
  },
  "configurationStatus": {
    "state": "ACTIVE"
  },
  "lastUpdateDate": "2023-10-25T15:59:46-07:00",
  "warmTier": "DISABLED"
}

```

Configure los ajustes de almacenamiento para el nivel cálido con AWS CLI

Ejecute el comando siguiente para configurar los ajustes de almacenamiento. `file-name` Sustitúyalo por el nombre del archivo que contiene la configuración AWS IoT SiteWise de almacenamiento.

```
aws iotsitewise put-storage-configuration --cli-input-json file://file-name.json
```

Example AWS IoT SiteWise configuración con niveles caliente y cálido

```

{
  "storageType": "SITEWISE_DEFAULT_STORAGE",
  "disassociatedDataStorage": "ENABLED",
  "warmTier": "ENABLED",
  "retentionPeriod": {
    "numberOfDays": hot-tier-retention-in-days
  }
}

```

`hot-tier-retention-in-days` debe ser un número entero mayor o igual que 30 días.

Example response

```
{
```

```
    "storageType": "SITEWISE_DEFAULT_STORAGE",
    "configurationStatus": {
      "state": "UPDATE_IN_PROGRESS"
    }
  }
```

Si tiene habilitado el almacenamiento de nivel de acceso inactivo, consulte [Configure los ajustes de almacenamiento con AWS CLI un nivel frío existente](#).

Configure los ajustes de almacenamiento con AWS CLI un nivel frío existente

Configure los ajustes de almacenamiento utilizando AWS CLI el almacenamiento en capa fría existente

- Ejecute el comando siguiente para configurar los ajustes de almacenamiento. Sustituya *file-name* por el nombre del archivo que contiene la configuración de almacenamiento de AWS IoT SiteWise .

```
aws iotsitewise put-storage-configuration --cli-input-json file://file-name.json
```

Example AWS IoT SiteWise configuración de almacenamiento

- Reemplace *amzn-s3-demo-bucket* por el nombre de su bucket de Amazon S3.
- *prefix* Sustitúyalo por el prefijo de Amazon S3.
- *aws-account-id* Sustitúyalo por tu ID AWS de cuenta.
- *role-name* Sustitúyalo por el nombre del rol de acceso de Amazon S3 que permite AWS IoT SiteWise enviar datos a Amazon S3.
- *hot-tier-retention-in-days* Sustitúyalo por un número entero mayor o igual a 30 días.
- *warm-tier-retention-in-days* Sustitúyalo por un número entero mayor o igual a 365 días.

Note

AWS IoT SiteWise eliminará todos los datos de la capa cálida que sean anteriores al período de retención de la capa fría. Si no establece un período de retención, sus datos se almacenarán indefinidamente.

```
{
  "storageType": "MULTI_LAYER_STORAGE",
  "multiLayerStorage": {
    "customerManagedS3Storage": {
      "s3ResourceArn": "arn:aws:s3:::amzn-s3-demo-bucket/prefix/",
      "roleArn": "arn:aws:iam::aws-account-id:role/role-name"
    }
  },
  "disassociatedDataStorage": "ENABLED",
  "retentionPeriod": {
    "numberOfDays": hot-tier-retention-in-days
  },
  "warmTier": "ENABLED",
  "warmTierRetentionPeriod": {
    "numberOfDays": warm-tier-retention-in-days
  }
}
```

Example response

```
{
  "storageType": "MULTI_LAYER_STORAGE",
  "configurationStatus": {
    "state": "UPDATE_IN_PROGRESS"
  }
}
```


Configuración de ajustes de almacenamiento para el nivel de acceso inactivo (consola)

El siguiente procedimiento muestra cómo configurar los ajustes de almacenamiento para replicar los datos en la capa fría de la AWS IoT SiteWise consola.

Para configurar los parámetros de disponibilidad en la consola


1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, en Configuración, seleccione Listas.
3. En la esquina superior derecha, elija Edit (Editar).

4. En la página Editar acción, haga lo siguiente:
 - a. En la Configuración de almacenamiento, seleccione Habilitar el almacenamiento en el nivel inactivo. El almacenamiento en el nivel inactivo está desactivado de forma predeterminada.
 - b. En Ubicación del bucket de S3, introduzca el nombre de un bucket de Amazon S3 existente y un prefijo.

 Note

- Amazon S3 utiliza el prefijo como nombre de carpeta en el bucket de Amazon S3. El prefijo debe tener entre 1 y 255 caracteres y terminar con una barra diagonal (/). Sus AWS IoT SiteWise datos se guardan en esta carpeta.
- Si no dispone de un bucket de Amazon S3, seleccione Ver y, a continuación, cree uno en la consola de Amazon S3. Para obtener más información, consulte [Creación del primer bucket de S3](#) en la Guía del usuario de Amazon S3.

- c. Para rol de acceso a S3, realice una de las operaciones siguientes:
 - Si selecciona Crear un rol a partir de una plantilla AWS gestionada, crea AWS automáticamente un rol de IAM que permite AWS IoT SiteWise enviar datos a Amazon S3.
 - Elija Usar un rol existente y, a continuación, elija el rol que creó de la lista.

 Note

- Debe usar el mismo nombre de bucket de Amazon S3 para la Ubicación del bucket de S3 que utilizó en el paso anterior y en su política de IAM.
- Asegúrese de que el rol tenga los permisos que se muestran en el siguiente ejemplo.

Example política de permisos:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```

```

    "Effect": "Allow",
    "Action": [
        "s3:PutObject",
        "s3:GetObject",
        "s3:DeleteObject",
        "s3:GetBucketLocation",
        "s3:ListBucket"
    ],
    "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket",
        "arn:aws:s3:::amzn-s3-demo-bucket/*"
    ]
}
]
}

```

Sustituya amzn-s3-demo-bucket por el nombre de su bucket de Amazon S3.

- Si el bucket de Amazon S3 se cifra con una clave de KMS gestionada por el cliente, la clave de KMS debe tener una política de acceso con una función de IAM para kms:Decrypt las kms:GenerateDataKey operaciones.

- Para configurar el nivel de acceso frecuente, consulte el paso 5 en [Configuración de ajustes de almacenamiento para el nivel de acceso semiactivo \(consola\)](#).
- (Opcional) Para la integración de AWS IoT Analytics , haga lo siguiente.

Note

Aviso de fin del soporte: el 15 de diciembre de 2025, AWS finalizará el soporte para AWS IoT Analytics. Después del 15 de diciembre de 2025, ya no podrás acceder a la AWS IoT Analytics consola ni a AWS IoT Analytics los recursos. Para obtener más información, consulta [AWS IoT Analytics el fin del soporte](#).

- Si desea utilizarlos AWS IoT Analytics para consultar sus datos, elija Almacén de AWS IoT Analytics datos activado.
- AWS IoT SiteWise genera un nombre para el banco de datos o puede introducir un nombre diferente.

AWS IoT SiteWise crea automáticamente un almacén de datos AWS IoT Analytics para guardar sus datos. Para consultar los datos, puede utilizarlos AWS IoT Analytics para crear conjuntos de datos. Para obtener más información, consulte [Trabajar con AWS IoT SiteWise datos](#) en la Guía del AWS IoT Analytics usuario.

- f. Seleccione Save.

En la sección Almacenamiento de AWS IoT SiteWise , el Almacenamiento en el nivel inactivo puede tener uno de los siguientes valores:

- **Habilitado:** AWS IoT SiteWise replica los datos en el bucket de Amazon S3 especificado.
- **Habilitación:** AWS IoT SiteWise está procesando su solicitud para habilitar el almacenamiento en capas frías. Este proceso puede tardar varios minutos en completarse.
- **Enable_Failed:** no se ha AWS IoT SiteWise podido procesar tu solicitud para habilitar el almacenamiento en capa fría. Si has habilitado AWS IoT SiteWise el envío de registros a Amazon CloudWatch Logs, puedes usar estos registros para solucionar problemas. Para obtener más información, consulte [Supervise con Amazon CloudWatch Logs](#).
- **Deshabilitado:** el almacenamiento en el nivel inactivo está desactivado.

Configuración de ajustes de almacenamiento para el nivel de acceso inactivo (AWS CLI)

El siguiente procedimiento muestra cómo configurar los ajustes de almacenamiento para replicar datos en el nivel inactivo mediante AWS CLI.

Para configurar los ajustes de almacenamiento mediante AWS CLI

1. Para exportar datos a un bucket de Amazon S3 en su cuenta, ejecute el siguiente comando para configurar los ajustes de almacenamiento. *file-name* Sustitúyalo por el nombre del archivo que contiene la configuración AWS IoT SiteWise de almacenamiento.

```
aws iotsitewise put-storage-configuration --cli-input-json file://file-name.json
```

Example AWS IoT SiteWise configuración de almacenamiento

- Reemplace *amzn-s3-demo-bucket* por el nombre de su bucket de Amazon S3.

- *prefix* Sustitúyalo por el prefijo de Amazon S3.
- *aws-account-id* Sustitúyalo por tu ID AWS de cuenta.
- *role-name* Sustitúyalo por el nombre del rol de acceso de Amazon S3 que permite AWS IoT SiteWise enviar datos a Amazon S3.
- *retention-in-days* Sustitúyalo por un número entero que sea mayor o igual a 30 días.

```
{
  "storageType": "MULTI_LAYER_STORAGE",
  "multiLayerStorage": {
    "customerManagedS3Storage": {
      "s3ResourceArn": "arn:aws:s3:::amzn-s3-demo-bucket/prefix/",
      "roleArn": "arn:aws:iam::aws-account-id:role/role-name"
    }
  },
  "retentionPeriod": {
    "numberOfDays": retention-in-days,
    "unlimited": false
  }
}
```

Note

- Debe usar el mismo nombre de bucket de Amazon S3 en la configuración de AWS IoT SiteWise almacenamiento y en la política de IAM.
- Asegúrese de que el rol tenga los permisos que se muestran en el siguiente ejemplo.

Example política de permisos:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:GetObject",
        "s3:DeleteObject",

```

```

        "s3:GetBucketLocation",
        "s3:ListBucket"
    ],
    "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket",
        "arn:aws:s3:::amzn-s3-demo-bucket/*"
    ]
}
]
}

```

Sustituya `amzn-s3-demo-bucket` por el nombre de su bucket de Amazon S3.

- Si el bucket de Amazon S3 se cifra con una clave de KMS gestionada por el cliente, la clave de KMS debe tener una política de acceso con una función de IAM para `kms:Decrypt` las `kms:GenerateDataKey` operaciones.

Example response

```

{
  "storageType": "MULTI_LAYER_STORAGE",
  "retentionPeriod": {
    "numberOfDays": 100,
    "unlimited": false
  },
  "configurationStatus": {
    "state": "UPDATE_IN_PROGRESS"
  }
}

```

Note

La actualización de la configuración de AWS IoT SiteWise almacenamiento puede tardar unos minutos.

2. Para recuperar la información de configuración del almacenamiento, ejecute el siguiente comando.

```
aws iotsitewise describe-storage-configuration
```

Example response

```
{
  "storageType": "MULTI_LAYER_STORAGE",
  "multiLayerStorage": {
    "customerManagedS3Storage": {
      "s3ResourceArn": "arn:aws:s3:::amzn-s3-demo-bucket/torque/",
      "roleArn": "arn:aws:iam::123456789012:role/SWAccessS3Role"
    }
  },
  "retentionPeriod": {
    "numberOfDays": 100,
    "unlimited": false
  },
  "configurationStatus": {
    "state": "ACTIVE"
  },
  "lastUpdateDate": "2021-03-30T15:54:14-07:00"
}
```

3. Para detener la exportación de datos al bucket de Amazon S3, ejecute el siguiente comando para configurar los ajustes de almacenamiento.

```
aws iotsitewise put-storage-configuration --storage-type SITEWISE_DEFAULT_STORAGE
```

Note

De forma predeterminada, los datos solo se almacenan en la capa activa de AWS IoT SiteWise.

Example response

```
{
  "storageType": "SITEWISE_DEFAULT_STORAGE",
  "configurationStatus": {
    "state": "UPDATE_IN_PROGRESS"
  }
}
```

4. Para recuperar la información de configuración del almacenamiento, ejecute el siguiente comando.

```
aws iotsitewise describe-storage-configuration
```

Example response

```
{
  "storageType": "SITEWISE_DEFAULT_STORAGE",
  "configurationStatus": {
    "state": "ACTIVE"
  },
  "lastUpdateDate": "2021-03-30T15:57:14-07:00"
}
```

(Opcional) Cree un almacén AWS IoT Analytics de datos (AWS CLI)

Note

Aviso de fin del soporte: el 15 de diciembre de 2025, AWS finalizará el soporte para AWS IoT Analytics. Después del 15 de diciembre de 2025, ya no podrás acceder a la AWS IoT Analytics consola ni a AWS IoT Analytics los recursos. Para obtener más información, consulta [AWS IoT Analytics el fin del soporte](#).

Un almacén de AWS IoT Analytics datos es un repositorio escalable y consultable que recibe y almacena datos. Puede usar la AWS IoT SiteWise consola o AWS IoT Analytics APIs crear un banco de AWS IoT Analytics datos para guardar los datos AWS IoT SiteWise . Para consultar los datos, cree conjuntos de datos mediante AWS IoT Analytics. Para obtener más información, consulte [Trabajo con datos de AWS IoT SiteWise](#) en la Guía del usuario de AWS IoT Analytics .

Los siguientes pasos se utilizan AWS CLI para crear un almacén de datos en AWS IoT Analytics.

Ejecute el siguiente comando para crear un almacén de datos. *file-name* Sustitúyalo por el nombre del archivo que contiene la configuración del almacén de datos.

```
aws iotanalytics create-datastore --cli-input-json file://file-name.json
```

Note

- Debe especificar el nombre de un bucket de Amazon S3 existente. Si no dispone de un bucket de Amazon S3, cree uno primero. Para obtener más información, consulte [Creación del primer bucket de S3](#) en la Guía del usuario de Amazon S3.
- Debe usar el mismo nombre de bucket de Amazon S3 en la configuración de AWS IoT SiteWise almacenamiento, la política de IAM y la configuración del almacén de AWS IoT Analytics datos.

Example AWS IoT Analytics configuración del almacén de datos

Sustituya *data-store-name* y *amzn-s3-demo-bucket* por el nombre del almacén de AWS IoT Analytics datos y el nombre del bucket de Amazon S3.

```
{
  "datastoreName": "data-store-name",
  "datastoreStorage": {
    "iotSiteWiseMultiLayerStorage": {
      "customerManagedS3Storage": {
        "bucket": "amzn-s3-demo-bucket"
      }
    }
  },
  "retentionPeriod": {
    "numberOfDays": 90
  }
}
```

Example response

```
{
  "datastoreName": "datastore_IoTSiteWise_demo",
  "datastoreArn": "arn:aws:iotanalytics:us-west-2:123456789012:datastore/
datastore_IoTSiteWise_demo",
  "retentionPeriod": {
    "numberOfDays": 90,
    "unlimited": false
  }
}
```

Solucionar problemas de configuración de almacenamiento para AWS IoT SiteWise

Utilice la siguiente información como ayuda para solucionar problemas con la configuración de almacenamiento.

Problemas

- [Error: el bucket no existe](#)
- [Error: acceso denegado a la ruta de Amazon S3](#)
- [Error: no se puede asumir el ARN del rol](#)
- [Error: no se ha podido acceder al bucket de Amazon S3 entre regiones](#)

Error: el bucket no existe

Solución: no AWS IoT SiteWise ha podido encontrar su bucket de Amazon S3. Asegúrese de introducir el nombre de un bucket de Amazon S3 existente en la región actual.

Error: acceso denegado a la ruta de Amazon S3

Solución: no AWS IoT SiteWise ha podido acceder a su bucket de Amazon S3. Haga lo siguiente:

- Asegúrese de utilizar el mismo bucket de Amazon S3 que especificó en la política de IAM.
- Asegúrese de que el rol tenga los permisos que se muestran en el siguiente ejemplo.

Example política de permisos

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:GetObject",
        "s3:DeleteObject",
```

```

        "s3:GetBucketLocation",
        "s3:ListBucket"
    ],
    "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket",
        "arn:aws:s3:::amzn-s3-demo-bucket/*"
    ]
}
]
}

```

Sustituya `amzn-s3-demo-bucket` por el nombre de su bucket de Amazon S3.

Error: no se puede asumir el ARN del rol

Solución: no AWS IoT SiteWise puede asumir la función de IAM en su nombre. Asegúrese de que su rol confíe en el siguiente servicio: `iotsitewise.amazonaws.com`. Para obtener más información, consulte [No puedo asumir un rol](#) en la Guía del usuario de IAM.

Error: no se ha podido acceder al bucket de Amazon S3 entre regiones

Solución: el bucket de Amazon S3 que especificó se encuentra en una AWS región diferente. Asegúrese de que su depósito de Amazon S3 y AWS IoT SiteWise sus activos estén en la misma región.

Rutas de archivos y esquemas de datos guardados en el nivel inactivo

AWS IoT SiteWise almacena los datos en la capa fría mediante la replicación de series temporales, incluidas las mediciones, las métricas, las transformaciones y los agregados, así como las definiciones de activos y modelos de activos. A continuación se describen las rutas de los archivos y los esquemas de datos que se envían al nivel inactivo.

Temas

- [Datos del equipo \(mediciones\)](#)
- [Métricas, transformaciones y agregados](#)
- [Metadatos de los activos](#)

- [Metadatos de jerarquía de los activos](#)
- [Almacenamiento de archivos índice de datos](#)

Datos del equipo (mediciones)

AWS IoT SiteWise exporta los datos del equipo (mediciones) a la capa fría una vez cada seis horas. Los datos sin procesar se guardan en el nivel inactivo en formato [Apache AVRO](#) (.avro).

Ruta de archivo

AWS IoT SiteWise almacena los datos del equipo (mediciones) en la capa fría mediante la siguiente plantilla.

```
{keyPrefix}/raw/startYear={startYear}/startMonth={startMonth}/startDay={startDay}/
seriesBucket={seriesBucket}/raw_{timeseriesId}_{startTimestamp}_{quality}.avro
```

Cada ruta de archivo a datos sin procesar en Amazon S3 contiene los siguientes componentes.

Componente de ruta	Description (Descripción)
keyPrefix	El prefijo de Amazon S3 que especificó en la configuración AWS IoT SiteWise de almacenamiento. Amazon S3 utiliza el prefijo como nombre de carpeta en el bucket.
raw	La carpeta que almacena los datos de serie temporal del equipo (mediciones). La carpeta raw se guarda en la carpeta de prefijos.
seriesBucket	Un número hexadecimal entre 00 y ff. Este número se deriva de timeSeriesId . Esta partición se utiliza para aumentar el rendimiento cuando se AWS IoT SiteWise escribe en la capa fría. Cuando se utiliza Amazon Athena para ejecutar consultas, la partición puede servir para realizar particiones refinadas a fin de mejorar la precisión de las consultas.

Componente de ruta	Description (Descripción)
	<p><code>seriesBucket</code> y <code>timeSeriesBucket</code> son el mismo número en los metadatos del activo.</p>
<code>startYear</code>	El año de la hora de inicio exclusiva asociada a los datos de serie temporal.
<code>startMonth</code>	El mes de la hora de inicio exclusiva asociada a los datos de serie temporal.
<code>startDay</code>	El día del mes de la hora de inicio exclusiva asociada a los datos de serie temporal.
<code>fileName</code>	<p>El nombre del archivo utiliza el carácter de subrayado (<code>_</code>) como delimitador para separar lo siguiente:</p> <ul style="list-style-type: none"> • El prefijo <code>raw</code>. • El valor <code>timeSeriesId</code> . • La marca temporal de fecha de inicio exclusiva asociada a los datos de serie temporal. • La calidad de datos. Valores aceptados: <code>GOOD</code>, <code>BAD</code> y <code>UNCERTAIN</code> . Para obtener más información, consulta AssetPropertyValue en la AWS IoT SiteWise Referencia de la API de . <p>El archivo se guarda en el formato <code>.avro</code> mediante la compresión Snappy.</p>

Example ruta del archivo a los datos sin procesar en el nivel inactivo

```
keyPrefix/raw/startYear=2021/startMonth=1/startDay=2/seriesBucket=a2/
raw_7020c8e2-e6db-40fa-9845-ed0dd4c77d_95e63da7-d34e-43e1-
bc6f-1b490154b07a_1609577700_G00D.avro
```

Fields

El esquema de datos sin procesar que se exporta al nivel inactivo contiene los siguientes campos.

AWS IoT SiteWise aconseja a los clientes que implementen el soporte para la evolución de esquemas en los sistemas que leen datos sin procesar de la capa fría, ya que es posible que se introduzcan campos adicionales en el futuro.

Los datos nulos se representan cuando todos los campos de valores son nulos. Sin embargo, los clientes seguirán recibiendo el tipo de datos correcto cuando realicen consultas. AWS IoT SiteWise APIs

Nombre del campo	Tipos admitidos	Tipo predeterminado	Description (Descripción)
<code>seriesId</code>	<code>string</code>	N/A	El identificador que identifica los datos de serie temporal del equipo (mediciones). Puede usar este campo para unir datos sin procesar y metadatos de activos en las consultas.
<code>timeInSeconds</code>	<code>long</code>	N/A	La marca temporal, en segundos, en formato de tiempo Unix. Los datos fraccionarios de nanosegundos los proporciona <code>offsetInNanos</code> .
<code>offsetInNanos</code>	<code>long</code>	N/A	El desfase de nanosegundos procedente de <code>timeInSeconds</code> .

Nombre del campo	Tipos admitidos	Tipo predeterminado	Description (Descripción)
quality	string	N/A	La calidad del valor de la serie temporal.
doubleValue	double o null	null	Datos de serie temporal de tipo doble (número de punto flotante).
stringValue	string o null	null	Datos de serie temporal de tipo cadena (secuencia de caracteres).
integerValue	int o null	null	Datos de serie temporal de tipo entero (número entero).
booleanValue	boolean o null	null	Datos de serie temporal de tipo booleano (verdadero o falso).
jsonValue	string o null	null	Datos de serie temporal de tipo JSON (tipos de datos complejos almacenados como una cadena).

Nombre del campo	Tipos admitidos	Tipo predeterminado	Description (Descripción)
recordVersion	long o null	null	El número de versión para el registro. Puede usar el número de versión para seleccionar el registro más reciente. Los registros más recientes tienen números de versión más grandes.

Example datos sin procesar en el nivel inactivo

```

{"seriesId":"e9687d2a-0dbe-4f65-9ed6-6f443cba41f7_95e63da7-d34e-43e1-bc6f-1b490154b07a","timeInSeconds":1625675887,"offsetInNanos":0,"quality":"GOOD","doubleValue":{"double":0.75},"stringValue":null,"integerValue":null,"booleanValue":null,"jsonValue":null,"recordVersion":1}
{"seriesId":"e9687d2a-0dbe-4f65-9ed6-6f443cba41f7_95e63da7-d34e-43e1-bc6f-1b490154b07a","timeInSeconds":1625675889,"offsetInNanos":0,"quality":"GOOD","doubleValue":{"double":0.69},"stringValue":null,"integerValue":null,"booleanValue":null,"jsonValue":null,"recordVersion":2}
{"seriesId":"e9687d2a-0dbe-4f65-9ed6-6f443cba41f7_95e63da7-d34e-43e1-bc6f-1b490154b07a","timeInSeconds":1625675890,"offsetInNanos":0,"quality":"GOOD","doubleValue":{"double":0.66},"stringValue":null,"integerValue":null,"booleanValue":null,"jsonValue":null,"recordVersion":3}
{"seriesId":"e9687d2a-0dbe-4f65-9ed6-6f443cba41f7_95e63da7-d34e-43e1-bc6f-1b490154b07a","timeInSeconds":1625675891,"offsetInNanos":0,"quality":"GOOD","doubleValue":{"double":0.92},"stringValue":null,"integerValue":null,"booleanValue":null,"jsonValue":null,"recordVersion":4}
{"seriesId":"e9687d2a-0dbe-4f65-9ed6-6f443cba41f7_95e63da7-d34e-43e1-bc6f-1b490154b07a","timeInSeconds":1625675892,"offsetInNanos":0,"quality":"GOOD","doubleValue":{"double":0.73},"stringValue":null,"integerValue":null,"booleanValue":null,"jsonValue":null,"recordVersion":5}

```

Métricas, transformaciones y agregados

AWS IoT SiteWise exporta métricas, transformaciones y agregados a la capa fría una vez cada seis horas. Las métricas, las transformaciones y los agregados se guardan en el nivel inactivo en el formato [Apache AVRO](#) (.avro).

Ruta de archivo

AWS IoT SiteWise almacena las métricas, las transformaciones y los agregados en la capa fría mediante la siguiente plantilla.

```
{keyPrefix}/agg/startYear={startYear}/startMonth={startMonth}/startDay={startDay}/seriesBucket={seriesBucket}/agg_{timeseriesId}_{startTimestamp}_{quality}.avro
```

Cada ruta de archivo a las métricas, las transformaciones y los agregados en Amazon S3 contiene los siguientes componentes.

Componente de ruta	Description (Descripción)
keyPrefix	El prefijo de Amazon S3 que especificó en la configuración AWS IoT SiteWise de almacenamiento. Amazon S3 utiliza el prefijo como nombre de carpeta en el bucket.
agg	La carpeta que almacena los datos de serie temporal de las métricas. La carpeta agg se guarda en la carpeta de prefijos.
seriesBucket	<p>Un número hexadecimal entre 00 y ff. Este número se deriva de <code>timeSeriesId</code>. Esta partición se utiliza para aumentar el rendimiento cuando se AWS IoT SiteWise escribe en la capa fría. Cuando se utiliza Amazon Athena para ejecutar consultas, la partición puede servir para realizar particiones refinadas a fin de mejorar la precisión de las consultas.</p> <p><code>seriesBucket</code> y <code>timeSeriesBucket</code> son el mismo número en los metadatos del activo.</p>
startYear	El año de la hora de inicio exclusiva asociada a los datos de serie temporal.
startMonth	El mes de la hora de inicio exclusiva asociada a los datos de serie temporal.

Componente de ruta	Description (Descripción)
<code>startDay</code>	El día del mes de la hora de inicio exclusiva asociada a los datos de serie temporal.
<code>fileName</code>	<p>El nombre del archivo utiliza el carácter de subrayado (<code>_</code>) como delimitador para separar lo siguiente:</p> <ul style="list-style-type: none"> • El prefijo <code>raw</code>. • El valor <code>timeSeriesId</code> . • La marca temporal de fecha de inicio exclusiva asociada a los datos de serie temporal. • La calidad de datos. Valores aceptados: <code>GOOD</code>, <code>BAD</code> y <code>UNCERTAIN</code> . Para obtener más información, consulta AssetPropertyValue en la AWS IoT SiteWise Referencia de la API de . <p>El archivo se guarda en el formato <code>.avro</code> mediante la compresión Snappy.</p>

Example ruta del archivo a las métricas en el nivel inactivo

```
keyPrefix/agg/startYear=2021/startMonth=1/startDay=2/seriesBucket=a2/agg_7020c8e2-e6db-40fa-9845-ed0dddd4c77d_95e63da7-d34e-43e1-bc6f-1b490154b07a_1609577700_G00D.avro
```

Fields

El esquema de las métricas, las transformaciones y los agregados que se exportan al nivel inactivo contiene los siguientes campos.

Nombre del campo	Tipos admitidos	Tipo predeterminado	Description (Descripción)
<code>seriesId</code>	<code>string</code>	N/A	El ID que identifica los datos de serie temporal procedentes del equipo, de las métricas o de las transformaciones. Puede usar este campo para unir datos sin procesar y metadatos de activos en las consultas.
<code>timeInSeconds</code>	<code>long</code>	N/A	La marca temporal, en segundos, en formato de tiempo Unix. Los datos fraccionarios de nanosegundos los proporciona <code>offsetInNanos</code> .
<code>offsetInNanos</code>	<code>long</code>	N/A	El desfase de nanosegundos procedente de <code>timeInSeconds</code> .
<code>quality</code>	<code>string</code>	N/A	La calidad con la que se filtran los datos de los activos.
<code>resolution</code>	<code>string</code>	N/A	El intervalo de tiempo durante el que se van a agregar los datos.

Nombre del campo	Tipos admitidos	Tipo predeterminado	Description (Descripción)
count	double o null	null	El número total de puntos de datos para las variables dadas durante el intervalo de tiempo actual.
average	double o null	null	La media de los valores de las variables dadas durante el intervalo de tiempo actual.
min	double o null	null	El mínimo de los valores de las variables dadas durante el intervalo de tiempo actual.
max	boolean o null	null	El máximo de los valores de las variables dadas durante el intervalo de tiempo actual.
sum	string o null	null	La suma de los valores de las variables dadas durante el intervalo de tiempo actual.

Nombre del campo	Tipos admitidos	Tipo predeterminado	Description (Descripción)
recordVersion	long o null	null	El número de versión para el registro. Puede usar el número de versión para seleccionar el registro más reciente. Los registros más recientes tienen números de versión más grandes.

Example Datos métricos en el nivel inactivo

```

{"seriesId":"f74c2828-5317-4df3-ba16-6d41b5bcb531","timeInSeconds":1637334060,"offsetInNanos":0,"quality":"GOOD","resolution":1000,"sum":{"double":16.0},"min":{"double":1.0},"max":{"double":31.0},"recordVersion":null}
{"seriesId":"f74c2828-5317-4df3-ba16-6d41b5bcb531","timeInSeconds":1637334120,"offsetInNanos":0,"quality":"GOOD","resolution":1000,"sum":{"double":46.0},"min":{"double":32.0},"max":{"double":60.0},"recordVersion":null}
{"seriesId":"f74c2828-5317-4df3-ba16-6d41b5bcb531","timeInSeconds":1637334540,"offsetInNanos":0,"quality":"GOOD","resolution":1000,"sum":{"double":16.0},"min":{"double":1.0},"max":{"double":31.0},"recordVersion":null}
{"seriesId":"f74c2828-5317-4df3-ba16-6d41b5bcb531","timeInSeconds":1637334600,"offsetInNanos":0,"quality":"GOOD","resolution":1000,"sum":{"double":46.0},"min":{"double":32.0},"max":{"double":60.0},"recordVersion":null}
{"seriesId":"f74c2828-5317-4df3-ba16-6d41b5bcb531","timeInSeconds":1637335020,"offsetInNanos":0,"quality":"GOOD","resolution":1000,"sum":{"double":16.0},"min":{"double":1.0},"max":{"double":31.0},"recordVersion":null}

```

Metadatos de los activos

Al habilitar AWS IoT SiteWise la exportación de datos a la capa fría por primera vez, los metadatos de los activos se exportan a la capa fría. Tras la configuración inicial, AWS IoT SiteWise exporta los metadatos de los activos al nivel solo cuando se cambian las definiciones del modelo de activos o las definiciones de activos. Los metadatos de activos se guardan en el nivel de acceso inactivo en el formato JSON delimitado por nuevas líneas (.ndjson).

Ruta de archivo

AWS IoT SiteWise almacena los metadatos de los activos en la capa fría mediante la siguiente plantilla.

```
{keyPrefix}/asset_metadata/asset_{assetId}.ndjson
```

Cada ruta de archivo a los metadatos de los activos en el nivel inactivo contiene los siguientes componentes.

Componente de ruta	Description (Descripción)
keyPrefix	El prefijo de Amazon S3 que especificó en la configuración de almacenamiento AWS IoT SiteWise s. Amazon S3 utiliza el prefijo como nombre de carpeta en el bucket.
asset_metadata	La carpeta que almacena los metadatos de los activos. La carpeta asset_metadata se guarda en la carpeta de prefijos.
fileName	<p>El nombre del archivo utiliza el carácter de subrayado (_) como delimitador para separar lo siguiente:</p> <ul style="list-style-type: none"> • El prefijo asset. • El valor assetId. <p>El archivo se guarda en el formato .ndjson.</p>

Example ruta del archivo a los metadatos de los activos en el nivel inferior

keyPrefix/asset_metadata/asset_35901915-d476-4dca-8637-d9ed4df939ed.ndjson

Fields

El esquema de metadatos de activos que se exporta al nivel inactivo contiene los siguientes campos.

Nombre del campo	Description (Descripción)
assetId	El ID del activo.
assetName	Nombre del activo.
assetExternalId	El ID externo del recurso.
assetModelId	Id. del modelo de activos usado para crear el activo.
assetModelName	El nombre del modelo del activo.
assetModelExternalId	ID externo del modelo de activos.
assetPropertyId	El ID de la propiedad del activo.
assetPropertyName	El nombre de la propiedad del activo.
assetPropertyExternalId	ID externo de la propiedad del activo.
assetPropertyDataType	El tipo de datos de la propiedad del activo.
assetPropertyUnit	La unidad que usa la propiedad del activo (por ejemplo, Newtons y RPM).
assetPropertyAlias	El alias que identifica la propiedad del activo, como una ruta de flujo de datos del servidor OPC UA (por ejemplo, /company/windfarm/3/turbine/7/temperature).

Nombre del campo	Description (Descripción)
<code>timeSeriesId</code>	El ID que identifica los datos de serie temporal procedentes del equipo, de las métricas o de las transformaciones. Puede usar este campo para unir datos sin procesar y metadatos de activos en las consultas.
<code>timeSeriesBucket</code>	<p>Un número hexadecimal entre 00 y ff. Este número se deriva de <code>timeSeriesId</code> . Esta partición se utiliza para aumentar el rendimiento cuando se AWS IoT SiteWise escribe en la capa fría. Cuando se utiliza Amazon Athena para ejecutar consultas, la partición puede servir para realizar particiones refinadas a fin de mejorar la precisión de las consultas.</p> <p><code>timeSeriesBucket</code> y <code>seriesBucket</code> son el mismo número en la ruta del archivo a los datos sin procesar.</p>
<code>assetCompositeModelId</code>	ID del modelo compuesto.
<code>assetCompositeModelExternalId</code>	ID externo del modelo compuesto.
<code>assetCompositeModelDescription</code>	La descripción del modelo compuesto.
<code>assetCompositeModelName</code>	El nombre del modelo compuesto.
<code>assetCompositeModelType</code>	El tipo del modelo compuesto. Para los modelos compuestos de alarma, este tipo es <code>AWS/ALARM</code> .
<code>assetCreationDate</code>	La fecha en que se creó el activo, en formato de tiempo UNIX.
<code>assetLastUpdateDate</code>	La fecha en que el activo se actualizó por última vez, en fecha de inicio Unix.

Nombre del campo	Description (Descripción)
assetStatusErrorCode	Código de error.
assetStatusErrorMessage	Mensaje de error.
assetStatusState	El estado actual del activo.

Example metadatos de activos en el nivel inactivo

```

{"assetId":"7020c8e2-e6db-40fa-9845-
ed0dddd4c77d","assetExternalId":null,"assetName":"Wind Turbine Asset
  2","assetModelId":"ec1d924f-f07d-444f-b072-
e2994c165d35","assetModelExternalId":null,"assetModelName":"Wind
  Turbine Asset Model","assetPropertyId":"95e63da7-d34e-43e1-
bc6f-1b490154b07a","assetPropertyExternalId":null,"assetPropertyName":"Temperature","assetPrope
Washington/Seattle/WT2/temp","timeSeriesId":"7020c8e2-e6db-40fa-9845-
ed0dddd4c77d_95e63da7-d34e-43e1-
bc6f-1b490154b07a","timeSeriesBucket":"f6","assetArn":null,"assetCompositeModelDescription":null
  {"assetId":"7020c8e2-e6db-40fa-9845-
ed0dddd4c77d","assetExternalId":null,"assetName":"Wind Turbine Asset
  2","assetModelId":"ec1d924f-f07d-444f-b072-
e2994c165d35","assetModelExternalId":null,"assetModelName":"Wind Turbine Asset
  Model","assetPropertyId":"c706d54d-4c11-42dc-9a01-63662fc697b4","assetPropertyExternalId":null
Washington/Seattle/WT2/pressure","timeSeriesId":"7020c8e2-e6db-40fa-9845-
ed0dddd4c77d_c706d54d-4c11-42dc-9a01-63662fc697b4","timeSeriesBucket":"1e","assetArn":null,"ass
  {"assetId":"7020c8e2-e6db-40fa-9845-
ed0dddd4c77d","assetExternalId":null,"assetName":"Wind Turbine Asset
  2","assetModelId":"ec1d924f-f07d-444f-b072-
e2994c165d35","assetModelExternalId":null,"assetModelName":"Wind
  Turbine Asset Model","assetPropertyId":"8cf1162f-dead-4fbe-b468-
c8e24cde9f50","assetPropertyExternalId":null,"assetPropertyName":"Max
  Temperature","assetPropertyDataType":"DOUBLE","assetPropertyUnit":null,"assetPropertyAlias":nu
e6db-40fa-9845-ed0dddd4c77d_8cf1162f-dead-4fbe-b468-
c8e24cde9f50","timeSeriesBucket":"d7","assetArn":null,"assetCompositeModelDescription":null,"as

  {"assetId":"3a5f2a22-3b37-4332-9c1c-404ea1d73fab","assetExternalId":null,"assetName":"BatchAss
ebc75e75e827","assetModelExternalId":null,"assetModelName":"FlashTestAssetModelDouble","assetPr
b410-
ab401a9176ed","assetPropertyExternalId":null,"assetPropertyName":"measurementProperty","assetPr
ae89-

```

```
ff316f5ff8aa", "timeSeriesBucket": "af", "assetArn": null, "assetCompositeModelDescription": null, "as
```

Metadatos de jerarquía de los activos

Cuando habilita AWS IoT SiteWise guardar datos en la capa inactiva por primera vez, los metadatos de la jerarquía de activos se exportan a la capa fría. Tras la configuración inicial, AWS IoT SiteWise exporta los metadatos de la jerarquía de activos a la capa fría solo cuando se realizan cambios en el modelo de activos o en las definiciones de activos. Los metadatos de la jerarquía de activos se guardan en el nivel inactivo en el formato JSON delimitado por nuevas líneas (.ndjson).

Al llamar a la [DescribeAsset](#) API, se recupera un identificador externo de la jerarquía, el activo de destino o el activo de origen.

Ruta de archivo

AWS IoT SiteWise almacena los metadatos de la jerarquía de activos en la capa fría mediante la siguiente plantilla.

```
{keyPrefix}/asset_hierarchy_metadata/{parentAssetId}_{hierarchyId}.ndjson
```

Cada ruta de archivo a los metadatos de jerarquía de los activos en el nivel inactivo contiene los siguientes componentes.

Componente de ruta	Description (Descripción)
keyPrefix	El prefijo de Amazon S3 que especificó en la configuración AWS IoT SiteWise de almacenamiento. Amazon S3 utiliza el prefijo como nombre de carpeta en el bucket.
asset_hierarchy_metadata	La carpeta que almacena los metadatos de jerarquía de los activos. La carpeta <code>asset_hierarchy_metadata</code> se guarda en la carpeta de prefijos.

Componente de ruta	Description (Descripción)
fileName	<p>El nombre del archivo utiliza el carácter de subrayado (_) como delimitador para separar lo siguiente:</p> <ul style="list-style-type: none"> • El valor parentAssetId . • El valor hierarchyId . <p>El archivo se guarda en el formato .ndjson.</p>

Example ruta del archivo a los metadatos de la jerarquía de activos en el nivel inactivo

keyPrefix/asset_hierarchy_metadata/35901915-d476-4dca-8637-d9ed4df939ed_c5b3ced8-589a-48c7-9998-cdcccfc9747a0.ndjson

Fields

El esquema de los metadatos de la jerarquía de activos que se exporta al nivel inactivo contiene los siguientes campos.

Nombre del campo	Description (Descripción)
sourceAssetId	El ID del activo de origen en esta relación de activos.
targetAssetId	El ID del activo de destino en esta relación de activos.
hierarchyId	El ID de la jerarquía.
associationType	<p>El tipo de asociación de esta relación de activos.</p> <p>El valor debe ser CHILD. El activo de destino es una entidad secundaria del activo de origen.</p>

Example los metadatos de jerarquía de los activos en el nivel inactivo

```
{
  "sourceAssetId": "80388e72-2284-44fb-9c89-bfbaf0dfedd2",
  "targetAssetId": "2b866c25-0c74-4750-bdf5-b73683c8a2a2",
  "hierarchyId": "bbed9f59-0412-4585-a61d-6044db526aee",
  "associationType": "CHILD"
}
{
  "sourceAssetId": "80388e72-2284-44fb-9c89-bfbaf0dfedd2",
  "targetAssetId": "6b51246e-984d-460d-bc0b-470ea47d1e31",
  "hierarchyId": "bbed9f59-0412-4585-a61d-6044db526aee",
  "associationType": "CHILD"
}
```

Para ver los datos en el nivel inactivo

1. Vaya a la [consola de Amazon S3](#).
2. En el panel de navegación, elija Buckets y, a continuación, elija el bucket de Amazon S3.
3. Navegue hasta la carpeta que contiene los datos sin procesar, los metadatos de los activos o los metadatos de jerarquía de los activos.
4. Seleccione los archivos y, a continuación, en Acciones, elija Descargar.

Almacenamiento de archivos índice de datos

AWS IoT SiteWise utiliza estos archivos para optimizar el rendimiento de las consultas de datos. Aparecen en u bucket de Amazon S3, pero no es necesario que los utilice.

Ruta de archivo

AWS IoT SiteWise almacena los archivos de índice de datos en la capa fría mediante la siguiente plantilla.

```
keyPrefix/index/series=timeseriesId/startYear=startYear/startMonth=startMonth/
startDay=startDay/index_timeseriesId_startTimestamp_quality
```

Example ruta del archivo al archivo índice de almacenamiento de datos

```
keyPrefix/index/series=7020c8e2-e6db-40fa-9845-ed0dddd4c77d_95e63da7-
d34e-43e1-bc6f-1b490154b07a/startYear=2022/startMonth=02/startDay=03/
index_7020c8e2-e6db-40fa-9845-ed0dddd4c77d_95e63da7-d34e-43e1-
bc6f-1b490154b07a_1643846400_G00D
```

Ejemplos de código para AWS IoT SiteWise usar AWS SDKs

Los siguientes ejemplos de código muestran cómo usarlo AWS IoT SiteWise con un kit de desarrollo de AWS software (SDK).

Los conceptos básicos son ejemplos de código que muestran cómo realizar las operaciones esenciales dentro de un servicio.

Las acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Mientras las acciones muestran cómo llamar a las distintas funciones de servicio, es posible ver las acciones en contexto en los escenarios relacionados.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte [Uso de este servicio con un AWS SDK](#). En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Ejemplos de código

- [Ejemplos básicos de AWS IoT SiteWise uso AWS SDKs](#)
 - [Hola AWS IoT SiteWise](#)
 - [Aprenda los conceptos básicos del AWS IoT SiteWise uso de un AWS SDK](#)
 - [Acciones de AWS IoT SiteWise uso AWS SDKs](#)
 - [Úselo BatchPutAssetPropertyValue con un AWS SDK o CLI](#)
 - [Úselo CreateAsset con un AWS SDK o CLI](#)
 - [Úselo CreateAssetModel con un AWS SDK o CLI](#)
 - [Úselo CreateGateway con un AWS SDK o CLI](#)
 - [Úselo DeleteAsset con un AWS SDK o CLI](#)
 - [Úselo DeleteAssetModel con un AWS SDK o CLI](#)
 - [Úselo DeleteGateway con un AWS SDK o CLI](#)
 - [Úselo DescribeAssetModel con un AWS SDK o CLI](#)
 - [Úselo DescribeGateway con un AWS SDK o CLI](#)
 - [Úselo GetAssetPropertyValue con un AWS SDK o CLI](#)
 - [ListAssetModelPropertiesÚselo con un AWS SDK](#)
 - [Úselo ListAssetModels con un AWS SDK o CLI](#)

Ejemplos básicos de AWS IoT SiteWise uso AWS SDKs

Los siguientes ejemplos de código muestran cómo utilizar los conceptos básicos de AWS IoT SiteWise with AWS SDKs.

Ejemplos

- [Hola AWS IoT SiteWise](#)
- [Aprenda los conceptos básicos del AWS IoT SiteWise uso de un AWS SDK](#)
- [Acciones de AWS IoT SiteWise uso AWS SDKs](#)
 - [Úselo BatchPutAssetPropertyValue con un AWS SDK o CLI](#)
 - [Úselo CreateAsset con un AWS SDK o CLI](#)
 - [Úselo CreateAssetModel con un AWS SDK o CLI](#)
 - [Úselo CreateGateway con un AWS SDK o CLI](#)
 - [Úselo DeleteAsset con un AWS SDK o CLI](#)
 - [Úselo DeleteAssetModel con un AWS SDK o CLI](#)
 - [Úselo DeleteGateway con un AWS SDK o CLI](#)
 - [Úselo DescribeAssetModel con un AWS SDK o CLI](#)
 - [Úselo DescribeGateway con un AWS SDK o CLI](#)
 - [Úselo GetAssetPropertyValue con un AWS SDK o CLI](#)
 - [ListAssetModelPropertiesÚselo con un AWS SDK](#)
 - [Úselo ListAssetModels con un AWS SDK o CLI](#)

Hola AWS IoT SiteWise

En los siguientes ejemplos de código se muestra cómo empezar a utilizar AWS IoT SiteWise.

Java

SDK para Java 2.x

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
public class HelloSitewise {
    private static final Logger logger =
    LoggerFactory.getLogger(HelloSitewise.class);
    public static void main(String[] args) {
        fetchAssetModels();
    }

    /**
     * Fetches asset models using the provided {@link IoTSiteWiseAsyncClient}.
     */
    public static void fetchAssetModels() {
        IoTSiteWiseAsyncClient siteWiseAsyncClient =
    IoTSiteWiseAsyncClient.create();
        ListAssetModelsRequest assetModelsRequest =
    ListAssetModelsRequest.builder()
            .assetModelTypes(AssetModelType.ASSET_MODEL)
            .build();

        // Asynchronous paginator - process paginated results.
        ListAssetModelsPublisher listModelsPaginator =
    siteWiseAsyncClient.listAssetModelsPaginator(assetModelsRequest);
        CompletableFuture<Void> future = listModelsPaginator.subscribe(response -
> {
            response.assetModelSummaries().forEach(assetSummary ->
                logger.info("Asset Model Name: {} ", assetSummary.name())
            );
        });

        // Wait for the asynchronous operation to complete
        future.join();
    }
}
```

- Para obtener más información sobre la API, consulta [ListAssetModels](#) la Referencia AWS SDK for Java 2.x de la API.

JavaScript

SDK para JavaScript (v3)

Note

Hay más información. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
import {
  paginateListAssetModels,
  IoTSiteWiseClient,
} from "@aws-sdk/client-iotsitewise";

// Call ListDocuments and display the result.
export const main = async () => {
  const client = new IoTSiteWiseClient();
  const listAssetModelsPaginated = [];
  console.log(
    "Hello, AWS Systems Manager! Let's list some of your documents:\n",
  );
  try {
    // The paginate function is a wrapper around the base command.
    const paginator = paginateListAssetModels({ client }, { maxResults: 5 });
    for await (const page of paginator) {
      listAssetModelsPaginated.push(...page.assetModelSummaries);
    }
  } catch (caught) {
    console.error(`There was a problem saying hello: ${caught.message}`);
    throw caught;
  }
  for (const { name, creationDate } of listAssetModelsPaginated) {
    console.log(`${name} - ${creationDate}`);
  }
};

// Call function if run directly.
import { fileURLToPath } from "node:url";
if (process.argv[1] === fileURLToPath(import.meta.url)) {
  main();
}
```

- Para obtener más información sobre la API, consulta [ListAssetModels](#) la Referencia AWS SDK para JavaScript de la API.

Python

SDK para Python (Boto3)

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
import boto3

def hello_iot_sitewise(iot_sitewise_client):
    """
    Use the AWS SDK for Python (Boto3) to create an AWS IoT SiteWise
    client and list the asset models in your account.
    This example uses the default settings specified in your shared credentials
    and config files.

    :param iot_sitewise_client: A Boto3 AWS IoT SiteWise Client object. This
    object wraps
                                the low-level AWS IoT SiteWise service API.
    """
    print("Hello, AWS IoT SiteWise! Let's list some of your asset models:\n")
    paginator = iot_sitewise_client.get_paginator("list_asset_models")
    page_iterator = paginator.paginate(PaginationConfig={"MaxItems": 10})

    asset_model_names: [str] = []
    for page in page_iterator:
        for asset_model in page["assetModelSummaries"]:
            asset_model_names.append(asset_model["name"])

    print(f"{len(asset_model_names)} asset model(s) retrieved.")
    for asset_model_name in asset_model_names:
        print(f"\t{asset_model_name}")
```

```
if __name__ == "__main__":  
    hello_iot_sitewise(boto3.client("iotsitewise"))
```

- Para obtener más información sobre la API, consulta [ListAssetModels](#) la AWS Referencia de API de SDK for Python (Boto3).

SAP ABAP

SDK para SAP ABAP

Note

Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
TRY.  
    oo_result = lo_ios->listassetmodels( ). " oo_result is returned for  
testing purposes. "  
    DATA(lt_asset_models) = oo_result->get_assetmodelsummaries( ).  
    MESSAGE 'Retrieved list of asset models.' TYPE 'I'.  
CATCH /aws1/cx_rt_generic.  
    MESSAGE 'Unable to list asset models.' TYPE 'E'.  
ENDTRY.
```

- Para obtener más información sobre la API, consulte [ListAssetModels](#) la referencia sobre la API ABAP del AWS SDK para SAP.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulte [Uso de este servicio con un AWS SDK](#). En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Aprenda los conceptos básicos del AWS IoT SiteWise uso de un AWS SDK

En el siguiente ejemplo de código, se muestra cómo:

- Cree un modelo AWS IoT SiteWise de activos.
- Cree un AWS IoT SiteWise activo.
- Recuperar los valores de ID de la propiedad.
- Envía datos a un AWS IoT SiteWise activo.
- Recupera el valor de la propiedad del AWS IoT SiteWise activo.
- Crea un AWS IoT SiteWise portal.
- Cree una AWS IoT SiteWise puerta de enlace.
- Describa la AWS IoT SiteWise puerta de enlace.
- Elimine los AWS IoT SiteWise activos.

Java

SDK para Java 2.x

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

Ejecute un escenario interactivo que demuestre AWS IoT SiteWise las funciones.

```
public class SitewiseScenario {

    public static final String DASHES = new String(new char[80]).replace("\0",
"-");

    private static final Logger logger =
LoggerFactory.getLogger(SitewiseScenario.class);
    static Scanner scanner = new Scanner(System.in);

    private static final String ROLES_STACK = "RoleSitewise";

    static SitewiseActions sitewiseActions = new SitewiseActions();

    public static void main(String[] args) throws Throwable {
        Scanner scanner = new Scanner(System.in);
        String contactEmail = "user@mydomain.com"; // Change email address.
```

```
String assetModelName = "MyAssetModel1";  
String assetName = "MyAsset1" ;  
String portalName = "MyPortal1" ;  
String gatewayName = "MyGateway1" ;  
String myThing = "MyThing1" ;
```

```
logger.info(""
```

```
    AWS IoT SiteWise is a fully managed software-as-a-service (SaaS)  
that
```

```
    makes it easy to collect, store, organize, and monitor data from  
industrial equipment and processes.
```

```
    It is designed to help industrial and manufacturing organizations  
collect data from their equipment and
```

```
    processes, and use that data to make informed decisions about their  
operations.
```

```
    One of the key features of AWS IoT SiteWise is its ability to connect  
to a wide range of industrial
```

```
    equipment and systems, including programmable logic controllers  
(PLCs), sensors, and other
```

```
    industrial devices. It can collect data from these devices and  
organize it into a unified data model,
```

```
    making it easier to analyze and gain insights from the data. AWS IoT  
SiteWise also provides tools for
```

```
    visualizing the data, setting up alarms and alerts, and generating  
reports.
```

```
    Another key feature of AWS IoT SiteWise is its ability to scale to  
handle large volumes of data.
```

```
    It can collect and store data from thousands of devices and process  
millions of data points per second,
```

```
    making it suitable for large-scale industrial operations.
```

```
Additionally, AWS IoT SiteWise is designed
```

```
    to be secure and compliant, with features like role-based access  
controls, data encryption,
```

```
    and integration with other AWS services for additional security and  
compliance features.
```

```
    Let's get started...
```

```
    "");
```

```
waitForInputToContinue(scanner);
```

```
logger.info(DASHES);
```

```

        try {
            runScenario(assetModelName, assetName, portalName, contactEmail,
gatewayName, myThing);
        } catch (RuntimeException e) {
            logger.info(e.getMessage());
        }
    }

    public static void runScenario(String assetModelName, String assetName,
String portalName, String contactEmail, String gatewayName, String myThing)
throws Throwable {
        logger.info("Use AWS CloudFormation to create an IAM role that is
required for this scenario.");
        CloudFormationHelper.deployCloudFormationStack(ROLES_STACK);
        Map<String, String> stackOutputs =
CloudFormationHelper.getStackOutputsAsync(ROLES_STACK).join();
        String iamRole = stackOutputs.get("SitewiseRoleArn");
        logger.info("The ARN of the IAM role is {}", iamRole);
        logger.info(DASHES);

        logger.info(DASHES);
        logger.info("1. Create an AWS SiteWise Asset Model");
        logger.info("""
            An AWS IoT SiteWise Asset Model is a way to represent the physical
assets, such as equipment,
                processes, and systems, that exist in an industrial environment.
This model provides a structured and
                hierarchical representation of these assets, allowing users to
define the relationships and properties
                of each asset.

            This scenario creates two asset model properties: temperature and
humidity.

            """);
        waitForInputToContinue(scanner);
        String assetModelId = null;
        try {
            CreateAssetModelResponse response =
sitewiseActions.createAssetModelAsync(assetModelName).join();
            assetModelId = response.assetModelId();
            logger.info("Asset Model successfully created. Asset Model ID: {}. ",
assetModelId);
        } catch (CompletionException ce) {
            Throwable cause = ce.getCause();

```

```

        if (cause instanceof ResourceAlreadyExistsException) {
            try {
                assetModelId =
sitewiseActions.getAssetModelIdAsync(assetModelName).join();
                logger.info("The Asset Model {} already exists. The id of the
existing model is {}. Moving on...", assetModelName, assetModelId);
            } catch (CompletionException cex) {
                logger.error("Exception thrown acquiring the asset model id:
{}", cex.getCause().getCause(), cex);
                return;
            }
        } else {
            logger.info("An unexpected error occurred: " +
cause.getMessage(), cause);
            return;
        }
    }
    waitForInputToContinue(scanner);

    logger.info(DASHES);
    logger.info("2. Create an AWS IoT SiteWise Asset");
    logger.info("""
        The IoT SiteWise model that we just created defines the structure
and metadata for your physical assets.
        Now we create an asset from the asset model.

        """);
    logger.info("Let's wait 30 seconds for the asset to be ready.");
    countdown(30);
    waitForInputToContinue(scanner);
    String assetId;
    try {
        CreateAssetResponse response =
sitewiseActions.createAssetAsync(assetName, assetModelId).join();
        assetId = response.assetId();
        logger.info("Asset created with ID: {}", assetId);
    } catch (CompletionException ce) {
        Throwable cause = ce.getCause();
        if (cause instanceof ResourceNotFoundException) {
            logger.info("The asset model id was not found: {}",
cause.getMessage(), cause);
        } else {
            logger.info("An unexpected error occurred: {}",
cause.getMessage(), cause);

```

```

    }
    return;
}
waitForInputToContinue(scanner);
logger.info(DASHES);

logger.info(DASHES);
logger.info("3. Retrieve the property ID values");
logger.info(""""
    To send data to an asset, we need to get the property ID values. In
this scenario, we access the
    temperature and humidity property ID values.
    """);
waitForInputToContinue(scanner);
Map<String, String> propertyIds = null;
try {
    propertyIds = sitewiseActions.getPropertyIds(assetModelId).join();
} catch (CompletionException ce) {
    Throwable cause = ce.getCause();
    if (cause instanceof IoTSiteWiseException) {
        logger.error("IoTSiteWiseException occurred: {}",
cause.getMessage(), ce);
    } else {
        logger.error("An unexpected error occurred: {}",
cause.getMessage(), ce);
    }
    return;
}
String humPropId = propertyIds.get("Humidity");
logger.info("The Humidity property Id is {}", humPropId);
String tempPropId = propertyIds.get("Temperature");
logger.info("The Temperature property Id is {}", tempPropId);

waitForInputToContinue(scanner);
logger.info(DASHES);

logger.info(DASHES);
logger.info("4. Send data to an AWS IoT SiteWise Asset");
logger.info(""""
    By sending data to an IoT SiteWise Asset, you can aggregate data
from
    multiple sources, normalize the data into a standard format, and
store it in a

```

centralized location. This makes it easier to analyze and gain insights from the data.

In this example, we generate sample temperature and humidity data and send it to the AWS IoT SiteWise asset.

```

        """);
    waitForInputToContinue(scanner);
    try {
        sitewiseActions.sendDataToSiteWiseAsync(assetId, tempPropId,
        humPropId).join();
        logger.info("Data sent successfully.");
    } catch (CompletionException ce) {
        Throwable cause = ce.getCause();
        if (cause instanceof ResourceNotFoundException) {
            logger.error("The AWS resource was not found: {}",
            cause.getMessage(), cause);
        } else {
            logger.error("An unexpected error occurred: {}",
            cause.getMessage(), cause);
        }
        return;
    }
    waitForInputToContinue(scanner);
    logger.info(DASHES);

    logger.info(DASHES);
    logger.info("5. Retrieve the value of the IoT SiteWise Asset property");
    logger.info("""
        IoT SiteWise is an AWS service that allows you to collect, process,
        and analyze industrial data
        from connected equipment and sensors. One of the key benefits of
        reading an IoT SiteWise property
        is the ability to gain valuable insights from your industrial data.

        """);
    waitForInputToContinue(scanner);
    try {
        Double assetVal = sitewiseActions.getAssetPropValueAsync(tempPropId,
        assetId).join();
        logger.info("The property name is: {}", "Temperature");
        logger.info("The value of this property is: {}", assetVal);

        waitForInputToContinue(scanner);
    }

```

```

        assetVal = sitewiseActions.getAssetPropValueAsync(humPropId,
assetId).join();
        logger.info("The property name is: {}", "Humidity");
        logger.info("The value of this property is: {}", assetVal);
    } catch (CompletionException ce) {
        Throwable cause = ce.getCause();
        if (cause instanceof ResourceNotFoundException) {
            logger.info("The AWS resource was not found: {}",
cause.getMessage(), cause);
        } else {
            logger.info("An unexpected error occurred: {}",
cause.getMessage(), cause);
        }
        return;
    }
    waitForInputToContinue(scanner);
    logger.info(DASHES);

    logger.info(DASHES);
    logger.info("6. Create an IoT SiteWise Gateway");
    logger.info(
        """"
            IoT SiteWise Gateway serves as the bridge between industrial
equipment, sensors, and the
            cloud-based IoT SiteWise service. It is responsible for securely
collecting, processing, and
            transmitting data from various industrial assets to the IoT
SiteWise platform,
            enabling real-time monitoring, analysis, and optimization of
industrial operations.

            """);
    waitForInputToContinue(scanner);
    String gatewayId = "";
    try {
        gatewayId = sitewiseActions.createGatewayAsync(gatewayName,
myThing).join();
        logger.info("Gateway creation completed successfully. id is {}",
gatewayId );
    } catch (CompletionException ce) {
        Throwable cause = ce.getCause();
        if (cause instanceof IoTSiteWiseException siteWiseEx) {

```

```

        logger.error("IoT SiteWise error occurred: Error message: {},
Error code {}",
                    siteWiseEx.getMessage(),
siteWiseEx.awsErrorDetails().errorCode(), siteWiseEx);
    } else {
        logger.error("An unexpected error occurred: {}",
cause.getMessage());
    }
    return;
}
logger.info(DASHES);
logger.info(DASHES);

logger.info("7. Describe the IoT SiteWise Gateway");
waitForInputToContinue(scanner);
try {
    sitewiseActions.describeGatewayAsync(gatewayId)
        .thenAccept(response -> {
            logger.info("Gateway Name: {}", response.gatewayName());
            logger.info("Gateway ARN: {}", response.gatewayArn());
            logger.info("Gateway Platform: {}",
response.gatewayPlatform());
            logger.info("Gateway Creation Date: {}",
response.creationDate());
        }).join();
} catch (CompletionException ce) {
    Throwable cause = ce.getCause();
    if (cause instanceof ResourceNotFoundException notFoundException) {
        logger.error("A ResourceNotFoundException occurred: Error
message: {}, Error code {}",
                    notFoundException.getMessage(),
notFoundException.awsErrorDetails().errorCode(), notFoundException);
    } else {
        logger.error("An unexpected error occurred: {}",
cause.getMessage(), cause);
    }
    return;
}
logger.info(DASHES);

logger.info(DASHES);
logger.info("8. Delete the AWS IoT SiteWise Assets");
logger.info(
    ""

```

```
Before you can delete the Asset Model, you must delete the assets.

""");
logger.info("Would you like to delete the IoT SiteWise Assets? (y/n)");
String delAns = scanner.nextLine().trim();
if (delAns.equalsIgnoreCase("y")) {
    logger.info("You selected to delete the SiteWise assets.");

    try {
        sitewiseActions.deleteGatewayAsync(gatewayId).join();
        logger.info("Gateway {} was deleted successfully.", gatewayId);
    } catch (CompletionException ce) {
        Throwable cause = ce.getCause();
        if (cause instanceof ResourceNotFoundException notFoundException)
        {
            logger.error("A ResourceNotFoundException occurred: Error
message: {}, Error code {}",
                notFoundException.getMessage(),
notFoundException.awsErrorDetails().errorCode(), notFoundException);
        } else {
            logger.error("An unexpected error occurred: {}",
cause.getMessage());
        }
    }

    try {
        sitewiseActions.deleteAssetAsync(assetId).join();
        logger.info("Request to delete asset {} sent successfully",
assetId);
    } catch (CompletionException ce) {
        Throwable cause = ce.getCause();
        if (cause instanceof ResourceNotFoundException notFoundException)
        {
            logger.error("A ResourceNotFoundException occurred: Error
message: {}, Error code {}",
                notFoundException.getMessage(),
notFoundException.awsErrorDetails().errorCode(), notFoundException);
        } else {
            logger.error("An unexpected error occurred: {}",
cause.getMessage());
        }
    }
    logger.info("Let's wait 1 minute for the asset to be deleted.");
    countdown(60);
}
```

```

        waitForInputToContinue(scanner);
        logger.info("Delete the AWS IoT SiteWise Asset Model");
        try {
            sitewiseActions.deleteAssetModelAsync(assetModelId).join();
            logger.info("Asset model deleted successfully.");
        } catch (CompletionException ce) {
            Throwable cause = ce.getCause();
            if (cause instanceof ResourceNotFoundException notFoundException)
            {
                logger.error("A ResourceNotFoundException occurred: Error
message: {}, Error code {}",
                    notFoundException.getMessage(),
notFoundException.awsErrorDetails().errorCode(), notFoundException);
            } else {
                logger.error("An unexpected error occurred: {}",
cause.getMessage());
            }
        }
        waitForInputToContinue(scanner);

    } else {
        logger.info("The resources will not be deleted.");
    }
    logger.info(DASHES);

    logger.info(DASHES);
    CloudFormationHelper.destroyCloudFormationStack(ROLES_STACK);
    logger.info("This concludes the AWS IoT SiteWise Scenario");
    logger.info(DASHES);
}

private static void waitForInputToContinue(Scanner scanner) {
    while (true) {
        logger.info("");
        logger.info("Enter 'c' followed by <ENTER> to continue:");
        String input = scanner.nextLine();

        if (input.trim().equalsIgnoreCase("c")) {
            logger.info("Continuing with the program...");
            logger.info("");
            break;
        } else {
            logger.info("Invalid input. Please try again.");
        }
    }
}

```

```

    }
}

public static void countdown(int totalSeconds) throws InterruptedException {
    for (int i = totalSeconds; i >= 0; i--) {
        int displayMinutes = i / 60;
        int displaySeconds = i % 60;
        System.out.printf("\r%02d:%02d", displayMinutes, displaySeconds);
        Thread.sleep(1000); // Wait for 1 second
    }
    System.out.println(); // Move to the next line after countdown
    logger.info("Countdown complete!");
}
}
}

```

Una clase contenedora para los métodos AWS IoT SiteWise del SDK.

```

public class SitewiseActions {

    private static final Logger logger =
    LoggerFactory.getLogger(SitewiseActions.class);

    private static IoTSiteWiseAsyncClient ioTSiteWiseAsyncClient;

    private static IoTSiteWiseAsyncClient getAsyncClient() {
        if (ioTSiteWiseAsyncClient == null) {
            SdkAsyncHttpClient httpClient = NettyNioAsyncHttpClient.builder()
                .maxConcurrency(100)
                .connectionTimeout(Duration.ofSeconds(60))
                .readTimeout(Duration.ofSeconds(60))
                .writeTimeout(Duration.ofSeconds(60))
                .build();

            ClientOverrideConfiguration overrideConfig =
            ClientOverrideConfiguration.builder()
                .apiCallTimeout(Duration.ofMinutes(2))
                .apiCallAttemptTimeout(Duration.ofSeconds(90))
                .retryStrategy(RetryMode.STANDARD)
                .build();

            ioTSiteWiseAsyncClient = IoTSiteWiseAsyncClient.builder()
                .httpClient(httpClient)

```

```
        .overrideConfiguration(overrideConfig)
        .build();
    }
    return ioTSiteWiseAsyncClient;
}

/**
 * Creates an asset model.
 *
 * @param name the name of the asset model to create.
 * @return a {@link CompletableFuture} that represents a {@link
CreateAssetModelResponse} result. The calling code
 *         can attach callbacks, then handle the result or exception by
calling {@link CompletableFuture#join()} or
 *         {@link CompletableFuture#get()}.
 *         <p>
 *         If any completion stage in this method throws an exception, the
method logs the exception cause and keeps it
 *         available to the calling code as a {@link CompletionException}. By
calling
 *         {@link CompletionException#getCause()}, the calling code can
access the original exception.
 */
public CompletableFuture<CreateAssetModelResponse>
createAssetModelAsync(String name) {
    PropertyType humidity = PropertyType.builder()
        .measurement(Measurement.builder().build())
        .build();

    PropertyType temperaturePropertyType = PropertyType.builder()
        .measurement(Measurement.builder().build())
        .build();

    AssetModelPropertyDefinition temperatureProperty =
AssetModelPropertyDefinition.builder()
        .name("Temperature")
        .dataType(PropertyDataType.DOUBLE)
        .type(temperaturePropertyType)
        .build();

    AssetModelPropertyDefinition humidityProperty =
AssetModelPropertyDefinition.builder()
        .name("Humidity")
```

```

        .dataType(PropertyDataType.DOUBLE)
        .type(humidity)
        .build();

        CreateAssetModelRequest createAssetModelRequest =
CreateAssetModelRequest.builder()
        .assetModelName(name)
        .assetModelDescription("This is my asset model")
        .assetModelProperties(temperatureProperty, humidityProperty)
        .build();

        return getAsyncClient().createAssetModel(createAssetModelRequest)
            .whenComplete((response, exception) -> {
                if (exception != null) {
                    logger.error("Failed to create asset model: {}",
exception.getCause().getMessage());
                }
            });
    }

/**
 * Creates an asset with the specified name and asset model Id.
 *
 * @param assetName the name of the asset to create.
 * @param assetModelId the Id of the asset model to associate with the asset.
 * @return a {@link CompletableFuture} that represents a {@link
CreateAssetResponse} result. The calling code can
 *     attach callbacks, then handle the result or exception by calling
{@link CompletableFuture#join()} or
 *     {@link CompletableFuture#get()}.
 *     <p>
 *     If any completion stage in this method throws an exception, the
method logs the exception cause and keeps it
 *     available to the calling code as a {@link CompletionException}. By
calling
 *     {@link CompletionException#getCause()}, the calling code can
access the original exception.
 */
    public CompletableFuture<CreateAssetResponse> createAssetAsync(String
assetName, String assetModelId) {
        CreateAssetRequest createAssetRequest = CreateAssetRequest.builder()
            .assetModelId(assetModelId)
            .assetDescription("Created using the AWS SDK for Java")

```

```

        .assetName(assetName)
        .build();

    return getAsyncClient().createAsset(createAssetRequest)
        .whenComplete((response, exception) -> {
            if (exception != null) {
                logger.error("Failed to create asset: {}",
exception.getCause().getMessage());
            }
        });
}

/**
 * Sends data to the SiteWise service.
 *
 * @param assetId      the ID of the asset to which the data will be sent.
 * @param tempPropertyId the ID of the temperature property.
 * @param humidityPropId the ID of the humidity property.
 * @return a {@link CompletableFuture} that represents a {@link
BatchPutAssetPropertyValueResponse} result. The
 *         calling code can attach callbacks, then handle the result or
exception by calling
 *         {@link CompletableFuture#join()} or {@link
CompletableFuture#get()}.
 *         <p>
 *         If any completion stage in this method throws an exception, the
method logs the exception cause and keeps it
 *         available to the calling code as a {@link CompletionException}. By
calling
 *         {@link CompletionException#getCause()}, the calling code can
access the original exception.
 */
public CompletableFuture<BatchPutAssetPropertyValueResponse>
sendDataToSiteWiseAsync(String assetId, String tempPropertyId, String
humidityPropId) {
    Map<String, Double> sampleData = generateSampleData();
    long timestamp = Instant.now().toEpochMilli();

    TimeInNanos time = TimeInNanos.builder()
        .timeInSeconds(timestamp / 1000)
        .offsetInNanos((int) ((timestamp % 1000) * 1000000))
        .build();

```

```
BatchPutAssetPropertyValueRequest request =
BatchPutAssetPropertyValueRequest.builder()
    .entries(Arrays.asList(
        PutAssetPropertyValueEntry.builder()
            .entryId("entry-3")
            .assetId(assetId)
            .propertyId(tempPropertyId)
            .propertyValues(Arrays.asList(
                AssetPropertyValue.builder()
                    .value(Variant.builder()
                        .doubleValue(sampleData.get("Temperature"))
                        .build())
                    .timestamp(time)
                    .build()
            ))
            .build(),
        PutAssetPropertyValueEntry.builder()
            .entryId("entry-4")
            .assetId(assetId)
            .propertyId(humidityPropId)
            .propertyValues(Arrays.asList(
                AssetPropertyValue.builder()
                    .value(Variant.builder()
                        .doubleValue(sampleData.get("Humidity"))
                        .build())
                    .timestamp(time)
                    .build()
            ))
            .build()
    ))
    .build();

return getAsyncClient().batchPutAssetPropertyValue(request)
    .whenComplete((response, exception) -> {
        if (exception != null) {
            logger.error("An exception occurred: {}",
exception.getCause().getMessage());
        }
    });
}

/**
 * Fetches the value of an asset property.
 *
 */
```

```

    * @param propId the ID of the asset property to fetch.
    * @param assetId the ID of the asset to fetch the property value for.
    * @return a {@link CompletableFuture} that represents a {@link Double}
result. The calling code can attach
    *     callbacks, then handle the result or exception by calling {@link
CompletableFuture#join()} or
    *     {@link CompletableFuture#get()}.
    *     <p>
    *     If any completion stage in this method throws an exception, the
method logs the exception cause and keeps
    *     it available to the calling code as a {@link CompletionException}.
By calling
    *     {@link CompletionException#getCause()}, the calling code can
access the original exception.
    */
    public CompletableFuture<Double> getAssetPropValueAsync(String propId, String
assetId) {
        GetAssetPropertyValueRequest assetPropertyValueRequest =
GetAssetPropertyValueRequest.builder()
            .propertyId(propId)
            .assetId(assetId)
            .build();

        return getAsyncClient().getAssetPropertyValue(assetPropertyValueRequest)
            .handle((response, exception) -> {
                if (exception != null) {
                    logger.error("Error occurred while fetching property
value: {}.", exception.getCause().getMessage());
                    throw (CompletionException) exception;
                }
                return response.propertyValue().value().doubleValue();
            });
    }

/**
    * Retrieves the property IDs associated with a specific asset model.
    *
    * @param assetModelId the ID of the asset model that defines the properties.
    * @return a {@link CompletableFuture} that represents a {@link Map} result
that associates the property name to the
    *     propert ID. The calling code can attach callbacks, then handle the
result or exception by calling
    *     {@link CompletableFuture#join()} or {@link
CompletableFuture#get()}.

```

```

    *      <p>
    *      If any completion stage in this method throws an exception, the
method logs the exception cause and keeps
    *      it available to the calling code as a {@link CompletionException}.
By calling
    *      {@link CompletionException#getCause()}, the calling code can
access the original exception.
    */
    public CompletableFuture<Map<String, String>> getPropertyIds(String
assetModelId) {
        ListAssetModelPropertiesRequest modelPropertiesRequest =
ListAssetModelPropertiesRequest.builder().assetModelId(assetModelId).build();
        return getAsyncClient().listAssetModelProperties(modelPropertiesRequest)
            .handle((response, throwable) -> {
                if (response != null) {
                    return response.assetModelPropertySummaries().stream()
                        .collect(Collectors
                            .toMap(AssetModelPropertySummary::name,
AssetModelPropertySummary::id));
                } else {
                    logger.error("Error occurred while fetching property IDs:
{}.\"", throwable.getCause().getMessage());
                    throw (CompletionException) throwable;
                }
            });
    }

/**
 * Deletes an asset.
 *
 * @param assetId the ID of the asset to be deleted.
 * @return a {@link CompletableFuture} that represents a {@link
DeleteAssetResponse} result. The calling code can
 *      attach callbacks, then handle the result or exception by calling
{@link CompletableFuture#join()} or
 *      {@link CompletableFuture#get()}.
 *      <p>
 *      If any completion stage in this method throws an exception, the
method logs the exception cause and keeps
 *      it available to the calling code as a {@link CompletionException}.
By calling
 *      {@link CompletionException#getCause()}, the calling code can
access the original exception.
 */

```

```

public CompletableFuture<DeleteAssetResponse> deleteAssetAsync(String
assetId) {
    DeleteAssetRequest deleteAssetRequest = DeleteAssetRequest.builder()
        .assetId(assetId)
        .build();

    return getAsyncClient().deleteAsset(deleteAssetRequest)
        .whenComplete((response, exception) -> {
            if (exception != null) {
                logger.error("An error occurred deleting asset with id: {}",
assetId);
            }
        });
}

/**
 * Deletes an Asset Model with the specified ID.
 *
 * @param assetModelId the ID of the Asset Model to delete.
 * @return a {@link CompletableFuture} that represents a {@link
DeleteAssetModelResponse} result. The calling code
 * can attach callbacks, then handle the result or exception by
calling {@link CompletableFuture#join()} or
 * {@link CompletableFuture#get()}.
 * <p>
 * If any completion stage in this method throws an exception, the
method logs the exception cause and keeps
 * it available to the calling code as a {@link CompletionException}.
By calling
 * {@link CompletionException#getCause()}, the calling code can
access the original exception.
 */
public CompletableFuture<DeleteAssetModelResponse>
deleteAssetModelAsync(String assetModelId) {
    DeleteAssetModelRequest deleteAssetModelRequest =
DeleteAssetModelRequest.builder()
        .assetModelId(assetModelId)
        .build();

    return getAsyncClient().deleteAssetModel(deleteAssetModelRequest)
        .whenComplete((response, exception) -> {
            if (exception != null) {
                logger.error("Failed to delete asset model with ID:{}",
exception.getMessage());
            }
        });
}

```

```

    }
    });
}

/**
 * Retrieves the asset model ID for the given asset model name.
 *
 * @param assetModelName the name of the asset model for the ID.
 * @return a {@link CompletableFuture} that represents a {@link String}
result of the asset model ID or null if the
 * asset model cannot be found. The calling code can attach
callbacks, then handle the result or exception
 * by calling {@link CompletableFuture#join()} or {@link
CompletableFuture#get()}.
 *
 * <p>
 * If any completion stage in this method throws an exception, the
method logs the exception cause and keeps
 * it available to the calling code as a {@link CompletionException}.
By calling
 * {@link CompletionException#getCause()}, the calling code can
access the original exception.
 */
public CompletableFuture<String> getAssetModelIdAsync(String assetModelName)
{
    ListAssetModelsRequest listAssetModelsRequest =
ListAssetModelsRequest.builder().build();
    return getAsyncClient().listAssetModels(listAssetModelsRequest)
        .handle((listAssetModelsResponse, exception) -> {
            if (exception != null) {
                logger.error("Failed to retrieve Asset Model ID: {}",
exception.getCause().getMessage());
                throw (CompletionException) exception;
            }
            for (AssetModelSummary assetModelSummary :
listAssetModelsResponse.assetModelSummaries()) {
                if (assetModelSummary.name().equals(assetModelName)) {
                    return assetModelSummary.id();
                }
            }
            return null;
        });
}

```

```
/**
 * Creates a new IoT SiteWise gateway.
 *
 * @param gatewayName The name of the gateway to create.
 * @param myThing      The name of the core device thing to associate with the
gateway.
 * @return a {@link CompletableFuture} that represents a {@link String}
result of the gateways ID. The calling code
 *         can attach callbacks, then handle the result or exception by
calling {@link CompletableFuture#join()} or
 *         {@link CompletableFuture#get()}.
 *         <p>
 *         If any completion stage in this method throws an exception, the
method logs the exception cause and keeps
 *         it available to the calling code as a {@link CompletionException}.
By calling
 *         {@link CompletionException#getCause()}, the calling code can
access the original exception.
 */
public CompletableFuture<String> createGatewayAsync(String gatewayName,
String myThing) {
    GreengrassV2 gg = GreengrassV2.builder()
        .coreDeviceThingName(myThing)
        .build();

    GatewayPlatform platform = GatewayPlatform.builder()
        .greengrassV2(gg)
        .build();

    Map<String, String> tag = new HashMap<>();
    tag.put("Environment", "Production");

    CreateGatewayRequest createGatewayRequest =
CreateGatewayRequest.builder()
        .gatewayName(gatewayName)
        .gatewayPlatform(platform)
        .tags(tag)
        .build();

    return getAsyncClient().createGateway(createGatewayRequest)
        .handle((response, exception) -> {
            if (exception != null) {
                logger.error("Error creating the gateway.");
                throw (CompletionException) exception;
            }
        });
}
```

```

        }
        logger.info("The ARN of the gateway is {}" ,
response.gatewayArn());
        return response.gatewayId();
    });
}

/**
 * Deletes the specified gateway.
 *
 * @param gatewayId the ID of the gateway to delete.
 * @return a {@link CompletableFuture} that represents a {@link
DeleteGatewayResponse} result.. The calling code
 * can attach callbacks, then handle the result or exception by
calling {@link CompletableFuture#join()} or
 * {@link CompletableFuture#get()}.
 *
 * <p>
 * If any completion stage in this method throws an exception, the
method logs the exception cause and keeps
 * it available to the calling code as a {@link CompletionException}.
By calling
 * {@link CompletionException#getCause()}, the calling code can
access the original exception.
 */
public CompletableFuture<DeleteGatewayResponse> deleteGatewayAsync(String
gatewayId) {
    DeleteGatewayRequest deleteGatewayRequest =
DeleteGatewayRequest.builder()
        .gatewayId(gatewayId)
        .build();

    return getAsyncClient().deleteGateway(deleteGatewayRequest)
        .whenComplete((response, exception) -> {
            if (exception != null) {
                logger.error("Failed to delete gateway: {}",
exception.getCause().getMessage());
            }
        });
}

/**
 * Describes the specified gateway.
 *
 * @param gatewayId the ID of the gateway to describe.

```

```
    * @return a {@link CompletableFuture} that represents a {@link
DescribeGatewayResponse} result. The calling code
    *         can attach callbacks, then handle the result or exception by
calling {@link CompletableFuture#join()} or
    *         {@link CompletableFuture#get()}.
    *         <p>
    *         If any completion stage in this method throws an exception, the
method logs the exception cause and keeps
    *         it available to the calling code as a {@link CompletionException}.
By calling
    *         {@link CompletionException#getCause()}, the calling code can
access the original exception.
    */
    public CompletableFuture<DescribeGatewayResponse> describeGatewayAsync(String
gatewayId) {
        DescribeGatewayRequest request = DescribeGatewayRequest.builder()
            .gatewayId(gatewayId)
            .build();

        return getAsyncClient().describeGateway(request)
            .whenComplete((response, exception) -> {
                if (exception != null) {
                    logger.error("An error occurred during the describeGateway
method: {}", exception.getCause().getMessage());
                }
            });
    }

    private static Map<String, Double> generateSampleData() {
        Map<String, Double> data = new HashMap<>();
        data.put("Temperature", 23.5);
        data.put("Humidity", 65.0);
        return data;
    }
}
```

JavaScript

SDK para JavaScript (v3)

Note

Hay más información. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
import {
  Scenario,
  ScenarioAction,
  ScenarioInput,
  ScenarioOutput,
  //} from "@aws-doc-sdk-examples/lib/scenario/index.js";
} from "../../libs/scenario/index.js";
import {
  IoTSiteWiseClient,
  CreateAssetModelCommand,
  CreateAssetCommand,
  ListAssetModelPropertiesCommand,
  BatchPutAssetPropertyValueCommand,
  GetAssetPropertyValueCommand,
  CreatePortalCommand,
  DescribePortalCommand,
  CreateGatewayCommand,
  DescribeGatewayCommand,
  DeletePortalCommand,
  DeleteGatewayCommand,
  DeleteAssetCommand,
  DeleteAssetModelCommand,
  DescribeAssetModelCommand,
} from "@aws-sdk/client-iotsitewise";
import {
  CloudFormationClient,
  CreateStackCommand,
  DeleteStackCommand,
  DescribeStacksCommand,
  waitUntilStackExists,
  waitUntilStackCreateComplete,
  waitUntilStackDeleteComplete,
```

```
} from "@aws-sdk/client-cloudformation";
import { wait } from "@aws-doc-sdk-examples/lib/utils/util-timers.js";
import { parseArgs } from "node:util";
import { readFileSync } from "node:fs";
import { fileURLToPath } from "node:url";
import { dirname } from "node:path";

const __filename = fileURLToPath(import.meta.url);
const __dirname = dirname(__filename);
const stackName = "SiteWiseBasicsStack";

/**
 * @typedef {{
 *   iotSiteWiseClient: import('@aws-sdk/client-iotsitewise').IotSiteWiseClient,
 *   cloudFormationClient: import('@aws-sdk/client-
cloudformation').CloudFormationClient,
 *   stackName,
 *   stack,
 *   askToDeleteResources: true,
 *   asset: {assetName: "MyAsset1"},
 *   assetModel: {assetModelName: "MyAssetModel1"},
 *   portal: {portalName: "MyPortal1"},
 *   gateway: {gatewayName: "MyGateway1"},
 *   propertyIds: [],
 *   contactEmail: "user@mydomain.com",
 *   thing: "MyThing1",
 *   sampleData: { temperature: 23.5, humidity: 65.0}
 * }} State
 */

/**
 * Used repeatedly to have the user press enter.
 * @type {ScenarioInput}
 */
const pressEnter = new ScenarioInput("continue", "Press Enter to continue", {
  type: "confirm",
});

const greet = new ScenarioOutput(
  "greet",
  `AWS IoT SiteWise is a fully managed industrial software-as-a-service (SaaS)
that makes it easy to collect, store, organize, and monitor data from industrial
equipment and processes. It is designed to help industrial and manufacturing`
);
```

organizations collect data from their equipment and processes, and use that data to make informed decisions about their operations.

One of the key features of AWS IoT SiteWise is its ability to connect to a wide range of industrial equipment and systems, including programmable logic controllers (PLCs), sensors, and other industrial devices. It can collect data from these devices and organize it into a unified data model, making it easier to analyze and gain insights from the data. AWS IoT SiteWise also provides tools for visualizing the data, setting up alarms and alerts, and generating reports. Another key feature of AWS IoT SiteWise is its ability to scale to handle large volumes of data. It can collect and store data from thousands of devices and process millions of data points per second, making it suitable for large-scale industrial operations. Additionally, AWS IoT SiteWise is designed to be secure and compliant, with features like role-based access controls, data encryption, and integration with other AWS services for additional security and compliance features.

```
Let's get started...`,
```

```
  { header: true },
);

const displayBuildCloudFormationStack = new ScenarioOutput(
  "displayBuildCloudFormationStack",
  "This scenario uses AWS CloudFormation to create an IAM role that is required
  for this scenario. The stack will now be deployed.",
);

const sdkBuildCloudFormationStack = new ScenarioAction(
  "sdkBuildCloudFormationStack",
  async (** @type {State} */ state) => {
    try {
      const data = readFileSync(
        `${__dirname}/../../../../resources/cfn/iotsitewise_basics/SitewiseRoles-
template.yml`,
        "utf8",
      );
      await state.cloudFormationClient.send(
        new CreateStackCommand({
          StackName: stackName,
          TemplateBody: data,
          Capabilities: ["CAPABILITY_IAM"],
        })),
      );
      await waitUntilStackExists(
        { client: state.cloudFormationClient },
```

```

    { StackName: stackName },
  );
  await waitUntilStackCreateComplete(
    { client: state.cloudFormationClient },
    { StackName: stackName },
  );
  const stack = await state.cloudFormationClient.send(
    new DescribeStacksCommand({
      StackName: stackName,
    }),
  );
  state.stack = stack.Stacks[0].Outputs[0];
  console.log(`The ARN of the IAM role is ${state.stack.OutputValue}`);
} catch (caught) {
  console.error(caught.message);
  throw caught;
}
},
);

```

```

const displayCreateAWSSiteWiseAssetModel = new ScenarioOutput(
  "displayCreateAWSSiteWiseAssetModel",
  `1. Create an AWS SiteWise Asset Model

```

An AWS IoT SiteWise Asset Model is a way to represent the physical assets, such as equipment, processes, and systems, that exist in an industrial environment. This model provides a structured and hierarchical representation of these assets, allowing users to define the relationships and properties of each asset.

This scenario creates two asset model properties: temperature and humidity.`,

```

);

```

```

const sdkCreateAWSSiteWiseAssetModel = new ScenarioAction(
  "sdkCreateAWSSiteWiseAssetModel",
  async (/** @type {State} */ state) => {
    let assetModelResponse;
    try {
      assetModelResponse = await state.iotSiteWiseClient.send(
        new CreateAssetModelCommand({
          assetModelName: state.assetModel.assetModelName,
          assetModelProperties: [
            {
              name: "Temperature",
              dataType: "DOUBLE",
              type: {

```

```

        measurement: {},
      },
    },
    {
      name: "Humidity",
      dataType: "DOUBLE",
      type: {
        measurement: {},
      },
    },
  ],
 )),
);
state.assetModel.assetModelId = assetModelResponse.assetModelId;
console.log(
  `Asset Model successfully created. Asset Model ID:
${state.assetModel.assetModelId}`,
);
} catch (caught) {
  if (caught.name === "ResourceAlreadyExistsException") {
    console.log(
      `The Asset Model ${state.assetModel.assetModelName} already exists.`,
    );
    throw caught;
  }
  console.error(`${caught.message}`);
  throw caught;
}
},
);

const displayCreateAWSIoTSiteWiseAssetModel = new ScenarioOutput(
  "displayCreateAWSIoTSiteWiseAssetModel",
  `2. Create an AWS IoT SiteWise Asset
The IoT SiteWise model that we just created defines the structure and metadata
for your physical assets. Now we create an asset from the asset model.

Let's wait 30 seconds for the asset to be ready.`,
);

const waitThirtySeconds = new ScenarioAction("waitThirtySeconds", async () => {
  await wait(30); // wait 30 seconds
  console.log("Time's up! Let's check the asset's status.");
});

```

```

const sdkCreateAWSIoTSiteWiseAssetModel = new ScenarioAction(
  "sdkCreateAWSIoTSiteWiseAssetModel",
  async (** @type {State} */ state) => {
    try {
      const assetResponse = await state.iotSiteWiseClient.send(
        new CreateAssetCommand({
          assetModelId: state.assetModel.assetModelId,
          assetName: state.asset.assetName,
        })),
      );
      state.asset.assetId = assetResponse.assetId;
      console.log(`Asset created with ID: ${state.asset.assetId}`);
    } catch (caught) {
      if (caught.name === "ResourceNotFoundException") {
        console.log(
          `The Asset ${state.assetModel.assetModelName} was not found.`);
      };
      throw caught;
    }
    console.error(`${caught.message}`);
    throw caught;
  }
},
);

```

```

const displayRetrievePropertyId = new ScenarioOutput(
  "displayRetrievePropertyId",
  `3. Retrieve the property ID values

```

To send data to an asset, we need to get the property ID values. In this scenario, we access the temperature and humidity property ID values.`

```
);
```

```

const sdkRetrievePropertyId = new ScenarioAction(
  "sdkRetrievePropertyId",
  async (state) => {
    try {
      const retrieveResponse = await state.iotSiteWiseClient.send(
        new ListAssetModelPropertiesCommand({
          assetModelId: state.assetModel.assetModelId,
        })),
      );
    }
  );

```

```
    for (const retrieveResponseKey in
retrieveResponse.assetModelPropertySummaries) {
      if (
        retrieveResponse.assetModelPropertySummaries[retrieveResponseKey]
          .name === "Humidity"
      ) {
        state.propertyIds.Humidity =
          retrieveResponse.assetModelPropertySummaries[
            retrieveResponseKey
          ].id;
      }
      if (
        retrieveResponse.assetModelPropertySummaries[retrieveResponseKey]
          .name === "Temperature"
      ) {
        state.propertyIds.Temperature =
          retrieveResponse.assetModelPropertySummaries[
            retrieveResponseKey
          ].id;
      }
    }
    console.log(`The Humidity propertyId is ${state.propertyIds.Humidity}`);
    console.log(
      `The Temperature propertyId is ${state.propertyIds.Temperature}`,
    );
  } catch (caught) {
    if (caught.name === "IoTSiteWiseException") {
      console.log(
        `There was a problem retrieving the properties: ${caught.message}`,
      );
      throw caught;
    }
    console.error(`${caught.message}`);
    throw caught;
  }
},
);

const displaySendDataToIoTSiteWiseAsset = new ScenarioOutput(
  "displaySendDataToIoTSiteWiseAsset",
  `4. Send data to an AWS IoT SiteWise Asset`
);
```

By sending data to an IoT SiteWise Asset, you can aggregate data from multiple sources, normalize the data into a standard format, and store it in a

centralized location. This makes it easier to analyze and gain insights from the data.

In this example, we generate sample temperature and humidity data and send it to the AWS IoT SiteWise asset.`
);

```
const sdkSendDataToIoTSiteWiseAsset = new ScenarioAction(
  "sdkSendDataToIoTSiteWiseAsset",
  async (state) => {
    try {
      const sendResponse = await state.iotSiteWiseClient.send(
        new BatchPutAssetPropertyValueCommand({
          entries: [
            {
              entryId: "entry-3",
              assetId: state.asset.assetId,
              propertyId: state.propertyIds.Humidity,
              propertyValues: [
                {
                  value: {
                    doubleValue: state.sampleData.humidity,
                  },
                  timestamp: {
                    timeInSeconds: Math.floor(Date.now() / 1000),
                  },
                },
              ],
            },
            {
              entryId: "entry-4",
              assetId: state.asset.assetId,
              propertyId: state.propertyIds.Temperature,
              propertyValues: [
                {
                  value: {
                    doubleValue: state.sampleData.temperature,
                  },
                  timestamp: {
                    timeInSeconds: Math.floor(Date.now() / 1000),
                  },
                },
              ],
            },
          ],
        })
      );
    }
  }
);
```

```

    ],
  )),
);
console.log("The data was sent successfully.");
} catch (caught) {
  if (caught.name === "ResourceNotFoundException") {
    console.log(`The Asset ${state.asset.assetName} was not found.`);
    throw caught;
  }
  console.error(`${caught.message}`);
  throw caught;
}
},
);

```

```

const displayRetrieveValueOfIoTSiteWiseAsset = new ScenarioOutput(
  "displayRetrieveValueOfIoTSiteWiseAsset",
  `5. Retrieve the value of the IoT SiteWise Asset property

```

```

IoT SiteWise is an AWS service that allows you to collect, process, and analyze
industrial data from connected equipment and sensors. One of the key benefits of
reading an IoT SiteWise property is the ability to gain valuable insights from
your industrial data.` ,
);

```

```

const sdkRetrieveValueOfIoTSiteWiseAsset = new ScenarioAction(
  "sdkRetrieveValueOfIoTSiteWiseAsset",
  async (** @type {State} */ state) => {
    try {
      const temperatureResponse = await state.iotSiteWiseClient.send(
        new GetAssetPropertyValueCommand({
          assetId: state.asset.assetId,
          propertyId: state.propertyIds.Temperature,
        }),
      );
      const humidityResponse = await state.iotSiteWiseClient.send(
        new GetAssetPropertyValueCommand({
          assetId: state.asset.assetId,
          propertyId: state.propertyIds.Humidity,
        }),
      );
      console.log(
        `The property value for Temperature is
        ${temperatureResponse.propertyValue.value.doubleValue}`,
      );
    }
  }
);

```

```

    );
    console.log(
      `The property value for Humidity is
    ${humidityResponse.propertyValue.value.doubleValue}`,
    );
  } catch (caught) {
    if (caught.name === "ResourceNotFoundException") {
      console.log(`The Asset ${state.asset.assetName} was not found.`);
      throw caught;
    }
    console.error(`${caught.message}`);
    throw caught;
  }
},
);

const displayCreateIoTSiteWisePortal = new ScenarioOutput(
  "displayCreateIoTSiteWisePortal",
  `6. Create an IoT SiteWise Portal

An IoT SiteWise Portal allows you to aggregate data from multiple industrial
sources, such as sensors, equipment, and control systems, into a centralized
platform.`
);

const sdkCreateIoTSiteWisePortal = new ScenarioAction(
  "sdkCreateIoTSiteWisePortal",
  async (** @type {State} */ state) => {
    try {
      const createPortalResponse = await state.iotSiteWiseClient.send(
        new CreatePortalCommand({
          portalName: state.portal.portalName,
          portalContactEmail: state.contactEmail,
          roleArn: state.stack.OutputValue,
        }),
      );
    };
    state.portal = { ...state.portal, ...createPortalResponse };
    await wait(5); // Allow the portal to properly propagate.
    console.log(
      `Portal created successfully. Portal ID
    ${createPortalResponse.portalId}`,
    );
  } catch (caught) {
    if (caught.name === "IoTSiteWiseException") {

```

```

        console.log(
            `There was a problem creating the Portal: ${caught.message}.`,
        );
        throw caught;
    }
    console.error(`${caught.message}`);
    throw caught;
}
},
);

```

```

const displayDescribePortal = new ScenarioOutput(
    "displayDescribePortal",
    `7. Describe the Portal

```

```

In this step, we get a description of the portal and display the portal URL.`
);

```

```

const sdkDescribePortal = new ScenarioAction(
    "sdkDescribePortal",
    async (** @type {State} */ state) => {
        try {
            const describePortalResponse = await state.iotSiteWiseClient.send(
                new DescribePortalCommand({
                    portalId: state.portal.portalId,
                }),
            );
            console.log(`Portal URL: ${describePortalResponse.portalStartUrl}`);
        } catch (caught) {
            if (caught.name === "ResourceNotFoundException") {
                console.log(`The Portal ${state.portal.portalName} was not found.`);
                throw caught;
            }
            console.error(`${caught.message}`);
            throw caught;
        }
    },
);

```

```

const displayCreateIoTSiteWiseGateway = new ScenarioOutput(
    "displayCreateIoTSiteWiseGateway",
    `8. Create an IoT SiteWise Gateway

```

```
IoT SiteWise Gateway serves as the bridge between industrial equipment, sensors,
and the cloud-based IoT SiteWise service. It is responsible for securely
collecting, processing, and transmitting data from various industrial assets
to the IoT SiteWise platform, enabling real-time monitoring, analysis, and
optimization of industrial operations.`
);

const sdkCreateIoTSiteWiseGateway = new ScenarioAction(
  "sdkCreateIoTSiteWiseGateway",
  async (** @type {State} */ state) => {
    try {
      const createGatewayResponse = await state.iotSiteWiseClient.send(
        new CreateGatewayCommand({
          gatewayName: state.gateway.gatewayName,
          gatewayPlatform: {
            greengrassV2: {
              coreDeviceThingName: state.thing,
            },
          },
        })),
      );
      console.log(
        `Gateway creation completed successfully. ID is
        ${createGatewayResponse.gatewayId}`,
      );
      state.gateway.gatewayId = createGatewayResponse.gatewayId;
    } catch (caught) {
      if (caught.name === "IoTSiteWiseException") {
        console.log(
          `There was a problem creating the gateway: ${caught.message}.`,
        );
        throw caught;
      }
      console.error(`${caught.message}`);
      throw caught;
    }
  },
);

const displayDescribeIoTSiteWiseGateway = new ScenarioOutput(
  "displayDescribeIoTSiteWiseGateway",
  "9. Describe the IoT SiteWise Gateway",
);
```

```

const sdkDescribeIoTSiteWiseGateway = new ScenarioAction(
  "sdkDescribeIoTSiteWiseGateway",
  async (/** @type {State} */ state) => {
    try {
      const describeGatewayResponse = await state.iotSiteWiseClient.send(
        new DescribeGatewayCommand({
          gatewayId: state.gateway.gatewayId,
        }),
      );
      console.log("Gateway creation completed successfully.");
      console.log(`Gateway Name: ${describeGatewayResponse.gatewayName}`);
      console.log(`Gateway ARN: ${describeGatewayResponse.gatewayArn}`);
      console.log(
        `Gateway Platform:
        ${Object.keys(describeGatewayResponse.gatewayPlatform)}`,
      );
      console.log(
        `Gateway Creation Date: ${describeGatewayResponse.creationDate}`,
      );
    } catch (caught) {
      if (caught.name === "ResourceNotFoundException") {
        console.log(`The Gateway ${state.gateway.gatewayId} was not found.`);
        throw caught;
      }
      console.error(`${caught.message}`);
      throw caught;
    }
  },
);

const askToDeleteResources = new ScenarioInput(
  "askToDeleteResources",
  `10. Delete the AWS IoT SiteWise Assets

Before you can delete the Asset Model, you must delete the assets.`,
  { type: "confirm" },
);

const displayConfirmDeleteResources = new ScenarioAction(
  "displayConfirmDeleteResources",
  async (/** @type {State} */ state) => {
    if (state.askToDeleteResources) {
      return "You selected to delete the SiteWise assets.";
    }
  }
);

```

```
    return "The resources will not be deleted. Please delete them manually to
    avoid charges.";
  },
);

const sdkDeleteResources = new ScenarioAction(
  "sdkDeleteResources",
  async (** @type {State} */ state) => {
    await wait(10); // Give the portal status time to catch up.
    try {
      await state.iotSiteWiseClient.send(
        new DeletePortalCommand({
          portalId: state.portal.portalId,
        }),
      );
      console.log(
        `Portal ${state.portal.portalName} was deleted successfully.`
      );
    } catch (caught) {
      if (caught.name === "ResourceNotFoundException") {
        console.log(`The Portal ${state.portal.portalName} was not found.`);
      } else {
        console.log(`When trying to delete the portal: ${caught.message}`);
      }
    }

    try {
      await state.iotSiteWiseClient.send(
        new DeleteGatewayCommand({
          gatewayId: state.gateway.gatewayId,
        }),
      );
      console.log(
        `Gateway ${state.gateway.gatewayName} was deleted successfully.`
      );
    } catch (caught) {
      if (caught.name === "ResourceNotFoundException") {
        console.log(`The Gateway ${state.gateway.gatewayId} was not found.`);
      } else {
        console.log(`When trying to delete the gateway: ${caught.message}`);
      }
    }

    try {
```

```
    await state.iotSiteWiseClient.send(
      new DeleteAssetCommand({
        assetId: state.asset.assetId,
      })),
    );
    await wait(5); // Allow the delete to finish.
    console.log(`Asset ${state.asset.assetName} was deleted successfully.`);
  } catch (caught) {
    if (caught.name === "ResourceNotFoundException") {
      console.log(`The Asset ${state.asset.assetName} was not found.`);
    } else {
      console.log(`When deleting the asset: ${caught.message}`);
    }
  }
}

await wait(30); // Allow asset deletion to finish.
try {
  await state.iotSiteWiseClient.send(
    new DeleteAssetModelCommand({
      assetModelId: state.assetModel.assetModelId,
    })),
  );
  console.log(
    `Asset Model ${state.assetModel.assetModelName} was deleted
successfully.` ,
  );
} catch (caught) {
  if (caught.name === "ResourceNotFoundException") {
    console.log(
      `The Asset Model ${state.assetModel.assetModelName} was not found.` ,
    );
  } else {
    console.log(`When deleting the asset model: ${caught.message}`);
  }
}

try {
  await state.cloudFormationClient.send(
    new DeleteStackCommand({
      StackName: stackName,
    })),
  );
  await waitUntilStackDeleteComplete(
    { client: state.cloudFormationClient },
```

```
    { StackName: stackName },
  );
  console.log("The stack was deleted successfully.");
} catch (caught) {
  console.log(
    `${caught.message}. The stack was NOT deleted. Please clean up the
resources manually.`
  );
}
},
{ skipWhen: (/** @type {} */ state) => !state.askToDeleteResources },
);

const goodbye = new ScenarioOutput(
  "goodbye",
  "This concludes the IoT SiteWise Basics scenario for the AWS Javascript SDK v3.
Thank you!",
);

const myScenario = new Scenario(
  "IoTSiteWise Basics",
  [
    greet,
    pressEnter,
    displayBuildCloudFormationStack,
    sdkBuildCloudFormationStack,
    pressEnter,
    displayCreateAWSSiteWiseAssetModel,
    sdkCreateAWSSiteWiseAssetModel,
    displayCreateAWSIoTSiteWiseAssetModel,
    pressEnter,
    waitThirtySeconds,
    sdkCreateAWSIoTSiteWiseAssetModel,
    pressEnter,
    displayRetrievePropertyId,
    sdkRetrievePropertyId,
    pressEnter,
    displaySendDataToIoTSiteWiseAsset,
    sdkSendDataToIoTSiteWiseAsset,
    pressEnter,
    displayRetrieveValueOfIoTSiteWiseAsset,
    sdkRetrieveValueOfIoTSiteWiseAsset,
    pressEnter,
    displayCreateIoTSiteWisePortal,
```

```

    sdkCreateIoTSiteWisePortal,
    pressEnter,
    displayDescribePortal,
    sdkDescribePortal,
    pressEnter,
    displayCreateIoTSiteWiseGateway,
    sdkCreateIoTSiteWiseGateway,
    pressEnter,
    displayDescribeIoTSiteWiseGateway,
    sdkDescribeIoTSiteWiseGateway,
    pressEnter,
    askToDeleteResources,
    displayConfirmDeleteResources,
    sdkDeleteResources,
    goodbye,
  ],
  {
    iotSiteWiseClient: new IoTSiteWiseClient({}),
    cloudFormationClient: new CloudFormationClient({}),
    asset: { assetName: "MyAsset1" },
    assetModel: { assetModelName: "MyAssetModel1" },
    portal: { portalName: "MyPortal1" },
    gateway: { gatewayName: "MyGateway1" },
    propertyIds: [],
    contactEmail: "user@mydomain.com",
    thing: "MyThing1",
    sampleData: { temperature: 23.5, humidity: 65.0 },
  },
);

/** @type {{ stepHandlerOptions: StepHandlerOptions }} */
export const main = async (stepHandlerOptions) => {
  await myScenario.run(stepHandlerOptions);
};

// Invoke main function if this file was run directly.
if (process.argv[1] === fileURLToPath(import.meta.url)) {
  const { values } = parseArgs({
    options: {
      yes: {
        type: "boolean",
        short: "y",
      },
    },
  },
);

```

```
});  
main({ confirmAll: values.yes });  
}
```

Python

SDK para Python (Boto3)

Note

Hay más información GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

Ejecutar un escenario interactivo en un símbolo del sistema.

```
class IoTSitewiseGettingStarted:  
    """  
    A scenario that demonstrates how to use Boto3 to manage IoT physical assets  
    using  
    the AWS IoT SiteWise.  
    """  
  
    def __init__(  
        self,  
        iot_sitewise_wrapper: IoTSitewiseWrapper,  
        cloud_formation_resource: ServiceResource,  
    ):  
        self.iot_sitewise_wrapper = iot_sitewise_wrapper  
        self.cloud_formation_resource = cloud_formation_resource  
        self.stack = None  
        self.asset_model_id = None  
        self.asset_id = None  
        self.portal_id = None  
        self.gateway_id = None  
  
    def run(self) -> None:  
        """  
        Runs the scenario.  
        """
```

```
print(
    """
AWS IoT SiteWise is a fully managed software-as-a-service (SaaS) that
makes it easy to collect, store, organize, and monitor data from industrial
equipment and processes.
It is designed to help industrial and manufacturing organizations collect data
from their equipment and
processes, and use that data to make informed decisions about their operations.

One of the key features of AWS IoT SiteWise is its ability to connect to a wide
range of industrial
equipment and systems, including programmable logic controllers (PLCs), sensors,
and other
industrial devices. It can collect data from these devices and organize it into a
unified data model,
making it easier to analyze and gain insights from the data. AWS IoT SiteWise
also provides tools for
visualizing the data, setting up alarms and alerts, and generating reports.

Another key feature of AWS IoT SiteWise is its ability to scale to handle large
volumes of data.
It can collect and store data from thousands of devices and process millions of
data points per second,
making it suitable for large-scale industrial operations. Additionally, AWS IoT
SiteWise is designed
to be secure and compliant, with features like role-based access controls, data
encryption,
and integration with other AWS services for additional security and compliance
features.

Let's get started...
    """
)
press_enter_to_continue()
print_dashes()
print(f"")
print(
    f"Use AWS CloudFormation to create an IAM role that is required for
this scenario."
)
template_file = IoTSitewiseGettingStarted.get_template_as_string()

self.stack = self.deploy_cloudformation_stack(
    "python-iot-sitewise-basics", template_file
```

```

    )
    outputs = self.stack.outputs
    iam_role = None

    for output in outputs:
        if output.get("OutputKey") == "SitewiseRoleArn":
            iam_role = output.get("OutputValue")

    if iam_role is None:
        error_string = f"Failed to retrieve iam_role from CloudFormation
stack."
        logger.error(error_string)
        raise ValueError(error_string)

    print(f"The ARN of the IAM role is {iam_role}")
    print_dashes()
    print_dashes()
    print(f"1. Create an AWS SiteWise Asset Model")
    print(
        """

```

An AWS IoT SiteWise Asset Model is a way to represent the physical assets, such as equipment, processes, and systems, that exist in an industrial environment. This model provides a structured and hierarchical representation of these assets, allowing users to define the relationships and values of each asset.

This scenario creates two asset model values: temperature and humidity.

```

    """
    )
    press_enter_to_continue()
    asset_model_name = "MyAssetModel1"
    temperature_property_name = "temperature"
    humidity_property_name = "humidity"
    try:
        properties = [
            {
                "name": temperature_property_name,
                "dataType": "DOUBLE",
                "type": {
                    "measurement": {},
                },
            },
        ],
    },

```

```

        {
            "name": humidity_property_name,
            "dataType": "DOUBLE",
            "type": {
                "measurement": {},
            },
        },
    ]
    self.asset_model_id = self.iot_sitewise_wrapper.create_asset_model(
        asset_model_name, properties
    )
    print(
        f"Asset Model successfully created. Asset Model ID:
{self.asset_model_id}. "
    )
    except ClientError as err:
        if err.response["Error"]["Code"] == "ResourceAlreadyExistsException":
            self.asset_model_id =
self.get_model_id_for_model_name(asset_model_name)
            print(
                f"Asset Model {asset_model_name} already exists. Asset Model
ID: {self.asset_model_id}. "
            )
        else:
            raise

    press_enter_to_continue()
    print_dashes()
    print(f"2. Create an AWS IoT SiteWise Asset")
    print(
        """

```

The IoT SiteWise model that we just created defines the structure and metadata for your physical assets.

Now we create an asset from the asset model.

```

        """
    )
    press_enter_to_continue()

    self.asset_id = self.iot_sitewise_wrapper.create_asset(
        "MyAsset1", self.asset_model_id
    )

    print(f"Asset created with ID: {self.asset_id}")

```

```

press_enter_to_continue()
print_dashes()
print_dashes()
print(f"3. Retrieve the property ID values")
print(
    """

```

To send data to an asset, we need to get the property ID values. In this scenario, we access the temperature and humidity property ID values.

```

    """
)
press_enter_to_continue()
property_ids = self.iot_sitewise_wrapper.list_asset_model_properties(
    self.asset_model_id
)
humidity_property_id = None
temperature_property_id = None
for property_id in property_ids:
    if property_id.get("name") == humidity_property_name:
        humidity_property_id = property_id.get("id")
    elif property_id.get("name") == temperature_property_name:
        temperature_property_id = property_id.get("id")
if humidity_property_id is None or temperature_property_id is None:
    error_string = f"Failed to retrieve property IDs from Asset Model."
    logger.error(error_string)
    raise ValueError(error_string)

print(f"The Humidity property Id is {humidity_property_id}")
print(f"The Temperature property Id is {temperature_property_id}")
press_enter_to_continue()
print_dashes()
print_dashes()

```

```

print(f"4. Send data to an AWS IoT SiteWise Asset")
print(
    """

```

By sending data to an IoT SiteWise Asset, you can aggregate data from multiple sources, normalize the data into a standard format, and store it in a centralized location. This makes it easier to analyze and gain insights from the data.

In this example, we generate sample temperature and humidity data and send it to the AWS IoT SiteWise asset.

```

"""
)
press_enter_to_continue()

values = [
    {
        "propertyId": humidity_property_id,
        "valueType": "doubleValue",
        "value": 65.0,
    },
    {
        "propertyId": temperature_property_id,
        "valueType": "doubleValue",
        "value": 23.5,
    },
]
self.iot_sitewise_wrapper.batch_put_asset_property_value(self.asset_id,
values)
print(f"Data sent successfully.")

press_enter_to_continue()
print_dashes()
print_dashes()

print(f"5. Retrieve the value of the IoT SiteWise Asset property")
print(
    """

```

IoT SiteWise is an AWS service that allows you to collect, process, and analyze industrial data from connected equipment and sensors. One of the key benefits of reading an IoT SiteWise property is the ability to gain valuable insights from your industrial data.

```

"""
)
press_enter_to_continue()

property_value = self.iot_sitewise_wrapper.get_asset_property_value(
    self.asset_id, temperature_property_id
)
print(f"The property name is '{temperature_property_name}'.")

print(

```

```

        f"The value of this property is: {property_value['value']}
['doubleValue']]"
    )
    press_enter_to_continue()

    property_value = self.iot_sitewise_wrapper.get_asset_property_value(
        self.asset_id, humidity_property_id
    )
    print(f"The property name is '{humidity_property_name}'.")
    print(
        f"The value of this property is: {property_value['value']}
['doubleValue']]"
    )
    press_enter_to_continue()
    print_dashes()
    print_dashes()

    print(f"6. Create an IoT SiteWise Portal")
    print(
        """
An IoT SiteWise Portal allows you to aggregate data from multiple industrial
sources,
such as sensors, equipment, and control systems, into a centralized platform.
        """
    )

    press_enter_to_continue()
    contact_email = q.ask("Enter a contact email for the portal:",
q.non_empty)
    print("Creating the portal. The portal may take a while to become
active.")
    self.portal_id = self.iot_sitewise_wrapper.create_portal(
        "MyPortal1", iam_role, contact_email
    )
    print(f"Portal created successfully. Portal ID {self.portal_id}")
    press_enter_to_continue()
    print_dashes()
    print_dashes()

    print(f"7. Describe the Portal")
    print(
        """
In this step, we get a description of the portal and display the portal URL.
        """

```

```
)
    press_enter_to_continue()
    portal_description =
self.iot_sitewise_wrapper.describe_portal(self.portal_id)
    print(f"Portal URL: {portal_description['portalStartUrl']}")
    press_enter_to_continue()
    print_dashes()
    print_dashes()

    print(f"8. Create an IoT SiteWise Gateway")
    press_enter_to_continue()
    self.gateway_id = self.iot_sitewise_wrapper.create_gateway(
        "MyGateway1", "MyThing1"
    )
    print(f"Gateway creation completed successfully. id is
{self.gateway_id}")
    print_dashes()
    print_dashes()
    print(f"9. Describe the IoT SiteWise Gateway")
    press_enter_to_continue()

    gateway_description = self.iot_sitewise_wrapper.describe_gateway(
        self.gateway_id
    )
    print(f"Gateway Name: {gateway_description['gatewayName']}")
    print(f"Gateway ARN: {gateway_description['gatewayArn']}")
    print(f"Gateway Platform:\n{gateway_description['gatewayPlatform']}")
    print(f"Gateway Creation Date: {gateway_description['gatewayArn']}")
    print_dashes()
    print_dashes()

    print(f"10. Delete the AWS IoT SiteWise Assets")
    if q.ask("Would you like to delete the IoT SiteWise Assets? (y/n)",
q.is_yesno):
        self.cleanup()
    else:
        print(f"The resources will not be deleted.")
    print_dashes()
    print_dashes()
    print(f"This concludes the AWS IoT SiteWise Scenario")

def cleanup(self) -> None:
    """
    Deletes the CloudFormation stack and the resources created for the demo.
```

```

"""

if self.gateway_id is not None:
    self.iot_sitewise_wrapper.delete_gateway(self.gateway_id)
    print(f"Deleted gateway with id {self.gateway_id}.")
    self.gateway_id = None
if self.portal_id is not None:
    self.iot_sitewise_wrapper.delete_portal(self.portal_id)
    print(f"Deleted portal with id {self.portal_id}.")
    self.portal_id = None
if self.asset_id is not None:
    self.iot_sitewise_wrapper.delete_asset(self.asset_id)
    print(f"Deleted asset with id {self.asset_id}.")
    self.iot_sitewise_wrapper.wait_asset_deleted(self.asset_id)
    self.asset_id = None
if self.asset_model_id is not None:
    self.iot_sitewise_wrapper.delete_asset_model(self.asset_model_id)
    print(f"Deleted asset model with id {self.asset_model_id}.")
    self.asset_model_id = None
if self.stack is not None:
    stack = self.stack
    self.stack = None
    self.destroy_cloudformation_stack(stack)

def deploy_cloudformation_stack(
    self, stack_name: str, cfn_template: str
) -> ServiceResource:
    """
    Deploys prerequisite resources used by the scenario. The resources are
    defined in the associated `SitewiseRoles-template.yaml` AWS
    CloudFormation script and are deployed
    as a CloudFormation stack, so they can be easily managed and destroyed.

    :param stack_name: The name of the CloudFormation stack.
    :param cfn_template: The CloudFormation template as a string.
    :return: The CloudFormation stack resource.
    """
    print(f"Deploying CloudFormation stack: {stack_name}.")
    stack = self.cloud_formation_resource.create_stack(
        StackName=stack_name,
        TemplateBody=cfn_template,
        Capabilities=["CAPABILITY_NAMED_IAM"],
    )
    print(f"CloudFormation stack creation started: {stack_name}")

```

```
print("Waiting for CloudFormation stack creation to complete...")
waiter = self.cloud_formation_resource.meta.client.get_waiter(
    "stack_create_complete"
)
waiter.wait(StackName=stack.name)
stack.load()
print("CloudFormation stack creation complete.")

return stack

def destroy_cloudformation_stack(self, stack: ServiceResource) -> None:
    """
    Destroys the resources managed by the CloudFormation stack, and the
    CloudFormation
    stack itself.

    :param stack: The CloudFormation stack that manages the example
    resources.
    """
    print(
        f"CloudFormation stack '{stack.name}' is being deleted. This may take
    a few minutes."
    )
    stack.delete()
    waiter = self.cloud_formation_resource.meta.client.get_waiter(
        "stack_delete_complete"
    )
    waiter.wait(StackName=stack.name)
    print(f"CloudFormation stack '{stack.name}' has been deleted.")

@staticmethod
def get_template_as_string() -> str:
    """
    Returns a string containing this scenario's CloudFormation template.
    """
    template_file_path = os.path.join(script_dir, "SitewiseRoles-
template.yaml")
    file = open(template_file_path, "r")
    return file.read()

def get_model_id_for_model_name(self, model_name: str) -> str:
    """
    Returns the model ID for the given model name.
```

```

:param model_name: The name of the model.
:return: The model ID.
"""
model_id = None
asset_models = self.iot_sitewise_wrapper.list_asset_models()
for asset_model in asset_models:
    if asset_model["name"] == model_name:
        model_id = asset_model["id"]
        break
return model_id

```

Clase IoTSitewise Wrapper que envuelve acciones. AWS IoT SiteWise

```

class IoTSitewiseWrapper:
    """Encapsulates AWS IoT SiteWise actions using the client interface."""

    def __init__(self, iotsitewise_client: client) -> None:
        """
        Initializes the IoTSitewiseWrapper with an AWS IoT SiteWise client.

        :param iotsitewise_client: A Boto3 AWS IoT SiteWise client. This client
        provides low-level
            access to AWS IoT SiteWise services.
        """
        self.iotsitewise_client = iotsitewise_client
        self.entry_id = 0 # Incremented to generate unique entry IDs for
        batch_put_asset_property_value.

    @classmethod
    def from_client(cls) -> "IoTSitewiseWrapper":
        """
        Creates an IoTSitewiseWrapper instance with a default AWS IoT SiteWise
        client.

        :return: An instance of IoTSitewiseWrapper initialized with the default
        AWS IoT SiteWise client.
        """
        iotsitewise_client = boto3.client("iotsitewise")
        return cls(iotsitewise_client)

```

```

def create_asset_model(
    self, asset_model_name: str, properties: List[Dict[str, Any]]
) -> str:
    """
    Creates an AWS IoT SiteWise Asset Model.

    :param asset_model_name: The name of the asset model to create.
    :param properties: The property definitions of the asset model.
    :return: The ID of the created asset model.
    """
    try:
        response = self.iotsitewise_client.create_asset_model(
            assetModelName=asset_model_name,
            assetModelDescription="This is a sample asset model
description.",
            assetModelProperties=properties,
        )
        asset_model_id = response["assetModelId"]
        waiter = self.iotsitewise_client.get_waiter("asset_model_active")
        waiter.wait(assetModelId=asset_model_id)
        return asset_model_id
    except ClientError as err:
        if err.response["Error"]["Code"] == "ResourceAlreadyExistsException":
            logger.error("Asset model %s already exists.", asset_model_name)
        else:
            logger.error(
                "Error creating asset model %s. Here's why %s",
                asset_model_name,
                err.response["Error"]["Message"],
            )
        raise

def create_asset(self, asset_name: str, asset_model_id: str) -> str:
    """
    Creates an AWS IoT SiteWise Asset.

    :param asset_name: The name of the asset to create.
    :param asset_model_id: The ID of the asset model to associate with the
asset.
    :return: The ID of the created asset.
    """
    try:

```

```
        response = self.iotsitewise_client.create_asset(
            assetName=asset_name, assetModelId=asset_model_id
        )
        asset_id = response["assetId"]
        waiter = self.iotsitewise_client.get_waiter("asset_active")
        waiter.wait(assetId=asset_id)
        return asset_id
    except ClientError as err:
        if err.response["Error"] == "ResourceNotFoundException":
            logger.error("Asset model %s does not exist.", asset_model_id)
        else:
            logger.error(
                "Error creating asset %s. Here's why %s",
                asset_name,
                err.response["Error"]["Message"],
            )
        raise

def list_asset_models(self) -> List[Dict[str, Any]]:
    """
    Lists all AWS IoT SiteWise Asset Models.

    :return: A list of dictionaries containing information about each asset
    model.

    """
    try:
        asset_models = []
        paginator =
self.iotsitewise_client.get_paginator("list_asset_models")
        pages = paginator.paginate()
        for page in pages:
            asset_models.extend(page["assetModelSummaries"])
        return asset_models
    except ClientError as err:
        logger.error(
            "Error listing asset models. Here's why %s",
            err.response["Error"]["Message"],
        )
        raise
```

```

def list_asset_model_properties(self, asset_model_id: str) -> List[Dict[str,
Any]]:
    """
    Lists all AWS IoT SiteWise Asset Model Properties.

    :param asset_model_id: The ID of the asset model to list values for.
    :return: A list of dictionaries containing information about each asset
model property.
    """
    try:
        asset_model_properties = []
        paginator = self.iotsitewise_client.get_paginator(
            "list_asset_model_properties"
        )
        pages = paginator.paginate(assetModelId=asset_model_id)
        for page in pages:
            asset_model_properties.extend(page["assetModelPropertySummaries"])
        return asset_model_properties
    except ClientError as err:
        logger.error(
            "Error listing asset model values. Here's why %s",
            err.response["Error"]["Message"],
        )
        raise

def batch_put_asset_property_value(
    self, asset_id: str, values: List[Dict[str, str]]
) -> None:
    """
    Sends data to an AWS IoT SiteWise Asset.

    :param asset_id: The asset ID.
    :param values: A list of dictionaries containing the values in the form
        {propertyId : property_id,
         valueType : [stringValue|integerValue|doubleValue|
booleanValue],
         value : the_value}.
    """
    try:
        entries = self.properties_to_values(asset_id, values)

self.iotsitewise_client.batch_put_asset_property_value(entries=entries)

```

```

except ClientError as err:
    if err.response["Error"]["Code"] == "ResourceNotFoundException":
        logger.error("Asset %s does not exist.", asset_id)
    else:
        logger.error(
            "Error sending data to asset. Here's why %s",
            err.response["Error"]["Message"],
        )
    raise

def properties_to_values(
    self, asset_id: str, values: list[dict[str, Any]]
) -> list[dict[str, Any]]:
    """
    Utility function to convert a values list to the entries parameter for
    batch_put_asset_property_value.
    :param asset_id : The asset ID.
    :param values : A list of dictionaries containing the values in the form
        {propertyId : property_id,
         valueType : [stringValue|integerValue|doubleValue|
booleanValue],
         value : the_value}.
    :return: An entries list to pass as the 'entries' parameter to
    batch_put_asset_property_value.
    """
    entries = []
    for value in values:
        epoch_ns = time.time_ns()
        self.entry_id += 1
        if value["valueType"] == "stringValue":
            property_value = {"stringValue": value["value"]}
        elif value["valueType"] == "integerValue":
            property_value = {"integerValue": value["value"]}
        elif value["valueType"] == "booleanValue":
            property_value = {"booleanValue": value["value"]}
        elif value["valueType"] == "doubleValue":
            property_value = {"doubleValue": value["value"]}
        else:
            raise ValueError("Invalid valueType: %s", value["valueType"])
        entry = {
            "entryId": f"{self.entry_id}",
            "assetId": asset_id,
            "propertyId": value["propertyId"],

```

```

        "propertyValues": [
            {
                "value": property_value,
                "timestamp": {
                    "timeInSeconds": int(epoch_ns / 1000000000),
                    "offsetInNanos": epoch_ns % 1000000000,
                },
            }
        ],
    }
    entries.append(entry)
return entries

def get_asset_property_value(
    self, asset_id: str, property_id: str
) -> Dict[str, Any]:
    """
    Gets the value of an AWS IoT SiteWise Asset Property.

    :param asset_id: The ID of the asset.
    :param property_id: The ID of the property.
    :return: A dictionary containing the value of the property.
    """
    try:
        response = self.iotsitewise_client.get_asset_property_value(
            assetId=asset_id, propertyId=property_id
        )
        return response["propertyValue"]
    except ClientError as err:
        if err.response["Error"]["Code"] == "ResourceNotFoundException":
            logger.error(
                "Asset %s or property %s does not exist.", asset_id,
property_id
            )
        else:
            logger.error(
                "Error getting asset property value. Here's why %s",
                err.response["Error"]["Message"],
            )
        raise

def create_portal(

```

```
    self, portal_name: str, iam_role_arn: str, portal_contact_email: str
) -> str:
    """
    Creates an AWS IoT SiteWise Portal.

    :param portal_name: The name of the portal to create.
    :param iam_role_arn: The ARN of an IAM role.
    :param portal_contact_email: The contact email of the portal.
    :return: The ID of the created portal.
    """
    try:
        response = self.iotsitewise_client.create_portal(
            portalName=portal_name,
            roleArn=iam_role_arn,
            portalContactEmail=portal_contact_email,
        )
        portal_id = response["portalId"]
        waiter = self.iotsitewise_client.get_waiter("portal_active")
        waiter.wait(portalId=portal_id, WaiterConfig={"MaxAttempts": 40})
        return portal_id
    except ClientError as err:
        if err.response["Error"]["Code"] == "ResourceAlreadyExistsException":
            logger.error("Portal %s already exists.", portal_name)
        else:
            logger.error(
                "Error creating portal %s. Here's why %s",
                portal_name,
                err.response["Error"]["Message"],
            )
        raise

def describe_portal(self, portal_id: str) -> Dict[str, Any]:
    """
    Describes an AWS IoT SiteWise Portal.

    :param portal_id: The ID of the portal to describe.
    :return: A dictionary containing information about the portal.
    """
    try:
        response =
self.iotsitewise_client.describe_portal(portalId=portal_id)
        return response
    except ClientError as err:
```

```
        logger.error(
            "Error describing portal %s. Here's why %s",
            portal_id,
            err.response["Error"]["Message"],
        )
        raise

def create_gateway(self, gateway_name: str, my_thing: str) -> str:
    """
    Creates an AWS IoT SiteWise Gateway.

    :param gateway_name: The name of the gateway to create.
    :param my_thing: The core device thing name.
    :return: The ID of the created gateway.
    """
    try:
        response = self.iotsitewise_client.create_gateway(
            gatewayName=gateway_name,
            gatewayPlatform={
                "greengrassV2": {"coreDeviceThingName": my_thing},
            },
            tags={"Environment": "Production"},
        )
        gateway_id = response["gatewayId"]
        return gateway_id
    except ClientError as err:
        if err.response["Error"]["Code"] == "ResourceAlreadyExistsException":
            logger.error("Gateway %s already exists.", gateway_name)
        else:
            logger.error(
                "Error creating gateway %s. Here's why %s",
                gateway_name,
                err.response["Error"]["Message"],
            )
            raise

def describe_gateway(self, gateway_id: str) -> Dict[str, Any]:
    """
    Describes an AWS IoT SiteWise Gateway.

    :param gateway_id: The ID of the gateway to describe.
    :return: A dictionary containing information about the gateway.
```

```

    """
    try:
        response =
self.iotsitewise_client.describe_gateway(gatewayId=gateway_id)
        return response
    except ClientError as err:
        if err.response["Error"]["Code"] == "ResourceNotFoundException":
            logger.error("Gateway %s does not exist.", gateway_id)
        else:
            logger.error(
                "Error describing gateway %s. Here's why %s",
                gateway_id,
                err.response["Error"]["Message"],
            )
        raise

def delete_gateway(self, gateway_id: str) -> None:
    """
    Deletes an AWS IoT SiteWise Gateway.

    :param gateway_id: The ID of the gateway to delete.
    """
    try:
        self.iotsitewise_client.delete_gateway(gatewayId=gateway_id)
    except ClientError as err:
        if err.response["Error"]["Code"] == "ResourceNotFoundException":
            logger.error("Gateway %s does not exist.", gateway_id)
        else:
            logger.error(
                "Error deleting gateway %s. Here's why %s",
                gateway_id,
                err.response["Error"]["Message"],
            )
        raise

def delete_portal(self, portal_id: str) -> None:
    """
    Deletes an AWS IoT SiteWise Portal.

    :param portal_id: The ID of the portal to delete.
    """
    try:
```

```
        self.iotsitewise_client.delete_portal(portalId=portal_id)
    except ClientError as err:
        if err.response["Error"]["Code"] == "ResourceNotFoundException":
            logger.error("Portal %s does not exist.", portal_id)
        else:
            logger.error(
                "Error deleting portal %s. Here's why %s",
                portal_id,
                err.response["Error"]["Message"],
            )
        raise

def delete_asset(self, asset_id: str) -> None:
    """
    Deletes an AWS IoT SiteWise Asset.

    :param asset_id: The ID of the asset to delete.
    """
    try:
        self.iotsitewise_client.delete_asset(assetId=asset_id)
    except ClientError as err:
        logger.error(
            "Error deleting asset %s. Here's why %s",
            asset_id,
            err.response["Error"]["Message"],
        )
        raise

def delete_asset_model(self, asset_model_id: str) -> None:
    """
    Deletes an AWS IoT SiteWise Asset Model.

    :param asset_model_id: The ID of the asset model to delete.
    """
    try:
        self.iotsitewise_client.delete_asset_model(assetModelId=asset_model_id)
    except ClientError as err:
        logger.error(
            "Error deleting asset model %s. Here's why %s",
            asset_model_id,
            err.response["Error"]["Message"],
        )
```

```
    )
    raise

def wait_asset_deleted(self, asset_id: str) -> None:
    """
    Waits for an AWS IoT SiteWise Asset to be deleted.

    :param asset_id: The ID of the asset to wait for.
    """
    try:
        waiter = self.iotsitewise_client.get_waiter("asset_not_exists")
        waiter.wait(assetId=asset_id)
    except ClientError as err:
        logger.error(
            "Error waiting for asset %s to be deleted. Here's why %s",
            asset_id,
            err.response["Error"]["Message"],
        )
    raise
```

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulte [Uso de este servicio con un AWS SDK](#). En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Acciones de AWS IoT SiteWise uso AWS SDKs

Los siguientes ejemplos de código muestran cómo realizar AWS IoT SiteWise acciones individuales con AWS SDKs. Cada ejemplo incluye un enlace a GitHub, donde puede encontrar instrucciones para configurar y ejecutar el código.

Los siguientes ejemplos incluyen solo las acciones que se utilizan con mayor frecuencia. Para ver una lista completa, consulte la [Referencia de la API de AWS IoT SiteWise](#).

Ejemplos

- [Úselo BatchPutAssetPropertyValue con un AWS SDK o CLI](#)
- [Úselo CreateAsset con un AWS SDK o CLI](#)

- [Úselo CreateAssetModel con un AWS SDK o CLI](#)
- [Úselo CreateGateway con un AWS SDK o CLI](#)
- [Úselo DeleteAsset con un AWS SDK o CLI](#)
- [Úselo DeleteAssetModel con un AWS SDK o CLI](#)
- [Úselo DeleteGateway con un AWS SDK o CLI](#)
- [Úselo DescribeAssetModel con un AWS SDK o CLI](#)
- [Úselo DescribeGateway con un AWS SDK o CLI](#)
- [Úselo GetAssetPropertyValue con un AWS SDK o CLI](#)
- [ListAssetModelPropertiesÚselo con un AWS SDK](#)
- [Úselo ListAssetModels con un AWS SDK o CLI](#)

Úselo **BatchPutAssetPropertyValue** con un AWS SDK o CLI

Los siguientes ejemplos de código muestran cómo utilizar BatchPutAssetPropertyValue.

CLI

AWS CLI

Para enviar datos a propiedades de los activos

En el siguiente ejemplo de batch-put-asset-property-value, se envían datos de potencia y temperatura a las propiedades de los activos identificadas por los alias de las propiedades.

```
aws iotsitewise batch-put-asset-property-value \  
  --cli-input-json file://batch-put-asset-property-value.json
```

Contenido de batch-put-asset-property-value.json:

```
{  
  "entries": [  
    {  
      "entryId": "1575691200-company-windfarm-3-turbine-7-power",  
      "propertyAlias": "company-windfarm-3-turbine-7-power",  
      "propertyValues": [  

```

```
        {
          "value": {
            "doubleValue": 4.92
          },
          "timestamp": {
            "timeInSeconds": 1575691200
          },
          "quality": "GOOD"
        }
      ]
    },
    {
      "entryId": "1575691200-company-windfarm-3-turbine-7-temperature",
      "propertyAlias": "company-windfarm-3-turbine-7-temperature",
      "propertyValues": [
        {
          "value": {
            "integerValue": 38
          },
          "timestamp": {
            "timeInSeconds": 1575691200
          }
        }
      ]
    }
  ]
}
```

Salida:


```
{
  "errorEntries": []
}
```

Para obtener más información, consulte [Ingesta de datos mediante la SiteWise API de AWS IoT](#) en la Guía del SiteWise usuario de AWS IoT.

- Para obtener más información sobre la API, consulte [BatchPutAssetPropertyValue](#) la Referencia de AWS CLI comandos.

Java

SDK para Java 2.x

 Note

Hay más información al respecto en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
/**
 * Sends data to the SiteWise service.
 *
 * @param assetId      the ID of the asset to which the data will be sent.
 * @param tempPropertyId the ID of the temperature property.
 * @param humidityPropId the ID of the humidity property.
 * @return a {@link CompletableFuture} that represents a {@link
BatchPutAssetPropertyValueResponse} result. The
 *         calling code can attach callbacks, then handle the result or
exception by calling
 *         {@link CompletableFuture#join()} or {@link
CompletableFuture#get()}.
 *         <p>
 *         If any completion stage in this method throws an exception, the
method logs the exception cause and keeps it
 *         available to the calling code as a {@link CompletionException}. By
calling
 *         {@link CompletionException#getCause()}, the calling code can
access the original exception.
 */
public CompletableFuture<BatchPutAssetPropertyValueResponse>
sendDataToSiteWiseAsync(String assetId, String tempPropertyId, String
humidityPropId) {
    Map<String, Double> sampleData = generateSampleData();
    long timestamp = Instant.now().toEpochMilli();

    TimeInNanos time = TimeInNanos.builder()
        .timeInSeconds(timestamp / 1000)
        .offsetInNanos((int) ((timestamp % 1000) * 1000000))
        .build();
```

```
BatchPutAssetPropertyValueRequest request =
BatchPutAssetPropertyValueRequest.builder()
    .entries(Arrays.asList(
        PutAssetPropertyValueEntry.builder()
            .entryId("entry-3")
            .assetId(assetId)
            .propertyId(tempPropertyId)
            .propertyValues(Arrays.asList(
                AssetPropertyValue.builder()
                    .value(Variant.builder()
                        .doubleValue(sampleData.get("Temperature"))
                        .build())
                    .timestamp(time)
                    .build()
            ))
            .build(),
        PutAssetPropertyValueEntry.builder()
            .entryId("entry-4")
            .assetId(assetId)
            .propertyId(humidityPropId)
            .propertyValues(Arrays.asList(
                AssetPropertyValue.builder()
                    .value(Variant.builder()
                        .doubleValue(sampleData.get("Humidity"))
                        .build())
                    .timestamp(time)
                    .build()
            ))
            .build()
    ))
    .build();

return getAsyncClient().batchPutAssetPropertyValue(request)
    .whenComplete((response, exception) -> {
        if (exception != null) {
            logger.error("An exception occurred: {}",
exception.getCause().getMessage());
        }
    });
}
```

- Para obtener más información sobre la API, consulta [BatchPutAssetPropertyValue](#) la Referencia AWS SDK for Java 2.x de la API.

JavaScript

SDK para JavaScript (v3)

Note

Hay más información. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
import {
  BatchPutAssetPropertyValueCommand,
  IoTSiteWiseClient,
} from "@aws-sdk/client-iotsitewise";
import { parseArgs } from "node:util";

/**
 * Batch put asset property values.
 * @param {{ entries : array }}
 */
export const main = async ({ entries }) => {
  const client = new IoTSiteWiseClient({});
  try {
    const result = await client.send(
      new BatchPutAssetPropertyValueCommand({
        entries: entries,
      }),
    );
    console.log("Asset properties batch put successfully.");
    return result;
  } catch (caught) {
    if (caught instanceof Error && caught.name === "ResourceNotFound") {
      console.warn(`${caught.message}. A resource could not be found.`);
    } else {
      throw caught;
    }
  }
};
```

- Para obtener más información sobre la API, consulta [BatchPutAssetPropertyValue](#) la Referencia AWS SDK para JavaScript de la API.

Python

SDK para Python (Boto3)

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
class IoTSitewiseWrapper:
    """Encapsulates AWS IoT SiteWise actions using the client interface."""

    def __init__(self, iotsitewise_client: client) -> None:
        """
        Initializes the IoTSitewiseWrapper with an AWS IoT SiteWise client.

        :param iotsitewise_client: A Boto3 AWS IoT SiteWise client. This client
        provides low-level
                                access to AWS IoT SiteWise services.
        """
        self.iotsitewise_client = iotsitewise_client
        self.entry_id = 0 # Incremented to generate unique entry IDs for
        batch_put_asset_property_value.

    @classmethod
    def from_client(cls) -> "IoTSitewiseWrapper":
        """
        Creates an IoTSitewiseWrapper instance with a default AWS IoT SiteWise
        client.

        :return: An instance of IoTSitewiseWrapper initialized with the default
        AWS IoT SiteWise client.
        """
        iotsitewise_client = boto3.client("iotsitewise")
        return cls(iotsitewise_client)
```

```

def batch_put_asset_property_value(
    self, asset_id: str, values: List[Dict[str, str]]
) -> None:
    """
    Sends data to an AWS IoT SiteWise Asset.

    :param asset_id: The asset ID.
    :param values: A list of dictionaries containing the values in the form
        {propertyId : property_id,
         valueType : [stringValue|integerValue|doubleValue|
booleanValue],
         value : the_value}.
    """
    try:
        entries = self.properties_to_values(asset_id, values)
    self.iotsitewise_client.batch_put_asset_property_value(entries=entries)
    except ClientError as err:
        if err.response["Error"]["Code"] == "ResourceNotFoundException":
            logger.error("Asset %s does not exist.", asset_id)
        else:
            logger.error(
                "Error sending data to asset. Here's why %s",
                err.response["Error"]["Message"],
            )
    raise

```

Una función auxiliar para generar el parámetro de entradas a partir de una lista de valores.

```

def properties_to_values(
    self, asset_id: str, values: list[dict[str, Any]]
) -> list[dict[str, Any]]:
    """
    Utility function to convert a values list to the entries parameter for
    batch_put_asset_property_value.
    :param asset_id : The asset ID.
    :param values : A list of dictionaries containing the values in the form
        {propertyId : property_id,

```

```

        valueType : [stringValue|integerValue|doubleValue|
booleanValue],
        value : the_value}.
    :return: An entries list to pass as the 'entries' parameter to
batch_put_asset_property_value.
    """
    entries = []
    for value in values:
        epoch_ns = time.time_ns()
        self.entry_id += 1
        if value["valueType"] == "stringValue":
            property_value = {"stringValue": value["value"]}
        elif value["valueType"] == "integerValue":
            property_value = {"integerValue": value["value"]}
        elif value["valueType"] == "booleanValue":
            property_value = {"booleanValue": value["value"]}
        elif value["valueType"] == "doubleValue":
            property_value = {"doubleValue": value["value"]}
        else:
            raise ValueError("Invalid valueType: %s", value["valueType"])
        entry = {
            "entryId": f"{self.entry_id}",
            "assetId": asset_id,
            "propertyId": value["propertyId"],
            "propertyValues": [
                {
                    "value": property_value,
                    "timestamp": {
                        "timeInSeconds": int(epoch_ns / 1000000000),
                        "offsetInNanos": epoch_ns % 1000000000,
                    },
                },
            ],
        }
        entries.append(entry)
    return entries

```

A continuación se muestra un ejemplo de una lista de valores que se pasan a la función auxiliar.

```
values = [
```

```
{
  "propertyId": humidity_property_id,
  "valueType": "doubleValue",
  "value": 65.0,
},
{
  "propertyId": temperature_property_id,
  "valueType": "doubleValue",
  "value": 23.5,
},
]
```

- Para obtener más información sobre la API, consulta [BatchPutAssetPropertyValue](#) la AWS Referencia de API de SDK for Python (Boto3).

SAP ABAP

SDK para SAP ABAP

Note

Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
TRY.
  lo_ios->batchputassetpropertyvalue(
    it_entries = it_entries
  ).
  MESSAGE 'Data sent to IoT SiteWise asset successfully.' TYPE 'I'.
CATCH /aws1/cx_iosresourcenotfoundex.
  MESSAGE 'Asset does not exist.' TYPE 'E'.
ENDTRY.
```

- Para obtener más información sobre la API, consulte [BatchPutAssetPropertyValue](#) la referencia sobre la API ABAP del AWS SDK para SAP.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulte [Uso de este servicio con un AWS SDK](#). En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **CreateAsset** con un AWS SDK o CLI

Los siguientes ejemplos de código muestran cómo utilizar CreateAsset.

CLI

AWS CLI

Para crear un activo

En el siguiente ejemplo de `create-asset`, se crea un activo de turbina eólica a partir de un modelo de activos de turbina eólica.

```
aws iotsitewise create-asset \  
  --asset-model-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \  
  --asset-name "Wind Turbine 1"
```

Salida:


```
{  
  "assetId": "a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",  
  "assetArn": "arn:aws:iotsitewise:us-west-2:123456789012:asset/  
a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",  
  "assetStatus": {  
    "state": "CREATING"  
  }  
}
```

Para obtener más información, consulte [Creación de activos](#) en la Guía del SiteWise usuario de AWS IoT.

- Para obtener más información sobre la API, consulte [CreateAsset](#) la Referencia de AWS CLI comandos.

Java

SDK para Java 2.x

 Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
/**
 * Creates an asset with the specified name and asset model Id.
 *
 * @param assetName    the name of the asset to create.
 * @param assetModelId the Id of the asset model to associate with the asset.
 * @return a {@link CompletableFuture} that represents a {@link
CreateAssetResponse} result. The calling code can
 *         attach callbacks, then handle the result or exception by calling
{@link CompletableFuture#join()} or
 *         {@link CompletableFuture#get()}.
 *         <p>
 *         If any completion stage in this method throws an exception, the
method logs the exception cause and keeps it
 *         available to the calling code as a {@link CompletionException}. By
calling
 *         {@link CompletionException#getCause()}, the calling code can
access the original exception.
 */
public CompletableFuture<CreateAssetResponse> createAssetAsync(String
assetName, String assetModelId) {
    CreateAssetRequest createAssetRequest = CreateAssetRequest.builder()
        .assetModelId(assetModelId)
        .assetDescription("Created using the AWS SDK for Java")
        .assetName(assetName)
        .build();

    return getAsyncClient().createAsset(createAssetRequest)
        .whenComplete((response, exception) -> {
            if (exception != null) {
                logger.error("Failed to create asset: {}",
exception.getCause().getMessage());
            }
        });
}
```

```

    }
    });
}

```

- Para obtener más información sobre la API, consulta [CreateAsset](#) la Referencia AWS SDK for Java 2.x de la API.

JavaScript

SDK para JavaScript (v3)

Note

Hay más información. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```

import {
  CreateAssetCommand,
  IoTSiteWiseClient,
} from "@aws-sdk/client-iotsitewise";
import { parseArgs } from "node:util";

/**
 * Create an Asset.
 * @param {{ assetName : string, assetModelId: string }}
 */
export const main = async ({ assetName, assetModelId }) => {
  const client = new IoTSiteWiseClient({});
  try {
    const result = await client.send(
      new CreateAssetCommand({
        assetName: assetName, // The name to give the Asset.
        assetModelId: assetModelId, // The ID of the asset model from which to
        create the asset.
      })),
    );
    console.log("Asset created successfully.");
    return result;
  } catch (caught) {

```

```

    if (caught instanceof Error && caught.name === "ResourceNotFound") {
        console.warn(
            `${caught.message}. The asset model could not be found. Please check the
            asset model id.`
        );
    } else {
        throw caught;
    }
}
};

```

- Para obtener más información sobre la API, consulta [CreateAsset](#) la Referencia AWS SDK para JavaScript de la API.

Python

SDK para Python (Boto3)

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```

class IoTSiteWiseWrapper:
    """Encapsulates AWS IoT SiteWise actions using the client interface."""

    def __init__(self, iotsitewise_client: client) -> None:
        """
        Initializes the IoTSiteWiseWrapper with an AWS IoT SiteWise client.

        :param iotsitewise_client: A Boto3 AWS IoT SiteWise client. This client
        provides low-level
            access to AWS IoT SiteWise services.
        """
        self.iotsitewise_client = iotsitewise_client
        self.entry_id = 0 # Incremented to generate unique entry IDs for
        batch_put_asset_property_value.

    @classmethod

```

```
def from_client(cls) -> "IoTSitewiseWrapper":
    """
    Creates an IoTSitewiseWrapper instance with a default AWS IoT SiteWise
    client.

    :return: An instance of IoTSitewiseWrapper initialized with the default
    AWS IoT SiteWise client.
    """
    iotsitewise_client = boto3.client("iotsitewise")
    return cls(iotsitewise_client)

def create_asset(self, asset_name: str, asset_model_id: str) -> str:
    """
    Creates an AWS IoT SiteWise Asset.

    :param asset_name: The name of the asset to create.
    :param asset_model_id: The ID of the asset model to associate with the
    asset.
    :return: The ID of the created asset.
    """
    try:
        response = self.iotsitewise_client.create_asset(
            assetName=asset_name, assetModelId=asset_model_id
        )
        asset_id = response["assetId"]
        waiter = self.iotsitewise_client.get_waiter("asset_active")
        waiter.wait(assetId=asset_id)
        return asset_id
    except ClientError as err:
        if err.response["Error"] == "ResourceNotFoundException":
            logger.error("Asset model %s does not exist.", asset_model_id)
        else:
            logger.error(
                "Error creating asset %s. Here's why %s",
                asset_name,
                err.response["Error"]["Message"],
            )
        raise
```

- Para obtener más información sobre la API, consulta [CreateAsset](#) la AWS Referencia de API de SDK for Python (Boto3).

SAP ABAP

SDK para SAP ABAP

Note

Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
TRY.  
    oo_result = lo_ios->createasset(  
        iv_assetname = iv_asset_name  
        iv_assetmodelid = iv_asset_model_id  
    ). " oo_result is returned for testing purposes. "  
    MESSAGE 'IoT SiteWise asset created' TYPE 'I'.  
CATCH /aws1/cx_iosresourcenotfoundex.  
    MESSAGE 'Asset model does not exist.' TYPE 'E'.  
ENDTRY.
```

- Para obtener más información sobre la API, consulte [CreateAsset](#) la referencia sobre la API ABAP del AWS SDK para SAP.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulte [Uso de este servicio con un AWS SDK](#). En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **CreateAssetModel** con un AWS SDK o CLI

Los siguientes ejemplos de código muestran cómo utilizar `CreateAssetModel`.

CLI

AWS CLI

Para crear un modelo de activos

En el siguiente ejemplo de `create-asset-model`, se crea un modelo de activos que define una turbina eólica con las siguientes propiedades:

Número de serie: el número de serie de una turbina eólica
Potencia generada por una turbina eólica
El flujo de datos de energía generada por una turbina eólica
Temperatura C - El flujo de datos de temperatura de una turbina eólica en Celsius
Temperatura F - Los puntos de datos de temperatura mapeados de grados Celsius a Fahrenheit

```
aws iotsitewise create-asset-model \  
  --cli-input-json file://create-wind-turbine-model.json
```

Contenido de `create-wind-turbine-model.json`:

```
{  
  "assetModelName": "Wind Turbine Model",  
  "assetModelDescription": "Represents a wind turbine",  
  "assetModelProperties": [  
    {  
      "name": "Serial Number",  
      "dataType": "STRING",  
      "type": {  
        "attribute": {}  
      }  
    },  
    {  
      "name": "Generated Power",  
      "dataType": "DOUBLE",  
      "unit": "kW",  
      "type": {  
        "measurement": {}  
      }  
    },  
    {  
      "name": "Temperature C",  
      "dataType": "DOUBLE",  
      "unit": "Celsius",  
      "type": {  
        "measurement": {}  
      }  
    },  
    {  
      "name": "Temperature F",
```

```
"dataType": "DOUBLE",
"unit": "Fahrenheit",
"type": {
  "transform": {
    "expression": "temp_c * 9 / 5 + 32",
    "variables": [
      {
        "name": "temp_c",
        "value": {
          "propertyId": "Temperature C"
        }
      }
    ]
  }
},
{
  "name": "Total Generated Power",
  "dataType": "DOUBLE",
  "unit": "kW",
  "type": {
    "metric": {
      "expression": "sum(power)",
      "variables": [
        {
          "name": "power",
          "value": {
            "propertyId": "Generated Power"
          }
        }
      ],
      "window": {
        "tumbling": {
          "interval": "1h"
        }
      }
    }
  }
}
]
```

Salida:

```
{
  "assetModelId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
  "assetModelArn": "arn:aws:iotsitewise:us-west-2:123456789012:asset-model/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
  "assetModelStatus": {
    "state": "CREATING"
  }
}
```

Para obtener más información, consulte [Definir modelos de activos](#) en la Guía del SiteWise usuario de AWS IoT.

- Para obtener más información sobre la API, consulte [CreateAssetModel](#) en la Referencia de AWS CLI comandos.

Java

SDK para Java 2.x

Note

Hay más información al respecto en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
/**
 * Creates an asset model.
 *
 * @param name the name of the asset model to create.
 * @return a {@link CompletableFuture} that represents a {@link
 * CreateAssetModelResponse} result. The calling code
 * can attach callbacks, then handle the result or exception by
 * calling {@link CompletableFuture#join()} or
 * {@link CompletableFuture#get()}.
 *
 * <p>
 * If any completion stage in this method throws an exception, the
 * method logs the exception cause and keeps it
 * available to the calling code as a {@link CompletionException}. By
 * calling
```

```
    *      {@link CompletionException#getCause()}, the calling code can
    access the original exception.
    */
    public CompletableFuture<CreateAssetModelResponse>
    createAssetModelAsync(String name) {
        PropertyType humidity = PropertyType.builder()
            .measurement(Measurement.builder().build())
            .build();

        PropertyType temperaturePropertyType = PropertyType.builder()
            .measurement(Measurement.builder().build())
            .build();

        AssetModelPropertyDefinition temperatureProperty =
        AssetModelPropertyDefinition.builder()
            .name("Temperature")
            .dataType(PropertyDataType.DOUBLE)
            .type(temperaturePropertyType)
            .build();

        AssetModelPropertyDefinition humidityProperty =
        AssetModelPropertyDefinition.builder()
            .name("Humidity")
            .dataType(PropertyDataType.DOUBLE)
            .type(humidity)
            .build();

        CreateAssetModelRequest createAssetModelRequest =
        CreateAssetModelRequest.builder()
            .assetModelName(name)
            .assetModelDescription("This is my asset model")
            .assetModelProperties(temperatureProperty, humidityProperty)
            .build();

        return getAsyncClient().createAssetModel(createAssetModelRequest)
            .whenComplete((response, exception) -> {
                if (exception != null) {
                    logger.error("Failed to create asset model: {}",
                    exception.getCause().getMessage());
                }
            });
    }
}
```

- Para obtener más información sobre la API, consulta [CreateAssetModel](#) la Referencia AWS SDK for Java 2.x de la API.

JavaScript

SDK para JavaScript (v3)

Note

Hay más información. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
import {
  CreateAssetModelCommand,
  IoTSiteWiseClient,
} from "@aws-sdk/client-iotsitewise";
import { parseArgs } from "node:util";

/**
 * Create an Asset Model.
 * @param {{ assetName : string, assetModelId: string }}
 */
export const main = async ({ assetModelName, assetModelId }) => {
  const client = new IoTSiteWiseClient({});
  try {
    const result = await client.send(
      new CreateAssetModelCommand({
        assetModelName: assetModelName, // The name to give the Asset Model.
      }),
    );
    console.log("Asset model created successfully.");
    return result;
  } catch (caught) {
    if (caught instanceof Error && caught.name === "IoTSiteWiseError") {
      console.warn(
        `${caught.message}. There was a problem creating the asset model.`
      );
    } else {
      throw caught;
    }
  }
}
```

```
}  
};
```

- Para obtener más información sobre la API, consulta [CreateAssetModel](#) la Referencia AWS SDK para JavaScript de la API.

Python

SDK para Python (Boto3)

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
class IoTSiteWiseWrapper:  
    """Encapsulates AWS IoT SiteWise actions using the client interface."""  
  
    def __init__(self, iotsitewise_client: client) -> None:  
        """  
        Initializes the IoTSiteWiseWrapper with an AWS IoT SiteWise client.  
  
        :param iotsitewise_client: A Boto3 AWS IoT SiteWise client. This client  
        provides low-level  
            access to AWS IoT SiteWise services.  
        """  
        self.iotsitewise_client = iotsitewise_client  
        self.entry_id = 0 # Incremented to generate unique entry IDs for  
        batch_put_asset_property_value.  
  
    @classmethod  
    def from_client(cls) -> "IoTSiteWiseWrapper":  
        """  
        Creates an IoTSiteWiseWrapper instance with a default AWS IoT SiteWise  
        client.  
  
        :return: An instance of IoTSiteWiseWrapper initialized with the default  
        AWS IoT SiteWise client.  
        """
```

```

iotsitewise_client = boto3.client("iotsitewise")
return cls(iotsitewise_client)

def create_asset_model(
    self, asset_model_name: str, properties: List[Dict[str, Any]]
) -> str:
    """
    Creates an AWS IoT SiteWise Asset Model.

    :param asset_model_name: The name of the asset model to create.
    :param properties: The property definitions of the asset model.
    :return: The ID of the created asset model.
    """
    try:
        response = self.iotsitewise_client.create_asset_model(
            assetModelName=asset_model_name,
            assetModelDescription="This is a sample asset model
description.",
            assetModelProperties=properties,
        )
        asset_model_id = response["assetModelId"]
        waiter = self.iotsitewise_client.get_waiter("asset_model_active")
        waiter.wait(assetModelId=asset_model_id)
        return asset_model_id
    except ClientError as err:
        if err.response["Error"]["Code"] == "ResourceAlreadyExistsException":
            logger.error("Asset model %s already exists.", asset_model_name)
        else:
            logger.error(
                "Error creating asset model %s. Here's why %s",
                asset_model_name,
                err.response["Error"]["Message"],
            )
        raise

```

A continuación se muestra un ejemplo de una lista de valores que se pasan a la función.

```

properties = [
    {
        "name": temperature_property_name,

```

```

        "dataType": "DOUBLE",
        "type": {
            "measurement": {},
        },
    },
    {
        "name": humidity_property_name,
        "dataType": "DOUBLE",
        "type": {
            "measurement": {},
        },
    },
]

```

- Para obtener más información sobre la API, consulta [CreateAssetModel](#) en la AWS Referencia de API de SDK for Python (Boto3).

SAP ABAP

SDK para SAP ABAP

Note

Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```

TRY.
    oo_result = lo_ios->createassetmodel(
        iv_assetmodelname = iv_asset_model_name
        iv_assetmodeldescription = 'This is a sample asset model description.'
        it_assetmodelproperties = it_properties
    ). " oo_result is returned for testing purposes. "
    MESSAGE 'IoT SiteWise asset model created' TYPE 'I'.
CATCH /aws1/cx_iosresrcalrddyexistsex.
    MESSAGE 'Asset model already exists.' TYPE 'E'.
ENDTRY.

```

- Para obtener más información sobre la API, consulte [CreateAssetModel](#) la referencia sobre la API ABAP del AWS SDK para SAP.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulte [Uso de este servicio con un AWS SDK](#). En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **CreateGateway** con un AWS SDK o CLI

Los siguientes ejemplos de código muestran cómo utilizar CreateGateway.

CLI

AWS CLI

Para crear una puerta de enlace

El siguiente create-gateway ejemplo crea una puerta de enlace que se ejecuta en AWS IoT Greengrass.

```
aws iotsitewise create-gateway \  
  --gateway-name ExampleCorpGateway \  
  --gateway-platform greengrass={groupArn=arn:aws:greengrass:us-west-2:123456789012:/greengrass/groups/a1b2c3d4-5678-90ab-cdef-1b1b1EXAMPLE}
```

Salida:


```
{  
  "gatewayId": "a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE",  
  "gatewayArn": "arn:aws:iotsitewise:us-west-2:123456789012:gateway/  
a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE"  
}
```

Para obtener más información, consulte [Configuración de una puerta de enlace](#) en la Guía del SiteWise usuario de AWS IoT.

- Para obtener más información sobre la API, consulte [CreateGateway](#) la Referencia de AWS CLI comandos.

Java

SDK para Java 2.x

 Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
/**
 * Creates a new IoT SiteWise gateway.
 *
 * @param gatewayName The name of the gateway to create.
 * @param myThing      The name of the core device thing to associate with the
 gateway.
 * @return a {@link CompletableFuture} that represents a {@link String}
 result of the gateways ID. The calling code
 *         can attach callbacks, then handle the result or exception by
 calling {@link CompletableFuture#join()} or
 *         {@link CompletableFuture#get()}.
 *         <p>
 *         If any completion stage in this method throws an exception, the
 method logs the exception cause and keeps
 *         it available to the calling code as a {@link CompletionException}.
 By calling
 *         {@link CompletionException#getCause()}, the calling code can
 access the original exception.
 */
public CompletableFuture<String> createGatewayAsync(String gatewayName,
String myThing) {
    GreengrassV2 gg = GreengrassV2.builder()
        .coreDeviceThingName(myThing)
        .build();

    GatewayPlatform platform = GatewayPlatform.builder()
        .greengrassV2(gg)
        .build();

    Map<String, String> tag = new HashMap<>();
    tag.put("Environment", "Production");
```

```
        CreateGatewayRequest createGatewayRequest =
CreateGatewayRequest.builder()
    .gatewayName(gatewayName)
    .gatewayPlatform(platform)
    .tags(tag)
    .build();

return getAsyncClient().createGateway(createGatewayRequest)
    .handle((response, exception) -> {
        if (exception != null) {
            logger.error("Error creating the gateway.");
            throw (CompletionException) exception;
        }
        logger.info("The ARN of the gateway is {}" ,
response.gatewayArn());
        return response.gatewayId();
    });
}
```

- Para obtener más información sobre la API, consulta [CreateGateway](#) la Referencia AWS SDK for Java 2.x de la API.

JavaScript

SDK para JavaScript (v3)

Note

Hay más información. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
import {
    CreateGatewayCommand,
    IoTSiteWiseClient,
} from "@aws-sdk/client-iotsitewise";
import { parseArgs } from "node:util";

/**
```

```

* Create a Gateway.
* @param {{ }}
*/
export const main = async ({ gatewayName }) => {
  const client = new IoTSiteWiseClient({});
  try {
    const result = await client.send(
      new CreateGatewayCommand({
        gatewayName: gatewayName, // The name to give the created Gateway.
      }),
    );
    console.log("Gateway created successfully.");
    return result;
  } catch (caught) {
    if (caught instanceof Error && caught.name === "IoTSiteWiseError") {
      console.warn(
        `${caught.message}. There was a problem creating the Gateway.`
      );
    } else {
      throw caught;
    }
  }
};

```

- Para obtener más información sobre la API, consulta [CreateGateway](#) la Referencia AWS SDK para JavaScript de la API.

Python

SDK para Python (Boto3)

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```

class IoTSitewiseWrapper:
    """Encapsulates AWS IoT SiteWise actions using the client interface."""

```

```
def __init__(self, iotsitewise_client: client) -> None:
    """
    Initializes the IoTSitewiseWrapper with an AWS IoT SiteWise client.

    :param iotsitewise_client: A Boto3 AWS IoT SiteWise client. This client
    provides low-level
        access to AWS IoT SiteWise services.
    """
    self.iotsitewise_client = iotsitewise_client
    self.entry_id = 0 # Incremented to generate unique entry IDs for
    batch_put_asset_property_value.

    @classmethod
    def from_client(cls) -> "IoTSitewiseWrapper":
        """
        Creates an IoTSitewiseWrapper instance with a default AWS IoT SiteWise
        client.

        :return: An instance of IoTSitewiseWrapper initialized with the default
        AWS IoT SiteWise client.
        """
        iotsitewise_client = boto3.client("iotsitewise")
        return cls(iotsitewise_client)

def create_gateway(self, gateway_name: str, my_thing: str) -> str:
    """
    Creates an AWS IoT SiteWise Gateway.

    :param gateway_name: The name of the gateway to create.
    :param my_thing: The core device thing name.
    :return: The ID of the created gateway.
    """
    try:
        response = self.iotsitewise_client.create_gateway(
            gatewayName=gateway_name,
            gatewayPlatform={
                "greengrassV2": {"coreDeviceThingName": my_thing},
            },
            tags={"Environment": "Production"},
        )
        gateway_id = response["gatewayId"]
        return gateway_id
    except ClientError as err:
```

```

    if err.response["Error"]["Code"] == "ResourceAlreadyExistsException":
        logger.error("Gateway %s already exists.", gateway_name)
    else:
        logger.error(
            "Error creating gateway %s. Here's why %s",
            gateway_name,
            err.response["Error"]["Message"],
        )
    raise

```

- Para obtener más información sobre la API, consulta [CreateGateway](#) la AWS Referencia de API de SDK for Python (Boto3).

SAP ABAP

SDK para SAP ABAP

Note

Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```

TRY.
    oo_result = lo_ios->creategateway(
        iv_gatewayname = iv_gateway_name
        io_gatewayplatform = NEW /aws1/cl_iosgatewayplatform(
            io_greenrassv2 = NEW /aws1/cl_iosgreenrassv2(
                iv_coredevicethingname = iv_core_device_thing_name
            )
        )
    )
    it_tags = VALUE /aws1/cl_iostagmap_w=>tt_tagmap(
        (
            VALUE /aws1/cl_iostagmap_w=>ts_tagmap_maprow(
                key = 'Environment'
                value = NEW /aws1/cl_iostagmap_w( 'Production' )
            )
        )
    )
)

```

```

    ). " oo_result is returned for testing purposes. "
    MESSAGE 'IoT SiteWise gateway created' TYPE 'I'.
  CATCH /aws1/cx_iosresrcalrddyexistsex.
    MESSAGE 'Gateway already exists.' TYPE 'E'.
ENDTRY.

```

- Para obtener más información sobre la API, consulte [CreateGateway](#) la referencia sobre la API ABAP del AWS SDK para SAP.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulte [Uso de este servicio con un AWS SDK](#). En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **DeleteAsset** con un AWS SDK o CLI

Los siguientes ejemplos de código muestran cómo utilizar DeleteAsset.

CLI

AWS CLI

Para eliminar un activo

En el siguiente ejemplo de `delete-asset`, se elimina un activo de turbina eólica.

```

aws iotsitewise delete-asset \
  --asset-id a1b2c3d4-5678-90ab-cdef-3333EXAMPLE

```

Salida:

```

{
  "assetStatus": {
    "state": "DELETING"
  }
}


```

Para obtener más información, consulte [Eliminar activos](#) en la Guía del SiteWise usuario de AWS IoT.

- Para obtener más información sobre la API, consulte [DeleteAsset](#) la Referencia de AWS CLI comandos.

Java

SDK para Java 2.x

 Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
/**
 * Deletes an asset.
 *
 * @param assetId the ID of the asset to be deleted.
 * @return a {@link CompletableFuture} that represents a {@link
DeleteAssetResponse} result. The calling code can
 *         attach callbacks, then handle the result or exception by calling
{@link CompletableFuture#join()} or
 *         {@link CompletableFuture#get()}.
 *         <p>
 *         If any completion stage in this method throws an exception, the
method logs the exception cause and keeps
 *         it available to the calling code as a {@link CompletionException}.
By calling
 *         {@link CompletionException#getCause()}, the calling code can
access the original exception.
 */
public CompletableFuture<DeleteAssetResponse> deleteAssetAsync(String
assetId) {
    DeleteAssetRequest deleteAssetRequest = DeleteAssetRequest.builder()
        .assetId(assetId)
        .build();

    return getAsyncClient().deleteAsset(deleteAssetRequest)
        .whenComplete((response, exception) -> {
            if (exception != null) {
                logger.error("An error occurred deleting asset with id: {}",
assetId);
            }
        });
}
```

- Para obtener más información sobre la API, consulta [DeleteAsset](#) la Referencia AWS SDK for Java 2.x de la API.

JavaScript

SDK para JavaScript (v3)

Note

Hay más información. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
import {
  DeleteAssetCommand,
  IoTSiteWiseClient,
} from "@aws-sdk/client-iotsitewise";
import { parseArgs } from "node:util";

/**
 * Delete an asset.
 * @param {{ assetId : string }}
 */
export const main = async ({ assetId }) => {
  const client = new IoTSiteWiseClient({});
  try {
    await client.send(
      new DeleteAssetCommand({
        assetId: assetId, // The model id to delete.
      }),
    );
    console.log("Asset deleted successfully.");
    return { assetDeleted: true };
  } catch (caught) {
    if (caught instanceof Error && caught.name === "ResourceNotFound") {
      console.warn(
        `${caught.message}. There was a problem deleting the asset.`
      );
    } else {
      throw caught;
    }
  }
}
```

```

    }
};

```

- Para obtener más información sobre la API, consulta [DeleteAsset](#) la Referencia AWS SDK para JavaScript de la API.

Python

SDK para Python (Boto3)

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```

class IoTSiteWiseWrapper:
    """Encapsulates AWS IoT SiteWise actions using the client interface."""

    def __init__(self, iotsitewise_client: client) -> None:
        """
        Initializes the IoTSiteWiseWrapper with an AWS IoT SiteWise client.

        :param iotsitewise_client: A Boto3 AWS IoT SiteWise client. This client
        provides low-level
            access to AWS IoT SiteWise services.
        """
        self.iotsitewise_client = iotsitewise_client
        self.entry_id = 0 # Incremented to generate unique entry IDs for
        batch_put_asset_property_value.

    @classmethod
    def from_client(cls) -> "IoTSiteWiseWrapper":
        """
        Creates an IoTSiteWiseWrapper instance with a default AWS IoT SiteWise
        client.

        :return: An instance of IoTSiteWiseWrapper initialized with the default
        AWS IoT SiteWise client.
        """

```

```

iotsitewise_client = boto3.client("iotsitewise")
return cls(iotsitewise_client)

def delete_asset(self, asset_id: str) -> None:
    """
    Deletes an AWS IoT SiteWise Asset.

    :param asset_id: The ID of the asset to delete.
    """
    try:
        self.iotsitewise_client.delete_asset(assetId=asset_id)
    except ClientError as err:
        logger.error(
            "Error deleting asset %s. Here's why %s",
            asset_id,
            err.response["Error"]["Message"],
        )
        raise

```

- Para obtener más información sobre la API, consulta [DeleteAsset](#) la AWS Referencia de API de SDK for Python (Boto3).

SAP ABAP

SDK para SAP ABAP

Note

Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```

TRY.
  lo_ios->deleteasset(
    iv_assetid = iv_asset_id
  ).
  MESSAGE 'IoT SiteWise asset deleted.' TYPE 'I'.
CATCH /aws1/cx_rt_generic.

```

```
MESSAGE 'Unable to delete asset.' TYPE 'E'.  
ENDTRY.
```

- Para obtener más información sobre la API, consulte [DeleteAsset](#) la referencia sobre la API ABAP del AWS SDK para SAP.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulte [Uso de este servicio con un AWS SDK](#). En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo `DeleteAssetModel` con un AWS SDK o CLI

Los siguientes ejemplos de código muestran cómo utilizar `DeleteAssetModel`.

CLI

AWS CLI

Para eliminar un modelo de activos

En el siguiente ejemplo de `delete-asset-model`, se elimina un modelo de activos de turbina eólica.

```
aws iotsitewise delete-asset-model \  
  --asset-model-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
```

Salida:


```
{  
  "assetModelStatus": {  
    "state": "DELETING"  
  }  
}
```

Para obtener más información, consulte [Eliminar modelos de activos](#) en la Guía del SiteWise usuario de AWS IoT.

- Para obtener más información sobre la API, consulte [DeleteAssetModel](#) la Referencia de AWS CLI comandos.

Java

SDK para Java 2.x

 Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
/**
 * Deletes an Asset Model with the specified ID.
 *
 * @param assetModelId the ID of the Asset Model to delete.
 * @return a {@link CompletableFuture} that represents a {@link
DeleteAssetModelResponse} result. The calling code
 *         can attach callbacks, then handle the result or exception by
calling {@link CompletableFuture#join()} or
 *         {@link CompletableFuture#get()}.
 *         <p>
 *         If any completion stage in this method throws an exception, the
method logs the exception cause and keeps
 *         it available to the calling code as a {@link CompletionException}.
By calling
 *         {@link CompletionException#getCause()}, the calling code can
access the original exception.
 */
public CompletableFuture<DeleteAssetModelResponse>
deleteAssetModelAsync(String assetModelId) {
    DeleteAssetModelRequest deleteAssetModelRequest =
DeleteAssetModelRequest.builder()
        .assetModelId(assetModelId)
        .build();

    return getAsyncClient().deleteAssetModel(deleteAssetModelRequest)
        .whenComplete((response, exception) -> {
            if (exception != null) {
                logger.error("Failed to delete asset model with ID:{}",
exception.getMessage());
            }
        });
}
```

- Para obtener más información sobre la API, consulta [DeleteAssetModel](#) la Referencia AWS SDK for Java 2.x de la API.

JavaScript

SDK para JavaScript (v3)

Note

Hay más información. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
import {
  DeleteAssetModelCommand,
  IoTSiteWiseClient,
} from "@aws-sdk/client-iotsitewise";
import { parseArgs } from "node:util";

/**
 * Delete an asset model.
 * @param {{ assetModelId : string }}
 */
export const main = async ({ assetModelId }) => {
  const client = new IoTSiteWiseClient({});
  try {
    await client.send(
      new DeleteAssetModelCommand({
        assetModelId: assetModelId, // The model id to delete.
      }),
    );
    console.log("Asset model deleted successfully.");
    return { assetModelDeleted: true };
  } catch (caught) {
    if (caught instanceof Error && caught.name === "ResourceNotFound") {
      console.warn(
        `${caught.message}. There was a problem deleting the asset model.`
      );
    } else {
```

```

        throw caught;
    }
}
};

```

- Para obtener más información sobre la API, consulta [DeleteAssetModel](#) la Referencia AWS SDK para JavaScript de la API.

Python

SDK para Python (Boto3)

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```

class IoTSitewiseWrapper:
    """Encapsulates AWS IoT SiteWise actions using the client interface."""

    def __init__(self, iotsitewise_client: client) -> None:
        """
        Initializes the IoTSitewiseWrapper with an AWS IoT SiteWise client.

        :param iotsitewise_client: A Boto3 AWS IoT SiteWise client. This client
        provides low-level
                                access to AWS IoT SiteWise services.
        """
        self.iotsitewise_client = iotsitewise_client
        self.entry_id = 0 # Incremented to generate unique entry IDs for
        batch_put_asset_property_value.

    @classmethod
    def from_client(cls) -> "IoTSitewiseWrapper":
        """
        Creates an IoTSitewiseWrapper instance with a default AWS IoT SiteWise
        client.

```

```

        :return: An instance of IoTSiteWiseWrapper initialized with the default
AWS IoT SiteWise client.
        """
        iotsitewise_client = boto3.client("iotsitewise")
        return cls(iotsitewise_client)

def delete_asset_model(self, asset_model_id: str) -> None:
    """
    Deletes an AWS IoT SiteWise Asset Model.

    :param asset_model_id: The ID of the asset model to delete.
    """
    try:

self.iotsitewise_client.delete_asset_model(assetModelId=asset_model_id)
    except ClientError as err:
        logger.error(
            "Error deleting asset model %s. Here's why %s",
            asset_model_id,
            err.response["Error"]["Message"],
        )
        raise

```

- Para obtener más información sobre la API, consulta [DeleteAssetModel](#) la AWS Referencia de API de SDK for Python (Boto3).

SAP ABAP

SDK para SAP ABAP

Note

Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

TRY.

```
lo_ios->deleteassetmodel(
```

```
        iv_assetmodelid = iv_asset_model_id
    ).
    MESSAGE 'IoT SiteWise asset model deleted.' TYPE 'I'.
CATCH /aws1/cx_rt_generic.
    MESSAGE 'Unable to delete asset model.' TYPE 'E'.
ENDTRY.
```

- Para obtener más información sobre la API, consulte [DeleteAssetModel](#) la referencia sobre la API ABAP del AWS SDK para SAP.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulte [Uso de este servicio con un AWS SDK](#). En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **DeleteGateway** con un AWS SDK o CLI

Los siguientes ejemplos de código muestran cómo utilizar DeleteGateway.

CLI

AWS CLI

Para eliminar una puerta de enlace

En el siguiente ejemplo de `delete-gateway`, se elimina una puerta de enlace.

```
aws iotsitewise delete-gateway \  
  --gateway-id a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE
```


Este comando no genera ninguna salida.

Para obtener más información, consulte [Ingesta de datos mediante una puerta de enlace](#) en la Guía del SiteWise usuario de AWS IoT.

- Para obtener más información sobre la API, consulte [DeleteGateway](#) la Referencia de AWS CLI comandos.

Java

SDK para Java 2.x

 Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
/**
 * Deletes the specified gateway.
 *
 * @param gatewayId the ID of the gateway to delete.
 * @return a {@link CompletableFuture} that represents a {@link
DeleteGatewayResponse} result.. The calling code
 *         can attach callbacks, then handle the result or exception by
calling {@link CompletableFuture#join()} or
 *         {@link CompletableFuture#get()}.
 *         <p>
 *         If any completion stage in this method throws an exception, the
method logs the exception cause and keeps
 *         it available to the calling code as a {@link CompletionException}.
By calling
 *         {@link CompletionException#getCause()}, the calling code can
access the original exception.
 */
public CompletableFuture<DeleteGatewayResponse> deleteGatewayAsync(String
gatewayId) {
    DeleteGatewayRequest deleteGatewayRequest =
DeleteGatewayRequest.builder()
        .gatewayId(gatewayId)
        .build();

    return getAsyncClient().deleteGateway(deleteGatewayRequest)
        .whenComplete((response, exception) -> {
            if (exception != null) {
                logger.error("Failed to delete gateway: {}",
exception.getCause().getMessage());
            }
        });
}
```

- Para obtener más información sobre la API, consulta [DeleteGateway](#) la Referencia AWS SDK for Java 2.x de la API.

JavaScript

SDK para JavaScript (v3)

Note

Hay más información. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
import {
  DeleteGatewayCommand,
  IoTSiteWiseClient,
} from "@aws-sdk/client-iotsitewise";
import { parseArgs } from "node:util";

/**
 * Create an SSM document.
 * @param {{ content: string, name: string, documentType?: DocumentType }}
 */
export const main = async ({ gatewayId }) => {
  const client = new IoTSiteWiseClient({});
  try {
    await client.send(
      new DeleteGatewayCommand({
        gatewayId: gatewayId, // The ID of the Gateway to describe.
      }),
    );
    console.log("Gateway deleted successfully.");
    return { gatewayDeleted: true };
  } catch (caught) {
    if (caught instanceof Error && caught.name === "ResourceNotFound") {
      console.warn(
        `${caught.message}. The Gateway could not be found. Please check the Gateway Id.`
      );
    }
  }
};
```

```
    } else {  
        throw caught;  
    }  
}  
};
```

- Para obtener más información sobre la API, consulta [DeleteGateway](#) la Referencia AWS SDK para JavaScript de la API.

Python

SDK para Python (Boto3)

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
class IoTSitewiseWrapper:  
    """Encapsulates AWS IoT SiteWise actions using the client interface."""  
  
    def __init__(self, iotsitewise_client: client) -> None:  
        """  
        Initializes the IoTSitewiseWrapper with an AWS IoT SiteWise client.  
  
        :param iotsitewise_client: A Boto3 AWS IoT SiteWise client. This client  
        provides low-level  
            access to AWS IoT SiteWise services.  
        """  
        self.iotsitewise_client = iotsitewise_client  
        self.entry_id = 0 # Incremented to generate unique entry IDs for  
        batch_put_asset_property_value.  
  
    @classmethod  
    def from_client(cls) -> "IoTSitewiseWrapper":  
        """  
        Creates an IoTSitewiseWrapper instance with a default AWS IoT SiteWise  
        client.
```

```
        :return: An instance of IoTSiteWiseWrapper initialized with the default
AWS IoT SiteWise client.
        """
        iotsitewise_client = boto3.client("iotsitewise")
        return cls(iotsitewise_client)

def delete_gateway(self, gateway_id: str) -> None:
    """
    Deletes an AWS IoT SiteWise Gateway.

    :param gateway_id: The ID of the gateway to delete.
    """
    try:
        self.iotsitewise_client.delete_gateway(gatewayId=gateway_id)
    except ClientError as err:
        if err.response["Error"]["Code"] == "ResourceNotFoundException":
            logger.error("Gateway %s does not exist.", gateway_id)
        else:
            logger.error(
                "Error deleting gateway %s. Here's why %s",
                gateway_id,
                err.response["Error"]["Message"],
            )
            raise
```

- Para obtener más información sobre la API, consulta [DeleteGateway](#) la AWS Referencia de API de SDK for Python (Boto3).

SAP ABAP

SDK para SAP ABAP

Note

Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```

TRY.
  lo_ios->deletegateway(
    iv_gatewayid = iv_gateway_id
  ).
  MESSAGE 'IoT SiteWise gateway deleted.' TYPE 'I'.
CATCH /aws1/cx_iosresourcenotfoundex.
  MESSAGE 'Gateway does not exist.' TYPE 'E'.
ENDTRY.

```

- Para obtener más información sobre la API, consulte [DeleteGateway](#) la referencia sobre la API ABAP del AWS SDK para SAP.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulte [Uso de este servicio con un AWS SDK](#). En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **DescribeAssetModel** con un AWS SDK o CLI

Los siguientes ejemplos de código muestran cómo utilizar `DescribeAssetModel`.

CLI

AWS CLI

Para describir un modelo de activos

En el siguiente ejemplo de `describe-asset-model`, se describe un modelo de activos de parque eólico.

```

aws iotsitewise describe-asset-model \
  --asset-model-id a1b2c3d4-5678-90ab-cdef-2222EXAMPLE

```

Salida:

```

{
  "assetModelId": "a1b2c3d4-5678-90ab-cdef-2222EXAMPLE",
  "assetModelArn": "arn:aws:iotsitewise:us-west-2:123456789012:asset-model/a1b2c3d4-5678-90ab-cdef-2222EXAMPLE",
  "assetModelName": "Wind Farm Model",

```

```

"assetModelDescription": "Represents a wind farm that comprises many wind
turbines",
"assetModelProperties": [
  {
    "id": "a1b2c3d4-5678-90ab-cdef-99999EXAMPLE",
    "name": "Total Generated Power",
    "dataType": "DOUBLE",
    "unit": "kW",
    "type": {
      "metric": {
        "expression": "sum(power)",
        "variables": [
          {
            "name": "power",
            "value": {
              "propertyId": "a1b2c3d4-5678-90ab-
cdef-66666EXAMPLE",
              "hierarchyId": "a1b2c3d4-5678-90ab-
cdef-77777EXAMPLE"
            }
          }
        ],
        "window": {
          "tumbling": {
            "interval": "1h"
          }
        }
      }
    }
  },
  {
    "id": "a1b2c3d4-5678-90ab-cdef-88888EXAMPLE",
    "name": "Region",
    "dataType": "STRING",
    "type": {
      "attribute": {
        "defaultValue": " "
      }
    }
  }
],
"assetModelHierarchies": [
  {
    "id": "a1b2c3d4-5678-90ab-cdef-77777EXAMPLE",

```

```

        "name": "Wind Turbines",
        "childAssetModelId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
    }
],
"assetModelCreationDate": 1575671284.0,
"assetModelLastUpdateDate": 1575671988.0,
"assetModelStatus": {
    "state": "ACTIVE"
}
}

```

Para obtener más información, consulte [Descripción de un modelo de activo específico](#) en la Guía del SiteWise usuario de AWS IoT.

- Para obtener más información sobre la API, consulte [DescribeAssetModel](#) la Referencia de AWS CLI comandos.

Java

SDK para Java 2.x

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```

/**
 * Retrieves the property IDs associated with a specific asset model.
 *
 * @param assetModelId the ID of the asset model that defines the properties.
 * @return a {@link CompletableFuture} that represents a {@link Map} result
 * that associates the property name to the
 *     property ID. The calling code can attach callbacks, then handle the
 * result or exception by calling
 *     {@link CompletableFuture#join()} or {@link
 * CompletableFuture#get()}.
 *     <p>
 *     If any completion stage in this method throws an exception, the
 * method logs the exception cause and keeps
 *     it available to the calling code as a {@link CompletionException}.
 * By calling

```

```

    *      {@link CompletionException#getCause()}, the calling code can
    access the original exception.
    */
    public CompletableFuture<Map<String, String>> getPropertyIds(String
assetModelId) {
        ListAssetModelPropertiesRequest modelPropertiesRequest =
ListAssetModelPropertiesRequest.builder().assetModelId(assetModelId).build();
        return getAsyncClient().listAssetModelProperties(modelPropertiesRequest)
            .handle((response, throwable) -> {
                if (response != null) {
                    return response.assetModelPropertySummaries().stream()
                        .collect(Collectors
                            .toMap(AssetModelPropertySummary::name,
AssetModelPropertySummary::id));
                } else {
                    logger.error("Error occurred while fetching property IDs:
{}.", throwable.getCause().getMessage());
                    throw (CompletionException) throwable;
                }
            });
    }
}

```

- Para obtener más información sobre la API, consulta [DescribeAssetModel](#) la Referencia AWS SDK for Java 2.x de la API.

JavaScript

SDK para JavaScript (v3)

Note

Hay más información. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```

import {
    DescribeAssetModelCommand,
    IoTSiteWiseClient,
} from "@aws-sdk/client-iotsitewise";
import { parseArgs } from "node:util";

```

```
/**
 * Describe an asset model.
 * @param {{ assetModelId : string }}
 */
export const main = async ({ assetModelId }) => {
  const client = new IoTSiteWiseClient({});
  try {
    const { assetModelDescription } = await client.send(
      new DescribeAssetModelCommand({
        assetModelId: assetModelId, // The ID of the Gateway to describe.
      }),
    );
    console.log("Asset model information retrieved successfully.");
    return { assetModelDescription: assetModelDescription };
  } catch (caught) {
    if (caught instanceof Error && caught.name === "ResourceNotFound") {
      console.warn(
        `${caught.message}. The asset model could not be found. Please check the
        asset model id.`
      );
    } else {
      throw caught;
    }
  }
};
```

- Para obtener más información sobre la API, consulta [DescribeAssetModel](#) la Referencia AWS SDK para JavaScript de la API.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulte [Uso de este servicio con un AWS SDK](#). En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **DescribeGateway** con un AWS SDK o CLI

Los siguientes ejemplos de código muestran cómo utilizar DescribeGateway.

CLI

AWS CLI

Para describir una puerta de enlace

En el siguiente ejemplo de `describe-gateway`, se describe una puerta de enlace.

```
aws iotsitewise describe-gateway \  
  --gateway-id a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE
```

Salida:


```
{  
  "gatewayId": "a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE",  
  "gatewayName": "ExampleCorpGateway",  
  "gatewayArn": "arn:aws:iotsitewise:us-west-2:123456789012:gateway/  
a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE",  
  "gatewayPlatform": {  
    "greengrass": {  
      "groupArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/  
groups/a1b2c3d4-5678-90ab-cdef-1b1b1EXAMPLE"  
    }  
  },  
  "gatewayCapabilitySummaries": [  
    {  
      "capabilityNamespace": "iotsitewise:opcuacollector:1",  
      "capabilitySyncStatus": "IN_SYNC"  
    }  
  ],  
  "creationDate": 1588369971.457,  
  "lastUpdateDate": 1588369971.457  
}
```

Para obtener más información, consulte [Ingesta de datos mediante una puerta de enlace](#) en la Guía del SiteWise usuario de AWS IoT.

- Para obtener más información sobre la API, consulte [DescribeGateway](#) la Referencia de AWS CLI comandos.

Java

SDK para Java 2.x

 Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
/**
 * Describes the specified gateway.
 *
 * @param gatewayId the ID of the gateway to describe.
 * @return a {@link CompletableFuture} that represents a {@link
DescribeGatewayResponse} result. The calling code
 *         can attach callbacks, then handle the result or exception by
calling {@link CompletableFuture#join()} or
 *         {@link CompletableFuture#get()}.
 *         <p>
 *         If any completion stage in this method throws an exception, the
method logs the exception cause and keeps
 *         it available to the calling code as a {@link CompletionException}.
By calling
 *         {@link CompletionException#getCause()}, the calling code can
access the original exception.
 */
public CompletableFuture<DescribeGatewayResponse> describeGatewayAsync(String
gatewayId) {
    DescribeGatewayRequest request = DescribeGatewayRequest.builder()
        .gatewayId(gatewayId)
        .build();

    return getAsyncClient().describeGateway(request)
        .whenComplete((response, exception) -> {
            if (exception != null) {
                logger.error("An error occurred during the describeGateway
method: {}", exception.getCause().getMessage());
            }
        });
}
```

- Para obtener más información sobre la API, consulta [DescribeGateway](#) la Referencia AWS SDK for Java 2.x de la API.

JavaScript

SDK para JavaScript (v3)

Note

Hay más información. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
import {
  DescribeGatewayCommand,
  IoTSiteWiseClient,
} from "@aws-sdk/client-iotsitewise";
import { parseArgs } from "node:util";

/**
 * Create an SSM document.
 * @param {{ content: string, name: string, documentType?: DocumentType }}
 */
export const main = async ({ gatewayId }) => {
  const client = new IoTSiteWiseClient({});
  try {
    const { gatewayDescription } = await client.send(
      new DescribeGatewayCommand({
        gatewayId: gatewayId, // The ID of the Gateway to describe.
      }),
    );
    console.log("Gateway information retrieved successfully.");
    return { gatewayDescription: gatewayDescription };
  } catch (caught) {
    if (caught instanceof Error && caught.name === "ResourceNotFound") {
      console.warn(
        `${caught.message}. The Gateway could not be found. Please check the Gateway Id.`
      );
    } else {
      throw caught;
    }
  }
};
```

```

    }
  }
};

```

- Para obtener más información sobre la API, consulta [DescribeGateway](#) la Referencia AWS SDK para JavaScript de la API.

Python

SDK para Python (Boto3)

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```

class IoTSitewiseWrapper:
    """Encapsulates AWS IoT SiteWise actions using the client interface."""

    def __init__(self, iotsitewise_client: client) -> None:
        """
        Initializes the IoTSitewiseWrapper with an AWS IoT SiteWise client.

        :param iotsitewise_client: A Boto3 AWS IoT SiteWise client. This client
        provides low-level
                                access to AWS IoT SiteWise services.
        """
        self.iotsitewise_client = iotsitewise_client
        self.entry_id = 0 # Incremented to generate unique entry IDs for
        batch_put_asset_property_value.

    @classmethod
    def from_client(cls) -> "IoTSitewiseWrapper":
        """
        Creates an IoTSitewiseWrapper instance with a default AWS IoT SiteWise
        client.

        :return: An instance of IoTSitewiseWrapper initialized with the default
        AWS IoT SiteWise client.

```

```
"""
    iotsitewise_client = boto3.client("iotsitewise")
    return cls(iotsitewise_client)

def describe_gateway(self, gateway_id: str) -> Dict[str, Any]:
    """
    Describes an AWS IoT SiteWise Gateway.

    :param gateway_id: The ID of the gateway to describe.
    :return: A dictionary containing information about the gateway.
    """
    try:
        response =
self.iotsitewise_client.describe_gateway(gatewayId=gateway_id)
        return response
    except ClientError as err:
        if err.response["Error"]["Code"] == "ResourceNotFoundException":
            logger.error("Gateway %s does not exist.", gateway_id)
        else:
            logger.error(
                "Error describing gateway %s. Here's why %s",
                gateway_id,
                err.response["Error"]["Message"],
            )
        raise
```

- Para obtener más información sobre la API, consulta [DescribeGateway](#) la AWS Referencia de API de SDK for Python (Boto3).

SAP ABAP

SDK para SAP ABAP

Note

Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```

TRY.
  oo_result = lo_ios->describegateway(
    iv_gatewayid = iv_gateway_id
  ). " oo_result is returned for testing purposes. "
  MESSAGE 'Retrieved gateway description.' TYPE 'I'.
CATCH /aws1/cx_iosresourcenotfoundex.
  MESSAGE 'Gateway does not exist.' TYPE 'E'.
ENDTRY.

```

- Para obtener más información sobre la API, consulte [DescribeGateway](#) la referencia sobre la API ABAP del AWS SDK para SAP.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulte [Uso de este servicio con un AWS SDK](#). En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **GetAssetPropertyValue** con un AWS SDK o CLI

Los siguientes ejemplos de código muestran cómo utilizar `GetAssetPropertyValue`.

CLI

AWS CLI

Para recuperar el valor actual de una propiedad de un activo

En el siguiente ejemplo de `get-asset-property-value`, se recupera la potencia total actual de un activo de turbina eólica.

```

aws iotsitewise get-asset-property-value \
  --asset-id a1b2c3d4-5678-90ab-cdef-3333EXAMPLE \
  --property-id a1b2c3d4-5678-90ab-cdef-6666EXAMPLE

```

Salida:

```

{
  "propertyValue": {
    "value": {
      "doubleValue": 6890.8677520453875
    },

```

```

        "timestamp": {
            "timeInSeconds": 1580853000,
            "offsetInNanos": 0
        },
        "quality": "GOOD"
    }
}

```

Para obtener más información, consulte [Consulta los valores actuales de las propiedades de los activos](#) en la Guía del SiteWise usuario de AWS IoT.

- Para obtener más información sobre la API, consulte [GetAssetPropertyValue](#) la Referencia de AWS CLI comandos.

Java

SDK para Java 2.x

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```

/**
 * Fetches the value of an asset property.
 *
 * @param propId the ID of the asset property to fetch.
 * @param assetId the ID of the asset to fetch the property value for.
 * @return a {@link CompletableFuture} that represents a {@link Double}
result. The calling code can attach
 *     callbacks, then handle the result or exception by calling {@link
CompletableFuture#join()} or
 *     {@link CompletableFuture#get()}.
 *     <p>
 *     If any completion stage in this method throws an exception, the
method logs the exception cause and keeps
 *     it available to the calling code as a {@link CompletionException}.
By calling
 *     {@link CompletionException#getCause()}, the calling code can
access the original exception.
 */

```

```

public CompletableFuture<Double> getAssetPropValueAsync(String propId, String
assetId) {
    GetAssetPropertyValueRequest assetPropertyValueRequest =
GetAssetPropertyValueRequest.builder()
        .propertyId(propId)
        .assetId(assetId)
        .build();

    return getAsyncClient().getAssetPropertyValue(assetPropertyValueRequest)
        .handle((response, exception) -> {
            if (exception != null) {
                logger.error("Error occurred while fetching property
value: {}.", exception.getCause().getMessage());
                throw (CompletionException) exception;
            }
            return response.propertyValue().value().doubleValue();
        });
}

```

- Para obtener más información sobre la API, consulta [GetAssetPropertyValuel](#) la Referencia AWS SDK for Java 2.x de la API.

JavaScript

SDK para JavaScript (v3)

Note

Hay más información. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```

import {
    GetAssetPropertyValueCommand,
    IoTSiteWiseClient,
} from "@aws-sdk/client-iotsitewise";
import { parseArgs } from "node:util";

/**
 * Describe an asset property value.

```

```

* @param {{ entryId : string }}
*/
export const main = async ({ entryId }) => {
  const client = new IoTSiteWiseClient({});
  try {
    const result = await client.send(
      new GetAssetPropertyValueCommand({
        entryId: entryId, // The ID of the Gateway to describe.
      }),
    );
    console.log("Asset property information retrieved successfully.");
    return result;
  } catch (caught) {
    if (caught instanceof Error && caught.name === "ResourceNotFound") {
      console.warn(
        `${caught.message}. The asset property entry could not be found. Please
        check the entry id.`
      );
    } else {
      throw caught;
    }
  }
};

```

- Para obtener más información sobre la API, consulta [GetAssetPropertyValue](#) la Referencia AWS SDK para JavaScript de la API.

Python

SDK para Python (Boto3)

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```

class IoTSiteWiseWrapper:
    """Encapsulates AWS IoT SiteWise actions using the client interface."""

```

```

def __init__(self, iotsitewise_client: client) -> None:
    """
    Initializes the IoTSitewiseWrapper with an AWS IoT SiteWise client.

    :param iotsitewise_client: A Boto3 AWS IoT SiteWise client. This client
    provides low-level
                                access to AWS IoT SiteWise services.
    """
    self.iotsitewise_client = iotsitewise_client
    self.entry_id = 0 # Incremented to generate unique entry IDs for
    batch_put_asset_property_value.

    @classmethod
    def from_client(cls) -> "IoTSitewiseWrapper":
        """
        Creates an IoTSitewiseWrapper instance with a default AWS IoT SiteWise
        client.

        :return: An instance of IoTSitewiseWrapper initialized with the default
        AWS IoT SiteWise client.
        """
        iotsitewise_client = boto3.client("iotsitewise")
        return cls(iotsitewise_client)

    def get_asset_property_value(
        self, asset_id: str, property_id: str
    ) -> Dict[str, Any]:
        """
        Gets the value of an AWS IoT SiteWise Asset Property.

        :param asset_id: The ID of the asset.
        :param property_id: The ID of the property.
        :return: A dictionary containing the value of the property.
        """
        try:
            response = self.iotsitewise_client.get_asset_property_value(
                assetId=asset_id, propertyId=property_id
            )
            return response["propertyValue"]
        except ClientError as err:
            if err.response["Error"]["Code"] == "ResourceNotFoundException":
                logger.error(

```

```

        "Asset %s or property %s does not exist.", asset_id,
property_id
    )
    else:
        logger.error(
            "Error getting asset property value. Here's why %s",
            err.response["Error"]["Message"],
        )
    raise

```

- Para obtener más información sobre la API, consulta [GetAssetPropertyValue](#) la AWS Referencia de API de SDK for Python (Boto3).

SAP ABAP

SDK para SAP ABAP

Note

Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```

TRY.
    oo_result = lo_ios->getassetpropertyvalue(
        iv_assetid = iv_asset_id
        iv_propertyid = iv_property_id
    ). " oo_result is returned for testing purposes. "
    MESSAGE 'Retrieved asset property value.' TYPE 'I'.
CATCH /aws1/cx_iosresourcenotfoundex.
    MESSAGE 'Asset or property does not exist.' TYPE 'E'.
ENDTRY.

```

- Para obtener más información sobre la API, consulte [GetAssetPropertyValue](#) la referencia sobre la API ABAP del AWS SDK para SAP.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulte [Uso de este servicio con un AWS SDK](#). En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

ListAssetModelProperties Úselo con un AWS SDK

En el siguiente ejemplo de código, se muestra cómo utilizar `ListAssetModelProperties`.

SAP ABAP

SDK para SAP ABAP

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
TRY.  
    oo_result = lo_ios->listassetmodelproperties(  
        iv_assetmodelid = iv_asset_model_id  
    ). " oo_result is returned for testing purposes. "  
    DATA(lt_properties) = oo_result->get_assetmodelpropertysums( ).  
    MESSAGE 'Retrieved list of asset model properties.' TYPE 'I'.  
CATCH /aws1/cx_rt_generic.  
    MESSAGE 'Unable to list asset model properties.' TYPE 'E'.  
ENDTRY.
```

- Para obtener más información sobre la API, consulte [ListAssetModelProperties](#) la referencia sobre la API ABAP del AWS SDK para SAP.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulte [Uso de este servicio con un AWS SDK](#). En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo `ListAssetModels` con un AWS SDK o CLI

Los siguientes ejemplos de código muestran cómo utilizar `ListAssetModels`.

CLI

AWS CLI

Para enumerar todos los modelos de activos

En el siguiente `list-asset-models` ejemplo, se enumeran todos los modelos de activos que están definidos en su AWS cuenta en la región actual.

```
aws iotsitewise list-asset-models
```

Salida:

```
{
  "assetModelSummaries": [
    {
      "id": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
      "arn": "arn:aws:iotsitewise:us-west-2:123456789012:asset-model/a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
      "name": "Wind Farm Model",
      "description": "Represents a wind farm that comprises many wind turbines",
      "creationDate": 1575671284.0,
      "lastUpdateDate": 1575671988.0,
      "status": {
        "state": "ACTIVE"
      }
    },
    {
      "id": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
      "arn": "arn:aws:iotsitewise:us-west-2:123456789012:asset-model/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
      "name": "Wind Turbine Model",
      "description": "Represents a wind turbine manufactured by Example Corp",
      "creationDate": 1575671207.0,
      "lastUpdateDate": 1575686273.0,
      "status": {
        "state": "ACTIVE"
      }
    }
  ]
}
```

Para obtener más información, consulte [Listar todos los modelos de activos](#) en la Guía del SiteWise usuario de AWS IoT.

- Para obtener más información sobre la API, consulte [ListAssetModels](#) la Referencia de AWS CLI comandos.

Java

SDK para Java 2.x

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
/**
 * Retrieves the asset model ID for the given asset model name.
 *
 * @param assetModelName the name of the asset model for the ID.
 * @return a {@link CompletableFuture} that represents a {@link String}
result of the asset model ID or null if the
 *       asset model cannot be found. The calling code can attach
callbacks, then handle the result or exception
 *       by calling {@link CompletableFuture#join()} or {@link
CompletableFuture#get()}.
 *       <p>
 *       If any completion stage in this method throws an exception, the
method logs the exception cause and keeps
 *       it available to the calling code as a {@link CompletionException}.
By calling
 *       {@link CompletionException#getCause()}, the calling code can
access the original exception.
 */
public CompletableFuture<String> getAssetModelIdAsync(String assetModelName)
{
    ListAssetModelsRequest listAssetModelsRequest =
ListAssetModelsRequest.builder().build();
    return getAsyncClient().listAssetModels(listAssetModelsRequest)
        .handle((listAssetModelsResponse, exception) -> {
            if (exception != null) {
```

```

        logger.error("Failed to retrieve Asset Model ID: {}",
exception.getCause().getMessage());
        throw (CompletionException) exception;
    }
    for (AssetModelSummary assetModelSummary :
listAssetModelsResponse.assetModelSummaries()) {
        if (assetModelSummary.name().equals(assetModelName)) {
            return assetModelSummary.id();
        }
    }
    return null;
});
}

```

- Para obtener más información sobre la API, consulta [ListAssetModels](#) la Referencia AWS SDK for Java 2.x de la API.

JavaScript

SDK para JavaScript (v3)

Note

Hay más información. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```

import {
    ListAssetModelsCommand,
    IoTSiteWiseClient,
} from "@aws-sdk/client-iotsitewise";
import { parseArgs } from "node:util";

/**
 * List asset models.
 * @param {{ assetModelTypes : array }}
 */
export const main = async ({ assetModelTypes = [] }) => {
    const client = new IoTSiteWiseClient({});
    try {

```

```

const result = await client.send(
  new ListAssetModelsCommand({
    assetModelTypes: assetModelTypes, // The model types to list
  }),
);
console.log("Asset model types retrieved successfully.");
return result;
} catch (caught) {
  if (caught instanceof Error && caught.name === "IoTSiteWiseError") {
    console.warn(
      `${caught.message}. There was a problem listing the asset model types.`
    );
  } else {
    throw caught;
  }
}
};

```

- Para obtener más información sobre la API, consulta [ListAssetModels](#) la Referencia AWS SDK para JavaScript de la API.

Python

SDK para Python (Boto3)

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```

class IoTSiteWiseWrapper:
    """Encapsulates AWS IoT SiteWise actions using the client interface."""

    def __init__(self, iotsitewise_client: client) -> None:
        """
        Initializes the IoTSiteWiseWrapper with an AWS IoT SiteWise client.

        :param iotsitewise_client: A Boto3 AWS IoT SiteWise client. This client
        provides low-level

```

```
        access to AWS IoT SiteWise services.

        """
        self.iotsitewise_client = iotsitewise_client
        self.entry_id = 0 # Incremented to generate unique entry IDs for
batch_put_asset_property_value.

    @classmethod
    def from_client(cls) -> "IoTSitewiseWrapper":
        """
        Creates an IoTSitewiseWrapper instance with a default AWS IoT SiteWise
client.

        :return: An instance of IoTSitewiseWrapper initialized with the default
AWS IoT SiteWise client.
        """
        iotsitewise_client = boto3.client("iotsitewise")
        return cls(iotsitewise_client)

    def list_asset_models(self) -> List[Dict[str, Any]]:
        """
        Lists all AWS IoT SiteWise Asset Models.

        :return: A list of dictionaries containing information about each asset
model.

        """
        try:
            asset_models = []
            paginator =
self.iotsitewise_client.get_paginator("list_asset_models")
            pages = paginator.paginate()
            for page in pages:
                asset_models.extend(page["assetModelSummaries"])
            return asset_models
        except ClientError as err:
            logger.error(
                "Error listing asset models. Here's why %s",
                err.response["Error"]["Message"],
            )
            raise
```

- Para obtener más información sobre la API, consulta [ListAssetModels](#) la AWS Referencia de API de SDK for Python (Boto3).

SAP ABAP

SDK para SAP ABAP

Note

Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
TRY.  
    oo_result = lo_ios->listassetmodels( ). " oo_result is returned for  
testing purposes. "  
    DATA(lt_asset_models) = oo_result->get_assetmodelsummaries( ).  
    MESSAGE 'Retrieved list of asset models.' TYPE 'I'.  
CATCH /aws1/cx_rt_generic.  
    MESSAGE 'Unable to list asset models.' TYPE 'E'.  
ENDTRY.
```

- Para obtener más información sobre la API, consulte [ListAssetModels](#) la referencia sobre la API ABAP del AWS SDK para SAP.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulte [Uso de este servicio con un AWS SDK](#). En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Seguridad en AWS IoT SiteWise

La seguridad en la nube AWS es la máxima prioridad. Como AWS cliente, usted se beneficia de una arquitectura de centro de datos y red diseñada para cumplir con los requisitos de las organizaciones más sensibles a la seguridad.

La seguridad es una responsabilidad compartida entre usted AWS y usted. El [modelo de responsabilidad compartida](#) la describe como seguridad de la nube y seguridad en la nube:

- Seguridad de la nube: AWS es responsable de proteger la infraestructura que ejecuta AWS los servicios en la AWS nube. AWS también le proporciona servicios que puede utilizar de forma segura. Los auditores externos prueban y verifican periódicamente la eficacia de nuestra seguridad como parte de los [AWS programas](#) de de . Para obtener más información sobre los programas de cumplimiento aplicables AWS IoT SiteWise, consulte los [AWS servicios incluidos en el ámbito de aplicación por programa de cumplimiento](#) y .
- Seguridad en la nube: su responsabilidad viene determinada por el AWS servicio que utilice. También es responsable de otros factores, incluida la confidencialidad de los datos, los requisitos de la empresa y la legislación y la normativa aplicables.

Esta documentación le ayuda a comprender cómo aplicar el modelo de responsabilidad compartida cuando se utiliza AWS IoT SiteWise. Los siguientes temas muestran cómo configurarlo AWS IoT SiteWise para cumplir sus objetivos de seguridad y conformidad. También aprenderá a utilizar otros AWS servicios que le ayudan a supervisar y proteger sus AWS IoT SiteWise recursos.

Temas

- [Protección de datos en AWS IoT SiteWise](#)
- [Cifrado de datos en AWS IoT SiteWise](#)
- [Gestión de identidades y accesos para AWS IoT SiteWise](#)
- [Validación de conformidad para AWS IoT SiteWise](#)
- [Resiliencia en AWS IoT SiteWise](#)
- [Seguridad de la infraestructura en AWS IoT SiteWise](#)
- [Análisis de configuración y vulnerabilidad en AWS IoT SiteWise](#)
- [Puntos finales de VPC para AWS IoT SiteWise](#)
- [Mejores prácticas de seguridad para AWS IoT SiteWise](#)

Protección de datos en AWS IoT SiteWise

El modelo de [responsabilidad AWS compartida modelo](#) se aplica a la protección de datos en AWS IoT SiteWise. Como se describe en este modelo, AWS es responsable de proteger la infraestructura global que ejecuta todos los Nube de AWS. Eres responsable de mantener el control sobre el contenido alojado en esta infraestructura. También eres responsable de las tareas de administración y configuración de seguridad para los Servicios de AWS que utiliza. Para obtener más información sobre la privacidad de los datos, consulte las [Preguntas frecuentes sobre la privacidad de datos](#). Para obtener información sobre la protección de datos en Europa, consulte la publicación de blog sobre el [Modelo de responsabilidad compartida de AWS y GDPR](#) en el Blog de seguridad de AWS .

Con fines de protección de datos, le recomendamos que proteja Cuenta de AWS las credenciales y configure los usuarios individuales con AWS IAM Identity Center o AWS Identity and Access Management (IAM). De esta manera, solo se otorgan a cada usuario los permisos necesarios para cumplir sus obligaciones laborales. También recomendamos proteger sus datos de la siguiente manera:

- Utiliza la autenticación multifactor (MFA) en cada cuenta.
- Se utiliza SSL/TLS para comunicarse con AWS los recursos. Exigimos TLS 1.2 y recomendamos TLS 1.3.
- Configure la API y el registro de actividad de los usuarios con AWS CloudTrail. Para obtener información sobre el uso de CloudTrail senderos para capturar AWS actividades, consulte [Cómo trabajar con CloudTrail senderos](#) en la Guía del AWS CloudTrail usuario.
- Utilice soluciones de AWS cifrado, junto con todos los controles de seguridad predeterminados Servicios de AWS.
- Utiliza servicios de seguridad administrados avanzados, como Amazon Macie, que lo ayuden a detectar y proteger la información confidencial almacenada en Amazon S3.
- Si necesita módulos criptográficos validados por FIPS 140-3 para acceder a AWS través de una interfaz de línea de comandos o una API, utilice un punto final FIPS. Para obtener más información sobre los puntos de conexión de FIPS disponibles, consulte [Estándar de procesamiento de la información federal \(FIPS\) 140-3](#).

Se recomienda encarecidamente no introducir nunca información confidencial o sensible, como por ejemplo, direcciones de correo electrónico de clientes, en etiquetas o campos de formato libre, tales como el campo Nombre. Esto incluye cuando trabaja AWS IoT SiteWise o Servicios de AWS utiliza la consola, la API o. AWS CLI AWS SDKs Cualquier dato que introduzca en etiquetas o campos

de formato libre utilizados para los nombres se pueden emplear para los registros de facturación o diagnóstico. Si proporciona una URL a un servidor externo, recomendamos encarecidamente que no incluya información de credenciales en la URL a fin de validar la solicitud para ese servidor.

Temas

- [Privacidad del tráfico entre redes para AWS IoT SiteWise](#)
- [AWS IoT SiteWise Asistente de mejora del servicio empresarial](#)

Privacidad del tráfico entre redes para AWS IoT SiteWise

Las conexiones entre aplicaciones locales AWS IoT SiteWise y entre ellas, como las puertas de enlace SiteWise Edge, se protegen mediante conexiones de seguridad de capa de transporte (TLS). Para obtener más información, consulte [Cifrado de datos en tránsito para AWS IoT SiteWise](#).

AWS IoT SiteWise no admite conexiones entre zonas de disponibilidad dentro de una AWS región ni conexiones entre cuentas. AWS

Solo puede configurar el Centro de identidades de IAM en una región a la vez. SiteWise El monitor se conecta a la región que configuró para el IAM Identity Center. Esto significa que utiliza una región para el acceso al Centro de identidades de IAM, pero puede crear portales en cualquier región.

AWS IoT SiteWise Asistente de mejora del servicio empresarial

AWS IoT SiteWise Assistant no utiliza los datos de los clientes para mejorar el servicio ni para mejorar los subyacentes LLMs.

Cifrado de datos en AWS IoT SiteWise

El cifrado de datos se refiere a la protección de los datos mientras están en tránsito (cuando viajan hacia y desde AWS IoT SiteWise las puertas de enlace SiteWise Edge y los servidores y entre ellos) y en reposo (mientras están almacenados en dispositivos o AWS servicios locales). Puede proteger los datos en tránsito mediante seguridad de la capa de transporte (TLS) o en reposo mediante el cifrado del cliente.

Note

AWS IoT SiteWise El procesamiento perimetral expone APIs lo que está alojado en las puertas de enlace SiteWise Edge y a los que se puede acceder a través de la red local.

APIs Se exponen a través de una conexión TLS respaldada por un certificado de servidor propiedad del conector Edge. AWS IoT SiteWise Para la autenticación del cliente, APIs utilizan una contraseña de control de acceso. Tanto la clave privada del certificado del servidor como la contraseña de control de acceso se almacenan en el disco. AWS IoT SiteWise El procesamiento perimetral se basa en el cifrado del sistema de archivos para garantizar la seguridad de estas credenciales inactivas.

Para obtener más información sobre el cifrado del lado del servidor y el cifrado del cliente, revise los temas que se enumeran a continuación.

Temas

- [El cifrado en reposo en AWS IoT SiteWise](#)
- [Cifrado de datos en tránsito para AWS IoT SiteWise](#)
- [Gestión de claves en AWS IoT SiteWise](#)

El cifrado en reposo en AWS IoT SiteWise

AWS IoT SiteWise almacena sus datos en la AWS nube y en las pasarelas AWS IoT SiteWise Edge.

Datos en reposo en la nube AWS

AWS IoT SiteWise almacena los datos en otros AWS servicios que cifran los datos en reposo de forma predeterminada. Encryption at rest se integra con AWS Key Management Service (AWS KMS) para administrar la clave de cifrado que se utiliza para cifrar los valores de las propiedades de los activos y los valores agregados. AWS IoT SiteWise Puede optar por utilizar una clave administrada por el cliente para cifrar los valores de propiedad de activo y los valores agregados en AWS IoT SiteWise. Puede crear, administrar y ver la clave de cifrado mediante AWS KMS.

Puede elegir una clave Clave propiedad de AWS para cifrar sus datos o elegir una clave gestionada por el cliente para cifrar los valores de las propiedades de sus activos y los valores agregados:

Funcionamiento

El cifrado en reposo se integra AWS KMS para administrar la clave de cifrado que se utiliza para cifrar los datos.

- Clave propiedad de AWS — Clave de cifrado predeterminada. AWS IoT SiteWise es el propietario de esta clave. No puedes ver esta clave en tu AWS cuenta. Tampoco puede ver las operaciones de la clave en los registros de AWS CloudTrail . Puede usar esta clave sin cargo adicional.
- Clave administrada por el cliente: la clave se almacena en la cuenta y usted la crea, posee y administra. Usted controla plenamente la clave KMS. Se aplican AWS KMS cargos adicionales.

Claves propiedad de AWS

Claves propiedad de AWS no están guardados en tu cuenta. Forman parte de una colección de claves de KMS que AWS posee y administra para su uso en varias AWS cuentas. AWS los servicios que puede utilizar Claves propiedad de AWS para proteger sus datos.

No puede ver, administrar Claves propiedad de AWS, usar ni auditar su uso. Sin embargo, no es necesario que realice ninguna acción ni que cambie programas para proteger las claves que cifran sus datos.

No se te cobra una cuota mensual ni una cuota de uso si las utilizas Claves propiedad de AWS, y estas no se tienen en cuenta para AWS KMS las cuotas de tu cuenta.

Claves administradas por el cliente

Las claves administradas por el cliente son claves KMS en su cuenta que usted ha creado, posee y administra. Tiene el control total sobre estas claves KMS, lo que significa que puede hacer lo siguiente:

- Establecer y mantener sus políticas de claves, políticas de IAM y concesiones.
- Activarlas y desactivarlas.
- Rotar sus materiales criptográficos.
- Agregar etiquetas.
- Crear alias que hagan referencia a ellas.
- Programar su eliminación.

También puedes usar CloudTrail Amazon CloudWatch Logs para realizar un seguimiento de las solicitudes que se AWS IoT SiteWise envían AWS KMS en tu nombre.

Si utilizas claves administradas por el cliente, debes conceder AWS IoT SiteWise acceso a la clave de KMS almacenada en tu cuenta. AWS IoT SiteWise utiliza el cifrado de sobres y la jerarquía de

claves para cifrar los datos. La clave de AWS KMS cifrado se utiliza para cifrar la clave raíz de esta jerarquía de claves. Para obtener más información, consulte [Cifrado de sobre](#) en la Guía para desarrolladores de AWS Key Management Service .

El siguiente ejemplo de política concede AWS IoT SiteWise permisos para crear una clave gestionada por el cliente en su nombre. Al crear la clave, debe permitir las acciones `kms:CreateGrant` y `kms:DescribeKey`.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1603902045292",
      "Action": [
        "kms:CreateGrant",
        "kms:DescribeKey"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

El contexto de cifrado para su concesión creada utiliza su `aws:iotsitewise:subscriberId` y su ID de cuenta.

Datos en reposo en las puertas de enlace SiteWise Edge

AWS IoT SiteWise las pasarelas almacenan los siguientes datos en el sistema de archivos local:

- Información de configuración de origen OPC UA
- Conjunto de rutas de flujo de datos OPC UA desde orígenes OPC UA conectados
- Los datos industriales se almacenan en caché cuando la puerta de enlace SiteWise Edge pierde la conexión a Internet

SiteWise Las puertas de enlace Edge se ejecutan en funcionamiento. AWS IoT Greengrass AWS IoT Greengrass se basa en los permisos de archivos de Unix y en el cifrado de disco completo (si

está activado) para proteger los datos almacenados en el núcleo. Es su responsabilidad proteger el sistema de archivos y el dispositivo.

Sin AWS IoT Greengrass embargo, cifra las copias locales de los secretos de su servidor OPC UA recuperados de Secrets Manager. Para obtener más información, consulte [Cifrado de secretos](#) en la Guía para desarrolladores de AWS IoT Greengrass Version 1 .

Para obtener más información sobre el cifrado en reposo en AWS IoT Greengrass núcleos, consulte [Cifrado en reposo en](#) la Guía para AWS IoT Greengrass Version 1 desarrolladores.

Cifrado de datos en tránsito para AWS IoT SiteWise

AWS IoT SiteWise utiliza el cifrado en tránsito para proteger los datos transmitidos entre sus dispositivos, pasarelas y la AWS nube. La comunicación con AWS IoT SiteWise usted se cifra mediante HTTPS y TLS 1.2, lo que garantiza que sus datos permanezcan confidenciales y protegidos contra el acceso o la interceptación no autorizados.

Hay tres modos de comunicación en los que los datos están en tránsito:

- [A través de Internet:](#) la comunicación entre dispositivos locales (incluidas las puertas de enlace SiteWise Edge) está cifrada AWS IoT SiteWise .
- [A través de la red local:](#) la comunicación entre las puertas de OpsHub enlace de SiteWise la aplicación y las de SiteWise Edge siempre está cifrada. La comunicación entre la aplicación de SiteWise monitorización que se ejecuta en el navegador y las pasarelas de SiteWise Edge siempre está cifrada. La comunicación entre las puertas de enlace SiteWise Edge y las fuentes OPC UA se puede cifrar.
- [Entre los componentes de las puertas de enlace SiteWise Edge:](#) la comunicación entre AWS IoT Greengrass los componentes de las puertas de enlace SiteWise Edge no está cifrada.

Temas

- [Datos en tránsito a través de Internet](#)
- [Datos en tránsito a través de la red local](#)
- [Datos en tránsito entre los componentes locales de SiteWise Edge](#)

Datos en tránsito a través de Internet

AWS IoT SiteWise utiliza Transport Layer Security (TLS) para cifrar todas las comunicaciones a través de Internet. Todos los datos que se envían a la AWS nube se envían a través de una conexión TLS mediante los protocolos MQTT o HTTPS, por lo que son seguros de forma predeterminada. SiteWise Las pasarelas perimetrales, que se ejecutan en ellas AWS IoT Greengrass, y las notificaciones del valor de las propiedades utilizan el modelo de seguridad del AWS IoT transporte. Para obtener más información, consulte [Seguridad de transporte](#) en la Guía del desarrollador de AWS IoT .

Datos en tránsito a través de la red local

SiteWise Las pasarelas perimetrales siguen las especificaciones del OPC UA para la comunicación con las fuentes OPC UA locales. Es su responsabilidad configurar los orígenes para utilizar un modo de seguridad de mensajes que cifre los datos en tránsito.

Si elige un modo de seguridad de mensajes de firma, los datos en tránsito entre las puertas de enlace y las fuentes de SiteWise Edge se firman pero no se cifran. Si elige un modo de seguridad para firmar y cifrar los mensajes, los datos en tránsito entre las puertas de enlace y las fuentes de SiteWise Edge se firman y cifran. Para obtener más información sobre la configuración de orígenes, consulte [Agregue fuentes de datos a su puerta de enlace AWS IoT SiteWise Edge](#).

La comunicación entre la aplicación de consola perimetral y las puertas de enlace SiteWise Edge siempre se cifra mediante TLS. El conector SiteWise Edge de la puerta de enlace SiteWise Edge genera y almacena un certificado autofirmado para poder establecer una conexión TLS con la consola Edge para la aplicación. AWS IoT SiteWise Deberá copiar este certificado de la puerta de enlace SiteWise Edge a la consola de Edge para su AWS IoT SiteWise aplicación antes de conectar la aplicación a la puerta de enlace SiteWise Edge. Esto garantiza que la consola perimetral de AWS IoT SiteWise la aplicación pueda comprobar que se ha conectado a la puerta de enlace SiteWise Edge de confianza.

Además del TLS para garantizar el secreto y la autenticidad del servidor, SiteWise Edge utiliza el protocolo SiGv4 para establecer la autenticidad de la aplicación de la consola perimetral. El conector SiteWise Edge de la puerta de enlace SiteWise Edge acepta y almacena una contraseña para poder verificar las conexiones entrantes desde la aplicación de consola perimetral, la aplicación de SiteWise supervisión que se ejecuta en los navegadores y otros clientes según el AWS IoT SiteWise SDK.

Para obtener más información sobre generación de la contraseña y el certificado del servidor, consulte [the section called “Administración de puertas de enlace”](#).

Datos en tránsito entre los componentes locales de SiteWise Edge

SiteWise Las pasarelas Edge funcionan AWS IoT Greengrass, lo que no cifra los datos intercambiados localmente en el AWS IoT Greengrass núcleo porque los datos no salen del dispositivo. Esto incluye la comunicación entre AWS IoT Greengrass componentes, como el AWS IoT SiteWise conector. Para obtener más información, consulte [Datos en el dispositivo central](#) en la Guía para desarrolladores de AWS IoT Greengrass Version 1 .

Gestión de claves en AWS IoT SiteWise

AWS IoT SiteWise administración de claves en la nube

De forma predeterminada, Claves administradas por AWS se AWS IoT SiteWise utiliza para proteger sus datos en la AWS nube. Puede actualizar sus ajustes para utilizar una clave administrada por el cliente para cifrar algunos datos en AWS IoT SiteWise. Puede crear, administrar y ver la clave de cifrado mediante AWS Key Management Service (AWS KMS).

AWS IoT SiteWise admite el cifrado del lado del servidor con claves administradas por el cliente almacenadas AWS KMS para cifrar los siguientes datos:

- Valores de propiedades de activos
- Valores agregados

Note

El resto de los datos y recursos se cifran mediante el cifrado predeterminado con claves gestionadas por. AWS IoT SiteWise Esta clave se almacena en la cuenta de AWS IoT SiteWise .

Para obtener más información, consulte [¿Qué es AWS Key Management Service?](#) en la Guía para AWS Key Management Service desarrolladores.

Habilitación del cifrado mediante claves administradas por el cliente

Para usar las claves administradas por el cliente AWS IoT SiteWise, debes actualizar AWS IoT SiteWise la configuración.

Para habilitar el cifrado mediante claves KMS

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. Elija Ajustes de la cuenta y, a continuación, Editar para abrir la página Editar ajustes de la cuenta.
3. En Tipo de clave de cifrado, elija Elegir una clave AWS KMS diferente. Esto habilita el cifrado con claves administradas por el cliente almacenadas en AWS KMS.

Note

En la actualidad, solo puede utilizar el cifrado con claves administradas por el cliente para valores de propiedad de activo y valores agregados.

4. Elija su clave KMS con una de las siguientes opciones:
 - Para usar una clave KMS existente: elija el alias de su clave KMS de la lista.
 - Para crear una clave KMS nueva, selecciona Crear una AWS KMS clave.

Note

Esto abre el panel de AWS KMS . Para obtener más información sobre cómo crear una clave KMS, consulte [Creación de claves](#) en la Guía para desarrolladores de AWS Key Management Service .

5. Elija Guardar para actualizar la configuración.

SiteWise Administración de claves de Edge Gateway

SiteWise Las puertas de enlace AWS IoT Greengrass perimetrales funcionan y los dispositivos AWS IoT Greengrass principales utilizan claves públicas y privadas para autenticarse en la AWS nube y cifrar los secretos locales, como los secretos de autenticación OPC UA. Para obtener más información, consulte [Administración de claves](#) en la Guía para desarrolladores de AWS IoT Greengrass Version 1 .

Gestión de identidades y accesos para AWS IoT SiteWise

AWS Identity and Access Management (IAM) es una herramienta Servicio de AWS que ayuda al administrador a controlar de forma segura el acceso a los AWS recursos. Los administradores de IAM controlan quién puede autenticarse (iniciar sesión) y quién puede autorizarse (tener permisos) para usar los recursos. AWS IoT SiteWise La IAM es una Servicio de AWS opción que puede utilizar sin coste adicional.

Temas

- [Audiencia por motivos de seguridad AWS IoT SiteWise](#)
- [Autentícate con identidades en AWS IoT SiteWise](#)
- [¿Cómo AWS IoT SiteWise funciona con IAM](#)
- [AWS políticas gestionadas para AWS IoT SiteWise](#)
- [Utilice funciones vinculadas al servicio para AWS IoT SiteWise](#)
- [Configure los permisos para las alarmas de eventos en AWS IoT SiteWise](#)
- [La prevención policial confusa entre servicios en AWS IoT SiteWise](#)
- [Solucione los problemas de AWS IoT SiteWise identidad y acceso](#)

Audiencia por motivos de seguridad AWS IoT SiteWise

La forma de usar AWS Identity and Access Management (IAM) varía según la función que desempeñes:

- Usuario del servicio: solicite permisos al administrador si no puede acceder a las características (consulte [Solucione los problemas de AWS IoT SiteWise identidad y acceso](#)).
- Administrador del servicio: determine el acceso de los usuarios y envíe las solicitudes de permiso (consulte [¿Cómo AWS IoT SiteWise funciona con IAM](#)).
- Administrador de IAM: escribe las políticas para administrar el acceso (consulte [AWS IoT SiteWise ejemplos de políticas basadas en la identidad](#)).

Autentícate con identidades en AWS IoT SiteWise

La autenticación es la forma en que inicias sesión AWS con tus credenciales de identidad. Debe autenticarse como usuario de Usuario raíz de la cuenta de AWS IAM o asumir una función de IAM.

Puede iniciar sesión como una identidad federada con las credenciales de una fuente de identidad, como AWS IAM Identity Center (IAM Identity Center), la autenticación de inicio de sesión único o las credenciales. Google/Facebook Para obtener más información sobre el inicio de sesión, consulte [Cómo iniciar sesión en la Cuenta de AWS](#) en la Guía del usuario de AWS Sign-In .

Para el acceso programático, AWS proporciona un SDK y una CLI para firmar criptográficamente las solicitudes. Para obtener más información, consulte [AWS Signature Version 4 para solicitudes de API](#) en la Guía del usuario de IAM.

Cuenta de AWS usuario root

Al crear un Cuenta de AWS, se comienza con una identidad de inicio de sesión denominada usuario Cuenta de AWS raíz que tiene acceso completo a todos Servicios de AWS los recursos. Se recomienda encarecidamente que no utilice el usuario raíz para las tareas diarias. Para ver la lista completa de las tareas que requieren credenciales de usuario raíz, consulte [Tareas que requieren credenciales de usuario raíz](#) en la Guía del usuario de IAM.

Usuarios y grupos de IAM

Un [usuario de IAM](#) es una identidad con permisos específicos para una sola persona o aplicación. Recomendamos el uso de credenciales temporales en lugar de usuarios de IAM con credenciales de larga duración. Para obtener más información, consulte [Exigir a los usuarios humanos que utilicen la federación con un proveedor de identidad para acceder AWS mediante credenciales temporales](#) en la Guía del usuario de IAM.

Un [grupo de IAM](#) especifica un conjunto de usuarios de IAM y facilita la administración de los permisos para grupos grandes de usuarios. Para obtener más información, consulte [Casos de uso para usuarios de IAM](#) en la Guía del usuario de IAM.

Roles de IAM

Un [Rol de IAM](#) es una identidad con permisos específicos que proporciona credenciales temporales. Puede asumir un rol [cambiando de un rol de usuario a uno de IAM \(consola\)](#) o llamando a una AWS CLI operación de AWS API. Para obtener más información, consulte [Métodos para asumir un rol](#) en la Guía del usuario de IAM.

Los roles de IAM son útiles para el acceso de usuario federado, los permisos de usuario de IAM temporales, el acceso entre cuentas, el acceso entre servicios y las aplicaciones que se ejecutan en Amazon EC2. Para obtener más información, consulte [Acceso a recursos entre cuentas en IAM](#) en la Guía del usuario de IAM.

¿Cómo AWS IoT SiteWise funciona con IAM

Antes de usar AWS Identity and Access Management (IAM) para administrar el acceso AWS IoT SiteWise, debe comprender con qué funciones de IAM está disponible. AWS IoT SiteWise

Característica de IAM	¿Compatible con AWS IoT SiteWise?
Políticas basadas en identidad con permisos de nivel de recursos	Sí
Acciones de políticas	Sí
Recursos de políticas	Sí
Claves de condición de política	Sí
Políticas basadas en recursos	No
Listas de control de acceso (ACLs)	No
Autorización basada en etiquetas (ABAC)	Sí
Credenciales temporales	Sí
Sesiones de acceso directo (FAS)	Sí
Roles vinculados al servicio	Sí
Roles de servicio	Sí

Para obtener una visión general de cómo AWS IoT SiteWise funcionan otros AWS servicios con IAM, consulte los [AWS servicios que funcionan con IAM](#) en la Guía del usuario de IAM.

Contenido

- [AWS IoT SiteWise Funciones de IAM](#)

- [Utilice credenciales temporales con AWS IoT SiteWise](#)
- [Sesiones de acceso directo \(FAS\) para AWS IoT SiteWise](#)
- [Roles vinculados a servicios](#)
- [Roles de servicio](#)
- [Elija un rol de IAM en AWS IoT SiteWise](#)
- [Autorización basada en AWS IoT SiteWise etiquetas](#)
- [AWS IoT SiteWise políticas basadas en la identidad](#)
 - [Acciones de políticas](#)
 - [BatchPutAssetPropertyValue autorización](#)
 - [Recursos de políticas](#)
 - [Claves de condición de políticas](#)
 - [Ejemplos](#)
- [AWS IoT SiteWise ejemplos de políticas basadas en la identidad](#)
 - [Prácticas recomendadas relativas a políticas](#)
 - [Utilice la consola AWS IoT SiteWise](#)
 - [Cómo permitir a los usuarios consultar sus propios permisos](#)
 - [Concesión de permisos para que los usuarios ingieran datos en activos de una jerarquía](#)
 - [Vea AWS IoT SiteWise los activos en función de las etiquetas](#)
- [Administre el acceso mediante políticas en AWS IoT SiteWise](#)
 - [Políticas basadas en identidades](#)
 - [Políticas basadas en recursos](#)
 - [Listas de control de acceso \(\) ACLs](#)
 - [Otros tipos de políticas](#)
 - [Varios tipos de políticas](#)

AWS IoT SiteWise Funciones de IAM

Un [rol de IAM](#) es una entidad de la cuenta de AWS que dispone de permisos específicos.

Utilice credenciales temporales con AWS IoT SiteWise

Puede utilizar credenciales temporales para iniciar sesión con federación, asumir un rol de IAM o asumir un rol de acceso entre cuentas. Las credenciales de seguridad temporales se

obtienen mediante una llamada a operaciones de la API de AWS STS , como [AssumeRole](#) o [GetFederationToken](#).

AWS IoT SiteWise admite el uso de credenciales temporales.

SiteWise Monitor permite a los usuarios federados acceder a los portales. Los usuarios del portal se autentican con sus credenciales del Centro de identidades de IAM o IAM.

Important

Los usuarios o roles deben tener el permiso de `iotsitewise:DescribePortal` para iniciar sesión en el portal.

Cuando un usuario inicia sesión en un portal, SiteWise Monitor genera una política de sesión que proporciona los siguientes permisos:

- Acceso de solo lectura a los activos y a los datos de los activos de AWS IoT SiteWise su cuenta a los que proporciona acceso la función de ese portal.
- Acceso a proyectos en ese portal para los que el usuario tiene acceso de administrador (propietario del proyecto) o de solo lectura (lector del proyecto).

Para obtener más información acerca de los permisos de usuario federado del portal, consulte [Utilice los roles de servicio para AWS IoT SiteWise Monitor](#).

Sesiones de acceso directo (FAS) para AWS IoT SiteWise

Admite sesiones de acceso directo (FAS): sí

Las sesiones de acceso directo (FAS) utilizan los permisos de la persona principal que realiza la llamada Servicio de AWS, junto con la solicitud, Servicio de AWS para realizar solicitudes a los servicios descendentes. Para obtener información sobre las políticas a la hora de realizar solicitudes de FAS, consulte [Sesiones de acceso directo](#).

Roles vinculados a servicios

Las [funciones vinculadas al servicio](#) permiten a AWS los servicios acceder a los recursos de otros servicios para completar una acción en tu nombre. Las funciones vinculadas al servicio aparecen en tu AWS cuenta y son propiedad del servicio. Un administrador de IAM puede ver, pero no editar, los permisos de los roles vinculados a servicios.

AWS IoT SiteWise admite funciones vinculadas al servicio. Para obtener más información acerca de cómo crear o administrar roles vinculados a servicios de AWS IoT SiteWise, consulte [Utilice funciones vinculadas al servicio para AWS IoT SiteWise](#).

Roles de servicio

Esta característica permite que un servicio asuma un [rol de servicio](#) en su nombre. Este rol permite que el servicio obtenga acceso a los recursos de otros servicios para completar una acción en su nombre. Los roles de servicio aparecen en su AWS cuenta y son propiedad de la cuenta. Esto significa que un administrador de IAM puede cambiar los permisos de este rol. Sin embargo, hacerlo podría deteriorar la funcionalidad del servicio.

AWS IoT SiteWise utiliza un rol de servicio para permitir a los usuarios del portal SiteWise Monitor acceder a algunos de sus AWS IoT SiteWise recursos en su nombre. Para obtener más información, consulte [Utilice los roles de servicio para AWS IoT SiteWise Monitor](#).

Debe disponer de los permisos necesarios para poder crear modelos de AWS IoT Events alarma en el AWS IoT SiteWise. Para obtener más información, consulte [Configure los permisos para las alarmas de eventos en AWS IoT SiteWise](#).

Elija un rol de IAM en AWS IoT SiteWise

Al crear un portal recurso en AWS IoT SiteWise, debe elegir un rol que permita a los usuarios federados de su portal SiteWise Monitor acceder AWS IoT SiteWise en su nombre. Si ya ha creado un rol de servicio, le AWS IoT SiteWise proporciona una lista de roles entre los que puede elegir. De lo contrario, puede crear un rol con los permisos necesarios al crear un portal. Es importante elegir un rol que permita acceder a los activos y a los datos de activos. Para obtener más información, consulte [Utilice los roles de servicio para AWS IoT SiteWise Monitor](#).

Autorización basada en AWS IoT SiteWise etiquetas

Puedes adjuntar etiquetas a AWS IoT SiteWise los recursos o pasarles etiquetas en una solicitud AWS IoT SiteWise. Para controlar el acceso en función de etiquetas, debe proporcionar información de las etiquetas en el [elemento de condición](#) de una política utilizando las claves de condición `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` o `aws:TagKeys`. Para obtener más información acerca del etiquetado de recursos de AWS IoT SiteWise, consulte [Etiquete sus AWS IoT SiteWise recursos](#).

Para consultar un ejemplo de política basada en la identidad para limitar el acceso a un recurso en función de las etiquetas de ese recurso, consulte [Vea AWS IoT SiteWise los activos en función de las etiquetas](#).

AWS IoT SiteWise políticas basadas en la identidad

Las políticas de IAM le permiten controlar quién puede hacer qué en AWS IoT SiteWise. Puede decidir qué acciones están permitidas o no y establecer condiciones específicas para estas acciones. Por ejemplo, puede establecer reglas sobre quién puede ver o cambiar la información AWS IoT SiteWise. AWS IoT SiteWise admite acciones, recursos y claves de condición específicos. Para obtener más información acerca de los elementos que utiliza en una política de JSON, consulte [Referencia de los elementos de las políticas de JSON de IAM](#) en la Guía del usuario de IAM.

Acciones de políticas

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Action` de una política JSON describe las acciones que puede utilizar para conceder o denegar el acceso en una política. Incluya acciones en una política para conceder permisos y así llevar a cabo la operación asociada.

Las acciones políticas AWS IoT SiteWise utilizan el siguiente prefijo antes de la acción: `iotsitewise:`. Por ejemplo, para conceder a alguien permiso para cargar datos de propiedades de activos AWS IoT SiteWise con la operación de la `BatchPutAssetPropertyValue` API, debes incluir la `iotsitewise:BatchPutAssetPropertyValue` acción en su política. Las declaraciones de política deben incluir un `NotAction` elemento `Action` o. AWS IoT SiteWise define su propio conjunto de acciones que describen las tareas que puede realizar con este servicio.

Para especificar varias acciones de en una única instrucción, sepárelas con comas del siguiente modo.

```
"Action": [
  "iotsitewise:action1",
  "iotsitewise:action2"
]
```

Puede utilizar caracteres comodín (*) para especificar varias acciones . Por ejemplo, para especificar todas las acciones que comiencen con la palabra `Describe`, incluya la siguiente acción.

```
"Action": "iotsitewise:Describe*"
```

Para ver una lista de AWS IoT SiteWise acciones, consulte [las acciones definidas AWS IoT SiteWise en la](#) Guía del usuario de IAM.

BatchPutAssetPropertyValue autorización

AWS IoT SiteWise autoriza el acceso a la [BatchPutAssetPropertyValue](#) acción de una manera inusual. En la mayoría de las acciones, al permitir o denegar acceso, dicha acción devuelve un error si no se conceden permisos. Con BatchPutAssetPropertyValue puede enviar varias entradas de datos a diferentes activos y propiedades de activos en una sola solicitud de la API. AWS IoT SiteWise autoriza cada entrada de datos de forma independiente. Para cualquier entrada individual que no supere la autorización de la solicitud, AWS IoT SiteWise incluye un error `AccessDeniedException` en la lista de errores devuelta. AWS IoT SiteWise recibe los datos de cualquier entrada que autorice y sea correcta, incluso si otra entrada de la misma solicitud no es válida.

Important

Antes de ingerir datos a un flujo de datos, realice lo siguiente:

- Autorice el recurso `time-series` si utiliza un alias de propiedad para identificar el flujo de datos.
- Autorice el recurso `asset` si utiliza un ID de activo para identificar el activo que contiene la propiedad de activo asociada.

Recursos de políticas

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Resource` de la política JSON especifica el objeto u objetos a los que se aplica la acción. Como práctica recomendada, especifique un recurso utilizando el [Nombre de recurso de Amazon \(ARN\)](#). En el caso de las acciones que no admiten permisos por recurso, utilice un carácter comodín (*) para indicar que la instrucción se aplica a todos los recursos.

```
"Resource": "*"
```

Cada declaración de política de IAM se aplica a los recursos que especifique mediante su ARNs uso. Un ARN tiene la siguiente sintaxis general.

```
arn:${Partition}:${Service}:${Region}:${Account}:${ResourceType}/${ResourcePath}
```

Para obtener más información sobre el formato de ARNs, consulte [Identificar AWS los recursos con los nombres de los recursos de Amazon \(ARNs\)](#).

Por ejemplo, para especificar el activo con el ID a1b2c3d4-5678-90ab-cdef-2222EXAMPLE en la instrucción, utilice el siguiente ARN.

```
"Resource": "arn:aws:iotsitewise:region:123456789012:asset/a1b2c3d4-5678-90ab-cdef-2222EXAMPLE"
```

Para especificar todos los flujos de datos que pertenecen a una cuenta específica, utilice el carácter comodín (*):

```
"Resource": "arn:aws:iotsitewise:region:123456789012:time-series/*"
```

Para especificar todos los activos que pertenecen a una cuenta específica, utilice el carácter comodín (*):

```
"Resource": "arn:aws:iotsitewise:region:123456789012:asset/*"
```

Algunas AWS IoT SiteWise acciones, como las de creación de recursos, no se pueden realizar en un recurso específico. En dichos casos, debe utilizar el carácter comodín (*).

```
"Resource": "*"
```

Para especificar varios recursos en una sola sentencia, sepárelos ARNs con comas.

```
"Resource": [  
  "resource1",  
  "resource2"  
]
```

Para ver una lista de los tipos de AWS IoT SiteWise recursos y sus correspondientes ARNs, consulte los [tipos de recursos definidos AWS IoT SiteWise en la](#) Guía del usuario de IAM. Para obtener

información sobre las acciones con las que puede especificar el ARN de cada recurso, consulte [Acciones definidas por AWS IoT SiteWise](#).

Claves de condición de políticas

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Condition` especifica cuándo se ejecutan las instrucciones en función de criterios definidos. Puede crear expresiones condicionales que utilizan [operadores de condición](#), tales como igual o menor que, para que la condición de la política coincida con los valores de la solicitud. Para ver todas las claves de condición AWS globales, consulte las claves de [contexto de condición AWS globales](#) en la Guía del usuario de IAM.

Important

Muchas claves de condición son específicas de un recurso y algunas acciones de API utilizan varios recursos. Si escribe una instrucción de política con una clave de condición, use el elemento `Resource` de la instrucción para especificar el recurso en el que se aplica la clave de condición. Si no lo hace, la política puede impedir que los usuarios ejecuten la acción, ya que la comprobación de la condición dará un error en el caso de los recursos en los que la clave de la condición no se aplica. Si no quiere especificar un recurso o si ha escrito el elemento `Action` de la política para que contenga varias acciones de API, debe utilizar el tipo de condición `...IfExists` para asegurarse de que no se tenga en cuenta la clave de condición en el caso de los recursos que no la utilicen. [Para obtener más información, consulte... IfExists](#) condiciones de la Guía del usuario de IAM.

AWS IoT SiteWise define su propio conjunto de claves de condición y también admite el uso de algunas claves de condición globales. Para ver todas las claves de condición AWS globales, consulte las claves de [contexto de condición AWS globales](#) en la Guía del usuario de IAM.

AWS IoT SiteWise claves de condición

Clave de condición	Description (Descripción)	Tipos
<code>iotsitewise:isAssociatedWithAssetProperty</code>	Si los flujos de datos están asociados a una propiedad de activo. Utilice esta clave	Cadena

Clave de condición	Description (Descripción)	Tipos
	<p>de condición para definir permisos basados en la existencia de una propiedad de activo asociada para flujos de datos.</p> <p>Ejemplo de valor: true</p>	
<code>iotsitewise:assetHierarchyPath</code>	<p>La ruta jerárquica del activo, que es una cadena de activos separados IDs cada uno por una barra diagonal. Utilice esta clave de condición para definir permisos basados en un subconjunto de la jerarquía de todos los activos de la cuenta.</p> <p>Ejemplo de valor: /a1b2c3d4-5678-90ab-cdef-2222EXAMPLE/a1b2c3d4-5678-90ab-cdef-6666EXAMPLE</p>	Cadena
<code>iotsitewise:propertyId</code>	<p>ID de una propiedad de activo. Utilice esta clave de condición para definir permisos basados en una propiedad específica de un modelo de activos. Esta clave de condición se aplica a todos los activos de ese modelo.</p> <p>Ejemplo de valor: a1b2c3d4-5678-90ab-cdef-3333EXAMPLE</p>	Cadena

Clave de condición	Description (Descripción)	Tipos
<code>iotsitewise:childAssetId</code>	<p>ID de un activo asociado como secundario a otro activo. Utilice esta clave de condición para definir permisos basados en activos secundarios. Para definir permisos basados en activos principales, utilice la sección de activos de una instrucción de política.</p> <p>Ejemplo de valor: a1b2c3d4-5678-90ab-cdef-6666EXAMPLE</p>	Cadena
<code>iotsitewise:iam</code>	<p>El ARN de una identidad de IAM al enumerar las políticas de acceso. Utilice esta clave de condición para definir permisos de política de acceso para una identidad de IAM.</p> <p>Ejemplo de valor: arn:aws:iam::123456789012:user/JohnDoe</p>	Cadena, null
<code>iotsitewise:propertyAlias</code>	<p>El alias que identifica una propiedad de activo o un flujo de datos. Utilice esta clave de condición para definir permisos basados en el alias.</p>	Cadena

Clave de condición	Description (Descripción)	Tipos
<code>iotsitewise:user</code>	<p>El ID de un usuario del Centro de identidades de IAM al enumerar las políticas de acceso. Utilice esta clave de condición para definir los permisos de la política de acceso para un usuario del Centro de identidades de IAM.</p> <p>Ejemplo de valor: a1b2c3d4e5-a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE</p>	Cadena, null
<code>iotsitewise:group</code>	<p>El ID de un grupo del Centro de identidades de IAM al enumerar las políticas de acceso. Utilice esta clave de condición para definir los permisos de la política de acceso para un grupo del Centro de identidades de IAM.</p> <p>Ejemplo de valor: a1b2c3d4e5-a1b2c3d4-5678-90ab-cdef-bbbbbbEXAMPLE</p>	Cadena, null

Clave de condición	Description (Descripción)	Tipos
<code>iotsitewise:portal</code>	<p>ID de un portal en una política de acceso. Utilice esta clave de condición para definir permisos de política de acceso basados en un portal.</p> <p>Ejemplo de valor: a1b2c3d4-5678-90ab-cdef-77777EXAMPLE</p>	Cadena, null
<code>iotsitewise:project</code>	<p>ID de un proyecto en una política de acceso o el ID de un proyecto para un panel. Utilice esta clave de condición para definir permisos de panel o política de acceso basados en un proyecto.</p> <p>Ejemplo de valor: a1b2c3d4-5678-90ab-cdef-88888EXAMPLE</p>	Cadena, null

Para saber con qué acciones y recursos puede utilizar una clave condicionada, consulte [Acciones definidas por AWS IoT SiteWise](#).

Ejemplos

Para ver ejemplos de políticas AWS IoT SiteWise basadas en la identidad, consulte. [AWS IoT SiteWise ejemplos de políticas basadas en la identidad](#)

AWS IoT SiteWise ejemplos de políticas basadas en la identidad

De forma predeterminada, las entidades (usuarios y roles) no tienen permiso para crear o modificar recursos de AWS IoT SiteWise . Tampoco pueden realizar tareas mediante la Consola de

administración de AWS, AWS Command Line Interface (AWS CLI) o la AWS API. Para ajustar los permisos, un administrador AWS Identity and Access Management (IAM) debe hacer lo siguiente:

1. Cree políticas de IAM que concedan permisos a los usuarios y a los roles para realizar operaciones de la API concretas en los recursos que necesiten.
2. Asocie esas políticas a los usuarios o usuarias o grupos que necesiten esos permisos.

Para obtener más información acerca de cómo crear una política basada en identidad de IAM con estos documentos de políticas de JSON de ejemplo, consulte [Creación de políticas en la pestaña JSON](#) en la Guía del usuario de IAM.

Temas

- [Prácticas recomendadas relativas a políticas](#)
- [Utilice la consola AWS IoT SiteWise](#)
- [Cómo permitir a los usuarios consultar sus propios permisos](#)
- [Concesión de permisos para que los usuarios ingieran datos en activos de una jerarquía](#)
- [Vea AWS IoT SiteWise los activos en función de las etiquetas](#)

Prácticas recomendadas relativas a políticas

Las políticas basadas en la identidad determinan si alguien puede crear AWS IoT SiteWise recursos de tu cuenta, acceder a ellos o eliminarlos. Estas acciones pueden generar costos adicionales para su Cuenta de AWS. Siga estas directrices y recomendaciones al crear o editar políticas basadas en identidades:

- Comience con las políticas AWS administradas y avance hacia los permisos con privilegios mínimos: para empezar a conceder permisos a sus usuarios y cargas de trabajo, utilice las políticas AWS administradas que otorgan permisos para muchos casos de uso comunes. Están disponibles en su Cuenta de AWS. Le recomendamos que reduzca aún más los permisos definiendo políticas administradas por el AWS cliente que sean específicas para sus casos de uso. Con el fin de obtener más información, consulte las [políticas administradas por AWS](#) o las [políticas administradas por AWS para funciones de tarea](#) en la Guía de usuario de IAM.
- Aplique permisos de privilegio mínimo: cuando establezca permisos con políticas de IAM, conceda solo los permisos necesarios para realizar una tarea. Para ello, debe definir las acciones que se pueden llevar a cabo en determinados recursos en condiciones específicas, también conocidos

como permisos de privilegios mínimos. Con el fin de obtener más información sobre el uso de IAM para aplicar permisos, consulte [Políticas y permisos en IAM](#) en la Guía del usuario de IAM.

- Utilice condiciones en las políticas de IAM para restringir aún más el acceso: puede agregar una condición a sus políticas para limitar el acceso a las acciones y los recursos. Por ejemplo, puede escribir una condición de políticas para especificar que todas las solicitudes deben enviarse utilizando SSL. También puedes usar condiciones para conceder el acceso a las acciones del servicio si se utilizan a través de una acción específica Servicio de AWS, por ejemplo CloudFormation. Para obtener más información, consulte [Elementos de la política de JSON de IAM: Condición](#) en la Guía del usuario de IAM.
- Utiliza el analizador de acceso de IAM para validar las políticas de IAM con el fin de garantizar la seguridad y funcionalidad de los permisos: el analizador de acceso de IAM valida políticas nuevas y existentes para que respeten el lenguaje (JSON) de las políticas de IAM y las prácticas recomendadas de IAM. El analizador de acceso de IAM proporciona más de 100 verificaciones de políticas y recomendaciones procesables para ayudar a crear políticas seguras y funcionales. Para más información, consulte [Validación de políticas con el Analizador de acceso de IAM](#) en la Guía del usuario de IAM.
- Requerir autenticación multifactor (MFA): si tiene un escenario que requiere usuarios de IAM o un usuario raíz en Cuenta de AWS su cuenta, active la MFA para mayor seguridad. Para exigir la MFA cuando se invoquen las operaciones de la API, añada condiciones de MFA a sus políticas. Para más información, consulte [Acceso seguro a la API con MFA](#) en la Guía del usuario de IAM.

Para obtener más información sobre las prácticas recomendadas de IAM, consulte [Prácticas recomendadas de seguridad en IAM](#) en la Guía del usuario de IAM.

Utilice la consola AWS IoT SiteWise

Para acceder a la AWS IoT SiteWise consola, necesita un conjunto básico de permisos. Estos permisos te permiten ver y administrar los detalles de los AWS IoT SiteWise recursos de tu AWS cuenta.

Si crea una política demasiado restrictiva, es posible que la consola no funcione como se espera para los usuarios o los roles (entidades) que tengan esa política. Para garantizar que esas entidades puedan seguir utilizando la AWS IoT SiteWise consola, adjúnteles la política [AWSIoTSiteWiseConsoleFullAccess](#) gestionada o defina permisos equivalentes para esas entidades. Para obtener más información, consulte [Adición de permisos a un usuario](#) en la Guía del usuario de IAM:

Si las entidades solo utilizan la AWS Command Line Interface (CLI) o la AWS IoT SiteWise API, y no la consola, no necesitan estos permisos mínimos. En ese caso, basta con darles acceso a las acciones específicas que necesitan para sus tareas de la API.

Cómo permitir a los usuarios consultar sus propios permisos

En este ejemplo, se muestra cómo podría crear una política que permita a los usuarios de IAM ver las políticas administradas e insertadas que se asocian a la identidad de sus usuarios. Esta política incluye permisos para completar esta acción en la consola o mediante programación mediante la API AWS CLI o AWS .

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupsWithUser",
        "iam:ListAttachedUserPolicies",
        "iam:ListUserPolicies",
        "iam:GetUser"
      ],
      "Resource": ["arn:aws:iam::*:user/${aws:username}"]
    },
    {
      "Sid": "NavigateInConsole",
      "Effect": "Allow",
      "Action": [
        "iam:GetGroupPolicy",
        "iam:GetPolicyVersion",
        "iam:GetPolicy",
        "iam:ListAttachedGroupPolicies",
        "iam:ListGroupPolicies",
        "iam:ListPolicyVersions",
        "iam:ListPolicies",
        "iam:ListUsers"
      ],
      "Resource": "*"
    }
  ]
}
```

```
}

```

Concesión de permisos para que los usuarios ingieran datos en activos de una jerarquía

En este ejemplo, desea conceder a un usuario de su AWS cuenta acceso para escribir datos en todas las propiedades de los activos de una jerarquía de activos específica, empezando por el activo raíz. a1b2c3d4-5678-90ab-cdef-22222EXAMPLE La política concede el permiso `iotsitewise:BatchPutAssetPropertyValue` al usuario. Esta política utiliza la clave de condición `iotsitewise:assetHierarchyPath` para restringir el acceso a los activos cuya ruta de jerarquía coincide con el activo o sus descendientes.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PutAssetPropertyValuesForHierarchy",
      "Effect": "Allow",
      "Action": "iotsitewise:BatchPutAssetPropertyValue",
      "Resource": "arn:aws:iotsitewise:*:*:asset/*",
      "Condition": {
        "StringLike": {
          "iotsitewise:assetHierarchyPath": [
            "/a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
            "/a1b2c3d4-5678-90ab-cdef-22222EXAMPLE/*"
          ]
        }
      }
    }
  ]
}
```

Vea AWS IoT SiteWise los activos en función de las etiquetas

Utilice las condiciones de su política basada en la identidad para controlar el acceso a AWS IoT SiteWise los recursos en función de las etiquetas. En este ejemplo se muestra cómo crear una política que permita visualizar un activo. Sin embargo, los activos solo se conceden si la etiqueta del activo `Owner` tiene el valor del nombre de usuario de dicho usuario. Esta política también concede permiso para llevar a cabo esta acción en la consola.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ListAllAssets",
      "Effect": "Allow",
      "Action": [
        "iotsitewise:ListAssets",
        "iotsitewise:ListAssociatedAssets"
      ],
      "Resource": "*"
    },
    {
      "Sid": "DescribeAssetIfOwner",
      "Effect": "Allow",
      "Action": "iotsitewise:DescribeAsset",
      "Resource": "arn:aws:iotsitewise:*:*:asset/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Owner": "${aws:username}"
        }
      }
    }
  ]
}
```

Asocie esta política a los usuarios de su cuenta. Si un usuario llamado `richard-roe` intenta ver un AWS IoT SiteWise activo, el activo debe estar etiquetado `Owner=richard-roe` o `owner=richard-roe`. De lo contrario, se deniega el acceso a Richard. Los nombres de claves de etiquetas de condición no distinguen entre mayúsculas y minúsculas. Por tanto, `Owner` coincide tanto con `Owner` como con `owner`. Para obtener más información, consulte [Elementos de la política de JSON de IAM: Condición](#) en la Guía del usuario de IAM.

Administre el acceso mediante políticas en AWS IoT SiteWise

El acceso se controla AWS creando políticas y adjuntándolas a AWS identidades o recursos. Una política define los permisos cuando están asociados a una identidad o un recurso. AWS evalúa estas políticas cuando un director hace una solicitud. La mayoría de las políticas se almacenan AWS como

documentos JSON. Para obtener más información sobre los documentos de políticas de JSON, consulte [Información general de políticas de JSON](#) en la Guía del usuario de IAM.

Mediante las políticas, los administradores especifican quién tiene acceso a qué, definiendo qué entidad principal puede realizar acciones sobre qué recursos y en qué condiciones.

De forma predeterminada, los usuarios y los roles no tienen permisos. Un administrador de IAM crea políticas de IAM y las agrega a roles, que los usuarios pueden asumir posteriormente. Las políticas de IAM definen permisos independientemente del método que se utilice para realizar la operación.

Políticas basadas en identidades

Las políticas basadas en identidad son documentos de política de permisos JSON que asocia a una identidad (usuario, grupo o rol). Estas políticas controlan qué acciones pueden realizar las identidades, en qué recursos y en qué condiciones. Para obtener más información sobre cómo crear una política basada en la identidad, consulte [Definición de permisos de IAM personalizados con políticas administradas por el cliente](#) en la Guía del usuario de IAM.

Las políticas basadas en identidad pueden ser políticas insertadas (incrustadas directamente en una sola identidad) o políticas administradas (políticas independientes asociadas a varias identidades). Para obtener información sobre cómo elegir entre políticas administradas e insertadas, consulte [Selección entre políticas administradas y políticas insertadas](#) en la Guía del usuario de IAM.

Políticas basadas en recursos

Las políticas basadas en recursos son documentos de políticas JSON que se asocian a un recurso. Los ejemplos incluyen las Políticas de confianza de roles de IAM y las Políticas de bucket de Amazon S3. En los servicios que admiten políticas basadas en recursos, los administradores de servicios pueden utilizarlos para controlar el acceso a un recurso específico. Debe [especificar una entidad principal](#) en una política basada en recursos.

Las políticas basadas en recursos son políticas insertadas que se encuentran en ese servicio. No puedes usar políticas AWS gestionadas de IAM en una política basada en recursos.

Listas de control de acceso (ACLs)

Las listas de control de acceso (ACLs) controlan qué responsables (miembros de la cuenta, usuarios o roles) tienen permisos para acceder a un recurso. ACLs son similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de políticas JSON.

Amazon S3 y Amazon VPC son ejemplos de servicios compatibles. AWS WAF ACLs Para obtener más información ACLs, consulte la [descripción general de la lista de control de acceso \(ACL\)](#) en la Guía para desarrolladores de Amazon Simple Storage Service.

Otros tipos de políticas

AWS admite tipos de políticas adicionales que pueden establecer los permisos máximos otorgados por los tipos de políticas más comunes:

- Límites de permisos: establecen los permisos máximos que una política basada en identidad puede conceder a una entidad de IAM. Para obtener más información, consulte [Límites de permisos para las entidades de IAM](#) en la Guía del usuario de IAM.
- Políticas de control de servicios (SCPs): especifican los permisos máximos para una organización o unidad organizativa en AWS Organizations. Para obtener más información, consulte [Políticas de control de servicios](#) en la Guía del usuario de AWS Organizations .
- Políticas de control de recursos (RCPs): establece los permisos máximos disponibles para los recursos de tus cuentas. Para obtener más información, consulte [Políticas de control de recursos \(RCPs\)](#) en la Guía del AWS Organizations usuario.
- Políticas de sesión: políticas avanzadas que se pasan como parámetro cuando se crea una sesión temporal para un rol o un usuario federado. Para obtener más información, consulte [Políticas de sesión](#) en la Guía del usuario de IAM.

Varios tipos de políticas

Cuando se aplican varios tipos de políticas a una solicitud, los permisos resultantes son más complicados de entender. Para saber cómo se AWS determina si se debe permitir una solicitud cuando se trata de varios tipos de políticas, consulte la [lógica de evaluación de políticas](#) en la Guía del usuario de IAM.

AWS políticas gestionadas para AWS IoT SiteWise

Simplifique la adición de permisos a los usuarios, grupos y roles mediante políticas AWS administradas en lugar de tener que escribir las políticas usted mismo. Se necesita tiempo y experiencia para [crear políticas administradas por el cliente de IAM](#) que proporcionen a su equipo los permisos necesarios. Para una configuración más rápida, considere usar nuestras políticas AWS administradas para casos de uso comunes. Busca políticas AWS administradas en tu AWS cuenta. Para obtener más información sobre las políticas administradas de AWS , consulte [Políticas administradas de AWS](#) en la Guía del usuario de IAM.

AWS los servicios se encargan de actualizar y mantener las políticas AWS administradas, lo que significa que no puedes modificar los permisos de estas políticas. Ocasionalmente, AWS IoT SiteWise pueden añadir permisos para adaptarse a nuevas funciones, lo que afecta a todas las identidades a las que se adjunta la política. Estas actualizaciones son habituales con la introducción de nuevos servicios o características. Sin embargo, los permisos nunca se eliminan, lo que garantiza que las configuraciones permanezcan intactas.

Además, AWS admite políticas gestionadas para funciones laborales que abarcan varios servicios. Por ejemplo, la política `ReadOnlyAccess` AWS gestionada proporciona acceso de solo lectura a todos los AWS servicios y recursos. Cuando un servicio lanza una nueva función, AWS agrega permisos de solo lectura para nuevas operaciones y recursos. Para obtener una lista con descripciones de las políticas de funciones laborales, consulte [Políticas administradas por AWS para funciones laborales](#) en la Guía del usuario de IAM.

AWS política gestionada: `AWSIoTSiteWiseReadOnlyAccess`

Utilice la política `AWSIoTSiteWiseReadOnlyAccess` AWS gestionada para permitir el acceso de solo lectura a AWS IoT SiteWise

Puede asociar la política `AWSIoTSiteWiseReadOnlyAccess` a las identidades de IAM.

Permisos de nivel de servicio

Esta política proporciona acceso de solo lectura a consultas SQL de solo lectura AWS IoT SiteWise, incluidos permisos para ejecutarlas. No se incluyen otros permisos de servicio en esta política.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iotsitewise:BatchGetAssetPropertyAggregates",
        "iotsitewise:BatchGetAssetPropertyValue",
        "iotsitewise:BatchGetAssetPropertyValueHistory",
        "iotsitewise:DescribeAccessPolicy",
        "iotsitewise:DescribeAction",
        "iotsitewise:DescribeAsset",
        "iotsitewise:DescribeAssetCompositeModel",
```

```
"iotsitewise:DescribeAssetModel",
"iotsitewise:DescribeAssetModelCompositeModel",
"iotsitewise:DescribeAssetModelInterfaceRelationship",
"iotsitewise:DescribeAssetProperty",
"iotsitewise:DescribeBulkImportJob",
"iotsitewise:DescribeComputationModel",
"iotsitewise:DescribeComputationModelExecutionSummary",
"iotsitewise:DescribeDashboard",
"iotsitewise:DescribeDataset",
"iotsitewise:DescribeDefaultEncryptionConfiguration",
"iotsitewise:DescribeExecution",
"iotsitewise:DescribeGateway",
"iotsitewise:DescribeGatewayCapabilityConfiguration",
"iotsitewise:DescribeLoggingOptions",
"iotsitewise:DescribePortal",
"iotsitewise:DescribeProject",
"iotsitewise:DescribeStorageConfiguration",
"iotsitewise:DescribeTimeSeries",
"iotsitewise:GetAssetPropertyAggregates",
"iotsitewise:GetAssetPropertyValue",
"iotsitewise:GetAssetPropertyValueHistory",
"iotsitewise:GetInterpolatedAssetPropertyValues",
"iotsitewise:ListAccessPolicies",
"iotsitewise:ListActions",
"iotsitewise:ListAssetModelCompositeModels",
"iotsitewise:ListAssetModelProperties",
"iotsitewise:ListAssetModels",
"iotsitewise:ListAssetProperties",
"iotsitewise:ListAssetRelationships",
"iotsitewise:ListAssets",
"iotsitewise:ListAssociatedAssets",
"iotsitewise:ListBulkImportJobs",
"iotsitewise:ListCompositionRelationships",
"iotsitewise:ListComputationModelDataBindingUsages",
"iotsitewise:ListComputationModelResolveToResources",
"iotsitewise:ListComputationModels",
"iotsitewise:ListDashboards",
"iotsitewise:ListDatasets",
"iotsitewise:ListExecutions",
"iotsitewise:ListGateways",
"iotsitewise:ListInterfaceRelationships",
"iotsitewise:ListPortals",
"iotsitewise:ListProjectAssets",
"iotsitewise:ListProjects",
```

```

        "iotsitewise:ListTagsForResource",
        "iotsitewise:ListTimeSeries"
    ],
    "Resource": "*"
}
]
}

```

AWS política gestionada: Wise AWSService RoleForIo TSite

El rol `AWSServiceRoleForIoTSiteWise` utiliza la política `AWSServiceRoleForIoTSiteWise` con los siguientes permisos. Esta política:

- Permite AWS IoT SiteWise implementar pasarelas SiteWise Edge (que se ejecutan en ellasAWS IoT Greengrass).
- Permite AWS IoT SiteWise realizar el registro.
- Permite AWS IoT SiteWise ejecutar una consulta de búsqueda de metadatos en la AWS IoT TwinMaker base de datos.

Si la utiliza AWS IoT SiteWise con una sola cuenta de usuario, el `AWSServiceRoleForIoTSiteWise` rol crea la `AWSServiceRoleForIoTSiteWise` política en su cuenta de IAM y la adjunta a las funciones vinculadas al `AWSServiceRoleForIoTSiteWise` [servicio](#). AWS IoT SiteWise

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowSiteWiseReadGreenGrass",
      "Effect": "Allow",
      "Action": [
        "greengrass:GetAssociatedRole",
        "greengrass:GetCoreDefinition",
        "greengrass:GetCoreDefinitionVersion",
        "greengrass:GetGroup",
        "greengrass:GetGroupVersion"
      ]
    }
  ]
}

```

```

    ],
    "Resource": "*"
  },
  {
    "Sid": "AllowSiteWiseAccessLogGroup",
    "Effect": "Allow",
    "Action": [
      "logs:CreateLogGroup",
      "logs:DescribeLogGroups"
    ],
    "Resource": "arn:aws:logs:*:*:log-group:/aws/iotsitewise*"
  },
  {
    "Sid": "AllowSiteWiseAccessLog",
    "Effect": "Allow",
    "Action": [
      "logs:CreateLogStream",
      "logs:DescribeLogStreams",
      "logs:PutLogEvents"
    ],
    "Resource": "arn:aws:logs:*:*:log-group:/aws/iotsitewise*:log-stream:*"
  },
  {
    "Sid": "AllowSiteWiseAccessSiteWiseManagedWorkspaceInTwinMaker",
    "Effect": "Allow",
    "Action": [
      "iottwinmaker:GetWorkspace",
      "iottwinmaker:ExecuteQuery"
    ],
    "Resource": "arn:aws:iottwinmaker:*:*:workspace/*",
    "Condition": {
      "ForAnyValue:StringEquals": {
        "iottwinmaker:linkedServices": [
          "IOTSITewise"
        ]
      }
    }
  }
]
}

```

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowSiteWiseReadGreenGrass",
      "Effect": "Allow",
      "Action": [
        "greengrass:GetAssociatedRole",
        "greengrass:GetCoreDefinition",
        "greengrass:GetCoreDefinitionVersion",
        "greengrass:GetGroup",
        "greengrass:GetGroupVersion"
      ],
      "Resource": "*"
    },
    {
      "Sid": "AllowSiteWiseAccessLogGroup",
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:DescribeLogGroups"
      ],
      "Resource": "arn:aws-us-gov:logs:*:*:log-group:/aws/iotsitewise*"
    },
    {
      "Sid": "AllowSiteWiseAccessLog",
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogStream",
        "logs:DescribeLogStreams",
        "logs:PutLogEvents"
      ],
      "Resource": "arn:aws-us-gov:logs:*:*:log-group:/aws/iotsitewise*:log-stream:*"
    },
    {
      "Sid": "AllowSiteWiseAccessSiteWiseManagedWorkspaceInTwinMaker",
      "Effect": "Allow",
      "Action": [
        "iottwinmaker:GetWorkspace",
        "iottwinmaker:ExecuteQuery"
      ]
    }
  ]
}

```

```

    ],
    "Resource": "arn:aws-us-gov:iottwinmaker:*:*:workspace/*",
    "Condition": {
      "ForAnyValue:StringEquals": {
        "iottwinmaker:linkedServices": [
          "IOTSITWISE"
        ]
      }
    }
  }
]
}

```

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowSiteWiseReadGreenGrass",
      "Effect": "Allow",
      "Action": [
        "greengrass:GetAssociatedRole",
        "greengrass:GetCoreDefinition",
        "greengrass:GetCoreDefinitionVersion",
        "greengrass:GetGroup",
        "greengrass:GetGroupVersion"
      ],
      "Resource": "*"
    },
    {
      "Sid": "AllowSiteWiseAccessLogGroup",
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:DescribeLogGroups"
      ],
      "Resource": "arn:aws-cn:logs:*:*:log-group:/aws/iotsitewise*"
    }
  ]
}

```

```

    "Sid": "AllowSiteWiseAccessLog",
    "Effect": "Allow",
    "Action": [
        "logs:CreateLogStream",
        "logs:DescribeLogStreams",
        "logs:PutLogEvents"
    ],
    "Resource": "arn:aws-cn:logs:*:*:log-group:/aws/iotsitewise*:log-stream:*"
},
{
    "Sid": "AllowSiteWiseAccessSiteWiseManagedWorkspaceInTwinMaker",
    "Effect": "Allow",
    "Action": [
        "iottwinmaker:GetWorkspace",
        "iottwinmaker:ExecuteQuery"
    ],
    "Resource": "arn:aws-cn:iottwinmaker:*:*:workspace/*",
    "Condition": {
        "ForAnyValue:StringEquals": {
            "iottwinmaker:linkedServices": [
                "IOTSITWISE"
            ]
        }
    }
}
]
}

```

AWS IoT SiteWise actualizaciones de las políticas gestionadas AWS

Puede ver los detalles sobre las actualizaciones de las políticas AWS administradas desde el momento en que este servicio comenzó a realizar el seguimiento de los cambios. AWS IoT SiteWise Para recibir alertas automáticas sobre los cambios en esta página, suscríbese a la fuente RSS de la página del historial del AWS IoT SiteWise documento.

Cambio	Descripción	Fecha
AWSServiceRoleForIoTSiteWise : actualice una política existente	AWS IoT SiteWise ahora puede ejecutar una consulta de búsqueda de metadatos en	6 de noviembre de 2023

Cambio	Descripción	Fecha
	la AWS IoT TwinMaker base de datos.	
AWSIoTSiteWiseRead OnlyAccess : actualización de una política actual	AWS IoT SiteWise agregó un nuevo prefijo de política, BatchGet* , que le permite realizar operaciones de lectura por lotes.	16 de septiembre de 2022
AWSIoTSiteWiseRead OnlyAccess : política nueva	AWS IoT SiteWise agregó una nueva política para conceder acceso de solo lectura a. AWS IoT SiteWise	24 de noviembre de 2021
AWS IoT SiteWise comenzó a rastrear los cambios	AWS IoT SiteWise comenzó a realizar un seguimiento de los cambios de sus políticas AWS gestionadas.	24 de noviembre de 2021

Utilice funciones vinculadas al servicio para AWS IoT SiteWise

AWS IoT SiteWise utiliza funciones AWS Identity and Access Management vinculadas al [servicio](#) (IAM). Un rol vinculado a un servicio es un tipo único de rol de IAM al que se vincula directamente AWS IoT SiteWise. Los roles vinculados al servicio están predefinidos AWS IoT SiteWise e incluyen todos los permisos que el servicio requiere para llamar a otros servicios en su nombre. AWS

Los roles vinculados al servicio simplifican la configuración al incluir automáticamente todos los permisos necesarios AWS IoT SiteWise . AWS IoT SiteWise define los permisos de sus funciones vinculadas al servicio y, a menos que se defina lo contrario, solo AWS IoT SiteWise puede asumir sus funciones. Los permisos definidos incluyen la política de confianza y la política de permisos. Y dicha política de permisos no se puede asociar a ninguna otra entidad de IAM.

Solo es posible eliminar un rol vinculado a un servicio después de eliminar sus recursos relacionados. Esto protege sus AWS IoT SiteWise recursos porque no puede eliminar inadvertidamente el permiso de acceso a los recursos.

Para obtener información sobre otros servicios que admiten funciones vinculadas a servicios, consulte los [AWS servicios que funcionan con IAM y busque los servicios con](#) la palabra Sí en la columna Función vinculada a servicios. Elija una opción Sí con un enlace para ver la documentación acerca del rol vinculado al servicio en cuestión.

Temas

- [Permisos de rol vinculados al servicio para AWS IoT SiteWise](#)
- [Cree un rol vinculado a un servicio para AWS IoT SiteWise](#)
- [Actualice un rol vinculado a un servicio para AWS IoT SiteWise](#)
- [Elimina un rol vinculado a un servicio para AWS IoT SiteWise](#)
- [Regiones compatibles para los roles vinculados al servicio AWS IoT SiteWise](#)
- [Utilice los roles de servicio para AWS IoT SiteWise Monitor](#)

Permisos de rol vinculados al servicio para AWS IoT SiteWise

AWS IoT SiteWise utiliza el rol vinculado al servicio denominado `AWSServiceRoleForIoTSiteWise`. AWS IoT SiteWise utiliza esta función vinculada a un servicio para implementar las puertas de enlace SiteWise Edge (que se ejecutan en ellas) y realizar el AWS IoT Greengrass registro.

El rol vinculado a servicios de `AWSServiceRoleForIoTSiteWise` usa la política de `AWSServiceRoleForIoTSiteWise` con los siguientes permisos. Esta política:

- Permite AWS IoT SiteWise implementar puertas de enlace SiteWise Edge (que se ejecutan en ellas). AWS IoT Greengrass
- Permite AWS IoT SiteWise realizar el registro.
- Permite AWS IoT SiteWise ejecutar una consulta de búsqueda de metadatos en la AWS IoT TwinMaker base de datos.

Para obtener más información sobre las acciones permitidas en `AWSServiceRoleForIoTSiteWise`, consulte [Políticas administradas por AWS para AWS IoT SiteWise](#).

JSON

```
{  
  "Version": "2012-10-17",
```

```

"Statement": [
  {
    "Sid": "AllowSiteWiseReadGreenGrass",
    "Effect": "Allow",
    "Action": [
      "greengrass:GetAssociatedRole",
      "greengrass:GetCoreDefinition",
      "greengrass:GetCoreDefinitionVersion",
      "greengrass:GetGroup",
      "greengrass:GetGroupVersion"
    ],
    "Resource": "*"
  },
  {
    "Sid": "AllowSiteWiseAccessLogGroup",
    "Effect": "Allow",
    "Action": [
      "logs:CreateLogGroup",
      "logs:DescribeLogGroups"
    ],
    "Resource": "arn:aws:logs:*:*:log-group:/aws/iotsitewise*"
  },
  {
    "Sid": "AllowSiteWiseAccessLog",
    "Effect": "Allow",
    "Action": [
      "logs:CreateLogStream",
      "logs:DescribeLogStreams",
      "logs:PutLogEvents"
    ],
    "Resource": "arn:aws:logs:*:*:log-group:/aws/iotsitewise*:log-stream:*"
  },
  {
    "Sid": "AllowSiteWiseAccessSiteWiseManagedWorkspaceInTwinMaker",
    "Effect": "Allow",
    "Action": [
      "iottwinmaker:GetWorkspace",
      "iottwinmaker:ExecuteQuery"
    ],
    "Resource": "arn:aws:iottwinmaker:*:*:workspace/*",
    "Condition": {
      "ForAnyValue:StringEquals": {
        "iottwinmaker:linkedServices": [
          "IOTSITWISE"
        ]
      }
    }
  }
]

```

```

    ]
  }
}
]
}

```

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowSiteWiseReadGreenGrass",
      "Effect": "Allow",
      "Action": [
        "greengrass:GetAssociatedRole",
        "greengrass:GetCoreDefinition",
        "greengrass:GetCoreDefinitionVersion",
        "greengrass:GetGroup",
        "greengrass:GetGroupVersion"
      ],
      "Resource": "*"
    },
    {
      "Sid": "AllowSiteWiseAccessLogGroup",
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:DescribeLogGroups"
      ],
      "Resource": "arn:aws-us-gov:logs:*:*:log-group:/aws/iotsitewise*"
    },
    {
      "Sid": "AllowSiteWiseAccessLog",
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogStream",
        "logs:DescribeLogStreams",
        "logs:PutLogEvents"
      ]
    }
  ]
}

```

```

    ],
    "Resource": "arn:aws-us-gov:logs:*:*:log-group:/aws/iotsitewise*:log-stream:*"
  },
  {
    "Sid": "AllowSiteWiseAccessSiteWiseManagedWorkspaceInTwinMaker",
    "Effect": "Allow",
    "Action": [
      "iottwinmaker:GetWorkspace",
      "iottwinmaker:ExecuteQuery"
    ],
    "Resource": "arn:aws-us-gov:iottwinmaker:*:*:workspace/*",
    "Condition": {
      "ForAnyValue:StringEquals": {
        "iottwinmaker:linkedServices": [
          "IOTSITWISE"
        ]
      }
    }
  }
]
}

```

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowSiteWiseReadGreenGrass",
      "Effect": "Allow",
      "Action": [
        "greengrass:GetAssociatedRole",
        "greengrass:GetCoreDefinition",
        "greengrass:GetCoreDefinitionVersion",
        "greengrass:GetGroup",
        "greengrass:GetGroupVersion"
      ],
      "Resource": "*"
    }
  ]
}

```

```

    "Sid": "AllowSiteWiseAccessLogGroup",
    "Effect": "Allow",
    "Action": [
        "logs:CreateLogGroup",
        "logs:DescribeLogGroups"
    ],
    "Resource": "arn:aws-cn:logs:*:*:log-group:/aws/iotsitewise*"
},
{
    "Sid": "AllowSiteWiseAccessLog",
    "Effect": "Allow",
    "Action": [
        "logs:CreateLogStream",
        "logs:DescribeLogStreams",
        "logs:PutLogEvents"
    ],
    "Resource": "arn:aws-cn:logs:*:*:log-group:/aws/iotsitewise*:log-stream:*"
},
{
    "Sid": "AllowSiteWiseAccessSiteWiseManagedWorkspaceInTwinMaker",
    "Effect": "Allow",
    "Action": [
        "iottwinmaker:GetWorkspace",
        "iottwinmaker:ExecuteQuery"
    ],
    "Resource": "arn:aws-cn:iottwinmaker:*:*:workspace/*",
    "Condition": {
        "ForAnyValue:StringEquals": {
            "iottwinmaker:linkedServices": [
                "IOTSITWISE"
            ]
        }
    }
}
]
}

```

Puede usar los registros para monitorear y solucionar problemas de sus puertas de enlace SiteWise Edge. Para obtener más información, consulte [Supervise los registros de SiteWise Edge Gateway](#).

Para permitir que una entidad de IAM (como un usuario, grupo o rol) cree, edite o elimine un rol vinculado al servicio, primero configure los permisos. Para obtener más información, consulte [Permisos de roles vinculados a servicios](#) en la Guía del usuario de IAM.

Cree un rol vinculado a un servicio para AWS IoT SiteWise

AWS IoT SiteWise requiere una función vinculada al servicio para realizar determinadas acciones y acceder a los recursos en su nombre. Un rol vinculado a un servicio es un tipo único de rol de AWS Identity and Access Management (IAM) al que se vincula directamente. AWS IoT SiteWise Al crear este rol, AWS IoT SiteWise concede los permisos necesarios para acceder a otros AWS servicios y recursos necesarios para su funcionamiento, como Amazon S3 para el almacenamiento de datos o AWS IoT para la comunicación entre dispositivos.

No necesita crear manualmente un rol vinculado a servicios. Al realizar las siguientes operaciones en la AWS IoT SiteWise consola, AWS IoT SiteWise crea automáticamente el rol vinculado al servicio.

- Cree una puerta de enlace Greengrass V1.
- Configure la opción de registro.
- Selección del botón de suscripción en el banner de ejecución de consultas.

Si elimina este rol vinculado a servicios y necesita crearlo de nuevo, puede utilizar el mismo proceso para volver a crear el rol en su cuenta. Al realizar cualquier operación en la AWS IoT SiteWise consola, vuelve a AWS IoT SiteWise crear el rol vinculado al servicio para usted.

También puede utilizar la consola o la API de IAM para crear un rol de IAM vinculado al servicio para AWS IoT SiteWise.

- Para hacerlo en la consola de IAM, crea un rol con la política de `AWSServiceRoleForIoTSiteWise` y una relación de confianza con `iotsitewise.amazonaws.com`
- Para hacerlo mediante la API AWS CLI o la API de IAM, cree un rol con la `arn:aws:iam::aws:policy/aws-service-role/AWSServiceRoleForIoTSiteWise` política y una relación de confianza con `iotsitewise.amazonaws.com`

Para obtener más información, consulte [Creación de un rol vinculado al servicio](#) en la Guía del usuario de IAM.

Si elimina este rol vinculado al servicio, puede utilizar este mismo proceso para volver a crear el rol.

Actualice un rol vinculado a un servicio para AWS IoT SiteWise

AWS IoT SiteWise no permite editar el rol vinculado al servicio de AWSService RoleForlo TSite Wise. Después de crear un rol vinculado a un servicio, no puede cambiarle el nombre, ya que varias entidades pueden hacer referencia a él. Sin embargo, puede editar la descripción del rol mediante IAM. Para obtener más información, consulte [Actualizar un rol vinculado a servicios](#) en la Guía del usuario de IAM.

Elimina un rol vinculado a un servicio para AWS IoT SiteWise

Si ya no se utiliza una característica o un servicio que requiere un rol vinculado a servicios, se recomienda eliminar el rol asociado. Esto evita tener una entidad inactiva que no se supervisa ni mantiene. Sin embargo, debe limpiar los recursos de su rol vinculado al servicio antes de eliminarlo manualmente.

Note

Si el AWS IoT SiteWise servicio utiliza el rol al intentar eliminar los recursos, es posible que la eliminación no se realice correctamente. Si eso sucede, espere unos minutos e inténtelo de nuevo.

Para eliminar AWS IoT SiteWise los recursos utilizados por The AWSService RoleForlo TSite Wise

1. Desactivar el registro de AWS IoT SiteWise. Para obtener más información, consulte [Cambio del nivel de registro](#)
2. Elimine todas las puertas de enlace SiteWise Edge activas.

Para eliminar manualmente el rol vinculado a servicios mediante IAM

Utilice la consola de IAM AWS CLI, la o la AWS API para eliminar el rol vinculado al servicio de AWSService RoleForlo TSite Wise. Para obtener más información, consulte [Eliminar roles o perfiles de instancia](#) en la Guía del usuario de IAM.

Regiones compatibles para los roles vinculados al servicio AWS IoT SiteWise

AWS IoT SiteWise admite el uso de funciones vinculadas al servicio en todas las regiones en las que el servicio está disponible. Para obtener más información, consulte [Puntos de conexión y cuotas de AWS IoT SiteWise](#).

Utilice los roles de servicio para AWS IoT SiteWise Monitor

Un rol de servicio es un [rol de IAM](#) que asume un servicio para realizar acciones en su nombre. Un administrador de IAM puede crear, modificar y eliminar un rol de servicio desde IAM. Para obtener más información, consulte [Crear un rol para delegar permisos a un Servicio de AWS](#) en la Guía del usuario de IAM.

Para permitir que los usuarios federados del portal SiteWise Monitor accedan a sus AWS IAM Identity Center recursos AWS IoT SiteWise a sus recursos, debe asignar un rol de servicio a cada portal que cree. El rol de servicio debe especificar a SiteWise Monitor como entidad de confianza e incluir la política [AWSIoTSiteWiseMonitorPortalAccess](#) administrada o definir [permisos equivalentes](#). Esta política la mantiene AWS y define el conjunto de permisos que SiteWise Monitor utiliza para acceder a sus recursos AWS IoT SiteWise y a los del IAM Identity Center.

Al crear un portal de SiteWise Monitor, debe elegir una función que permita a los usuarios de ese portal acceder a sus recursos AWS IoT SiteWise a los del IAM Identity Center. La AWS IoT SiteWise consola puede crear y configurar el rol por usted. Puede editar el rol en IAM más adelante. Los usuarios de su portal tendrán problemas al usar sus portales de SiteWise Monitor si quita los permisos necesarios del rol o elimina el rol.

Note

Los portales creados antes del 29 de abril de 2020 no requerían roles de servicio. Si creó portales antes de esta fecha, deberá asociarles roles de servicio para poder seguir usándolos. Para ello, vaya a la página Portales de la [consola de AWS IoT SiteWise](#) y elija Migrar todos los portales a fin de utilizar roles de IAM.

En las siguientes secciones se describe cómo crear y administrar el rol del servicio de SiteWise monitoreo en el Consola de administración de AWS o el AWS Command Line Interface.

Contenido

- [Permisos de rol de servicio para SiteWise Monitor \(Classic\)](#)
- [Permisos de rol de servicio para SiteWise Monitor \(compatibles con la IA\)](#)
- [Administre la función de servicio SiteWise Supervisar \(consola\)](#)
 - [Búsqueda del rol de servicio de un portal \(consola\)](#)
 - [Cree un rol de servicio de SiteWise monitoreo \(AWS IoT SiteWise consola\)](#)

- [Cree un rol de servicio de SiteWise supervisión \(consola de IAM\)](#)
- [Cambio del rol de servicio de un portal \(consola\)](#)
- [Administrar la función de servicio de SiteWise supervisión \(CLI\)](#)
 - [Búsqueda del rol de servicio de un portal \(CLI\)](#)
 - [Crear el rol de servicio de SiteWise monitoreo \(CLI\)](#)
- [SiteWise Supervise las actualizaciones de AWS IoT SiteWise Monitor Service Role](#)

Permisos de rol de servicio para SiteWise Monitor (Classic)

Al crear un portal, AWS IoT SiteWise permite crear un rol cuyo nombre comience por `AWSIoTSiteWiseMonitorServiceRole`. Esta función permite a los usuarios federados de SiteWise Monitor acceder a la configuración del portal, a los activos, a los datos de activos y a los datos de configuración del IAM Identity Center.

El rol confía en el siguiente servicio para asumir el rol:

- `monitor.iotsitewise.amazonaws.com`

El rol usa la siguiente política de permisos, que comienza con `AWSIoTSiteWiseMonitorServicePortalPolicy`, para permitir a los usuarios de SiteWise Monitor realizar acciones en los recursos de su cuenta. La política administrada [AWSIoTSiteWiseMonitorPortalAccess](#) define permisos equivalentes.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iotsitewise:DescribePortal",
        "iotsitewise:CreateProject",
        "iotsitewise:DescribeProject",
        "iotsitewise:UpdateProject",
        "iotsitewise>DeleteProject",
        "iotsitewise:ListProjects",
        "iotsitewise:BatchAssociateProjectAssets",
```

```

        "iotsitewise:BatchDisassociateProjectAssets",
        "iotsitewise:ListProjectAssets",
        "iotsitewise:CreateDashboard",
        "iotsitewise:DescribeDashboard",
        "iotsitewise:UpdateDashboard",
        "iotsitewise>DeleteDashboard",
        "iotsitewise:ListDashboards",
        "iotsitewise:CreateAccessPolicy",
        "iotsitewise:DescribeAccessPolicy",
        "iotsitewise:UpdateAccessPolicy",
        "iotsitewise>DeleteAccessPolicy",
        "iotsitewise:ListAccessPolicies",
        "iotsitewise:DescribeAsset",
        "iotsitewise:ListAssets",
        "iotsitewise:ListAssociatedAssets",
        "iotsitewise:DescribeAssetProperty",
        "iotsitewise:GetAssetPropertyValue",
        "iotsitewise:GetAssetPropertyValueHistory",
        "iotsitewise:GetAssetPropertyAggregates",
        "iotsitewise:BatchPutAssetPropertyValue",
        "iotsitewise:ListAssetRelationships",
        "iotsitewise:DescribeAssetModel",
        "iotsitewise:ListAssetModels",
        "iotsitewise:UpdateAssetModel",
        "iotsitewise:UpdateAssetModelPropertyRouting",
        "sso-directory:DescribeUsers",
        "sso-directory:DescribeUser",
        "iotevents:DescribeAlarmModel",
        "iotevents:ListTagsForResource"
    ],
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "iotevents:BatchAcknowledgeAlarm",
        "iotevents:BatchSnoozeAlarm",
        "iotevents:BatchEnableAlarm",
        "iotevents:BatchDisableAlarm"
    ],
    "Resource": "*",
    "Condition": {
        "Null": {
            "iotevents:keyValue": "false"
        }
    }
}

```

```

    }
  }
},
{
  "Effect": "Allow",
  "Action": [
    "iotevents:CreateAlarmModel",
    "iotevents:TagResource"
  ],
  "Resource": "*",
  "Condition": {
    "Null": {
      "aws:RequestTag/iotsitewisemonitor": "false"
    }
  }
},
{
  "Effect": "Allow",
  "Action": [
    "iotevents:UpdateAlarmModel",
    "iotevents>DeleteAlarmModel"
  ],
  "Resource": "*",
  "Condition": {
    "Null": {
      "aws:ResourceTag/iotsitewisemonitor": "false"
    }
  }
},
{
  "Effect": "Allow",
  "Action": [
    "iam:PassRole"
  ],
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "iam:PassedToService": [
        "iotevents.amazonaws.com"
      ]
    }
  }
}
]

```

}

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iotsitewise:CreateProject",
        "iotsitewise:DescribeProject",
        "iotsitewise:UpdateProject",
        "iotsitewise>DeleteProject",
        "iotsitewise:ListProjects",
        "iotsitewise:BatchAssociateProjectAssets",
        "iotsitewise:BatchDisassociateProjectAssets",
        "iotsitewise:ListProjectAssets",
        "iotsitewise:CreateDashboard",
        "iotsitewise:DescribeDashboard",
        "iotsitewise:UpdateDashboard",
        "iotsitewise>DeleteDashboard",
        "iotsitewise:ListDashboards",
        "iotsitewise:CreateAccessPolicy",
        "iotsitewise:DescribeAccessPolicy",
        "iotsitewise:UpdateAccessPolicy",
        "iotsitewise>DeleteAccessPolicy",
        "iotsitewise:ListAccessPolicies",
        "iotsitewise:DescribeAsset",
        "iotsitewise:ListAssets",
        "iotsitewise:ListAssociatedAssets",
        "iotsitewise:DescribeAssetProperty",
        "iotsitewise:GetAssetPropertyValue",
        "iotsitewise:GetAssetPropertyValueHistory",
        "iotsitewise:GetAssetPropertyAggregates"
      ],
      "Resource": "*"
    }
  ]
}


```

Para obtener más información sobre los permisos necesarios para las alarmas, consulte [Configure los permisos para las alarmas de eventos en AWS IoT SiteWise](#).

Cuando un usuario del portal inicia sesión, SiteWise Monitor crea una [política de sesión](#) basada en la intersección del rol de servicio y las políticas de acceso de ese usuario. Las políticas de acceso definen el nivel de acceso de las identidades de a sus portales y proyectos. Para obtener más información sobre los permisos y las políticas de acceso al portal, consulte [Administre sus portales de SiteWise Monitor](#) y [CreateAccessPolicy](#).

Permisos de rol de servicio para SiteWise Monitor (compatibles con la IA)

Al crear un portal, AWS IoT SiteWise permite crear un rol cuyo nombre comience por lo. TSite WisePortalRole Esta función permite a los usuarios federados de SiteWise Monitor acceder a la configuración del portal, a los activos, a los datos de activos y a los datos de configuración del IAM Identity Center.

 Warning

SiteWise Monitor (compatible con la IA) no admite los roles de propietario del proyecto y visor del proyecto.

El rol confía en el siguiente servicio para asumir el rol:

- `monitor.iotsitewise.amazonaws.com`

El rol usa la siguiente política de permisos, que comienza con lo TSite Wise AIPortal AccessPolicy, para permitir a los usuarios de SiteWise Monitor realizar acciones en los recursos de tu cuenta.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iotsitewise:CreateProject",
        "iotsitewise:DescribePortal",
        "iotsitewise:ListProjects",
```

```

        "iotsitewise:DescribeProject",
        "iotsitewise:UpdateProject",
        "iotsitewise>DeleteProject",
        "iotsitewise>CreateDashboard",
        "iotsitewise:DescribeDashboard",
        "iotsitewise:UpdateDashboard",
        "iotsitewise>DeleteDashboard",
        "iotsitewise:ListDashboards",
        "iotsitewise:ListAssets",
        "iotsitewise:DescribeAsset",
        "iotsitewise:ListAssociatedAssets",
        "iotsitewise:ListAssetProperties",
        "iotsitewise:DescribeAssetProperty",
        "iotsitewise:GetAssetPropertyValue",
        "iotsitewise:GetAssetPropertyValueHistory",
        "iotsitewise:GetAssetPropertyAggregates",
        "iotsitewise:GetInterpolatedAssetPropertyValues",
        "iotsitewise:BatchGetAssetPropertyAggregates",
        "iotsitewise:BatchGetAssetPropertyValue",
        "iotsitewise:BatchGetAssetPropertyValueHistory",
        "iotsitewise:ListAssetRelationships",
        "iotsitewise:DescribeAssetModel",
        "iotsitewise:ListAssetModels",
        "iotsitewise:DescribeAssetCompositeModel",
        "iotsitewise:DescribeAssetModelCompositeModel",
        "iotsitewise:ListAssetModelProperties",
        "iotsitewise:ExecuteQuery",
        "iotsitewise:ListTimeSeries",
        "iotsitewise:DescribeTimeSeries",
        "iotsitewise:InvokeAssistant",
        "iotsitewise:DescribeDataset",
        "iotsitewise:ListDatasets",
        "iotevents:DescribeAlarmModel",
        "iotevents:ListTagsForResource",
        "iottwinmaker:ListWorkspaces",
        "iottwinmaker:ExecuteQuery",
        "iottwinmaker:GetWorkspace",
        "identitystore:DescribeUser"
    ],
    "Resource": "*"
}
]
}

```

Cuando un usuario del portal inicia sesión, SiteWise Monitor crea una [política de sesión](#) basada en la intersección del rol de servicio y las políticas de acceso de ese usuario.

Administre la función de servicio SiteWise Supervisar (consola)

Esto Consola de AWS IoT SiteWise facilita la administración de la función de servicio de SiteWise monitoreo para los portales. Al crear un portal, la consola comprueba si los roles existentes son aptos para su asociación. Si no hay ninguno disponible, la consola puede crear y configurar un rol de servicio para usted. Para obtener más información, consulte [Cree un portal en SiteWise Monitor](#).

Temas

- [Búsqueda del rol de servicio de un portal \(consola\)](#)
- [Cree un rol de servicio de SiteWise monitoreo \(AWS IoT SiteWise consola\)](#)
- [Cree un rol de servicio de SiteWise supervisión \(consola de IAM\)](#)
- [Cambio del rol de servicio de un portal \(consola\)](#)

Búsqueda del rol de servicio de un portal (consola)

Siga los siguientes pasos para buscar el rol de servicio asociado a un portal de SiteWise Monitor.

Para buscar el rol de servicio de un portal

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación izquierdo, elija Portales.
3. Elija el portal para el que desea buscar el rol de servicio.

El rol asociado al portal aparece en Permisos, Rol de servicio.

Cree un rol de servicio de SiteWise monitoreo (AWS IoT SiteWise consola)

Al crear un portal de SiteWise Monitor, puede crear un rol de servicio para su portal. Para obtener más información, consulte [Cree un portal en SiteWise Monitor](#).

También puede crear un rol de servicio para un portal existente en la AWS IoT SiteWise consola. Esto sustituye el rol de servicio existente del portal.

Para crear un rol de servicio para un portal existente

1. Vaya a la [consola de AWS IoT SiteWise](#).

2. En el panel de navegación, elija Portales.
3. Elija el portal para el que desea crear un nuevo rol de servicio.
4. En Detalles del portal, elija Editar.
5. En Permisos, elija Crear y usar un nuevo rol de servicio de la lista.
6. Escriba un nombre para el nuevo rol.
7. Seleccione Save.

Cree un rol de servicio de SiteWise supervisión (consola de IAM)

Puede crear un rol de servicio a partir de la plantilla de rol de servicio de la consola de IAM. Esta plantilla de roles incluye la política [AWSIoTSiteWiseMonitorPortalAccess](#) administrada y especifica a SiteWise Monitor como una entidad de confianza.

Para crear un rol de servicio a partir de la plantilla de roles de servicio del portal

1. Vaya a la [consola de IAM](#).
2. Seleccione Roles en el panel de navegación.
3. Elija Create role (Crear rol).
4. En Elige un caso de uso, elige IoT SiteWise.
5. En Seleccionar su caso de uso, elija IoT SiteWise Monitor - Portal.
6. Elija Siguiente: Permisos.
7. Elija Siguiente: Etiquetas.
8. Elija Siguiente: Revisar.
9. Introduzca un Nombre de rol para el nuevo rol de servicio.
10. Elija Crear rol.

Cambio del rol de servicio de un portal (consola)

Utilice el siguiente procedimiento para elegir un rol de servicio de SiteWise supervisión diferente para un portal.

Para cambiar el rol de servicio de un portal

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, elija Portales.

3. Elija el portal para el que desea cambiar el rol de servicio.
4. En Detalles del portal, elija Editar.
5. En Permisos, elija Usar un rol existente.
6. Elija un rol existente para asociar a este portal.
7. Seleccione Save.

Administrar la función de servicio de SiteWise supervisión (CLI)

Puede utilizarla AWS CLI para las siguientes tareas de administración de funciones de servicio del portal:

Temas

- [Búsqueda del rol de servicio de un portal \(CLI\)](#)
- [Crear el rol de servicio de SiteWise monitoreo \(CLI\)](#)

Búsqueda del rol de servicio de un portal (CLI)

Para encontrar el rol de servicio asociado a un portal de SiteWise Monitor, ejecute el siguiente comando para ver una lista de todos los portales de la región actual.

```
aws iotsitewise list-portals
```

La operación devuelve una respuesta que contiene los resúmenes de su portal en el siguiente formato.

```
{
  "portalSummaries": [
    {
      "id": "a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE",
      "name": "WindFarmPortal",
      "description": "A portal that contains wind farm projects for Example Corp.",
      "roleArn": "arn:aws:iam::123456789012:role/service-role/role-name",
      "startUrl": "https://a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE.app.iotsitewise.aws",
      "creationDate": "2020-02-04T23:01:52.90248068Z",
      "lastUpdateDate": "2020-02-04T23:01:52.90248078Z"
    }
  ]
}
```

También puede usar la [DescribePortal](#) operación para encontrar la función de su portal si conoce el ID de su portal.

Crear el rol de servicio de SiteWise monitoreo (CLI)

Siga los siguientes pasos para crear un nuevo rol de servicio de SiteWise monitoreo.

Para crear un rol de servicio de SiteWise monitoreo

1. Cree un rol con una política de confianza que permita a SiteWise Monitor asumir el rol. En este ejemplo se crea un rol denominado **MySiteWiseMonitorPortalRole** a partir de una política de confianza almacenada en una cadena JSON.

Linux, macOS, or Unix

```
aws iam create-role --role-name MySiteWiseMonitorPortalRole --assume-role-policy-document '{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "monitor.iotsitewise.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}'
```

Windows command prompt

```
aws iam create-role --role-name MySiteWiseMonitorPortalRole --assume-role-policy-document "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":\"Allow\",\"Principal\":{\"Service\":\"monitor.iotsitewise.amazonaws.com\"},\"Action\":\"sts:AssumeRole\"}]}"
```

2. Copie el ARN del rol de los metadatos del rol en la salida. Al crear un portal, debe utilizar este ARN para asociar el rol al portal. Para obtener más información sobre la creación de un portal, consulte [CreatePortal](#) la referencia de la AWS IoT SiteWise API.
3. a. Para el SiteWise monitor (clásico): adjunte la `AWSIoTSiteWiseMonitorPortalAccess` política al rol o adjunte una política que defina permisos equivalentes.

```
aws iam attach-role-policy --role-name MySiteWiseMonitorPortalRole --policy-arn
arn:aws:iam::aws:policy/service-role/AWSIoTSiteWiseMonitorPortalAccess
```

- b. Para el SiteWise monitor (compatible con la IA): asocie la `IoTSiteWiseAIPortalAccessPolicy` política al rol o adjunte una política que defina permisos equivalentes. Por ejemplo, cree una política con permisos de acceso al portal. En el siguiente ejemplo, se crea una política denominada `MySiteWiseMonitorPortalAccess`.

```
aws iam create-policy \
  --policy-name MySiteWiseMonitorPortalAccess \
  --policy-document '{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iotsitewise:CreateProject",
        "iotsitewise:DescribePortal",
        "iotsitewise:ListProjects",
        "iotsitewise:DescribeProject",
        "iotsitewise:UpdateProject",
        "iotsitewise>DeleteProject",
        "iotsitewise:CreateDashboard",
        "iotsitewise:DescribeDashboard",
        "iotsitewise:UpdateDashboard",
        "iotsitewise>DeleteDashboard",
        "iotsitewise:ListDashboards",
        "iotsitewise:ListAssets",
        "iotsitewise:DescribeAsset",
        "iotsitewise:ListAssociatedAssets",
        "iotsitewise:ListAssetProperties",
        "iotsitewise:DescribeAssetProperty",
        "iotsitewise:GetAssetPropertyValue",
        "iotsitewise:GetAssetPropertyValueHistory",
        "iotsitewise:GetAssetPropertyAggregates",
        "iotsitewise:GetInterpolatedAssetPropertyValues",
        "iotsitewise:BatchGetAssetPropertyAggregates",
        "iotsitewise:BatchGetAssetPropertyValue",
        "iotsitewise:BatchGetAssetPropertyValueHistory",
        "iotsitewise:ListAssetRelationships",
```

```

        "iotsitewise:DescribeAssetModel",
        "iotsitewise:ListAssetModels",
        "iotsitewise:DescribeAssetCompositeModel",
        "iotsitewise:DescribeAssetModelCompositeModel",
        "iotsitewise:ListAssetModelProperties",
        "iotsitewise:ExecuteQuery",
        "iotsitewise:ListTimeSeries",
        "iotsitewise:DescribeTimeSeries",
        "iotsitewise:InvokeAssistant",
        "iotsitewise:DescribeDataset",
        "iotsitewise:ListDatasets",
        "iotevents:DescribeAlarmModel",
        "iotevents:ListTagsForResource",
        "iottwinmaker:ListWorkspaces",
        "iottwinmaker:ExecuteQuery",
        "iottwinmaker:GetWorkspace",
        "identitystore:DescribeUser"
    ],
    "Resource": "*"
}
]
}'

```

Para asociar un rol de servicio a un portal existente

1. Para recuperar los detalles existentes del portal, ejecute el siguiente comando. *portal-id* Sustitúyalo por el ID del portal.

```
aws iotsitewise describe-portal --portal-id portal-id
```

La operación devuelve una respuesta que contiene los detalles del portal en el siguiente formato.

```

{
  "portalId": "a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE",
  "portalArn": "arn:aws:iotsitewise:region:account-id:portal/a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE",
  "portalName": "WindFarmPortal",
  "portalDescription": "A portal that contains wind farm projects for Example Corp."
}

```

```

    "portalClientId": "E-1a2b3c4d5e6f_sn6tbqHVzLWVEXAMPLE",
    "portalStartUrl": "https://a1b2c3d4-5678-90ab-cdef-
aaaaaEXAMPLE.app.iotsitewise.aws",
    "portalContactEmail": "support@example.com",
    "portalStatus": {
      "state": "ACTIVE"
    },
    "portalCreationDate": "2020-04-29T23:01:52.90248068Z",
    "portalLastUpdateDate": "2020-04-29T00:28:26.103548287Z",
    "roleArn": "arn:aws:iam::123456789012:role/service-role/
AWSIoTSiteWiseMonitorServiceRole_1aEXAMPLE"
  }

```

- Para asociar un rol de servicio a un portal, ejecute el siguiente comando. Sustituya *role-arn* por el ARN del rol de servicio y sustituya los parámetros restantes por los valores existentes del portal.

```

aws iotsitewise update-portal \
  --portal-id portal-id \
  --role-arn role-arn \
  --portal-name portal-name \
  --portal-description portal-description \
  --portal-contact-email portal-contact-email

```

SiteWise Supervise las actualizaciones de AWSIoTSiteWiseMonitorServiceRole

Puede ver los detalles sobre las actualizaciones de AWSIoTSiteWiseMonitorServiceRolefor SiteWise Monitor, empezando por el momento en que este servicio comenzó a realizar el seguimiento de los cambios. Para recibir alertas automáticas sobre los cambios en esta página, suscríbese a la fuente RSS de la página del historial del AWS IoT SiteWise documento.

Cambio	Descripción	Fecha
AWSIoTSiteWiseMonitorPortal Access : política actualizada	AWS IoT SiteWise actualizó la política AWSIoTSiteWiseMonitorPortalAccess gestionada para la función de alarmas.	27 de mayo de 2021

Cambio	Descripción	Fecha
AWS IoT SiteWise comenzó a rastrear los cambios	AWS IoT SiteWise comenzó a rastrear los cambios de su función de servicio.	15 de diciembre de 2020

Configure los permisos para las alarmas de eventos en AWS IoT SiteWise

Cuando utiliza un modelo de AWS IoT Events alarma para supervisar la propiedad de un AWS IoT SiteWise activo, debe tener los siguientes permisos de IAM:

- Un rol AWS IoT Events de servicio que permite AWS IoT Events enviar datos a AWS IoT SiteWise. Para obtener más información, consulte [Administración de identidades y accesos de AWS IoT Events](#) en la Guía para desarrolladores de AWS IoT Events .
- Debe tener los siguientes permisos de AWS IoT SiteWise acción:
 - `iotsitewise:DescribeAssetModel`
 - `yiotsitewise:UpdateAssetModelPropertyRouting`. Estos permisos permiten AWS IoT SiteWise enviar los valores de las propiedades de los activos a los modelos de AWS IoT Events alarma.

Para obtener más información, consulte [Políticas basadas en recursos](#) en la Guía del usuario de IAM.

Permisos de acción necesarios

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones. El elemento `Action` de una política JSON describe las acciones que puede utilizar para conceder o denegar el acceso en una política.

Antes de definir un modelo de AWS IoT Events alarma, debe conceder los siguientes permisos que permiten AWS IoT SiteWise enviar los valores de las propiedades de los activos al modelo de alarma.

- `iotsitewise:DescribeAssetModel`, `iotsitewise:ListAssetModels` — Permite AWS IoT Events comprobar si existe una propiedad de un activo.

- `iotsitewise:UpdateAssetModelPropertyRouting`— Permite AWS IoT SiteWise crear automáticamente suscripciones que permiten AWS IoT SiteWise enviar datos a AWS IoT Events.

Para obtener más información sobre las acciones AWS IoT SiteWise compatibles, consulte [las acciones definidas AWS IoT SiteWise](#) en la Referencia de autorización de servicios.

Example Ejemplo 1 de política de permisos

La siguiente política permite AWS IoT SiteWise enviar los valores de las propiedades de los activos a cualquier modelo de AWS IoT Events alarma.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iotevents:CreateAlarmModel",
        "iotevents:UpdateAlarmModel"
      ],
      "Resource": "arn:aws:iotevents:us-east-1:123456789012:alarmModel/*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "iotsitewise:DescribeAssetModel",
        "iotsitewise>ListAssetModels",
        "iotsitewise:UpdateAssetModelPropertyRouting"
      ],
      "Resource": "arn:aws:iotsitewise:us-east-1:123456789012:asset-model/*"
    }
  ]
}
```

Example Ejemplo 2 de política de permisos

La siguiente política permite AWS IoT SiteWise enviar los valores de una propiedad de activo específica a un modelo de AWS IoT Events alarma específico.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iotevents:CreateAlarmModel",
        "iotevents:UpdateAlarmModel"
      ],
      "Resource": "arn:aws:iotevents:us-east-1:123456789012:alarmModel/*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "iotsitewise:DescribeAssetModel",
        "iotsitewise>ListAssetModels"
      ],
      "Resource": "arn:aws:iotsitewise:us-east-1:123456789012:asset-model/*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "iotsitewise:UpdateAssetModelPropertyRouting"
      ],
      "Resource": [
        "arn:aws:iotsitewise:us-east-1:123456789012:asset-model/12345678-90ab-cdef-1234-567890abcdef"
      ],
      "Condition": {
        "StringLike": {
          "iotsitewise:propertyId": "abcdef12-3456-7890-abcdef1234567890",
          "aws:ResourceTag/AlarmModel": "arn:aws:iotevents:us-east-1:123456789012:alarmModel/MyAlarmModel"
        }
      }
    }
  ]
}
```

```

    }
  }
]
}

```

ListInputRoutings Permiso (opcional)

Al actualizar o eliminar un modelo de activo, AWS IoT SiteWise puede comprobar si un modelo de alarma AWS IoT Events monitorea una propiedad de activo asociada a este modelo de activo. Esto le impide eliminar una propiedad de activo que una AWS IoT Events alarma esté utilizando actualmente. Para activar esta función AWS IoT SiteWise, debe tener el `iotevents:ListInputRoutings` permiso. Este permiso permite AWS IoT SiteWise realizar llamadas a la operación de [ListInputRoutings](#) API admitida por AWS IoT Events.

Note

Le recomendamos encarecidamente que añada el permiso `ListInputRoutings`.

Example Ejemplo de política de permisos

La siguiente política le permite actualizar y eliminar modelos de activos y utilizar la `ListInputRoutings` API en ellos AWS IoT SiteWise.

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iotsitewise:UpdateAssetModel",
        "iotsitewise>DeleteAssetModel",
        "iotevents:ListInputRoutings"
      ],
      "Resource": "arn:aws:iotsitewise:us-east-1:123456789012:asset-model/*"
    }
  ]
}

```

```
}
```

Permisos necesarios para SiteWise Monitor

Si desea utilizar la función de alarmas en los portales de SiteWise Monitor, debe actualizar la [función del servicio de SiteWise Monitor](#) con la siguiente política:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iotsitewise:DescribePortal",
        "iotsitewise:CreateProject",
        "iotsitewise:DescribeProject",
        "iotsitewise:UpdateProject",
        "iotsitewise>DeleteProject",
        "iotsitewise:ListProjects",
        "iotsitewise:BatchAssociateProjectAssets",
        "iotsitewise:BatchDisassociateProjectAssets",
        "iotsitewise:ListProjectAssets",
        "iotsitewise:CreateDashboard",
        "iotsitewise:DescribeDashboard",
        "iotsitewise:UpdateDashboard",
        "iotsitewise>DeleteDashboard",
        "iotsitewise:ListDashboards",
        "iotsitewise:CreateAccessPolicy",
        "iotsitewise:DescribeAccessPolicy",
        "iotsitewise:UpdateAccessPolicy",
        "iotsitewise>DeleteAccessPolicy",
        "iotsitewise:ListAccessPolicies",
        "iotsitewise:DescribeAsset",
        "iotsitewise:ListAssets",
        "iotsitewise:ListAssociatedAssets",
        "iotsitewise:DescribeAssetProperty",
        "iotsitewise:GetAssetPropertyValue",
        "iotsitewise:GetAssetPropertyValueHistory",
        "iotsitewise:GetAssetPropertyAggregates",
```

```

        "iotsitewise:BatchPutAssetPropertyValue",
        "iotsitewise:ListAssetRelationships",
        "iotsitewise:DescribeAssetModel",
        "iotsitewise:ListAssetModels",
        "iotsitewise:UpdateAssetModel",
        "iotsitewise:UpdateAssetModelPropertyRouting",
        "sso-directory:DescribeUsers",
        "sso-directory:DescribeUser",
        "iotevents:DescribeAlarmModel",
        "iotevents:ListTagsForResource"
    ],
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "iotevents:BatchAcknowledgeAlarm",
        "iotevents:BatchSnoozeAlarm",
        "iotevents:BatchEnableAlarm",
        "iotevents:BatchDisableAlarm"
    ],
    "Resource": "*",
    "Condition": {
        "Null": {
            "iotevents:keyValue": "false"
        }
    }
},
{
    "Effect": "Allow",
    "Action": [
        "iotevents>CreateAlarmModel",
        "iotevents:TagResource"
    ],
    "Resource": "*",
    "Condition": {
        "Null": {
            "aws:RequestTag/iotsitewisemonitor": "false"
        }
    }
},
{
    "Effect": "Allow",
    "Action": [

```

```

        "iotevents:UpdateAlarmModel",
        "iotevents>DeleteAlarmModel"
    ],
    "Resource": "*",
    "Condition": {
        "Null": {
            "aws:ResourceTag/iotsitewisemonitor": "false"
        }
    }
},
{
    "Effect": "Allow",
    "Action": [
        "iam:PassRole"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "iam:PassedToService": [
                "iotevents.amazonaws.com"
            ]
        }
    }
}
]
}

```

La prevención policial confusa entre servicios en AWS IoT SiteWise

El problema de la sustitución confusa es un problema de seguridad en el que una entidad que no tiene permiso para realizar una acción puede obligar a una entidad con más privilegios a realizar la acción. En AWS, la suplantación de identidad entre servicios puede provocar el confuso problema de un diputado. La suplantación entre servicios puede producirse cuando un servicio (el servicio que lleva a cabo las llamadas) llama a otro servicio (el servicio al que se llama). El servicio que lleva a cabo las llamadas se puede manipular para utilizar sus permisos a fin de actuar en función de los recursos de otro cliente de una manera en la que no debe tener permiso para acceder. Para evitarlo, AWS proporciona herramientas que le ayudan a proteger sus datos para todos los servicios con entidades principales de servicio a las que se les ha dado acceso a los recursos de su cuenta.

Se recomienda utilizar las claves de contexto de condición [aws:SourceAccount](#) global [aws:SourceArny](#) las claves de contexto en las políticas de recursos para limitar los permisos que

se AWS IoT SiteWise otorgan a otro servicio al recurso. Si el valor de `aws:SourceArn` no contiene el ID de cuenta, como un nombre de recurso de Amazon (ARN) de bucket de Amazon S3, debe utilizar ambas claves de contexto de condición global para limitar los permisos. Si utiliza claves de contexto de condición global y el valor de `aws:SourceArn` contiene el ID de cuenta, el valor de `aws:SourceAccount` y la cuenta en el valor de `aws:SourceArn` deben utilizar el mismo ID de cuenta cuando se utiliza en la misma instrucción de política.

- Utiliza `aws:SourceArn` si desea que solo se asocie un recurso al acceso entre servicios.
- Utiliza `aws:SourceAccount` si quiere permitir que cualquier recurso de esa cuenta se asocie al uso entre servicios.

El valor de `aws:SourceArn` debe ser el recurso AWS IoT SiteWise del cliente asociado a la `sts:AssumeRole` solicitud.

La forma más eficaz de protegerse contra el problema de la sustitución confusa es utilizar la clave de contexto de condición global de `aws:SourceArn` con el ARN completo del recurso. Si no conoce el ARN completo del recurso o si especifica varios recursos, utilice la clave de condición de contexto global `aws:SourceArn` con comodines (*) para las partes desconocidas del ARN. Por ejemplo, `arn:aws:servicename:*:123456789012:*`.

Example– Prevención del suplente confuso

El siguiente ejemplo muestra cómo puede utilizar las claves de contexto de condición `aws:SourceAccount` global `aws:SourceArn` y las claves contextuales AWS IoT SiteWise para evitar el confuso problema de los diputados.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Sid": "ConfusedDeputyPreventionExamplePolicy",
    "Effect": "Allow",
    "Principal": {
      "Service": "iotsitewise.amazonaws.com"
    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "ArnLike": {
```

```
    "aws:SourceArn": "arn:aws:iotsitewise:*:123456789012:*"  
  },  
  "StringEquals": {  
    "aws:SourceAccount": "123456789012"  
  }  
}  
}
```

Solucione los problemas de AWS IoT SiteWise identidad y acceso

Utilice la siguiente información para ayudarle a diagnosticar y solucionar los problemas más comunes que pueden surgir al trabajar con AWS IoT SiteWise y AWS Identity and Access Management (IAM).

Temas

- [No estoy autorizado a realizar ninguna acción en AWS IoT SiteWise](#)
- [No tengo autorización para realizar iam:PassRole](#)
- [Quiero permitir que personas ajenas a mi AWS cuenta accedan a mis AWS IoT SiteWise recursos](#)

No estoy autorizado a realizar ninguna acción en AWS IoT SiteWise

Si Consola de administración de AWS le indica que no está autorizado a realizar una acción, debe ponerse en contacto con su administrador para obtener ayuda. El administrador es la persona que le facilitó el nombre de usuario y la contraseña.

El siguiente ejemplo de error se produce cuando el usuario de IAM mateojackson intenta utilizar la consola para ver los detalles de un activo, pero no dispone de los permisos `iotsitewise:DescribeAsset`.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:  
iotsitewise:DescribeAsset on resource: a1b2c3d4-5678-90ab-cdef-22222EXAMPLE
```

En este caso, Mateo pide a su administrador que actualice sus políticas de forma que pueda obtener acceso al activo del activo con el ID `a1b2c3d4-5678-90ab-cdef-22222EXAMPLE` mediante la acción `iotsitewise:DescribeAsset`.

No tengo autorización para realizar **iam:PassRole**

Si recibe un error que indica que no tiene autorización para realizar la acción `iam:PassRole`, las políticas deben actualizarse a fin de permitirle pasar un rol a AWS IoT SiteWise.

Algunas Servicios de AWS permiten transferir una función existente a ese servicio en lugar de crear una nueva función de servicio o una función vinculada a un servicio. Para ello, debe tener permisos para transferir la función al servicio.

En el siguiente ejemplo, el error se produce cuando un usuario de IAM denominado `marymajor` intenta utilizar la consola para realizar una acción en AWS IoT SiteWise. Sin embargo, la acción requiere que el servicio cuente con permisos que otorguen un rol de servicio. Mary no tiene permisos para transferir el rol al servicio.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

En este caso, las políticas de Mary se deben actualizar para permitirle realizar la acción `iam:PassRole`.

Si necesita ayuda, póngase en contacto con su AWS administrador. El administrador es la persona que le proporcionó las credenciales de inicio de sesión.

Quiero permitir que personas ajenas a mi AWS cuenta accedan a mis AWS IoT SiteWise recursos

Se puede crear un rol que los usuarios de otras cuentas o las personas externas a la organización puedan utilizar para acceder a sus recursos. Se puede especificar una persona de confianza para que asuma el rol. En el caso de los servicios que admiten políticas basadas en recursos o listas de control de acceso (ACLs), puedes usar esas políticas para permitir que las personas accedan a tus recursos.

Para obtener más información, consulte lo siguiente:

- Para saber si AWS IoT SiteWise es compatible con estas funciones, consulte [¿Cómo AWS IoT SiteWise funciona con IAM](#)
- Para obtener información sobre cómo proporcionar acceso a los recursos de su Cuentas de AWS propiedad, consulte [Proporcionar acceso a un usuario de IAM en otro usuario de su propiedad Cuenta de AWS en](#) la Guía del usuario de IAM.

- Para obtener información sobre cómo proporcionar acceso a tus recursos a terceros Cuentas de AWS, consulta [Cómo proporcionar acceso a recursos que Cuentas de AWS son propiedad de terceros](#) en la Guía del usuario de IAM.
- Para obtener información sobre cómo proporcionar acceso mediante una federación de identidades, consulte [Proporcionar acceso a usuarios autenticados externamente \(identidad federada\)](#) en la Guía del usuario de IAM.
- Para conocer sobre la diferencia entre las políticas basadas en roles y en recursos para el acceso entre cuentas, consulte [Acceso a recursos entre cuentas en IAM](#) en la Guía del usuario de IAM.

Validación de conformidad para AWS IoT SiteWise

AWS IoT SiteWise no está dentro del ámbito de ningún programa de AWS cumplimiento.

Para obtener una lista de AWS los servicios incluidos en el ámbito de los programas de cumplimiento específicos, consulte los [AWS servicios incluidos en el ámbito de aplicación por programa de cumplimiento](#) y . Para obtener información general, consulte Programas de [AWS cumplimiento > Programas AWS](#) .

Puede descargar informes de auditoría de terceros utilizando AWS Artifact. Para obtener más información, consulte [Descargar informes en AWS Artifact](#) .

Su responsabilidad de conformidad al AWS IoT SiteWise utilizarlos viene determinada por la confidencialidad de sus datos, los objetivos de cumplimiento de su empresa y las leyes y reglamentos aplicables. AWS proporciona los siguientes recursos para ayudar con el cumplimiento:

- [Guías de inicio rápido](#) sobre sobre seguridad y cumplimiento: estas guías de implementación analizan las consideraciones arquitectónicas y proporcionan los pasos para implementar entornos básicos centrados en la seguridad y el cumplimiento. AWS
- Documento técnico sobre [cómo diseñar una arquitectura basada en la seguridad y el cumplimiento de la HIPAA: en este documento técnico](#) se describe cómo pueden utilizar las empresas para crear aplicaciones que cumplan con la HIPAA. AWS
- [AWS Recursos de cumplimiento Recursos](#) de de trabajo y guías puede aplicarse a su sector y ubicación.
- [Evaluación de los recursos con las reglas](#) de la Guía para AWS Config desarrolladores: el AWS Config servicio evalúa en qué medida las configuraciones de los recursos cumplen con las prácticas internas, las directrices del sector y las normas.

- [AWS Security Hub CSPM](#)— Este AWS servicio proporciona una visión integral del estado de su seguridad AWS que le ayuda a comprobar su conformidad con los estándares y las mejores prácticas del sector de la seguridad.
- [Diez reglas de oro de seguridad para las soluciones de IoT industrial](#): esta entrada de blog presenta diez reglas de oro que ayudan a proteger sus sistemas de control industrial (ICS), Internet de las cosas (IIoT) industrial y entornos de nube.
- [Mejores prácticas de seguridad para la fabricación de OT](#): este documento técnico describe las mejores prácticas de seguridad para diseñar, implementar y diseñar estas cargas de trabajo de fabricación híbrida locales para la nube. AWS

Resiliencia en AWS IoT SiteWise

La infraestructura AWS global se basa en AWS regiones y zonas de disponibilidad. AWS Las regiones proporcionan varias zonas de disponibilidad aisladas y separadas físicamente, que están conectadas mediante redes de baja latencia, alto rendimiento y alta redundancia. Con las zonas de disponibilidad, puede diseñar y utilizar aplicaciones y bases de datos que realizan una conmutación por error automática entre las zonas sin interrupciones. Las zonas de disponibilidad tienen una mayor disponibilidad, tolerancia a errores y escalabilidad que las infraestructuras tradicionales de uno o varios centros de datos.

AWS IoT SiteWise está totalmente gestionado y utiliza AWS servicios duraderos y de alta disponibilidad, como Amazon S3 y Amazon EC2. Para garantizar la disponibilidad en caso de que se produzca una interrupción en la zona de disponibilidad, AWS IoT SiteWise opera en varias zonas de disponibilidad.

Para obtener más información sobre AWS las regiones y las zonas de disponibilidad, consulte [Infraestructura AWS global](#).

Además de la infraestructura AWS global, AWS IoT SiteWise ofrece varias funciones para ayudarlo a satisfacer sus necesidades de respaldo y resiliencia de datos:

- Puede publicar actualizaciones del valor de las propiedades AWS IoT Core mediante mensajes MQTT y, a continuación, configurar reglas para actuar en función de esos datos. Con esta función, puede realizar copias de seguridad de los datos en otros AWS servicios, como Amazon S3 y Amazon DynamoDB. Para obtener más información, consulte [Interactúa con otros AWS servicios y Exportación de datos a Amazon S3 con notificaciones sobre propiedades de activos](#).

- Puede utilizarla AWS IoT SiteWise Get* APIs para recuperar y hacer copias de seguridad de los datos históricos de propiedades de los activos. Para obtener más información, consulte [Consulte los valores históricos de las propiedades de los activos en AWS IoT SiteWise](#).
- Puede utilizarlos AWS IoT SiteWise Describe* APIs para recuperar las definiciones de sus recursos, como activos y modelos. Puede realizar una copia de seguridad de estas definiciones y luego utilizarlas para volver a crear los recursos. Para obtener más información, consulte la [Referencia de la API de AWS IoT SiteWise](#).

Seguridad de la infraestructura en AWS IoT SiteWise

Como servicio gestionado, AWS IoT SiteWise está protegido por la seguridad de la red AWS global. Para obtener información sobre los servicios AWS de seguridad y cómo se AWS protege la infraestructura, consulte [Seguridad AWS en la nube](#). Para diseñar su AWS entorno utilizando las mejores prácticas de seguridad de la infraestructura, consulte [Protección de infraestructuras en un marco](#) de buena AWS arquitectura basado en el pilar de la seguridad.

Utiliza las llamadas a la API AWS publicadas para acceder a AWS IoT SiteWise través de la red. Los clientes deben admitir lo siguiente:

- Seguridad de la capa de transporte (TLS). Exigimos TLS 1.2 y recomendamos TLS 1.3.
- Conjuntos de cifrado con confidencialidad directa total (PFS) como DHE (Ephemeral Diffie-Hellman) o ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). La mayoría de los sistemas modernos como Java 7 y posteriores son compatibles con estos modos.

SiteWise Las pasarelas perimetrales, que se ejecutan en AWS IoT Greengrass ella, utilizan certificados X.509 y claves criptográficas para conectarse a la nube y autenticarse en ella. AWS Para obtener más información, consulte [Autenticación y autorización de dispositivos para AWS IoT Greengrass](#) en la Guía para desarrolladores de AWS IoT Greengrass Version 1.

Análisis de configuración y vulnerabilidad en AWS IoT SiteWise

Las flotas de IoT pueden constar de un gran número de dispositivos que tienen diversas funcionalidades, son de larga duración y están distribuidos geográficamente. Estas características hacen que la configuración de la flota sea compleja y propensa a errores. Dado que los dispositivos suelen tener una capacidad limitada de procesamiento, memoria y almacenamiento, no siempre son compatibles con el cifrado y otras medidas de seguridad. Además, los dispositivos a menudo usan

software con vulnerabilidades conocidas. Estos factores hacen que las flotas de IoT sean un objetivo atractivo para los piratas informáticos y dificultan la protección continuada de la flota de dispositivos.

AWS IoT Device Defender aborda estos desafíos proporcionando herramientas para identificar los problemas de seguridad y las desviaciones de las mejores prácticas. Úselo AWS IoT Device Defender para analizar, auditar y monitorear los dispositivos conectados a fin de detectar comportamientos anormales y mitigar los riesgos de seguridad. AWS IoT Device Defender puede auditar las flotas de dispositivos para garantizar que cumplen las mejores prácticas de seguridad y detectar comportamientos anormales en los dispositivos. Esto permite aplicar políticas de seguridad coherentes en toda su flota de AWS IoT dispositivos y responder rápidamente cuando los dispositivos se ven comprometidos. Para obtener más información, consulte [¿Qué es AWS IoT Device Defender?](#) en la Guía para desarrolladores de AWS IoT Device Defender .

Si utiliza las puertas de enlace SiteWise Edge para introducir datos en el servicio, es su responsabilidad configurar y mantener el entorno de su puerta de enlace SiteWise Edge. Esta responsabilidad incluye la actualización a las versiones más recientes del software del sistema, AWS IoT Greengrass el software y el conector de la AWS IoT SiteWise puerta de enlace SiteWise Edge. Para obtener más información, consulte [Configurar el AWS IoT Greengrass núcleo](#) en la Guía para AWS IoT Greengrass Version 1 desarrolladores y [Administre las puertas de enlace SiteWise Edge](#).

Puntos finales de VPC para AWS IoT SiteWise

Un punto final de VPC de interfaz establece una conexión privada entre su nube privada virtual (VPC) y AWS IoT SiteWise [AWS PrivateLink](#) potencia los puntos finales de la interfaz, lo que permite el acceso privado a AWS IoT SiteWise las operaciones de la API. AWS IoT SiteWise admite ambos tipos IPv4 y IPv6 (doble pila) a través de los puntos finales de su interfaz. Puede evitar la necesidad de una puerta de enlace de Internet, un dispositivo NAT, una conexión VPN o AWS Direct Connect. Las instancias de su VPC no necesitan direcciones IP públicas para comunicarse con las operaciones de la API de AWS IoT SiteWise . El tráfico entre su VPC y AWS IoT SiteWise no sale de la AWS red.

Cada punto de conexión de la interfaz está representado por una o más [interfaces de redes elásticas](#) en las subredes.

Antes de configurar un punto de enlace de VPC de interfaz AWS IoT SiteWise, consulte la sección [Acceder a un AWS servicio mediante un punto de enlace de VPC](#) de interfaz en la Guía.AWS PrivateLink

Operaciones de API para puntos finales de VPC en AWS IoT SiteWise

AWS IoT SiteWise admite realizar llamadas a las siguientes operaciones de AWS IoT SiteWise API desde su VPC:

- Para todas las operaciones de la API del plano de datos, utilice el siguiente punto de conexión: Sustituya *region* por su Región de AWS

```
data.iotsitewise.region.amazonaws.com
```

Las operaciones de la API del plano de datos incluyen lo siguiente:


- [BatchGetAssetPropertyValue](#)
 - [BatchGetAssetPropertyValueHistory](#)
 - [BatchPutAssetPropertyValue](#)
 - [GetAssetPropertyAggregates](#)
 - [GetAssetPropertyValue](#)
 - [GetAssetPropertyValueHistory](#)
 - [GetInterpolatedAssetPropertyValues](#)
- Para las operaciones de la API del plano de control que utiliza para gestionar los modelos de activos, los activos, las pasarelas de SiteWise Edge, las etiquetas y las configuraciones de cuentas, utilice el siguiente punto final. Sustituya *region* por su Región de AWS.

```
api.iotsitewise.region.amazonaws.com
```

Las operaciones de la API del plano de control admitidas incluyen lo siguiente:

- [AssociateAssets](#)
- [CreateAsset](#)
- [CreateAssetModel](#)
- [DeleteAsset](#)
- [DeleteAssetModel](#)
- [DeleteDashboard](#)
- [DescribeAsset](#)
- [DescribeAssetModel](#)
- [DescribeAssetProperty](#)

- [DescribeDashboard](#)
- [DescribeLoggingOptions](#)
- [DisassociateAssets](#)
- [ListAssetModels](#)
- [ListAssetRelationships](#)
- [ListAssets](#)
- [ListAssociatedAssets](#)
- [PutLoggingOptions](#)
- [UpdateAsset](#)
- [UpdateAssetModel](#)
- [UpdateAssetProperty](#)
- [CreateGateway](#)
- [DeleteGateway](#)
- [DescribeDefaultEncryptionConfiguration](#)
- [DescribeGateway](#)
- [DescribeGatewayCapabilityConfiguration](#)
- [DescribeStorageConfiguration](#)
- [ListGateways](#)
- [ListTagsForResource](#)
- [UpdateGateway](#)
- [UpdateGatewayCapabilityConfiguration](#)
- [PutDefaultEncryptionConfiguration](#)
- [PutStorageConfiguration](#)
- [TagResource](#)
- [UntagResource](#)

 Note

El punto final de la interfaz de VPC para las operaciones de la API del plano de control actualmente no admite realizar llamadas a las siguientes operaciones de la API de

- [BatchAssociateProjectAssets](#)

- [BatchDisassociateProjectAssets](#)
- [CreateAccessPolicy](#)
- [CreateDashboard](#)
- [CreatePortal](#)
- [CreateProject](#)
- [DeleteAccessPolicy](#)
- [DeletePortal](#)
- [DeleteProject](#)
- [DescribeAccessPolicy](#)
- [DescribePortal](#)
- [DescribeProject](#)
- [ListAccessPolicies](#)
- [ListDashboards](#)
- [ListPortals](#)
- [ListProjects](#)
- [ListProjectAssets](#)
- [UpdateAccessPolicy](#)
- [UpdateDashboard](#)
- [UpdatePortal](#)
- [UpdateProject](#)

Cree un punto final de VPC de interfaz para AWS IoT SiteWise

Para crear un punto de enlace de VPC para el AWS IoT SiteWise servicio, utilice la consola de Amazon VPC o (). AWS Command Line Interface AWS CLI Para obtener más información, consulte [Acceder a un AWS servicio mediante un punto final de VPC de interfaz](#) en la AWS PrivateLink Guía.

Cree un punto de enlace de VPC con uno de los siguientes nombres de servicio: AWS IoT SiteWise

- Para las operaciones de la API del plano de datos, utilice el siguiente nombre de servicio:

- Para las operaciones de la API del plano de control, utilice el siguiente nombre de servicio:

```
com.amazonaws.region.iotsitewise.api
```

Acceso a AWS IoT SiteWise través de un punto final de VPC de interfaz

Al crear un punto final de interfaz, generamos nombres de host DNS específicos del punto final con los que puede comunicarse. AWS IoT SiteWise La opción de DNS privado está habilitada de forma predeterminada. Para obtener más información, consulte [Uso de zonas alojadas privadas](#) en la Guía del usuario de Amazon VPC.

Si habilitas el DNS privado para el punto final, puedes realizar solicitudes de API a AWS IoT SiteWise través de uno de los siguientes puntos finales de VPC.

- Para las operaciones de la API del plano de datos, usa el siguiente punto final: Sustitúyalo por el tuyo *region*. Región de AWS

```
data.iotsitewise.region.amazonaws.com
```

- Para las operaciones de la API del plano de control, utilice el siguiente punto final: Reemplace por *region* el suyo Región de AWS.

```
api.iotsitewise.region.amazonaws.com
```

Si inhabilita el DNS privado para el punto final, debe hacer lo siguiente para acceder a AWS IoT SiteWise través del punto final:

1. Especifique la url del punto de conexión de VPC en las solicitudes de la API.

- Para las operaciones de la API del plano de datos, utilice la siguiente url de punto de conexión. Sustituya *vpc-endpoint-id* y *region* por su ID de punto final de VPC y su región.

```
vpc-endpoint-id.data.iotsitewise.region.vpce.amazonaws.com
```

- Para las operaciones de la API del plano de control, utilice la siguiente url de punto de conexión. Sustituya *vpc-endpoint-id* y *region* por su ID de punto final de VPC y su región.

```
vpc-endpoint-id.api.iotsitewise.region.vpce.amazonaws.com
```

2. Inhabilite la inyección de prefijos de host. Al llamar a AWS SDKs cada operación de API, coloque varios prefijos de host antes del punto final del servicio. AWS CLI Esta función hace que se AWS SDKs produzcan AWS CLI y URLs que no son válidos AWS IoT SiteWise cuando se especifica un punto final de VPC.

 Important

No puede deshabilitar la inyección de prefijos de host en el AWS CLI o en. Herramientas de AWS para PowerShell Esto significa que si inhabilitas el DNS privado, no podrás usar estas herramientas para acceder a AWS IoT SiteWise través del punto final de la VPC. Habilita el DNS privado para usar el AWS CLI o Herramientas de AWS para PowerShell para acceder a AWS IoT SiteWise través del punto final.

Para obtener más información sobre cómo deshabilitar la inyección de prefijos de host en el AWS SDKs, consulta las siguientes secciones de la documentación de cada SDK:

- [AWS SDK para C++](#)
- [AWS SDK para Go](#)
- [AWS SDK para Go v2](#)
- [AWS SDK para Java](#)
- [AWS SDK for Java 2.x](#)
- [AWS SDK para JavaScript](#)
- [AWS SDK para .NET](#)
- [AWS SDK para PHP](#)
- [AWS SDK para Python \(Boto3\)](#)
- [AWS SDK para Ruby](#)

Para obtener más información, consulte [Acceder a un AWS servicio mediante un punto final de VPC de interfaz](#) en la AWS PrivateLink Guía.

Cree una política de puntos de conexión de VPC para AWS IoT SiteWise

Puede asociar una política de punto de conexión con su punto de conexión de VPC que controla el acceso a AWS IoT SiteWise. La política especifica la siguiente información:

- La entidad principal que puede realizar operaciones.
- Las operaciones que se pueden realizar.
- Los recursos con los que se pueden realizar las operaciones.

Para obtener más información, consulte [Controlar el acceso a los puntos de conexión de la VPC](#) mediante políticas de puntos de conexión en la Guía de usuario de Amazon VPC.

Ejemplo: política de puntos finales de VPC para acciones AWS IoT SiteWise

El siguiente es un ejemplo de una política de puntos finales para AWS IoT SiteWise. Cuando se adjunta a un punto final, esta política otorga acceso a las AWS IoT SiteWise acciones enumeradas al usuario de la AWS cuenta *iotsitewiseadmin 123456789012* en el activo especificado.

```
{
  "Statement": [
    {
      "Action": [
        "iotsitewise:CreateAsset",
        "iotsitewise:ListGateways",
        "iotsitewise:ListTagsForResource"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:iotsitewise:us-west-2:123456789012:asset/a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
      "Principal": {
        "AWS": [
          "123456789012:user/iotsitewiseadmin"
        ]
      }
    }
  ]
}
```

Mejores prácticas de seguridad para AWS IoT SiteWise

En este tema se incluyen las prácticas recomendadas de seguridad para AWS IoT SiteWise.

Usar credenciales de autenticación en los servidores OPC UA

Exija las credenciales de autenticación para conectarse a sus servidores OPC UA. Consulte la documentación de los servidores para hacerlo. A continuación, para permitir que la puerta de enlace SiteWise Edge se conecte a los servidores OPC UA, añada los secretos de autenticación del servidor a la puerta de enlace SiteWise Edge. Para obtener más información, consulte [Configurar la autenticación de fuentes de datos para SiteWise Edge](#).

Usar modos de comunicación cifrados para los servidores OPC UA

Elija un modo de seguridad de mensajes cifrados y no obsoleto al configurar las fuentes OPC UA para su puerta de enlace Edge. SiteWise Esto ayuda a proteger sus datos industriales cuando se transfieren de sus servidores OPC UA a la puerta de enlace Edge. SiteWise Para obtener más información, consulte [Datos en tránsito a través de la red local](#) y [Configure una fuente OPC UA en Edge SiteWise](#).

Mantener los componentes actualizados

Si utiliza las puertas de enlace SiteWise Edge para introducir datos en el servicio, es su responsabilidad configurar y mantener el entorno de su puerta de enlace Edge. SiteWise Esta responsabilidad incluye la actualización del software del sistema de la puerta de enlace, del software de AWS IoT Greengrass y de los conectores a sus últimas versiones.

Note

El conector AWS IoT SiteWise Edge almacena información confidencial en su sistema de archivos. Estos secretos controlan quién puede ver los datos almacenados en caché en su puerta de enlace SiteWise Edge. Se recomienda encarecidamente activar el cifrado del disco o del sistema de archivos en el sistema en el que se ejecuta la puerta de enlace Edge. SiteWise

Para obtener información sobre cómo actualizar los componentes de la AWS IoT SiteWise consola, consulte. [Cambie la versión de los paquetes de componentes de SiteWise Edge Gateway](#)

Cifre el sistema de archivos de su puerta de enlace SiteWise Edge

Cifre y proteja su puerta de enlace SiteWise Edge para que sus datos industriales estén seguros a medida que se mueven a través de la puerta de enlace SiteWise Edge. Si su puerta de

enlace SiteWise Edge tiene un módulo de seguridad de hardware, puede configurarlo AWS IoT Greengrass para proteger su puerta de enlace SiteWise Edge. Para obtener más información, consulte [Integración de la seguridad del hardware](#) en la Guía para desarrolladores de AWS IoT Greengrass Version 1 . De lo contrario, consulte la documentación del sistema operativo para obtener información sobre cómo cifrar y proteger el sistema de archivos.

Acceso seguro a la configuración de su periferia

No compartas la contraseña de la aplicación de la consola perimetral ni la contraseña de la aplicación SiteWise Monitor. No coloque esta contraseña en lugares donde cualquiera pueda verla. Implemente una política de rotación de contraseñas saludable configurando una caducidad apropiada para su contraseña.

Protección de los datos en Siemens Industrial Edge Management

Los datos del dispositivo que decida compartir con AWS IoT SiteWise Edge se determinan en los temas Siemens IEM Databus de configuración. Al elegir los temas para compartir con SiteWise Edge, compartes datos a nivel de tema. AWS IoT SiteWise Siemens Industrial Edge Marketplace es un mercado independiente, separado de AWS. Para proteger sus datos compartidos, la aplicación SiteWise Edge no se ejecutará a menos que la utilice Siemens Secured Storage. Para obtener más información, consulte las [Almacenamiento seguro](#) en la documentación de Siemens.

Otorgue a los usuarios de SiteWise Monitor los permisos mínimos posibles

Siga el principio de privilegio mínimo utilizando el conjunto mínimo de permisos de la política de acceso para los usuarios de su portal.

- Al crear un portal, defina un rol que permita el conjunto mínimo de activos necesarios para ese portal. Para obtener más información, consulte [Utilice los roles de servicio para AWS IoT SiteWise Monitor](#).
- Cuando los administradores del portal y usted creen y compartan proyectos, utilice el conjunto mínimo de activos necesarios para ese proyecto.
- Cuando una identidad ya no necesite acceder a un portal o proyecto, elimínela de ese recurso. Si esa identidad ya no es aplicable a su organización, elimínela de su almacén de identidades.

La práctica recomendada de principio mínimo también se aplica a los roles de IAM. Para obtener más información, consulte [Prácticas recomendadas relativas a políticas](#).

No exponer información confidencial

Debe evitar el registro de credenciales y otra información confidencial, como la información de identificación personal (PII). Le recomendamos que implemente las siguientes medidas de seguridad, aunque el acceso a los registros locales en una puerta de enlace de SiteWise Edge requiera privilegios de root y el acceso a CloudWatch los registros requiera permisos de IAM.

- No utilice información confidencial en nombres, descripciones o propiedades de sus activos o modelos.
- No utilices información confidencial en los nombres de las fuentes o la puerta de enlace de SiteWise Edge.
- No utilice información confidencial en nombres o descripciones de sus portales, proyectos o paneles.

Siga las prácticas recomendadas de AWS IoT Greengrass seguridad

Siga las prácticas recomendadas de AWS IoT Greengrass seguridad para su puerta de enlace SiteWise Edge. Para obtener más información, consulte [Prácticas recomendadas de seguridad](#) en la Guía para desarrolladores de AWS IoT Greengrass Version 1 .

Véase también

- [Prácticas recomendadas de seguridad](#) en la Guía para desarrolladores de AWS IoT
- [Diez reglas de oro de seguridad para soluciones IoT industrial](#)

Iniciar sesión y monitorear AWS IoT SiteWise

La supervisión es una parte importante del mantenimiento de la confiabilidad, la disponibilidad y el rendimiento de AWS IoT SiteWise sus demás AWS soluciones. AWS IoT SiteWise admite las siguientes herramientas de supervisión para supervisar el servicio, informar cuando algo va mal y tomar medidas automáticas cuando sea necesario:

- Amazon CloudWatch supervisa tus AWS recursos y las aplicaciones en las que ejecutas AWS en tiempo real. Recopile métricas y realice un seguimiento de ellas, cree paneles personalizados y defina alarmas que realizan advertencias o acciones cuando una métrica alcanza el umbral determinado. Por ejemplo, puedes CloudWatch hacer un seguimiento del uso de la CPU u otras métricas de tus EC2 instancias de Amazon y lanzar automáticamente nuevas instancias cuando sea necesario. Para obtener más información, consulta la [Guía del CloudWatch usuario de Amazon](#).
- Amazon CloudWatch Logs supervisa, almacena y accede a sus archivos de registro desde las puertas de enlace de SiteWise Edge y otras fuentes. CloudTrail CloudWatch Los registros pueden monitorear la información de los archivos de registro y notificarle cuando se alcanzan ciertos umbrales. También se pueden archivar los datos del registro en un almacenamiento de larga duración. Para obtener más información, consulta la [Guía del usuario CloudWatch de Amazon Logs](#).
- AWS CloudTrail captura las llamadas a la API y los eventos relacionados realizados por su AWS cuenta o en su nombre. A continuación, CloudTrail entrega los archivos de registro a un bucket de Amazon S3 que especifique. Puede identificar qué usuarios y cuentas llamaron AWS, la dirección IP de origen desde la que se realizaron las llamadas y cuándo se produjeron. Para obtener más información, consulte la [AWS CloudTrail Guía del usuario de](#) .

Temas

- [Supervise con Amazon CloudWatch Logs](#)
- [Supervise los registros de SiteWise Edge Gateway](#)
- [Supervisa AWS IoT SiteWise con las CloudWatch métricas de Amazon](#)
- [Registra las llamadas a la AWS IoT SiteWise API con AWS CloudTrail](#)

Supervise con Amazon CloudWatch Logs

AWS IoT SiteWise Configúrelo para registrar la información en CloudWatch los registros a fin de supervisar y solucionar los problemas del servicio.

Cuando utilizas la AWS IoT SiteWise consola, AWS IoT SiteWise crea un rol vinculado al servicio que permite al servicio registrar la información en tu nombre. Si no usa la AWS IoT SiteWise consola, debe crear manualmente un rol vinculado al servicio para recibir los registros. Para obtener más información, consulte [Cree un rol vinculado a un servicio para AWS IoT SiteWise](#).

Debe tener una política de recursos que permita colocar AWS IoT SiteWise los eventos de registro en CloudWatch las transmisiones. Para crear y actualizar una política de recursos para CloudWatch los registros, ejecute el siguiente comando. *logging-policy-name* Sustitúyalo por el nombre de la política que se va a crear.

```
aws logs put-resource-policy --policy-name logging-policy-name --policy-document "{ \"Version\": \"2012-10-17\", \"Statement\": [ { \"Sid\": \"IoTSiteWiseToCloudWatchLogs\", \"Effect\": \"Allow\", \"Principal\": { \"Service\": [ \"iotsitewise.amazonaws.com\" ] }, \"Action\": \"logs:PutLogEvents\", \"Resource\": \"*\" } ] }"
```

CloudWatch Los registros también admiten [las claves de contexto aws: SourceArn y aws: SourceAccount](#) condition. Estas claves de contexto de condición son opcionales.

Para crear o actualizar una política de recursos que AWS IoT SiteWise permita colocar únicamente los registros asociados al AWS IoT SiteWise recurso especificado en las CloudWatch transmisiones, ejecute el comando y haga lo siguiente:

- *logging-policy-name* Sustitúyalo por el nombre de la política que se va a crear.
- *source-ARN* Sustitúyalo por el ARN de tu AWS IoT SiteWise recurso, como un modelo de activo o un activo. Para encontrar el ARN de cada tipo de AWS IoT SiteWise recurso, consulte [Tipos de recursos definidos AWS IoT SiteWise en la Referencia](#) de autorización de servicio.
- *account-ID* Sustitúyalo por el ID de AWS cuenta asociado al AWS IoT SiteWise recurso especificado.

```
aws logs put-resource-policy --policy-name logging-policy-name --policy-document "{ \"Version\": \"2012-10-17\", \"Statement\": [ { \"Sid\": \"IoTSiteWiseToCloudWatchLogs\", \"Effect\": \"Allow\", \"Principal\": { \"Service
```

```
\": [ \"iotsitewise.amazonaws.com\" ] }, \"Action\": \"logs:PutLogEvents\", \"Resource\": \"*\", \"Condition\": { \"StringLike\": { \"aws:SourceArn\": [ \"source-ARN\" ], \"aws:SourceAccount\": [ \"account-ID\" ] } } }
```

De forma predeterminada, AWS IoT SiteWise no registra la información en los CloudWatch registros. Para activar el registro, elige un nivel de registro que no sea Desactivado (OFF). AWS IoT SiteWise admite los siguientes niveles de registro:

- OFF: el registro está desactivado.
- ERROR: se registran los errores.
- INFO: se registran los errores y los mensajes informativos.

Puede configurar las puertas de enlace SiteWise Edge para que registren la información en CloudWatch Logs Through AWS IoT Greengrass. Para obtener más información, consulte [Supervise los registros de SiteWise Edge Gateway](#).

También puede configurarlas AWS IoT Core para que registren información en los CloudWatch registros si está solucionando una acción de AWS IoT SiteWise regla. Para obtener más información, consulte [Solucionar problemas con una acción de AWS IoT SiteWise regla](#).

Contenido

- [Administra el inicio de sesión AWS IoT SiteWise](#)
 - [Búsqueda de su nivel de registro](#)
 - [Cambio del nivel de registro](#)
- [Ejemplo: entradas de archivos de AWS IoT SiteWise registro](#)

Administra el inicio de sesión AWS IoT SiteWise

Utilice la AWS IoT SiteWise consola o AWS CLI para las siguientes tareas de configuración de registro.

Búsqueda de su nivel de registro

Console

Utilice el procedimiento siguiente para buscar el nivel de registro actual en la consola de AWS IoT SiteWise .

Para encontrar su nivel de AWS IoT SiteWise registro actual

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación izquierdo, elija Logging options (Opciones de registro).

El estado actual de registro aparece en Logging status (Estado de registro). Si el registro está activado, el nivel de registro actual aparece en Nivel de detalle.

AWS CLI

Ejecute el siguiente comando para encontrar su nivel de AWS IoT SiteWise registro actual con el AWS CLI.

```
aws iotsitewise describe-logging-options
```

La operación devuelve una respuesta que contiene el nivel de registro en el siguiente formato.

```
{
  "loggingOptions": {
    "level": "String"
  }
}
```

Cambio del nivel de registro

Utilice el siguiente procedimiento para cambiar el nivel de registro en la AWS IoT SiteWise consola o mediante AWS CLI.

Console

Para cambiar el nivel de AWS IoT SiteWise registro

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación izquierdo, elija Logging options (Opciones de registro).
3. Elija Editar.
4. Elija el Nivel de detalle que desea activar.
5. Seleccione Guardar.

AWS CLI

Ejecute el siguiente AWS CLI comando para cambiar el nivel de AWS IoT SiteWise registro. *logging-level* Sustitúyalo por el nivel de registro que desee.

```
aws iotsitewise put-logging-options --logging-options level=logging-level
```

Ejemplo: entradas de archivos de AWS IoT SiteWise registro

Cada entrada de AWS IoT SiteWise registro incluye información sobre el evento y los recursos relevantes para ese evento, de modo que pueda comprender y analizar los datos del registro.

En el siguiente ejemplo, se muestra una CloudWatch entrada de AWS IoT SiteWise registros que registra cuándo se crea correctamente un modelo de activos.

```
{
  "eventTime": "2020-05-05T00:10:22.902Z",
  "logLevel": "INFO",
  "eventType": "AssetModelCreationSuccess",
  "message": "Successfully created asset model.",
  "resources": {
    "assetModelId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
  }
}
```

Supervise los registros de SiteWise Edge Gateway

Puede configurar su puerta de enlace AWS IoT SiteWise Edge para registrar la información en Amazon CloudWatch Logs o en el sistema de archivos local.

Temas

- [Usa Amazon CloudWatch Logs](#)
- [Usa los registros del servicio en AWS IoT SiteWise](#)

Usa Amazon CloudWatch Logs

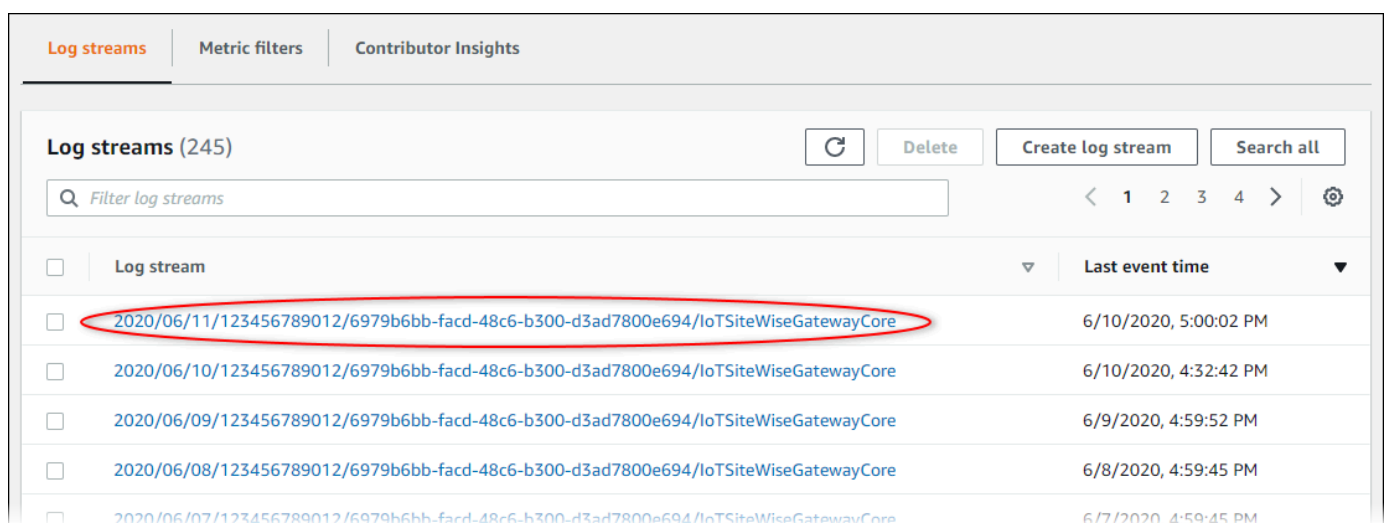
Puede configurar su puerta de enlace SiteWise Edge para enviar CloudWatch registros a Logs. Para obtener más información, consulte [Habilitar el registro de CloudWatch registros en la Guía para AWS IoT Greengrass Version 2](#) desarrolladores.

Para configurar los CloudWatch registros y acceder a ellos (consola)

1. Vaya a la [consola de CloudWatch](#).
2. En el panel de navegación, seleccione Grupos de registro.
3. Puede encontrar los registros de los AWS IoT SiteWise componentes en los siguientes grupos de registros:
 - `/aws/greengrass/UserComponent/region/aws.iot.SiteWiseEdgeCollectorOpcua`— Los registros del componente de la puerta de enlace SiteWise Edge que recopila datos de las fuentes OPC UA de la puerta de enlace SiteWise Edge.
 - `/aws/greengrass/UserComponent/region/aws.iot.SiteWiseEdgePublisher`— Los registros del componente de la puerta de enlace SiteWise Edge que publica los flujos de datos OPC UA en. AWS IoT SiteWise

Elija el grupo de registro de la función que desea depurar.

4. Elija un flujo de registro que tenga un nombre que termine con el nombre de su AWS IoT Greengrass grupo. De forma predeterminada, CloudWatch muestra primero el flujo de registro más reciente.



Log stream	Last event time
2020/06/11/123456789012/6979b6bb-facd-48c6-b300-d3ad7800e694/IoTSiteWiseGatewayCore	6/10/2020, 5:00:02 PM
2020/06/10/123456789012/6979b6bb-facd-48c6-b300-d3ad7800e694/IoTSiteWiseGatewayCore	6/10/2020, 4:32:42 PM
2020/06/09/123456789012/6979b6bb-facd-48c6-b300-d3ad7800e694/IoTSiteWiseGatewayCore	6/9/2020, 4:59:52 PM
2020/06/08/123456789012/6979b6bb-facd-48c6-b300-d3ad7800e694/IoTSiteWiseGatewayCore	6/8/2020, 4:59:45 PM
2020/06/07/123456789012/6979b6bb-facd-48c6-b300-d3ad7800e694/IoTSiteWiseGatewayCore	6/7/2020, 4:59:45 PM

5. Para mostrar los registros de los últimos 5 minutos, haga lo siguiente:
 - a. Elija custom (personalizado) en la esquina superior derecha.
 - b. Elija Relative (Relativo).
 - c. Elija 5 minutos.
 - d. Seleccione Aplicar.

6. (Opcional) Para ver menos registros, puede seleccionar 1m (1 minuto) en la esquina superior derecha.
7. Desplácese hasta la parte inferior de las entradas de registro para ver los registros más recientes.

Usa los registros del servicio en AWS IoT SiteWise

SiteWise Los dispositivos Edge Gateway incluyen archivos de registro de servicio para ayudar a depurar los problemas. Las siguientes secciones le ayudarán a encontrar y utilizar los archivos de registro de servicio de los componentes AWS IoT SiteWise OPC UA Collector y AWS IoT SiteWise Publisher.

AWS IoT SiteWise Archivo de registro del servicio OPC UA Collector

El componente AWS IoT SiteWise OPC UA Collector utiliza el siguiente archivo de registro.

Linux

```
/greengrass/v2/logs/aws.iot.SiteWiseEdgeCollectorOpcua.log
```

Windows

```
C:\greengrass\v2\logs\aws.iot.SiteWiseEdgeCollectorOpcua.log
```

Visualización de los registros de este componente

- Ejecute el siguiente comando en el dispositivo de núcleo para ver el archivo de registro de este componente en tiempo real. Sustituya */greengrass/v2* o *C:\greengrass\v2* por la ruta a la carpeta AWS IoT Greengrass raíz.

Linux

```
sudo tail -f /greengrass/v2/logs/aws.iot.SiteWiseEdgeCollectorOpcua.log
```

Windows (PowerShell)

```
Get-Content C:\greengrass\v2\logs\aws.iot.SiteWiseEdgeCollectorOpcua.log -Tail  
10 -Wait
```

AWS IoT SiteWise Archivo de registro del servicio de Publisher

El componente AWS IoT SiteWise Publisher utiliza el siguiente archivo de registro.

Linux

```
/greengrass/v2/logs/aws.iot.SiteWiseEdgePublisher.log
```

Windows

```
C:\greengrass\v2\logs\aws.iot.SiteWiseEdgePublisher.log
```

Visualización de los registros de este componente

- Ejecute el siguiente comando en el dispositivo de núcleo para ver el archivo de registro de este componente en tiempo real. Sustituya `/greengrass/v2` o `C:\greengrass\v2` por la ruta a la carpeta AWS IoT Greengrass raíz.

Linux

```
sudo tail -f /greengrass/v2/logs/aws.iot.SiteWiseEdgePublisher.log
```

Windows (PowerShell)

```
Get-Content C:\greengrass\v2\logs\aws.iot.SiteWiseEdgePublisher.log -Tail 10 -  
Wait
```

Supervisa AWS IoT SiteWise con las CloudWatch métricas de Amazon

Puede monitorizar el AWS IoT SiteWise uso CloudWatch, que recopila datos sin procesar y los procesa para convertirlos en métricas legibles prácticamente en tiempo real. Estas estadísticas se mantienen durante 15 meses, de forma que pueda obtener acceso a información histórica y disponer de una mejor perspectiva sobre el desempeño de su aplicación web o servicio. También puede establecer alarmas que vigilen determinados umbrales y enviar notificaciones o realizar acciones cuando se cumplan dichos umbrales. Para obtener más información, consulta la [Guía del CloudWatch usuario de Amazon](#).

AWS IoT SiteWise publica las métricas y dimensiones que se enumeran en las secciones siguientes en el espacio de `AWS/IoTSiteWise` nombres.

Tip

AWS IoT SiteWise publica las métricas en un intervalo de un minuto. Cuando visualice estas métricas en gráficos en la CloudWatch consola, le recomendamos que elija un período de 1 minuto. Esto le permite ver la resolución más alta disponible de los datos de métrica.

Temas

- [AWS IoT Greengrass Version 2 métricas de puerta de enlace](#)

AWS IoT Greengrass Version 2 métricas de puerta de enlace

AWS IoT SiteWise publica métricas de pasarela para las transmisiones clásicas, las pasarelas V2 y las pasarelas V3 habilitadas para MQTT. A menos que se indique lo contrario, cada métrica es aplicable a ambas versiones de puertas de enlace autohospedadas. Todas las métricas de SiteWise Edge Gateway se publican en un intervalo de un minuto.

SiteWise Métricas de Edge Gateway

Métrica	Descripción
<code>Gateway.AvailableMemory</code>	<p>La memoria disponible de una puerta de enlace SiteWise Edge.</p> <p>Unidad: bytes</p> <p>Dimensión: None</p>
<code>Gateway.AvailableDiskSpace</code>	<p>El espacio en disco disponible de una puerta de enlace SiteWise Edge.</p> <p>Unidad: bytes</p> <p>Dimensión: None</p>
<code>Gateway.CloudConnectivity</code>	<p>El estado de conectividad a la nube de una puerta de enlace SiteWise Edge.</p> <p>Unidad: ninguna</p> <p>Dimensión: GatewayId</p>
<code>Gateway.CpuUsage</code>	<p>El uso de la CPU de una puerta de enlace SiteWise Edge.</p> <p>Unidad: porcentaje</p> <p>Dimensión: None</p>

Métrica	Descripción
<code>Gateway.TotalDiskSpace</code>	<p>El espacio total en disco de una puerta de enlace SiteWise Edge.</p> <p>Unidad: bytes</p> <p>Dimensión: None</p>
<code>Gateway.TotalMemory</code>	<p>La memoria total de una puerta de enlace SiteWise Edge.</p> <p>Unidad: bytes</p> <p>Dimensión: None</p>
<code>Gateway.UsedDiskSpace</code>	<p>El espacio en disco utilizado de una puerta de enlace SiteWise Edge.</p> <p>Unidad: bytes</p> <p>Dimensión: None</p>
<code>Gateway.UsedMemory</code>	<p>La memoria utilizada de una puerta de enlace SiteWise Edge.</p> <p>Unidad: bytes</p> <p>Dimensión: None</p>
<code>Gateway.UsedPercentageDiskSpace</code>	<p>El porcentaje de espacio en disco utilizado de una puerta de enlace SiteWise Edge.</p> <p>Unidad: bytes</p> <p>Dimensión: None</p>

Métrica	Descripción
<code>Gateway.UsedPercentageMemory</code>	<p>El porcentaje de memoria utilizado de una puerta de enlace SiteWise Edge.</p> <p>Unidad: bytes</p> <p>Dimensión: None</p>

AWS IoT SiteWise métricas del editor

Métrica	Descripción
<code>IoTSiteWisePublisher.ComponentBuildVersion</code>	<p>Esta métrica indica la versión de compilación del componente de SiteWise editor de IoT que se ejecuta en la puerta de enlace. Un valor de 1 significa que la puerta de enlace ejecuta una versión del editor correspondiente a la <code>ComponentBuildVersion</code> dimensión.</p> <p>Unidad: 1</p> <p>Dimensiones: <code>GatewayId</code>, <code>ComponentBuildVersion</code></p>
<code>IoTSiteWisePublisher.DroppedCount</code>	<p>La cantidad de puntos de datos que descarta una puerta de enlace SiteWise Edge (<code>GatewayId</code>) y que no se publican en la nube, generados cada minuto.</p> <p>Unidad: recuento</p> <p>Dimensiones: <code>GatewayId</code></p>
<code>IoTSiteWisePublisher.Heartbeat</code>	<p>Generado cada minuto por el editor en la puerta de enlace SiteWise Edge.</p> <p>Unidad: 1 (1 representa que el publicador está en ejecución y si falta el punto de datos</p>

Métrica	Descripción
	<p>representa que el publicador no está en ejecución).</p> <p>Dimensiones: GatewayId</p>
<p><code>IoTSiteWisePublisher.IsConnectedToMqttBroker</code></p>	<p>Generado cada minuto por el editor en la puerta de enlace SiteWise Edge.</p> <p>Unidad: 1 (1 representa que el publicador está conectado a un agente MQTT).</p> <p>Dimensiones: GatewayId</p>
<p><code>IoTSiteWisePublisher.MessageCheckpointPersistenceErrorCount</code></p>	<p>La métrica indica que la puerta de enlace ha detectado un problema con el archivo de puntos de comprobación utilizado para hacer un seguimiento de los datos procesados por el publicador. Un valor de 1 significa que se ha producido un error.</p> <p>Unidad: ninguna</p> <p>Dimensiones: AccountId, GatewayId</p>
<p><code>IoTSiteWisePublisher.MqttMessageReceivedSuccessCount</code></p>	<p>Número de mensajes recibidos correctamente por el publicador del agente MQTT, generado cada minuto.</p> <p>Unidad: recuento</p> <p>Dimensiones: GatewayId</p>
<p><code>IoTSiteWisePublisher.MqttReceivedSuccessBytes</code></p>	<p>Número de bytes de datos de mensajes recibidos correctamente por el publicador del agente MQTT, generado cada minuto.</p> <p>Unidad: recuento</p> <p>Dimensiones: GatewayId</p>

Métrica	Descripción
<code>IoTSiteWisePublisher.NumberOfSubscriptionsToMqttBroker</code>	<p>Número de temas suscritos al agente MQTT por el procesador, generado cada minuto. Un tema comodín de varios niveles se cuenta como 1.</p> <p>Unidad: recuento</p> <p>Dimensiones: GatewayId</p>
<code>IoTSiteWisePublisher.NumberOfUniqueMqttTopicsReceived</code>	<p>Número de temas únicos recibidos por el publicador del agente MQTT, generado cada minuto.</p> <p>Unidad: recuento</p> <p>Dimensiones: GatewayId</p>
<code>IoTSiteWisePublisher.PublisherFailureCount</code>	<p>La cantidad de puntos de datos que una puerta de enlace SiteWise Edge (GatewayId) no pudo publicar, generada cada minuto.</p> <p>Unidad: recuento</p> <p>Dimensiones: GatewayId</p>
<code>IoTSiteWisePublisher.PublisherRejectedCount</code>	<p>La cantidad de puntos de datos que una puerta de enlace SiteWise Edge (GatewayId) rechazó desde la nube y generó cada minuto.</p> <p>Unidad: recuento</p> <p>Dimensiones: GatewayId</p>

Métrica	Descripción
<code>IoTSiteWisePublisher.PublisherSuccessCount</code>	<p>La cantidad de puntos de datos que una puerta de enlace SiteWise Edge (<code>GatewayId</code>) publicó correctamente en la nube y generó cada minuto.</p> <p>Unidad: recuento</p> <p>Dimensiones: <code>GatewayId</code></p>
<code>IoTSiteWisePublisher.PublisherToS3FailureCount</code>	<p>Número de puntos de datos que una puerta de enlace (<code>GatewayId</code>) no ha publicado en un bucket de Amazon S3.</p> <p>Unidad: recuento</p> <p>Dimensiones: <code>GatewayId</code></p>
<code>IoTSiteWisePublisher.PublisherToS3SuccessCount</code>	<p>Número de puntos de datos que una puerta de enlace (<code>GatewayId</code>) ha publicado correctamente en un bucket de Amazon S3.</p> <p>Unidad: recuento</p> <p>Dimensiones: <code>GatewayId</code></p>

Métricas del recopilador de OPC UA

Métrica	Descripción
<code>OpcUaCollector.ActiveDataStreamCount</code>	<p>El número de flujos de datos a los que se ha suscrito una puerta de enlace SiteWise Edge (<code>gatewayId</code>) para una fuente OPC UA (<code>sourceName</code>).</p> <p>Unidad: recuento</p>

Métrica	Descripción
<p><code>OpcUaCollector.ComponentBuildVersion</code> (no disponible en las transmisiones clásicas ni en las pasarelas V2)</p>	<p>Dimensiones: GatewayId,, SourceName PropertyGroup</p> <p>Esta métrica indica la versión de compilación del componente recopilador IoT SiteWise OPC UA que se ejecuta en la puerta de enlace. Un valor de 1 significa que la puerta de enlace ejecuta una versión del recopilador correspondiente a la <code>ComponentBuildVersion</code> dimensión.</p> <p>Unidad: 1</p> <p>Dimensiones: GatewayId, ComponentBuildVersion</p>
<p><code>OpcUaCollector.ConversionErrors</code></p>	<p>El número de puntos de datos que una puerta de enlace SiteWise Edge (<code>gatewayId</code>) recibió para una fuente OPC UA (<code>sourceName</code>) y que provocaron errores de conversión al AWS IoT SiteWise enviar los datos. El recopilador de OPC UA no ingerirá estos puntos de datos.</p> <p>Unidad: recuento</p> <p>Dimensiones: GatewayId, SourceName</p>
<p><code>OpcUaCollector.Heartbeat</code></p>	<p>Se genera cada minuto para cada fuente OPC UA (<code>sourceName</code>) conectada a una puerta de enlace SiteWise Edge (<code>gatewayId</code>).</p> <p>Unidad: Recuento (1 representa que el origen está conectado y 0 que el origen está desconectado).</p> <p>Dimensiones: GatewayId, SourceName</p>

Métrica	Descripción
<p><code>OpcUaCollector.IncomingValuesCount</code></p>	<p>La cantidad de puntos de datos que una puerta de enlace SiteWise Edge (<code>gatewayId</code>) recibe para una fuente OPC UA (<code>sourceName</code>), generada cada minuto.</p> <p>Unidad: recuento</p> <p>Dimensiones: GatewayId,, SourceName PropertyGroup</p>
<p><code>OpcUaCollector.IncomingValueErrors</code></p>	<p>El número de puntos de datos que una puerta de enlace SiteWise Edge (<code>gatewayId</code>) recibe de una fuente OPC UA (<code>sourceName</code>) que no son valores válidos. El recopilador de OPC UA no ingiere estos puntos de datos, generados cada minuto.</p> <p>Unidad: recuento</p> <p>Dimensiones: GatewayId,, SourceName PropertyGroup</p>
<p><code>OpcUaCollector.IsConnectedToMqttBroker</code> (no disponible en las transmisiones clásicas ni en las pasarelas V2)</p>	<p>Generado cada minuto por el componente recopilador IoT SiteWise OPC UA en la puerta de enlace SiteWise Edge.</p> <p>Unidad: 1 (1 representa que el componente recopilador SiteWise OPC UA de IoT está conectado a un intermediario MQTT)</p> <p>Dimensiones: GatewayId</p>

Métrica	Descripción
<code>OpcUaCollector.MqttMessagesDroppedCount</code> (no disponible en transmisiones clásicas y pasarelas V2)	<p>La cantidad de mensajes MQTT descartados por el componente recopilador IoT SiteWise OPC UA.</p> <p>Unidad: recuento</p> <p>Dimensiones: GatewayId SourceName</p>
<code>OpcUaCollector.MqttMessagesPublishedBytes</code> (no disponible en transmisiones clásicas y pasarelas V2)	<p>El número de bytes de datos de mensajes MQTT publicados correctamente por el componente recopilador IoT SiteWise OPC UA en el broker MQTT.</p> <p>Unidad: recuento</p> <p>Dimensiones: GatewayId SourceName</p>
<code>OpcUaCollector.MqttMessagesPublishedCount</code> (no disponible en transmisiones clásicas y pasarelas V2)	<p>El número de mensajes MQTT publicados correctamente por el componente recopilador SiteWise OPC UA de IoT en el broker MQTT.</p> <p>Unidad: recuento</p> <p>Dimensiones: GatewayId SourceName</p>
<code>OpcUaCollector.NullValueCount</code> (no disponible en transmisiones clásicas y pasarelas V2)	<p>El número de valores nulos recibidos por el componente recopilador SiteWise OPC UA de IoT desde el servidor OPC UA.</p> <p>Unidad: recuento</p> <p>Dimensiones: GatewayId, SourceName PropertyGroup</p>

Métrica	Descripción
<code>OpcUaCollector.NumberOfUniqueMqttTopicsPublished</code> (no disponible en las transmisiones clásicas ni en las pasarelas V2)	<p>El número de temas únicos de MQTT publicados por el recopilador de IoT SiteWise OPC UA para el bróker de MQTT.</p> <p>Unidad: recuento</p> <p>Dimensiones: GatewayId SourceName</p>

AWS IoT SiteWise métricas del procesador

Métrica	Descripción
<code>Gateway.DataProcessor.IngestionThrottled</code> (no disponible en las pasarelas V3 habilitadas para MQTT)	<p>Número de puntos de datos que se han limitado, generado cada minuto.</p> <p>Unidad: recuento</p> <p>Dimensiones: ThrottledAt</p>
<code>Gateway.DataProcessor.MeasurementRejected</code> (no disponible en las pasarelas V3 habilitadas para MQTT)	<p>Número de mediciones que se rechazaron, generado cada minuto.</p> <p>Unidad: recuento</p> <p>Dimensiones: Reason</p>
<code>Gateway.DataProcessor.MessagesRemaining</code> (no disponible en las pasarelas V3 habilitadas para MQTT)	<p>Número de mensajes que quedan en un flujo, generado cada minuto.</p> <p>Unidad: recuento</p> <p>Dimensiones: StreamName</p>
<code>Gateway.DataProcessor.ProcessingError</code> (no disponible en las pasarelas V3 habilitadas para MQTT)	<p>Número de errores de procesamiento, generado cada minuto.</p> <p>Unidad: recuento</p>

Métrica	Descripción
	Dimensiones: Reason

Registra las llamadas a la AWS IoT SiteWise API con AWS CloudTrail

AWS IoT SiteWise está integrado con AWS CloudTrail un servicio que proporciona un registro de las acciones realizadas por un usuario, un rol o un AWS servicio en AWS IoT SiteWise. CloudTrail captura las llamadas a la API AWS IoT SiteWise como eventos. Las llamadas capturadas incluyen llamadas desde la AWS IoT SiteWise consola y llamadas en código a las operaciones de la AWS IoT SiteWise API. Si crea una ruta, puede activar la entrega continua de CloudTrail eventos a un bucket de Amazon S3, incluidos los eventos de AWS IoT SiteWise. Si no configura una ruta, podrá ver los eventos más recientes en la CloudTrail consola, en el historial de eventos. Con la información recopilada por usted CloudTrail, puede determinar el destinatario de la solicitud AWS IoT SiteWise, la dirección IP desde la que se realizó la solicitud, quién la realizó, cuándo se realizó y detalles adicionales.

Para obtener más información al respecto CloudTrail, consulte la [Guía AWS CloudTrail del usuario](#).

AWS IoT SiteWise información en CloudTrail

CloudTrail se activa en su AWS cuenta al crear la cuenta. Cuando se produce una actividad de eventos admitida AWS IoT SiteWise, esa actividad se registra en un CloudTrail evento junto con otros eventos de AWS servicio en el historial de eventos. Puedes ver, buscar y descargar los eventos recientes en tu AWS cuenta. Para obtener más información, consulta [Ver eventos con el historial de CloudTrail eventos](#).

Para tener un registro continuo de los eventos de tu AWS cuenta, incluidos los eventos de tu cuenta AWS IoT SiteWise, crea una ruta. Un rastro permite CloudTrail entregar archivos de registro a un bucket de Amazon S3. De manera predeterminada, cuando se crea un registro de seguimiento en la consola, el registro de seguimiento se aplica a todas las regiones de AWS. La ruta registra los eventos de todas las regiones de la AWS partición y envía los archivos de registro al bucket de Amazon S3 que especifique. Además, puede configurar otros AWS servicios para analizar más a fondo los datos de eventos recopilados en los CloudTrail registros y actuar en función de ellos. Para más información, consulte los siguientes temas:

- [Introducción a la creación de registros de seguimiento](#)
- [CloudTrail servicios e integraciones compatibles](#)
- [Configuración de las notificaciones de Amazon SNS para CloudTrail](#)
- [Recibir archivos de CloudTrail registro de varias regiones](#) y [recibir archivos de CloudTrail registro de varias cuentas](#)

Cada entrada de registro o evento contiene información sobre quién generó la solicitud. La información de identidad del usuario le ayuda a determinar lo siguiente:

- Si la solicitud se realizó con credenciales de usuario root o AWS Identity and Access Management (IAM).
- Si la solicitud se realizó con credenciales de seguridad temporales de un rol o fue un usuario federado.
- Si la solicitud la realizó otro AWS servicio.

Para obtener más información, consulte el [elemento userIdentity de CloudTrail](#).

AWS IoT SiteWise eventos de datos en CloudTrail

Los [eventos de datos](#) proporcionan información sobre las operaciones de recursos realizadas en o dentro de un recurso (por ejemplo, leer o escribir en un objeto de Amazon S3). Se denominan también operaciones del plano de datos. Los eventos de datos suelen ser actividades de gran volumen. De forma predeterminada, CloudTrail no registra los eventos de datos. El historial de CloudTrail eventos no registra los eventos de datos.

Se aplican cargos adicionales a los eventos de datos. Para obtener más información sobre CloudTrail los precios, consulta [AWS CloudTrail Precios](#).

Puede registrar eventos de datos para los tipos de AWS IoT SiteWise recursos mediante la CloudTrail consola o las operaciones de la CloudTrail API. AWS CLI La [tabla](#) de esta sección muestra los tipos de recursos disponibles para AWS IoT SiteWise.

- Para registrar eventos de datos mediante la CloudTrail consola, cree un [almacén de datos de rutas o eventos](#) para registrar eventos de datos, o [actualice un banco de datos de seguimiento o evento existente](#) para registrar eventos de datos.
 1. Para registrar eventos de datos, elija Eventos de datos.

2. En la lista Tipo de evento de datos, elija el tipo de recurso en el que desea registrar los eventos de datos.
 3. Elija la plantilla de selector de registro que desea usar. Puede registrar todos los eventos de datos del tipo de recurso, registre todos los eventos `readOnly`, registre todos los eventos `writeOnly` o cree una plantilla de selector de registro personalizada para filtrar por los campos `readOnly`, `eventName` y `resources.ARN`.
- Para registrar eventos de datos mediante el AWS CLI, configure el `--advanced-event-selectors` parámetro para que el `eventCategory` campo sea igual al valor del tipo de recurso `Data` y el `resources.type` campo igual al valor del tipo de recurso (consulte [la tabla](#)). Puede agregar condiciones para filtrar los valores de los campos `readOnly`, `eventName` y `resources.ARN`.
 - Para configurar una ruta para registrar eventos de datos, ejecute el [AWS CloudTrail put-event-selectors](#) comando. Para obtener más información, consulte [Registro de eventos de datos para registros de seguimiento en la AWS CLI](#).
 - Para configurar un banco de datos de eventos para registrar eventos de datos, ejecute el [AWS CloudTrail create-event-data-store](#) comando para crear un nuevo banco de datos de eventos para registrar eventos de datos o ejecute el [AWS CloudTrail update-event-data-store](#) comando para actualizar un banco de datos de eventos existente. Para obtener más información, consulte [Registro de eventos de datos para eventos de almacenes de datos con AWS CLI](#).

En la siguiente tabla se enumeran los tipos de AWS IoT SiteWise recursos. La columna Tipo de evento de datos (consola) muestra el valor que se puede elegir en la lista de tipos de eventos de datos de la CloudTrail consola. La columna de valores `resources.type` muestra el `resources.type` valor que se debe especificar al configurar los selectores de eventos avanzados mediante o. AWS CLI CloudTrail APIs La CloudTrail columna Datos APIs registrados muestra las llamadas a la API registradas CloudTrail para el tipo de recurso.

Tipo de evento de datos (consola)	<code>resources.type</code> value	Datos APIs registrados en CloudTrail *
AWS IoT SiteWise recurso	<code>AWS::IoTSiteWise::Asset</code>	<ul style="list-style-type: none"> • BatchPutAssetPropertyValue • GetAssetPropertyValue • GetAssetPropertyValueHistory

Tipo de evento de datos (consola)	resources.type value	Datos APIs registrados en CloudTrail *
		<ul style="list-style-type: none"> • GetAssetPropertyAggregates • GetInterpolatedAssetPropertyValues • BatchGetAssetPropertyValue • BatchGetAssetPropertyValueHistory • BatchGetAssetPropertyAggregates
AWS IoT SiteWise series temporales	AWS::IoTSiteWise::TimeSeries	<ul style="list-style-type: none"> • BatchPutAssetPropertyValue • GetAssetPropertyValue • GetAssetPropertyValueHistory • GetAssetPropertyAggregates • GetInterpolatedAssetPropertyValues • BatchGetAssetPropertyValue • BatchGetAssetPropertyValueHistory • BatchGetAssetPropertyAggregates
AWS IoT SiteWise Asistente	AWS::SitewiseAssistant::Conversation	<ul style="list-style-type: none"> • InvokeAssistant

Note

El `resources.type` registrado en el evento de Cloudtrail depende del identificador utilizado en la solicitud de la API. Si se especifica un identificador de activo en la solicitud, se registra el `Asset resources.type`; de lo contrario, se registra el `TimeSeries resources.type`.

*Puede configurar selectores de eventos avanzados para filtrar según los campos `eventName`, `readOnly` y `resources.ARN` y así registrar solo los eventos que son importantes para usted. Para obtener más información acerca de estos campos, consulte [AdvancedFieldSelector](#).

AWS IoT SiteWise eventos de gestión en CloudTrail

El [registro de los eventos](#) de administración proporciona información sobre las operaciones de administración que se realizan en los recursos de su AWS cuenta. Se denominan también operaciones del plano de control. De forma predeterminada, CloudTrail registra los eventos de administración.

AWS IoT SiteWise registra todas las operaciones del plano de AWS IoT SiteWise control como eventos de administración. Para obtener una lista de las operaciones del plano de AWS IoT SiteWise control en las que se AWS IoT SiteWise registra CloudTrail, consulte la [referencia de la AWS IoT SiteWise API](#).

Ejemplo: entradas de archivos de AWS IoT SiteWise registro

Un rastro es una configuración que permite la entrega de eventos como archivos de registro a un bucket de Amazon S3 que usted especifique. CloudTrail Los archivos de registro contienen una o más entradas de registro. Un evento representa una solicitud única de cualquier fuente e incluye información sobre la operación solicitada, la fecha y la hora de la operación, los parámetros de la solicitud, etc. CloudTrail Los archivos de registro no son un seguimiento ordenado de las llamadas a la API pública, por lo que no aparecen en ningún orden específico.

En el siguiente ejemplo, se muestra una entrada de CloudTrail registro que demuestra la `CreateAsset` operación.

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
```

```
"principalId": "AIDACKCEVSQ6C2EXAMPLE",
"arn": "arn:aws:iam::123456789012:user/Administrator",
"accountId": "123456789012",
"accessKeyId": "AKIAIOSFODNN7EXAMPLE",
"userName": "Administrator",
"sessionContext": {
  "sessionIssuer": {},
  "webIdFederationData": {},
  "attributes": {
    "mfaAuthenticated": "false",
    "creationDate": "2020-03-11T17:26:40Z"
  }
},
"invokedBy": "signin.amazonaws.com"
},
"eventTime": "2020-03-11T18:01:22Z",
"eventSource": "iotsitewise.amazonaws.com",
"eventName": "CreateAsset",
"awsRegion": "us-east-1",
"sourceIPAddress": "203.0.113.0",
"userAgent": "signin.amazonaws.com",
"requestParameters": {
  "assetName": "Wind Turbine 1",
  "assetModelId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
  "clientToken": "a1b2c3d4-5678-90ab-cdef-00000EXAMPLE"
},
"responseElements": {
  "assetId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
  "assetArn": "arn:aws:iotsitewise:us-east-1:123456789012:asset/a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
  "assetStatus": {
    "state": "CREATING"
  }
},
"requestID": "a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE",
"eventID": "a1b2c3d4-5678-90ab-cdef-bbbbbEXAMPLE",
"eventType": "AwsApiCall",
"recipientAccountId": "123456789012"
}
```

Etiquete sus AWS IoT SiteWise recursos

Etiquetar AWS IoT SiteWise los recursos proporciona una forma eficaz de clasificar, gestionar y recuperar los activos de la organización de forma eficiente. Al asignar etiquetas, que consisten en pares clave-valor, puede asociar metadatos descriptivos a sus recursos. Los metadatos de las etiquetas se pueden utilizar para agilizar las operaciones. Por ejemplo, en el escenario de un parque eólico, las etiquetas permiten etiquetar turbinas con atributos específicos, como ubicación, capacidad y estado operativo, lo que permite identificarlas y administrarlas rápidamente en AWS IoT SiteWise.

La integración de las etiquetas con las políticas AWS Identity and Access Management (IAM) mejora la seguridad y el control operativo mediante la definición de reglas de acceso condicional. Esto significa que puede especificar que solo los usuarios tengan determinadas etiquetas. Por ejemplo, solo las personas etiquetadas con un determinado rol o departamento pueden acceder a determinados recursos o modificarlos.

Utilice etiquetas en AWS IoT SiteWise

Use etiquetas para clasificar sus AWS IoT SiteWise recursos por propósito, propietario, entorno o cualquier otra clasificación para su caso de uso. Cuando tenga muchos recursos del mismo tipo, podrá identificar rápidamente un recurso específico en función de las etiquetas.

Cada etiqueta consta de una clave y un valor opcional que usted especifica. Por ejemplo, puede establecer una serie de etiquetas para sus modelos de activos a fin de hacer su seguimiento de acuerdo con los procesos industriales a los que dan soporte. Se recomienda desarrollar un conjunto personalizado de claves de etiquetas para cada tipo de recurso que administre. El uso de un conjunto coherente de claves de etiquetas facilita la administración de los recursos.

Etiquete con la Consola de administración de AWS

El editor de etiquetas del Consola de administración de AWS proporciona una forma centralizada y unificada de crear y administrar las etiquetas para los recursos de todos los AWS servicios. Para obtener más información, consulte [Introducción al editor de etiquetas](#) en los AWS recursos de etiquetado y en la Guía del usuario del editor de etiquetas.

Etiquete con la API AWS IoT SiteWise

La AWS IoT SiteWise API también usa etiquetas. Antes de crear etiquetas, tenga en cuenta las restricciones de etiquetado. Para obtener más información, consulte este artículo sobre las [convenciones de nomenclatura y el uso de etiquetas](#) en la Referencia general de AWS.

- Para agregar etiquetas al crear un recurso, deberá definir las en la propiedad `tags` del recurso.
- Para añadir etiquetas a un recurso existente o para actualizar los valores de las etiquetas, utilice la [TagResource](#) operación.
- Para eliminar etiquetas de un recurso, utilice la [UntagResource](#) operación.
- Para recuperar las etiquetas asociadas a un recurso, utilice la [ListTagsForResource](#) operación o describa el recurso e inspeccione su `tags` propiedad.

En la siguiente tabla, se enumeran los recursos que puedes etiquetar mediante la AWS IoT SiteWise API `Create` y sus `Describe` operaciones correspondientes.

Recursos etiquetables AWS IoT SiteWise

Recurso	Crear operación	Describir operación
Modelo de activos o modelo de componentes	CreateAssetModel	DescribeAssetModel
Activo	CreateAsset	DescribeAsset
SiteWise Puerta de enlace Edge	CreateGateway	DescribeGateway
Portal	CreatePortal	DescribePortal
Proyecto	CreateProject	DescribeProject
Panel de control	CreateDashboard	DescribeDashboard
Política de acceso	CreateAccessPolicy	DescribeAccessPolicy
Serie temporal	BatchPutAssetPropertyValue	DescribeTimeSeries

Por [BatchPutAssetPropertyValue](#) ejemplo, puede configurar sus fuentes de datos para enviar datos industriales AWS IoT SiteWise antes de crear modelos y activos de activos. AWS IoT SiteWise crea automáticamente flujos de datos para recibir flujos de datos sin procesar de su equipo. Para obtener más información, consulte [Administración de la ingesta de datos](#).

Utilice las siguientes operaciones para ver y administrar etiquetas para los recursos que admiten etiquetas:

- [TagResource](#)— Añade etiquetas a un recurso o actualiza el valor de una etiqueta existente.
- [ListTagsForResource](#)— Muestra las etiquetas de un recurso.
- [UntagResource](#)— Elimina las etiquetas de un recurso.

Agregue o quite etiquetas de un recurso en cualquier momento. Para actualizar el valor de una clave de etiqueta existente, añada al recurso una nueva etiqueta con la misma clave y el nuevo valor deseado. Esta acción sustituye el valor anterior por el nuevo. Si bien es posible asignar una cadena vacía como valor de etiqueta, no se puede asignar un valor nulo.

Al eliminar un recurso, también se eliminan todas las etiquetas vinculadas a él.

Uso de etiquetas con políticas de IAM

Utilice etiquetas de recursos en sus políticas de IAM para controlar los permisos y el acceso de usuarios. Por ejemplo, las políticas pueden permitir a los usuarios crear solo aquellos recursos que tengan una etiqueta específica asociada. Las políticas también puede limitar la creación o modificación de recursos que tengan determinadas etiquetas por parte de los usuarios.

Note

Si utiliza etiquetas para permitir o denegar el acceso de los usuarios a los recursos, debe denegar a los usuarios la capacidad de agregar o eliminar esas etiquetas para los mismos recursos. De lo contrario, un usuario podría eludir sus restricciones y obtener acceso a un recurso modificando sus etiquetas.

Puede utilizar los siguientes valores y claves de contexto de condición en el elemento `Condition` (también llamado bloque `Condition`) de una instrucción de política.

`aws:ResourceTag/tag-key: tag-value`

Permitir o denegar acciones en recursos con etiquetas específicas.

`aws:RequestTag/tag-key: tag-value`

Exigir que se utilice (o no se utilice) una etiqueta específica al crear o modificar un recurso etiquetable.

`aws:TagKeys: [tag-key, ...]`

Exigir que se utilice (o no se utilice) un conjunto específico de claves de etiqueta al crear o modificar un recurso etiquetable.

Note

Las claves y valores de contexto de condición en una política de IAM se aplican solo a las acciones que tienen un recurso etiquetable como parámetro requerido. Por ejemplo, puede configurar el acceso condicional basado en etiquetas para [ListAssets](#). No puedes activar el acceso condicional basado en etiquetas [PutLoggingOptions](#) porque en la solicitud no se hace referencia a ningún recurso etiquetable.

Para obtener más información, consulte [Controlar el acceso a AWS los recursos mediante etiquetas de recursos](#) y la [referencia a la política JSON de IAM](#) en la Guía del usuario de IAM.

Ejemplo de políticas de IAM que utilizan etiquetas

- [Vea AWS IoT SiteWise los activos en función de las etiquetas](#)

Solución de problemas AWS IoT SiteWise

Utilice la siguiente información para solucionar problemas con AWS IoT SiteWise.

Temas

- [Solución de problemas de una puerta de enlace SiteWise Edge](#)
- [Solucionar problemas de un portal AWS IoT SiteWise](#)
- [Solucionar problemas con una acción de AWS IoT SiteWise regla](#)
- [Solución de operaciones de importación y exportación masivas](#)

Solución de problemas de una puerta de enlace SiteWise Edge

Solucione los problemas más comunes de la puerta de enlace AWS IoT SiteWise Edge explorando los temas relevantes.

También puede ver CloudWatch las métricas reportadas por sus puertas de enlace SiteWise Edge para solucionar problemas de conectividad o flujos de datos. Para obtener más información, consulte [Supervisa AWS IoT SiteWise con las CloudWatch métricas de Amazon](#).

Temas

- [Configure y acceda a los registros de la puerta de SiteWise enlace Edge](#)
- [Solución de problemas con la puerta de enlace SiteWise Edge](#)
- [Solución de problemas de la aplicación AWS IoT SiteWise Edge en Siemens Industrial Edge](#)
- [Solución de problemas de integraciones de código abierto en Edge](#)
- [Solución de problemas AWS IoT Greengrass](#)

Configure y acceda a los registros de la puerta de SiteWise enlace Edge

Para poder ver los registros de SiteWise Edge Gateway, debe configurar su puerta de enlace SiteWise Edge para que envíe los registros a Amazon CloudWatch Logs o almacene los registros en el sistema de archivos local.

- Utilice CloudWatch los registros si quiere usarlos Consola de administración de AWS para ver los archivos de registro de su puerta de enlace SiteWise Edge. Para obtener más información, consulte [Usa Amazon CloudWatch Logs](#).

- Use los registros del sistema de archivos local si quiere usar la línea de comandos o el software local para ver los archivos de registro de su puerta de enlace SiteWise Edge. Para obtener más información, consulte [Usa los registros del servicio en AWS IoT SiteWise](#).

Solución de problemas con la puerta de enlace SiteWise Edge

Utilice la siguiente información para solucionar los problemas de las puertas de enlace SiteWise Edge.

Problemas

- [No se han podido implementar paquetes en las puertas de enlace de SiteWise Edge](#)
- [AWS IoT SiteWise no recibe datos de los servidores OPC UA](#)
- [No se muestran datos en el panel](#)
- [En aws.iot aparece el mensaje «No se pudo encontrar ni cargar la clase principal». SiteWiseEdgePublisher inicia sesión en/error greengrass/v2/logs](#)
- [Aparece «SESSION_TAKEN_OVER» o «com.aws.greengrass.mqttclient». MqttClient: No se pudo publicar el mensaje a través de Spooler y lo volveré a intentar. ' en los registros](#)
- [Veo «com.aws.greengrass.deployment». lotJobsHelper: No se encontró ningún trabajo de despliegue. o 'Ya se ha informado del resultado de la implementación'. en los registros](#)
- [Aparece el estado «SYNC_FAILED» cuando intento configurar la configuración de la marca de tiempo en un grupo de propiedades de una fuente de datos OPC UA](#)
- [Los tipos de datos convertidos no se incluyen](#)
- [Problemas con el almacén de confianza](#)
- [Problemas de instalación con proxy](#)

No se han podido implementar paquetes en las puertas de enlace de SiteWise Edge

Si el componente AWS IoT Greengrass núcleo (`aws.greengrass.Nucleus`) está desactualizado, es posible que no pueda implementar paquetes en su puerta de enlace SiteWise Edge. Puede usar la AWS IoT Greengrass V2 consola para actualizar el componente AWS IoT Greengrass Nucleo.

Para actualizar el componente AWS IoT Greengrass Nucleus (consola)

1. Vaya a la [consola de AWS IoT Greengrass](#).
2. En el panel de navegación, en AWS IoT Greengrass, seleccione Implementaciones.

3. En la lista Implementaciones, seleccione la implementación que desee revisar.
4. Seleccione Revisar.
5. En la página Especificar destino, seleccione Siguiente.
6. En la página Seleccionar componentes, en Componentes públicos, en el cuadro de búsqueda, introduzca **aws.greengrass.Nucleus** y, a continuación, seleccione aws.greengrass.Nucleus.
7. Seleccione Siguiente.
8. En la página Configurar componentes, seleccione Siguiente.
9. En la página Configurar opciones avanzadas, seleccione Siguiente.
10. En la página Revisar, elija Implementar.

AWS IoT SiteWise no recibe datos de los servidores OPC UA

Si tus AWS IoT SiteWise activos no reciben los datos enviados por tus servidores OPC UA, puedes buscar en los registros de la puerta de enlace SiteWise Edge para solucionar problemas. Busque registros de `swPublisher` a nivel de información que contengan el siguiente mensaje.

```
Emitting diagnostic name=PublishError.SomeException
```

Según el tipo de registro, utiliza *SomeException* los siguientes tipos de excepciones y los problemas correspondientes para solucionar los problemas de tu SiteWise puerta de enlace Edge:

- **ResourceNotFoundException**— Sus servidores OPC UA envían datos que no coinciden con el alias de propiedad de ningún activo. Esta excepción puede ocurrir en dos casos:
 - Sus alias de propiedad no coinciden exactamente con sus variables OPC UA, incluido cualquier prefijo de origen que haya definido. Compruebe que los alias de propiedad y los prefijos de origen son correctos.
 - No ha asignado las variables OPC UA a las propiedades de los activos. Para obtener más información, consulte [Gestione los flujos de datos para AWS IoT SiteWise](#).

Si ya ha mapeado todas las variables OPC UA que desea incluir AWS IoT SiteWise, puede filtrar las variables OPC UA que envía la SiteWise puerta de enlace Edge. Para obtener más información, consulte [Utilice filtros de nodo OPC UA en Edge SiteWise](#).

- **InvalidRequestException**— Los tipos de datos de las variables OPC UA no coinciden con los tipos de datos de propiedades de sus activos. Por ejemplo, si una variable de OPC UA tiene un tipo de datos entero, la propiedad de activo correspondiente debe ser del tipo de datos entero. Una

propiedad de activo de tipo doble no puede recibir valores OPC UA enteros. Para solucionar este problema, defina nuevas propiedades con los tipos de datos correctos.

- `TimestampOutOfRangeException`— Su puerta de enlace SiteWise Edge envía datos que están fuera del rango que AWS IoT SiteWise acepta. AWS IoT SiteWise rechaza todos los puntos de datos con marcas de tiempo anteriores a 7 días o posteriores a 5 minutos en el futuro. Si la puerta de enlace SiteWise Edge se ha quedado sin alimentación o sin conexión a la AWS nube, es posible que tengas que borrar la memoria caché de la puerta de enlace SiteWise Edge.
- `ThrottlingException` o bien `LimitExceededException`: tu solicitud superó una cuota de AWS IoT SiteWise servicio, como la tasa de puntos de datos ingeridos o la tasa de solicitudes de operaciones de la API de datos sobre propiedades de activos. Compruebe que su configuración no exceda el [AWS IoT SiteWise cuotas](#).

No se muestran datos en el panel

Si no se muestran datos en el panel, es posible que la configuración de Publisher y la fuente de datos de la puerta de enlace de SiteWise Edge no estén sincronizadas. Si no están sincronizados, la actualización del nombre del origen de datos podría acelerar la sincronización desde la nube hasta la periferia y solucionar el error asincronía.

Actualización del nombre de un origen de datos

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, seleccione Puertas de enlace Edge.
3. Seleccione la puerta de enlace SiteWise Edge conectada al panel de control.
4. En Origen de datos, seleccione Editar.
5. Seleccione un nuevo Nombre de origen y, a continuación, Guardar para confirmar el cambio.
6. Verifique sus cambios confirmando que el nombre del origen de datos se haya actualizado en la tabla Orígenes de datos.

En `aws.iot` aparece el mensaje «No se pudo encontrar ni cargar la clase principal».
`SiteWiseEdgePublisher` inicia sesión en `/error greengrass/v2/logs`

Si aparece este error, es posible que deba actualizar la versión java de su puerta de enlace SiteWise Edge.

- En un terminal, ejecute el comando siguiente:

```
java -version
```

La versión de Java con la que se ejecuta su puerta de enlace SiteWise Edge aparecerá en `OpenJDK Runtime Environment`. Verá una respuesta como la siguiente:

```
openjdk version "11.0.20" 2023-07-18 LTS
OpenJDK Runtime Environment Corretto011.0.20.8.1 (build 11.0.20+8-LTS
OpenJDK 64-Bit Server VM Corretto-11.0.20.8.1 (build 11.0.20+8-LTS, mixed node)
```

Si ejecuta la versión 11.0.20.8.1 de Java, debe actualizar el paquete IoT SiteWise Publisher a la versión 2.4.1 o posterior. Solo se ve afectada la versión 11.0.20.8.1 de Java. Los entornos con otras versiones de Java pueden seguir utilizando versiones anteriores del componente IoT SiteWise Publisher. Para obtener más información sobre la actualización de un paquete de componentes, consulte [Cambie la versión de los paquetes de componentes de SiteWise Edge Gateway](#).

Aparece «SESSION_TAKEN_OVER» o «com.aws.greengrass.mqttclient». `MqttClient: No se pudo publicar el mensaje a través de Spooler y lo volveré a intentar.` ' en los registros

Si ve una advertencia `SESSION_TAKEN_OVER` o un error `com.aws.greengrass.mqttclient.MqttClient: Failed to publish the message via Spooler and will retry.` en sus registros `/greengrass/v2/logs/greengrass.log`, puede que esté intentando utilizar el mismo archivo de configuración para varias puertas de enlace SiteWise Edge en varios dispositivos. Cada puerta de enlace SiteWise Edge necesita un archivo de configuración único para conectarse a su AWS cuenta.

Veo «com.aws.greengrass.deployment». `lotJobsHelper: No se encontró ningún trabajo de despliegue. o 'Ya se ha informado del resultado de la implementación'.` en los registros

Si ve `com.aws.greengrass.deployment.IotJobsHelper: No deployment job found.` o `Deployment result already reported.` en sus registros en `/greengrass/v2/logs/greengrass.log`, puede que esté intentando volver a utilizar el mismo archivo de configuración.

Hay varias soluciones:

- Si desea reutilizar el archivo de configuración, haga lo siguiente:

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, seleccione Puertas de enlace Edge.
3. Elija la puerta de enlace SiteWise Edge que desee reutilizar.
4. Seleccione la pestaña Actualizaciones.
5. Seleccione una versión de Publicador diferente y elija Implementar.

Siga los pasos de [Creación de una puerta de enlace para Siemens Industrial Edge](#) para crear un nuevo archivo de configuración.

Aparece el estado «SYNC_FAILED» cuando intento configurar la configuración de la marca de tiempo en un grupo de propiedades de una fuente de datos OPC UA

Cuando AWS IoT SiteWise actualizamos el componente recopilador OPC UA AWS IoT Greengrass en la versión 2.5.0, introdujimos una nueva opción de configuración de marca de tiempo. Puedes usar la marca de tiempo de tu dispositivo o la marca de tiempo del servidor. Las versiones anteriores del componente recopilador OPC UA no admiten esta opción y no se sincronizan.

Hay dos formas de resolver un estado de sincronización defectuoso de una fuente de datos. La forma recomendada es actualizar el componente recopilador IoT SiteWise OPC UA a la versión 2.5.0 o superior. Como alternativa, puedes seguir utilizando la versión anterior del componente recopilador OPC UA si configuras la marca de tiempo en. `Source` Para obtener información sobre cómo actualizar el componente recopilador IoT SiteWise OPC UA, consulte [Actualice la versión de un componente AWS IoT SiteWise](#). Recomendamos utilizar las versiones más recientes de todos los componentes.

Note

Los datos no se interrumpen cuando se produce un error en el estado de sincronización de una fuente de datos. Los datos de origen siguen fluyendo hacia AWS IoT SiteWise. La configuración simplemente no se sincroniza con el componente recopilador IoT SiteWise OPC UA de su AWS IoT Greengrass V2 implementación.

Para cambiar la configuración de la marca de tiempo de un grupo de propiedades

1. Vaya a la [consola de AWS IoT SiteWise](#).

2. En el panel de navegación, seleccione Puertas de enlace Edge.
3. Seleccione la puerta de enlace que desee editar.
4. En la sección Fuentes de datos, seleccione la fuente de datos con el estado de sincronización fallida y elija Editar.
5. Expanda Configuración avanzada y, a continuación, expanda Configuración de grupo.
6. En Timestamp, selecciona Fuente. Al seleccionar Fuente, se elimina la `timestampToReturn` propiedad de la configuración. Esta configuración permite recopilar la marca de tiempo de la fuente de datos de su dispositivo de forma predeterminada, lo que permite que la fuente de datos se sincronice con el componente recopilador SiteWise OPC UA de IoT.
7. Seleccione Save.

Los tipos de datos convertidos no se incluyen

Si aparece un error al convertir tipos de datos OPC UA no compatibles en cadenas AWS IoT SiteWise, puede deberse a varios motivos:

- El tipo de datos que intenta convertir es un tipo de datos complejo. No se admite en tipos de datos complejos.
- Al utilizar Destinos como AWS IoT SiteWise almacenado en búfer con Amazon S3, el valor completo de la cadena se conserva en los archivos que se envían a un bucket de Amazon S3. Al introducir datos más adelante AWS IoT SiteWise, se rechazan los valores de cadena completa de más de 1024 bytes.

Problemas con el almacén de confianza

Si encuentra problemas relacionados con los almacenes de confianza en SiteWise Edge, tenga en cuenta los siguientes pasos de solución de problemas:

- Compruebe que el certificado de CA AWS IoT Greengrass raíz esté presente y tenga el formato correcto en los almacenes de confianza adecuados
- Asegúrese de que la KeyStore contraseña de Java esté configurada correctamente y de que los componentes de SiteWise Edge puedan acceder a ella
- Compruebe que todos los certificados personalizados (como los de los proxies HTTPS) estén en el formato correcto (normalmente PEM) y que se hayan importado correctamente a los almacenes de confianza

- Confirme que los almacenes de confianza tengan los permisos de archivo correctos y que los procesos de Edge puedan acceder a ellos SiteWise
- Revise los registros de SiteWise Edge para ver si hay algún error SSL/TLS relacionado, que pueda indicar problemas con el almacén de confianza
- Pruebe SSL/TLS las conexiones de forma independiente mediante herramientas como las `openssl` de verificación del funcionamiento del almacén de confianza

Problemas de instalación con proxy

Si encuentra problemas durante el proceso de configuración del proxy, tenga en cuenta los siguientes pasos de solución de problemas:

- Compruebe que la URL del proxy tenga el formato correcto e incluya el esquema adecuado (`http://`/`https://`)
- Asegúrese de que todas las credenciales de proxy estén codificadas en URL si contienen caracteres especiales
- Confirme que la lista de usuarios sin proxy incluya todas las direcciones locales y los puntos de conexión del servicio necesarios AWS
- En el caso de los proxies HTTPS, compruebe que el certificado de CA proporcionado esté en formato PEM
- Revise los registros de instalación para ver si hay mensajes de error específicos que puedan indicar el origen del problema
- Pruebe la conexión proxy de forma independiente para asegurarse de que funciona correctamente

Solución de problemas de la aplicación AWS IoT SiteWise Edge en Siemens Industrial Edge

Para solucionar los problemas de la aplicación AWS IoT SiteWise Edge en su Siemens Industrial Edge dispositivo, puede acceder a los registros de la aplicación a través de los portales Siemens Industrial Edge Management o Siemens Industrial Edge dispositivos (IED). Para obtener más información, consulte [Downloading Logs](#) en la documentación de Siemens.

Mis datos no aparecen en AWS IoT SiteWise

- Asegúrese de que no hay problemas con sus usuarios de Databus y de que el icono de marca de verificación de Databus_Configuration esté verde en lugar de gris.
- Es posible que no esté utilizando Siemens Industrial Edge Management en una versión que contenga Secure Storage. Actualice su versión de Siemens OS. Para obtener más información, consulte [Siemens Secure Storage en la aplicación AWS IoT SiteWise Edge](#).

Veo «Falta el archivo de configuración AWS_REGION» en los registros

Si aparece Config file missing AWS_REGION en los registros de Siemens, el JSON del archivo de configuración está dañado. Deberá crear un nuevo archivo de configuración. Siga los pasos de [Creación de una puerta de enlace para Siemens Industrial Edge](#) para crear un nuevo archivo de configuración.

Aparece el mensaje de error «No está sincronizado» en la configuración de la puerta de enlace Edge

Si aparece un mensaje de Out of sync error en la Siemens Industrial Edge puerta de enlace una vez completada la implementación, significa que el componente IoT SiteWise Publisher no está sincronizado con la puerta de enlace. El componente de SiteWise editor de IoT funciona en segundo plano en las Siemens Industrial Edge pasarelas para proporcionar la funcionalidad de temas MQTT. Hemos actualizado Siemens Industrial Edge las pasarelas para utilizar el espacio de nombres de capacidades en lugar de `hacerloiotstewise:publisher:3.iotsitewise:publisher:2`. Puede actualizar a la última versión del editor para resolver este problema.

Para actualizar a la última versión del SiteWise editor de IoT

1. Vaya a la [consola de AWS IoT SiteWise](#).
2. En el panel de navegación, seleccione Puertas de enlace Edge.
3. Seleccione la Siemens Industrial Edge puerta de enlace que desee editar.
4. En la sección de capacidades de Edge, selecciona Ver versiones de software.
5. Seleccione la versión más reciente del SiteWise editor de IoT en el menú desplegable Publicador.
6. Seleccione Listo.

Solución de problemas de integraciones de código abierto en Edge

En esta sección, se proporcionan soluciones a los problemas más comunes que pueden surgir al integrar herramientas de código abierto con SiteWise Edge.

Note

Node-RED®, InfluxDB® y Grafana® no son vendedores ni proveedores de Edge. SiteWise

Problemas de conectividad

Node-RED no puede conectarse al broker MQTT

Compruebe que el broker MQTT esté funcionando y sea accesible en el puerto especificado. Compruebe la configuración de la red y asegúrese de que la dirección del intermediario sea correcta.

Para comprobar el estado del corredor de MQTT, ejecute:

```
docker ps | grep emqx
```

Errores de conexión de InfluxDB

Asegúrese de que su token de autenticación sea válido y de que haya especificado los nombres de organización y bucket correctos. Compruebe que InfluxDB esté en ejecución y sea accesible.

Para verificar el estado de InfluxDB, ejecute:

```
curl -I http://localhost:8086
```

Grafana no puede conectarse a InfluxDB

Compruebe que la configuración de la fuente de datos de InfluxDB en Grafana sea correcta, incluida la URL, el token de autenticación, la organización y el depósito.

Problemas de flujo de datos

No aparece ningún dato en AWS IoT SiteWise

Compruebe que el alias de su propiedad en el flujo de Node-RED coincide con el formato esperado. Compruebe que la estructura temática de MQTT sea correcta y que la puerta de enlace SiteWise Edge esté configurada correctamente para recibir datos del broker de MQTT.

No hay datos de SiteWise Edge almacenados en InfluxDB

Compruebe que el flujo de retención de Node-RED esté configurado correctamente y que el nodo de grabación de InfluxDB tenga los ajustes de cubeta y medición adecuados. Compruebe si hay errores en el resultado de depuración de Node-RED.

Errores de formato de datos

Asegúrese de que sus funciones de transformación de datos conviertan correctamente los datos entre formatos. Utilice los nodos de depuración de Node-RED para inspeccionar los datos en cada etapa del flujo.

Problemas de rendimiento

Uso elevado de CPU o memoria

Supervise el uso de los recursos y ajuste la configuración de sus componentes según sea necesario. Considere reducir la frecuencia de recopilación de datos o implementar un filtrado de datos para reducir la carga de procesamiento.

Para supervisar el uso de los recursos, ejecute:

```
docker stats
```

Carga lenta del panel de Grafana

Optimice sus consultas de InfluxDB y considere agregar límites de rango de tiempo a los paneles de su panel de control. Reduzca la cantidad de puntos de datos que se muestran mediante el uso de las funciones de agregación adecuadas.

Registro y diagnóstico

Para solucionar problemas, compruebe los registros de cada componente:

Registros de Node-RED

Vea los registros en la consola de Node-RED o ejecute:

```
docker logs node-red
```

Registros de InfluxDB

Acceda a los registros ejecutando:

```
docker logs influxdb
```

Registros de Grafana

Para ver los registros, ejecute:

```
docker logs grafana
```

SiteWise Registros de Edge

Compruebe los registros de la puerta de enlace SiteWise Edge para ver si hay problemas de conexión con MQTT y de procesamiento de datos. Para obtener más información, consulte [Solución de problemas de una puerta de enlace SiteWise Edge](#).

Solución de problemas AWS IoT Greengrass

Para encontrar soluciones a muchos problemas al configurar o implementar tu puerta de enlace SiteWise Edge AWS IoT Greengrass, consulta [Solución de problemas AWS IoT Greengrass](#) en la Guía para AWS IoT Greengrass desarrolladores.


Solucionar problemas de un portal AWS IoT SiteWise

Solucione los problemas habituales de sus portales. AWS IoT SiteWise

Los usuarios y los administradores no pueden acceder al portal AWS IoT SiteWise

Si los usuarios o los administradores no pueden acceder a su AWS IoT SiteWise portal, es posible que tenga permisos restringidos en las políticas adjuntas AWS Identity and Access Management (IAM) que impiden iniciar sesión correctamente.

Consulte los siguientes ejemplos de políticas de IAM que provocan errores de inicio de sesión:

 Note

Cualquier política de IAM asociada que incluya un elemento "Condition" provoca un error de inicio de sesión.

Ejemplo 1: La condición aquí es una IP limitada, lo cual causa un fallo de inicio de sesión.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iotsitewise:DescribePortal"
      ],
      "Resource": "*",
      "Condition": {
        "IpAddress": {
          "aws:SourceIp": [
            "203.0.113.0/24"
          ]
        }
      }
    }
  ]
}
```

Ejemplo 2: La condición aquí es una etiqueta incluida, lo cual causa un fallo de inicio de sesión.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
```

```
{
  "Effect": "Allow",
  "Action": [
    "iotsitewise:DescribePortal"
  ],
  "Resource": "*",
  "Condition": {
    "StringLike": {
      "aws:ResourceTag/project": "*"
    }
  }
}
```

Al añadir usuarios o administradores al portal, evite crear políticas de IAM que restrinjan los permisos de los usuarios, como IP limitada. Las políticas adjuntas con permisos restringidos no podrán conectarse al AWS IoT SiteWise portal.

Solucionar problemas con una acción de AWS IoT SiteWise regla

Para solucionar los problemas de AWS IoT SiteWise la acción de la regla AWS IoT Core, puede realizar uno de los siguientes procedimientos:

- Configurar Amazon CloudWatch Logs
- Configuración de una acción de error de republicación de su regla

A continuación, compare los mensajes de error con los errores de este tema para solucionar el problema.

Temas

- [Configure AWS IoT Core los registros](#)
- [Configuración de una acción de error de volver a publicar](#)
- [Solución de problemas de reglas](#)
- [Solución de problemas de una regla \(AWS IoT SiteWise\)](#)
- [Solución de problemas de una regla \(DynamoDB\)](#)

Configure AWS IoT Core los registros

Puede configurarlo AWS IoT para registrar varios niveles de información en los CloudWatch registros.

Para configurar los CloudWatch registros y acceder a ellos

1. Para configurar el registro AWS IoT Core, consulte [Supervisión con CloudWatch registros](#) en la Guía para AWS IoT desarrolladores.
2. Vaya a la [consola de CloudWatch](#).
3. En el panel de navegación, seleccione Grupos de registro.
4. Elija el grupo AWSIoTde registros.
5. Elija una secuencia de registro reciente. De forma predeterminada, CloudWatch muestra primero el flujo de registro más reciente.
6. Elija una entrada de registro para expandir el mensaje de registro. Su entrada de registro podría parecerse a la siguiente captura de pantalla.

The screenshot shows the AWS CloudWatch console interface. The breadcrumb navigation is: CloudWatch > Log Groups > AWSIoTLogs > 9ca6614a-00fc-4f9e-8100-5c2a34918e90_123456789012_0. The interface includes a search bar for events, a filter dropdown set to 'all', and a date range selector for '2020-02-10 (19:36:11)'. Below the search bar is a table with columns 'Time (UTC +00:00)' and 'Message'. A log entry is expanded, showing a timestamp of '2020-02-11 19:36:11' and a message starting with '2020-02-11 19:36:11.823 TRACEID:d4cd3bd0-ac41-cd4a-4f59-74a242ec70e6 PRINCIPALID:AIDAZ2YMUHYHIEDEL3VA3 [ERROR] EVENT:IotSiteWise'. The message body contains details about an 'IotSiteWiseActionFailure' with a specific error code and message: 'InvalidRequestException, Message: Property value does not match data type DOUBLE'. The console also shows 'No older events found at the moment. Retry.' and 'No newer events found at the moment. Retry.' messages.

7. Compare los mensajes de error con los errores de este tema para solucionar el problema.

Configuración de una acción de error de volver a publicar

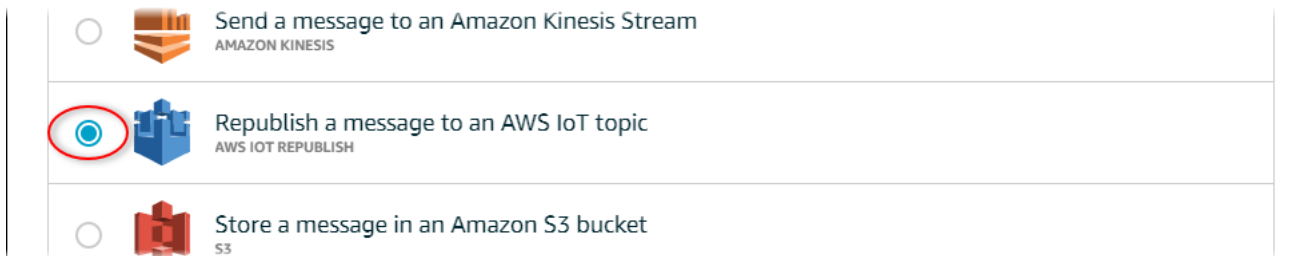
Puede configurar una acción de error en la regla para controlar los mensajes de error. En este procedimiento, configure la acción de la regla Republicar como una acción de error para ver los mensajes de error en el cliente de prueba de MQTT.

Note

La acción de error de republicar genera solo el equivalente de los registros de nivel ERROR. Si desea registros más detallados, debe [configurar CloudWatch](#) los registros.

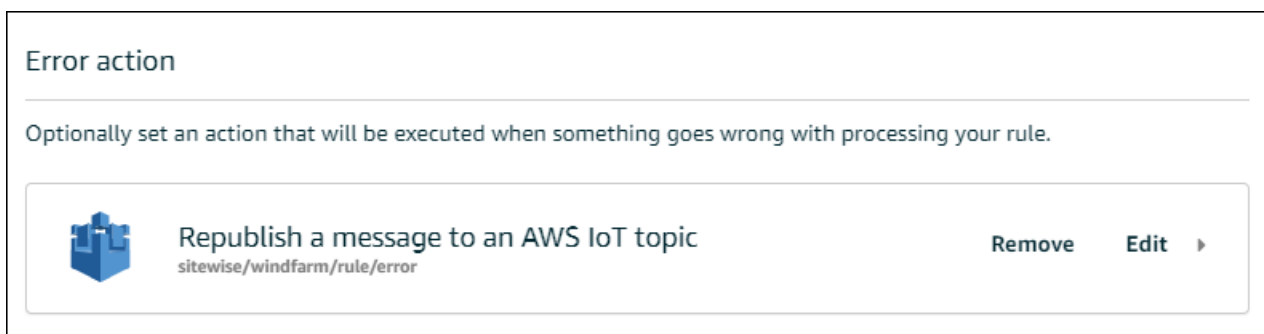
Para añadir una acción de error de republicar a una regla

1. Vaya a la [consola de AWS IoT](#).
2. En el panel de navegación izquierdo, elija Act (Acción) y, a continuación, elija Rules (Reglas).
3. Elija su regla.
4. En Error action (Acción de error), elija Add action (Agregar acción).
5. Elige Volver a publicar un mensaje en un tema. AWS IoT



6. Seleccione Configure action (Configurar acción) en la parte inferior de la página.
7. En Tema, introduce un tema único (por ejemplo, **sitewise/windfarm/rule/error**). AWS IoT Core volverá a publicar los mensajes de error relacionados con este tema.
8. Elija Seleccionar para conceder AWS IoT Core acceso y realizar la acción de error.
9. Elija Select (Seleccionar) junto al rol creado para la regla.
10. Elija Update role (Actualizar rol) para añadir los permisos adicionales al rol.
11. Seleccione Agregar acción.

La acción de error de la regla debería ser similar a la siguiente captura de pantalla.



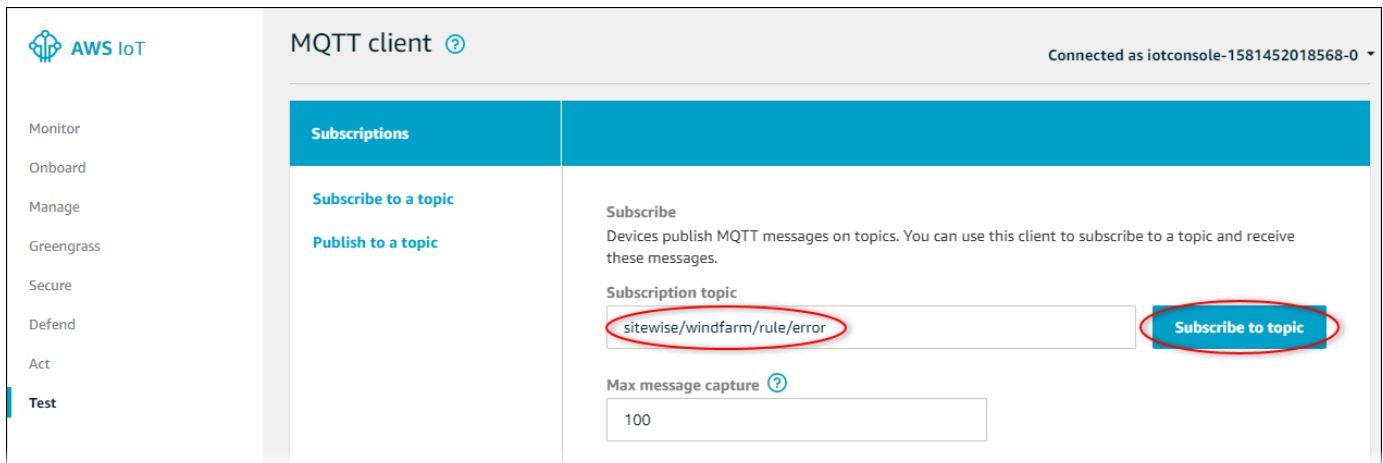
12. Selecciona la flecha hacia atrás situada en la esquina superior izquierda de la consola para volver a la página de inicio de la AWS IoT consola.

Después de configurar la acción de error de Republish (Volver a publicar), puede ver los mensajes de error en el cliente de prueba de MQTT en AWS IoT Core.

En el siguiente procedimiento, se suscribe al tema de error en el cliente de prueba de MQTT. En el cliente de prueba de MQTT, puede recibir mensajes de error de su regla para solucionar el problema.

Para suscribirse al tema de acción de error

1. Vaya a la [consola de AWS IoT](#).
2. En la página de navegación izquierda, elija Test (Pruebas) para abrir el cliente de pruebas de MQTT.
3. En el campo Subscription topic (Tema de suscripción), escriba el tema de error que configuró anteriormente (por ejemplo, **sitewise/windfarm/rule/error**) y elija Subscribe to topic (Suscribirse al tema).



4. Observe si aparecen mensajes de error y, a continuación, expanda la matriz failures en cualquier mensaje de error.

A continuación, compare los mensajes de error con los errores de este tema para solucionar el problema.

Solución de problemas de reglas

Utilice la siguiente información para solucionar problemas de reglas.

Problemas

- [Error: el miembro debe estar dentro de los 604800 segundos antes y 300 segundos después de la marca de tiempo actual](#)
- [Error: el valor de la propiedad no coincide con el tipo de datos <type>](#)
- [Error: Usuario: <role-arn>no está autorizado a realizar: iotsitewise: on resource BatchPutAssetPropertyValue](#)
- [Error: iot.amazonaws.com no puede funcionar: sts: on resource: AssumeRole <role-arn>](#)
- [Información: no se envió ninguna solicitud. PutAssetPropertyValueEntries estaba vacío después de realizar las plantillas de sustitución.](#)

Error: el miembro debe estar dentro de los 604800 segundos antes y 300 segundos después de la marca de tiempo actual

Su marca de tiempo es anterior a 7 días o posterior a 5 minutos, en comparación con la hora epoch de Unix. Pruebe lo siguiente:

- Verifique que la marca de tiempo esté en formato de tiempo de Unix (UTC). Si proporciona una marca temporal con una zona horaria diferente, recibirá este error.
- Compruebe que la marca de tiempo esté en segundos. AWS IoT SiteWise espera que las marcas de tiempo se dividan en tiempo en segundos (en el tiempo de época de Unix) y compensadas en nanosegundos.
- Compruebe que está cargando datos cuya fecha no sea anterior a 7 días.

Error: el valor de la propiedad no coincide con el tipo de datos <type>

Una entrada de la acción de regla tiene un tipo de datos diferente al de la propiedad del activo de destino. Por ejemplo, la propiedad del activo de destino es una DOUBLE y el tipo de datos seleccionado es Integer (Entero) o ha transferido el valor en integerValue. Pruebe lo siguiente:

- Si configura la regla desde la AWS IoT consola, compruebe que ha elegido el tipo de datos correcto para cada entrada.
- Si configuras la regla desde la API o AWS Command Line Interface (AWS CLI), compruebe que tu value objeto utilice el campo de tipo correcto (por ejemplo, doubleValue para una DOUBLE propiedad).

Error: Usuario: <role-arn>no está autorizado a realizar: iotsitewise: on resource BatchPutAssetPropertyValue

La regla no está autorizada a acceder a la propiedad del activo de destino o la propiedad del activo de destino no existe. Pruebe lo siguiente:

- Compruebe que su alias de propiedad es correcto y que tiene una propiedad de activo con el alias de propiedad dado. Para obtener más información, consulte [Gestione los flujos de datos para AWS IoT SiteWise](#).
- Compruebe que la regla tiene un rol y que el rol permite los permisos `iotsitewise:BatchPutAssetPropertyValue` a la propiedad del activo de destino, por ejemplo, a través de la jerarquía del activo de destino. Para obtener más información, consulte [Conceda AWS IoT el acceso requerido](#).

Error: iot.amazonaws.com no puede funcionar: sts: on resource: AssumeRole <role-arn>

Tu usuario no está autorizado a asumir la función que figura en tu regla en (IAM). AWS Identity and Access Management

Compruebe que su usuario tenga permiso `iam:PassRole` para asumir el rol en su regla. Para obtener más información, consulte [Permisos de pase de roles](#) en la Guía para desarrolladores de AWS IoT .

Información: no se envió ninguna solicitud. `PutAssetPropertyValueEntries` estaba vacío después de realizar las plantillas de sustitución.

Note

Este mensaje es un registro de nivel INFO.

Su solicitud debe tener como mínimo una entrada con todos los parámetros requeridos.

Compruebe que los parámetros de la regla, incluidas las plantillas de sustitución, den como resultado valores no vacíos. Las plantillas de sustitución no pueden acceder a los valores definidos en las cláusulas AS de la instrucción de consulta de la regla. Para obtener más información, consulte [Plantillas de sustitución](#) en la Guía para desarrolladores de AWS IoT .

Solución de problemas de una regla (AWS IoT SiteWise)

Siga los pasos de este procedimiento para solucionar los problemas de la regla si los datos de uso de la CPU y la memoria no aparecen AWS IoT SiteWise como se esperaba. En este procedimiento, configure la acción de la regla Republicar como una acción de error para ver los mensajes de error en el cliente de prueba de MQTT. También puedes configurar el registro en los CloudWatch registros para solucionar los problemas. Para obtener más información, consulte [Solucionar problemas con una acción de AWS IoT SiteWise regla](#).

Para añadir una acción de error de republicar a una regla

1. Vaya a la [consola de AWS IoT](#).
2. En el panel de navegación de la izquierda, elija Redirección de mensajes y, a continuación, seleccione Reglas.
3. Elija la regla que creó anteriormente y elija Editar.
4. En Acción de error opcional, elija Agregar acción de error.
5. Selecciona Volver a publicar un mensaje en un AWS IoT tema.
6. En Tema, introduce la ruta del error (por ejemplo, **sitewise/rule/tutorial/error**). AWS IoT Core volverá a publicar los mensajes de error relacionados con este tema.
7. Elija el rol que creó anteriormente (por ejemplo, SiteWiseTutorialDeviceRuleRole).
8. Elija Actualizar.

Después de configurar la acción de error de Republish (Volver a publicar), puede ver los mensajes de error en el cliente de prueba de MQTT en AWS IoT Core.

En el siguiente procedimiento, se suscribe al tema de error en el cliente de prueba de MQTT.

Para suscribirse al tema de acción de error

1. Vaya a la [consola de AWS IoT](#).
2. En la página de navegación izquierda, elija cliente de pruebas de MQTT para abrir el cliente de pruebas de MQTT.
3. En el campo Filtro por temas, introduzca **sitewise/rule/tutorial/error** y seleccione Suscribirse.

Cuando aparecen mensajes de error, vea la matriz `failures` en cualquier mensaje de error para diagnosticar problemas. Para obtener más información acerca de posibles problemas y cómo resolverlos, consulte [Solucionar problemas con una acción de AWS IoT SiteWise regla](#).

Si no aparecen errores, compruebe que la regla esté habilitada y que se suscribió al mismo tema que configuró en la acción de error de Republish. Si los errores siguen sin aparecer después de hacerlo, compruebe que el script del dispositivo se esté ejecutando y actualizando correctamente la sombra del dispositivo.

Note

También puedes suscribirte al tema de actualizaciones alternativas de tu dispositivo para ver la carga útil que analiza tu AWS IoT SiteWise acción. Para ello, suscríbete al siguiente tema.

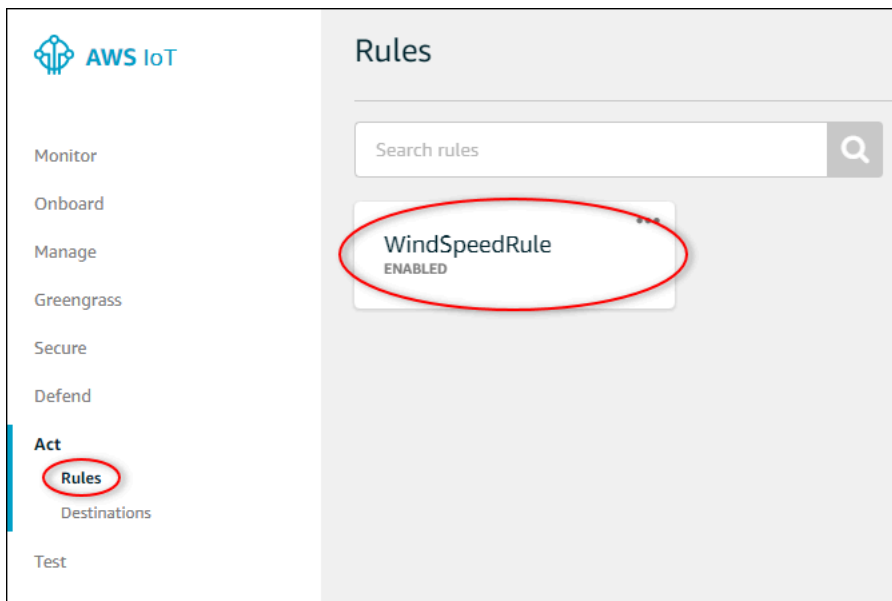
```
$aws/things/+/shadow/update/accepted
```

Solución de problemas de una regla (DynamoDB)

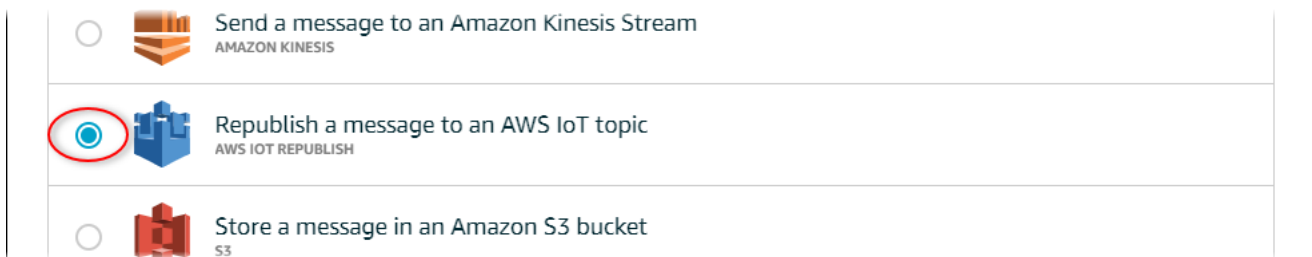
Siga los pasos de este procedimiento para solucionar problemas de la regla si los datos de activos de demostración no aparecen en la tabla de DynamoDB como se esperaba. En este procedimiento, configure la acción de la regla Republish como una acción de error para ver los mensajes de error en el cliente de prueba de MQTT. También puede configurar el registro en CloudWatch Logs para solucionar problemas. Para obtener más información, consulte [Supervisión con CloudWatch registros](#) en la Guía para AWS IoT desarrolladores.

Para añadir una acción de error de republicar a una regla

1. Vaya a la [consola de AWS IoT](#).
2. En el panel de navegación izquierdo, elija Act (Acción) y, a continuación, elija Rules (Reglas).
3. Elija la regla que creó anteriormente.




4. En Error action (Acción de error), elija Add action (Agregar acción).
5. Selecciona Volver a publicar un mensaje en un AWS IoT tema.





6. Seleccione Configure action (Configurar acción) en la parte inferior de la página.
7. En Tema, escriba **windspeed/error**. AWS IoT Core volverá a publicar los mensajes de error relacionados con este tema.

Configure action

 **Republish a message to an AWS IoT topic**
AWS IOT REPUBLISH

This action will republish the message to another AWS IoT topic.

*Topic 

Quality of Service 
 0 - The message is delivered zero or more times.
 1 - The message is delivered one or more times.

Choose or create a role to grant AWS IoT access to perform this action.

No role selected Create Role Select

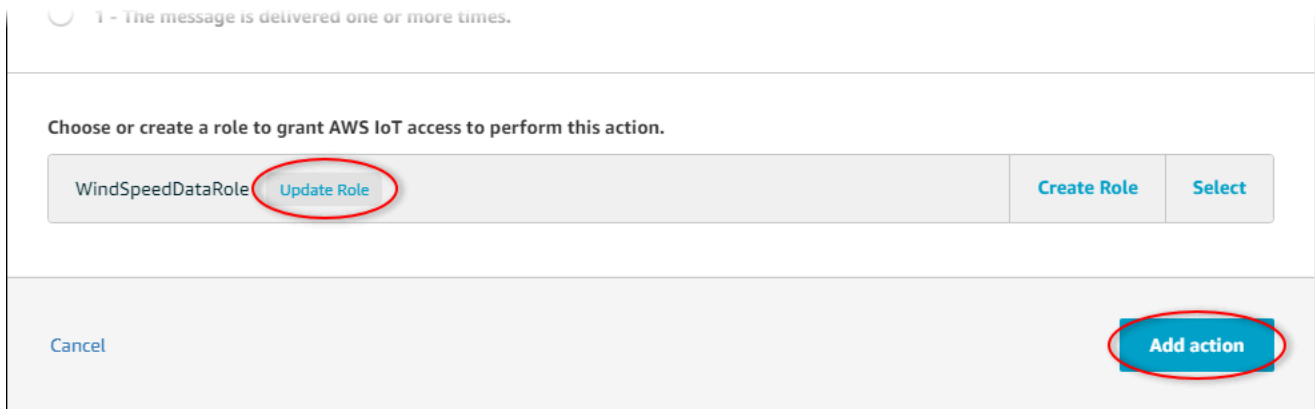
Cancel Add action

8. Elija Seleccionar para conceder acceso a AWS IoT Core y realizar la acción de error utilizando el rol que creó anteriormente.
9. Elija Select (Seleccionar) junto a su rol.

Choose or create a role to grant AWS IoT access to perform this action.

No role selected	Refresh	Create Role	Close
<input type="text" value="Search for IAM roles"/>			
WindSpeedDataRole	Select		

10. Elija Update role (Actualizar rol) para añadir los permisos adicionales al rol.



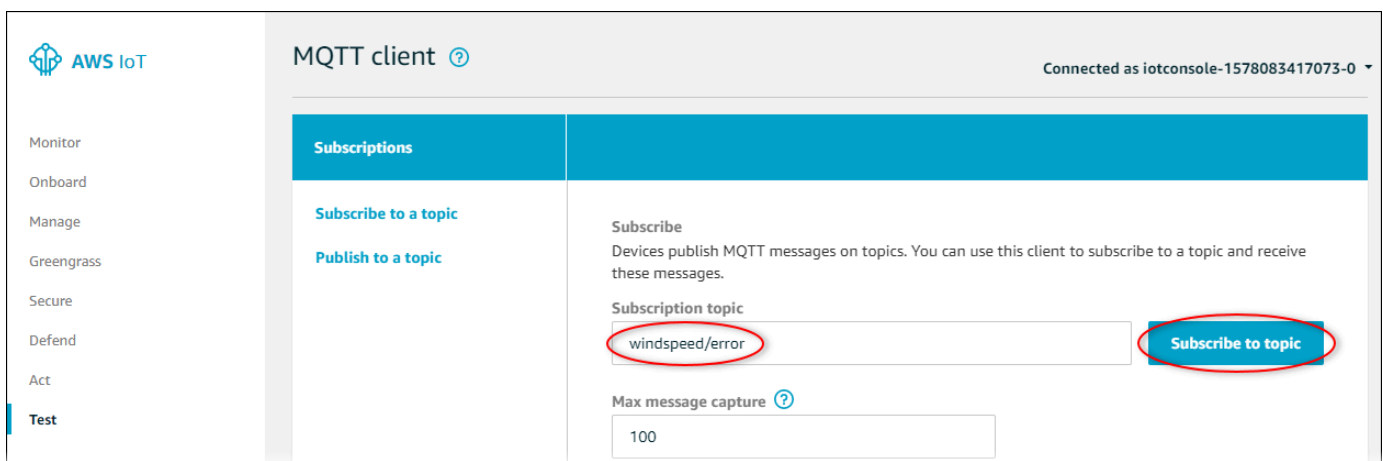
11. Elija Add action (Añadir acción) para terminar de añadir la acción de error.
12. Selecciona la flecha hacia atrás situada en la esquina superior izquierda de la consola para volver a la página de inicio de la consola AWS IoT Core.

Tras configurar la acción de error de republicación, podrá ver los mensajes de error en el cliente de prueba de MQTT de Core. AWS IoT

En el siguiente procedimiento, se suscribe al tema de error en el cliente de prueba de MQTT.

Para suscribirse al tema de acción de error

1. En la página de navegación izquierda de la consola AWS IoT Core, selecciona Probar.
2. En el campo Subscription topic (Tema de suscripción) escriba **windspeed/error** y elija Subscribe to topic (Suscribirse al tema).



3. Observe si aparecen mensajes de error y explore la matriz de **failures** en un mensaje de error para diagnosticar los siguientes problemas comunes:
 - Errores tipográficos en la instrucción de consulta de regla

- Permisos de rol insuficientes

Si no aparecen errores, compruebe que la regla esté habilitada y que se suscribió al mismo tema que configuró en la acción de error de Republish. Si los errores siguen sin aparecer, comprueba que los activos de demostración del parque eólico siguen existiendo y que has activado las notificaciones sobre las propiedades de la velocidad del viento. Si sus activos de demostración han caducado o han desaparecido AWS IoT SiteWise, puede crear una nueva versión de demostración y actualizar la declaración de consulta de reglas para que refleje el modelo de activo y la propiedad IDs actualizados.

Solución de operaciones de importación y exportación masivas

Para gestionar y diagnosticar los errores producidos durante un trabajo de transferencia, consulta la AWS IoT TwinMaker GetMetadataTransferJobAPI:

1. Después de crear y ejecutar un trabajo de transferencia, llama a la GetMetadataTransferJobAPI:

```
aws iottwinmaker get-metadata-transfer-job \  
--metadata-transfer-job-id your_metadata_transfer_job_id \  
--region us-east-1
```

2. El estado del trabajo cambia a uno de los siguientes estados:
 - COMPLETED
 - CANCELLED
 - ERROR
3. La GetMetadataTransferJobAPI devuelve un [MetadataTransferJobProgress](#) objeto.
4. El MetadataTransferJobProgress objeto contiene los siguientes parámetros:
 - failedCount: indica el recuento de activos que han fallado durante el proceso.
 - skippedCount: indica el recuento de activos que se han omitido durante el proceso de transferencia.
 - succeededCount: indica el recuento de activos transferidos correctamente durante el proceso.
 - totalCount: indica el recuento total de activos implicados en el proceso de transferencia.

5. Además, la llamada a la API devuelve un elemento `reportUrl`, que contiene una URL prefirmada. Si su trabajo de transferencia contiene errores que deban investigarse, puede descargar un informe de errores completo en esta URL.

AWS IoT SiteWise puntos finales y cuotas

En las siguientes secciones se describen los puntos finales y las cuotas de AWS IoT SiteWise.

Temas

- [AWS IoT SiteWise puntos finales](#)
- [AWS IoT SiteWise cuotas](#)

AWS IoT SiteWise puntos finales

La Referencia general de AWS guía enumera los AWS IoT SiteWise puntos finales de un. Cuenta de AWS Para obtener más información, consulte [AWS IoT SiteWise endpoints and quotas](#) en la Guía de Referencia general de AWS .

AWS IoT SiteWise cuotas

En las siguientes tablas se describen las cuotas en. AWS IoT SiteWise Para obtener más información sobre cuotas y cómo solicitar aumentos de cuota, consulte [service quotas de AWS](#) en la página Referencia general de AWS. Para obtener más información sobre AWS IoT SiteWise las cuotas, consulte [las cuotas de AWS IoT SiteWise servicio](#) en Referencia general de AWS.

Cuotas de AWS IoT SiteWise activos y modelos de activos

Recurso	Description (Descripción)	Cuota	Ajustable
Número de modelos de activos en cada uno Región de AWS para cada Cuenta de AWS	El número máximo de modelos de activos que puede crear en un Región de AWS para un Cuenta de AWS.	10000	Sí
Número de activos en cada modelo de activos	El número máximo de activos que puede	10 000	Sí

Recurso	Description (Descripción)	Cuota	Ajustable
	crear para cada modelo de activos.		
Número de activos secundarios en cada activo principal	El número máximo de activos secundarios que puede asociar a un activo principal.	2000	Sí
Profundidad del árbol de jerarquía del modelo de activos	La profundidad máxima del árbol jerárquico de activos para un modelo de activos.	30	Sí
Número de definiciones jerárquicas en cada modelo de activos	El número máximo de definiciones jerárquicas que puede tener un modelo de activos.	30	Sí
Número de propiedades en el nivel raíz de cada modelo de activos	El número máximo de <code>assetModelProperties</code> para cada modelo de activos. Este recuento no incluye <code>compositeModelProperties</code> . Esta cuota también se aplica a cualquier activo único creado a partir de este modelo de activos.	500	Sí

Recurso	Description (Descripción)	Cuota	Ajustable
Número de propiedades en un modelo de activos	Número máximo de propiedades de un modelo de activos de tipo ASSET_MODEL o COMPONENT_MODEL . Este número se determina combinando las propiedades del modelo de activos raíz y cualquier modelo compuesto incluido component-model-based o en línea. Esta cuota también se aplica a cualquier activo único creado a partir de este modelo de activos.	5000	Sí
Número de propiedades de cada modelo compuesto	Número máximo de propiedades permitidas para modelos compuestos. Además, número máximo de propiedades permitidas para un modelo de activos de tipo COMPONENT_MODEL .	100	Sí

Recurso	Description (Descripción)	Cuota	Ajustable
Profundidad del árbol de propiedades en un modelo de activos	Por ejemplo, un modelo con una propiedad de transformación C que consuma una propiedad de transformación B, que a su vez consuma una propiedad de medición A tendrá una profundidad de 3.	10	No
Número de modelos de activos en cada árbol jerárquico	El número máximo de modelos de activos que puede incluir en un único árbol jerárquico.	100	Sí

Recurso	Description (Descripción)	Cuota	Ajustable
Número de propiedad es directamente dependientes de un modelo de activos	Esta cuota limita el número de propiedad es que pueden depender directamente de una sola propiedad, tal y como se define en las expresiones de fórmulas de propiedad. El número de propiedades dependientes debe ser mayor que el número de propiedad es directamente dependientes de un modelo de activos. Solicite un aumento de ambas cuotas si hay propiedad es que dependen más directamente que propiedades dependientes en un modelo de activos.	20	Sí

Recurso	Description (Descripción)	Cuota	Ajustable
Número de propiedades dependientes en un modelo de activos	Esta cuota limita el número de propiedades que pueden depender directa o indirectamente de una sola propiedad, tal y como se define en las expresiones de fórmulas de propiedad.	30	No
Número de modelos compuestos en un modelo de activos	El número máximo de modelos compuestos que puede tener en un único modelo de activos.	50	Sí
Profundidad del modelo compuesto	La profundidad máxima del árbol del modelo compuesto en cada modelo de activo, incluidos los modelos en línea y component-model-based compuestos.	2	Sí

Recurso	Description (Descripción)	Cuota	Ajustable
Número de modelos de activos únicos que utilizan el mismo modelo de componentes	El número máximo de modelos de activos únicos que tienen al menos un modelo component-model-based compuesto que hace referencia directamente a un modelo de activos específico del tipo COMPONENT_MODEL.	20	Sí
Número de variables de propiedad en una expresión de fórmula de propiedad	Por ejemplo, hay dos variables de propiedad, power y temp, en la expresión <code>avg(power) + max(temp)</code> . Esto también se aplica a los resultados del cálculo de transformaciones.	10	No
Número de funciones en una expresión de fórmula de propiedad	Por ejemplo, hay dos funciones, avg y max, en la expresión <code>avg(power) + max(temp)</code> .	10	No

Cuotas de AWS IoT SiteWise interfaces

Cuotas de AWS IoT SiteWise interfaces

Recurso	Description (Descripción)	Cuota	Ajustable
Número de interfaces en cada una Región de AWS para cada Cuenta de AWS	El número máximo de interfaces que puede crear en un Región de AWS para un Cuenta de AWS.	100	Sí
Número de propiedades en cada interfaz	El número máximo de propiedades que se pueden definir en una interfaz.	200	Sí
Número de modelos de activos únicos que utilizan el mismo modelo de componentes	El número máximo de modelos de activos únicos que pueden usar el mismo modelo de interfaz.	500	Sí
Número de interfaces en cada árbol jerárquico	El número máximo de interfaces que puede incluir en un único árbol jerárquico.	100	Sí
Profundidad del árbol jerárquico de la interfaz	La profundidad máxima del árbol jerárquico de la interfaz.	30	Sí
Número de definiciones de jerarquías en cada interfaz	El número máximo de definiciones de jerarquías que puede tener en una interfaz.	20	Sí

Cuotas de datos AWS IoT SiteWise de propiedades de activos

Recurso	Description (Descripción)	Cuota	Ajustable
Velocidad de solicitudes para operaciones de la API de datos de propiedad de activos	El número máximo de solicitudes de API de datos sobre propiedades de activos por segundo que puedes realizar Región de AWS en cada una de ellas Cuenta de AWS. Esta cuota se aplica a las operaciones de la API, como GetAssetPropertyValue y BatchPutAssetPropertyValue .	1 000	Sí
Número de puntos de datos por segundo para cada calidad de datos de cada activo y propiedad	Esta cuota se aplica al número máximo de puntos de datos timestamp-quality-value (TQV) con la misma marca de tiempo en segundos para cada calidad de datos de cada propiedad del activo. Puede almacenar hasta este número de puntos de datos de calidad buena, calidad incierta y	10	No

Recurso	Description (Descripción)	Cuota	Ajustable
	calidad mala por cada segundo dado para cada propiedad de activo.		
Número de BatchPutAssetPropertyValue entradas ingeridas cada segundo en cada propiedad de activo que no sea un atributo para cada una de ellas. Región de AWS Cuenta de AWS	El número máximo de entradas en cada propiedad de activo sin atributos de todas las fuentes, incluidas las pasarelas BatchPutAssetPropertyValue de SiteWise Edge, las reglas y las llamadas a la API. AWS IoT Core	10	No
Número de BatchPutAssetPropertyValue entradas ingeridas cada segundo para cada propiedad de activo de atributo de cada una. Región de AWS Cuenta de AWS	Esta cuota se aplica a las entradas de cada propiedad de activo de atributo BatchPutAssetPropertyValue procedentes de todas las fuentes, incluidas las pasarelas de SiteWise Edge, AWS IoT Core las reglas y las llamadas a la API.	1	No

Recurso	Description (Descripción)	Cuota	Ajustable
Tasa de puntos de datos ingeridos	El número máximo de puntos de datos timestamp-quality-value (TQV) ingeridos por segundo en cada uno Región de AWS de ellos para un. Cuenta de AWS	5000	Sí
Tasa de solicitudes para BatchGetAssetPropertyAggregates	El número máximo de BatchGetAssetPropertyAggregates solicitudes por segundo que puede realizar Región de AWS en cada una de ellas. Cuenta de AWS	200	Sí
Tasa de solicitudes para BatchGetAssetPropertyValue	El número máximo de BatchGetAssetPropertyValue solicitudes por segundo que puedes realizar Región de AWS en cada una de ellas Cuenta de AWS.	500	Sí

Recurso	Description (Descripción)	Cuota	Ajustable
Tasa de solicitudes para BatchGetAssetPropertyHistory	El número máximo de BatchGetAssetPropertyHistory solicitudes por segundo que puede realizar.	200	Sí
Número de BatchPutAssetPropertyValue entradas ingeridas cada segundo para cada propiedad de activo y para cada una Región de AWS Cuenta de AWS de ellas.	Esta cuota se aplica a las entradas de cada propiedad de activo BatchPutAssetPropertyValue procedentes de todas las fuentes, incluidas las pasarelas de SiteWise Edge, AWS IoT Core las reglas y las llamadas a la API.	10	No
Tasa de GetAssetPropertyAggregates solicitudes y consultas de BatchGetAssetPropertyAggregates entrada para cada propiedad de activo	El número máximo de GetAssetPropertyAggregates solicitudes y BatchGetAssetPropertyAggregates entradas totales para cada propiedad de activo por segundo Región de AWS en cada uno de ellos Cuenta de AWS.	50	No

Recurso	Description (Descripción)	Cuota	Ajustable
Tasa de GetAssetPropertyValue solicitudes y consultas de BatchGetAssetPropertyValue entrada para cada activo (propiedad)	El número máximo de GetAssetPropertyValue solicitudes y BatchGetAssetPropertyValue entradas totales para cada propiedad de activo por segundo Región de AWS en cada una de ellas Cuenta de AWS.	500	No
Tasa de GetAssetPropertyValueHistory solicitudes y consultas de BatchGetAssetPropertyValueHistory entrada para cada activo (propiedad)	El número máximo de GetAssetPropertyValueHistory y solicitudes y BatchGetAssetPropertyValueHistory entradas totales para cada propiedad de activo por segundo Región de AWS en cada una de ellas Cuenta de AWS.	30	No

Recurso	Description (Descripción)	Cuota	Ajustable
Tasa de solicitudes GetInterpolatedAssetPropertyValues	El número máximo de GetInterpolatedAssetPropertyValues solicitudes por segundo que puede realizar Región de AWS en cada una de ellas Cuenta de AWS.	500	Sí
Número de resultados en cada GetInterpolatedAssetPropertyValues solicitud	El número máximo de resultados que se devolverán en una GetInterpolatedAssetPropertyValues solicitud paginada.	10	Sí

Recurso	Description (Descripción)	Cuota	Ajustable
<p>Tasa de puntos de datos recuperados de y GetAssetPropertyValueHistory y BatchGetAssetPropertyValueHistory</p>	<p>La velocidad máxima de bytes (MB/segundo) de los puntos de datos recuperados cada segundo para cada uno de los puntos de datos Región de AWS transversales. Cuenta de AWS GetAssetPropertyValueHistory y BatchGetAssetPropertyValueHistory La carga útil de respuesta evaluada para esta cuota utiliza campos Timestamp-Quality-Value (TQV) para cada punto de datos y redondea el tamaño de los bytes de cada solicitud de API al siguiente incremento de 4 KB.</p> <p>Timestamp-quality-value (TQV) los puntos de datos recuperados cada segundo varían según el tipo de datos:</p>	100	Sí

Recurso	Description (Descripción)	Cuota	Ajustable
	<ul style="list-style-type: none"> • Entero: hasta 5 millones de TQV por segundo • Doble: hasta 4 millones de TQV por segundo • Booleano: hasta 6 millones de TQV por segundo • Cadena: varía en función del tamaño del valor de cada cadena. 		
Número de propiedades que dependen de una sola propiedad dentro de un modelo de activos vinculados	El número máximo de propiedades que dependen directa o indirectamente de una sola propiedad dentro de un modelo de activos vinculados.	10	Sí

Cuotas para las pasarelas SiteWise Edge

Recurso	Description (Descripción)	Cuota	Ajustable
Número de puertas de enlace SiteWise Edge en cada una Región de AWS de ellas Cuenta de AWS	El número máximo de puertas de enlace SiteWise Edge que puede crear en y	100	Sí

Recurso	Description (Descripción)	Cuota	Ajustable
	Región de AWS para un. Cuenta de AWS		
Número de fuentes OPC UA en una SiteWise puerta de enlace Edge	El número máximo de fuentes OPC UA que puede configurar en una puerta de enlace SiteWise Edge.	100	No
Número total de destinos en una puerta de enlace SiteWise Edge	La cantidad máxima de destinos que puede configurar en una puerta de enlace SiteWise Edge.	100	No

Cuotas para AWS IoT SiteWise Monitor

Note

La función SiteWise Supervisar ya no está disponible para los nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte [SiteWise Supervisar el cambio de disponibilidad](#).

Recurso	Description (Descripción)	Cuota	Ajustable
Número de portales en cada uno Región de AWS para un Cuenta de AWS	El número máximo de portales de SiteWise Monitor que puede crear en y Región de AWS para un Cuenta de AWS.	100	Sí

Recurso	Description (Descripción)	Cuota	Ajustable
Número de proyectos en un portal	El número máximo de proyectos que puede crear en un portal de SiteWise Monitor.	100	Sí
Número de paneles en un proyecto	El número máximo de paneles que puede crear dentro de un proyecto en SiteWise Monitor.	100	Sí
Número de activos raíz de un proyecto	El número máximo de activos de nivel superior que puede añadir a un proyecto en SiteWise Monitor.	1	No
Número de visualizaciones en un panel	El número máximo de elementos visuales (como cuadros, gráficos o tablas) que puede añadir a un panel en SiteWise Monitor.	10	Sí
Número de métricas en cada visualización del panel	El número máximo de métricas o puntos de datos que se pueden mostrar en una sola visualización en un panel de control de SiteWise Monitor.	5	Sí

Recurso	Description (Descripción)	Cuota	Ajustable
Número de umbrales para cada visualización del panel	El número máximo de niveles de umbral que puede establecer para cada visualización en un panel de control en SiteWise Monitor.	12	No

Cuotas de importación y exportación AWS IoT SiteWise masivas de metadatos

Recurso	Description (Descripción)	Cuota	Ajustable
Número de trabajos de transferencia de metadatos en cola	Número máximo de trabajos de transferencia de metadatos de PENDING en la cola.	10	Sí
Tamaño del archivo de importación de trabajos de transferencia de metadatos	Tamaño máximo del archivo importado (en MB).	100	Sí
Número de recursos de AWS IoT SiteWise importación en una tarea	El número máximo de recursos de AWS IoT SiteWise importación en un solo trabajo. Un recurso incluye activos y modelos de activos.	5000	Sí
Número de recursos de AWS IoT SiteWise	El número máximo de recursos de AWS IoT	5000	Sí

Recurso	Description (Descripción)	Cuota	Ajustable
exportación en un trabajo	SiteWise exportación en un solo trabajo. Un recurso incluye activos y modelos de activos.		

Cuotas para la importación AWS IoT SiteWise masiva de datos

Recurso	Description (Descripción)	Cuota	Ajustable
Número de trabajos de importación masiva en ejecución	El número máximo de trabajos de importación masiva que se pueden ejecutar simultáneamente.	100	No
Tamaño del archivo CSV	El tamaño máximo del archivo CSV (en GB) de un trabajo de importación masiva.	10	No
Tamaño del archivo parquet sin comprimir	El tamaño máximo de archivo (en MB) para un archivo de parquet sin comprimir en un trabajo de importación masiva.	256 MB	No
Tamaño del archivo CSV para la ingesta almacenada en búfer	El tamaño máximo del archivo CSV (en MB) cuando se utiliza la ingesta en búfer en un	256 MB	No

Recurso	Description (Descripción)	Cuota	Ajustable
	trabajo de importación masiva.		
Tamaño del grupo de filas parquet sin comprimir	El tamaño máximo de un grupo de hileras de parquet sin comprimir.	64 MB	No
Número de medidas únicas en una lima de parquet	El número máximo de medidas únicas en una lima de parquet.	10000	No
Número de días entre la marca de tiempo anterior y la actual para ingesta almacenada en búfer	El número máximo de días entre una marca de tiempo pasada y la fecha de hoy cuando se utiliza la ingestión almacenada en búfer.	30	Sí
Solicita una tarifa para cada uno de ellos CreateBulkImportJobs Región de AWS Cuenta de AWS		10	Sí
Solicita una tarifa ListBulkImportJobs para cada uno Región de AWS de ellos Cuenta de AWS		50	Sí

Recurso	Description (Descripción)	Cuota	Ajustable
Solicita una tarifa DescribeBulkImportJobs para cada uno Región de AWS de ellos Cuenta de AWS		50	Sí

Cuotas de regulación de la API AWS IoT SiteWise Assistant

Recurso	Description (Descripción)	Cuota	Ajustable
Tasa de solicitud de InvokeAssistant operación	El número máximo de transacciones por minuto (TPM) que se pueden realizar en la AWS IoT SiteWise InvokeAssistant API en un Cuenta de AWS. Los límites del TPM se aplican a todas las regiones compatibles y se pueden ajustar en algunas regiones.	10	No

Cuotas de cuotas de detección de anomalías

Recurso	Description (Descripción)	Cuota	Ajustable
Número máximo de modelos computacionales	El número máximo de modelos computacionales.	1 000	Sí
Número máximo de propiedades de entrada para cada modelo computacional	El número máximo de propiedades de entrada que puede tener un modelo computacional.	80	No
Número máximo de TQV en los datos de entrenamiento (después del remuestreo)	El número máximo de TQV en los datos de entrenamiento tras el remuestreo.	1 500 000	No
Número máximo de TQV en los datos de evaluación (después del remuestreo)	El número máximo de TQV en los datos de evaluación tras el remuestreo.	1 500 000	No
Plazo de tiempo mínimo de los datos de entrenamiento	El intervalo de tiempo mínimo para los datos de entrenamiento. Para obtener más información, consulte Comprenda el intervalo de fechas mínimo .	14 días	No
Número máximo de trabajos de ACTIVE	El número máximo de trabajos de ACTIVE formación simultánea	25	Sí

Recurso	Description (Descripción)	Cuota	Ajustable
formación simultáneos	os en un momento dado.		
Tamaño máximo del archivo de datos de etiquetas	El tamaño máximo del archivo de datos de etiquetas.	2 MB	No
Número máximo de inferencias de ACTIVE alta frecuencia	El número máximo de inferencias de ACTIVE alta frecuencia en un momento dado.	250	Sí
Número máximo de inferencias de ACTIVE baja frecuencia	El número máximo de inferencias de ACTIVE baja frecuencia en un momento dado.	500	Sí
Tiempo máximo de duración del ACTIVE entrenamiento	El tiempo máximo durante el que puede tener lugar el ACTIVE entrenamiento.	24 horas	Sí

Cuotas de AWS IoT SiteWise SQL

Recurso	Description (Descripción)	Cuota	Ajustable
Tasa de solicitud de API SiteWise ExecuteQuery	El número máximo de ExecuteQuery y solicitudes por segundo que puedes realizar Región de	30	Sí

Recurso	Description (Descripción)	Cuota	Ajustable
	AWS en cada una de ellas Cuenta de AWS.		
Número máximo de consultas en ejecución simultánea	El número máximo de consultas que se pueden ejecutar simultáneamente.	30	Sí

Historial de documentos de la Guía AWS IoT SiteWise del usuario

En la siguiente tabla se describe la documentación de esta versión de AWS IoT SiteWise.

- Versión de API: 2019-12-02

Cambio	Descripción	Fecha
<u>El paquete de procesamiento de datos (DPP) dejará de estar abierto a nuevos clientes a partir del 7 de noviembre de 2025.</u>	DPP dejará de estar abierto a nuevos clientes a partir del 7 de noviembre de 2025. Si desea utilizar la función de DPP, inscríbese antes de esa fecha. Los clientes existentes pueden seguir utilizando el servicio con normalidad.	7 de octubre de 2025
<u>AWS IoT SiteWise Monitor dejará de estar abierto a nuevos clientes a partir del 7 de noviembre de 2025.</u>	AWS IoT SiteWise Monitor dejará de estar abierto a nuevos clientes a partir del 7 de noviembre de 2025. Si desea utilizar la función SiteWise Monitor, inscríbase antes de esa fecha. Los clientes existentes pueden seguir utilizando el servicio con normalidad.	7 de octubre de 2025
<u>Interfaces para modelos de activos</u>	Defina una estructura común que garantice la coherencia entre los modelos de activos y, al mismo tiempo, permita variaciones en la implementación.	5 de agosto de 2025

[Se agregó soporte nativo para anomalías a AWS IoT SiteWise](#)

AWS IoT SiteWise La detección nativa de anomalías es un servicio de aprendizaje automático (ML) para monitorear equipos industriales que detecta el comportamiento anormal de los equipos e identifica posibles fallas. Con la detección de anomalías nativa, puede implementar programas de mantenimiento predictivo e identificar los procesos de los equipos que no son óptimos.

28 de julio de 2025

[Se agregaron funciones Rich SQL y nuevos tipos de datos, cláusulas y operadores de SQL al SQL existente](#)

Se agregó soporte para nuevos tipos de datos, cláusulas y operadores.

22 de julio de 2025

[Procese y visualice datos con SiteWise Edge y herramientas de código abierto](#)

Integre herramientas de código abierto como Node-RED®, InfluxDB® y Grafana® en pasarelas V3 habilitadas para MQTT. AWS IoT SiteWise Utilice el procesamiento y la visualización locales para mejorar sus capacidades de gestión de datos industriales.

3 de julio de 2025

[Conecte aplicaciones externas al broker EMQX con nombres de usuario y contraseñas](#)

Se agregó soporte para conectar aplicaciones externas mediante nombres de usuario y contraseñas al agente AWS IoT Greengrass EMQX que se implementa de forma predeterminada al crear una nueva puerta de enlace V3 compatible con MQTT.

1 de mayo de 2025

[Support para pasarelas V3 habilitadas para MQTT en Edge SiteWise](#)

26 de febrero de 2025

Se agregaron nuevas funciones y se eliminó el contenido obsoleto

- Se agregó soporte para puertas de enlace V3 habilitadas para MQTT. Configuración de destino mejorada mediante filtros de ruta para suscribirse a temas de MQTT, incluida la ingesta de datos en tiempo real directamente AWS IoT SiteWise o la ingesta de datos en búfer mediante Amazon S3.
- Publicada la versión 3.0.0 del recopilador IoT SiteWise OPC UA y la versión 4.0.0 del componente IoT SiteWise Publisher para AWS IoT Greengrass V2.
- Se cambió el nombre de la versión anterior de las pasarelas SiteWise Edge autohospedadas por Classic Streams, compuertas V2.
- Se eliminaron las referencias a ella AWS IoT Greengrass V1 en la documentación de SiteWise Edge porque ya no se admite su uso con AWS IoT SiteWise

<u>Compatibilidad con AWS IoT SiteWise Assistant</u>	Se agregó soporte para el AWS IoT SiteWise Asistente , un asistente generativo impulsado por IA.	18 de noviembre de 2024
<u>Se agregaron tiempos de espera de sesión configurables para Edge SiteWise APIs</u>	Se agregaron ajustes de tiempo de espera de sesión configurables para administrar los períodos de inactividad de Edge. AWS OpsHub SiteWise APIs	31 de octubre de 2024
<u>Se agregaron ajustes de proxy configurables para Edge SiteWise APIs</u>	Se agregó la administración de la información del almacén de confianza para permitir la compatibilidad con el proxy HTTPS en las puertas de enlace SiteWise Edge.	31 de octubre de 2024
<u>Habilite CORS para Edge SiteWise APIs</u>	Se agregó la compatibilidad con CORS para SiteWise Edge APIs a fin de permitir el acceso seguro a las aplicaciones web entre dominios.	30 de septiembre de 2024
<u>Compatibilidad con orígenes de datos CloudRail y Litmus Edge de socios</u>	Se ha agregado compatibilidad tanto para los orígenes de datos CloudRail y Litmus Edge de socios.	5 de septiembre de 2024
<u>Disponibilidad general para ejecutar SiteWise Edge en Siemens Industrial Edge</u>	AWS IoT SiteWise ahora es compatible con la disponibilidad general de SiteWise Edge en dispositivos Siemens Industrial Edge.	24 de julio de 2024

[Se ha añadido compatibilidad con la configuración de marcas de tiempo en orígenes de datos OPC UA](#)

AWS IoT SiteWise ahora admite la configuración de marcas de tiempo para las fuentes de datos OPC UA.

24 de julio de 2024

[Se ha añadido compatibilidad con la conversión de tipos de datos en orígenes de datos OPC UA](#)

AWS IoT SiteWise ahora admite la conversión de tipos de datos para tipos de datos OPC UA no compatibles.

24 de julio de 2024

[Se agregó soporte para ejecutar una vista previa de SiteWise Edge en Siemens Industrial Edge](#)

AWS IoT SiteWise ahora permite ejecutar una vista previa de SiteWise Edge en los dispositivos Siemens Industrial Edge.

26 de noviembre de 2023

[Se ha añadido compatibilidad con almacenamiento en niveles de acceso semiaactivo](#)

AWS IoT SiteWise ahora es compatible con el almacenamiento en caliente, un nivel de almacenamiento totalmente gestionado que facilita a los clientes el almacenamiento seguro de los datos industriales y el acceso a ellos.

15 de noviembre de 2023

[Se ha añadido compatibilidad con identificadores únicos definidos por el usuario](#)

AWS IoT SiteWise ahora admite el uso de identificadores únicos definidos por el usuario para los activos, los modelos de activos, las propiedades y las jerarquías.

15 de noviembre de 2023

[Se ha añadido compatibilidad con detección de anomalías multivariantes en activos industriales](#)

AWS IoT SiteWise ahora admite la detección de anomalías multivariante de activos industriales mediante la integración de datos históricos y en tiempo real de los equipos con Amazon Lookout for Equipment.

15 de noviembre de 2023

[Se agregó soporte para la ingesta rentable y escalable de datos de series temporales en AWS IoT SiteWise](#)

AWS IoT SiteWise ahora permite la ingesta rentable y escalable de los datos de series temporales necesarios para los casos de uso analíticos.

15 de noviembre de 2023

[Se ha añadido compatibilidad con importación, exportación y actualización masivas](#)

AWS IoT SiteWise ahora admite la importación, exportación y actualización masivas de metadatos de equipos industriales.

15 de noviembre de 2023

[Se ha añadido compatibilidad con componentes del modelo de activos](#)

AWS IoT SiteWise ahora es compatible con los componentes del modelo Asset para ayudar a los clientes industriales a crear componentes reutilizables.

15 de noviembre de 2023

[Se ha añadido compatibilidad con la aplicación de panel de IoT](#)

AWS IoT SiteWise ahora es compatible con una aplicación de panel de código abierto en la que puede visualizar los datos operativos e interactuar con ellos.

15 de noviembre de 2023

<u>Se actualizaron las funciones vinculadas al servicio para AWS IoT SiteWise</u>	AWS IoT SiteWise tiene nuevas funciones vinculadas al servicio y puede ejecutar una consulta de búsqueda de metadatos en la base de datos. AWS IoT TwinMaker	6 de noviembre de 2023
<u>Se ha actualizado el etiquetado de los recursos del flujo de datos AWS IoT SiteWise</u>	Se añadió soporte para etiquetar los recursos del flujo de datos.	18 de agosto de 2022
<u>Puertas de enlace SiteWise Edge actualizadas</u>	Ahora se puede configurar el publicador para que controle qué datos deben enviarse desde la periferia a la nube y el orden en que deben enviarse a la nube.	12 de enero de 2022
<u>Se actualizó la demostración AWS IoT SiteWise</u>	Ahora puede utilizar la demostración para crear un portal de SiteWise monitores.	10 de enero de 2022
<u>Se actualizó la administración del almacenamiento</u>	Ahora puede definir un período de retención para controlar cuánto tiempo deben conservarse sus datos en el nivel de acceso frecuente.	29 de noviembre de 2021
<u>Se añadió soporte para la administración del flujo de datos</u>	Ahora puede incorporar datos AWS IoT SiteWise antes de crear modelos y activos de activos.	24 de noviembre de 2021
<u>Se actualizaron las jerarquías de los modelos de activos</u>	Ahora se puede asociar un modelo de entidad secundaria a varios modelos de activos principales.	28 de octubre de 2021

Lanzamiento regional	Lanzado AWS IoT SiteWise en AWS GovCloud (EE. UU., oeste).	29 de septiembre de 2021
Funciones de actualización	<p>Se añadieron las siguientes características</p> <ul style="list-style-type: none">• En las métricas, puede usar expresiones anidadas en las funciones de agregación y las funciones temporales.• En las transformaciones, puede utilizar la función <code>pretrigger()</code> para recuperar el valor de una variable antes de actualizar la propiedad que activó el cálculo de la transformación actual.	10 de agosto de 2021
Intervalo de tiempo personalizado para las métricas	Se añadió soporte para los intervalos de tiempo personalizados y los desfases en las métricas.	3 de agosto de 2021
AWS IoT SiteWise Utilizándolo en el borde	La característica de procesamiento en la periferia está ahora disponible de forma general.	29 de julio de 2021
Exportación de datos a Amazon S3	AWS IoT SiteWise ahora puede exportar datos a Amazon S3.	27 de julio de 2021
Puntos de conexión de VPC (AWS PrivateLink)	El punto de conexión de VPC en la interfaz para las operaciones de API de plano de control ahora está disponible con carácter general.	15 de julio de 2021

Transformaciones	Las transformaciones ahora pueden introducir múltiples variables de propiedades de activos.	8 de julio de 2021
Se actualizó la función <code>marca temporal()</code>	En las transformaciones, ahora se puede proporcionar una variable como argumento a la función <code>timestamp()</code> .	16 de junio de 2021
Disponibilidad general de alarmas	La característica de alarmas ya está disponible con carácter general.	27 de mayo de 2021
Se lanzó la versión 2 del adaptador de protocolo Modbus-TCP	Está disponible la versión 2 del conector del adaptador de protocolo Modbus-TCP . Esta versión agregó soporte para cadenas fuente ISO8859 codificadas y ASCII. UTF8	24 de mayo de 2021
Service quotas actualizadas	Se agregaron las siguientes cuotas para la GetInterpolatedAssetPropertyValues API: tasa de <code>GetInterpolatedAssetPropertyValues</code> solicitudes, cantidad de resultados por <code>GetInterpolatedAssetPropertyValues</code> solicitud y cantidad de días entre la fecha de inicio anterior y la fecha actual de <code>GetInterpolatedAssetPropertyValues</code> .	29 de abril de 2021

[Se actualizaron las expresiones de fórmula](#)

Se añadieron los siguientes operadores y funciones:

22 de abril de 2021

- Se añadieron los siguientes [operadores](#): <, >, <=, >=, ==, !=, !, and, or y not.
- Se añadió la siguiente [función de comparación](#): neq(x, y).
- Se añadieron las siguientes [funciones de cadena](#): join(), format() y f' '.

[Puntos de conexión de VPC \(AWS PrivateLink\)](#)

Se agregó información sobre cómo establecer una conexión privada entre la nube privada virtual (VPC) y el plano de AWS IoT SiteWise control APIs mediante la creación de un punto final de VPC de interfaz.

16 de marzo de 2021

[Federación de IAM](#)

Los administradores y usuarios del portal SiteWise Monitor ahora pueden iniciar sesión en los portales asignados con sus credenciales de IAM.

16 de marzo de 2021

[Lanzamiento regional](#)

Lanzado AWS IoT SiteWise en China (Pekín).

3 de febrero de 2021

<u>Lanzamiento de la versión 10 SiteWise del conector IoT</u>	Está disponible la versión 10 SiteWise del conector IoT. Esta versión configura StreamManager para mejorar el control cuando se pierda y se restablezca la conexión de origen. Esta versión también acepta valores OPC UA con una ServerTimestamp cuando no hay una SourceTimestamp disponible.	22 de enero de 2021
<u>Funciones de fecha y hora</u>	AWS IoT SiteWise ahora admite las funciones de fecha y hora.	21 de enero de 2021
<u>Sintaxis de las funciones</u>	Ahora puede utilizar la sintaxis uniforme de llamadas a funciones (UFCS) para AWS IoT SiteWise las funciones.	11 de enero de 2021
<u>Integración con Grafana</u>	Se agregó información sobre cómo visualizar los AWS IoT SiteWise datos en los paneles de Grafana.	15 de diciembre de 2020

[AWS IoT SiteWise lanzamiento de funciones](#)

15 de diciembre de 2020

Ahora puede monitorear sus datos con alarmas, procesar datos industriales en la periferia, usar Modbus TCP y Ethernet/IP las fuentes de su puerta de enlace SiteWise Edge, filtrar los datos entrantes con bandas muertas y mucho más.

- Se añadió la sección [Monitoreo de datos con alarmas](#) que se puede utilizar para definir, configurar y responder a las alarmas en AWS IoT SiteWise.
- Se añadió la sección [Procesamiento en la periferia](#) que se puede usar para configurar el procesamiento de sus datos industriales en sus dispositivos de periferia.
- Se agregaron las secciones [Modbus TCP y EtherNet/IP](#) a la documentación fuente de la puerta de enlace Edge. SiteWise
- Se añadió la sección [destino del origen](#) que se puede usar para personalizar el lugar al que deben enviarse los datos industriales entrantes.

- Se agregó la sección de [filtrado OPC UA](#) que puede usar para controlar la frecuencia y el tipo de datos que se envían a su puerta de enlace SiteWise Edge desde su servidor local industrial.

[AWS IoT SiteWise ahora es compatible con la gestión CMKs del cliente.](#)

AWS IoT SiteWise ahora admite el cifrado gestionado por el cliente CMKs.

24 de noviembre de 2020

[Lanzamiento de la versión 8 SiteWise del conector IoT](#)

Está disponible la versión 8 SiteWise del conector IoT. Esta versión mejora la estabilidad cuando el conector experimenta una conectividad de red intermitente.

19 de noviembre de 2020

[Uso de cadenas y condicionales en expresiones de fórmula](#)

Se agregó información sobre cómo encadenar y funciones condicionales en expresiones de fórmula para transformaciones y métricas.

16 de noviembre de 2020

[Ingerir datos mediante el administrador de AWS IoT Greengrass flujos](#)

Se agregó información sobre cómo ingerir grandes volúmenes de datos de IoT de fuentes de datos locales mediante un dispositivo AWS IoT Greengrass perimetral.

16 de septiembre de 2020

[Puntos de conexión de VPC \(AWS PrivateLink\)](#)

Se agregó información sobre cómo establecer una conexión privada entre su nube privada virtual (VPC) y los AWS IoT SiteWise datos APIs mediante la creación de un punto final de VPC de interfaz.

4 de septiembre de 2020

[Lanzamiento de la versión 7 SiteWise del conector IoT](#)

Está disponible la versión 7 SiteWise del conector IoT. Esta versión corrige un problema con las métricas de la puerta de enlace SiteWise Edge.

14 de agosto de 2020

[Creación de usuarios del IAM Identity Center desde la consola AWS IoT SiteWise](#)

Se agregó información sobre cómo crear usuarios del IAM Identity Center en la AWS IoT SiteWise consola. Ahora puede crear usuarios de IAM Identity Center cuando asigne usuarios a un portal nuevo o existente. Se actualizó el tutorial [Visualización y uso compartido de datos de parques eólicos](#) para utilizar esta característica. Este cambio reduce el número de pasos del tutorial.

4 de agosto de 2020

[Solución de problemas mejorada de SiteWise Edge Gateway](#)

Se agregó información adicional sobre cómo solucionar problemas en una puerta de enlace SiteWise Edge y cómo [exportar el certificado de cliente OPC UA](#) a una fuente.

18 de junio de 2020

[Documentación de tareas de la consola](#)

Se ha agregado documentación de tareas de la consola para [Modelado de activos industriales](#), [Consulta de datos de propiedades de activos](#) e [Interacción con otros servicios](#). Puede seguir estas instrucciones para completar las tareas en la consola de AWS IoT SiteWise.

11 de junio de 2020

[Tutorial de análisis de datos exportados](#)

Se ha añadido un tutorial que puede seguir para aprender a utilizar Amazon Athena para analizar los datos de activos que ha exportado a Amazon S3 con la plantilla de [funciones CloudFormation de exportación](#).

27 de mayo de 2020

[Mejorado mediante el uso de expresiones de fórmula](#)

Se agregó información detallada sobre el comportamiento de las propiedades de las AWS IoT SiteWise fórmulas y se agregó un ejemplo de cómo contar los puntos de datos filtrados.

18 de mayo de 2020

[Lanzamiento de la versión 6 SiteWise del conector IoT](#)

Está disponible la versión 6 SiteWise del conector IoT. Esta versión añade compatibilidad con las CloudWatch métricas y la detección automática de nuevas etiquetas OPC UA. Esto significa que no es necesario reiniciar la puerta de enlace SiteWise Edge cuando cambien las etiquetas de las fuentes OPC UA. Esta versión del conector requiere el administrador de transmisiones y el software AWS IoT Greengrass Core v1.10.0 o superior.

29 de abril de 2020

[AWS IoT SiteWise versión de funciones](#)

AWS IoT SiteWise lanzamiento de funciones. Ahora puede administrar las puertas de enlace de SiteWise Edge con la API, añadir su logotipo a los portales, ver las métricas de las puertas de enlace de SiteWise Edge y mucho más.

29 de abril de 2020

- Se agregó la sección [Exportación de datos a Amazon S3](#) con una CloudFormation plantilla que puede usar para exportar nuevos valores de datos a un bucket de Amazon S3.
- Se agregó la sección [de configuración de fuentes de datos](#), que mejora la documentación fuente de SiteWise Edge Gateway e incluye la nueva puerta de enlace SiteWise Edge APIs.
- Se agregó la sección de métricas de las puertas de enlace de SiteWise Edge, que describe las CloudWatch métricas que publican las puertas de enlace de SiteWise Edge.
- Se agregó la sección Configuración de una puerta de enlace SiteWise Edge en Amazon EC2 con una CloudFormation plantilla que puede usar para configurar

rápidamente las dependencias de la puerta de enlace SiteWise Edge en una instancia de Amazon EC2.

- Se agregó la sección de [funciones de servicio del portal](#) que describe la nueva función de permisos de SiteWise los portales Monitor.
- Se ha actualizado la [documentación del portal](#) para roles de servicio del portal y logotipos del portal.
- Se agregó la sección [Cómo etiquetar AWS IoT SiteWise los recursos](#).
- Se ha actualizado la sección [Creación de paneles \(CLI\)](#) para la nueva estructura de definición de panel.
- Se ha agregado la sección [Seguridad](#).

[Ingerir datos de AWS IoT Events](#)

Se agregó información sobre cómo ingerir datos del AWS IoT Events momento en que se produce un evento.

20 de abril de 2020

[Tutorial sobre visualización y uso compartido de datos de parques eólicos en Monitor SiteWise](#)

Se ha añadido un tutorial que puede seguir para aprender a utilizarlos AWS IoT SiteWise Monitor para visualizar y compartir datos de activos.

12 de marzo de 2020

AWS IoT SiteWise conceptos	Se ha añadido un glosario de AWS IoT SiteWise conceptos que puede utilizar para obtener información sobre el servicio y sus términos comunes.	5 de marzo de 2020
Se eliminaron AWS IoT Greengrass las instrucciones de instalación	Se eliminaron las instrucciones de instalación del software AWS IoT Greengrass básico de la Guía AWS IoT SiteWise del usuario. La Guía para AWS IoT Greengrass desarrolladores incluye un script de configuración del dispositivo e instrucciones para configurarlo AWS IoT Greengrass en otras plataformas, como Amazon EC2 y Docker.	14 de febrero de 2020
Se ha mejorado la ingesta de datos mediante reglas AWS IoT Core	Se ha añadido información detallada sobre cómo utilizar y solucionar los problemas de la acción de la AWS IoT SiteWise regla, que se puede utilizar para ingerir datos de los mensajes de MQTT mediante ella. AWS IoT Core	14 de febrero de 2020
Lanzamiento de la versión 5 SiteWise del conector IoT	Está disponible la versión 5 SiteWise del conector IoT. Esta versión corrige un problema de compatibilidad con la versión AWS IoT Greengrass 1.9.4 del software Core.	12 de febrero de 2020

[Lanzamiento de la versión 4 SiteWise del conector IoT](#)

Está disponible la versión 4 SiteWise del conector IoT. Esta versión corrige un problema con la reconexión del servidor OPC UA.

7 de febrero de 2020

[Activos industriales de modelado reestructurados](#)

Se ha reestructurado la sección Actualización de activos y modelos en varios temas dentro de Modelado de activos industriales.

4 de febrero de 2020

- [Estados de activos y modelos](#)
- [Gestione los flujos de datos para AWS IoT SiteWise](#)
- [Actualización de valores de atributos](#)
- [Asociación y desasociación de activos](#)
- [Actualización de activos y modelos](#)
- [Elimine activos y modelos en AWS IoT SiteWise](#)

[Tutorial sobre cómo ingerir datos de AWS IoT cosas](#)

Se agregó un tutorial que puede seguir para aprender a configurar una acción de AWS IoT SiteWise regla para ingerir datos de una flota de AWS IoT cosas nueva o existente.

4 de febrero de 2020

Reestructurada: recuperación de datos de AWS IoT SiteWise	Se reestructuró la sección de recuperación de datos en dos secciones de nivel superior: Consulta de valores y agregados de propiedades de activos e Interacción con otros servicios. AWS	21 de enero de 2020
Publicación de actualizaciones de valores de propiedades en el tutorial de Amazon DynamoDB	Se ha agregado un tutorial que puede seguir para aprender a utilizar las notificaciones de valor de propiedad para almacenar datos de activos en DynamoDB.	8 de enero de 2020
Uso de expresiones de fórmula	Se ha agregado la referencia de expresión de fórmula para organizar las constantes y las funciones disponibles para su uso en propiedades de transformación y métricas. Propiedades de activos se ha reestructurado en varios temas independientes en función de cada tipo de propiedad.	7 de enero de 2020
Uso de filtros de nodos OPC UA	Se agregó información sobre cómo usar los filtros de nodos OPC UA para mejorar el rendimiento de las puertas de enlace de Edge al agregar fuentes de puertas de enlace de SiteWise Edge. SiteWise	3 de enero de 2020

Actualización de un conector	Se agregó información sobre cómo actualizar una puerta de enlace SiteWise Edge cuando se publique una nueva versión del conector.	30 de diciembre de 2019
Lanzamiento de la versión 3 SiteWise del conector IoT	Está disponible la versión 3 SiteWise del conector IoT. Esta versión elimina el requisito de permisos de <code>iot:*</code> .	17 de diciembre de 2019
Lanzamiento SiteWise del conector IoT versión 2	Está disponible la versión 2 SiteWise del conector IoT. Esta versión agrega compatibilidad con múltiples recursos de secretos de OPC UA.	10 de diciembre de 2019
Creación de paneles de control (AWS CLI)	Se agregó información sobre cómo crear un panel de control al AWS IoT SiteWise Monitor usar el AWS CLI.	6 de diciembre de 2019

[AWS IoT SiteWise publicada la versión 2](#)

Vista previa publicada de la versión 2 de AWS IoT SiteWise. Ahora puede ingerir datos a través de OPC UA, MQTT y HTTP, modelar sus datos en jerarquías de activos y visualizarlos con Monitor. SiteWise

2 de diciembre de 2019

- Se ha reescrito la sección de [modelado de activos](#) en cambios en activos, modelos de activos y jerarquías de activos.
- Se actualizó la sección de [ingesta de datos](#) para incluir los pasos de AWS IoT Greengrass conexión y las secciones de ingesta de datos que no son de pasarela.
- Se agregó la [AWS IoT SiteWise Monitor](#) sección y una [guía de aplicación independiente](#) que muestra cómo usar la aplicación web SiteWise Monitor.
- Se han agregado las secciones [Consulta datos de AWS IoT SiteWise](#) y [Interactúa con otros AWS servicios](#).
- Se ha reescrito la sección de [introducción](#) para que coincida con la experiencia

de la demostración actualiza
da.

[AWS IoT SiteWise publicada
la versión 1](#)

Publicada la vista previa inicial 25 de febrero de 2019
de la versión 1 de AWS IoT
SiteWise.

Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la versión original de inglés, prevalecerá la versión en inglés.