



Guía del usuario de

AWS Batch



AWS Batch: 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 Batch?	1
¿Es la primera vez que usa AWS Batch?	3
Servicios relacionados	4
Acceso a AWS Batch	4
Componentes de AWS Batch	5
Entorno de computación	5
Colas de trabajo	6
Definiciones de trabajo	6
Tareas	6
Política de programación	6
Recursos consumibles	7
Entorno de servicio	7
Trabajo de servicio	7
Cómo configurar AWS Batch	8
Creación de una cuenta y un usuario administrador de IAM	8
Inscríbase en una Cuenta de AWS	8
Creación de un usuario con acceso administrativo	9
Cree roles de IAM	10
Crear un par de claves	11
Creación de una VPC	13
Creación de un grupo de seguridad	14
Instala la AWS CLI	16
Tutoriales de introducción	17
Introducción a Amazon EC2 con el asistente	17
Descripción general	17
Requisitos previos	18
Paso 1: Crear un entorno de computación	18
Paso 2: Crear una cola de trabajo	19
Paso 3: crear una definición de trabajo	20
Paso 4: Crear un trabajo	20
Paso 5: Revisar y crear	20
Paso 6: Ver la salida de la tarea	21
Paso 7: Eliminar los recursos del tutorial	21
Recursos adicionales	22

Introducción a la orquestación de Fargate con el asistente	22
Descripción general de	22
Requisitos previos	23
Paso 1: Crear un entorno de computación	23
Paso 2: Crear una cola de trabajo	24
Paso 3: Crea una definición de trabajo	24
Paso 4: Crear un trabajo	25
Paso 5: Revisar y crear	26
Paso 6: Ver la salida de la tarea	26
Paso 7: Eliminar los recursos del tutorial	26
Recursos adicionales	27
Cómo empezar a AWS Batch usar Fargate el AWS CLI	27
Requisitos previos	28
Creación de un rol de ejecución de IAM	28
Creación de un entorno de computación	30
Cree una cola de trabajos	31
crear una definición de trabajo, ...	31
Envío y supervisión de un trabajo	32
Visualización de la salida de un trabajo	34
Eliminar recursos	34
Paso a la producción	35
Sigüientes pasos	36
Introducción a Amazon EKS	37
Descripción general de	38
Requisitos previos	39
Paso 1: Cree su clúster de Amazon EKS para AWS Batch	40
Paso 2: Prepare el clúster de Amazon EKS para AWS Batch	41
Paso 3: Crear un entorno de computación de Amazon EKS	44
Paso 4: Cree una cola de trabajos y adjunte el entorno informático	46
Paso 5: Crear una definición de trabajo	47
Paso 6: Presentar un trabajo	48
Paso 7: Ver la salida del trabajo	49
Paso 8: (opcional) Enviar un trabajo con anulaciones	49
Paso 9: Eliminar los recursos del tutorial	50
Recursos adicionales	51
Cómo empezar a utilizar AWS Batch Amazon EKS Private Clusters	51

Descripción general de	38
Requisitos previos	54
Paso 1: Cree su clúster de EKS para AWS Batch	55
Paso 2: Prepare su clúster EKS para AWS Batch	57
Paso 3: Crear un entorno de computación de Amazon EKS	60
Paso 4: Cree una cola de trabajos y adjunte el entorno informático	62
Paso 5: Crear un Amazon ECR con pull through cache	63
Paso 6: Registrar una definición de trabajo	64
Paso 7: Enviar un trabajo para ejecutarlo	65
Paso 8: Ver la salida del trabajo	65
Paso 9: (opcional) Enviar un trabajo con anulaciones	65
Paso 10: Eliminar los recursos del tutorial	66
Recursos adicionales	51
Resolución de problemas	67
¿Cómo empezar con AWS Batch la SageMaker IA	68
Descripción general de	68
Requisitos previos	70
Paso 1: Crear un rol de ejecución de SageMaker IA	70
Paso 2: Crear el entorno de servicio	72
Paso 3: Crea tu cola de SageMaker trabajos	73
Paso 4: Crear y enviar un trabajo de entrenamiento	74
Paso 5: Supervisar el estado del trabajo	76
Paso 6: Ver la salida del trabajo	78
Paso 7: Eliminar los recursos del tutorial	81
Recursos adicionales	81
El panel del widget de AWS Batch	83
Agregar el widget de cola de trabajos única	83
Cómo agregar el widget de CloudWatch Container Insights	84
Agregar el widget de registros de trabajos	84
Entornos informáticos para AWS Batch	86
Entornos de computación administrados	86
Consideraciones a la hora de crear trabajos paralelos de varios nodos	89
Entornos de computación no administrados	90
Creación de un entorno de computación	91
Tutorial: creación de un entorno de computación gestionado con los recursos de Fargate	92

Tutorial: creación de un entorno de computación gestionado con los recursos de Amazon EC2	94
Tutorial: creación de un entorno de computación no gestionado con los recursos de Amazon EC2	104
Tutorial: creación de un entorno de computación gestionado con los recursos de Amazon EKS	105
Tutorial: Creación de un entorno informático no gestionado con los recursos de Amazon EKS	111
Recurso: plantilla de entorno de computación	112
Tabla de computación para el tipo de instancia	114
Actualización de un entorno de computación	117
Estrategias de actualización del entorno de computación	118
Cómo elegir la mejor estrategia de actualización	119
Consideraciones sobre la actualización de AMI	120
Cómo realizar actualizaciones de escalado	121
Cómo realizar actualizaciones de la infraestructura	124
Realice Blue/green actualizaciones	129
AMI de recursos de computación	134
Especificaciones de AMI de recursos de computación	136
Tutorial: creación de una AMI de recursos de computación	138
Uso de una AMI de carga de trabajo de GPU	141
Obsolescencia de Amazon Linux	147
Obsolescencia de la AMI de Amazon Linux 2 en Amazon EKS	148
Obsolescencia de la AMI de Amazon Linux 2 en Amazon ECS	148
Uso de plantillas de lanzamiento de Amazon EC2	149
Plantillas de lanzamiento predeterminadas y anuladas	151
Datos de usuario de Amazon EC2 en las plantillas de lanzamiento	152
Referencia: ejemplos de plantillas de lanzamiento	153
Configuración del servicio de metadatos de instancias (IMDS)	156
Casos de configuración	156
Configuraciones de EC2	158
Cómo migrar de ECS AL2 a ECS AL2023	159
Estrategias de asignación de tipos de instancia	161
Administración de la memoria	163
Reserva de memoria del sistema	164
Tutorial: visualización de la memoria de recurso de computación	164

Consideraciones sobre memoria y vCPU para AWS Batch en Amazon EKS	165
Entornos de computación en Fargate	170
Cuándo usar Fargate	171
Definiciones de trabajo en Fargate	172
Colas de trabajo en Fargate	174
Entornos de computación en Fargate	174
Entornos de computación de Amazon EKS	175
Amazon EKS	178
AMI predeterminada de Amazon EKS	179
Entornos de AMI mixtos	180
Versiones de Kubernetes compatibles	181
Actualización de la versión Kubernetes del entorno de computación.	182
Responsabilidad compartida de los nodos Kubernetes	182
Ejecute un DaemonSet nodo AWS Batch gestionado	183
Personalización de plantillas de lanzamiento de Amazon EKS	184
¿Cómo actualizar de EKS AL2 a EKS AL2023	189
Entornos de servicio	191
Qué son los entornos de servicio	191
Cómo funcionan los entornos de servicio con otros componentes AWS Batch	192
Prácticas recomendadas para los entornos de servicios	192
Estados y ciclo de vida del entorno de servicio	194
Definiciones de estado del entorno de servicios	194
Creación de un entorno de servicio	196
Requisitos previos	196
Actualización de un entorno de servicio	198
Eliminación de un entorno de servicio	200
Requisitos previos de eliminación	200
Colas de trabajos	203
Creación de una cola de trabajos	203
Cómo crear una cola de Amazon EC2	204
Cómo crear una cola de trabajos de Fargate	205
Cómo crear una cola de trabajos de Amazon EKS	206
Creación de una cola de trabajos de SageMaker	207
Plantilla de cola de trabajos	210
Ver una cola de trabajos	211
Visualización de información de la cola de trabajos	211

Eliminación de una cola de trabajos	213
Políticas de programación de reparto justo	214
Uso de identificadores de reparto	215
Uso de políticas de programación	216
Uso de una programación de reparto justo	216
Tutorial: Cree una política de programación de reparto justo	218
Referencia: plantilla de política de programación de reparto equitativo	219
Programación basada en los recursos	220
Creación de recursos consumibles	221
Cómo especificar los recursos para un trabajo	222
Cómo comprobar el uso de recursos	223
Cómo actualizar las cantidades de recursos en uso	225
Cómo buscar trabajos que requieran un recurso consumible	226
Cómo eliminar un recurso consumible	227
Gestión de cuotas	228
Cuotas	229
Preferencia	230
Cree recursos de administración de cuotas	231
Crear cuotas compartidas	235
Someter los trabajos a una cuota	237
Definiciones de trabajo	239
Creación de una definición de trabajo de un solo nodo	239
Tutorial: creación de una definición de trabajo de un solo nodo en los recursos de Amazon EC2	240
Creación de una definición de trabajo de un solo nodo en los recursos de Fargate	246
Creación de una definición de trabajo de un solo nodo en los recursos de Amazon EKS	253
Creación de una definición de trabajo de un solo nodo con varios contenedores en los recursos de Amazon EC2	258
Creación de una definición de trabajo paralelo de varios nodos	265
Tutorial: creación de una definición de trabajo paralelo de varios nodos en los recursos de Amazon EC2	266
Referencia: plantilla de definición de trabajo que usa ContainerProperties	274
Parámetros de definición de trabajo para ContainerProperties	282
Cree definiciones de trabajo mediante EcsProperties	328
ContainerProperties frente a las definiciones de trabajo de EcsProperties	328
Cambios generales en el AWS Batch APIs	329

Definiciones de trabajo en varios contenedores para Amazon ECS	330
Definiciones de trabajos con varios contenedores para Amazon EKS	331
Referencia: escenarios de trabajo de AWS Batch que utilizan EcsProperties	332
Uso del controlador de registros awslogs	338
opciones del controlador de registro de awslogs en el tipo de datos AWS Batch	
JobDefiniton	339
Especificación de una configuración de registro en la definición de trabajo	342
Especificación de información confidencial	343
Uso de Secrets Manager	344
Uso del almacén de parámetros de Systems Manager	352
Autenticación de registros privados para trabajos	356
Permisos de IAM requeridos para la autenticación de registros privados	358
Tutorial: creación de un secreto para la autenticación de registros privados	359
Volúmenes de Amazon EFS	360
Consideraciones acerca de volúmenes de Amazon EFS	360
Uso de puntos de acceso de Amazon EFS	361
Indique un sistema de archivos de Amazon EFS en la definición del trabajo	362
Ejemplos de definición de trabajo	365
Variables de entorno	366
Sustitución de parámetros	367
Funcionalidad de GPU de prueba	368
Trabajo paralelo de varios nodos	369
Tareas	370
Tutorial: envío de un trabajo	371
Trabajos de servicio	374
Cargas útiles de trabajos de servicio	375
Envío de un trabajo de servicio	376
Estado del trabajo de servicio	378
Estrategias de reintento de trabajos de servicio	379
Supervisión de los trabajos de servicio en la cola	382
Finalización de los trabajos de servicio	384
Estados de trabajo	384
Variables de entorno de los trabajos	388
Reintentos automáticos de trabajo	389
Dependencias de trabajos	391
Tiempos de espera de trabajo	391

Trabajos de Amazon EKS	393
Tutorial: asignación de un trabajo en ejecución a un pod y un nodo	394
Tutorial: reasignación de un pod en ejecución a su función	395
Trabajos paralelos de varios nodos	396
Variables de entorno	398
Grupos de nodos	398
Ciclo de vida del trabajo	399
Consideraciones del entorno de computación	400
Trabajos paralelos de varios nodos en Amazon EKS	401
Ejecución de trabajos de MNP	402
Creación de una definición de trabajo de MNP de Amazon EKS	403
Envío de un trabajo de Amazon EKS MNP	406
Anulación de una definición de trabajo de MNP de Amazon EKS	406
Trabajos de matrices	407
Ejemplo de un flujo de trabajos de matriz	411
Uso del índice de trabajo de matriz	414
Ejecución de trabajos de GPU	420
Creación de un clúster de Kubernetes basado en GPU en Amazon EKS	425
Creación de una definición de trabajo de GPU de Amazon EKS	427
Ejecución de un trabajo de GPU en el clúster de Amazon EKS	428
Ver trabajos en una cola de trabajos	428
Buscar para trabajos en una cola de trabajos	429
Buscar AWS Batch trabajos (AWS consola)	429
Buscar y filtrar AWS Batch trabajos (AWS CLI)	431
Modos de red para AWS Batch trabajos	432
Vea los registros de trabajos en CloudWatch los registros	434
Revise la información del AWS Batch trabajo	435
Seguridad en AWS Batch	437
Gestión de identidad y acceso	438
Público	438
Autenticación con identidades	438
Administración del acceso con políticas	440
¿Cómo AWS Batch funciona con IAM	442
Ejemplos de políticas basadas en identidades	446
AWS políticas gestionadas	449
Políticas, roles y permisos de IAM	454

Estructura de la política de IAM	455
Recurso: ejemplos de políticas	459
Recurso: política AWS Batch gestionada	470
AWS Batch Función de ejecución de IAM	470
Permisos de nivel de recursos admitidos	471
Tutorial: creación del rol de ejecución de IAM	472
Tutorial: comprobación del rol de ejecución de IAM	472
Cómo utilizar roles vinculados a servicios	473
Rol de instancia de Amazon ECS	488
Rol de flota de spot de Amazon EC2	491
EventBridge Rol de IAM	494
Creación de una nube virtual privada	496
Creación de una VPC	496
Siguiendo pasos	497
Puntos de conexión de VPC	498
Consideraciones	498
Creación de un punto de conexión de interfaz	499
Creación de una política de punto de conexión	501
Validación de conformidad	502
Seguridad de la infraestructura	502
Prevención de la sustitución confusa entre servicios	503
Ejemplo: rol para acceder a un solo entorno de computación	504
Ejemplo: rol para acceder a varios entornos de computación	504
CloudTrail	505
AWS Batch información en CloudTrail	506
Referencia: Descripción de las entradas de los archivos de AWS Batch registro	507
Solucionar problemas de IAM AWS Batch	509
No estoy autorizado a realizar ninguna acción en AWS Batch	509
No estoy autorizado a realizar iam: PassRole	509
Quiero permitir que personas ajenas a mi AWS cuenta accedan a mis AWS Batch recursos	510
AWS Step Functions	511
Tutorial: visualización de detalles de las máquinas de estado	511
Tutorial: edición de una máquina de estado	512
Tutorial: ejecución de una máquina de estado	512
Amazon EventBridge	514

AWS Batch eventos	515
Eventos de cambio de estado de los trabajos	515
Eventos bloqueados en la cola de trabajos	517
Eventos de cambio de estado de los trabajos	519
Eventos bloqueados en el trabajo de servicio	520
Eventos de prevención de tareas de servicio	521
Tutorial: Usa las notificaciones AWS de usuario con AWS Batch	523
Trabajos de AWS Batch como destinos de EventBridge	524
Tutorial: creación de un trabajo programado	525
Tutorial: creación de una regla con un patrón de evento	527
Tutorial: paso del transformador de entrada	530
Tutorial: escucha de los eventos de trabajo de AWS Batch	533
Requisitos previos	533
Tutorial: creación de la función de Lambda	534
Tutorial: registro de la regla de eventos	535
Tutorial: prueba de la configuración	537
Tutorial: envío de alertas de Amazon Simple Notification Service para eventos de trabajos fallidos	537
Requisitos previos	537
Tutorial: creación de un tema de Amazon SNS y suscribirse a él	538
Tutorial: registro de una regla de eventos	538
Tutorial: prueba de la regla	540
Regla alternativa: cola de trabajos por lotes bloqueada	540
Elastic Fabric Adapter	542
Supervisar AWS Batch	545
CloudWatch Registros	545
Tutorial: Añadir una política de CloudWatch IAM de registros	546
Instale y configure el CloudWatch agente	548
Tutorial: Ver CloudWatch registros	549
CloudWatch Container Insights	551
Active Container Insights	551
Utilice CloudWatch los registros para supervisar AWS Batch los trabajos de Amazon EKS	552
Requisitos previos	553
Instalación del complemento	553
Etiquetar los recursos	555
Conceptos básicos de etiquetas	555

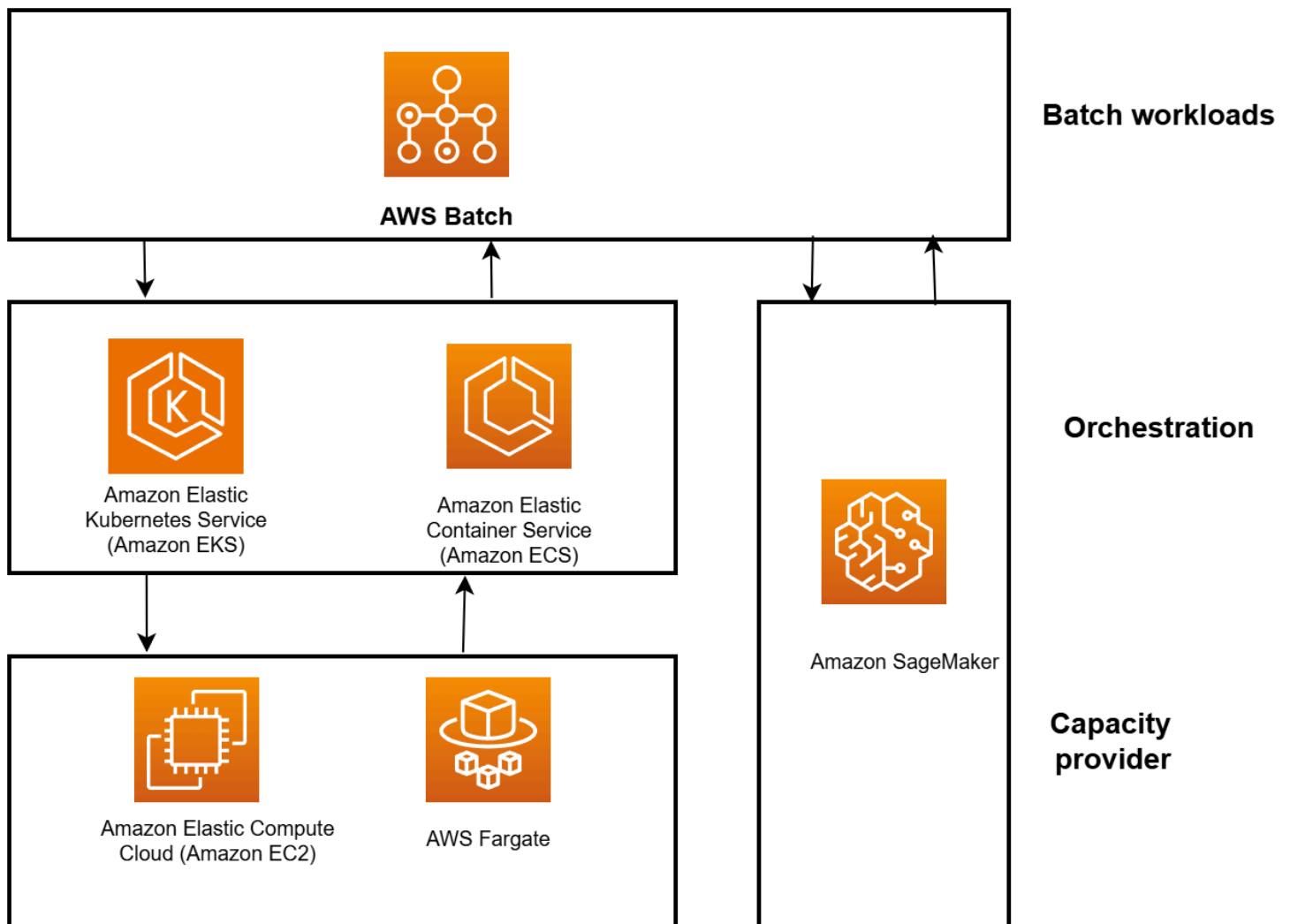
Etiquetar los recursos	556
Restricciones de las etiquetas	557
Tutorial: administración de etiquetas con la consola	558
Adición de etiquetas a un recurso individual durante su creación	558
Agregar y eliminar etiquetas en un recurso individual	558
Administración de etiquetas mediante el uso de CLI o API	559
Prácticas recomendadas	561
Cuándo se debe usar AWS Batch	561
Lista de comprobación para ejecutarla a escala	562
Optimice los contenedores y las AMI	563
Elija el recurso de entorno de computación adecuado	564
Amazon EC2 bajo demanda o Amazon EC2 Spot	565
Prácticas recomendadas para instancias de spot de Amazon EC2 para AWS Batch	566
Errores comunes y solución de problemas	568
Resolución de problemas	571
AWS Batch	572
Configuración óptima del tipo de instancia para recibir actualizaciones automáticas de la familia de instancias	573
Entorno de computación de INVALID	573
Trabajos bloqueados en estado RUNNABLE	576
Instancias de spot no etiquetadas en el momento de su creación	582
Las instancias de spot no se están reduciendo verticalmente	582
No puedo recuperar los secretos de Secrets Manager	584
No se pueden anular los requisitos de recursos de la definición del trabajo	584
Mensaje de error al actualizar la configuración de desiredvCpus	586
AWS Batch en Amazon EKS	586
Entorno de computación de INVALID	586
AWS Batch en Amazon EKS, el trabajo está RUNNABLE estancado	590
AWS Batch en Amazon EKS, el trabajo está STARTING estancado	591
Compruebe que aws-auth ConfigMap se ha configurado correctamente	592
Los permisos o enlaces de RBAC no están configurados correctamente	593
Recurso: cuotas de servicio	595
Historial de documentos	597
.....	dcv

¿Qué es () AWS Batch?

AWS Batch le ayuda a ejecutar cargas de trabajo de computación por lotes en Nube de AWS.

La informática por lotes es una forma común de acceso a grandes cantidades de recursos de computación utilizada por desarrolladores, científicos e ingenieros. AWS Batch elimina la ardua tarea de configurar y administrar la infraestructura necesaria, de forma similar al software de informática por lotes tradicional. Este servicio puede aprovisionar recursos de forma eficaz en respuesta a los trabajos enviados para eliminar limitaciones de capacidad, reducir costos de computación y ofrecer resultados con rapidez.

Al ser un servicio completamente administrado, AWS Batch le ayuda a ejecutar cargas de trabajo de computación por lotes de cualquier escala. AWS Batch proporciona automáticamente recursos de computación y optimiza la distribución de la carga de trabajo en función de la cantidad y la escala. Con AWS Batch, no es necesario instalar ni administrar software de informática por lotes, lo que le permite centrarse en analizar resultados y resolver problemas.

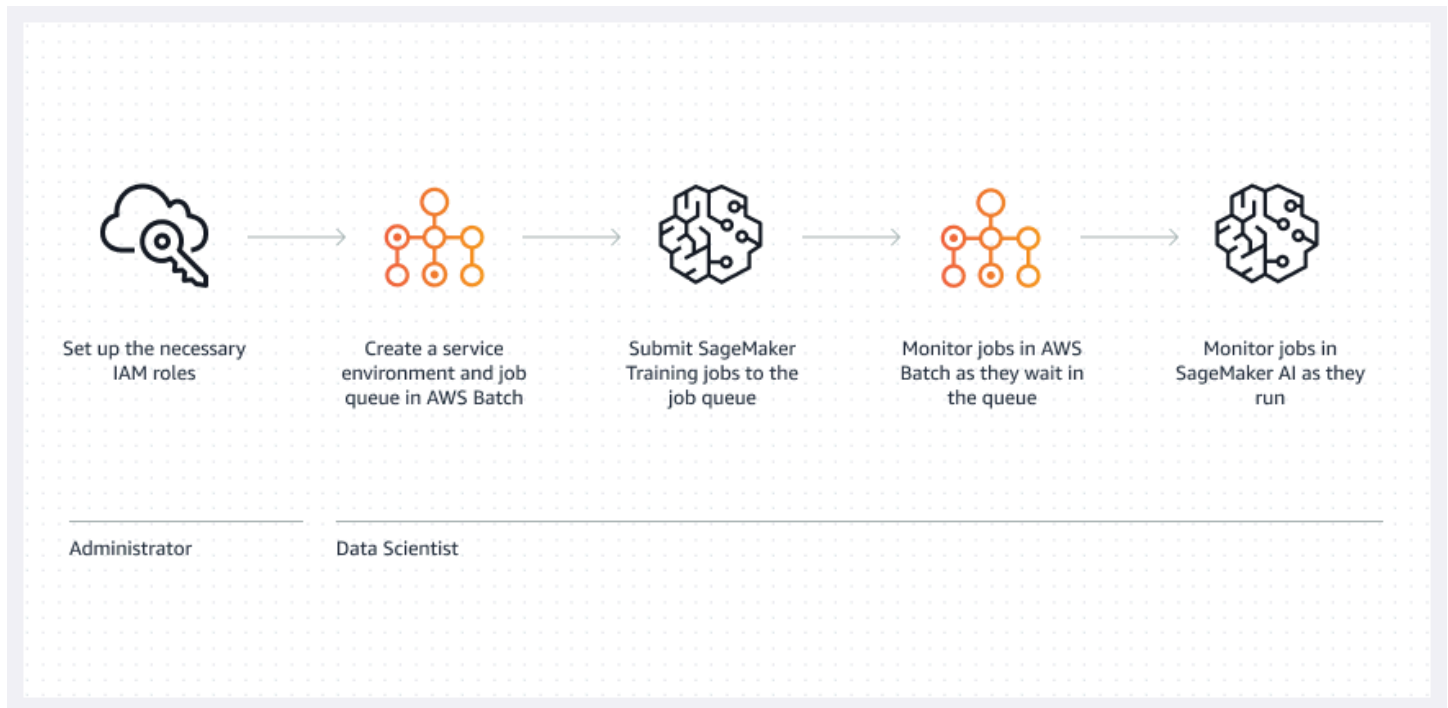


AWS Batch brinda todas las funciones necesarias para ejecutar cargas de trabajo a gran escala y con un uso intensivo de computación, además de los servicios administrados de AWS para la orquestación de contenedores, Amazon ECS y Amazon EKS. AWS Batch es capaz de escalar la capacidad de computación de instancias de Amazon EC2 y recursos de Fargate.

AWS Batch brinda un servicio totalmente administrado para cargas de trabajo por lotes y ofrece las capacidades operativas necesarias para optimizar este tipo de cargas de trabajo en términos de rendimiento, velocidad, eficiencia de los recursos y costos.

AWS Batch también permite crear colas en los trabajos de entrenamiento de SageMaker, lo que permite a los científicos de datos y a los ingenieros de machine learning enviar los trabajos de entrenamiento con prioridades a colas configurables. Esta capacidad garantiza que las cargas de trabajo de machine learning se ejecuten de forma automática en cuanto los recursos estén disponibles, lo que elimina la necesidad de realizar una coordinación manual y mejora la utilización de los recursos.

En el caso de las cargas de trabajo de machine learning, AWS Batch ofrece funciones de creación de colas para los trabajos de entrenamiento de SageMaker. Se pueden configurar las colas con políticas específicas para optimizar los costos, el rendimiento y la asignación de recursos para sus cargas de trabajo de formación en machine learning.



Esto presenta un modelo de responsabilidad compartida en el que los administradores configuran la infraestructura y los permisos, logrando al mismo tiempo que los científicos de datos puedan centrarse en enviar y supervisar sus cargas de trabajo de entrenamiento de machine learning. De forma automática, los trabajos se ponen en cola y se ejecutan según las prioridades configuradas y la disponibilidad de los recursos.

¿Es la primera vez que usa AWS Batch?

Si es la primera vez que usa AWS Batch, le recomendamos que empiece leyendo las siguientes secciones:

- [Componentes de AWS Batch](#)
- [Creación de una cuenta y un usuario administrador de IAM](#)
- [Cómo configurar AWS Batch](#)
- [Cómo empezar con AWS Batch los tutoriales](#)
- [Primeros pasos con AWS Batch la SageMaker IA](#)

Servicios relacionados

AWS Batch es un servicio de computación por lotes completamente administrado que planifica, programa y ejecuta las cargas de trabajo de machine learning, la simulación y el análisis por lotes en contenedores para la amplia variedad de ofertas de computación de AWS, como Amazon ECS, Amazon EKS, AWS Fargate e instancias de spot o bajo demanda. Para obtener más información sobre cada servicio de computación administrado, consulte:

- [Guía del usuario de Amazon EC2](#)
- [AWS Fargate Guía para desarrolladores de](#)
- [Guía del usuario de Amazon EKS](#)
- [Amazon SageMaker AI Guía para desarrolladores de](#)

Acceso a AWS Batch

Puede obtener acceso a AWS Batch utilizando lo siguiente:

AWS Batch Consola de

Interfaz web en la que se crean y administran los recursos.

AWS Command Line Interface

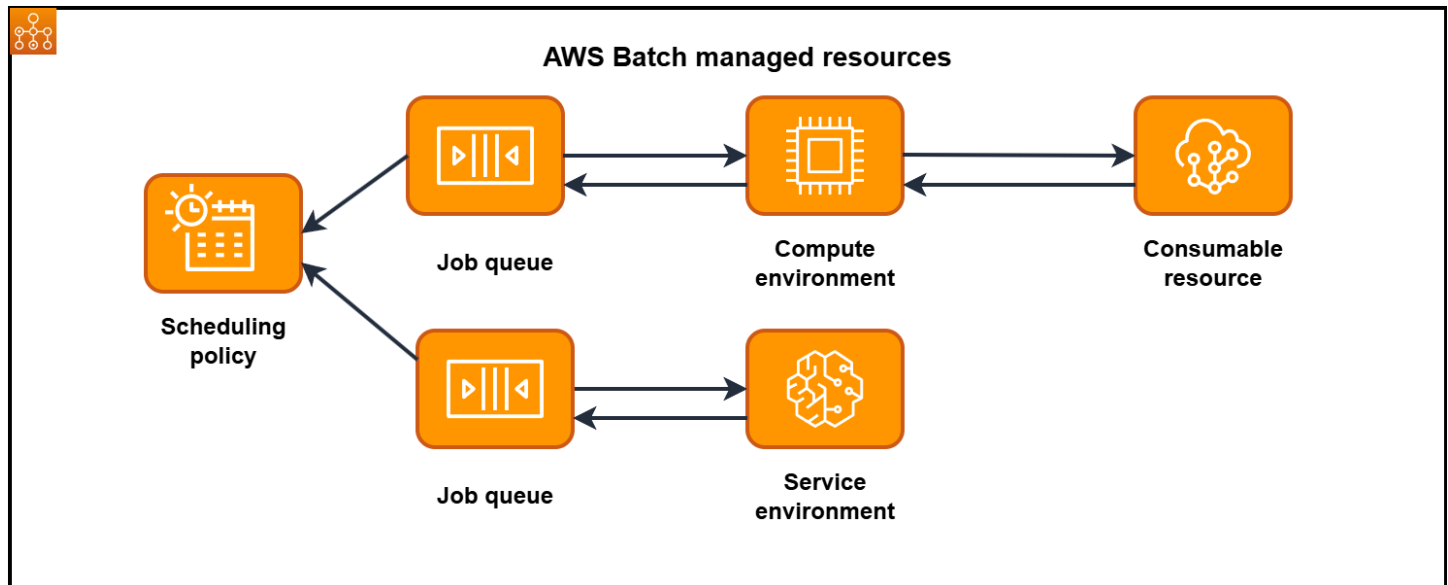
Interactúe con los Servicios de AWS mediante comandos en el shell de la línea de comandos. La AWS Command Line Interface es compatible con Windows, macOS y Linux. Para obtener más información sobre la AWS CLI, consulte la [Guía del usuario de AWS Command Line Interface](#). Puede encontrar los comandos de AWS Batch en la [Referencia de comandos de la AWS CLI](#).

AWS SDK

Si prefiere crear aplicaciones usando API específicas de un lenguaje en lugar de enviar una solicitud a través de HTTP o HTTPS, AWS proporciona bibliotecas, ejemplos de código, tutoriales y otros recursos. Estas bibliotecas proporcionan funciones básicas que automatizan tareas, como la firma criptográfica de las solicitudes, el reintento de las solicitudes o el tratamiento de las respuestas de error. Estas funciones permiten empezar de forma más eficiente. Para obtener más información, consulte [herramientas para crear en AWS](#).

Componentes de AWS Batch

AWS Batch simplifica los trabajos por lotes en ejecución en múltiples zonas de disponibilidad dentro de una región. Puede crear entornos de computación de AWS Batch dentro de una VPC nueva o existente. Después de que un entorno de computación se ha activado y asociado a una cola de trabajos, puede precisar las definiciones de trabajo que especifican cuáles imágenes de contenedor de Docker ejecutarán sus trabajos. Las imágenes de contenedor se almacenan y se extraen desde registros de contenedor, que podrían existir dentro o fuera de la infraestructura de AWS.



Entorno de computación

Un entorno de computación es un conjunto de recursos de computación administrados o no administrados que se utilizan para ejecutar trabajos. Con los entornos de procesamiento gestionados, usted puede especificar el tipo de procesamiento deseado (Fargate o EC2) con varios niveles de detalle. Puede configurar entornos de computación que utilicen un tipo de instancia EC2 determinado, como `c5.2xlarge` o `m5.10xlarge`. O bien, usted puede elegir especificar únicamente que desea utilizar los tipos de instancias más recientes. También puede especificar el número mínimo, el número deseado y el número máximo de CPU virtuales del entorno, junto con la cantidad que está dispuesto a pagar por una instancia de spot como un porcentaje del precio de las instancias bajo demanda y un conjunto de destino de subredes de la VPC. AWS Batch lanzará, administrará y terminará en forma eficiente los tipos de computación según sea necesario. También puede administrar sus propios entornos de computación. En este caso, usted es responsable de configurar y escalar las instancias en un clúster de Amazon ECS que AWS Batch crea para usted. Para obtener más información, consulte [Entornos informáticos para AWS Batch](#).

Colas de trabajo

Al enviar un trabajo de AWS Batch, lo envía a una cola de trabajos determinada, donde el trabajo permanecerá hasta que se programe en un entorno de computación. Asocie uno o más entornos de computación con una cola de trabajos. También puede asignar valores de prioridad a estos entornos de computación e incluso a las propias colas de trabajos. Por ejemplo, puede tener una cola de prioridad alta a la que envía trabajos prioritarios, y una cola de prioridad baja para los trabajos que pueden ejecutarse en cualquier momento, cuando los recursos de computación son más económicos. Para obtener más información, consulte [Colas de trabajos](#).

Definiciones de trabajo

Una definición de trabajo especifica cómo se ejecutan los trabajos. Puede considerar una definición de trabajo como un esquema de los recursos en su trabajo. Puede asignar a su trabajo un rol de IAM para proporcionar acceso a otros recursos de AWS. También puede especificar los requisitos de memoria y CPU. La definición de trabajo también puede controlar las propiedades de contenedor, las variables de entorno y los puntos de montaje para un almacenamiento persistente. Muchas de las especificaciones de una definición de trabajo pueden anularse mediante la especificación de nuevos valores al enviar trabajos individuales. Para obtener más información, consulte [Definiciones de trabajo](#)

Tareas

Una unidad de trabajo (como un script de shell, un ejecutable en Linux o una imagen de contenedor Docker) que envía a AWS Batch. Tiene un nombre y se ejecuta como una aplicación en contenedor en AWS Fargate o recursos de Amazon EC2 de su entorno de computación, utilizando los parámetros que se especifican en una definición de trabajo. Los trabajos pueden hacer referencia a otros trabajos por nombre o ID, y puede que de ellos dependa la correcta realización de otros trabajos o de la disponibilidad de los [recursos](#) que usted especifique. Para obtener más información, consulte [Tareas](#).

Política de programación

Puede utilizar las políticas de programación para configurar cómo se asignan los recursos de computación de una cola de trabajos entre los usuarios o las cargas de trabajo. Con las políticas de programación equitativas, puede asignar diferentes identificadores de participación a las cargas de trabajo o a los usuarios. El programador de trabajos de AWS Batch utilizará de forma predeterminada

la estrategia FIFO (primero en entrar, primero en salir). Para obtener más información, consulte [Políticas de programación de reparto justo](#).

Recursos consumibles

Un recurso consumible es un recurso que se necesita para ejecutar los trabajos, como un token de licencia de terceros, el ancho de banda de acceso a la base de datos, la necesidad de limitar las llamadas a una API de terceros, etc. Estos recursos consumibles se deben especificar para ejecutar un trabajo, por lo tanto, Batch contempla dichas dependencias de recursos cuando realiza la programación de uno. La reducción de la infrautilización de los recursos de computación se lleva a cabo con la asignación de los trabajos que contienen todos los recursos necesarios. Para obtener más información, consulte [Programación basada en los recursos](#).

Entorno de servicio

Un entorno de servicio define la forma en que AWS Batch se integra con SageMaker para la ejecución de tareas. Los entornos de servicio permiten que AWS Batch envíe trabajos y los administre en SageMaker, sin dejar de brindar las capacidades de creación de colas, programación y gestión de prioridades de AWS Batch. Estos definen los límites de capacidad para los tipos de servicios específicos, como los trabajos de entrenamiento de SageMaker. Los límites de capacidad controlan la cantidad máxima de recursos que pueden utilizar los trabajos de servicio en el entorno. Para obtener más información, consulte [Entornos de servicio para AWS Batch](#).

Trabajo de servicio

Un trabajo de servicio es una unidad de trabajo que se envía a AWS Batch para que se ejecute en un entorno de servicio. Los trabajos de servicio aprovechan las capacidades de preparación de colas y programación de AWS Batch, sin dejar de delegar la ejecución del servicio externo. Por ejemplo, los trabajos de entrenamiento de SageMaker enviados como trabajos de servicio se ponen en cola y AWS Batch los prioriza, pero la ejecución de los trabajos de entrenamiento de SageMaker se lleva a cabo dentro de la infraestructura de SageMaker AI. Esta integración permite a los científicos de datos y a los ingenieros de machine learning beneficiarse de la administración automatizada de las cargas de trabajos de AWS Batch, y las colas de prioridad, para sus cargas de trabajos de SageMaker AI. Los trabajos de servicio pueden hacer referencia a otros trabajos por su nombre o ID y admitir las dependencias de trabajos. Para obtener más información, consulte [Trabajos de servicio en AWS Batch](#).

Cómo configurar AWS Batch

Si ya se ha registrado en Amazon Web Services (AWS) y está utilizando Amazon Elastic Compute Cloud (Amazon EC2) o Amazon Elastic Container Service (Amazon ECS), pronto podrá usar AWS Batch. El proceso de configuración para estos servicios es similar. Esto se debe a que AWS Batch utiliza instancias de contenedor de Amazon ECS en sus entornos de computación. Para usar la AWS CLI con AWS Batch debe usar una versión de la AWS CLI que admita las últimas características de AWS Batch. Si alguna de las características de AWS Batch no son compatibles con la AWS CLI, debe actualizar a la última versión. Para obtener más información, consulte <http://aws.amazon.com/cli/>.

Note

Como AWS Batch utiliza componentes de Amazon EC2, se utiliza la consola de Amazon EC2 para muchos de estos pasos.

Lleve a cabo las siguientes tareas para obtener la configuración de AWS Batch.

Temas

- [Creación de una cuenta y un usuario administrador de IAM](#)
- [Creación de roles de IAM para los entornos de computación e instancias de contenedor](#)
- [Creación de un par de claves para las instancias](#)
- [Creación de una VPC](#)
- [Creación de un grupo de seguridad](#)
- [Instala la AWS CLI](#)

Creación de una cuenta y un usuario administrador de IAM

Para empezar, debe crear una AWS cuenta y un usuario único al que normalmente se le concedan derechos administrativos. Para ello, complete los siguientes tutoriales:

Inscríbase en una Cuenta de AWS

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

Para suscribirte 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 un Cuenta de AWS, Usuario raíz de la cuenta de AWS se crea un. 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 .

Creación de roles de IAM para los entornos de computación e instancias de contenedor

Las instancias de contenedor y los entornos de computación de AWS Batch requieren credenciales de Cuenta de AWS para realizar llamadas a otras API de AWS en su nombre. Cree un rol de AWS Identity and Access Management que proporcione estas credenciales a los entornos de computación e instancias de contenedor y, a continuación, asociar el rol a los entornos de computación.

Note

Para comprobar que Cuenta de AWS dispone de los permisos necesarios, consulte [Configuración inicial del servicio de IAM para su cuenta](#).

Los roles de instancia de contenedor y de entorno de computación de AWS Batch se crean automáticamente en la experiencia de primera ejecución de la consola. Por lo tanto, si tiene intención de utilizar la consola AWS Batch, puede pasar a la siguiente sección. Si, en cambio, tiene previsto utilizar la AWS CLI, siga el procedimiento descrito en [Utilizar funciones vinculadas al servicio para AWS Batch](#), [Rol de instancia de Amazon ECS](#) y [Tutorial: creación del rol de ejecución de IAM](#) antes de crear el primer entorno de computación.

Creación de un par de claves para las instancias

AWS utiliza criptografía de clave pública para proteger la información de inicio de sesión de la instancia. Una instancia de Linux, como una instancia contenedora de un entorno de AWS Batch cómputo, no tiene ninguna contraseña que pueda usarse para acceder a SSH. Utilice un par de claves para iniciar sesión de forma segura en la instancia. Primero se especifica el nombre del par de claves al crearse el entorno de computación y, a continuación, se proporciona la clave privada al iniciar sesión con SSH.

Si aún no ha creado un par de claves, puede crear uno con la consola de Amazon EC2. Tenga en cuenta que, si planea lanzar instancias en varias Regiones de AWS, cree un key pair en cada región. Para obtener más información acerca de las regiones, consulte [Regiones y zonas de disponibilidad](#) en la Guía del usuario de Amazon EC2.

Creación de un par de claves

1. Abra la consola de Amazon EC2 en <https://console.aws.amazon.com/ec2/>.
2. En la barra de navegación, seleccione una Región de AWS para el key pair. Se puede seleccionar cualquier región disponible, independientemente de su ubicación. Sin embargo, cada par de claves corresponde a una región específica. Por ejemplo, si tiene previsto lanzar una instancia en la región Oeste de EE. UU. (Oregón), debe crear un par de claves para la instancia en la misma región.
3. En el panel de navegación, seleccione Key Pairs (Pares de claves), Create Key Pair (Creación de par de claves).

4. En el cuadro de diálogo Create Key Pair (Creación de par de claves), en Key pair name (Nombre del par de claves), escriba un nombre para el nuevo par de claves y, a continuación, seleccione Create (Creación de). Elija un nombre que pueda recordar, como su nombre de usuario, seguido de -key-pair y del nombre de la región. Por ejemplo, yopar-de-clavesuswest2.
5. Su navegador descargará el archivo de clave privada automáticamente. El nombre de archivo base es el nombre que especificó como nombre del par de claves y la extensión del archivo es .pem. Guarde el archivo de clave privada en un lugar seguro.

 Important

Esta es la única oportunidad para guardar el archivo de clave privada. Necesita proporcionar el nombre del par de claves al lanzar una instancia y la clave privada correspondiente cada vez que se conecte a dicha instancia.

6. Si usa un cliente SSH en un equipo Mac o Linux para conectarse a su instancia de Linux, utilice el comando a continuación para establecer los permisos de su archivo de clave privada. De esa forma, solo usted podrá leerlo.

```
$ chmod 400 your_user_name-key-pair-region_name.pem
```

Para más información, consulte [Amazon EC2 Key Pairs](#) en la Guía del usuario de Amazon EC2.

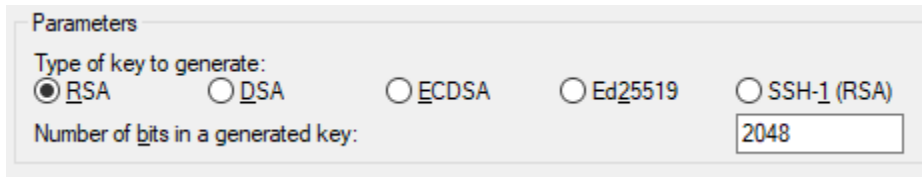
Para conectarse a la instancia mediante el par de claves

Para conectarse a la instancia de Linux desde un equipo que ejecute Mac OS o Linux, especifique el archivo .pem a su cliente SSH con la opción `-i` y la ruta a su clave privada. Para conectarte a tu instancia de Linux desde un ordenador que ejecute Windows, usa PuTTY, MindTerm o PuTTY. Si tiene previsto utilizar PuTTY, instálelo y utilice el siguiente procedimiento para convertir el archivo .pem en un archivo .ppk.

(Opcional) Para prepararse para conectarse a una instancia de Linux desde Windows mediante PuTTY

1. Descargue PuTTY de <http://www.chiark.greenend.org.uk/~sgtatham/putty/> e instálelo. Asegúrese de instalar el conjunto completo.
2. Iniciar TTYgen (por ejemplo, en el menú Inicio, elija Todos los programas, PuTTY y PuTTYgen).

3. En Type of key to generate (Tipo de clave a generar), elija RSA. Si utilizas una versión anterior de PuTTYgen, selecciona SSH-2 RSA.



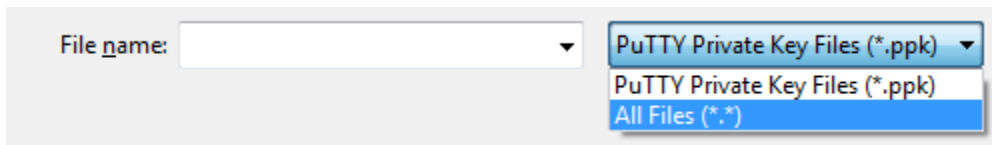
Parameters

Type of key to generate:

☒ RSA ☐ DSA ☐ ECDSA ☐ Ed25519 ☐ SSH-1 (RSA)

Number of bits in a generated key: 2048

4. Elija Load (Cargar). De forma predeterminada, PuTTYgen muestra solo los archivos con la extensión. .ppk Para localizar el archivo .pem, seleccione la opción de mostrar todos los tipos de archivo.



File name:

PuTTY Private Key Files (*.ppk) ▼

PuTTY Private Key Files (*.ppk)

All Files (*.*)

5. Seleccione el archivo de clave privada que creó en el procedimiento anterior y elija Open (Abrir). Elija OK (Aceptar) para descartar el cuadro de diálogo de confirmación.
6. Elija Save private key (Guardar clave privada). PuTTYgen muestra una advertencia sobre cómo guardar la clave sin contraseña. Seleccione Yes.
7. Especifique para la clave el mismo nombre que utilizó para el par de claves. PuTTY añade la extensión de archivo .ppk automáticamente.

Creación de una VPC

Con Amazon Virtual Private Cloud (Amazon VPC), puede lanzar recursos de AWS en una red virtual que haya definido. Recomendamos lanzar las instancias de contenedor en una VPC.

Si dispone de una VPC predeterminada, también puede omitir esta sección y pasar a la siguiente tarea [Creación de un grupo de seguridad](#). Para determinar si dispone de una VPC predeterminada, consulte [Plataformas compatibles con la consola de Amazon EC2](#) en la Guía del usuario de Amazon EC2.

Para obtener información acerca de cómo crear una VPC de Amazon, consulte [Creación de una VPC](#) en la Guía del usuario de Amazon VPC. Consulte la siguiente tabla para determinar qué opciones seleccionar.

Opción	Valor	
Recursos para crear	VPC solo	
Nombre	De manera opcional, indique un nombre para su VPC.	
IPv4 CIDR block	Entrada manual de IPv4 CIDR El bloque de CIDR debe ser de un tamaño de entre /16 y /28.	
IPv6 CIDR block	No hay bloque de CIDR IPv6	
Propiedad	Predeterminado/a	

Para obtener más información acerca de Amazon VPC, consulte [¿Qué es Amazon VPC?](#) en la Guía del usuario de Amazon VPC.

Creación de un grupo de seguridad

Los grupos de seguridad actúan como un cortafuegos para las instancias de contenedor de entorno de computación asociadas, al controlar el tráfico entrante y saliente en el nivel de instancia del contenedor. El grupo de seguridad solo se puede utilizar en la VPC para la que se creó.

Es posible añadir reglas a un grupo de seguridad que le permita conectarse a la instancia de contenedor desde su dirección IP mediante SSH. También se pueden añadir reglas que permitan HTTP de entrada y salida y acceso HTTPS desde cualquier lugar. Añada reglas para abrir los puertos requeridos por las tareas.

Recuerde que si pretende lanzar instancias de contenedor en varias regiones deberá crear un grupo de seguridad por región. Para obtener más información, consulte [Regiones y zonas de disponibilidad](#) en la Guía del usuario de Amazon EC2.

Note

Necesitará la dirección IP pública de su equipo local, que puede obtener mediante un servicio. Por ejemplo, proporcionamos el siguiente servicio: <http://checkip.amazonaws.com/>

o <https://checkip.amazonaws.com/>. Para buscar otro servicio que le brinde su dirección IP, utilice la frase de búsqueda "what is my IP address" (cuál es mi dirección IP). Si se conecta a través de un proveedor de servicios de internet (ISP) o protegido por un firewall sin una dirección IP estática, deberá identificar el rango de direcciones IP utilizadas por los equipos cliente.

Para crear un grupo de seguridad con la consola

1. Abra la consola de Amazon VPC en <https://console.aws.amazon.com/vpc/>.
2. En el panel de navegación, elija Security Groups (Grupos de seguridad).
3. Elija Create Security Group (Creación de grupo de seguridad).
4. Ingrese un nombre y una descripción para el grupo de seguridad. No puede cambiar el nombre ni la descripción de un grupo de seguridad después de crearlo.
5. En VPC, elija la VPC.
6. (Opcional) De forma predeterminada, los grupos de seguridad nuevos comienzan con una única regla de salida que permite que todo el tráfico salga del recurso. Debe añadir reglas para permitir el tráfico entrante o restringir el tráfico saliente.

Las instancias de contenedor de AWS Batch no requieren la apertura de ningún puerto de entrada. Sin embargo, es posible que desee agregar una regla SSH. De ese modo, puede iniciar sesión en la instancia de contenedor y examinar los contenedores en los trabajos con comandos de Docker. Si desea que su instancia de contenedor aloje un trabajo que ejecute un servidor web, también puede agregar reglas para HTTP. Siga los pasos a continuación para añadir estas reglas de grupo de seguridad opcionales.

En la pestaña Entrada cree las siguientes reglas y seleccione Creación de:

- Seleccione Add Rule (Agregar regla). En Tipo, seleccione HTTP. En Fuente, elija Cualquier lugar (0.0.0.0/0).
- Seleccione Add Rule (Agregar regla). En Tipo, seleccione SSH. En Fuente, elija IP personalizada y especifique la dirección IP pública de su computadora o red en notación de enrutamiento entre dominios sin clases (CIDR). Si su empresa asigna direcciones de un rango, especifíquelo; por ejemplo, 203.0.113.0/24. Para especificar una dirección IP individual en notación CIDR, elija Mi IP. Esto añade el prefijo de enrutamiento /32 a la dirección IP pública.

 Note

Por motivos de seguridad, recomendamos que no permita acceso SSH desde todas las direcciones IP (0.0.0.0/0) a su instancia, excepto con fines de prueba y solamente durante un breve periodo.

7. Puede agregar etiquetas ahora o más adelante. Para agregar una etiqueta, elija Agregar nueva etiqueta y, a continuación, ingrese la clave y el valor de la etiqueta.
8. Elija Creación de grupo de seguridad.

Para crear un grupo de seguridad mediante una línea de comando, vea [>create-security-group](#) (AWS CLI)

Para obtener más información acerca de los grupos de seguridad, consulte [Trabajar con grupos de seguridad](#).

Instala la AWS CLI

Para utilizar la AWS CLI con AWS Batch, instale la última versión de la AWS CLI. Para obtener más información acerca de cómo instalar la AWS CLI o cómo actualizarla a la versión más reciente, consulte [Instalación de la interfaz de la línea de comandos de AWS](#) en la Guía del usuario de AWS Command Line Interface.

Cómo empezar con AWS Batch los tutoriales

Puede utilizar el asistente AWS Batch de primera ejecución para empezar rápidamente. AWS Batch Tras completar los requisitos previos, puede utilizar el asistente de primera ejecución para crear un entorno de computación, una definición de trabajos y una cola de trabajos.

También puede enviar un ejemplo de trabajo de «Hello World» mediante el asistente de AWS Batch primera ejecución para probar la configuración. Si ya tiene una imagen de Docker con la que quiere lanzarla AWS Batch, puede utilizarla para crear una definición de trabajo.

Después, puedes usar el asistente de AWS Batch primera ejecución para crear un entorno informático, una cola de trabajos y enviar un ejemplo de trabajo de Hello World.

Introducción a la orquestación de Amazon EC2 con el asistente

Amazon Elastic Compute Cloud (Amazon EC2) brinda capacidad de computación escalable en la Nube de AWS. El uso de Amazon EC2 elimina la necesidad de invertir inicialmente en hardware, de manera que puede desarrollar e implementar aplicaciones en menos tiempo.

Puede usar Amazon EC2 para lanzar tantos servidores virtuales como necesite, configurar la seguridad y las redes, y administrar el almacenamiento. Amazon EC2 le permite escalar hacia arriba o hacia abajo para controlar los cambios en los requisitos o los picos de popularidad, con lo que se reduce la necesidad de prever el tráfico.

Descripción general

En este tutorial se muestra cómo realizar la configuración de AWS Batch con el asistente para configurar Amazon EC2 y ejecutar Hello World.

Destinatarios previstos

Este tutorial está diseñado para los administradores y desarrolladores de sistemas responsables de la configuración, las pruebas y la implementación de AWS Batch.

Características utilizadas

Este tutorial muestra cómo utilizar el asistente de la consola de AWS Batch para realizar lo siguiente:

- crear y configurar un entorno de computación de Amazon EC2,

- crear una cola de trabajos,
- Creación de una definición de trabajo
- crear y enviar un trabajo para que se ejecute,
- ver la salida del trabajo en CloudWatch.

Tiempo necesario

Debe tardar entre 10 y 15 minutos en completar este tutorial.

Restricciones regionales

No hay restricciones regionales o de país asociadas al uso de esta solución.

Costos de uso de recursos

La creación de una cuenta de AWS no supone ningún coste. Sin embargo, mediante la implementación de esta solución, es posible que incurra en algunos o todos los costos que se enumeran en la siguiente tabla.

Descripción	Costo (dólares estadounidenses)
Instancia de Amazon EC2	Usted paga por cada instancia de Amazon EC2 creada. Para obtener más información acerca de los precios, consulte Precios de Amazon EC2 .

Requisitos previos

Antes de empezar:

- Cree una Cuenta de AWS si aún no dispone de una.
- Cree la [instancia del rol `ecsInstanceRole`](#).

Paso 1: Crear un entorno de computación

Important

Para comenzar de la manera más sencilla y rápida posible, en este tutorial se incluirán los pasos con la configuración predeterminada. Antes de crear para su uso en producción,

recomendamos que conozca toda la configuración e implemente la configuración que satisfaga sus requisitos.

Para crear un entorno de computación para una orquestación de Amazon EC2, haga lo siguiente:

1. Abra el [Asistente de primer uso de la consola AWS Batch](#).
2. En Configuración el tipo de orquestación y trabajo, seleccione Amazon Elastic Compute Cloud (Amazon EC2).
3. Seleccione Siguiente.
4. En la sección de Configuración de entorno de computación de Nombre, especifique un nombre único para su entorno de computación. El nombre puede tener una longitud máxima de 128 caracteres. Puede contener letras mayúsculas y minúsculas, números, guiones (-) y guiones bajos (_).
5. En Rol de instancia, elija un rol de instancia existente que tenga asociados los permisos de IAM necesarios. Este rol de instancia permite a las instancias de contenedor de Amazon ECS en su entorno de computación realizar llamadas a las API de AWS necesarias. Para obtener más información, consulte [Rol de instancia de Amazon ECS](#).

El nombre predeterminado de la instancia del rol es `ecsInstanceRole`.

6. En Configuración de instancia, puede dejar la configuración predeterminada.
7. En Configuración de red, utilice su VPC predeterminada para la Región de AWS.
8. Elija Siguiente.

Paso 2: Crear una cola de trabajo

Una cola de trabajo almacena los trabajos que ha presentado hasta que el programador de AWS Batch los ejecuta en un recurso dentro del entorno de computación. Para obtener más información, consulte [Colas de trabajos](#)

Para crear una cola de trabajos para una orquestación de Amazon EC2, haga lo siguiente:

1. En Configuración de la cola de trabajos, Nombre, especifique un nombre único para la cola de trabajos. El nombre puede tener una longitud máxima de 128 caracteres. Puede contener letras mayúsculas y minúsculas, números, guiones (-) y guiones bajos (_).
2. Puede dejar el valor predeterminado en todas las demás opciones de configuración.

3. Elija Siguiente.

Paso 3: crear una definición de trabajo

AWS Batch Las definiciones de trabajo de especifican cómo se ejecutan los trabajos. Si bien cada trabajo debe hacer referencia a una definición de trabajo, muchos de los parámetros especificados en dicha definición pueden ser ignorados en tiempo de ejecución.

Para crear la definición de trabajo:

1. En Crear una definición de trabajo
 - a. en Nombre, escriba un nombre único para su cola de trabajos. El nombre puede tener una longitud máxima de 128 caracteres. Puede contener letras mayúsculas y minúsculas, números, guiones (-) y guiones bajos (_).
 - b. En Comando: optional, puede cambiar `hello world` a un mensaje personalizado o dejarlo como está.
2. Puede dejar el valor predeterminado en todas las demás opciones de configuración.
3. Elija Siguiente.

Paso 4: Crear un trabajo

Para crear un trabajo, haga lo siguiente:

1. En Configuración de trabajo, especifique un Nombre único para el trabajo. El nombre puede tener una longitud máxima de 128 caracteres. Puede contener letras mayúsculas y minúsculas, números, guiones (-) y guiones bajos (_).
2. Puede dejar el valor predeterminado en todas las demás opciones de configuración.
3. Seleccione Siguiente.

Paso 5: Revisar y crear

En la página Revisar y crear, revise los pasos de configuración. Si necesita realizar cambios, elija Editar. Cuando haya terminado, seleccione Creación de recursos.

1. En Revisar y crear, seleccione Crear recursos.

2. Se abre una ventana cuando AWS Batch comienza a asignar sus recursos. Una vez completado, seleccione Ir al panel. En el panel, verá todos los recursos que tiene asignados y también que el trabajo se encuentra en el estado `Runnable`. Su tarea está programada para ejecutarse y se completará en un plazo de 2 a 3 minutos.

Paso 6: Ver la salida de la tarea

Para ver la salida del trabajo, realice lo siguiente:

1. En el panel de navegación, seleccione Trabajos.
2. En el menú desplegable de Colas de trabajo, elija la cola de trabajos que creó para el tutorial.
3. La tabla Trabajos muestra todos los trabajos y su estado actual. Una vez que el Estado del trabajo sea `Correcto`, seleccione el Nombre del trabajo para consultar sus detalles.
4. En el panel Detalles, elija el Nombre del flujo de registro. Se abrirá la consola CloudWatch del trabajo junto con un evento con el Mensaje de `hello world` o su mensaje personalizado.

Paso 7: Eliminar los recursos del tutorial

Se le cobrará por la instancia de Amazon EC2 mientras esté habilitada. Puede eliminar la instancia para dejar de generar cargos.

Para eliminar los recursos que usted creó, realice lo que se indica a continuación:

1. En el panel de navegación, elija Cola de trabajos.
2. En la tabla Cola de trabajos, elija la cola de trabajos que creó para el tutorial.
3. Elija Deshabilitar. Una vez que el Estado de la cola de trabajos esté deshabilitado, puede elegir Eliminar.
4. Una vez se elimine la cola de trabajos, en el panel de navegación, elija Entornos de computación.
5. Elija el entorno de computación que creó para este tutorial y, a continuación, elija Deshabilitar. El entorno de computación puede tardar entre 1 y 2 minutos en terminar de deshabilitarse.
6. Una vez que el Estado del entorno de computación esté deshabilitado, seleccione Eliminar. La eliminación del entorno de computación puede tardar entre 1 y 2 minutos.

Recursos adicionales

Después de completar el tutorial, recomendamos consultar alguno de los siguientes temas:

- Consulte los componentes de AWS Batch principales. Para obtener más información, consulte [Componentes de AWS Batch](#).
- Obtenga más información sobre los diferentes [entornos de computación](#) disponibles en AWS Batch.
- Obtenga más información sobre las [colas de trabajos](#) y sus diferentes opciones de programación.
- Obtenga más información sobre las [definiciones de trabajos](#) y las diferentes opciones de configuración.
- Obtenga más información sobre los distintos tipos de [trabajos](#).

Introducción a la orquestación AWS Batch de Fargate con el Mago

AWS Fargate lanza y escala el cómputo para que se ajuste perfectamente a los requisitos de recursos que especifique para el contenedor. Con Fargate, no es necesario aprovisionar en exceso los servidores ni pagar por ellos. Para obtener más información, consulte [Fargate](#).

Descripción general de

En este tutorial se muestra cómo realizar la configuración AWS Batch con el asistente para configurar AWS Fargate y ejecutarlo. Hello World

Público objetivo

Este tutorial está diseñado para los administradores y desarrolladores de sistemas responsables de la configuración, las pruebas y la implementación de AWS Batch.

Características utilizadas

En este tutorial se muestra cómo utilizar el asistente de AWS Batch consola para:

- Cree y configure un entorno AWS informático Fargate
- crear una cola de trabajos,
- crear una definición de trabajo,
- crear y enviar un flujo de trabajo para ejecutarlo,

- Ve el resultado del trabajo en CloudWatch

Tiempo requerido

Completar este tutorial debería tardar entre 10 y 15 minutos.

Restricciones regionales

No hay restricciones regionales o de país asociadas al uso de esta solución.

Costos de uso de recursos

La creación de una AWS cuenta es gratuita. Sin embargo, mediante la implementación de esta solución, es posible que incurra en algunos o todos los costos que se enumeran en la siguiente tabla.

Description (Descripción)	Costo (dólares estadounidenses)
Los precios se basan en la vCPU, la memoria, los sistemas operativos, la arquitectura de la CPU y los recursos de almacenamiento solicitados para la tarea o el pod.	Para obtener más información acerca de los precios, consulte la página Precios de Fargate .

Requisitos previos

Antes de empezar:

- Crea una Cuenta de AWS si no tienes una.
- cree el rol de ejecución de tareas. Si aún no ha creado el [rol de ejecución de tareas](#), puede crearlo como parte de este tutorial.

Paso 1: Crear un entorno de computación

Important

Para comenzar de la manera más sencilla y rápida posible, en este tutorial se incluirán los pasos con la configuración predeterminada. Antes de crear para su uso en producción, recomendamos que conozca toda la configuración e implemente la configuración que satisfaga sus requisitos.

Para crear un entorno de computación para una orquestación de Fargate, haga lo siguiente:

1. Abra el [Asistente de primer uso de la consola AWS Batch](#).
2. En Configurar trabajo y tipo de orquestación, elija Fargate.
3. Elija Siguiente.
4. En la sección de Configuración de entorno de computación de Nombre, especifique un nombre único para su entorno de computación. El nombre puede tener una longitud máxima de 128 caracteres. Puede contener letras mayúsculas y minúsculas, números, guiones (-) y guiones bajos (_).
5. Puede dejar el valor predeterminado en todas las demás opciones de configuración.
6. Elija Siguiente.

Paso 2: Crear una cola de trabajo

Una cola de trabajos almacena los trabajos enviados hasta que el AWS Batch programador ejecute el trabajo en un recurso de su entorno informático. Para crear una cola de trabajos:

Para crear una cola de trabajos para una orquestación de Fargate, haga lo siguiente:

1. En la sección de Configuración de cola de trabajo de Nombre, especifique un nombre único para su entorno de computación. El nombre puede tener una longitud máxima de 128 caracteres. Puede contener letras mayúsculas y minúsculas, números, guiones (-) y guiones bajos (_).
2. En Prioridad, ingrese 900 para la cola de trabajos.
3. Puede dejar el valor predeterminado en todas las demás opciones de configuración.
4. Elija Siguiente.

Paso 3: Crea una definición de trabajo

Para crear la definición de trabajo:

1. En la sección de Configuración general:
 - En la sección de Configuración general de Nombre, especifique un nombre único para su entorno de computación. El nombre puede tener una longitud máxima de 128 caracteres. Puede contener letras mayúsculas y minúsculas, números, guiones (-) y guiones bajos (_).
2. En la sección de Configuración de la plataforma Fargate:

- a. Active Asignar IP pública para asignar una dirección IP pública. Necesita una IP pública para descargar la imagen del contenedor, a menos que haya configurado un repositorio de imágenes privado.
- b. Para la función de ejecución, elija una función de ejecución de tareas que permita a los agentes de Amazon Elastic Container Service (Amazon ECS) AWS realizar llamadas en su nombre. Elija `ecsTaskExecutionRole` o `BatchEcsTaskExecutionRole`.

Para crear el Rol de ejecución, seleccione Crear un rol de ejecución. En el modal Crear un rol de IAM, seleccione Crear rol de IAM.

- i. La consola de IAM ya tiene la configuración de permisos configurada para crear el rol de ejecución.
- ii. En Tipo de entidad de confianza, verifique que el Servicio de AWS esté seleccionado.
- iii. En Caso de servicio o usuario, compruebe que Elastic Container Service esté seleccionado.
- iv. Elija Siguiente.
- v. Para las políticas de permisos, verifica que Amazon ECSTask ExecutionRolePolicy esté seleccionado.
- vi. Elija Siguiente.
- vii. En Nombre, revise y cree, compruebe que el nombre del rol es `BatchEcsTaskExecutionRole`.
- viii. Seleccione Crear rol.
- ix. En la AWS Batch consola, pulse el botón de actualización situado junto a la función de ejecución. Elija el rol `BatchEcsTaskExecutionRole` de ejecución.

3. En la sección Configuración del contenedor:

- En Comando, puede cambiar `hello world` con un mensaje personalizado o dejarlo como está.

4. Puede dejar el valor predeterminado en todas las demás opciones de configuración.

5. Elija Siguiente.

Paso 4: Crear un trabajo

Para crear un trabajo de Fargate, haga lo siguiente:

1. En Configuración de trabajo, especifique un Nombre único para el trabajo. El nombre puede tener una longitud máxima de 128 caracteres. Puede contener letras mayúsculas y minúsculas, números, guiones (-) y guiones bajos (_).
2. Puede dejar el valor predeterminado en todas las demás opciones de configuración.
3. Elija Siguiente.

Paso 5: Revisar y crear

En la página Revisar y crear, revise los pasos de configuración. Si necesita realizar cambios, elija Editar. Cuando haya terminado, seleccione Creación de recursos.

Paso 6: Ver la salida de la tarea

Para ver la salida del trabajo, realice lo siguiente:

1. En el panel de navegación, seleccione Trabajos.
2. En el menú desplegable de Colas de trabajo, elija la cola de trabajos que creó para el tutorial.
3. La tabla Trabajos muestra todos los trabajos y su estado actual. Una vez que el Estado del trabajo sea Correcto, seleccione el Nombre del trabajo para consultar sus detalles.
4. En el panel Detalles, elija el Nombre del flujo de registro. Se abrirá la CloudWatch consola del Job y debería haber un evento con el Mensaje de `hello world` o su mensaje personalizado.

Paso 7: Eliminar los recursos del tutorial

Se le cobrará por la EC2 instancia de Amazon mientras esté habilitada. Puede eliminar la instancia para dejar de generar cargos.

Para eliminar los recursos que usted creó, haga lo que se indica a continuación:

1. En el panel de navegación, elija Cola de trabajos.
2. En la tabla Cola de trabajos, elija la cola de trabajos que creó para el tutorial.
3. Elija Deshabilitar. Una vez que el Estado de la cola de trabajos esté deshabilitado, puede elegir Eliminar.
4. Una vez se elimine la cola de trabajos, en el panel de navegación, elija Entornos de computación.

5. Elija el entorno de computación que creó para este tutorial y, a continuación, elija Deshabilitar. El entorno de computación puede tardar entre 1 y 2 minutos en terminar de deshabilitarse.
6. Una vez que el Estado del entorno de computación esté deshabilitado, seleccione Eliminar. La eliminación del entorno de computación puede tardar entre 1 y 2 minutos.

Recursos adicionales

Después de completar el tutorial, recomendamos consultar alguno de los siguientes temas:

- Obtenga más información sobre las [prácticas recomendadas](#).
- Explore los componentes AWS Batch principales. Para obtener más información, consulte [Componentes de AWS Batch](#).
- Obtenga más información sobre los diferentes [entornos de computación](#) disponibles en AWS Batch.
- Obtenga más información sobre las [colas de trabajos](#) y sus diferentes opciones de programación.
- Obtenga más información sobre las [definiciones de trabajos](#) y las diferentes opciones de configuración.
- Obtenga más información sobre los distintos tipos de [trabajos](#).

Cómo empezar a AWS Batch usar Fargate el AWS CLI

En este tutorial se muestra cómo configurar AWS Batch la AWS Fargate orquestación y cómo ejecutar un sencillo trabajo de «Hello World» con la AWS Command Line Interface (AWS CLI). Aprenderá a crear entornos de computación, colas de trabajos, definiciones de trabajos y enviar trabajos a AWS Batch.

Temas

- [Requisitos previos](#)
- [Creación de un rol de ejecución de IAM](#)
- [Creación de un entorno de computación](#)
- [Cree una cola de trabajos](#)
- [crear una definición de trabajo,](#)
- [Envío y supervisión de un trabajo](#)

- [Visualización de la salida de un trabajo](#)
- [Eliminar recursos](#)
- [Paso a la producción](#)
- [Siguiendo pasos](#)

Requisitos previos

Antes de comenzar este tutorial, asegúrese de tener lo siguiente:

1. El AWS CLI. Si necesita instalarla, siga la [guía de instalación de la AWS CLI](#). También puede [usar AWS CloudShell](#), que incluye el AWS CLI.
2. Lo configuró AWS CLI con las credenciales adecuadas. Si aún no ha configurado sus credenciales, ejecute `aws configure`
3. Conocimiento básico de las interfaces de línea de comandos y los conceptos de contenedorización
4. [¿Cómo AWS Batch funciona con IAM](#) para crear y gestionar AWS Batch recursos, funciones de IAM y recursos de VPC en su. Cuenta de AWS
5. Un ID de subred y un ID de grupo de seguridad de una VPC de su. Cuenta de AWS Si no dispone de una VPC, puede [crear una](#). Para obtener más información sobre cómo usar el recurso AWS CLI para recuperar estos recursos IDs, consulte [describe-subnets](#) y en la Referencia de comandos. [describe-security-groups](#) AWS CLI

Tiempo necesario: Este tutorial se completa entre aproximadamente 15 y 20 minutos.

Costo: Este tutorial utiliza los recursos de computación de Fargate. El costo estimado para completar este tutorial es menos de 0,01 USD, siempre que siga las instrucciones de eliminación para eliminar los recursos inmediatamente después de completarlo. Los precios de Fargate se basan en los recursos de vCPU y memoria consumidos, y se cobran por segundo con un mínimo de 1 minuto. Para obtener información acerca de los precios, consulte [Precios de AWS Fargate](#).

Creación de un rol de ejecución de IAM

AWS Batch requiere una función de ejecución que permita a los agentes de Amazon Elastic Container Service (Amazon ECS) AWS realizar llamadas a la API en su nombre. Esta función es necesaria para que Fargate extraiga imágenes de contenedores y escriba registros para Amazon CloudWatch

Creación de un documento de política de confianza

Primero, cree una política de confianza que permita al servicio de Amazon ECS asumir el rol.

```
cat > batch-execution-role-trust-policy.json << EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "ecs-tasks.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
EOF
```

Cree el rol de ejecución

El siguiente comando crea un rol de IAM con el nombre `BatchEcsTaskExecutionRoleTutorial` mediante la política de confianza que usted creó.

```
aws iam create-role \
  --role-name BatchEcsTaskExecutionRoleTutorial \
  --assume-role-policy-document file://batch-execution-role-trust-policy.json
```

Asociación de la política requerida

Adjunte la política AWS gestionada que proporciona los permisos necesarios para la ejecución de las tareas de Amazon ECS.

```
aws iam attach-role-policy \
  --role-name BatchEcsTaskExecutionRoleTutorial \
  --policy-arn arn:aws:iam::aws:policy/service-role/AmazonECSTaskExecutionRolePolicy
```

La función ya está lista para ser utilizada en la ejecución AWS Batch de tareas de Fargate.

Creación de un entorno de computación

Un entorno de computación define los recursos de computación en los que se ejecutarán los trabajos por lotes. Para este tutorial, creará un entorno de computación administrado por Fargate que aprovisiona y escale automáticamente los recursos según los requisitos del trabajo.

Creación de un entorno de computación

El siguiente comando crea un entorno de computación para Fargate. Sustituya la subred y el grupo de seguridad de ejemplo por IDs los suyos propios según el. [Requisitos previos](#)

```
aws batch create-compute-environment \  
  --compute-environment-name my-fargate-compute-env \  
  --type MANAGED \  
  --state ENABLED \  
  --compute-resources type=FARGATE,maxvCpus=128,subnets=subnet-  
a123456b,securityGroupIds=sg-a12b3456
```

A continuación, se muestra cómo se ve la salida cuando el comando se ejecuta correctamente.

```
{  
  "computeEnvironmentName": "my-fargate-compute-env",  
  "computeEnvironmentArn": "arn:aws:batch:us-west-2:123456789012:compute-environment/  
my-fargate-compute-env"  
}
```

Espera del entorno de computación hasta que esté listo

Compruebe el estado de su entorno de computación para asegurarse de que esté listo antes de continuar.

```
aws batch describe-compute-environments \  
  --compute-environments my-fargate-compute-env \  
  --query 'computeEnvironments[0].status'
```

```
"VALID"
```

Cuando se muestre el estado VALID, su entorno de computación estará preparado para aceptar trabajos.

Cree una cola de trabajos

Una cola de trabajos almacena los trabajos enviados hasta que el AWS Batch programador los ejecute en los recursos de su entorno informático. Los trabajos se procesan en orden de prioridad dentro de la cola.

Creación de una cola de trabajos

El siguiente comando crea una cola de trabajos con prioridad 900 que utiliza su entorno informático de Fargate.

```
aws batch create-job-queue \  
  --job-queue-name my-fargate-job-queue \  
  --state ENABLED \  
  --priority 900 \  
  --compute-environment-order order=1,computeEnvironment=my-fargate-compute-env
```

A continuación, se muestra cómo se ve la salida cuando el comando se ejecuta correctamente.

```
{  
  "jobQueueName": "my-fargate-job-queue",  
  "jobQueueArn": "arn:aws:batch:us-west-2:123456789012:job-queue/my-fargate-job-queue"  
}
```

Comprobación de que la cola de trabajos esté lista

Compruebe que la cola de trabajos está en el estado ENABLED y lista para aceptar trabajos.

```
aws batch describe-job-queues \  
  --job-queues my-fargate-job-queue \  
  --query 'jobQueues[0].state' "ENABLED"
```

crear una definición de trabajo,

Una definición de trabajo especifica cómo se ejecutan los trabajos, como la imagen de Docker que se va a usar, los requisitos de recursos y otros parámetros. Para Fargate, utilizará los requisitos de recursos en lugar de los parámetros tradicionales de vCPU y memoria.

Creación de la definición de trabajo

El siguiente comando crea una definición de trabajo que ejecuta un comando simple de “Hola, mundo” utilizando la imagen del contenedor de busybox. 123456789012Sustitúyelo por su Cuenta de AWS ID real y sustituya el ejemplo por el Región de AWS suyo propio.

```
aws batch register-job-definition \
  --job-definition-name my-fargate-job-def \
  --type container \
  --platform-capabilities FARGATE \
  --container-properties '{
    "image": "busybox",
    "resourceRequirements": [
      {"type": "VCPU", "value": "0.25"},
      {"type": "MEMORY", "value": "512"}
    ],
    "command": ["echo", "hello world"],
    "networkConfiguration": {
      "assignPublicIp": "ENABLED"
    },
    "executionRoleArn": "arn:aws:iam::123456789012:role/
BatchEcsTaskExecutionRoleTutorial"
  },
  {
    "jobDefinitionName": "my-fargate-job-def",
    "jobDefinitionArn": "arn:aws:batch:us-west-2:123456789012:job-definition/my-
fargate-job-def:1",
    "revision": 1
  }'
```

La definición del trabajo especifica 0,25 vCPU y 512 MB de memoria, que son los recursos mínimos para una tarea de Fargate. La configuración assignPublicIp está habilitada para que el contenedor pueda extraer la imagen de Busybox de Docker Hub.

Envío y supervisión de un trabajo

Ahora que tiene todos los componentes necesarios, puede enviar un trabajo a la cola y supervisar su progreso.

Envía un trabajo

El siguiente comando envía un trabajo a la cola utilizando la definición de trabajo que ha creado.

```
aws batch submit-job \
```

```
--job-name my-hello-world-job \  
--job-queue my-fargate-job-queue \  
--job-definition my-fargate-job-def
```

A continuación, se muestra cómo se ve la salida cuando el comando se ejecuta correctamente.

```
{  
  "jobArn": "arn:aws:batch:us-west-2:123456789012:job/my-hello-world-job",  
  "jobName": "my-hello-world-job",  
  "jobId": "1509xmpl-4224-4da6-9ba9-1d1acc96431a"  
}
```

Observe el `jobId` devuelto en la respuesta, ya que lo utilizará para supervisar el progreso del trabajo.

Supervisión del estado de trabajo

Utilice el ID del trabajo para verificar el estado del trabajo. El trabajo progresará a través de diversos estados: SUBMITTED, PENDING, RUNNABLE, STARTING, RUNNING, y finalmente, SUCCEEDED o FAILED.

```
aws batch describe-jobs --jobs 1509xmpl-4224-4da6-9ba9-1d1acc96431a
```

A continuación, se muestra cómo se ve la salida cuando el comando se ejecuta correctamente.

```
{  
  "jobs": [  
    {  
      "jobArn": "arn:aws:batch:us-west-2:123456789012:job/my-hello-world-job",  
      "jobName": "my-hello-world-job",  
      "jobId": "1509xmpl-4224-4da6-9ba9-1d1acc96431a",  
      "jobQueue": "arn:aws:batch:us-west-2:123456789012:job-queue/my-fargate-job-queue",  
      "status": "SUCCEEDED",  
      "createdAt": 1705161908000,  
      "jobDefinition": "arn:aws:batch:us-west-2:123456789012:job-definition/my-fargate-job-def:1"  
    }  
  ]  
}
```

Cuando se muestra el estado SUCCEEDED, el trabajo se habrá completado correctamente.

Visualización de la salida de un trabajo

Una vez finalizado el trabajo, podrás ver su resultado en Amazon CloudWatch Logs.

Obtención del nombre del flujo de registro

En primer lugar, recupere el nombre del flujo de registro desde los detalles del trabajo. Sustituya el ID de trabajo de ejemplo por el suyo.

```
aws batch describe-jobs --jobs 1509xmpl-4224-4da6-9ba9-1d1acc96431a \
  --query 'jobs[0].attempts[0].containers[0].logStreamName' \
  --output text
```

```
my-fargate-job-def/default/1509xmpl-4224-4da6-9ba9-1d1acc96431a
```

Visualización de los registros de trabajo

Utilice el nombre del flujo de registro para recuperar el resultado del trabajo desde CloudWatch Logs.

```
aws logs get-log-events \
  --log-group-name /aws/batch/job \
  --log-stream-name my-fargate-job-def/default/1509xmpl-4224-4da6-9ba9-1d1acc96431a \
  --query 'events[*].message' \
  --output text
```

La salida muestra “Hola, mundo”, lo que confirma que su trabajo se ha ejecutado correctamente.

Eliminar recursos

Elimine los recursos que haya creado en este tutorial para evitar cargos recurrentes. Debe eliminar los recursos en el orden correcto debido a las dependencias.

Deshabilite y elimine la cola de trabajos

En primer lugar, deshabilite la cola de trabajos y, a continuación, elimínela.

```
aws batch update-job-queue \
  --job-queue my-fargate-job-queue \
  --state DISABLED
```

```
aws batch delete-job-queue \  
  --job-queue my-fargate-job-queue
```

Deshabilite y elimine el entorno de computación

Una vez eliminada la cola de trabajos, deshabilite y elimine el entorno de computación.

```
aws batch update-compute-environment \  
  --compute-environment my-fargate-compute-env \  
  --state DISABLED
```

```
aws batch delete-compute-environment \  
  --compute-environment my-fargate-compute-env
```

Elimine el rol de IAM

Elimine la asociación de la política y elimine el rol de IAM.

```
aws iam detach-role-policy \  
  --role-name BatchEcsTaskExecutionRoleTutorial \  
  --policy-arn arn:aws:iam::aws:policy/service-role/AmazonECSTaskExecutionRolePolicy
```

```
aws iam delete-role \  
  --role-name BatchEcsTaskExecutionRoleTutorial
```

Elimine los archivos temporales

Elimine el archivo de política de confianza que creó.

```
rm batch-execution-role-trust-policy.json
```

Todos los recursos se han eliminado correctamente.

Paso a la producción

Este tutorial está diseñado para ayudarle a entender cómo AWS Batch funciona con Fargate. Para las implementaciones de producción, tenga en cuenta los siguientes requisitos adicionales:

Consideraciones de seguridad:

- Crear grupos de seguridad dedicados con un acceso mínimo obligatorio en lugar de utilizar los grupos de seguridad predeterminados.
- Utilizar subredes privadas con la puerta de enlace NAT en lugar de asignar direcciones IP públicas para los contenedores.
- Almacenar imágenes de contenedor en Amazon ECR en lugar de usar repositorios públicos.
- Implemente puntos finales de VPC para la comunicación del AWS servicio a fin de evitar el tráfico de Internet

Consideraciones de arquitectura:

- Desplegar en varias zonas de disponibilidad para lograr una alta disponibilidad.
- Implementar estrategias de reintentos de tareas y colas de mensajes fallidos para gestionar los errores.
- Utilizar varias colas de trabajos con diferentes prioridades para la administración de la carga de trabajo.
- Configurar políticas de escalado automático según la profundidad de las colas y la utilización de los recursos.
- Implementar la supervisión y las alertas en caso de que se produzcan errores en los trabajos y se utilicen los recursos.

Consideraciones operativas:

- Configure CloudWatch paneles y alarmas para la supervisión
- Implementar registros y registros de auditoría adecuados.
- Utilice CloudFormation o la AWS CDK infraestructura como código
- Establecer procedimientos de copia de seguridad y recuperación ante desastres.

Para obtener una guía completa sobre arquitecturas listas para producción, consulte [Maro de AWS Well-Architected](#) y [AWS Security Best Practices](#).

Siguientes pasos

Ahora que ha completado este tutorial, puede explorar AWS Batch funciones más avanzadas:

- [Colas de trabajos](#): obtenga información sobre la programación de colas de trabajos y la administración de prioridades
- [Definiciones de trabajo](#): conozca las configuraciones avanzadas de definición de trabajos, como las variables de entorno, los volúmenes y las estrategias de reintento
- [Entornos informáticos para AWS Batch](#): comprenda los diferentes tipos de entornos de computación y las opciones de escalado
- [Trabajos paralelos de varios nodos](#): ejecute trabajos que abarquen varios nodos de computación
- [Trabajos de matrices](#): envíe un gran número de trabajos similares de manera eficiente
- [Prácticas recomendadas de AWS Batch](#): aprenda técnicas de optimización para las cargas de trabajo de producción

Cómo empezar con AWS Batch Amazon EKS

AWS Batch on Amazon EKS es un servicio gestionado para programar y escalar cargas de trabajo por lotes en clústeres de Amazon EKS existentes. AWS Batch no crea, administra ni realiza operaciones del ciclo de vida de sus clústeres de Amazon EKS en su nombre. AWS Batch la orquestación amplía y reduce los nodos administrados por dichos nodos AWS Batch y los ejecuta en ellos.

AWS Batch no afecta a los nodos, ni a los grupos de nodos con escalado automático ni a los ciclos de vida de los pods que no estén asociados a los entornos AWS Batch informáticos de su clúster de Amazon EKS. AWS Batch Para funcionar de forma eficaz, su [función vinculada al servicio](#) necesita permisos de control de acceso Kubernetes basado en funciones (RBAC) en su clúster Amazon EKS existente. Para obtener más información, consulte [Utilización de la autorización de RBAC](#) en la documentación Kubernetes.

AWS Batch requiere un espacio de Kubernetes nombres en el que pueda clasificar los pods como trabajos. AWS Batch Recomendamos un espacio de nombres dedicado para aislar los AWS Batch pods de las demás cargas de trabajo del clúster.

Una vez AWS Batch que se le haya otorgado acceso a RBAC y se haya establecido un espacio de nombres, puede asociar ese clúster de Amazon EKS a un entorno AWS Batch informático mediante la operación de API. [CreateComputeEnvironment](#) Se puede asociar una cola de trabajos a este nuevo entorno informático de Amazon EKS. AWS Batch los trabajos se envían a la cola de trabajos en función de una definición de trabajo de Amazon EKS mediante la operación de [SubmitJob](#) API. AWS Batch a continuación, lanza los nodos AWS Batch gestionados y coloca los trabajos de la

cola de trabajos en forma de Kubernetes pods en el clúster EKS asociado a un entorno AWS Batch informático.

En las siguientes secciones se explica cómo configurar AWS Batch Amazon EKS.

Contenido

- [Descripción general de](#)
- [Requisitos previos](#)
- [Paso 1: Cree su clúster de Amazon EKS para AWS Batch](#)
- [Paso 2: Prepare el clúster de Amazon EKS para AWS Batch](#)
- [Paso 3: Crear un entorno de computación de Amazon EKS](#)
- [Paso 4: Cree una cola de trabajos y adjunte el entorno informático](#)
- [Paso 5: Crear una definición de trabajo](#)
- [Paso 6: Presentar un trabajo](#)
- [Paso 7: Ver la salida del trabajo](#)
- [Paso 8: \(opcional\) Enviar un trabajo con anulaciones](#)
- [Paso 9: Eliminar los recursos del tutorial](#)
- [Recursos adicionales](#)

Descripción general de

En este tutorial se muestra cómo realizar la configuración AWS Batch con Amazon EKS mediante AWS CLI, `kubectl` y `eksctl`.

Público al que va dirigido

Este tutorial está diseñado para los administradores y desarrolladores de sistemas responsables de la configuración, las pruebas y la implementación de AWS Batch.

Características utilizadas

En este tutorial se muestra cómo utilizar el AWS CLI, para:

- crear y configurar un entorno de computación de Amazon EKS,
- crear una cola de trabajos,
- crear una definición de trabajo,

- crear y enviar un flujo de trabajo para ejecutarlo,
- enviar un trabajo con anulaciones.

Tiempo requerido

Completar este tutorial debería tardar entre 30 y 40 minutos.

Restricciones regionales

No hay restricciones regionales o de país asociadas al uso de esta solución.

Costos de uso de recursos

La creación de una AWS cuenta no conlleva ningún cargo. Sin embargo, mediante la implementación de esta solución, es posible que incurra en algunos o todos los costos que se enumeran en la siguiente tabla.

Description (Descripción)	Costo (dólares estadounidenses)
El cobro será por hora de uso del clúster	Varía según la instancia. Consulte los precios de Amazon EKS

Requisitos previos

Antes de comenzar este tutorial, debe instalar y configurar las siguientes herramientas y recursos que necesita para crear y administrar tanto AWS Batch los recursos de Amazon EKS.

- **AWS CLI**: una herramienta de línea de comandos para trabajar con servicios de AWS , incluido Amazon EKS. Esta guía requiere que utilices la versión 2.8.6 o posterior o la 1.26.0 o posterior. Para obtener más información, consulte [Instalar, actualizar y desinstalar la AWS CLI](#) en la Guía del usuario de AWS Command Line Interface . Tras instalarlo AWS CLI, le recomendamos que también lo configure. Para obtener más información, consulte [Configuración rápida con aws configure](#) en la Guía del usuario de AWS Command Line Interface .
- **kubectl**: una herramienta de línea de comandos para trabajar con clústeres de Kubernetes. Esta guía requiere que utilice la versión 1.23 o una posterior. Para obtener más información, consulte [Instalación o actualización de kubectl](#) en la Guía del usuario de Amazon EKS.
- **eksctl**: una herramienta de línea de comandos para trabajar con clústeres de Amazon EKS que automatiza varias tareas individuales. Esta guía requiere que utilice la versión 0.115.0 o una

posterior. Para obtener más información, consulte [Instalación o actualización de eksctl](#) en la Guía del usuario de Amazon EKS.

- Permisos de IAM necesarios: el responsable de seguridad de IAM que utilice debe tener permisos para trabajar con las funciones de IAM de Amazon EKS y las funciones vinculadas a servicios CloudFormation, así como con una VPC y los recursos relacionados. Para obtener más información, consulte [Acciones, recursos y claves de condición de Amazon Elastic Kubernetes Service](#) y [Uso de roles vinculados a servicios](#) en la Guía del usuario de IAM. Debe completar todos los pasos de esta guía como el mismo usuario.
- Permisos: los usuarios que llamen a la operación de [CreateComputeEnvironment](#) API para crear un entorno de cómputo que utilice los recursos de Amazon EKS necesitan permisos para la operación de `eks:DescribeCluster` API.
- Cuenta de AWS número: necesita saber su Cuenta de AWS ID. Siga las instrucciones que se indican en [Visualización de su ID de Cuenta de AWS](#).
- (Opcional) CloudWatch: para examinar los detalles de [\(opcional\) Enviar un trabajo con anulaciones](#), se debe configurar el registro. Para obtener más información, consulte [Utilice CloudWatch los registros para supervisar AWS Batch los trabajos de Amazon EKS](#).

Paso 1: Cree su clúster de Amazon EKS para AWS Batch

Important

Para comenzar de la manera más sencilla y rápida posible, en este tutorial se incluirán los pasos con la configuración predeterminada. Antes de crear para su uso en producción, recomendamos que conozca toda la configuración e implemente la configuración que satisfaga sus requisitos.

Tras haber instalado los requisitos previos, debe crear su clúster con `eksctl`. Crear el clúster puede tardar entre 10 y 15 minutos.

```
$ eksctl create cluster --name my-cluster-name --region region-code
```

Sustituya en el comando anterior:

- *my-cluster-name* Sustitúyalo por el nombre que desee usar para su clúster.
- *region-code* Sustitúyalo por el Región de AWS para crear el clúster, por ejemplo `us-west-2`.

El nombre y la región del clúster serán necesarios en otro momento para este tutorial.

Paso 2: Prepare el clúster de Amazon EKS para AWS Batch

Todos los pasos son obligatorios.

1. Cree un espacio de nombres dedicado a los trabajos AWS Batch

Se utiliza `kubectl` para crear un nuevo espacio de nombres.

```
$ namespace=my-aws-batch-namespace
```

```
$ cat - <<EOF | kubectl create -f -
{
  "apiVersion": "v1",
  "kind": "Namespace",
  "metadata": {
    "name": "${namespace}",
    "labels": {
      "name": "${namespace}"
    }
  }
}
EOF
```

Salida:

```
namespace/my-aws-batch-namespace created
```

2. Habilite el acceso a través del control de acceso basado en roles (RBAC)

Utilícelo `kubectl` para crear un Kubernetes rol para el clúster que AWS Batch permita vigilar los nodos y los pods y vincular el rol. Debe hacerlo una vez para cada clúster de EKS.

```
$ cat - <<EOF | kubectl apply -f -
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: aws-batch-cluster-role
rules:
  - apiGroups: [""]
```

```

    resources: ["namespaces"]
    verbs: ["get"]
  - apiGroups: [""]
    resources: ["nodes"]
    verbs: ["get", "list", "watch"]
  - apiGroups: [""]
    resources: ["pods"]
    verbs: ["get", "list", "watch"]
  - apiGroups: [""]
    resources: ["events"]
    verbs: ["list"]
  - apiGroups: [""]
    resources: ["configmaps"]
    verbs: ["get", "list", "watch"]
  - apiGroups: ["apps"]
    resources: ["daemonsets", "deployments", "statefulsets", "replicasets"]
    verbs: ["get", "list", "watch"]
  - apiGroups: ["rbac.authorization.k8s.io"]
    resources: ["clusterroles", "clusterrolebindings"]
    verbs: ["get", "list"]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: aws-batch-cluster-role-binding
subjects:
- kind: User
  name: aws-batch
  apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: ClusterRole
  name: aws-batch-cluster-role
  apiGroup: rbac.authorization.k8s.io
EOF

```

Salida:

```

clusterrole.rbac.authorization.k8s.io/aws-batch-cluster-role created
clusterrolebinding.rbac.authorization.k8s.io/aws-batch-cluster-role-binding created

```

3. Cree un Kubernetes rol dentro del espacio de nombres para administrar los pods durante AWS Batch todo el ciclo de vida y vincúlelo. Debe hacerlo una vez para cada espacio de nombres único.

```
$ namespace=my-aws-batch-namespace
```

```
$ cat - <<EOF | kubectl apply -f - --namespace "${namespace}"
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: aws-batch-compute-environment-role
  namespace: ${namespace}
rules:
  - apiGroups: ["" ]
    resources: ["pods"]
    verbs: ["create", "get", "list", "watch", "delete", "patch"]
  - apiGroups: ["" ]
    resources: ["serviceaccounts"]
    verbs: ["get", "list"]
  - apiGroups: ["rbac.authorization.k8s.io"]
    resources: ["roles", "rolebindings"]
    verbs: ["get", "list"]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: aws-batch-compute-environment-role-binding
  namespace: ${namespace}
subjects:
  - kind: User
    name: aws-batch
    apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: Role
  name: aws-batch-compute-environment-role
  apiGroup: rbac.authorization.k8s.io
EOF
```

Salida:

```
role.rbac.authorization.k8s.io/aws-batch-compute-environment-role created
rolebinding.rbac.authorization.k8s.io/aws-batch-compute-environment-role-binding
created
```


4. Actualice el Kubernetes `aws-auth` mapa de configuración para asignar los permisos RBAC anteriores al rol vinculado al servicio. AWS Batch

Reemplace en el siguiente comando:

- Reemplace `<your-account-number>` por el ID de su Cuenta de AWS .

```
$ eksctl create iamidentitymapping \  
  --cluster my-cluster-name \  
  --arn "arn:aws:iam::<your-account-number>:role/AWSServiceRoleForBatch" \  
  --username aws-batch
```

Salida:

```
2022-10-25 20:19:57 [#] adding identity "arn:aws:iam::<your-account-number>:role/  
AWSServiceRoleForBatch" to auth ConfigMap
```

Note

La ruta `aws-service-role/batch.amazonaws.com/` se ha eliminado del ARN del rol vinculado a un servicio. Esto se debe a un problema con el mapa de configuración de `aws-auth`. Para obtener más información, consulte [Roles con rutas que no funcionan cuando la ruta está incluida en su ARN en el aws-authconfigmap](#).

Paso 3: Crear un entorno de computación de Amazon EKS

AWS Batch los entornos informáticos definen los parámetros de los recursos informáticos para satisfacer sus necesidades de carga de trabajo por lotes. En un entorno informático gestionado, le AWS Batch ayuda a gestionar la capacidad y los tipos de instancia de los recursos informáticos (Kubernetesnodos) de su clúster de Amazon EKS. Se basa en la especificación de recursos de computación que se define al crear el entorno de computación. Puede utilizar instancias bajo demanda EC2 o instancias de spot EC2.

Ahora que el rol `AWSServiceRoleForBatch` vinculado al servicio tiene acceso a su clúster de Amazon EKS, puede crear AWS Batch recursos. En primer lugar, cree un entorno de computación que apunte a su clúster de Amazon EKS.

- En subnets ejecute `eksctl get cluster my-cluster-name` para obtener las subredes que el clúster utilizó.
- En el parámetro `securityGroupIds`, puede utilizar el mismo grupo de seguridad que utiliza el clúster de Amazon EKS. Este comando recupera el ID del grupo de seguridad del clúster.

```
$ aws eks describe-cluster \
  --name my-cluster-name \
  --query cluster.resourcesVpcConfig.clusterSecurityGroupId
```

- El `instanceRole` se creará con la formación del clúster. Si desea encontrar la lista `instanceRole` de todas las entidades que utilizan la política `AmazonEKSWorkerNodePolicy`:

```
$ aws iam list-entities-for-policy --policy-arn arn:aws:iam::aws:policy/
AmazonEKSWorkerNodePolicy
```

El nombre del rol de política contiene el nombre del clúster que se creó en `eksctl-my-cluster-name-nodegroup-example`.

Ejecute el siguiente comando para buscar el ARN de `instanceRole`:

```
$ aws iam list-instance-profiles-for-role --role-name eksctl-my-cluster-name-
nodegroup-example
```

Salida:

```
INSTANCEPROFILES      arn:aws:iam::<your-account-number>:instance-profile/
eks-04cb2200-94b9-c297-8dbe-87f12example
```

Para obtener más información, consulte [Crear el rol de IAM de nodo de Amazon EKS](#) y [Habilitar el acceso de entidad principal para el clúster de IAM](#) en la Guía del usuario de Amazon EKS. Si utiliza redes de pod, consulte [Configuración del complemento de CNI de Amazon VPC de Kubernetes para utilizar roles de IAM en las cuentas de servicio](#) en la Guía del usuario de Amazon EKS.

```
$ cat <<EOF > ./batch-eks-compute-environment.json
{
  "computeEnvironmentName": "My-Eks-CE1",
  "type": "MANAGED",
```

```
"state": "ENABLED",
"eksConfiguration": {
  "eksClusterArn": "arn:aws:eks:region-code:your-account-number:cluster/my-cluster-name",
  "kubernetesNamespace": "my-aws-batch-namespace"
},
"computeResources": {
  "type": "EC2",
  "allocationStrategy": "BEST_FIT_PROGRESSIVE",
  "minvCpus": 0,
  "maxvCpus": 128,
  "instanceTypes": [
    "m5"
  ],
  "subnets": [
    "<eks-cluster-subnets-with-access-to-internet-for-image-pull>"
  ],
  "securityGroupIds": [
    "<eks-cluster-sg>"
  ],
  "instanceRole": "<eks-instance-profile>"
}
}
EOF
```

```
$ aws batch create-compute-environment --cli-input-json file:///./batch-eks-compute-environment.json
```

Notas

- El mantenimiento de un entorno de computación de Amazon EKS es una responsabilidad compartida. Para obtener más información, consulte [Responsabilidad compartida de los nodos Kubernetes](#).

Paso 4: Cree una cola de trabajos y adjunte el entorno informático

Important

Es importante confirmar que el entorno de computación está en buen estado antes de continuar. Para ello, se puede utilizar la operación de la [DescribeComputeEnvironments](#) API.

```
$ aws batch describe-compute-environments --compute-environments My-Eks-CE1
```

Confirme que el parámetro `status` no es `INVALID`. Si es así, busque la causa en el parámetro `statusReason`. Para obtener más información, consulte [Solución de problemas AWS Batch](#).

Los trabajos enviados a esta nueva cola de trabajos se ejecutan como pods en los nodos AWS Batch gestionados que se unieron al clúster de Amazon EKS asociado a su entorno de cómputo.

```
$ cat <<EOF > ./batch-eks-job-queue.json
{
  "jobQueueName": "My-Eks-JQ1",
  "priority": 10,
  "computeEnvironmentOrder": [
    {
      "order": 1,
      "computeEnvironment": "My-Eks-CE1"
    }
  ]
}
EOF
```

```
$ aws batch create-job-queue --cli-input-json file:///./batch-eks-job-queue.json
```

Paso 5: Crear una definición de trabajo

La siguiente definición de trabajo le indica al pod que permanezca en espera durante 60 segundos.

```
$ cat <<EOF > ./batch-eks-job-definition.json
{
  "jobDefinitionName": "MyJobOnEks_Sleep",
  "type": "container",
  "eksProperties": {
    "podProperties": {
      "hostNetwork": true,
      "containers": [
        {
          "image": "public.ecr.aws/amazonlinux/amazonlinux:2",

```

```

        "command": [
            "sleep",
            "60"
        ],
        "resources": {
            "limits": {
                "cpu": "1",
                "memory": "1024Mi"
            }
        }
    },
    "metadata": {
        "labels": {
            "environment": "test"
        }
    }
}
EOF

```

```
$ aws batch register-job-definition --cli-input-json file://./batch-eks-job-definition.json
```

Notas

- Hay que tener en cuenta los parámetros cpu y memory. Para obtener más información, consulte [Consideraciones sobre memoria y vCPU para AWS Batch en Amazon EKS](#).

Paso 6: Presentar un trabajo

Ejecute el siguiente AWS CLI comando para enviar un nuevo Job.

```
$ aws batch submit-job --job-queue My-Eks-JQ1 \
    --job-definition MyJobOnEks_Sleep --job-name My-Eks-Job1
```

Cómo comprobar el estado de un trabajo:

```
$ aws batch describe-jobs --job <jobId-from-submit-response>
```

Notas

- Para obtener más información sobre la ejecución de trabajos en recursos de Amazon EKS, consulte [Trabajos de Amazon EKS](#).

Paso 7: Ver la salida del trabajo

Para ver la salida del trabajo, realice lo siguiente:

1. Abra la AWS Batch consola en <https://console.aws.amazon.com/batch/>.
2. En el panel de navegación, seleccione Trabajos.
3. En el menú desplegable de Colas de trabajo, elija la cola de trabajos que creó para el tutorial.
4. La tabla Trabajos muestra todos los trabajos y su estado actual. Una vez que el estado del trabajo sea correcto, elija el nombre del trabajo *My-Eks-JQ1*, para ver los detalles del trabajo.
5. En el panel Detalles, las horas de Iniciado a las y Terminado a las deben estar separadas por un minuto.

Paso 8: (opcional) Enviar un trabajo con anulaciones

Este trabajo anula el comando pasado al contenedor. AWS Batch limpia agresivamente las cápsulas una vez finalizadas las tareas para reducir la carga. Kubernetes Para examinar los detalles de un trabajo, se debe configurar el registro. Para obtener más información, consulte [Utilice CloudWatch los registros para supervisar AWS Batch los trabajos de Amazon EKS](#).

```
$ cat <<EOF > ./submit-job-override.json
{
  "jobName": "EksWithOverrides",
  "jobQueue": "My-Eks-JQ1",
  "jobDefinition": "MyJobOnEks_Sleep",
  "eksPropertiesOverride": {
    "podProperties": {
      "containers": [
        {
          "command": [
            "/bin/sh"
          ],
          "args": [
            "-c",
```

```
        "echo hello world"
      ]
    }
  ]
}
EOF
```

```
$ aws batch submit-job --cli-input-json file:///./submit-job-override.json
```

Notas

- Para mejorar la visibilidad de los detalles de las operaciones, habilite el registro del plano de control de Amazon EKS. Para obtener más información, consulte [Registros del plano de control del clúster de Amazon EKS](#) en la Guía del usuario de Amazon EKS.
- La sobrecarga Daemonsets y kubelets afectan a los recursos de vCPU y de memoria disponibles, específicamente al escalado y la colocación de trabajos. Para obtener más información, consulte [Consideraciones sobre memoria y vCPU para AWS Batch en Amazon EKS](#).

Para ver la salida del trabajo, realice lo siguiente:

1. Abra la AWS Batch consola en <https://console.aws.amazon.com/batch/>
2. En el panel de navegación, seleccione Trabajos.
3. En el menú desplegable de Colas de trabajo, elija la cola de trabajos que creó para el tutorial.
4. La tabla Trabajos muestra todos los trabajos y su estado actual. Una vez que el Estado del trabajo sea Correcto, seleccione el Nombre del trabajo para consultar sus detalles.
5. En el panel Detalles, elija el Nombre del flujo de registro. Se abrirá la CloudWatch consola del Job y debería haber un evento con el Mensaje de hello world o su mensaje personalizado.

Paso 9: Eliminar los recursos del tutorial

Se le cobrará por la instancia de Amazon EC2 mientras esté habilitada. Puede eliminar la instancia para dejar de generar cargos.

Para eliminar los recursos que usted creó, haga lo que se indica a continuación:

1. Abra la AWS Batch consola en <https://console.aws.amazon.com/batch/>.

2. En el panel de navegación, elija Cola de trabajos.
3. En la tabla Cola de trabajos, elija la cola de trabajos que creó para el tutorial.
4. Elija Deshabilitar. Una vez que el Estado de la cola de trabajos esté deshabilitado, puede elegir Eliminar.
5. Una vez se elimine la cola de trabajos, en el panel de navegación, elija Entornos de computación.
6. Elija el entorno de computación que creó para este tutorial y, a continuación, elija Deshabilitar. El entorno de computación puede tardar entre 1 y 2 minutos en terminar de deshabilitarse.
7. Una vez que el Estado del entorno de computación esté deshabilitado, seleccione Eliminar. La eliminación del entorno de computación puede tardar entre 1 y 2 minutos.

Recursos adicionales

Después de completar el tutorial, recomendamos consultar alguno de los siguientes temas:

- Obtenga más información sobre las [prácticas recomendadas](#).
- Explore los componentes AWS Batch principales. Para obtener más información, consulte [Componentes de AWS Batch](#).
- Obtenga más información sobre los diferentes [entornos de computación](#) disponibles en AWS Batch.
- Obtenga más información sobre las [colas de trabajos](#) y sus diferentes opciones de programación.
- Obtenga más información sobre las [definiciones de trabajos](#) y las diferentes opciones de configuración.
- Obtenga más información sobre los distintos tipos de [trabajos](#).

Cómo empezar a utilizar AWS Batch Amazon EKS Private Clusters

AWS Batch es un servicio gestionado que organiza las cargas de trabajo por lotes en los clústeres de Amazon Elastic Kubernetes Service (Amazon EKS). Esto incluye la creación de colas, el seguimiento de dependencias, la gestión de reintentos y las prioridades de los trabajos, la gestión de pods y el escalado de nodos. Esta función conecta su clúster privado de Amazon EKS existente con el AWS Batch fin de ejecutar sus trabajos a escala. Puede utilizar [eksctl](#) (una interfaz de línea de comandos para Amazon EKS), la AWS consola o la [AWS Command Line Interface](#) para crear un clúster privado de Amazon EKS con todos los demás recursos necesarios.

De forma predeterminada, [los clústeres privados de Amazon EKS](#) no tienen acceso a inbound/outbound Internet y solo puede acceder al servidor de API desde su VPC o desde una red conectada. Los puntos de enlace de VPC de Amazon se utilizan para permitir el acceso privado a otros servicios de AWS. `eksctl` admite la creación de clústeres totalmente privados mediante una VPC de Amazon y subredes preexistentes. `eksctl` también crea puntos de enlace de VPC de Amazon en la VPC de Amazon suministrada y modifica las tablas de enrutamiento de las subredes proporcionadas.

Cada subred debe tener asociada una tabla de enrutamiento explícita, ya que `eksctl` no modifica la tabla de enrutamiento principal. El [clúster](#) debe extraer imágenes de un registro de contenedores que esté en su VPC de Amazon. Además, puede crear un Amazon Elastic Container Registry en su VPC de Amazon y copiar las imágenes del contenedor para que sus nodos puedan extraerlas. Para obtener más información, consulte [Copiar una imagen de contenedor de un repositorio en otro repositorio](#). Para empezar a utilizar los repositorios privados de Amazon ECR, consulte [Repositorios privados de Amazon ECR](#).

También puede crear una [regla de caché de extracción](#) con Amazon ECR. Una vez creada una regla de caché de extracción para un registro público externo, extraiga una imagen de ese registro público externo mediante el identificador de recursos uniforme (URI) de su registro privado de Amazon ECR. A continuación, Amazon ECR crea un repositorio y almacena la imagen en la caché. Cuando se extrae una imagen en caché mediante el URI del registro privado de Amazon ECR, Amazon ECR verifica el registro remoto para ver si hay una nueva versión de la imagen y actualiza su registro privado hasta una vez cada 24 horas.

Contenido

- [Descripción general de](#)
- [Requisitos previos](#)
- [Paso 1: Cree su clúster de EKS para AWS Batch](#)
- [Paso 2: Prepare su clúster EKS para AWS Batch](#)
- [Paso 3: Crear un entorno de computación de Amazon EKS](#)
- [Paso 4: Cree una cola de trabajos y adjunte el entorno informático](#)
- [Paso 5: Crear un Amazon ECR con pull through cache](#)
- [Paso 6: Registrar una definición de trabajo](#)
- [Paso 7: Enviar un trabajo para ejecutarlo](#)
- [Paso 8: Ver la salida del trabajo](#)

- [Paso 9: \(opcional\) Enviar un trabajo con anulaciones](#)
- [Paso 10: Eliminar los recursos del tutorial](#)
- [Recursos adicionales](#)
- [Resolución de problemas](#)

Descripción general de

En este tutorial se muestra cómo realizar la configuración AWS Batch con un Amazon EKS privado mediante AWS CloudShell, `kubectl` y `yeksctl`.

Público al que va dirigido

Este tutorial está diseñado para los administradores y desarrolladores de sistemas responsables de la configuración, las pruebas y la implementación de AWS Batch.

Características utilizadas

En este tutorial se muestra cómo utilizar el AWS CLI, para:

- utilizar Amazon Elastic Container Registry (Amazon ECR) para almacenar imágenes de contenedor,
- crear y configurar un entorno de computación de Amazon EKS,
- crear una cola de trabajos,
- crear una definición de trabajo,
- crear y enviar un flujo de trabajo para ejecutarlo,
- enviar un trabajo con anulaciones.

Tiempo requerido

Completar este tutorial debería tardar entre 40 y 50 minutos.

Restricciones regionales

No hay restricciones regionales o de país asociadas al uso de esta solución.

Costos de uso de recursos

La creación de una AWS cuenta no conlleva ningún cargo. Sin embargo, mediante la implementación de esta solución, es posible que incurra en algunos o todos los costos que se enumeran en la siguiente tabla.

Description (Descripción)	Costo (dólares estadounidenses)
El cobro será por hora de uso del clúster	Varía según la instancia. Consulte los precios de Amazon EKS
Instancia de Amazon EC2	Usted paga por cada instancia de Amazon EC2 creada. Para obtener más información acerca de los precios, consulte Precios de Amazon EC2 .

Requisitos previos

Este tutorial AWS CloudShell utiliza un shell preautenticado y basado en un navegador que se inicia directamente desde. Consola de administración de AWS Esto permite acceder al clúster cuando deje de tener acceso público a Internet. El AWS CLI `kubectl`, y `eksctl` puede que ya estén instalados como parte de. AWS CloudShell Para obtener más información al respecto AWS CloudShell, consulte la [Guía AWS CloudShell del usuario](#). Una alternativa AWS CloudShell es conectarse a la VPC del clúster o a una [red conectada](#).

Para ejecutar comandos de `kubectl`, necesitará acceso privado a su clúster de Amazon EKS. Esto significa que todo el tráfico al servidor de la API del clúster debe proceder de dentro de la VPC de su clúster o de una red conectada.

- **AWS CLI**— Una herramienta de línea de comandos para trabajar con AWS servicios, incluido Amazon EKS. Esta guía requiere que utilices la versión 2.8.6 o posterior o la 1.26.0 o posterior. Para obtener más información, consulte [Instalar, actualizar y desinstalar la AWS CLI](#) en la Guía del usuario de AWS Command Line Interface . Tras instalarlo AWS CLI, le recomendamos que también lo configure. Para obtener más información, consulte [Configuración rápida con aws configure](#) en la Guía del usuario de AWS Command Line Interface .
- **`kubectl`**: una herramienta de línea de comandos para trabajar con clústeres de Kubernetes. Esta guía requiere que utilice la versión 1.23 o una posterior. Para obtener más información, consulte [Instalación o actualización de kubectl](#) en la Guía del usuario de Amazon EKS.
- **`eksctl`**: una herramienta de línea de comandos para trabajar con clústeres de Amazon EKS que automatiza varias tareas individuales. Esta guía requiere que utilice la versión 0.115.0 o una posterior. Para obtener más información, consulte [Instalación o actualización de eksctl](#) en la Guía del usuario de Amazon EKS.

- Permisos: los usuarios que llamen a la operación de [CreateComputeEnvironment](#) API para crear un entorno de cómputo que utilice los recursos de Amazon EKS necesitan permisos para la operación de `eks:ListClusters` API `eks:DescribeCluster` y de API. Puede adjuntar la política [AWSBatchFullAccess](#) gestionada a su cuenta de usuario siguiendo las instrucciones para [añadir y eliminar permisos de identidad de IAM](#) que aparecen en la Guía del usuario de IAM.
- InstanceRole— Debe crear una InstanceRole para sus nodos de Amazon EKS que tenga las `AmazonEC2ContainerRegistryPullOnly` políticas `AmazonEKSEWorkerNodePolicy` y. Para obtener instrucciones sobre cómo crear el InstanceRole, consulte [Crear el rol de IAM de nodo de Amazon EKS](#). Necesitará el ARN del InstanceRole.
- Cuenta de AWS ID: necesita saber su Cuenta de AWS ID. Sigue las instrucciones que se indican en Cómo [ver tu Cuenta de AWS identificación](#).
- (Opcional) CloudWatch: para examinar los detalles de [\(opcional\) Enviar un trabajo con anulaciones](#), se debe configurar el registro. Para obtener más información, consulte [Utilice CloudWatch los registros para supervisar AWS Batch los trabajos de Amazon EKS](#).

Paso 1: Cree su clúster de EKS para AWS Batch

Important

Para comenzar de la manera más sencilla y rápida posible, en este tutorial se incluirán los pasos con la configuración predeterminada. Antes de crear para su uso en producción, recomendamos que conozca toda la configuración e implemente la configuración que satisfaga sus requisitos.

Recomendamos utilizar `eksctl` y el siguiente archivo de configuración para crear el clúster. Para configurar el clúster de forma manual, siga las instrucciones en [Implementación de clústeres privados con acceso limitado a Internet](#) de la Guía del usuario de Amazon EKS.

1. Abra la [consola de AWS CloudShell](#) y establezca la región en `us-east-1`. Para el resto del tutorial, asegúrese de estar utilizando `us-east-1`.
2. Cree un clúster de EKS privado en la región `us-east-1` mediante el ejemplo de archivo de configuración `eksctl`. Guarde el archivo `yaml` en su AWS CloudShell entorno y asígnele `clusterConfig.yaml` un nombre. Puede *my-test-cluster* cambiarlo por el nombre que desee usar para su clúster.

```
kind: ClusterConfig
apiVersion: eksctl.io/v1alpha5
metadata:
  name: my-test-cluster
  region: us-east-1
availabilityZones:
  - us-east-1a
  - us-east-1b
  - us-east-1c
managedNodeGroups:
  - name: ng-1
    privateNetworking: true
privateCluster:
  enabled: true
  skipEndpointCreation: false
```

3. Cree los recursos con el siguiente comando: `eksctl create cluster -f clusterConfig.yaml` La creación del cluster puede tardar entre 10 y 15 minutos.
4. Una vez que el clúster haya terminado de crearse, tendrás que añadir tu dirección AWS CloudShell IP a la lista de direcciones permitidas. Para encontrar su dirección AWS CloudShell IP, ejecute el siguiente comando:

```
curl http://checkip.amazonaws.com
```

Luego de conseguir la dirección IP pública, debe crear una regla de lista de permitidos:

```
aws eks update-cluster-config \
  --name my-test-cluster \
  --region us-east-1 \
  --resources-vpc-config
endpointPublicAccess=true,endpointPrivateAccess=true,publicAccessCidrs=["<Public
IP>/32"]
```

A continuación, aplique la actualización al archivo de configuración de kubectl:

```
aws eks update-kubeconfig --name my-test-cluster --region us-east-1
```

5. Ejecute el siguiente comando para comprobar que tiene acceso a los nodos:

```
kubectl get nodes
```

La salida del comando será:

NAME	STATUS	ROLES	AGE	VERSION
ip-192-168-107-235.ec2.internal	Ready	none	1h	v1.32.3-eks-473151a
ip-192-168-165-40.ec2.internal	Ready	none	1h	v1.32.3-eks-473151a
ip-192-168-98-54.ec2.internal	Ready	none	1h	v1.32.1-eks-5d632ec

Paso 2: Prepare su clúster EKS para AWS Batch

Todos los pasos son obligatorios y deben realizarse en el AWS CloudShell.

1. Cree un espacio de nombres dedicado a los trabajos AWS Batch

Se utiliza `kubectl` para crear un nuevo espacio de nombres.

```
$ namespace=my-aws-batch-namespace
```

```
$ cat - <<EOF | kubectl create -f -
{
  "apiVersion": "v1",
  "kind": "Namespace",
  "metadata": {
    "name": "${namespace}",
    "labels": {
      "name": "${namespace}"
    }
  }
}
EOF
```

Salida:

```
namespace/my-aws-batch-namespace created
```

2. Habilite el acceso a través del control de acceso basado en roles (RBAC)

Se utiliza `kubectl` para crear un rol de Kubernetes para el clúster que permita a AWS Batch vigilar los nodos y los pods y vincular el rol. Haga esto una vez para cada clúster de Amazon EKS.

```
$ cat - <<EOF | kubectl apply -f -
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: aws-batch-cluster-role
rules:
  - apiGroups: [""]
    resources: ["namespaces"]
    verbs: ["get"]
  - apiGroups: [""]
    resources: ["nodes"]
    verbs: ["get", "list", "watch"]
  - apiGroups: [""]
    resources: ["pods"]
    verbs: ["get", "list", "watch"]
  - apiGroups: [""]
    resources: ["events"]
    verbs: ["list"]
  - apiGroups: [""]
    resources: ["configmaps"]
    verbs: ["get", "list", "watch"]
  - apiGroups: ["apps"]
    resources: ["daemonsets", "deployments", "statefulsets", "replicasets"]
    verbs: ["get", "list", "watch"]
  - apiGroups: ["rbac.authorization.k8s.io"]
    resources: ["clusterroles", "clusterrolebindings"]
    verbs: ["get", "list"]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: aws-batch-cluster-role-binding
subjects:
  - kind: User
    name: aws-batch
    apiGroup: rbac.authorization.k8s.io
roleRef:
```

```
kind: ClusterRole
name: aws-batch-cluster-role
apiGroup: rbac.authorization.k8s.io
EOF
```

Salida:

```
clusterrole.rbac.authorization.k8s.io/aws-batch-cluster-role created
clusterrolebinding.rbac.authorization.k8s.io/aws-batch-cluster-role-binding created
```

Cree un Kubernetes rol centrado en el espacio de nombres para administrar los pods y vincularlos durante su ciclo AWS Batch de vida. Debe hacerlo una vez para cada espacio de nombres único.

```
$ namespace=my-aws-batch-namespace
```

```
$ cat - <<EOF | kubectl apply -f - --namespace "${namespace}"
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: aws-batch-compute-environment-role
  namespace: ${namespace}
rules:
- apiGroups: ["" ]
  resources: ["pods"]
  verbs: ["create", "get", "list", "watch", "delete", "patch"]
- apiGroups: ["" ]
  resources: ["serviceaccounts"]
  verbs: ["get", "list"]
- apiGroups: ["rbac.authorization.k8s.io"]
  resources: ["roles", "rolebindings"]
  verbs: ["get", "list"]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: aws-batch-compute-environment-role-binding
  namespace: ${namespace}
subjects:
- kind: User
  name: aws-batch
```



```
apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: Role
  name: aws-batch-compute-environment-role
apiGroup: rbac.authorization.k8s.io
EOF
```

Salida:

```
role.rbac.authorization.k8s.io/aws-batch-compute-environment-role created
rolebinding.rbac.authorization.k8s.io/aws-batch-compute-environment-role-binding
created
```

Actualice el Kubernetes aws-auth mapa de configuración para asignar los permisos RBAC anteriores al rol vinculado al servicio. AWS Batch

```
$ eksctl create iamidentitymapping \
  --cluster my-test-cluster \
  --arn "arn:aws:iam::<your-account-ID>:role/AWSServiceRoleForBatch" \
  --username aws-batch
```

Salida:

```
2022-10-25 20:19:57 [#] adding identity "arn:aws:iam::<your-account-ID>:role/
AWSServiceRoleForBatch" to auth ConfigMap
```

Note

La ruta `aws-service-role/batch.amazonaws.com/` se ha eliminado del ARN del rol vinculado a un servicio. Esto se debe a un problema con el mapa de configuración de aws-auth. Para obtener más información, consulte [Roles con rutas que no funcionan cuando la ruta está incluida en su ARN en el aws-authconfigmap](#).

Paso 3: Crear un entorno de computación de Amazon EKS

AWS Batch los entornos informáticos definen los parámetros de los recursos informáticos para satisfacer sus necesidades de carga de trabajo por lotes. En un entorno informático gestionado, le AWS Batch ayuda a gestionar la capacidad y los tipos de instancia de los recursos informáticos

(Kubernetesnodos) de su clúster de Amazon EKS. Se basa en la especificación de recursos de computación que se define al crear el entorno de computación. Puede utilizar instancias bajo demanda EC2 o instancias de spot EC2.

Ahora que el rol `AWSServiceRoleForBatch` vinculado al servicio tiene acceso a su clúster de Amazon EKS, puede crear AWS Batch recursos. En primer lugar, cree un entorno de computación que apunte a su clúster de Amazon EKS.

- En subnets ejecute `eksctl get cluster my-test-cluster` para obtener las subredes que el clúster utilizó.
- En el parámetro `securityGroupIds`, puede utilizar el mismo grupo de seguridad que utiliza el clúster de Amazon EKS. Este comando recupera el ID del grupo de seguridad del clúster.

```
$ aws eks describe-cluster \
  --name my-test-cluster \
  --query cluster.resourcesVpcConfig.clusterSecurityGroupId
```

- Utilice el ARN del `instanceRole` que creó en los requisitos previos.

```
$ cat <<EOF > ./batch-eks-compute-environment.json
{
  "computeEnvironmentName": "My-Eks-CE1",
  "type": "MANAGED",
  "state": "ENABLED",
  "eksConfiguration": {
    "eksClusterArn": "arn:aws:eks:us-east-1:<your-account-ID>:cluster/my-test-cluster",
    "kubernetesNamespace": "my-aws-batch-namespace"
  },
  "computeResources": {
    "type": "EC2",
    "allocationStrategy": "BEST_FIT_PROGRESSIVE",
    "minvCpus": 0,
    "maxvCpus": 128,
    "instanceTypes": [
      "m5"
    ],
    "subnets": [
      "<eks-cluster-subnets-with-access-to-the-image-for-image-pull>"
    ],
    "securityGroupIds": [
      "<eks-cluster-sg>"
    ]
  }
}
```

```
    ],  
    "instanceRole": "<eks-instance-profile>"  
  }  
}  
EOF
```

```
$ aws batch create-compute-environment --cli-input-json file:///./batch-eks-compute-environment.json
```

Notas

- El mantenimiento de un entorno de computación de Amazon EKS es una responsabilidad compartida. Para obtener más información, consulte [Seguridad en Amazon EKS](#).

Paso 4: Cree una cola de trabajos y adjunte el entorno informático

Important

Es importante confirmar que el entorno de computación está en buen estado antes de continuar. Para ello, se puede utilizar la operación de la [DescribeComputeEnvironments](#) API.

```
$ aws batch describe-compute-environments --compute-environments My-Eks-CE1
```

Confirme que el parámetro `status` no es `INVALID`. Si es así, busque la causa en el parámetro `statusReason`. Para obtener más información, consulte [Solución de problemas AWS Batch](#).

Los trabajos enviados a esta nueva cola de trabajos se ejecutan como pods en los nodos AWS Batch gestionados que se unieron al clúster de Amazon EKS asociado a su entorno de cómputo.

```
$ cat <<EOF > ./batch-eks-job-queue.json  
{  
  "jobQueueName": "My-Eks-JQ1",  
  "priority": 10,  
  "computeEnvironmentOrder": [  
    {  
      "order": 1,
```

```

        "computeEnvironment": "My-Eks-CE1"
    }
]
}
EOF

```

```
$ aws batch create-job-queue --cli-input-json file:///./batch-eks-job-queue.json
```

Paso 5: Crear un Amazon ECR con pull through cache

Como el clúster no tiene acceso público a Internet, deberá crear un Amazon ECR para las imágenes del contenedor. Las siguientes instrucciones crean un Amazon ECR con pull-through cache rule (regla de cache de extracción) para almacenar la imagen.

1. El siguiente comando crea la pull-through cache rule. Puede sustituirlo por *tutorial-prefix* un prefijo diferente.

```
aws ecr create-pull-through-cache-rule \
  --ecr-repository-prefix "my-prefix" \
  --upstream-registry-url "public.ecr.aws" \
  --region us-east-1
```

2. Autentíquese con el ECR público.

```
aws ecr get-login-password --region us-east-1 | docker login --username AWS --
password-stdin <your-account-ID>.dkr.ecr.us-east-1.amazonaws.com
```

Ahora puede extraer una imagen.

```
docker pull <your-account-ID>.dkr.ecr.us-east-1.amazonaws.com/my-prefix/
amazonlinux/amazonlinux:2
```

3. Verifique el repositorio y la imagen ejecutando los siguientes comandos:

```
aws ecr describe-repositories
```

```
aws ecr describe-images --repository-name my-prefix/amazonlinux/amazonlinux
```

4. La cadena de imagen que se utilizará para extraer el contenedor tiene el siguiente formato:

```
<your-account-ID>.dkr.ecr.us-east-1.amazonaws.com/my-prefix/amazonlinux/  
amazonlinux:2
```

Paso 6: Registrar una definición de trabajo

La siguiente definición de trabajo le indica al pod que permanezca en espera durante 60 segundos.

En el campo de imagen de la definición del trabajo, en lugar de proporcionar un enlace a la imagen de un repositorio de ECR público, facilite el enlace a la imagen almacenada en nuestro repositorio de ECR privado. Consulte el siguiente ejemplo de definición del trabajo:

```
$ cat <<EOF > ./batch-eks-job-definition.json  
{  
  "jobDefinitionName": "MyJobOnEks_Sleep",  
  "type": "container",  
  "eksProperties": {  
    "podProperties": {  
      "hostNetwork": true,  
      "containers": [  
        {  
          "image": "<your-account-ID>.dkr.ecr.us-east-1.amazonaws.com/my-prefix/  
amazonlinux/amazonlinux:2",  
          "command": [  
            "sleep",  
            "60"  
          ],  
          "resources": {  
            "limits": {  
              "cpu": "1",  
              "memory": "1024Mi"  
            }  
          }  
        }  
      ],  
      "metadata": {  
        "labels": {  
          "environment": "test"  
        }  
      }  
    }  
  }  
}
```

```
}  
EOF
```

```
$ aws batch register-job-definition --cli-input-json file:///./batch-eks-job-  
definition.json
```

Notas

- Hay que tener en cuenta los parámetros cpu y memory. Para obtener más información, consulte [Consideraciones sobre memoria y vCPU para AWS Batch en Amazon EKS](#).

Paso 7: Enviar un trabajo para ejecutarlo

Ejecuta el siguiente AWS CLI comando AWS CloudShell para enviar un nuevo trabajo y devuelve el JobId único.

```
$ aws batch submit-job --job-queue My-Eks-JQ1 \  
--job-definition MyJob0nEks_Sleep - --job-name My-Eks-Job1
```

Notas

- Para obtener más información sobre la ejecución de trabajos en recursos de Amazon EKS, consulte [Trabajos de Amazon EKS](#).

Paso 8: Ver la salida del trabajo

Cómo comprobar el estado de un trabajo:

```
$ aws batch describe-jobs --job <JobID-from-submit-response>
```

Los startedAt y stoppedAt deben estar separados por un minuto.

Paso 9: (opcional) Enviar un trabajo con anulaciones

Este trabajo anula el comando transferido al contenedor.

```
$ cat <<EOF > ./submit-job-override.json
```

```
{
  "jobName": "EksWithOverrides",
  "jobQueue": "My-Eks-JQ1",
  "jobDefinition": "MyJobOnEks_Sleep",
  "eksPropertiesOverride": {
    "podProperties": {
      "containers": [
        {
          "command": [
            "/bin/sh"
          ],
          "args": [
            "-c",
            "echo hello world"
          ]
        }
      ]
    }
  }
}
```

```
$ aws batch submit-job - -cli-input-json file:///./submit-job-override.json
```

Notas

- Para mejorar la visibilidad de los detalles de las operaciones, habilite el registro del plano de control de Amazon EKS. Para obtener más información, consulte [Registros del plano de control del clúster de Amazon EKS](#) en la Guía del usuario de Amazon EKS.
- La sobrecarga Daemonsets y kubelets afectan a los recursos de vCPU y de memoria disponibles, específicamente al escalado y la colocación de trabajos. Para obtener más información, consulte [Consideraciones sobre memoria y vCPU para AWS Batch en Amazon EKS](#).

Paso 10: Eliminar los recursos del tutorial

Se le cobrará por la instancia de Amazon EC2 mientras esté habilitada. Puede eliminar la instancia para dejar de generar cargos.

Para eliminar los recursos que usted creó, haga lo que se indica a continuación:

1. Abre la AWS Batch consola en. <https://console.aws.amazon.com/batch/>
2. En el panel de navegación, elija Cola de trabajos.
3. En la tabla Cola de trabajos, elija la cola de trabajos que creó para el tutorial.
4. En Acciones, seleccione Deshabilitar. Una vez que el Estado de la cola de trabajos esté deshabilitado, puede elegir Eliminar.
5. Una vez se elimine la cola de trabajos, en el panel de navegación, elija Entornos de computación.
6. Elija el entorno de computación que creó para este tutorial y, a continuación, elija Deshabilitar desde Acciones. El entorno de computación puede tardar entre 1 y 2 minutos en terminar de deshabilitarse.
7. Una vez que el Estado del entorno de computación esté deshabilitado, seleccione Eliminar. La eliminación del entorno de computación puede tardar entre 1 y 2 minutos.

Recursos adicionales

Después de completar el tutorial, recomendamos consultar alguno de los siguientes temas:

- Obtenga más información sobre las [prácticas recomendadas](#).
- Explore los componentes AWS Batch principales. Para obtener más información, consulte [Componentes de AWS Batch](#).
- Obtenga más información sobre los diferentes [entornos de computación](#) disponibles en AWS Batch.
- Obtenga más información sobre las [colas de trabajos](#) y sus diferentes opciones de programación.
- Obtenga más información sobre las [definiciones de trabajos](#) y las diferentes opciones de configuración.
- Obtenga más información sobre los distintos tipos de [trabajos](#).

Resolución de problemas

Si los nodos lanzados por AWS Batch no tienen acceso al repositorio de Amazon ECR (o a ningún otro repositorio) que almacene su imagen, sus trabajos podrían permanecer en el estado STARTING. Esto se debe a que el pod no podrá descargar la imagen ni ejecutar su AWS Batch trabajo. Si haces clic en el nombre del pod lanzado por, AWS Batch deberías poder ver el mensaje de error y confirmar el problema. El mensaje de error debería ser similar al siguiente:


```
Failed to pull image "public.ecr.aws/amazonlinux/amazonlinux:2": rpc error: code =
Unknown desc = failed to pull and unpack image
"public.ecr.aws/amazonlinux/amazonlinux:2": failed to resolve reference
"public.ecr.aws/amazonlinux/amazonlinux:2": failed to do request: Head
"https://public.ecr.aws/v2/amazonlinux/amazonlinux/manifests/2": dial tcp: i/o timeout
```

Para ver otros escenarios de solución de problemas comunes, consulte [Solución de problemas de AWS Batch](#). Para solucionar problemas relacionados con el estado del pod, consulte [¿Cómo puedo solucionar los problemas relacionados con el estado de los pods en Amazon EKS?](#)

Primeros pasos con AWS Batch la SageMaker IA

AWS Batch Los trabajos de servicio le permiten enviar trabajos de SageMaker formación a través de colas de AWS Batch trabajos con funciones de programación, priorización y preparación de colas. En este tutorial se muestra cómo configurar y ejecutar un trabajo de SageMaker formación sencillo mediante AWS Batch trabajos de servicio.

Contenido

- [Descripción general de](#)
- [Requisitos previos](#)
- [Paso 1: Crear un rol de ejecución de SageMaker IA](#)
- [Paso 2: Crear el entorno de servicio](#)
- [Paso 3: Crea tu cola de SageMaker trabajos](#)
- [Paso 4: Crear y enviar un trabajo de entrenamiento](#)
- [Paso 5: Supervisar el estado del trabajo](#)
- [Paso 6: Ver la salida del trabajo](#)
- [Paso 7: Eliminar los recursos del tutorial](#)
- [Recursos adicionales](#)

Descripción general de

En este tutorial se muestra cómo configurar los trabajos AWS Batch de servicio para los trabajos de SageMaker formación mediante el AWS CLI.

Público objetivo

Este tutorial está dirigido a científicos de datos y desarrolladores responsables de configurar y ejecutar trabajos de entrenamiento en machine learning a gran escala.

Características utilizadas

En este tutorial se muestra cómo utilizarlas AWS CLI para:

- Crear un entorno de servicios para los trabajos SageMaker de formación
- Cree una cola SageMaker de trabajos de formación
- enviar trabajos de servicio mediante la API `SubmitServiceJob`,
- supervisar el estado del trabajo y ver las salidas,
- Acceda a CloudWatch los registros para los trabajos de formación

Tiempo requerido

Debe tardar aproximadamente 15 minutos en completar este tutorial.

Restricciones regionales

Este tutorial se puede completar en cualquier AWS región en la que estén disponibles AWS Batch tanto la SageMaker IA como la IA.

Costos de uso de recursos

La creación de una AWS cuenta no conlleva ningún cargo. Sin embargo, si se implementa esta solución, es posible que se generen gastos para los siguientes recursos:

Description (Descripción)	Costo (dólares estadounidenses)
SageMaker Instancias de entrenamiento de IA	Usted paga por cada instancia de entrenamiento de SageMaker IA utilizada. Para obtener más información sobre los precios, consulta SageMaker AI Pricing .
Almacenamiento de Amazon S3	Costo mínimo de almacenamiento de las salidas de los trabajos de entrenamiento. Para obtener más información, consulte Precios de Amazon S3 .

Requisitos previos

Antes de comenzar este tutorial, debe instalar y configurar las siguientes herramientas y recursos que necesita para crear y administrar tanto AWS Batch los recursos de SageMaker IA.

- **AWS CLI**— Una herramienta de línea de comandos para trabajar con AWS servicios, incluida AWS Batch la SageMaker IA. Esta guía requiere que utilices la versión 2.8.6 o posterior. Para obtener más información, consulte [Instalar, actualizar y desinstalar la AWS CLI](#) en la Guía del usuario de AWS Command Line Interface . Tras instalar el AWS CLI, le recomendamos que también lo configure. Para obtener más información, consulte [Configuración rápida con aws configure](#) en la Guía del usuario de AWS Command Line Interface .

Paso 1: Crear un rol de ejecución de SageMaker IA

SageMaker La IA utiliza las funciones de ejecución para realizar operaciones en su nombre mediante otros AWS servicios. Debe crear un rol de ejecución y conceder permisos a la SageMaker IA para usar los servicios y recursos necesarios para los trabajos de formación. Utilice la política administrada de AmazonSageMakerFullAccess, ya que incluye permisos para Amazon S3.

Note

Siga las instrucciones siguientes para crear el rol de ejecución de SageMaker IA para este tutorial.

Antes de crear una función de ejecución para su entorno de producción, le recomendamos que consulte la guía [Cómo utilizar las funciones de ejecución de SageMaker IA](#) en la [guía para desarrolladores de SageMaker IA](#).

1. Creación del rol de IAM

Cree un archivo JSON denominado `sagemaker-trust-policy.json` con la siguiente política de confianza:

JSON

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": "iam:CreateRole",  
      "Resource": "*"
    }
  ]
}
```

```
{
  "Effect": "Allow",
  "Principal": {
    "Service": "sagemaker.amazonaws.com"
  },
  "Action": "sts:AssumeRole"
}
```

Cree un rol de IAM mediante la política de confianza:

```
aws iam create-role \
  --role-name SageMakerExecutionRole \
  --assume-role-policy-document file:///sagemaker-trust-policy.json \
  --description "Execution role for SageMaker training jobs"
```

2. Asocie las políticas administradas

Asocie las políticas administradas necesarias al rol:

```
aws iam attach-role-policy \
  --role-name SageMakerExecutionRole \
  --policy-arn arn:aws:iam::aws:policy/AmazonSageMakerFullAccess
```

```
aws iam attach-role-policy \
  --role-name SageMakerExecutionRole \
  --policy-arn arn:aws:iam::aws:policy/AmazonS3FullAccess
```

3. Observe el ARN del rol

Obtenga el rol ARN que necesitará en los siguientes pasos:

```
aws iam get-role --role-name SageMakerExecutionRole --query 'Role.Arn' --output text
```

Guarde este ARN, ya que lo usará al crear la carga útil de su trabajo de entrenamiento.

Paso 2: Crear el entorno de servicio

Un entorno de servicios define las limitaciones de capacidad para los trabajos de SageMaker formación. El entorno de servicio encapsula la cantidad máxima de instancias de entrenamiento que se pueden ejecutar simultáneamente.

Important

Cuando crea su primer entorno de servicios para la SageMaker formación, crea AWS Batch automáticamente un rol vinculado al servicio denominado `AWSServiceRoleForAWSBatchWithSagemaker` en su cuenta. Esta función le permite AWS Batch poner en cola los trabajos de SageMaker formación y gestionarlos en tu nombre. Para obtener más información sobre los roles vinculados a servicios y sus permisos, consulte [the section called “SageMaker rol de integración”](#).

Cree un entorno de servicios que pueda gestionar hasta 5 instancias:

```
aws batch create-service-environment \
  --service-environment-name TutorialServiceEnvironment \
  --service-environment-type SAGEMAKER_TRAINING \
  --capacity-limits capacityUnit=NUM_INSTANCES,maxCapacity=5
```

Salida:

```
{
  "serviceEnvironmentName": "TutorialServiceEnvironment",
  "serviceEnvironmentArn": "arn:aws:batch:your-region:your-account-id:service-
environment/TutorialServiceEnvironment"
}
```

Compruebe que el entorno de servicio se haya creado correctamente:

```
aws batch describe-service-environments --service-
environments TutorialServiceEnvironment
```

Salida:

```
{
```

```

    "serviceEnvironments": [
      {
        "serviceEnvironmentName": "TutorialServiceEnvironment",
        "serviceEnvironmentArn": "arn:aws:batch:your-region:your-account-id:service-environment/TutorialServiceEnvironment",
        "serviceEnvironmentType": "SAGEMAKER_TRAINING",
        "state": "ENABLED",
        "status": "VALID",
        "capacityLimits": [
          {
            "maxCapacity": 5,
            "capacityUnit": "NUM_INSTANCES"
          }
        ],
        "tags": {}
      }
    ]
  }
}

```

Para obtener más información sobre los entornos de servicios, consulte [Entornos de servicio](#).

Paso 3: Crea tu cola de SageMaker trabajos

Una cola de SageMaker trabajos gestiona la programación y la ejecución de los trabajos de servicio. Los trabajos enviados a esta cola se entregarán a su entorno de servicio según la disponibilidad de la capacidad.

Cree una cola SageMaker de trabajos de formación:

```

aws batch create-job-queue \
  --job-queue-name my-sm-training-fifo-jq \
  --job-queue-type SAGEMAKER_TRAINING \
  --priority 1 \
  --service-environment-order order=1,serviceEnvironment=TutorialServiceEnvironment

```

Salida:

```

{
  "jobQueueName": "my-sm-training-fifo-jq",
  "jobQueueArn": "arn:aws:batch:your-region:your-account-id:job-queue/my-sm-training-fifo-jq"
}

```

Compruebe que la cola de trabajos se haya creado correctamente:

```
aws batch describe-job-queues --job-queues my-sm-training-fifo-jq
```

Salida:

```
{
  "jobQueues": [
    {
      "jobQueueName": "my-sm-training-fifo-jq",
      "jobQueueArn": "arn:aws:batch:your-region:your-account-id:job-queue/my-sm-training-fifo-jq",
      "state": "ENABLED",
      "status": "VALID",
      "statusReason": "JobQueue Healthy",
      "priority": 1,
      "computeEnvironmentOrder": [],
      "serviceEnvironmentOrder": [
        {
          "order": 1,
          "serviceEnvironment": "arn:aws:batch:your-region:your-account-id:service-environment/TutorialServiceEnvironment"
        }
      ],
      "jobQueueType": "SAGEMAKER_TRAINING",
      "tags": {}
    }
  ]
}
```

Para obtener más información sobre las colas de SageMaker trabajos, consulte. [the section called “Creación de una cola de trabajos de SageMaker”](#)

Paso 4: Crear y enviar un trabajo de entrenamiento

Ahora creará un trabajo de entrenamiento sencillo y lo enviará a la lista de trabajos. En este ejemplo se utiliza un trabajo de entrenamiento básico de “hola, mundo” que demuestra la funcionalidad del trabajo de servicio.

Cree un archivo llamado *my_training_job.json* con el siguiente contenido. *your-account-id* Sustitúyalo por tu ID AWS de cuenta:

Note

S3OutputPath es necesario para crear el trabajo de SageMaker formación, pero los resultados de este tutorial no se almacenan en el bucket de Amazon S3 y puede utilizar la ruta en el siguiente JSON. En su entorno de producción, necesitará un bucket de Amazon S3 válido para almacenar allí la salida, si así lo desea.

```
{
  "TrainingJobName": "my-simple-training-job",
  "RoleArn": "arn:aws:iam::your-account-id:role/SageMakerExecutionRole",
  "AlgorithmSpecification": {
    "TrainingInputMode": "File",
    "TrainingImage": "763104351884.dkr.ecr.us-west-2.amazonaws.com/pytorch-
training:2.0.0-cpu-py310",
    "ContainerEntrypoint": [
      "echo",
      "hello world"
    ]
  },
  "ResourceConfig": {
    "InstanceType": "ml.c5.xlarge",
    "InstanceCount": 1,
    "VolumeSizeInGB": 1
  },
  "OutputDataConfig": {
    "S3OutputPath": "s3://your-s3-bucket/output"
  },
  "StoppingCondition": {
    "MaxRuntimeInSeconds": 30
  }
}
```

Envíe el trabajo de formación mediante la [SubmitServiceJob](#) API:

```
aws batch submit-service-job \
  --job-queue my-sm-training-fifo-jq \
  --job-name my-batch-sm-job \
  --service-job-type SAGEMAKER_TRAINING \
  --retry-strategy attempts=1 \
  --timeout-config attemptDurationSeconds=60 \
```



```
--service-request-payload file://my_training_job.json
```

Salida:

```
{
  "jobArn": "arn:aws:batch:your-region:your-account-id:service-job/your-job-id",
  "jobName": "my-batch-sm-job",
  "jobId": "your-job-id"
}
```

Para obtener más información sobre la carga útil del trabajo de servicio, consulte [the section called “Cargas útiles de trabajos de servicio”](#). Y para obtener más información sobre el envío de trabajos de servicio, consulte [the section called “Envío de un trabajo de servicio”](#).

Paso 5: Supervisar el estado del trabajo

Puede supervisar sus trabajos de formación mediante lo siguiente AWS Batch APIs: [DescribeServiceJobListServiceJobs](#), y [GetJobQueueSnapshot](#). En esta sección se muestran diferentes formas de comprobar el estado de los trabajos y la información de las colas.

Cómo ver los trabajos en ejecución en la cola:

```
aws batch list-service-jobs \
  --job-queue my-sm-training-fifo-jq --job-status RUNNING
```

Salida:

```
{
  "jobSummaryList": [
    {
      "latestAttempt": {
        "serviceResourceId": {
          "name": "TrainingJobArn",
          "value": "arn:aws:sagemaker:your-region:your-account-id:training-job/AWSBatch<my-simple-training-job><your-attempt-id>"
        }
      },
      "createdAt": 1753718760,
      "jobArn": "arn:aws:batch:your-region:your-account-id:service-job/your-job-id",
      "jobId": "your-job-id",
      "jobName": "my-batch-sm-job",
    }
  ]
}
```

```

        "serviceJobType": "SAGEMAKER_TRAINING",
        "status": "RUNNING",
        "startedAt": 1753718820
    }
]
}

```

Cómo ver los trabajos que se encuentran en el estado RUNNABLE:

```

aws batch list-service-jobs \
  --job-queue my-sm-training-fifo-jq --job-status RUNNABLE

```

Cómo obtener una instantánea de los próximos trabajos en la cola:

```

aws batch get-job-queue-snapshot --job-queue my-sm-training-fifo-jq

```

Salida:

```

{
  "frontOfQueue": {
    "jobs": [
      {
        "jobArn": "arn:aws:batch:your-region:your-account-id:service-job/your-job-id",
        "earliestTimeAtPosition": 1753718880
      },
      {
        "jobArn": "arn:aws:batch:your-region:your-account-id:service-job/your-job-id-2",
        "earliestTimeAtPosition": 1753718940
      }
    ],
    "lastUpdatedAt": 1753718970
  }
}

```

Cómo buscar trabajos por nombre:

```

aws batch list-service-jobs \
  --job-queue my-sm-training-fifo-jq \
  --filters name=JOB_NAME,values="my-batch-sm-job"

```

Salida:

```
{
  "jobSummaryList": [
    {
      "latestAttempt": {
        "serviceResourceId": {
          "name": "TrainingJobArn",
          "value": "arn:aws:sagemaker:your-region:your-account-id:training-
job/AWSBatch<my-simple-training-job><your-attempt-id>"
        }
      },
      "createdAt": 1753718760,
      "jobArn": "arn:aws:batch:your-region:your-account-id:service-job/your-job-
id",
      "jobId": "your-job-id",
      "jobName": "my-batch-sm-job",
      "serviceJobType": "SAGEMAKER_TRAINING",
      "status": "RUNNING"
    }
  ]
}
```

Para obtener más información sobre la asignación del estado de trabajos, consulte [the section called “Estado del trabajo de servicio”](#).

Paso 6: Ver la salida del trabajo

Una vez que termine su trabajo, podrá ver su resultado y sus registros a través de la SageMaker IA APIs. AWS Batch

Obtenga información detallada sobre su trabajo en AWS Batch:

```
aws batch describe-service-job \
  --job-id your-job-id
```

Salida:

```
{
  "attempts": [
    {
      "serviceResourceId": {
```

```

        "name": "TrainingJobArn",
        "value": "arn:aws:sagemaker:your-region:your-account-id:training-job/
AWSBatch<my-simple-training-job><your-attempt-id>"
    },
    "startedAt": 1753718820,
    "stoppedAt": 1753718880,
    "statusReason": "Received status from SageMaker: Training job completed"
  }
],
"createdAt": 1753718760,
"jobArn": "arn:aws:batch:your-region:your-account-id:service-job/your-job-id",
"jobId": "your-job-id",
"jobName": "my-batch-sm-job",
"jobQueue": "arn:aws:batch:your-region:your-account-id:job-queue/my-sm-training-
fifo-jq",
"latestAttempt": {
  "serviceResourceId": {
    "name": "TrainingJobArn",
    "value": "arn:aws:sagemaker:your-region:your-account-id:training-job/
AWSBatch<my-simple-training-job><your-attempt-id>"
  }
},
"retryStrategy": {
  "attempts": 1,
  "evaluateOnExit": []
},
"serviceRequestPayload": "your-training-job-request-json",
"serviceJobType": "SAGEMAKER_TRAINING",
"startedAt": 1753718820,
"status": "SUCCEEDED",
"statusReason": "Received status from SageMaker: Training job completed",
"stoppedAt": 1753718880,
"tags": {},
"timeoutConfig": {
  "attemptDurationSeconds": 60
}
}

```

Este comando devuelve información completa sobre el trabajo, incluido el SageMaker ARN del trabajo de formación, que puede utilizar para acceder al trabajo directamente a través SageMaker de la IA:

```
aws sagemaker describe-training-job \
```

```
--training-job-name AWSBatch<my-simple-training-job><your-attempt-id>
```

Para ver los CloudWatch registros de tu trabajo de formación, primero obtén el nombre del flujo de registro:

```
aws logs describe-log-streams \  
  --log-group-name /aws/sagemaker/TrainingJobs \  
  --log-stream-name-prefix AWSBatchmy-simple-training-job
```

Salida:

```
{  
  "logStreams": [  
    {  
      "logStreamName": "your-log-stream-name",  
      "creationTime": 1753718830,  
      "firstEventTimestamp": 1753718840,  
      "lastEventTimestamp": 1753718850,  
      "lastIngestionTime": 1753718860,  
      "uploadSequenceToken": upload-sequence-token,  
      "arn": "arn:aws:logs:your-region:your-account-id:log-group:/aws/sagemaker/  
TrainingJobs:log-stream:AWSBatch<my-simple-training-job><your-attempt-id>/algo-1-algo-  
id",  
      "storedBytes": 0  
    }  
  ]  
}
```

A continuación, recupere los registros con el nombre del flujo de registro de la respuesta anterior:

```
aws logs get-log-events \  
  --log-group-name /aws/sagemaker/TrainingJobs \  
  --log-stream-name your-log-stream-name
```

Salida:

```
{  
  "events": [  
    {  
      "timestamp": 1753718845,  
      "message": "hello world",
```

```
        "ingestionTime": 1753718865
      }
    ],
    "nextForwardToken": "next-forward-token",
    "nextBackwardToken": "next-backward-token"
  }
```

La salida del registro muestra el mensaje “hola, mundo” de su trabajo de entrenamiento, que confirma que el trabajo se ha ejecutado correctamente.

Paso 7: Eliminar los recursos del tutorial

Cuando haya completado este tutorial, elimine los recursos que creó para evitar la generación de gastos innecesarios.

En primer lugar, deshabilite y elimine la cola de trabajos:

```
aws batch update-job-queue \  
  --job-queue my-sm-training-fifo-jq \  
  --state DISABLED
```

Espere a que se deshabilite la cola de trabajos y, a continuación, elimínela:

```
aws batch delete-job-queue \  
  --job-queue my-sm-training-fifo-jq
```

Deshabilite y elimine el entorno de servicio:

```
aws batch update-service-environment \  
  --service-environment TutorialServiceEnvironment \  
  --state DISABLED
```

Espere a que se deshabilite el entorno de servicio y, finalmente, elimínelo:

```
aws batch delete-service-environment \  
  --service-environment TutorialServiceEnvironment
```

Recursos adicionales

Después de completar el tutorial, recomendamos consultar alguno de los siguientes temas:

- Recomendamos usar PySDK para crear trabajos de servicio y enviarlos a la cola de trabajos, ya que contiene clases y utilidades auxiliares. Para ver un ejemplo del uso de PySDK, consulta los [ejemplos de SageMaker IA](#) en GitHub.
- Obtener más información sobre [the section called “Trabajos de servicio”](#).
- Consulte [Cargas útiles de trabajos de servicio en AWS Batch](#) para las configuraciones del trabajo de entrenamiento más complejos.
- Obtenga información sobre [Envíe un trabajo de servicio en AWS Batch](#) y la API `SubmitServiceJob`.
- Revise [Asignar el estado del trabajo del AWS Batch servicio al estado de la SageMaker IA](#) para comprender las transiciones entre los estados de trabajo.
- Visite la [documentación del SDK de Python para SageMaker IA](#) para obtener más formas ricas en funciones de crear y enviar trabajos de SageMaker formación con Python.
- Consulta [SageMakerejemplos de cuadernos para flujos](#) de trabajo de aprendizaje automático más complejos.

El panel del widget de AWS Batch

En el panel de AWS Batch, puede supervisar los trabajos recientes, las colas de trabajos y los entornos de computación. De forma predeterminada, se muestran los siguientes widgets del panel:

- Descripción general de trabajos: para obtener más información sobre trabajos de AWS Batch, consulte [Tareas](#).
- Descripción general de las colas de trabajos: para obtener más información sobre las colas de trabajos de AWS Batch, consulte [Colas de trabajos](#).
- Descripción general del entorno de computación: para obtener más información sobre los entornos de computación de AWS Batch, consulte [Entornos informáticos para AWS Batch](#).

Puede personalizar los widgets que se muestran en la página del panel de control. En las siguientes secciones se describen los widgets adicionales que puede agregar.

Temas

- [Cómo agregar el widget de cola de trabajos única al panel de AWS Batch](#)
- [Cómo agregar el widget de CloudWatch Container Insights al panel de AWS Batch](#)
- [Cómo agregar el widget de registros de trabajos al panel de AWS Batch](#).

Cómo agregar el widget de cola de trabajos única al panel de AWS Batch

El widget de cola de trabajos única muestra información detallada sobre una única cola de trabajos.

Para añadir este widget, siga estos pasos.

1. Abra la [consola de AWS Batch](#).
2. En la barra de navegación, seleccione la Región de AWS que desea utilizar.
3. En el panel de navegación, elija Panel.
4. Elija Añadir widget.
5. En Cola de trabajos única, elija Añadir widget.
6. En Cola de trabajos, elija la cola de trabajos que desee.
7. En Estado del trabajo, elija los estados del trabajo que desee mostrar.

8. (Opcional) Desactive Mostrar entornos de computación conectados si no desea mostrar las propiedades de los entornos de computación.
9. En Propiedades del entorno de computación, elija las propiedades que desee.
10. Elija Añadir.

Cómo agregar el widget de CloudWatch Container Insights al panel de AWS Batch

Este widget muestra métricas agregadas para trabajos y entornos de computación de AWS Batch. Para obtener más información sobre los contenedores, consulte [the section called “CloudWatch Container Insights”](#).

Para añadir este widget, siga estos pasos.

1. Abra la [consola de AWS Batch](#).
2. En la barra de navegación, seleccione la Región de AWS que desea utilizar.
3. En el panel de navegación, elija Panel.
4. Elija Añadir widget.
5. En Información de contenedores, elija Añadir widget.
6. En Entorno de computación, elija el entorno de computación que desee.
7. Elija Añadir.

Cómo agregar el widget de registros de trabajos al panel de AWS Batch.

Este widget muestra distintos registros desde los trabajos en una ubicación práctica. Para obtener más información acerca de los registros de trabajo, consulte [the section called “Vea los registros de trabajos en CloudWatch los registros”](#).

Para añadir este widget, siga estos pasos.

1. Abra la [consola de AWS Batch](#).
2. En la barra de navegación, seleccione la Región de AWS que desea utilizar.
3. En el panel de navegación, elija Panel.

4. Elija Añadir widget.
5. Para Registros de trabajos, elija Añadir widget.
6. En ID de trabajo, introduzca el ID de trabajo del trabajo que desee.
7. Elija Añadir.

Entornos informáticos para AWS Batch

Las colas de trabajos se asignan a uno o varios entornos de computación. Los entornos de computación incluyen instancias de contenedor de Amazon ECS que se utilizan para ejecutar trabajos por lotes en contenedores. Un entorno de computación específico también se puede asignar a una o más de una cola de trabajos. Dentro de una cola de trabajos, los entornos de computación asociados tienen cada uno un orden que utiliza el programador para determinar dónde se ejecutarán los trabajos que están listos para ejecutarse. Si el entorno de computación tiene un estado de `VALID` y tiene recursos gratuitos, el trabajo se programa para una instancia de contenedor dentro de dicho entorno de computación. Si el entorno de computación tiene un estado de `INVALID` o no puede proporcionar un recurso de computación apropiado, el programador intenta ejecutar el trabajo en el siguiente entorno de computación.

Temas

- [Entornos de computación administrados](#)
- [Entornos de computación no administrados](#)
- [Creación de un entorno de computación](#)
- [Actualizar un entorno informático en AWS Batch](#)
- [AMI de recursos de computación](#)
- [Utilice las plantillas de lanzamiento de Amazon EC2 con AWS Batch](#)
- [Configuración del servicio de metadatos de instancias \(IMDS\)](#)
- [Configuraciones de EC2](#)
- [Estrategias de asignación de tipos de instancia para AWS Batch](#)
- [Administración de la memoria de recurso de computación](#)
- [Entornos de computación en Fargate](#)
- [Entornos de computación de Amazon EKS](#)

Entornos de computación administrados

Puede utilizar un entorno informático gestionado para AWS Batch gestionar la capacidad y los tipos de instancias de los recursos informáticos del entorno. Esto se basa en la especificación del recurso de cálculo que usted defina al crear el entorno de computación. Puede optar por utilizar Instancias bajo demanda de Amazon EC2 o Instancias de spot de Amazon EC2. O bien, también puede utilizar

la capacidad de Fargate y Fargate Spot en su entorno de computación administrado. Si utiliza instancias de spot, también puede establecer un precio máximo. De este modo, las instancias de spot solo se lanzan cuando el precio de oferta de spot está por debajo de un determinado porcentaje del precio bajo demanda.

 Important

Las instancias Fargate Spot no son compatibles con. Windows containers on AWS Fargate. Se bloqueará una cola de trabajos si se envía un FargateWindows trabajo a una cola de trabajos que solo utilice entornos de cómputo Fargate Spot.

 Important

AWS Batch crea y administra varios AWS recursos en su nombre y dentro de su cuenta, incluidas las plantillas de lanzamiento de Amazon EC2, los grupos de Auto Scaling de Amazon EC2, las flotas puntuales de Amazon EC2 y los clústeres de Amazon ECS. Estos recursos administrados están configurados específicamente para garantizar un funcionamiento óptimo de AWS Batch. La modificación manual de estos recursos AWS Batch gestionados, a menos que se indique explícitamente en la AWS Batch documentación, puede provocar un comportamiento inesperado, como los entornos INVALID informáticos, un comportamiento de escalado de instancias subóptimo, retrasos en el procesamiento de la carga de trabajo o costes inesperados. El servicio no puede respaldar estas modificaciones manuales de forma determinista. AWS Batch Utilice siempre la consola AWS Batch APIs o la AWS Batch consola compatibles para gestionar sus entornos informáticos.

Las modificaciones manuales no admitidas incluyen ejecutar sus propias tareas o servicios de Amazon ECS en AWS Batch clústeres de Amazon ECS administrados o iniciar procesos, daemons o servicios adicionales directamente en instancias administradas. AWS Batch AWS Batch asume el control total de los recursos informáticos en un entorno informático gestionado y puede finalizar instancias, detener tareas o escalar el clúster en cualquier momento. Cualquier carga de trabajo que ejecute fuera de la presentación de AWS Batch trabajos en estos recursos gestionados puede interrumpirse sin previo aviso. La ejecución de AWS Batch cargas de trabajo no AWS Batch gestionadas en clústeres e instancias gestionados también puede interferir con la programación de los AWS Batch trabajos y el escalado de las instancias.

Los entornos de computación gestionados lanzan instancias de Amazon EC2 en la VPC y las subredes que especifique y, a continuación, las registran en un clúster de Amazon ECS. Las instancias de Amazon EC2 necesitan acceso de red externo para comunicarse con el punto de conexión de servicio de Amazon ECS. Algunas subredes no proporcionan direcciones IP públicas a las instancias de Amazon EC2. Si las instancias de Amazon EC2 no tienen una dirección IP pública, deberán utilizar traducción de direcciones de red (NAT) para obtener este acceso. Para obtener información, consulte [Gateways NAT](#) en la Guía del usuario de Amazon VPC. Para obtener más información acerca de cómo crear una VPC, consulte [Creación de una nube virtual privada](#).

De forma predeterminada, los entornos informáticos AWS Batch gestionados utilizan una versión reciente y aprobada de la AMI optimizada de Amazon ECS para los recursos informáticos. Sin embargo, es posible que desee crear sus propias AMI para utilizarlas en sus entornos de computación administrados por varias razones. Para obtener más información, consulte [AMI de recursos de computación](#).

Note

AWS Batch no se actualiza automáticamente AMIs en un entorno informático una vez creado. Por ejemplo, no actualiza el AMIs entorno informático cuando se publica una versión más reciente de la AMI optimizada para Amazon ECS. Usted es responsable de la administración del sistema operativo invitado. Esto incluye actualizaciones y parches de seguridad. También es el responsable de cualquier otra utilidad o software de aplicaciones que se instale en los recursos de computación. Hay dos maneras de utilizar una AMI nueva para sus AWS Batch trabajos. El método original consiste en completar estos pasos:

1. Cree un nuevo entorno de computación con la nueva AMI.
2. Añada el entorno de computación a una cola de trabajos existente.
3. Quite el entorno de computación anterior de la cola de trabajos.
4. Elimine el entorno de computación anterior.

En abril de 2022, AWS Batch se agregó un soporte mejorado para actualizar los entornos de cómputo. Para obtener más información, consulte [Actualizar un entorno informático en AWS Batch](#). Para utilizar la actualización mejorada de los entornos de cómputo para actualizar AMIs, siga estas reglas:

- No establezca el parámetro service role ([serviceRole](#)) o establézcalo en el rol `AWSServiceRoleForBatch` vinculado al servicio.

- Defina el parámetro de estrategia de asignación ([allocationStrategy](#)) en `BEST_FIT_PROGRESSIVE`, `SPOT_CAPACITY_OPTIMIZED` o `SPOT_PRICE_CAPACITY_OPTIMIZED`.
- Defina el parámetro de actualización a la última versión de la imagen ([updateToLatestImageVersion](#)) en `true`.
- No especifique un ID de AMI en [imageId](#), [imageIdOverride](#) (en [ec2Configuration](#)) o en la plantilla de lanzamiento ([launchTemplate](#)). En ese caso, AWS Batch selecciona la última AMI optimizada para Amazon ECS compatible AWS Batch en el momento en que se inicia la actualización de la infraestructura. Como alternativa, puede especificar el ID de la AMI en los parámetros `imageId` o `imageIdOverride`, o la plantilla de lanzamiento identificada por las propiedades `LaunchTemplate`. El cambio de cualquiera de estas propiedades inicia una actualización de la infraestructura. Si el ID de AMI se especifica en la plantilla de lanzamiento, no se puede reemplazar especificando un ID de AMI en los parámetros `imageId` o `imageIdOverride`. Solo se puede reemplazar especificando una plantilla de lanzamiento diferente. O bien, si la versión de la plantilla de lanzamiento está configurada en `$Default` o `$Latest`, configurando una nueva versión predeterminada para la plantilla de lanzamiento (si es `$Default`) o añadiendo una nueva versión a la plantilla de lanzamiento (si es `$Latest`).

Si se siguen estas reglas, cualquier actualización que inicie una actualización de la infraestructura hará que se vuelva a seleccionar el ID de la AMI. Si la configuración [version](#) en la plantilla de lanzamiento ([launchTemplate](#)) toma el valor `$Latest` o `$Default`, se evaluará la versión más reciente o predeterminada de la plantilla de lanzamiento en el momento de la actualización de la infraestructura, incluso si [launchTemplate](#) no se ha actualizado.

Consideraciones a la hora de crear trabajos paralelos de varios nodos

AWS Batch recomienda crear entornos de cómputo dedicados para ejecutar trabajos paralelos (MNP) con varios nodos y trabajos que no sean MNP. Esto se debe a la forma en que se crea la capacidad informática en su entorno de computación administrado. Al crear un nuevo entorno informático gestionado, si especificas un `minvCpu` valor superior a cero, AWS Batch crea un grupo de instancias para usarlo únicamente con trabajos que no son MNP. Si se envía un trabajo paralelo de varios nodos, AWS Batch crea una nueva capacidad de instancia para ejecutar los trabajos paralelos de varios nodos. En los casos en los que haya trabajos paralelos de un solo nodo y de

varios nodos ejecutándose en el mismo entorno informático en el que se establezca un `maxvCpus` valor `minvCpus` o, si los recursos informáticos necesarios no están disponibles, AWS Batch esperará a que finalicen los trabajos actuales antes de crear los recursos informáticos necesarios para ejecutar los nuevos trabajos.

Entornos de computación no administrados

En un entorno informático no administrado, usted administra sus propios recursos informáticos. AWS Batch admite entornos informáticos no gestionados tanto para Amazon ECS como para Amazon EKS, lo que le permite mantener el control de su infraestructura y, al mismo tiempo, aprovechar las capacidades de programación de trabajos de Batch.

Note

AWS Los recursos de Fargate no son compatibles con los entornos informáticos no gestionados.

Entornos de cómputo Amazon ECS no gestionados

En el caso de los entornos informáticos de Amazon ECS no administrados, debe comprobar que la AMI que utiliza para sus recursos informáticos cumple con la especificación de la AMI de la instancia de contenedor de Amazon ECS. Para obtener más información, consulte [Especificaciones de AMI de recursos de computación](#) y [Tutorial: creación de una AMI de recursos de computación](#).

Después de crear el entorno informático no administrado, utilice la operación de la [DescribeComputeEnvironments](#) API para ver los detalles del entorno informático. Encuentre el clúster de Amazon ECS asociado al entorno y, a continuación, lance manualmente las instancias de contenedor en ese clúster de Amazon ECS.

El siguiente AWS CLI comando también proporciona el ARN del clúster Amazon ECS.

```
$ aws batch describe-compute-environments \
  --compute-environments unmanagedCE \
  --query "computeEnvironments[].ecsClusterArn"
```

Para obtener más información, consulte [Lanzamiento de una instancia de contenedor de Amazon ECS](#) en la Guía del desarrollador de Amazon Elastic Container Service. Al lanzar los recursos de

computación, especifique el ARN del clúster de Amazon ECS que los recursos deben registrar con los siguientes datos de usuario de Amazon EC2. *ecsClusterArn* Sustitúyalo por el ARN del clúster que obtuvo con el comando anterior.

```
#!/bin/bash
echo "ECS_CLUSTER=ecsClusterArn" >> /etc/ecs/ecs.config
```

Entornos de cómputo Amazon EKS no gestionados

En un entorno informático Amazon EKS no gestionado, usted administra sus propios nodos de Kubernetes y, al mismo tiempo, AWS Batch gestiona la programación y la colocación de los trabajos. Le permite controlar directamente su infraestructura de Kubernetes para cumplir con los requisitos operativos, de seguridad o de conformidad. Usted es responsable de aprovisionar y configurar sus nodos de Amazon EKS y, al mismo tiempo, de AWS Batch integrarlos con su clúster de Amazon EKS existente para programar y ejecutar los trabajos.

Para obtener más información, consulte el [tutorial: Creación de un entorno informático no gestionado con los recursos de Amazon EKS](#).

Creación de un entorno de computación

Antes de poder ejecutar trabajos AWS Batch, debe crear un entorno informático. Puede crear un entorno informático gestionado en el que AWS Batch administre las instancias de Amazon EC2 o los recursos de AWS Fargate dentro del entorno en función de sus especificaciones. O bien, puede crear un entorno de computación no administrado en el que gestione la configuración de la instancia de Amazon EC2 dentro del entorno.

Important

Las instancias de spot de Fargate no se admiten en los siguientes escenarios:

- Windows containers on AWS Fargate

En estas situaciones, se bloqueará una cola de trabajos si un trabajo se envía a una cola de trabajos que solo utiliza entornos de computación Fargate Spot.

Temas

- [Tutorial: creación de un entorno de computación gestionado con los recursos de Fargate](#)
- [Tutorial: creación de un entorno de computación gestionado con los recursos de Amazon EC2](#)
- [Tutorial: creación de un entorno de computación no gestionado con los recursos de Amazon EC2](#)
- [Tutorial: creación de un entorno de computación gestionado con los recursos de Amazon EKS](#)
- [Tutorial: Creación de un entorno informático no gestionado con los recursos de Amazon EKS](#)
- [Recurso: plantilla de entorno de computación](#)
- [Tabla de computación para el tipo de instancia](#)

Tutorial: creación de un entorno de computación gestionado con los recursos de Fargate

Siga estos pasos para crear un entorno de computación administrado mediante recursos de AWS Fargate .

1. Abra la AWS Batch consola en <https://console.aws.amazon.com/batch/>.
2. En la barra de navegación, seleccione la Región de AWS que desee utilizar.
3. En el panel de navegación, elija Entornos de computación.
4. Seleccione Creación de.
5. Configure el entorno de computación.

Note

Los entornos de cómputo para los Windows containers on AWS Fargate trabajos deben tener al menos una vCPU.

- a. Para la Configuración del entorno de cómputo, elija Fargate.
- b. En Nombre, especifique un nombre único para el entorno de computación. El nombre puede contener hasta 128 caracteres. Puede contener letras mayúsculas y minúsculas, números, guiones (-) y guiones bajos (_).
- c. En el rol de servicio, elige un rol vinculado al servicio que permita al AWS Batch servicio realizar llamadas a las operaciones de AWS API requeridas en tu nombre. En este ejemplo, elija AWSServiceRoleForBatch. Para obtener más información, consulte [Utilizar funciones vinculadas al servicio para AWS Batch](#).

- d. (Opcional) Amplíe las Etiquetas. Para agregar una etiqueta, elija Add tag (Añadir etiqueta). Ingrese un nombre de Clave y un Valor opcional. Seleccione Agregar etiqueta.
 - e. Seleccione Página siguiente.
6. Sección de Configuración de instancias:
- a. (Opcional) En Utilizar la capacidad de Fargate Spot, active Fargate Spot. Para obtener información sobre Fargate Spot, consulte [Uso de Amazon EC2 Spot y Fargate_SPOT](#).
 - b. En Maximum v CPUs, elija el número máximo de v al CPUs que se pueda escalar su entorno informático, independientemente de la demanda de colas de trabajos.
 - c. Seleccione Página siguiente.
7. Configure redes.

 Important

Los recursos de computación de las deben obtener acceso para comunicarse con el punto de conexión del servicio de Amazon ECS. Esto puede ser a través de un punto de conexión de VPC de la interfaz o a través de recursos de computación de las con direcciones IP públicas.

Para obtener más información acerca de los puntos de enlace de la VPC de la interfaz, consulte [Puntos de enlace de la VPC de la interfaz de Amazon ECS \(AWS PrivateLink\)](#) en la Guía para desarrolladores de Amazon Elastic Container Service.

Si no tiene configurado un punto de conexión de la VPC de la interfaz y los recursos de computación de las no tienen direcciones IP públicas, deberán utilizar traducción de direcciones de red (NAT) para proporcionar este acceso. Para obtener más información, consulte [Gateways NAT](#) en la Guía del usuario de Amazon VPC. Para obtener más información, consulte [the section called “Creación de una VPC”](#).

- a. Para el ID de la nube privada virtual (VPC), elija una VPC en la que quiera lanzar sus instancias.
- b. En Subredes, elija las subredes que vaya a utilizar. De forma predeterminada, se escogen todas las subredes dentro de la VPC disponible.

 Note

AWS Batch on Fargate no es compatible actualmente con las Zonas Locales. Para obtener más información, consulte los [Clústeres de Amazon ECS en Local Zones, Wavelength Zones y AWS Outposts](#) en la Guía para desarrolladores de Amazon Elastic Container Service.

- c. En Grupos de seguridad, seleccione un su grupo de seguridad para asociarlo a las instancias. De forma predeterminada, se escoge el grupo de seguridad predeterminado para la VPC.
 - d. Seleccione Página siguiente.
8. Para la Revisión, revise los pasos de configuración. Si necesita realizar cambios, elija Editar. Cuando haya terminado, seleccione Creación de entorno de computación.

Tutorial: creación de un entorno de computación gestionado con los recursos de Amazon EC2

Siga estos pasos para crear un entorno de computación administrado mediante recursos de Amazon Elastic Compute Cloud (Amazon EC2).

1. Abra la AWS Batch consola en <https://console.aws.amazon.com/batch/>.
2. En la barra de navegación, seleccione la Región de AWS que desee utilizar.
3. En el panel de navegación, elija Entornos.
4. Elija Crear entorno y, a continuación, Entorno de computación.
5. Configure el entorno.
 - a. Para la Configuración del entorno de computación, elija Amazon Elastic Compute Cloud (Amazon EC2).
 - b. Para el Tipo de orquestación, seleccione Administrado.
 - c. En Nombre, especifique un nombre único para el entorno de computación. El nombre puede contener hasta 128 caracteres. Puede contener letras mayúsculas y minúsculas, números, guiones (-) y guiones bajos (_).
 - d. En Rol de servicio, elige un rol vinculado al servicio que permita al AWS Batch servicio realizar llamadas a las operaciones de AWS API requeridas en tu nombre. En este ejemplo,

elija `AWSServiceRoleForBatch`. Para obtener más información, consulte [Utilizar funciones vinculadas al servicio para AWS Batch](#).

- e. En Instance role (Rol de instancia), elija si desea crear un perfil de instancia nuevo o utilizar uno ya existente que tenga asociados los permisos de IAM necesarios. Este perfil de instancia permite que las instancias de contenedor de Amazon ECS que se crean para su entorno de cómputo realicen llamadas a las operaciones de AWS API requeridas en su nombre. Para obtener más información, consulte [Rol de instancia de Amazon ECS](#). Si decide crear un perfil de instancia nuevo, se creará el rol requerido (`ecsInstanceRole`).
- f. (Opcional) Amplíe las Etiquetas.
 - i. (Opcional) En el caso de las Etiquetas de EC2, seleccione Agregar etiqueta para añadir una etiqueta a los recursos que se lanzan en el entorno de computación. Ingrese un nombre de Clave y un Valor opcional. Seleccione Agregar etiqueta.
 - ii. (Opcional) En Etiquetas, seleccione Agregar etiqueta. Ingrese un nombre de Clave y un Valor opcional. Seleccione Agregar etiqueta.

Para obtener más información, consulte [Etiquetar los recursos de AWS Batch](#).

- g. Elija Siguiente.

6. Sección de Configuración de instancias:

- a. (Opcional) En Habilitar el uso de instancias de spot, active Spot. Para obtener más información, consulte [Instancias de spot de](#).
- b. (Opcional) En Precio máximo % bajo demanda, seleccione el porcentaje máximo del precio que puede tener una instancia de spot en relación con su precio bajo demanda antes de que lancen las instancias. Por ejemplo, si el precio máximo es el 20%, el precio de spot de esa instancia de EC2 deberá ser inferior al 20% del precio bajo demanda que tenga en ese momento. Siempre se paga el precio más bajo (de mercado) y nunca más que lo marcado por el porcentaje máximo. Si se deja este campo en blanco, el valor predeterminado es el 100% del precio bajo demanda.
- c. (Solo Spot) En Rol de la flota de spot, seleccione un rol de IAM para la flota de spot de Amazon EC2 que quiera aplicar a su entorno de computación de spot. Si aún no tiene un rol de IAM para la flota de spot de Amazon EC2, primero debe crear uno. Para obtener más información, consulte [Rol de flota de spot de Amazon EC2](#).

⚠ Important

Para etiquetar las instancias puntuales al crearlas, su función de IAM de Amazon EC2 Spot Fleet debe utilizar la política gestionada por Amazon EC2 SpotFleetTaggingRole más reciente. La política EC2 SpotFleetRole gestionada por Amazon no tiene los permisos necesarios para etiquetar instancias puntuales. Para obtener más información, consulte [Instancias de spot no etiquetadas en el momento de su creación](#) y [the section called “Etiquetar los recursos”](#).

- d. Como mínimo `v CPUs`, elija el número mínimo de `v CPUs` que mantenga su entorno informático, independientemente de la demanda de colas de trabajos.
- e. En el campo `v deseado CPUs`, elija el número de `v` con el `CPUs` que se inicia el entorno informático. A medida que aumente la demanda de colas de trabajos, AWS Batch puede aumentar la cantidad deseada de `vCPU` en su entorno informático y añadir instancias EC2, hasta un máximo de `v CPUs`. A medida que disminuya la demanda, AWS Batch puede reducir la cantidad deseada de `v CPUs` en su entorno de cómputo y eliminar instancias, hasta el mínimo `v CPUs`.
- f. En `Maximum v CPUs`, elige el número máximo de `v al CPUs` que se puede escalar tu entorno de cómputo, independientemente de la demanda de colas de trabajos.
- g. (Opcional) Para reducir el retraso (minutos), elija el tiempo mínimo (en minutos) que AWS Batch permite que las instancias se ejecuten en el entorno de procesamiento una vez finalizados sus trabajos.
- h. En `Tipos de instancias permitidos`, elija los tipos de instancia de Amazon EC2 que se pueden lanzar. Se pueden especificar familias de instancias para lanzar cualquier tipo de instancia en esas familias (por ejemplo, `c5`, `c5n` o `p3`). O bien puede especificar tamaños específicos dentro de una familia (por ejemplo, `c5.8xlarge`). Los tipos de instancias metálicas no están en las familias de instancias. Por ejemplo, `c5` no incluye `c5.metal`.

AWS Batch puede seleccionar el tipo de instancia por ti si eliges una de las siguientes opciones:

- `optimal` para elegir tipos de instancias (de las familias de instancias `c4`, `m4`, `r4`, `c5`, `m5` y `r5`) que se correspondan con la demanda de las colas de trabajos.
- `default_x86_64` para elegir tipos de instancias basados en x86 (de las familias de instancias `m6i`, `c6i`, `r6i` y `c7i`) que se ajusten a las demandas de recursos de la cola de trabajos.

- `default_arm64` para elegir tipos de instancias basados en x86 (de las familias de instancias m6g, c6g, r6g y c7g) que se ajusten a las demandas de recursos de la cola de trabajos.

 Note

Desde el 1 de noviembre de 2025, el comportamiento de `optimal` se cambiará para que coincida con `default_x86_64`. Las familias de instancias podrían actualizarse a una generación más nueva durante el proceso del cambio. No es necesario realizar ninguna acción para realizar la actualización. Para obtener más información sobre los cambios, consulte

 Note

Desde el 1 de noviembre de 2025, el comportamiento de `optimal` se cambiará para que coincida con

`default_x86_64`. Las familias de instancias podrían

actualizarse a una generación más nueva durante el

proceso del cambio. No es necesario realizar ninguna acción para realizar la actualización.

AWS Batch admitía una sola opción en `InstanceTypes` `optimal` para adaptarse a la demanda de sus colas de trabajos. Integramos dos nuevas opciones de tipos de instancias: `default_x86_64` y `default_arm64`. Si no se selecciona el tipo de instancia, se utilizará `default_x86_64`. Estas nuevas opciones seleccionarán automáticamente los tipos de instancias rentables para diferentes familias y generaciones según los requisitos de la cola de trabajo, lo que permitirá ejecutar rápidamente las cargas de trabajo.

A medida que haya una capacidad suficiente de nuevos tipos de instancias disponibles en una Región de AWS, el grupo predeterminado correspondiente se actualizará automáticamente con el nuevo tipo de instancia. La opción `optimal` existente seguirá siendo compatible y no quedará obsoleta, ya que será compatible con los grupos predeterminados subyacentes para

luego brindar instancias actualizadas. Si está utilizando `optimal`, no tiene que hacer nada.

Sin embargo, ten en cuenta que solo `ENABLED` los entornos de `VALID` cómputo (CEs) se actualizarán automáticamente con los nuevos tipos de instancias. Si tienes `DISABLED` alguna `INVALID` CEs, recibirá las actualizaciones una vez que se vuelva a habilitar y se establezca en un `VALID` estado.

 Note

- La disponibilidad de las familias de instancias varía según la Región de AWS. Por ejemplo, Región de AWS es posible que algunas no tengan familias de instancias de cuarta generación, pero sí familias de instancias de quinta y sexta generación.
- Al usar la agrupación de instancias `default_x86_64` o `default_arm64`, AWS Batch selecciona las familias de instancias según un equilibrio entre rentabilidad y rendimiento. Si bien las instancias de nueva generación suelen ofrecer una mejor relación precio-rendimiento, AWS Batch puede elegir una familia de instancias de generación anterior si proporciona la combinación óptima de disponibilidad, coste y rendimiento para su carga de trabajo. Por ejemplo, en un país en el Región de AWS que estén disponibles tanto las instancias `c6i` como las `c7i`, AWS Batch puede seleccionar instancias `c6i` si ofrecen una mejor relación costo-beneficio para los requisitos específicos de su trabajo. [Para obtener más información sobre los tipos de AWS Batch instancias y su Región de AWS disponibilidad, consulta la tabla de cálculo de tipos de instancias.](#)
- AWS Batch actualiza periódicamente las instancias en paquetes predeterminados con opciones más nuevas y rentables. Las actualizaciones se realizan automáticamente sin que sea necesario que realice ninguna acción. Sus cargas de trabajo seguirán ejecutándose durante las actualizaciones sin interrupciones.

 Note

Cuando se crea un entorno de computación, los tipos de instancias que se seleccionen para dicho entorno de computación deben compartir la misma arquitectura. Por ejemplo, no se puede mezclar instancias x86 y ARM en el mismo entorno de computación.

 Note

AWS Batch se escalará en GPUs función de la cantidad requerida en sus colas de trabajos. Para utilizar la programación de GPU, el entorno de computación debe incluir tipos de instancia de las familias p3, p4, p5, p6, g3, g3s, g4, g5 o g6.

- i. Para Allocation strategy (Estrategia de asignación), elija la estrategia de asignación que se utilizará al seleccionar los tipos de instancia de la lista de tipos de instancia permitidos. BEST_FIT_PROGRESSIVE suele ser la mejor opción para los entornos de computación bajo demanda de EC2, SPOT_CAPACITY_OPTIMIZED y SPOT_PRICE_CAPACITY_OPTIMIZED para los entornos de computación Spot de EC2. Para obtener más información, consulte [the section called “Estrategias de asignación de tipos de instancia”](#).
- j. Expanda Configuración adicional.
 - i. (Opcional) En Grupo de ubicación, introduzca un nombre de grupo de ubicación para agrupar los recursos en el entorno de computación.
 - ii. (Opcional) En Par de claves EC2, elija un par de claves pública y privada como credenciales de seguridad cuando se conecte a la instancia. Para obtener más información sobre pares de claves de Amazon EC2, consulte [pares de claves de Amazon EC2 e instancias de Linux](#).
 - iii. (Opcional) Para la configuración de EC2, elija los valores de anulación de tipo de imagen e ID de imagen AWS Batch para proporcionar información y seleccionar Amazon Machine Images (AMIs) para las instancias del entorno informático. Si no se especifica la anulación del ID de imagen para cada tipo de imagen, AWS Batch selecciona una [AMI reciente optimizada para Amazon ECS](#). Si no se especifica ningún

tipo de imagen, el valor predeterminado es una instancia de Amazon Linux 2 para instancias que no sean de GPU ni de AWS Graviton.

 Important

Para usar una AMI personalizada, elija el tipo de imagen y, a continuación, introduzca el ID de AMI personalizado en el cuadro de Cambio de ID de imagen.

[Amazon Linux 2](#)

Es el valor predeterminado para todas las familias de instancias AWS basadas en Graviton (por ejemplo, C6g M6gR6g, yT4g) y se puede usar para todos los tipos de instancias que no sean de GPU.

[Amazon Linux 2 \(GPU\)](#)

Es el valor predeterminado para todas las familias de instancias de GPU (por ejemplo, P4 yG4) y se puede usar para todos los tipos de instancias que no estén basadas en AWS Graviton.

[Amazon Linux 2023](#)

AWS Batch es compatible con Amazon Linux 2023.

 Note

Amazon Linux 2023 no es compatible con las instancias A1.

[Amazon Linux 2023 \(GPU\)](#)

Es el valor predeterminado para todas las familias de instancias de GPU (por ejemplo, P4 yG4) y se puede usar para todos los tipos de instancias que no estén AWS basados en Graviton.

 Note

La AMI que elija para un entorno informático debe coincidir con la arquitectura de los tipos de instancias que tenga previsto utilizar para dicho entorno informático. Por ejemplo, si su entorno informático utiliza tipos de instancias A1, la AMI de recursos informáticos que elija debe admitir instancias ARM. Amazon ECS ofrece versiones x86 y ARM de la AMI Amazon Linux 2 optimizada para Amazon ECS. Para obtener más información, consulte la sección sobre [AMI Amazon Linux 2 optimizada para Amazon ECS](#) en la Guía para desarrolladores de Amazon Elastic Container Service.

- k. (Opcional) Amplíe las plantillas de lanzamiento
 - i. (Opcional) En Plantilla de lanzamiento predeterminada, seleccione una plantilla de lanzamiento de Amazon EC2 existente para configurar sus recursos de computación. La versión predeterminada de la plantilla se rellena automáticamente. Para obtener más información, consulte [Utilice las plantillas de lanzamiento de Amazon EC2 con AWS Batch](#).

 Note

En una plantilla de lanzamiento, puede especificar una AMI personalizada que haya creado.

- ii. (Opcional) En Versión predeterminada, ingrese `$Default`, `$Latest` o el número de versión específico que desea utilizar.

 Note

Nota: Si usa una variable de sustitución (`$Default` or `$Latest`), aplicarán el número de versión actual predeterminado o el más reciente en el momento en que se guarde esta configuración. Si la versión predeterminada o la última versión cambian en el futuro, debe actualizar la información; no se actualizará automáticamente.

 Important

Si el parámetro de versión de la plantilla de lanzamiento es `$Default` o `$Latest`, la versión predeterminada o más reciente de la plantilla de lanzamiento especificada se evalúa durante una actualización de la infraestructura. Si se selecciona un ID de AMI diferente de forma predeterminada o se selecciona la última versión de la plantilla de lanzamiento, ese ID de AMI se utiliza en la actualización. Para obtener más información, consulte [the section called “Selección de AMI durante las actualizaciones de la infraestructura”](#).

- iii. (Opcional) En Anular la plantilla de lanzamiento, seleccione Agregar plantilla de lanzamiento de anulación
 - A. (Opcional) En Plantilla de lanzamiento, seleccione una plantilla de lanzamiento de Amazon EC2 existente para usarla en familias y tipos de instancias específicos.
 - B. (Opcional) En Versión predeterminada, ingrese un número de versión específico, `$Default` o `$Latest`.

 Note

Si utiliza la variable `$Default` o `$Latest`, AWS Batch aplicará la información actual en el momento en que se cree el entorno de computación. Si la versión predeterminada o la última versión cambian en el futuro, debe actualizar la información mediante [UpdateComputeEnvironment](#) mediante el botón Consola de administración de AWS - AWS Batch.

- C. (Opcional) En Tipos de instancia de destino, seleccione el tipo de instancia o la familia a la que desee aplicar la plantilla de lanzamiento de anulación.

 Note

Si especifica una plantilla de lanzamiento de anulación, los Tipos de instancias de destino serán obligatorios. Para obtener más información, consulte [LaunchTemplateSpecificationOverride.targetInstanceTypes](#).

 Note

Si el tipo de instancia o la familia que desea seleccionar no aparecen en esta lista, revise las selecciones que realizó en `Allowed instance types`.

I. Elija Siguiente.

7. En la sección Configuración de red:

 Important

Los recursos de computación de las deben obtener acceso para comunicarse con el punto de conexión del servicio de Amazon ECS. Esto puede ser a través de un punto de conexión de VPC de la interfaz o a través de recursos de computación de las con direcciones IP públicas.

Para obtener más información acerca de los puntos de enlace de la VPC de la interfaz, consulte [Puntos de enlace de la VPC de la interfaz de Amazon ECS \(AWS PrivateLink\)](#) en la Guía para desarrolladores de Amazon Elastic Container Service.

Si no tiene configurado un punto de conexión de la VPC de la interfaz y los recursos de computación de las no tienen direcciones IP públicas, deberán utilizar traducción de direcciones de red (NAT) para proporcionar este acceso. Para obtener más información, consulte [Gateways NAT](#) en la Guía del usuario de Amazon VPC. Para obtener más información, consulte [the section called “Creación de una VPC”](#).

- a. Para el ID de la nube privada virtual (VPC), elija una VPC donde lanzar sus instancias.
- b. En Subredes, elija las subredes que vaya a utilizar. De forma predeterminada, se escogen todas las subredes dentro de la VPC disponible.

 Note

AWS Batch en Amazon EC2 es compatible con Zonas Locales. Para obtener más información, consulte [Zonas locales](#) en la Guía del usuario de Amazon EC2 y [clústeres de Amazon ECS en zonas locales, zonas de Wavelength y AWS Outposts](#) en la Guía para desarrolladores de Amazon Elastic Container Service.

- c. (Opcional) En Grupos de seguridad, seleccione su grupo de seguridad para asociarlo a las instancias. De forma predeterminada, se escoge el grupo de seguridad predeterminado para la VPC.

 Note

Nota: Si usa una variable de sustitución (\$Default or \$Latest), aplicarán el número de versión actual predeterminado o el más reciente en el momento en que se guarde esta configuración. Si la versión predeterminada o la última versión cambian en el futuro, debe actualizar la información; no se actualizará automáticamente.

8. Seleccione Página siguiente.
9. Para la Revisión, revise los pasos de configuración. Si necesita realizar cambios, elija Editar. Cuando haya terminado, seleccione Creación de entorno de computación.

Tutorial: creación de un entorno de computación no gestionado con los recursos de Amazon EC2

Siga estos pasos para crear un entorno de computación no gestionado mediante recursos de Amazon Elastic Compute Cloud (Amazon EC2).

1. Abra la AWS Batch consola en <https://console.aws.amazon.com/batch/>.
2. En la barra de navegación, seleccione la Región de AWS que desee utilizar.
3. En la página Entornos de computación, elija Creación de.
4. Configure el entorno.
 - a. Para la Configuración del entorno de computación, elija Amazon Elastic Compute Cloud (Amazon EC2).
 - b. Para el Tipo de orquestación, seleccione No administrado.
5. En Nombre, especifique un nombre único para el entorno de computación. El nombre puede tener una longitud máxima de 128 caracteres. Puede contener letras mayúsculas y minúsculas, números, guiones (-) y guiones bajos (_).
6. En Función de servicio, elige una función que permita al AWS Batch servicio realizar llamadas a las operaciones de AWS API requeridas en tu nombre.

 Note

No se puede utilizar `AWSServiceRoleForBatch` para entornos de computación no gestionados.

7. En Maximum v CPUs, elija el número máximo de v al CPUs que se pueda escalar su entorno informático, independientemente de la demanda de colas de trabajos.
8. (Opcional) Amplíe las Etiquetas. Para agregar una etiqueta, elija Add tag (Añadir etiqueta). Ingrese un nombre de Clave y un Valor opcional. Seleccione Agregar etiqueta. Para obtener más información, consulte [Etiquetar los recursos de AWS Batch](#).
9. Seleccione Página siguiente.
10. Para la Revisión, revise los pasos de configuración. Si necesita realizar cambios, elija Editar. Cuando haya terminado, seleccione Creación de entorno de computación.

Tutorial: creación de un entorno de computación gestionado con los recursos de Amazon EKS

Siga estos pasos para crear un entorno de computación administrado mediante recursos de Amazon Elastic Kubernetes Service (Amazon EKS).

1. Abra la AWS Batch consola en <https://console.aws.amazon.com/batch/>.
2. En la barra de navegación, seleccione la Región de AWS que desee utilizar.
3. En el panel de navegación, elija Entornos de computación.
4. Seleccione Creación de.
5. Para la Configuración del entorno de computación, elija Amazon Elastic Kubernetes Service (Amazon EKS).
6. En Nombre, especifique un nombre único para el entorno de computación. El nombre puede tener una longitud máxima de 128 caracteres. Puede contener letras mayúsculas y minúsculas, números, guiones (-) y guiones bajos (_).
7. En Rol de instancia, elija un perfil de instancia existente que tenga asociados los permisos de IAM necesarios.

 Note

Para crear un entorno de procesamiento en la AWS Batch consola, elija un perfil de instancia que tenga los `eks:DescribeCluster` permisos `eks:ListClusters` y.

8. Para el clúster de EKS, elija un clúster de Amazon EKS existente.
9. En Espacio de nombres, introduzca un espacio de nombres Kubernetes para agrupar los procesos AWS Batch en el clúster.
10. (Opcional) Amplíe las Etiquetas. Elija Agregar etiqueta y, a continuación, introduzca un par clave-valor.
11. Seleccione Página siguiente.
12. (Opcional) En Utilizar las instancias de spot de EC2, active Habilitar el uso de instancias spot para utilizar las instancias de spot de Amazon EC2.
13. (Opcional) En Precio máximo % bajo demanda, seleccione el porcentaje máximo del precio que puede tener una instancia de spot en relación con su precio bajo demanda antes de que lancen las instancias. Por ejemplo, si el precio máximo es el 20%, el precio de spot de esa instancia de EC2 deberá ser inferior al 20% del precio bajo demanda que tenga en ese momento. Siempre se paga el precio más bajo (de mercado) y nunca más que lo marcado por el porcentaje máximo. Si se deja este campo en blanco, el valor predeterminado es el 100% del precio bajo demanda.
14. (Solo Spot) En Rol de la flota de spot, seleccione un rol de IAM para la flota de spot de Amazon EC2 que quiera aplicar a su entorno de computación SPOT.

 Important

Este rol es necesario si la estrategia de asignación está establecida en `BEST_FIT` o no se ha especificado.

15. (Opcional) Para obtener un valor mínimo de `v` CPUs, elija el número mínimo de `v` CPUs que mantenga su entorno de cómputo, independientemente de la demanda de colas de trabajos.
16. (Opcional) Para obtener un valor máximo de `v` CPUs, elija el número máximo de `v` al CPUs que pueda ampliarse su entorno informático, independientemente de la demanda de colas de trabajos.
17. (Opcional) Para reducir el retraso (minutos), elige el tiempo mínimo (en minutos) que AWS Batch permite que las instancias se ejecuten en el entorno de procesamiento una vez finalizados sus trabajos.

18. En Tipos de instancias permitidos, elija los tipos de instancia de Amazon EC2 que se pueden lanzar. Se pueden especificar familias de instancias para lanzar cualquier tipo de instancia en esas familias (por ejemplo, c5, c5n o p3). O bien puede especificar tamaños específicos dentro de una familia (por ejemplo, c5.8xlarge). Los tipos de instancias metálicas no están en las familias de instancias. Por ejemplo, c5 no incluye c5.metal.

AWS Batch puede seleccionar el tipo de instancia si eliges una de las siguientes opciones:

- `optimal` para elegir tipos de instancias (de las familias de instancias c4, m4, r4, c5, m5 y r5) que se correspondan con la demanda de las colas de trabajos.
- `default_x86_64` para elegir tipos de instancias basados en x86 (de las familias de instancias m6i, c6i, r6i y c7i) que se ajusten a las demandas de recursos de la cola de trabajos.
- `default_arm64` para elegir tipos de instancias basados en ARM (de las familias de instancias m6g, c6g, r6g y c7g) que se ajusten a las demandas de recursos de la cola de trabajos.

 Note

Desde el 1 de noviembre de 2025, el comportamiento de `optimal` se cambiará para que coincida con `default_x86_64`. Las familias de instancias podrían actualizarse a una generación más nueva durante el proceso del cambio. No es necesario realizar ninguna acción para realizar la actualización. Para obtener más información sobre los cambios, consulte [Configuración óptima del tipo de instancia para recibir actualizaciones automáticas de la familia de instancias](#).

 Note

- La disponibilidad de las familias de instancias varía según la Región de AWS. Por ejemplo, Región de AWS es posible que algunas no tengan familias de instancias de cuarta generación, pero sí familias de instancias de quinta y sexta generación.
- Al usar nuestros `default_x86_64` paquetes de `default_arm64` instancias, AWS Batch selecciona las familias de instancias en función de un equilibrio entre rentabilidad y rendimiento. Si bien las instancias de nueva generación suelen ofrecer una mejor relación precio-rendimiento, AWS Batch puede elegir una familia de instancias de generación anterior si proporciona la combinación óptima de

disponibilidad, coste y rendimiento para su carga de trabajo. Por ejemplo, en un país en el Región de AWS que estén disponibles tanto las instancias c6i como las c7i, AWS Batch puede seleccionar instancias c6i si ofrecen una mejor rentabilidad para los requisitos específicos de su trabajo. [Para obtener más información sobre los tipos de AWS Batch instancias y su Región de AWS disponibilidad, consulta la tabla de cálculo de tipos de instancias.](#)

- AWS Batch actualiza periódicamente las instancias en paquetes predeterminados con opciones más nuevas y rentables. Las actualizaciones se realizan automáticamente sin que sea necesario que realice ninguna acción. Sus cargas de trabajo seguirán ejecutándose durante las actualizaciones sin interrupciones.

 Note

Cuando se crea un entorno de computación, los tipos de instancias que se seleccionen para dicho entorno de computación deben compartir la misma arquitectura. Por ejemplo, no se puede mezclar instancias x86 y ARM en el mismo entorno de computación.

 Note

AWS Batch se escalará en GPUs función de la cantidad requerida en sus colas de trabajos. Para utilizar la programación de GPU, el entorno de computación debe incluir tipos de instancia de las familias p3, p4, p5, p6, g3, g3s, g4, g5 o g6.

19. (Opcional) Expandir Configuración adicional.

- (Opcional) En Grupo de ubicación, introduzca un nombre de grupo de ubicación para agrupar los recursos en el entorno de computación.
- Para la Estrategia de asignación, elija BEST_FIT_PROGRESSIVE.
- (Opcional) Para la configuración de Amazon Machine Images (AMIs), seleccione la configuración Añadir imágenes de Amazon Machine Images (AMI).

Puede utilizar una AMI de Amazon Linux optimizada para Amazon EKS o una AMI personalizada.

- Cómo utilizar una [AMI de Amazon Linux optimizada para Amazon EKS](#):

A. En Tipo de imagen, seleccione una de las siguientes opciones:

- [Amazon Linux 2](#): predeterminado para todas las familias de instancias AWS basadas en Graviton (por ejemplo, C6g M6gR6g, yT4g) y se puede usar para todos los tipos de instancias que no son de GPU.
- [Amazon Linux 2 \(acelerado\)](#): predeterminado para todas las familias de instancias de GPU (por ejemplo, P4 yG4) y se puede usar para todos los tipos de instancias que no estén AWS basados en Graviton.
- [Amazon Linux 2023](#): AWS Batch es compatible con Amazon Linux 2023 (AL2023).
- [Amazon Linux 2023 \(acelerado\)](#): Familias de instancias de GPU que se pueden utilizar para todos los tipos de instancias no basados en AWS Graviton.

B. En versión de Kubernetes, ingrese un [número de versión de Kubernetes](#).

ii. Cómo utilizar una AMI personalizada:

A. En Tipo de imagen, elija el tipo de AMI en el que se basa la AMI personalizada:

- [Amazon Linux 2](#): predeterminado para todas las familias de instancias AWS basadas en Graviton (por ejemplo, C6g M6gR6g, yT4g) y se puede usar para todos los tipos de instancias que no son de GPU.
- [Amazon Linux 2 \(acelerado\)](#): predeterminado para todas las familias de instancias de GPU (por ejemplo, P4 yG4) y se puede usar para todos los tipos de instancias que no estén AWS basados en Graviton.
- [Amazon Linux 2023](#): AWS Batch compatible AL2023.
- [Amazon Linux 2023 \(acelerado\)](#): Familias de instancias de GPU que se pueden utilizar para todos los tipos de instancias no basados en AWS Graviton.

B. En Anular el ID de imagen, ingrese el ID de AMI personalizado.

C. En versión de Kubernetes, ingrese un [número de versión de Kubernetes](#).

d. (Opcional) En Plantilla de lanzamiento, elija una plantilla de [lanzamiento existente](#).

e. (Opcional) En Versión de la plantilla de lanzamiento, introduzca **\$Default**, **\$Latest** o un número de versión.

f. (Opcional) En Anular la plantilla de lanzamiento, para agregar una sobreescritura, seleccione Agregar plantilla de lanzamiento de anulación:

- i. (Opcional) En Plantilla de lanzamiento, elija la plantilla de lanzamiento a la que desee anular la sobreescritura.
- ii. (Opcional) En Versión de la plantilla de lanzamiento, elija el número de versión de la plantilla de lanzamiento, `$Default` o `$Latest`.
- iii. (Opcional) En Tipos de instancias de destino, elija el tipo de instancia o la familia a la que se debe aplicar esta anulación. Esto solo puede dirigirse a los tipos y familias de instancias que estén incluidos en los Tipos de instancias permitidos.
- iv. (Opcional) En `userDataType`, elija la inicialización del nodo de EKS. Utilice este campo únicamente si tiene una AMI especificada en la plantilla de lanzamiento o como anulación de la plantilla de lanzamiento. Elija `EKS_NODEADM` para personalizar AMIs según **EKS_AL2023** `EKS_AL2023_NVIDIA` o `EKS_BOOSTRAP_SH` para y. `EKS_AL2` `EKS_AL_NVIDIA` El valor predeterminado es `EKS_BOOSTRAP_SH`.

Utilizaría `UserData` cuando tenga un [entorno mixto en el que utilice ambos y un entorno personalizado basado en el mismo entorno informático](#). `AL2` `AL2023` AMIs

20. Seleccione Página siguiente.
21. Para el ID de la nube privada virtual (VPC), elija una VPC donde lanzar las instancias.
22. En Subredes, elija las subredes que vaya a utilizar. De forma predeterminada, se escogen todas las subredes dentro de la VPC disponible.

 Note

AWS Batch en Amazon EKS admite Zonas Locales. Para obtener más información, consulte [Amazon EKS y Zonas AWS Locales](#) en la Guía del usuario de Amazon EKS.

23. (Opcional) En Grupos de seguridad, seleccione su grupo de seguridad para asociarlo a las instancias. De forma predeterminada, se elige el grupo de seguridad predeterminado para la VPC.
24. Seleccione Página siguiente.
25. Para la Revisión, revise los pasos de configuración. Si necesita realizar cambios, elija Editar. Cuando haya terminado, seleccione Creación de entorno de computación.

Tutorial: Creación de un entorno informático no gestionado con los recursos de Amazon EKS

Complete los siguientes pasos para crear un entorno informático no administrado con los recursos de Amazon Elastic Kubernetes Service (Amazon EKS).

1. Abra la consola en. AWS Batch <https://console.aws.amazon.com/batch/>
2. En la barra de navegación situada en la parte superior de la página, seleccione la Región de AWS que desee utilizar.
3. En el panel de navegación, elija Entornos de computación.
4. Seleccione Creación de.
5. Configure el entorno.
 - a. Para la Configuración del entorno de computación, elija Amazon Elastic Kubernetes Service (Amazon EKS).
 - b. Para el Tipo de orquestación, seleccione No administrado.
6. En Nombre, especifique un nombre único para el entorno de computación. El nombre puede tener una longitud máxima de 128 caracteres. Puede contener letras mayúsculas y minúsculas, números, guiones (-) y guiones bajos (_).
7. Para el clúster de EKS, elija un clúster de Amazon EKS existente. Para crear un nuevo clúster de EKS, siga los pasos de la [página Crear un clúster de Amazon EKS](#).
8. En Espacio de nombres, introduzca un espacio de nombres Kubernetes para agrupar los procesos AWS Batch en el clúster.
9. (Opcional) Para Maximum v CPUs, especifique el número máximo de v CPUs disponible para la programación de trabajos con la capacidad aprovisionada.
10. (Opcional) Amplíe las Etiquetas. Elija Agregar etiqueta y, a continuación, introduzca un par clave-valor.
11. Seleccione Página siguiente.
12. Para la Revisión, revise los pasos de configuración. Si necesita realizar cambios, elija Editar. Cuando haya terminado, seleccione Creación de entorno de computación.

Asignación de nodos de clúster de Amazon EKS a un entorno informático no gestionado

Tras crear el entorno informático no gestionado, debe etiquetar los nodos de Amazon EKS con el UUID del entorno informático.

En primer lugar, obtenga el UUID del entorno de cómputo del resultado de la `APIDescribeComputeEnvironments`:

```
$ aws batch describe-compute-environments \
  --compute-environments unmanagedEksCE \
  --query "computeEnvironments[].{name: computeEnvironmentName, uuid: uuid}"
```

Obtenga la información del nodo:

```
kubectl get nodes -o name
```

Etiquete los nodos con el UUID del entorno de AWS Batch cómputo:

```
kubectl label <node-name> batch.amazonaws.com/compute-environment-uuid=uuid
```

Recurso: plantilla de entorno de computación

En el siguiente ejemplo, se muestra una plantilla de entorno de computación vacía. Puede utilizar esta plantilla para crear un entorno de computación que puede guardar en un archivo y utilizarse con la opción `--cli-input-json` de la AWS CLI. Para obtener más información sobre estos parámetros, consulte [CreateComputeEnvironment](#) en la Referencia de la API AWS Batch.

Note

Puede generar una plantilla de entorno de computación con el siguiente comando de la AWS CLI:

```
$ aws batch create-compute-environment --generate-cli-skeleton
```

```
{
  "computeEnvironmentName": "",
```

```
"type": "UNMANAGED",
"state": "DISABLED",
"unmanagedvCpus": 0,
"computeResources": {
  "type": "EC2",
  "allocationStrategy": "BEST_FIT_PROGRESSIVE",
  "minvCpus": 0,
  "maxvCpus": 0,
  "desiredvCpus": 0,
  "instanceTypes": [
    ""
  ],
  "imageId": "",
  "subnets": [
    ""
  ],
  "securityGroupIds": [
    ""
  ],
  "ec2KeyPair": "",
  "instanceRole": "",
  "tags": {
    "KeyName": ""
  },
  "placementGroup": "",
  "bidPercentage": 0,
  "spotIamFleetRole": "",
  "launchTemplate": {
    "launchTemplateId": "",
    "launchTemplateName": "",
    "version": ""
  },
  "ec2Configuration": [
    {
      "imageType": "",
      "imageIdOverride": "",
      "imageKubernetesVersion": ""
    }
  ]
},
"serviceRole": "",
"tags": {
  "KeyName": ""
},
```

```

    "eksConfiguration": {
      "eksClusterArn": "",
      "kubernetesNamespace": ""
    }
  }
}

```

Tabla de computación para el tipo de instancia

En la siguiente tabla Región de AWS, se muestra la palabra clave de familia de instancias y las familias de instancias disponibles. AWS Batch intentará asignar una instancia de la familia más reciente, pero dado que la disponibilidad de la familia de instancias varía según la familia de instancias, Región de AWS es posible que obtenga una generación de familias de instancias anterior.

default_x86_64

Region	Familias de instancias
Todo Región de AWS es ese apoyo AWS Batch	m6i, c6i, r6i c7i

default_arm64

Region	Familias de instancias
Todo Región de AWS es ese apoyo AWS Batch	m6g, c6g, r6g c7g

Óptimo

Region	Familias de instancias
- ap-northeast-1 - ap-northeast-2 - ap-south-1 - ap-southeast-1 - ap-southeast-2	m4, c4, r4

Region	Familias de instancias
• ca-central-1 • cn-north-1 • cn-northwest-1 • eu-central-1 • eu-west-1 • eu-west-2 • sa-east-1 • us-east-1 • us-east-2 • us-gov-west-1 • us-west-1 • us-west-2	

Region	Familias de instancias
• af-south-1 • ap-east-1 • ap-northeast-3 • ap-south-2 • ap-southeast-3 • ap-southeast-4 • ca-west-1 • eu-central-2 • eu-north-1 • eu-south-1 • eu-south-2 • eu-west-3 • il-central-1 • me-central-1 • me-south-1 • us-gov-east-1 • us-isob-east-1 • us-iso-east-1 • us-isof-south-1 • us-isof-east-1 • eu-isoe-west-1	m5, c5, r5
• ap-southeast-5 • ap-southeast-7 • ap-east-2 • mx-central-1	m6, c6, r6

Actualizar un entorno informático en AWS Batch

AWS Batch proporciona varias estrategias para actualizar los entornos de cómputo, cada una diseñada para escenarios y requisitos de actualización específicos. Estos enfoques utilizan la misma API de actualización subyacente, pero representan diferentes métodos prescriptivos para administrar las actualizaciones de forma eficaz. Puede gestionar estas actualizaciones mediante la AWS Batch consola o el AWS CLI. Comprender estas estrategias permite elegir el método más adecuado para sus necesidades, sin dejar de disminuir las interrupciones de las cargas de trabajo.

En este tema se entrega información general sobre las estrategias de actualización disponibles y sobre cuándo usar cada enfoque. Para obtener información sobre los procedimientos detallados, consulte las secciones individuales de cada estrategia de actualización.

Important

AWS Batch crea y administra varios AWS recursos en su nombre y dentro de su cuenta, incluidas las plantillas de lanzamiento de Amazon EC2, los grupos de Auto Scaling de Amazon EC2, las flotas puntuales de Amazon EC2 y los clústeres de Amazon ECS. Estos recursos administrados están configurados específicamente para garantizar un funcionamiento óptimo de AWS Batch. La modificación manual de estos recursos AWS Batch gestionados, a menos que se indique explícitamente en la AWS Batch documentación, puede provocar un comportamiento inesperado, como los entornos INVALID informáticos, un comportamiento de escalado de instancias subóptimo, retrasos en el procesamiento de la carga de trabajo o costes inesperados. El servicio de AWS Batch no puede respaldar estas modificaciones manuales de forma determinista. Utilice siempre el soporte AWS Batch APIs o la AWS Batch consola para gestionar sus entornos informáticos.

Las modificaciones manuales no admitidas incluyen ejecutar sus propias tareas o servicios de Amazon ECS en AWS Batch clústeres de Amazon ECS administrados o iniciar procesos, daemons o servicios adicionales directamente en instancias administradas. AWS Batch asume el control total de los recursos informáticos en un entorno informático gestionado y puede finalizar instancias, detener tareas o escalar el clúster en cualquier momento. Cualquier carga de trabajo que ejecute fuera de la presentación de AWS Batch trabajos en estos recursos gestionados puede interrumpirse sin previo aviso. La ejecución de AWS Batch cargas de trabajo no AWS Batch gestionadas en clústeres e instancias gestionados también puede interferir con la programación de los AWS Batch trabajos y el escalado de las instancias.

Temas

- [Estrategias de actualización del entorno de computación](#)
- [Cómo elegir la mejor estrategia de actualización](#)
- [Consideraciones sobre la actualización de AMI](#)
- [Cómo realizar actualizaciones de escalado](#)
- [Cómo realizar actualizaciones de la infraestructura](#)
- [Realice blue/green actualizaciones para entornos de cómputo](#)

Estrategias de actualización del entorno de computación

El entorno de computación se actualiza en su lugar cuando se utilizan actualizaciones de escalado o de infraestructura. Para la estrategia de blue/green actualización, debe crear un nuevo entorno informático (verde) y, a continuación, migrar la carga de trabajo del entorno informático anterior (azul) al nuevo entorno informático (verde).

AWS Batch proporciona tres estrategias diferentes para las actualizaciones del entorno informático:

Actualizaciones de escalado

Las actualizaciones de escalado ajustan la capacidad del entorno de computación al agregar o eliminar instancias sin reemplazar las existentes. Este es el escenario de actualización más rápido y no requiere tiempo de inactividad. Utilice las actualizaciones de escalado cuando necesite cambiar la configuración de capacidad (vCPUs). Es normal que estas actualizaciones se completen en tan solo minutos.

Las actualizaciones de Fargate se realizan mediante los mismos procedimientos que las actualizaciones de escalado. Para obtener más información, consulte [Cómo realizar actualizaciones de escalado](#).

Actualizaciones de la infraestructura

Las actualizaciones de infraestructura sustituyen las instancias del entorno de computación por unas nuevas, ya que tienen una configuración actualizada. Estas actualizaciones requieren configuraciones específicas de rol de servicio y estrategia de asignación, pero proporcionan un tiempo de inactividad mínimo, y es posible que se interrumpan las tareas en ejecución. Utilice las actualizaciones de infraestructura cuando necesite modificar los tipos de instancias, la configuración de la AMI, los ajustes de red, el rol de servicio, el estado del entorno u otros

componentes de la infraestructura. Estas actualizaciones suelen completarse en un plazo de 10 a 30 minutos, según la finalización del trabajo.

Para obtener más información, consulte [Cómo realizar actualizaciones de la infraestructura](#).

Actualizaciones azul/verde

Blue/green updates create a new compute environment alongside your existing environment, allowing gradual workload transition with zero downtime. This approach provides the safest update path but requires running two environments temporarily. Use blue/green updates when you don't need any downtime, want to test changes before full implementation, need a fast rollback capability, or use configurations incompatible with infrastructure updates. You can regulate and control the completion time.

Para obtener más información, consulte [Realice blue/green actualizaciones para entornos de cómputo](#).

Cómo elegir la mejor estrategia de actualización

Con esta guía de decisiones puede seleccionar la estrategia de actualización más adecuada de acuerdo a sus necesidades:

Estrategia de actualizaciones de escalado

Elija la estrategia de actualización escalable cuando solo necesite ajustar la capacidad de cómputo (vCPUs). Las actualizaciones de escalado son ideales cuando se necesitan actualizaciones rápidas sin tiempo de inactividad ni cambios en la configuración de la infraestructura.

Para obtener procedimientos detallados, consulte [Cómo realizar actualizaciones de escalado](#).

Estrategia de actualizaciones de infraestructura

Elija la estrategia de actualización de infraestructura cuando necesite modificar los tipos de instancias, la configuración de la AMI, el rol de servicio, el estado del entorno o la configuración de la red. Su entorno debe utilizar la función `AWSServiceRoleForBatch` vinculada al servicio y una estrategia de asignación de `BEST_FIT_PROGRESSIVE_SPOT_CAPACITY_OPTIMIZED`, o `SPOT_PRICE_CAPACITY_OPTIMIZED`. Las actualizaciones de infraestructura funcionan bien cuando se acepta alguna interrupción del trabajo durante la actualización y se desean actualizaciones automáticas a la última AMI optimizada para Amazon ECS.

Para obtener procedimientos detallados, consulte [Cómo realizar actualizaciones de la infraestructura](#).

Elija blue/green las actualizaciones cuando

Elija la estrategia de blue/green actualización cuando sus cargas de trabajo no requieran ningún tiempo de inactividad o necesite probar los cambios antes de realizar la transición de las cargas de trabajo de producción. Este enfoque es esencial cuando es importante contar con una capacidad de reversión rápida, si su entorno utiliza una estrategia de BEST_FIT asignación o si su entorno no utiliza la función vinculada al servicio. AWSServiceRoleForBatch Blue/green Las actualizaciones también son la mejor opción cuando se utilizan actualizaciones personalizadas AMIs que requieren actualizaciones manuales o si es necesario realizar cambios importantes en la configuración.

Para obtener procedimientos detallados, consulte [Realice blue/green actualizaciones para entornos de cómputo](#).

Consideraciones sobre la actualización de AMI

El enfoque de actualización AMIs depende de la configuración del entorno informático.

Actualización de la AMI predeterminada AWS Batch proporcionada a la última

AWS Batch puede actualizarse a la última AMI optimizada para Amazon ECS durante las actualizaciones de [infraestructura](#) cuando se cumplen todas estas condiciones:

Note

Tras completada la actualización de la infraestructura, `updateToLatestImageVersion` se establece en `false`. Para iniciar otra actualización, `updateToLatestImageVersion` se debe establecer en `true`.

- El entorno informático utiliza la función vinculada al servicio AWSServiceRoleForBatch
- La estrategia de asignación se establece en `BEST_FIT_PROGRESSIVE`, `SPOT_CAPACITY_OPTIMIZED` o `SPOT_PRICE_CAPACITY_OPTIMIZED`
- No se especifica explícitamente ningún ID de AMI en `imageId`, `imageIdOverride` o en la plantilla de lanzamiento
- `updateToLatestImageVersion` se establece en `true`

Actualizaciones de la AMI mediante blue/green la implementación

Debe utilizar la blue/green implementación para actualizar AMIs en los siguientes escenarios:

- Si se utiliza la estrategia de asignación de BEST_FIT (no admite actualizaciones de infraestructura)
- Cuando no utilice la función vinculada al [AWSServiceRoleForBatch](#) [servicio](#)

Actualizaciones de la AMI para una AMI personalizada

Si especifica una AMI personalizada en la plantilla de lanzamiento del entorno informático, el `imageId` parámetro o el parámetro de la `imageIdOverride` configuración de EC2 no AWS Batch actualizarán automáticamente la AMI personalizada durante las actualizaciones de infraestructura. Puedes actualizar un identificador de AMI personalizado especificando el nuevo identificador en el parámetro que se usó originalmente durante la creación del entorno de cómputo. Si desea cambiar a una AMI AWS Batch proporcionada, puede hacerlo eliminando el ID de AMI personalizado en la actualización de su entorno informático.

Cómo realizar actualizaciones de escalado

Las actualizaciones de escalado ajustan la capacidad del entorno de computación al agregar o eliminar instancias. Esta es la estrategia de actualización más rápida y no requiere reemplazar las instancias existentes. Las actualizaciones de escalado funcionan con cualquier tipo de función de servicio y estrategia de asignación, lo que las convierte en la opción de actualización más flexible.

Cambios que activan una actualización de escalado

AWS Batch realiza una actualización de escalado cuando se modifican únicamente los siguientes ajustes. Si modifica alguna de estas configuraciones junto con otras configuraciones del entorno de cómputo, AWS Batch realiza una [actualización de infraestructura](#) en su lugar.

Los siguientes ajustes activan las actualizaciones de escalado cuando únicamente se modifica:

- `desiredvCpus`— Establece el número objetivo de v CPUs para el entorno.
- `maxvCpus`— Define el número máximo de v CPUs que se puede lanzar.
- `minvCpus`— Especifica el número mínimo de v CPUs que se debe mantener.
- `minScaleDownDelayMinutes`— Especifica el tiempo mínimo (en minutos) que AWS Batch permite que las instancias se ejecuten en el entorno informático una vez finalizados sus trabajos.

 Note

`minScaleDownDelayMinutes` no se aplica a las instancias que se sustituyen durante las actualizaciones de infraestructura.

Para los entornos de cómputo de Fargate, también se pueden modificar estos ajustes para las actualizaciones de escalado:

- `securityGroupIds`— Grupo de seguridad IDs para el entorno informático.
- `subnets`: Subredes para el entorno de computación.

 Note

Recomendamos no utilizarlo `desiredvCpus` para iniciar una actualización de escalado, ya que AWS Batch se ajustará de forma dinámica `desiredvCpus`. En su lugar, debe utilizar `minvCpus`.

Al actualizar `desiredvCpus`, el valor debe estar comprendido entre `minvCpus` y `maxvCpus`. El nuevo valor debe ser mayor o igual que `desiredvCpus`. Para obtener más información, consulte [the section called “Mensaje de error al actualizar la configuración de `desiredvCpus`”](#).

 Important

Si modifica alguna de estas configuraciones de escalado junto con otras configuraciones del entorno de cómputo (como tipos de instancias IDs, AMI o plantillas de lanzamiento), AWS Batch realiza una actualización de infraestructura en lugar de una actualización de escalado. Las actualizaciones de infraestructura tardan más tiempo y pueden sustituir a las instancias existentes.

Performing scaling updates using the Consola de administración de AWS

1. Abra la AWS Batch consola en <https://console.aws.amazon.com/batch/>.

2. En el panel de navegación, seleccione Entornos y luego en la pestaña Entornos de computación.
3. Seleccione el entorno de computación para actualizarlo.
4. Elija Acciones y, a continuación, elija Editar.
5. Modifique una o más de las [configuraciones que admiten actualizaciones de escalado](#). Por ejemplo:
 - En Minimum v CPUs, introduzca el número mínimo de vCPUs.
 - Para la v deseada CPUs, introduzca el número de v deseado CPUs.
 - En Máximo v CPUs, introduzca el número máximo de vCPUs.
6. Seleccione Save changes (Guardar cambios).
7. Supervise el estado del entorno de computación. La actualización debe completarse rápidamente, ya que solo implica operaciones de escalado.

Performing scaling updates using the AWS CLI

Utilice el comando `update-compute-environment` para realizar actualizaciones de escalado. En los siguientes ejemplos se muestran las operaciones de escalado comunes. Puede modificar una o más de las siguientes [configuraciones que admiten actualizaciones de escalado](#).

- En este ejemplo se actualizan los valores v deseado, mínimo y máximo CPUs:

```
aws batch update-compute-environment \  
  --compute-environment your-compute-environment-name \  
  --compute-resources minvCpus=2,maxvCpus=8
```

Supervisión de las actualizaciones de escalado

Supervise las actualizaciones de escalado mediante la AWS Batch consola para ver el estado del entorno de procesamiento y comprobar el recuento de instancias y las métricas de vCPU. También puedes usar el `describe-compute-environments` comando AWS CLI with para comprobar el estado y supervisar los recuentos de instancias y los valores de vCPU.

Cómo realizar actualizaciones de la infraestructura

Las actualizaciones de infraestructura sustituyen las instancias del entorno de computación por unas nuevas, ya que tienen una configuración actualizada. Esta estrategia de actualización tarda más que el escalado de las actualizaciones y requiere una configuración específica del rol de servicio y la estrategia de asignación. Las actualizaciones de infraestructura brindan una forma de modificar las configuraciones fundamentales del entorno de computación sin perder la disponibilidad del servicio.

Important

Las actualizaciones de infraestructura requieren una función `AWSServiceRoleForBatch` vinculada al servicio y una estrategia de asignación de `BEST_FIT_PROGRESSIVE_SPOT_CAPACITY_OPTIMIZED`, o. `SPOT_PRICE_CAPACITY_OPTIMIZED`. Si su entorno no cumple con estos requisitos, utilice `blue/green` actualizaciones en su lugar.

Cambios que activan las actualizaciones de la infraestructura

Al modificar cualquiera de las siguientes configuraciones, AWS Batch realiza una actualización de la infraestructura. Las actualizaciones de la infraestructura también se producen cuando se modifican estos ajustes junto con el escalado de los ajustes de actualización.

Los siguientes ajustes activan las actualizaciones de infraestructura:

Configuración de computación

- `allocationStrategy`— Determina cómo AWS Batch selecciona los tipos de instancias.
- `instanceTypes`: Especifica los tipos de instancias de EC2 que se deben utilizar
- `bidPercentage`: Porcentaje máximo del precio bajo demanda para las instancias de spot
- `type`: Tipo de entorno de computación (EC2 o SPOT)

AMI y configuración de lanzamiento

- `imageId`: AMI específica para utilizar en las instancias
- `ec2Configuration`: Configuración de EC2, como `imageIdOverride`
- `launchTemplate`: Configuración de la plantilla de lanzamiento de EC2

- `ec2KeyPair`: Par de claves SSH para el acceso a la instancia
- `updateToLatestImageVersion`: Configuración de actualizaciones automáticas de AMI

Redes y seguridad

- `subnets`: Subredes de VPC donde se lanzan las instancias (para entornos de computación de EC2)
- `securityGroupIds`: Grupos de seguridad para instancias (para entornos de computación de EC2)
- `placementGroup`: Configuración de grupos de ubicación de EC2

Otra configuración

- `instanceRole`: Rol de IAM para instancias de EC2
- `tags`: Etiquetas aplicadas a instancias de EC2

Important

Si modifica alguna configuración de actualización de la infraestructura junto con la configuración de actualización de escala (por ejemplo `desiredvCpus`, `maxvCpus`, o `minvCpus`), AWS Batch realiza una actualización de la infraestructura. Las actualizaciones de la infraestructura tardan más tiempo que las actualizaciones de escalado.

Selección de AMI durante las actualizaciones de la infraestructura

Durante una actualización de infraestructura, el ID de AMI del entorno informático puede cambiar en función de si AMIs se especifica en alguna de estas tres configuraciones. AMIs se especifican en la plantilla `imageId` (`incomputeResources`), `imageIdOverride` (`in ec2Configuration`) o de lanzamiento especificada en `launchTemplate`. Suponga que no IDs se especifica ninguna AMI en ninguna de esas configuraciones y la `updateToLatestImageVersion` configuración es `true`. A continuación, se utiliza la última AMI optimizada para Amazon ECS compatible con cualquier actualización de infraestructura. AWS Batch

Si se especifica una ID de AMI en al menos una de estas configuraciones, la actualización depende de la configuración proporcionada por la ID de AMI utilizada antes de la actualización. Al crear un

entorno de computación, la prioridad a la hora de seleccionar un ID de AMI es primero la plantilla de lanzamiento, después la configuración `imageId` y, por último, la configuración `imageIdOverride`. Sin embargo, si el ID de la AMI que se utiliza proviene de la plantilla de lanzamiento, al actualizar la configuración `imageId` o `imageIdOverride`, no se actualiza el ID de la AMI. La única forma de actualizar un ID de AMI seleccionado en la plantilla de lanzamiento es actualizar la plantilla de lanzamiento. Si el parámetro de versión de la plantilla de lanzamiento es `$Default` o `$Latest`, se evalúa la versión por defecto o la más reciente de la plantilla de lanzamiento especificada. Si se selecciona un ID de AMI diferente de forma predeterminada o se selecciona la última versión de la plantilla de lanzamiento, ese ID de AMI se utiliza en la actualización.

Si la plantilla de lanzamiento no se usó para seleccionar el ID de AMI, se usa el ID de AMI que se especifica en los parámetros `imageId` o `imageIdOverride`. Si se especifican ambos, se utiliza el ID de AMI especificado en el parámetro `imageIdOverride`.

Supongamos que el entorno de computación utiliza un ID de AMI especificado por los parámetros `imageId`, `imageIdOverride`, o `launchTemplate`, y usted desea utilizar la última AMI optimizada para Amazon ECS compatible con AWS Batch. A continuación, la actualización debe eliminar la configuración que proporcionaba la AMI IDs. Para `imageId`, es necesario especificar una cadena vacía para ese parámetro. Para `imageIdOverride`, es necesario especificar una cadena vacía para el parámetro de `ec2Configuration`.

Si el ID de la AMI proviene de la plantilla de lanzamiento, puede cambiarlo AWS Batch por la última AMI optimizada para Amazon ECS que sea compatible con una de las siguientes formas:

- Elimine la plantilla de lanzamiento especificando una cadena vacía para el parámetro `launchTemplateId` o `launchTemplateName`. Esto elimina toda la plantilla de lanzamiento, en lugar de solo el ID de la AMI.
- Si la versión actualizada de la plantilla de lanzamiento no especifica un ID de AMI, el parámetro `updateToLatestImageVersion` debe estar establecido en `true`.

Gestión de trabajos durante las actualizaciones

Configure el modo en que se gestionan los trabajos en ejecución durante una actualización de la infraestructura mediante la política de actualización. Cuando se configura `terminateJobsOnUpdate=true`, los trabajos en ejecución finalizan de inmediato, así como se ignora la configuración de `jobExecutionTimeoutMinutes` y la actualización se realiza en cuanto se puedan reemplazar las instancias. Si configura `terminateJobsOnUpdate=false`, los

trabajos en ejecución continúan durante el periodo de espera especificado, con un tiempo de espera predeterminado de 30 minutos, y los trabajos se cancelan si superan el tiempo de espera.

 Note

Para volver a intentar trabajos finalizados durante una actualización, configure una estrategia de reintento de trabajo. Para obtener más información, consulte [the section called “Reintentos automáticos de trabajo”](#).

Performing infrastructure updates using the Consola de administración de AWS

1. Abra la AWS Batch consola en <https://console.aws.amazon.com/batch/>.
2. En el panel de navegación, seleccione Entornos en la pestaña Entornos de computación.
3. Seleccione el entorno de computación para actualizarlo.
4. Elija Acciones y, a continuación, elija Editar.
5. En la sección Comportamiento de actualización, configure la forma en que se van a gestionar los trabajos en ejecución:
 - Elija Actualizar la AMI a la versión más reciente para actualizar la AMI a la versión más reciente.
 - Elija Finalizar los trabajos inmediatamente al actualizar para finalizar los trabajos cuando se ejecute el proceso de actualización.
 - En Tiempo de espera de ejecución del trabajo, ingrese el número de minutos que se debe esperar antes de iniciar el proceso de actualización.
6. Modifique una o más de las [configuraciones que requieren una actualización de la infraestructura](#). Por ejemplo:
 - Función de instancia
 - Uso de instancias de spot de EC2
 - Tipos de instancias permitidos
 - Grupo de ubicación
 - EC2 key pair (Par de claves de EC2)
 - Configuración de EC2
 - Plantillas de lanzamiento

- Subredes
 - Grupos de seguridad
7. Seleccione Save changes (Guardar cambios).
 8. Supervise el estado del entorno de computación. El entorno se mostrará UPDATING durante el proceso de actualización.

Performing infrastructure updates using the AWS CLI

Utilice el comando `update-compute-environment` para cambiar una o más de las [configuraciones que requieren una actualización de la infraestructura](#). Los siguientes tres ejemplos son operaciones de infraestructura comunes.

- En este ejemplo, se actualizan los tipos de instancias y se configura la política de actualización:

```
aws batch update-compute-environment \  
  --compute-environment your-compute-environment-name \  
  --compute-resources instanceTypes=default_x86_64 \  
  --update-policy terminateJobsOnUpdate=false,jobExecutionTimeoutMinutes=30
```

- En este ejemplo, se actualizan las subredes de la VPC y los grupos de seguridad:

```
aws batch update-compute-environment \  
  --compute-environment your-compute-environment-name \  
  --compute-resources subnets=subnet-abcd1234,subnet-efgh5678 \  
  securityGroupIds=sg-abcd1234 \  
  --update-policy terminateJobsOnUpdate=true
```

- En este ejemplo, se habilitan las actualizaciones automáticas de la última AMI optimizada de Amazon ECS:

```
aws batch update-compute-environment \  
  --compute-environment your-compute-environment-name \  
  --compute-resources updateToLatestImageVersion=true \  
  --update-policy terminateJobsOnUpdate=false,jobExecutionTimeoutMinutes=60
```

Supervisión de las actualizaciones de la infraestructura

Supervise las actualizaciones de la infraestructura mediante la AWS Batch consola para observar cómo cambia el estado del entorno de procesamiento `UPDATING`, supervisar el progreso del reemplazo de las instancias y comprobar si hay algún error en las actualizaciones. La actualización se realiza correctamente una vez que el estado del entorno de computación es `VAILED`. También puede utilizarla CloudWatch para realizar un seguimiento de los eventos de finalización de instancias y supervisar el estado de los trabajos durante la actualización. Con el AWS CLI, utilice el `describe-compute-environments` comando para comprobar el estado y supervisar los eventos del ciclo de vida de la instancia.

Realice blue/green actualizaciones para entornos de cómputo

Una blue/green actualización es una estrategia de actualización que reduce el tiempo de inactividad y el riesgo mediante la creación de un nuevo entorno informático (verde) junto con el entorno informático existente (azul). Este enfoque le permite realizar la transición gradual de las cargas de trabajo al nuevo entorno y, al mismo tiempo, mantener operativo el entorno existente. Blue/green las actualizaciones proporcionan la ruta de actualización más segura y funcionan con cualquier tipo de función de servicio o estrategia de asignación.

Descripción general de

Las actualizaciones azul/verde ofrecen varias ventajas que las hacen ideales para los entornos de producción. Proporcionan un mínimo tiempo de inactividad al mantener sus cargas de trabajo en funcionamiento continuo durante el proceso de actualización. Este enfoque ofrece funciones de reversión sencillas, lo que permite volver rápidamente al entorno original en caso de que surjan problemas. Puede implementar una estrategia de transición progresiva verificando el rendimiento del nuevo entorno antes de cambiar por completo las cargas de trabajo de producción. Este método también proporciona una excelente mitigación de riesgos, ya que el entorno original permanece inalterado y operativo hasta que se decida su eliminación.

Cuando se requieren blue/green actualizaciones

Debe utilizar blue/green las actualizaciones en las siguientes situaciones:

- cuando su entorno de computación utiliza una estrategia de asignación `BEST_FIT` (no admite actualizaciones de infraestructura),
- Cuando su entorno informático no utiliza la función vinculada al `AWSServiceRoleForBatch` servicio

- cuando necesite realizar la transición entre distintos tipos de rol de servicio.

Cuándo se recomiendan blue/green las actualizaciones

Blue/green updates are particularly recommended for production environments where zero downtime is critical for your workloads. This approach works well when you need to test new configurations before transitioning production workloads, ensuring that changes meet your performance and reliability requirements. Choose blue/green updates when the ability to quickly revert is important for your operations, especially if you are updating in a customized AMI with important changes. Additionally, this method is ideal for validating the characteristics and the behavior of the performance before making complete changes, as it provides confidence in the update process.

Requisitos previos

Antes de realizar una blue/green actualización, asegúrese de disponer de lo siguiente:

- [Permisos de IAM](#) adecuados para crear y administrar entornos de computación
- Acceso para ver y modificar la configuración de la cola de trabajos
- Estrategias de reintento de trabajo que estén configuradas para las definiciones de trabajo a fin de gestionar los posibles errores durante la transición. Para obtener más información, consulte [Reintentos automáticos de trabajo](#).
- El ID de AMI para el nuevo entorno de computación. Puede ser:
 - Una versión aprobada y reciente de la AMI optimizada para Amazon ECS (utilizada de manera predeterminada)
 - Una AMI personalizada que cumple con la especificación de la AMI de instancia de contenedor de Amazon ECS. Con una AMI personalizada, puede especificarla de una de las siguientes formas:
 - Uso del campo de Anulación del ID de imagen en la configuración de EC2
 - Especificación de la información en una plantilla de lanzamiento

Para obtener más información sobre la creación de una personalización AMIs, consulte [Tutorial: creación de una AMI de recursos de computación](#).

Antes de crear el nuevo entorno, debe registrar la configuración del entorno de computación existente. Puede hacerlo utilizando el Consola de administración de AWS o el AWS CLI.

 Note

Los siguientes procedimientos detallan cómo realizar una blue/green actualización que solo cambia la AMI. Puede actualizar otros ajustes para el nuevo entorno.

 Important

Al eliminar el entorno de computación anterior (azul), cualquier trabajo que se esté ejecutando ahora en esas instancias fallará porque las instancias se cancelarán. Configure las estrategias de reintento de tareas en las definiciones de tareas para gestionar estos errores de forma automática. Para obtener más información, consulte [Reintentos automáticos de trabajo](#).

Una vez que confíe en el nuevo entorno:

1. Edite la cola de trabajos para eliminar el entorno de computación anterior.
2. Espere a que se complete cualquier trabajo en ejecución en el entorno anterior.
3. Elimine el entorno informático antiguo.

Performing blue/green updates using the Consola de administración de AWS

1. Clone su entorno de computación actual.
 - a. Abra la AWS Batch consola en <https://console.aws.amazon.com/batch/>.
 - b. Seleccione su entorno de computación actual.
 - c. Elija Acciones y, a continuación, elija Clonar.
 - d. En Nombre, ingrese un nombre único para el nuevo entorno de computación.
 - e. Elija Siguiente.
 - f. En la sección Configuración de instancia, actualice la configuración de AMI:
 - i. Expanda Configuración adicional.
 - ii. En Configuración de EC2, especifique el nuevo tipo de AMI en Tipo de imagen y el ID de AMI en el campo Anulación del ID de imagen.
 - g. Elija Siguiente.
 - h. En Configuración de red, seleccione Siguiente.

- i. Revise las demás configuraciones que se copian automáticamente del entorno actual.
 - j. Seleccione Crear entorno de computación.
 - k. Espere a que el estado del nuevo entorno de computación cambie a VALID.
2. Cambie el orden de la cola de trabajos
 - a. En el panel de navegación, elija Colas de trabajos.
 - b. Seleccione la cola de trabajos asociada al entorno de computación actual.
 - c. Elija Edit (Edición de).
 - d. En Entorno de computación conectado, agregue el nuevo entorno de computación:
 - Agregue el nuevo entorno de computación con un número de orden superior al del entorno existente para realizar la transición de la carga de trabajo.
 - Una vez que compruebe que el nuevo entorno funciona correctamente, puede convertirlo en el entorno principal asignándole un número de orden inferior.
 - e. Seleccione Actualizar cola de trabajos.
3. Limpieza
 - a. Supervise la ejecución de los trabajos en el nuevo entorno para asegurarse de que todo funcione según lo esperado.
 - b. Una vez que confíe en el nuevo entorno:
 1. Edite la cola de trabajos para eliminar el entorno de computación anterior.
 2. Espere a que se complete cualquier trabajo en ejecución en el entorno anterior.
 3. Elimine el entorno informático antiguo.

Performing blue/green updates using the AWS CLI

1. Para obtener la configuración mediante el AWS CLI, utilice el siguiente comando:

```
aws batch describe-compute-environments \
  --compute-environments your-compute-environment-name
```

Guarde la salida como referencia al crear el nuevo entorno.

2. Cree un nuevo entorno de computación con la configuración de su entorno existente, pero con la nueva AMI. A continuación, se muestra un ejemplo de la estructura del comando:

Sustituya los valores de ejemplo por la configuración actual del paso anterior:

```
cat <<EOF > ./blue-green-compute-environment.json
{
  "computeEnvironmentName": "your-new-compute-environment-name",
  "type": "MANAGED",
  "state": "ENABLED",
  "computeResources": {
    "instanceRole": "arn:aws:iam::012345678901:instance-profile/
ecsInstanceRole",
    "type": "EC2",
    "minvCpus": 2,
    "desiredvCpus": 2,
    "maxvCpus": 256,
    "instanceTypes": [
      "optimal"
    ],
    "allocationStrategy": "BEST_FIT_PROGRESSIVE",
    "ec2Configuration": [
      {
        "imageType": "ECS_AL2023",
        "imageIdOverride": "ami-0abcdef1234567890"
      }
    ],
    "subnets": [
      "subnet-0abcdef1234567890"
    ],
    "securityGroupIds": [
      "sg-0abcdef1234567890"
    ]
  }
}
EOF
```

```
$ aws batch create-compute-environment --cli-input-json file:///./blue-green-
compute-environment.json
```

3. Espera para que el nuevo entorno esté disponible:

```
aws batch describe-compute-environments \
  --compute-environments your-new-compute-environment-name \
```

```
--query 'computeEnvironments[].status'
```

4. Agregue el entorno de computación a la cola de trabajos:

```
aws batch update-job-queue \  
  --job-queue your-job-queue \  
  --compute-environment-order order=1,computeEnvironment=your-existing-environment \  
  order=2,computeEnvironment=your-new-compute-environment-name
```

5. En cuanto lo verifique, vuelva a actualizarlo para que el nuevo entorno sea principal:

```
aws batch update-job-queue \  
  --job-queue your-job-queue \  
  --compute-environment-order order=1,computeEnvironment=your-new-compute-environment-name
```

Una vez finalizados todos los trabajos en el entorno anterior, desactívelo y elimínelo:

```
aws batch update-compute-environment \  
  --compute-environment your-existing-environment \  
  --state DISABLED
```

```
aws batch delete-compute-environment \  
  --compute-environment your-existing-environment
```

AMI de recursos de computación

De manera predeterminada, los entornos de computación administrados de AWS Batch usan una versión aprobada y reciente de la AMI optimizada para Amazon ECS para los recursos de computación. Sin embargo, es posible que desee crear sus propias AMI; para utilizarlas en sus entornos de computación administrados y no administrados. Si necesita alguna de las siguientes opciones, le recomendamos que cree su propia AMI:

- Aumentar el tamaño de su almacenamiento AMI; raíz o volúmenes de datos
- Agregar volúmenes de almacenamiento de instancias para los tipos de instancias de Amazon EC2; admitidos
- Personalizar el agente de contenedor de Amazon ECS

- Personalización de Docker
- Configurar una AMI de carga de trabajo de GPU que permite a los contenedores acceder a hardware de GPU en los tipos de instancias de Amazon EC2 admitidos

Note

Una vez que se ha creado un entorno de computación, AWS Batch no actualiza las AMI del entorno de computación. AWS Batch tampoco actualiza las AMI de su entorno de computación cuando hay disponible una versión más reciente de la AMI optimizada de Amazon ECS. Usted es responsable de la administración del sistema operativo invitado. Esto incluye actualizaciones y parches de seguridad. También es el responsable de cualquier otra utilidad o software de aplicaciones que se instale en los recursos de computación. A fin de utilizar una AMI nueva para sus trabajos de AWS Batch, siga los siguientes pasos:

1. Cree un nuevo entorno de computación con la nueva AMI.
2. Añada el entorno de computación a una cola de trabajos existente.
3. Quite el entorno de computación anterior de la cola de trabajos.
4. Elimine el entorno de computación anterior.

En abril de 2022, AWS Batch agregó compatibilidad mejorado para actualizar los entornos de computación. Para obtener más información, consulte [Actualizar un entorno informático en AWS Batch](#). Para usar la actualización mejorada de los entornos de computación para actualizar las AMI, siga estas reglas:

- No configure el parámetro de rol de servicio ([serviceRole](#)) o establézcalo en el rol `AWSServiceRoleForBatch` vinculado al servicio.
- Defina el parámetro de estrategia de asignación ([allocationStrategy](#)) en `BEST_FIT_PROGRESSIVE`, `SPOT_CAPACITY_OPTIMIZED` o `SPOT_PRICE_CAPACITY_OPTIMIZED`.
- Defina el parámetro de actualización a la última versión de la imagen ([updateToLatestImageVersion](#)) en `true`.
- No especifique un ID de AMI en [imageId](#), [imageIdOverride](#) (en [ec2Configuration](#)) o en la plantilla de lanzamiento ([launchTemplate](#)). Si no especifica un ID de AMI, AWS Batch selecciona la última AMI optimizada de Amazon ECS compatible con AWS Batch en el momento en que se inicia la actualización de la infraestructura. También,

puede especificar el ID de AMI en los parámetros `imageId` o `imageIdOverride`. También puede especificar la plantilla de lanzamiento que se identifica mediante las propiedades de `LaunchTemplate`. El cambio de cualquiera de estas propiedades inicia una actualización de la infraestructura. Si el ID de AMI se especifica en la plantilla de lanzamiento, este se puede reemplazar especificando un ID de AMI en los parámetros `imageId` o `imageIdOverride`. El ID de AMI solo se puede reemplazar especificando una plantilla de lanzamiento diferente. Si la versión de la plantilla de lanzamiento está configurada en `$Default` o `$Latest`, el ID de AMI puede sustituirse configurando una nueva versión predeterminada para la plantilla de lanzamiento (si es `$Default`) o añadiendo una nueva versión a la plantilla de lanzamiento (si es `$Latest`).

Si se siguen estas reglas, cualquier actualización que inicie una actualización de la infraestructura hará que se vuelva a seleccionar el ID de la AMI. Si la configuración [version](#) en la plantilla de lanzamiento ([launchTemplate](#)) toma el valor `$Latest` o `$Default`, se evaluará la versión más reciente o predeterminada de la plantilla de lanzamiento en el momento de la actualización de la infraestructura, incluso si [launchTemplate](#) no se ha actualizado.

Temas

- [Especificaciones de AMI de recursos de computación](#)
- [Tutorial: creación de una AMI de recursos de computación](#)
- [Uso de una AMI de carga de trabajo de GPU](#)
- [Obsolescencia de Amazon Linux](#)
- [Obsolescencia de la AMI de Amazon Linux 2 en Amazon EKS](#)
- [Obsolescencia de la AMI de Amazon Linux 2 en Amazon ECS](#)

Especificaciones de AMI de recursos de computación

La especificación de AWS Batch de recursos de computación de AMI consta de lo siguiente:

Obligatorio

- Una moderna distribución de Linux que ejecuta al menos la versión 3,10 de kernel de Linux en una AMI; de tipo de virtualización HVM. No se admiten los contenedores de Windows.

 Important

Los trabajos paralelos de varios nodos solo se pueden ejecutar en recursos de computación que se hayan lanzado en una instancia de Amazon Linux con el paquete `ecs-init` instalado. Recomendamos utilizar la AMI optimizada para Amazon ECS predeterminada al crear el entorno de computación. Para ello, no especifique una AMI personalizada. Para obtener más información, consulte [Trabajos paralelos de varios nodos](#).

- El agente de contenedor de Amazon ECS. Le recomendamos que utilice la última versión de . Para obtener más información, consulte [Instalar el agente contenedor de instancia de Amazon ECS](#) en la Guía para desarrolladores de Amazon Elastic Container Service.
- El controlador de registros `awslogs` debe especificarse como un controlador de registros disponible con la variable de entorno `ECS_AVAILABLE_LOGGING_DRIVERS` cuando el agente de contenedor de Amazon ECS se inicia. Para obtener más información, consulte [Configuración del agente de contenedor de Amazon ECS](#) en la Guía para desarrolladores de Amazon Elastic Container Service.
- Un daemon de Docker que ejecuta al menos la versión 1.9, y cualquier dependencia de tiempo de ejecución de Docker. Para obtener más información, consulte [Check runtime dependencies](#) en la documentación de Docker.

 Note

Le recomendamos la versión de Docker que se envía y se prueba con la versión de agente de Amazon ECS correspondiente que está utilizando. Amazon ECS proporciona un registro de cambios para la variante Linux de la AMI optimizada para Amazon ECS en GitHub. Para obtener más información, consulte [Changelog](#) (Registro de cambios).

Recomendado

- Un proceso de inicialización y nanny para ejecutar y monitorear el agente de Amazon ECS. La AMI optimizada para Amazon ECS usa el proceso `upstart ecs-init` y otros sistemas operativos pueden utilizar `systemd`. Para obtener más información y ejemplos, consulte [Scripts de configuración de datos de usuario de instancias de contenedor de ejemplo](#) en la Guía para desarrolladores de Amazon Elastic Container Service. Para obtener más información sobre `ecs-`

`init`, consulte el [proyecto ecs-init](#) en GitHub. Los entornos de computación administrados requieren, como mínimo, que el agente de Amazon ECS se inicie al arrancar. Si el agente de Amazon ECS no se ejecuta en el recurso de computación, no puede aceptar trabajos desde AWS Batch.

Las AMI optimizadas para Amazon ECS están preconfiguradas con estos requisitos y recomendaciones. Recomendamos que utilice la AMI optimizada para Amazon ECS o una AMI de Amazon Linux con el paquete `ecs-init` instalado para sus recursos de computación. Elija otra AMI si la aplicación requiere un sistema operativo específico o una versión de Docker que aún no esté disponible en estas AMI. Para obtener más información, consulte la [AMI optimizada para Amazon ECS](#) en la Guía para desarrolladores de Amazon Elastic Container Service.

Tutorial: creación de una AMI de recursos de computación

Es posible crear sus propias AMI de recursos de computación personalizadas para usarlas en entornos de computación administrados y sin administrar. Para obtener instrucciones, consulte la [Especificaciones de AMI de recursos de computación](#). Después de crear la AMI personalizada, puede crear un entorno de computación que utilice dicha AMI al que puede asociar una cola de trabajos. Por último, comience a enviar trabajos a esa cola.

Para crear una AMI de recursos de computación personalizada

1. Elija una AMI de base como punto de partida. Las AMI de base deben utilizar virtualización HVM. Las AMI de base no pueden ser una AMI de Windows.

Note

La AMI que elija para un entorno informático debe coincidir con la arquitectura de los tipos de instancias que tenga previsto utilizar para dicho entorno informático. Por ejemplo, si su entorno informático utiliza tipos de instancias A1, la AMI de recursos informáticos que elija debe admitir instancias ARM. Amazon ECS ofrece versiones x86 y ARM de la AMI Amazon Linux 2 optimizada para Amazon ECS. Para obtener más información, consulte la sección sobre [AMI Amazon Linux 2 optimizada para Amazon ECS](#) en la Guía para desarrolladores de Amazon Elastic Container Service.

La AMI Amazon Linux 2 optimizada para Amazon ECS es la predeterminada para recursos de computación en entornos de computación administrados. La AMI Amazon Linux 2 optimizada

para Amazon ECS viene preconfigurada y probada AWS Batch por AWS ingenieros. Se trata de una AMI mínima con la que puede empezar y que permite que sus recursos informáticos se ejecuten AWS rápidamente. Para obtener más información, consulte [AMI Linux optimizada para Amazon ECS](#) en la Guía para desarrolladores de Amazon Elastic Container Service.

También puede elegir otra variante de Amazon Linux 2 e instalar el paquete `ecs-init` con los siguientes comandos. Para obtener más información, consulte [Instalación del agente contenedor Amazon ECS en una EC2 instancia de Amazon Linux 2](#) en la Guía para desarrolladores de Amazon Elastic Container Service:

```
$ sudo amazon-linux-extras disable docker
$ sudo amazon-linux-extras install ecs-init
```

Por ejemplo, si desea ejecutar cargas de trabajo de GPU en sus recursos AWS Batch informáticos, puede empezar con la [AMI de aprendizaje profundo de Amazon Linux](#). A continuación, configure la AMI para ejecutar los AWS Batch trabajos. Para obtener más información, consulte [Uso de una AMI de carga de trabajo de GPU](#).

Important

Puede elegir una AMI básica que no sea compatible con el paquete `ecs-init`. Sin embargo, si lo hace, debe configurar una forma de iniciar el agente Amazon ECS durante el arranque y mantenerlo en funcionamiento. También puede ver varios ejemplos de scripts de configuración de datos de usuario que utilizan `systemd` para iniciar y supervisar el agente contenedor de Amazon ECS. Para obtener más información, consulte [Scripts de configuración de datos de usuario de instancias de contenedor de ejemplo](#) en la Guía para desarrolladores de Amazon Elastic Container Service.

2. Lance una instancia desde su AMI de base seleccionada con las opciones de almacenamiento adecuadas para su AMI. Es posible configurar el tamaño y la cantidad de volúmenes de Amazon EBS o de almacenamiento de instancias, si el tipo de instancia que ha seleccionado es compatible con ellos. Para obtener más información, consulte [Lanzamiento de una instancia y Amazon EC2 Instance Store](#) en la Guía del EC2 usuario de Amazon.
3. Conéctese a la instancia con SSH y lleve a cabo las tareas de configuración que sean necesarias. Esto puede incluir una de las siguientes etapas, o todas:

- Instalación del agente de contenedor de Amazon ECS. Para obtener más información, consulte [Instalar el agente contenedor de instancia de Amazon ECS](#) en la Guía para desarrolladores de Amazon Elastic Container Service.
- Configurar un script para formatear volúmenes de almacén de instancias.
- Añadir un volumen de almacén de instancias o sistemas de archivos de Amazon EFS al archivo `/etc/fstab` para que puedan montarse al arrancar.
- Configurar opciones de Docker, como habilitar la depuración o ajustar el tamaño de la imagen de base.
- Instalar paquetes o copiar archivos.

Para obtener más información, consulte [Conexión a su instancia de Linux mediante SSH](#) en la Guía del EC2 usuario de Amazon.

4. Si ha iniciado el agente contenedor de Amazon ECS en su instancia, debe detenerlo y eliminar cualquier archivo de comprobación de datos persistente antes de crear la AMI. De lo contrario, si no lo hace, el agente no se iniciará en las instancias que se lancen desde su AMI.

- a. Detenga el agente de contenedor de Amazon ECS.

- AMI de Amazon Linux 2 optimizada para Amazon ECS:

```
sudo systemctl stop ecs
```

- AMI de Amazon Linux optimizada para Amazon ECS:

```
sudo stop ecs
```

- b. Elimine los archivos de comprobación de datos persistentes. De forma predeterminada, estos archivos se ubican en el directorio `/var/lib/ecs/data/`. Use el siguiente comando para eliminar estos archivos, si los hay.

```
sudo rm -rf /var/lib/ecs/data/*
```

5. Cree una nueva AMI; desde su instancia en ejecución. Para obtener más información, consulte [Creación de una AMI de Linux respaldada por Amazon EBS](#) en la guía del EC2 usuario de Amazon.

Para usar su nueva AMI con AWS Batch

1. Cuando se haya creado una nueva AMI, cree un nuevo entorno de computación con la nueva AMI. Para ello, elija el tipo de imagen e introduzca el ID de AMI personalizado en el cuadro de anulación del ID de imagen al crear el entorno AWS Batch informático. Para obtener más información, consulte [the section called “Tutorial: creación de un entorno de computación gestionado con los recursos de Amazon EC2”](#).

Note

La AMI que elija para un entorno informático debe coincidir con la arquitectura de los tipos de instancias que tenga previsto utilizar para dicho entorno informático. Por ejemplo, si su entorno informático utiliza tipos de instancias A1, la AMI de recursos informáticos que elija debe admitir instancias ARM. Amazon ECS ofrece versiones x86 y ARM de la AMI Amazon Linux 2 optimizada para Amazon ECS. Para obtener más información, consulte la sección sobre [AMI Amazon Linux 2 optimizada para Amazon ECS](#) en la Guía para desarrolladores de Amazon Elastic Container Service.

2. Cree una cola de trabajos y asocie el entorno de computación nuevo. Para obtener más información, consulte [Creación de una cola de trabajos](#).

Note

Todos los entornos de computación asociados a una cola de trabajos deben compartir la misma arquitectura. AWS Batch no admite la mezcla de tipos de arquitecturas de entorno de computación en una sola cola de trabajos.

3. (Opcional) Envíe un trabajo de muestra a la cola de trabajos nueva. Para obtener más información, consulte [Ejemplos de definición de trabajo](#), [Creación de una definición de trabajo de un solo nodo](#) y [Tutorial: envío de un trabajo](#).

Uso de una AMI de carga de trabajo de GPU

Para ejecutar cargas de trabajo de GPU en los recursos de computación de AWS Batch, debe utilizar una AMI compatible con GPU. Para obtener más información, consulte [Trabajar con GPU en Amazon ECS](#) y [AMI optimizadas para Amazon ECS](#) en la Guía para desarrolladores de Amazon Elastic Container Service.

En entornos de computación administrados, si el entorno de computación especifica cualquier familia de instancia o tipo de instancia de p3, p4, p5, p6, g3, g3s, g4, g5 o g6, entonces AWS Batch utiliza una AMI optimizada para GPU de Amazon ECS.

En entornos de computación no administrados, se recomienda una AMI optimizada para GPU de Amazon ECS. Puede utilizar las operaciones AWS Command Line Interface o AWS Systems Manager del almacén de parámetros [GetParameter](#), [GetParameters](#) y [GetParametersByPath](#) para recuperar los metadatos de las AMI optimizadas para GPU de Amazon ECS recomendadas.

Note

La familia de instancias p5 solo se admite en versiones iguales o posteriores a la AMI optimizada para GPU de Amazon ECS 20230912 y son incompatibles los tipos de instancia p2 y g2. Si necesita usar instancias p5, asegúrese de que su entorno de computación no instancias p2 o g2 y utilice la última AMI de Batch predeterminada. La creación de un nuevo entorno de computación utilizará la AMI más reciente, pero si actualiza su entorno de computación para incluir p5, puede asegurarse de utilizar la AMI más reciente configurando [updateToLatestImageVersion](#) en true en propiedades ComputeResource. Para obtener más información sobre la compatibilidad de AMI con instancias de [GPU](#), consulte [Working with GPUs on Amazon ECS](#) en Amazon Elastic Container Service Developer Guide.

Los siguientes ejemplos muestran cómo utilizar el comando [GetParameter](#).

AWS CLI

```
$ aws ssm get-parameter --name /aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended \
                        --region us-east-2 --output json
```

El resultado incluye la información de AMI en el parámetro de valor Value.

```
{
  "Parameter": {
    "Name": "/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended",
    "LastModifiedDate": 1555434128.664,
    "Value": "{\"schema_version\":1,\"image_name\": \"amzn2-ami-ecs-gpu-hvm-2.0.20190402-x86_64-eb\", \"image_id\": \"ami-083c800fe4211192f\", \"os\": \"Amazon Linux 2\", \"ecs_runtime_version\": \"Docker version 18.06.1-ce\", \"ecs_agent_version\": \"1.27.0\"}"
```

```

    "Version": 9,
    "Type": "String",
    "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ecs/optimized-ami/
amazon-linux-2/gpu/recommended"
  }
}

```

Python

```

from __future__ import print_function

import json
import boto3

ssm = boto3.client('ssm', 'us-east-2')

response = ssm.get_parameter(Name='/aws/service/ecs/optimized-ami/amazon-linux-2/
gpu/recommended')
jsonVal = json.loads(response['Parameter']['Value'])
print("image_id  = " + jsonVal['image_id'])
print("image_name = " + jsonVal['image_name'])

```

El resultado solo incluye el ID y el nombre de AMI:

```

image_id  = ami-083c800fe4211192f
image_name = amzn2-ami-ecs-gpu-hvm-2.0.20190402-x86_64-ebs

```

En los siguientes ejemplos se muestra el uso de [GetParameters](#).

AWS CLI

```

$ aws ssm get-parameters --names /aws/service/ecs/optimized-ami/amazon-linux-2/gpu/
recommended/image_name \
                               /aws/service/ecs/optimized-ami/amazon-linux-2/gpu/
recommended/image_id \
                               --region us-east-2 --output json

```

El resultado incluye todos los metadatos de cada uno de los parámetros:

```

{

```

```

    "InvalidParameters": [],
    "Parameters": [
        {
            "Name": "/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/
image_id",
            "LastModifiedDate": 1555434128.749,
            "Value": "ami-083c800fe4211192f",
            "Version": 9,
            "Type": "String",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ecs/optimized-ami/
amazon-linux-2/gpu/recommended/image_id"
        },
        {
            "Name": "/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/
image_name",
            "LastModifiedDate": 1555434128.712,
            "Value": "amzn2-ami-ecs-gpu-hvm-2.0.20190402-x86_64-ebs",
            "Version": 9,
            "Type": "String",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ecs/optimized-ami/
amazon-linux-2/gpu/recommended/image_name"
        }
    ]
}

```

Python

```

from __future__ import print_function

import boto3

ssm = boto3.client('ssm', 'us-east-2')

response = ssm.get_parameters(
    Names=['/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/
image_name',
          '/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/
image_id'])
for parameter in response['Parameters']:
    print(parameter['Name'] + " = " + parameter['Value'])

```

El resultado incluye el ID de AMI y el nombre de AMI, con la ruta completa de los nombres.

```
/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/image_id =  
ami-083c800fe4211192f  
/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/image_name = amzn2-  
ami-ecs-gpu-hvm-2.0.20190402-x86_64-ebs
```

Los siguientes ejemplos muestran cómo utilizar el comando [GetParametersByPath](#).

AWS CLI

```
$ aws ssm get-parameters-by-path --path /aws/service/ecs/optimized-ami/amazon-  
linux-2/gpu/recommended \  
--region us-east-2 --output json
```

El resultado incluye todos los metadatos de todos los parámetros de la ruta especificada.

```
{  
  "Parameters": [  
    {  
      "Name": "/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/  
ecs_agent_version",  
      "LastModifiedDate": 1555434128.801,  
      "Value": "1.27.0",  
      "Version": 8,  
      "Type": "String",  
      "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ecs/optimized-ami/  
amazon-linux-2/gpu/recommended/ecs_agent_version"  
    },  
    {  
      "Name": "/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/  
ecs_runtime_version",  
      "LastModifiedDate": 1548368308.213,  
      "Value": "Docker version 18.06.1-ce",  
      "Version": 1,  
      "Type": "String",  
      "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ecs/optimized-ami/  
amazon-linux-2/gpu/recommended/ecs_runtime_version"  
    },  
    {  
      "Name": "/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/  
image_id",  
      "LastModifiedDate": 1555434128.749,
```

```

        "Value": "ami-083c800fe4211192f",
        "Version": 9,
        "Type": "String",
        "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ecs/optimized-ami/
amazon-linux-2/gpu/recommended/image_id"
    },
    {
        "Name": "/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/
image_name",
        "LastModifiedDate": 1555434128.712,
        "Value": "amzn2-ami-ecs-gpu-hvm-2.0.20190402-x86_64-ebs",
        "Version": 9,
        "Type": "String",
        "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ecs/optimized-ami/
amazon-linux-2/gpu/recommended/image_name"
    },
    {
        "Name": "/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/
os",
        "LastModifiedDate": 1548368308.143,
        "Value": "Amazon Linux 2",
        "Version": 1,
        "Type": "String",
        "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ecs/optimized-ami/
amazon-linux-2/gpu/recommended/os"
    },
    {
        "Name": "/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/
schema_version",
        "LastModifiedDate": 1548368307.914,
        "Value": "1",
        "Version": 1,
        "Type": "String",
        "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ecs/optimized-ami/
amazon-linux-2/gpu/recommended/schema_version"
    }
]
}

```

Python

```
from __future__ import print_function
```

```
import boto3

ssm = boto3.client('ssm', 'us-east-2')

response = ssm.get_parameters_by_path(Path='/aws/service/ecs/optimized-ami/amazon-
linux-2/gpu/recommended')
for parameter in response['Parameters']:
    print(parameter['Name'] + " = " + parameter['Value'])
```

El resultado incluye los valores de todos los nombres de parámetros de la ruta especificada, con la ruta completa de los nombres.

```
/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/ecs_agent_version =
1.27.0
/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/ecs_runtime_version =
Docker version 18.06.1-ce
/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/image_id =
ami-083c800fe4211192f
/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/image_name = amzn2-
ami-ecs-gpu-hvm-2.0.20190402-x86_64-eb6
/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/os = Amazon Linux 2
/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/schema_version = 1
```

Para obtener más información, consulte [Retrieving Amazon ECS-Optimized AMI Metadata](#) en la Guía para desarrolladores de Amazon Elastic Container Service.

Obsolescencia de Amazon Linux

La AMI de Amazon Linux (también llamada Amazon Linux 1) llegó al final de su vida útil el 31 de diciembre de 2023. AWS Batch ya no es compatible con Amazon Linux AMI, por lo que no recibe actualizaciones de seguridad ni correcciones de errores desde el 1 de enero de 2024. Para obtener más información, sobre el fin de la vida útil de la AMI de Amazon Linux, consulte [Preguntas frecuentes de AL](#).

Le recomendamos que actualice los entornos de computación actuales basados en Amazon Linux a Amazon Linux 2023 para evitar interrupciones imprevistas de la carga de trabajo y seguir recibiendo actualizaciones de seguridad u otro tipo.

Es posible que los entornos de computación que utilizan la AMI de Amazon Linux sigan funcionando después de la fecha de fin de vida útil (31 de diciembre de 2023). Sin embargo, estos entornos

de computación ya no recibirán nuevas actualizaciones de software, parches de seguridad ni correcciones de errores de AWS. Es su responsabilidad mantener estos entornos de computación en la AMI de Amazon Linux una vez finalizada su vida útil. Le recomendamos migrar los entornos de computación de AWS Batch a Amazon Linux 2023 o Amazon Linux 2 para mantener un rendimiento y una seguridad óptimos.

Para obtener ayuda al migrar AWS Batch de la AMI de Amazon Linux a Amazon Linux 2023 o Amazon Linux 2, consulte [Actualización de entornos de computación - AWS Batch](#).

Obsolescencia de la AMI de Amazon Linux 2 en Amazon EKS

Desde el 26/11/25, AWS finalizará el soporte para las AMI de Amazon Linux 2 optimizadas para Amazon EKS. Recomendamos migrar los entornos de computación de Amazon EKS de AWS Batch a Amazon Linux 2023 antes del 26/11/25 para mantener un rendimiento y una seguridad óptimos.

Si bien puede seguir utilizando las AMI Amazon Linux 2 optimizadas para Amazon EKS proporcionadas por Batch en sus entornos de computación de Amazon EKS después de la fecha de fin del soporte del 26/11/25, estos entornos de computación ya no recibirán nuevas actualizaciones de software, parches de seguridad o correcciones de errores de parte de AWS. Es su responsabilidad mantener estos entornos de computación en la AMI de Amazon Linux 2 optimizada para Amazon EKS una vez finalizada su vida útil.

Para obtener más información sobre el fin de la vida útil de AL2 en Amazon EKS, consulte las [preguntas frecuentes sobre la obsolescencia de las AMI de Amazon EKS](#) en la Guía del usuario de Amazon EKS.

Para obtener ayuda sobre la migración de entornos de computación AWS Batch Amazon EKS de Amazon Linux 2 to Amazon Linux 2023, consulte [¿Cómo actualizar de EKS AL2 a EKS AL2023](#).

Obsolescencia de la AMI de Amazon Linux 2 en Amazon ECS

AWS finalizará su compatibilidad con Amazon Linux 2. A partir de enero de 2026, AWS Batch se cambiará la AMI predeterminada para los nuevos entornos de computación de Amazon ECS de Amazon Linux 2 a Amazon Linux 2023. Le recomendamos migrar los entornos de computación de Amazon ECS de AWS Batch a Amazon Linux 2023 para mantener un rendimiento y una seguridad óptimos.

Para obtener más información sobre el fin de la vida útil de Amazon Linux 2, consulte [Preguntas frecuentes de Amazon Linux 2](#).

Para obtener información sobre las diferencias entre Amazon Linux 2 y Amazon Linux 2023, consulte [Comparación entre Amazon Linux 2 y Amazon Linux 2023](#) en la Guía del usuario de Amazon Linux 2023.

Para obtener información sobre los cambios de Amazon Linux 2023 con la AMI optimizada para Amazon ECS, consulte [Migración de una AMI de Amazon Linux 2 a una AMI de Amazon Linux 2023 optimizada para Amazon ECS](#) en la Guía de usuario de Amazon ECS.

Para obtener ayuda sobre la migración de entornos de computación AWS Batch Amazon ECS de Amazon Linux 2 to Amazon Linux 2023, consulte [Cómo migrar de ECS AL2 a ECS AL2023](#).

Utilice las plantillas de lanzamiento de Amazon EC2 con AWS Batch

AWS Batch admite el uso de plantillas de lanzamiento de Amazon EC2 con sus entornos informáticos de EC2. Con las plantillas de lanzamiento, puede modificar la configuración predeterminada de sus recursos AWS Batch informáticos sin necesidad de crear plantillas personalizadas. AMIs

Note

Los recursos de AWS Fargate no admiten plantillas de lanzamiento.

Es necesario crear una plantilla de lanzamiento para poder asociarla a un entorno de computación. Puede crear una plantilla de lanzamiento en la consola de Amazon EC2. O bien, puede usar el AWS CLI o un AWS SDK. Por ejemplo, el siguiente archivo JSON representa una plantilla de lanzamiento que cambia el tamaño del volumen de datos de Docker para la AMI de AWS Batch recursos informáticos predeterminada y también lo configura para que esté cifrado.

```
{
  "LaunchTemplateName": "increase-container-volume-encrypt",
  "LaunchTemplateData": {
    "BlockDeviceMappings": [
      {
        "DeviceName": "/dev/xvda",
        "Ebs": {
          "Encrypted": true,
```

```
        "VolumeSize": 100,  
        "VolumeType": "gp2"  
      }  
    ]  
  }  
}
```

Para crear la plantilla de lanzamiento anterior, guarda el JSON en un archivo llamado `lt-data.json` y ejecuta el siguiente AWS CLI comando.

```
aws ec2 --region <region> create-launch-template --cli-input-json file://lt-data.json
```

Para obtener más información sobre las plantillas de lanzamiento, consulte [Lanzar una instancia desde una plantilla de lanzamiento](#) en la Guía del usuario de Amazon EC2.

Si utiliza una plantilla de lanzamiento para crear el entorno de computación, puede mover los siguientes parámetros del entorno de computación existente a la plantilla de lanzamiento:

Note

Supongamos que alguno de estos parámetros (excepto las etiquetas de Amazon EC2) se especifica tanto en la plantilla de lanzamiento como en la configuración del entorno de procesamiento. Entonces, los parámetros del entorno de cómputo tienen prioridad. Las etiquetas Amazon EC2 se combinan entre la plantilla de lanzamiento y la configuración del entorno de cómputo. Si se produce una colisión en la clave de la etiqueta, el valor de la configuración del entorno de computación tiene prioridad.

- Pares de claves de Amazon EC2
- ID de la AMI de Amazon EC2
- Grupo de seguridad IDs
- Etiquetas de Amazon EC2

Los siguientes parámetros de la plantilla de lanzamiento son ignorados por AWS Batch:

- Tipo de instancia (especifique los tipos de instancia que desea utilizar al crear el entorno de computación)

- Rol de instancia (especifique el rol de instancia que desea utilizar al crear el entorno de computación)
- Subredes de interfaz de red (especifique las subredes que desea utilizar al crear el entorno de computación)
- Opciones del mercado de instancias (AWS Batch debe controlar la configuración de las instancias puntuales)
- Inhabilite la terminación de la API (AWS Batch debe controlar el ciclo de vida de la instancia)

AWS Batch solo actualiza la plantilla de lanzamiento con una nueva versión de la plantilla de lanzamiento durante las actualizaciones de infraestructura. Para obtener más información, consulte [Actualizar un entorno informático en AWS Batch](#).

Plantillas de lanzamiento predeterminadas y anuladas

Puede definir una plantilla de lanzamiento predeterminada para el entorno de computación y una plantilla de lanzamiento de anulación para los tipos y las familias de instancias específicos. Esto puede resultar útil para que la plantilla predeterminada se utilice en la mayoría de los tipos de instancias de los entornos de computación.

Las variables de sustitución `$Default` y `$Latest` se pueden utilizar en lugar de asignar un nombre a una versión específica. Si no proporciona una plantilla de lanzamiento de anulación, se aplicará automáticamente la predeterminada.

Si utiliza la `$Latest` variable `$Default` o, AWS Batch aplicará la información actual en el momento en que se cree el entorno informático. Si la versión predeterminada o la última versión cambian en el futuro, debe actualizar la información mediante [UpdateComputeEnvironment](#) mediante el botón Consola de administración de AWS - AWS Batch.

Para brindar una flexibilidad adicional, puede definir plantillas de lanzamiento de anulación que se apliquen a tipos o familias de instancias de computación específicos.

Note

Puede especificar hasta diez (10) anulaciones de plantillas de lanzamiento para cada entorno de computación.

Utilice el parámetro `targetInstanceTypes` para seleccionar el tipo de instancia o la familia que se debe usar en esta anulación de plantilla de lanzamiento. El parámetro [instanceTypes](#) debe identificar primero el tipo o la familia de instancias.

Si se definen las anulaciones de plantillas de lanzamiento y se decide eliminarlas más adelante, puede pasar una matriz vacía para anular la configuración del parámetro [overrides](#) en la operación de la API [UpdateComputeEnvironment](#). También puede optar por no incluir el parámetro `overrides` al enviar la operación de la API `UpdateComputeEnvironment`. Para obtener más información, consulte: [LaunchTemplateSpecification.overrides](#)

Para obtener más información, consulte

[LaunchTemplateSpecificationOverride.targetInstanceTypes](#) la guía de referencia de la AWS Batch API.

Datos de usuario de Amazon EC2 en las plantillas de lanzamiento

Puede proporcionar datos de usuario de Amazon EC2 en la plantilla de lanzamiento que será ejecutada por [cloud-init](#) al lanzar las instancias. Los datos de usuario pueden realizar escenarios de configuración comunes, incluidos pero sin limitarse a los siguientes:

- [Inclusión de usuarios o grupos](#)
- [Instalación de paquetes](#)
- [Creación de particiones y sistemas de archivos](#)

Los datos de usuario de Amazon EC2 en las plantillas de lanzamiento deben estar en el formato [archivo multiparte MIME](#). Esto se debe a que los datos de AWS Batch usuario se combinan con otros datos de usuario necesarios para configurar los recursos informáticos. Puede combinar varios bloques de datos de usuario en un único archivo multiparte MIME. Por ejemplo, es posible que desee combinar un boothook de nube que configure el daemon de Docker con un script de shell de datos de usuario que escribe información de configuración para el agente de contenedor de Amazon ECS.

Si los usas AWS CloudFormation, el [AWS::CloudFormation::Init](#) tipo se puede usar con el script de ayuda [cfn-init](#) para realizar escenarios de configuración comunes.

Un archivo multiparte MIME consta de los siguientes componentes:

- El tipo de contenido y declaración de límite de partes: `Content-Type: multipart/mixed; boundary=="==BOUNDARY=="`
- La declaración de versión de MIME: `MIME-Version: 1.0`

- Uno o más bloques de datos de usuario, que contienen los siguientes componentes:
 - El límite de apertura, que señala el inicio de un bloque de datos de usuario: `--==BOUNDARY==`. Debe dejar en blanco la línea anterior a este límite.
 - La declaración de tipo de contenido para el bloque: `Content-Type: text/cloud-config; charset="us-ascii"`. Para obtener más información sobre los tipos de contenido, consulte la [documentación de Cloud-Init](#). Debe dejar en blanco la línea que sigue a la declaración de tipo de contenido.
 - El contenido de los datos de usuario, por ejemplo, una lista de shell o políticas de `cloud-init`.
- El límite de cierre, que señala el final del archivo multiparte MIME: `--==BOUNDARY==--`. Debe dejar en blanco la línea anterior al límite de cierre.

Note

Si añade datos de usuario a una plantilla de lanzamiento en la consola de Amazon EC2, puede pegarlos como texto sin formato. O bien, puede cargarlos desde un archivo. Si utilizas el SDK AWS CLI o un AWS SDK, primero debes base64 codificar los datos del usuario y enviar esa cadena como el valor del `UserData` parámetro cuando llames [CreateLaunchTemplate](#), como se muestra en este archivo JSON.

```
{
  "LaunchTemplateName": "base64-user-data",
  "LaunchTemplateData": {
    "UserData":
      ""ewogICAgIkxhdW5jaFRlbXBsYXRlTmFtZSI6ICJpbmNyZWZzZS1jb250YWluZXItZm9sdW...""
  }
}
```

Temas

- [Referencia: ejemplos de plantillas de lanzamiento de Amazon EC2](#)

Referencia: ejemplos de plantillas de lanzamiento de Amazon EC2

A continuación, se muestra un ejemplo de un archivo multiparte MIME que puede utilizar para crear sus propias plantillas.

Ejemplos

- [Ejemplo: montaje de un sistema de archivos de Amazon EFS existente](#)
- [Ejemplo: anulación de la configuración predeterminada del agente de contenedor de Amazon ECS](#)
- [Ejemplo: montar un sistema de archivos Amazon FSx for Lustre existente](#)

Ejemplo: montaje de un sistema de archivos de Amazon EFS existente

Example

Este archivo multiparte MIME de ejemplo configura el recurso de computación para instalar el paquete `amazon-efs-utils` y montar un sistema de archivos de Amazon EFS existente en `/mnt/efs`.

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary=="MYBOUNDARY=="

--MYBOUNDARY--
Content-Type: text/cloud-config; charset="us-ascii"

packages:
- amazon-efs-utils

runcmd:
- file_system_id_01=fs-abcdef123
- efs_directory=/mnt/efs

- mkdir -p ${efs_directory}
- echo "${file_system_id_01}:/ ${efs_directory} efs tls,_netdev" >> /etc/fstab
- mount -a -t efs defaults

--MYBOUNDARY--
```

Ejemplo: anulación de la configuración predeterminada del agente de contenedor de Amazon ECS

Example

Este archivo multiparte MIME de ejemplo anula la configuración de limpieza predeterminada de las imágenes de Docker de un recurso de computación.

```

MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="==MYBOUNDARY=="

--==MYBOUNDARY==
Content-Type: text/x-shellscript; charset="us-ascii"

#!/bin/bash
echo ECS_IMAGE_CLEANUP_INTERVAL=60m >> /etc/ecs/ecs.config
echo ECS_IMAGE_MINIMUM_CLEANUP_AGE=60m >> /etc/ecs/ecs.config

--==MYBOUNDARY==

```

Ejemplo: montar un sistema de archivos Amazon FSx for Lustre existente

Example

Este ejemplo de archivo MIME de varias partes configura el recurso informático para instalar el `lustre2.10` paquete de la biblioteca Extras y montar un sistema de archivos existente FSx para Lustre en `/scratch` y con el nombre de montaje de `fsx`. Este ejemplo es para Amazon Linux 2. Para obtener instrucciones de instalación para otras distribuciones de Linux, consulte [Instalación del cliente de Lustre](#) en la Guía del usuario de Amazon FSx for Lustre. Para obtener más información, consulta [Cómo montar automáticamente el sistema de FSx archivos de Amazon](#) en la Guía del usuario de Amazon FSx for Lustre.

```

MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="==MYBOUNDARY=="

--==MYBOUNDARY==
Content-Type: text/cloud-config; charset="us-ascii"

runcmd:
- file_system_id_01=fs-0abcdef1234567890
- region=us-east-2
- fsx_directory=/scratch
- amazon-linux-extras install -y lustre2.10
- mkdir -p ${fsx_directory}
- mount -t lustre ${file_system_id_01}.fsx.${region}.amazonaws.com@tcp:fsx
  ${fsx_directory}

--==MYBOUNDARY==

```


En los miembros [volumes](#) y [mountPoints](#) de las propiedades del contenedor, se deben asignar los puntos de montaje al contenedor.

```
{
  "volumes": [
    {
      "host": {
        "sourcePath": "/scratch"
      },
      "name": "Scratch"
    }
  ],
  "mountPoints": [
    {
      "containerPath": "/scratch",
      "sourceVolume": "Scratch"
    }
  ],
}
```

Configuración del servicio de metadatos de instancias (IMDS)

El servicio de metadatos de instancias (IMDS) brinda metadatos sobre las instancias de EC2 para las aplicaciones que se ejecutan en dichas instancias. Utilice IMDSv2 para todas las cargas de trabajo nuevas y migre las existentes de IMDSv1 a IMDSv2 para mejorar la seguridad. Para obtener más información sobre el IMDS y su configuración, consulte [Uso de los metadatos de la instancia para administrar su instancia de EC2](#) y [Configurar las opciones de metadatos para instancias nuevas](#) en la Guía del usuario de Amazon EC2.

Casos de configuración

Elija el método de configuración adecuado según la configuración de su entorno de computación:

AMI predeterminada sin plantilla de lanzamiento

Si utiliza la AMI predeterminada de AWS Batch y no especifica una plantilla de lanzamiento, elija una de estas opciones:

1. Uso de la AMI predeterminada de Amazon Linux 2023: Amazon Linux 2023 requiere IMDSv2 de forma predeterminada. Cuando se cree su entorno de computación, seleccione Amazon Linux 2023 como tipo de imagen.

2. Ajuste de la configuración de IMDSv2 a nivel de cuenta: Configure su cuenta de AWS para que requiera IMDSv2 en todas las instancias nuevas. Esta configuración afecta a todas las instancias nuevas que se lancen en la cuenta. Para obtener instrucciones, consulte [Ajuste de IMDSv2 como valor predeterminado de la cuenta](#) en la Guía del usuario de Amazon EC2.

 Note

La configuración de IMDS a nivel de cuenta se puede anular mediante la plantilla de lanzamiento o la configuración de AMI. La configuración de plantilla de lanzamiento tiene prioridad sobre la configuración a nivel de cuenta.

AMI personalizada sin plantilla de lanzamiento

Cuando utilice una AMI personalizada sin una plantilla de lanzamiento, elija una de estas opciones:

1. Uso de Amazon Linux 2023 como base: Cree su AMI personalizada con Amazon Linux 2023 como imagen base. Para obtener información acerca de cómo crear AMI personalizadas para Batch, consulte [Tutorial: creación de una AMI de recursos de computación](#).
2. Configuración de IMDSv2 en la AMI personalizada: Cuando cree su AMI personalizada, configúrela para que requiera IMDSv2. Para obtener instrucciones, consulte [Configurar las opciones de metadatos de la instancia](#) en la Guía del usuario de Amazon EC2.
3. AWSServiceRoleForBatch Ajuste de la configuración de IMDSv2 a nivel de cuenta: Configure su cuenta de AWS para que requiera IMDSv2 en todas las instancias nuevas. Esta configuración afecta a todas las instancias nuevas que se lancen en la cuenta. Para obtener instrucciones, consulte [Ajuste de IMDSv2 como valor predeterminado de la cuenta](#) en la Guía del usuario de Amazon EC2.

 Note

La configuración de IMDS a nivel de cuenta se puede anular mediante la plantilla de lanzamiento o la configuración de AMI. La configuración de plantilla de lanzamiento tiene prioridad sobre la configuración a nivel de cuenta.

Uso de plantillas de lanzamiento

Cuando utilice plantillas de lanzamiento en su entorno de computación, agregue opciones de metadatos a la plantilla de lanzamiento para que requiera IMDSv2. Para obtener más información acerca del uso de plantillas de lanzamiento con Batch, consulte [Utilice las plantillas de lanzamiento de Amazon EC2 con AWS Batch](#).

```
{
  "LaunchTemplateName": "batch-imdsv2-template",
  "VersionDescription": "IMDSv2 only template for Batch",
  "LaunchTemplateData": {
    "MetadataOptions": {
      "HttpTokens": "required"
    }
  }
}
```

Cómo crear la plantilla de lanzamiento con la CLI de AWS:

```
aws ec2 create-launch-template --cli-input-json file://imds-template.json
```

Configuraciones de EC2

AWS Batch utiliza AMI optimizadas para Amazon ECS para entornos de computación EC2 y EC2 de spot. El predeterminado es [Amazon Linux 2](#) (ECS_AL2). Desde enero de 2026, el valor predeterminado cambiará a [AL2023](#) (ECS_AL2023).

AWS finalizará su compatibilidad con Amazon Linux 2. Le recomendamos migrar los entornos de computación de Amazon ECS de AWS Batch a Amazon Linux 2023 para mantener un rendimiento y una seguridad óptimos. Para obtener más información, consulte [Obsolescencia de la AMI de Amazon Linux 2 en Amazon ECS](#).

Le recomendamos que actualice los entornos de computación actuales basados en Amazon Linux a Amazon Linux 2023 para evitar interrupciones imprevistas de la carga de trabajo y seguir recibiendo actualizaciones de seguridad u otro tipo.

Para obtener ayuda al migrar AWS Batch de la AMI de Amazon Linux a Amazon Linux 2023, consulte [Cómo migrar de ECS AL2 a ECS AL2023](#)

Temas

- [Cómo migrar de ECS AL2 a ECS AL2023](#)

Cómo migrar de ECS AL2 a ECS AL2023

AL2023 es un sistema operativo basado en Linux diseñado para proporcionar un entorno seguro, estable y de alto rendimiento para las aplicaciones en la nube. Para obtener más información sobre las diferencias entre AL2 y AL2023, consulte [Comparación entre Amazon Linux 2 y Amazon Linux 2023](#) en la Guía del usuario de Amazon Linux 2023.

Desde enero de 2026, AWS Batch cambiará la AMI predeterminada para los nuevos entornos de computación de Amazon ECS desde Amazon Linux 2 a Amazon Linux 2023, ya que AWS [finalizará su compatibilidad con Amazon Linux 2](#). La AMI predeterminada se utiliza cuando no se especifica un valor para el campo [imageType.Ec2Configuration](#) al crear un nuevo entorno de computación. Le recomendamos migrar los entornos de computación de Amazon ECS de AWS Batch a Amazon Linux 2023 para mantener un rendimiento y una seguridad óptimos.

Según cómo esté configurado su entorno de computación, puede utilizar una de las siguientes rutas de actualización de AL2 a AL2023.

Actualización con Ec2Configuration.ImageType

- Si no utiliza una plantilla de lanzamiento ni anulaciones de la plantilla de lanzamiento, entonces cambie de [Ec2Configuration.ImageType](#) a ECS_AL2023 (o a ECS_AL2023_NVIDIA cuando utilice instancias de GPU) y, a continuación, ejecute [UpdateComputeEnvironment](#).
- Si especifica un [Ec2Configuration.ImageIdOverride](#), entonces [Ec2Configuration.ImageType](#) debe coincidir con el tipo de AMI especificado en [Ec2Configuration.ImageIdOverride](#).

Si ImageIdOverride y ImageType no coinciden, entonces es probable que el entorno de computación no funcione correctamente.

Actualización con plantillas de lanzamiento

- Si utiliza una plantilla de lanzamiento que especifica una AMI basada en ECS_AL2023, asegúrese de que dicha plantilla sea compatible con Amazon Linux 2023. Para obtener información sobre los cambios de Amazon Linux 2023 con la AMI optimizada para Amazon ECS, consulte [Migración de una AMI de Amazon Linux 2 a una AMI de Amazon Linux 2023 optimizada para Amazon ECS](#) en la Guía de usuario de Amazon ECS.

- En el caso de las AMI AL2023, verifique que los datos de usuario personalizados o los scripts de inicialización sean compatibles con el entorno AL2023 y el sistema de administración de paquetes.

Actualización con CloudFormation

- Si utiliza CloudFormation para administrar los entornos de computación, actualice la plantilla para cambiar la propiedad `ImageType` en la `Ec2Configuration` de `ECS_AL2` a `ECS_AL2023` (o a `ECS_AL2023_NVIDIA` cuando utilice instancias de GPU):

```
ComputeEnvironment:
  Type: AWS::Batch::ComputeEnvironment
  Properties:
    ComputeResources:
      Ec2Configuration:
        - ImageType: ECS_AL2023
```

Finalmente, actualice la pila de CloudFormation para que los cambios sean efectivos.

- Si la plantilla de CloudFormation especifica una AMI personalizada con el uso de `ImageIdOverride`, asegúrese de que el ID de la AMI corresponda a una AMI basada en AL2023 y que coincida con la configuración `ImageType`.

Consideraciones sobre la migración

Al migrar de Amazon Linux 2 a Amazon Linux 2023, tenga presente lo siguiente:

- Administración de paquetes: Amazon Linux 2023 usa `dnf` en vez de `yum` para la administración de paquetes.
- Servicios del sistema: Algunos servicios del sistema y sus configuraciones pueden diferir entre AL2 y AL2023.
- Tiempo de ejecución del contenedor: Tanto AL2 como AL2023 son compatibles con Docker, pero AL2023 puede tener diferentes configuraciones predeterminadas.
- Seguridad: AL2023 cuenta con funciones de seguridad mejoradas y puede requerir actualizaciones en las configuraciones relacionadas con la seguridad.
- Servicio de metadatos de instancia versión 2 (IMDSv2): IMDSv2 es un servicio orientado a la sesión que requiere una autenticación basada en tokens para acceder a los metadatos de las instancias de EC2, lo que permite entregar una seguridad mejorada. Para obtener más información

sobre IMDS, consulte [Funcionamiento de Servicio de metadatos de instancia versión 2](#) en la Guía del usuario de Amazon EC2.

Para obtener una lista completa de los cambios y las consideraciones sobre la migración, consulte [Migración de una AMI de Amazon Linux 2 a una AMI de Amazon Linux 2023 optimizada para Amazon ECS](#) en la Guía del usuario de Amazon ECS.

Estrategias de asignación de tipos de instancia para AWS Batch

Cuando se crea un entorno de computación administrado, AWS Batch seleccionará los tipos de instancia de entre los [instanceTypes](#) especificados que mejor se adapten a las necesidades de los trabajos. La estrategia de asignación define el comportamiento cuando AWS Batch necesita capacidad adicional. Este parámetro no es aplicable a trabajos que se ejecutan en recursos de Fargate. No los especifique en ambos lugares.

BEST_FIT (predeterminado)

AWS Batch selecciona el tipo de instancia que mejor se adapte a las necesidades de los trabajos con preferencia por el tipo de instancia de menor costo. Si no hay instancias adicionales disponibles del tipo de instancia seleccionada, AWS Batch espera a que estén disponibles. Si no hay suficientes instancias disponibles, o si el usuario alcanza los [Service Quotas de Amazon EC2](#), no se ejecutarán trabajos adicionales hasta que finalicen aquellos que se estén ejecutando en ese momento. Esta estrategia de asignación mantiene los costos más bajos, pero puede limitar el escalado. Si está utilizando la flota de spot con BEST_FIT, el rol de IAM de la flota spot debe especificarse. BEST_FIT no se admite cuando se actualizan los entornos de computación. Para obtener más información, consulte [Actualizar un entorno informático en AWS Batch](#).

Note

AWS Batch administra recursos de AWS en su cuenta. Los entornos de computación con la estrategia de asignación BEST_FIT utilizaban originalmente configuraciones de lanzamiento de forma predeterminada. Sin embargo, el uso de configuraciones de lanzamiento con cuentas de AWS nuevas se restringirá con el tiempo. Por lo tanto, desde finales de abril de 2024, los entornos de computación BEST_FIT recién creados lanzarán plantillas de forma predeterminada. Si su rol de servicio no tiene permisos para administrar las plantillas de lanzamiento, AWS Batch puede seguir utilizando

configuraciones de lanzamiento. Los entornos de computación existentes seguirán utilizando las configuraciones de lanzamiento.

BEST_FIT_PROGRESSIVE

AWS Batch selecciona tipos de instancia adicionales que sean lo suficientemente grandes como para cumplir los requisitos de los trabajos en la cola. Se prefieren los tipos de instancias con un costo menor para cada unidad de vCPU. Si las instancias adicionales de los tipos de instancia previamente seleccionados no están disponibles, AWS Batch selecciona nuevos tipos de instancia.

Note

Para los [trabajos paralelos de varios nodos](#), AWS Batch elige el tipo de instancia óptimo disponible. Si el tipo de instancia deja de estar disponible debido a la capacidad insuficiente, no se lanzarán otros tipos de instancias del grupo.

SPOT_CAPACITY_OPTIMIZED

AWS Batch selecciona uno o más tipos de instancia que sean lo suficientemente grandes como para cumplir los requisitos de los trabajos en la cola. Se prefieren los tipos de instancias que tienen menos probabilidades de interrumpirse. Esta estrategia de asignación solo está disponible para los recursos de computación de instancia de spot.

SPOT_PRICE_CAPACITY_OPTIMIZED

La estrategia de asignación optimizada por precio y capacidad analiza tanto el precio como la capacidad para seleccionar los grupos de instancias de spot que tienen menos probabilidades de interrupción y el precio más bajo posible. Esta estrategia de asignación solo está disponible para los recursos de computación de instancia de spot.

Note

Le recomendamos que utilice SPOT_PRICE_CAPACITY_OPTIMIZED en vez de SPOT_CAPACITY_OPTIMIZED la mayoría de los casos.

Las estrategias `BEST_FIT_PROGRESSIVE` y `BEST_FIT` utilizan instancias de spot o de bajo demanda, y las estrategias `SPOT_CAPACITY_OPTIMIZED` y `SPOT_PRICE_CAPACITY_OPTIMIZED` utilizan instancias de spot. Sin embargo, AWS Batch podría necesitar superar a `maxvCpus` para cumplir con sus requisitos de capacidad. En este caso, AWS Batch nunca supera `maxvCpus` en más de una sola instancia.

Administración de la memoria de recurso de computación

Cuando el agente de contenedores de Amazon ECS registra un recurso de computación en un entorno de computación, el agente debe determinar cuánta memoria tiene disponible el recurso de computación para reservarla para sus trabajos. Debido a la sobrecarga de memoria de la plataforma y a la memoria ocupada por el núcleo del sistema, este número es diferente de la cantidad de memoria instalada para las instancias de Amazon EC2. Por ejemplo, una instancia `m4.large` tiene 8 GiB de memoria instalada. Sin embargo, esto no siempre se traduce en exactamente 8192 MiB de memoria disponible para los trabajos cuando se registra el recurso de computación.

Suponga que especifica 8192 MiB para el trabajo y que ninguno de sus recursos de computación tiene 8192 MiB o más de memoria disponible para cumplir este requisito. Entonces, el trabajo no se puede colocar en su entorno de computación. Si está utilizando un entorno de computación gestionado, AWS Batch debe lanzar un tipo de instancia mayor para dar cabida a la solicitud.

La AMI predeterminada de recursos de computación de AWS Batch también reserva 32 MiB de memoria para el agente de contenedor de Amazon ECS y otros procesos críticos del sistema. Esta memoria no está disponible para la asignación de trabajos. Para obtener más información, consulte [Reserva de memoria del sistema](#).

El agente de contenedor de Amazon ECS utiliza la función `ReadMemInfo()` de Docker para consultar la memoria total disponible al sistema operativo. Linux proporciona utilidades de línea de comandos para determinar la memoria total.

Example- Determinar la memoria total en Linux

El comando `free` devuelve la memoria total reconocida por el sistema operativo.

```
$ free -b
```

A continuación, se muestra un ejemplo de salida para una instancia `m4.large` que ejecuta la AMI de Amazon Linux optimizada para Amazon ECS.

	total	used	free	shared	buffers	cached
Mem:	8373026816	348180480	8024846336	90112	25534464	205418496
-/+ buffers/cache:		117227520	8255799296			

Esta instancia tiene 8373026816 bytes de memoria total. Esto significa que hay 7985 MiB disponibles para tareas.

Temas

- [Reserva de memoria del sistema](#)
- [Tutorial: visualización de la memoria de recurso de computación](#)
- [Consideraciones sobre memoria y vCPU para AWS Batch en Amazon EKS](#)

Reserva de memoria del sistema

Si ocupa toda la memoria de un recurso de computación con sus trabajos, es posible que estos compitan por la memoria con procesos críticos del sistema y provoquen un fallo del sistema. El agente contenedor Amazon ECS proporciona una variable de configuración que se denomina `ECS_RESERVED_MEMORY`. Puede utilizar esta variable de configuración para eliminar un número determinado de MiB de memoria del pool asignado a sus trabajos. Este es un mecanismo eficaz que permite reservar memoria para los procesos críticos del sistema.

La AMI predeterminada de recursos de computación de AWS Batch reserva 32 MiB de memoria para el agente de contenedor de Amazon ECS y otros procesos críticos del sistema. Recomendamos reservar un búfer de memoria del 5 % para el agente de contenedor de Amazon ECS y otros procesos esenciales del sistema.

Tutorial: visualización de la memoria de recurso de computación

Puede ver con cuánta memoria registra un recurso de computación en la consola de Amazon ECS o con la operación de la API [DescribeContainerInstances](#). Si intenta maximizar la utilización de sus recursos proporcionando a sus trabajos tanta memoria como sea posible para un tipo de instancia en particular, puede observar la memoria disponible para ese recurso de computación y luego asignar a sus trabajos esa cantidad de memoria.

Para ver la memoria de recurso de computación

1. Abra la consola en <https://console.aws.amazon.com/ecs/v2>.

2. Elija Clústeres y, a continuación, elija el clúster que aloja los recursos de computación que desee ver.

El nombre del clúster del entorno de computación comienza por el nombre del entorno de computación.

3. Elija Infraestructura.
4. En Instancias de contenedor, elija la instancia de contenedor.
5. La sección Recursos y redes muestra la memoria registrada y disponible para el recurso de computación.

El valor de memoria de la Capacidad total es lo que el recurso de computación registró con Amazon ECS cuando se lanzó por primera vez, y el valor de memoria Disponible es lo que aún no se ha asignado a los trabajos.

Consideraciones sobre memoria y vCPU para AWS Batch en Amazon EKS

En AWS Batch en Amazon EKS, puede especificar los recursos que se ponen a disposición de un contenedor. Por ejemplo, puede especificar los valores `requests` o `limits` para los recursos vCPU y memoria.

Las siguientes son restricciones para especificar los recursos de vCPU:

- Debe especificarse al menos una vCPU de valor `requests` o `limits`.
- Una unidad vCPU equivale a un núcleo físico o virtual.
- El valor de vCPU debe introducirse en números enteros o en incrementos de 0,25.
- El valor de vCPU válido más bajo es 0,25.
- Si se especifican ambos, el valor `requests` debe ser menor o igual que el valor `limits`. De esta forma, puede configurar configuraciones de vCPU flexibles e invariables.
- Los valores de vCPU no se pueden especificar en formato milliCPU. Por ejemplo, `100m` no es un valor válido.
- AWS Batch utiliza el valor `requests` para tomar decisiones de escalado. Si no se especifica un valor `requests`, el valor `limits` se copia al valor `requests`.

Las siguientes son restricciones para especificar los recursos de memoria:

- Debe especificarse al menos una memoria de valor `requests` o `limits`.

- Los valores de memoria deben estar en mebibytes (MiBs).
- Si se especifican ambos, el valor `requests` debe ser igual que el valor `limits`.
- AWS Batch utiliza el valor `requests` para tomar decisiones de escalado. Si no se especifica un valor `requests`, el valor `limits` se copia al valor `requests`.

Las siguientes son restricciones para especificar los recursos de GPU:

- Si se especifican ambos, el valor `requests` debe ser igual que el valor `limits`.
- AWS Batch utiliza el valor `requests` para tomar decisiones de escalado. Si no se especifica un valor `requests`, el valor `limits` se copia al valor `requests`.

Ejemplo: definiciones de trabajo

La siguiente AWS Batch de Amazon EKS configura los recursos compartidos de vCPU flexibles. Esto permite a AWS Batch de Amazon EKS utilizar toda la capacidad de vCPU para el tipo de instancia. Sin embargo, si hay otros trabajos en ejecución, al trabajo se le asigna un máximo de 2 vCPU. La memoria está limitada a 2 GB.

```
{
  "jobDefinitionName": "MyJobOnEks_Sleep",
  "type": "container",
  "eksProperties": {
    "podProperties": {
      "containers": [
        {
          "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
          "command": ["sleep", "60"],
          "resources": {
            "requests": {
              "cpu": "2",
              "memory": "2048Mi"
            }
          }
        }
      ]
    }
  }
}
```

La siguiente AWS Batch de trabajo de Amazon EKS tiene un request valor de 1 y asigna un máximo de 4 vCPU al trabajo.

```
{
  "jobDefinitionName": "MyJobOnEks_Sleep",
  "type": "container",
  "eksProperties": {
    "podProperties": {
      "containers": [
        {
          "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
          "command": ["sleep", "60"],
          "resources": {
            "requests": {
              "cpu": "1"
            },
            "limits": {
              "cpu": "4",
              "memory": "2048Mi"
            }
          }
        }
      ]
    }
  }
}
```

La siguiente AWS Batch de trabajo de Amazon EKS establece un valor de vCPU limits de 1 y un valor de memoria limits de 1 GB.

```
{
  "jobDefinitionName": "MyJobOnEks_Sleep",
  "type": "container",
  "eksProperties": {
    "podProperties": {
      "containers": [
        {
          "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
          "command": ["sleep", "60"],
          "resources": {
            "limits": {
              "cpu": "1",
              "memory": "1024Mi"
            }
          }
        }
      ]
    }
  }
}
```

```
}
}
}
}
}
}
```

Cuando AWS Batch traduce un `AWS Batch` en Amazon EKS a un pod de Amazon EKS, AWS Batch copia el valor `limits` al valor `requests`. Esto ocurre si no se especifica un valor `requests`. Al enviar la definición de trabajo del ejemplo anterior, el pod spec es el siguiente.

```
apiVersion: v1
kind: Pod
...
spec:
  ...
  containers:
    - command:
      - sleep
      - 60
      image: public.ecr.aws/amazonlinux/amazonlinux:2
      resources:
        limits:
          cpu: 1
          memory: 1024Mi
        requests:
          cpu: 1
          memory: 1024Mi
      ...
```

Reservas de CPU y memoria de los nodos

AWS Batch se basa en la lógica predeterminada del archivo `bootstrap.sh` para las reservas de vCPU y memoria. Para obtener más información sobre el archivo `bootstrap.sh`, consulte [bootstrap.sh](https://aws.amazon.com/batch/docs/bootstrap-container-definitions.html). Al ajustar el tamaño de los recursos de vCPU y memoria, tenga en cuenta los ejemplos siguientes.

 Note

Si no se está ejecutando ninguna instancia, las reservas de vCPU y memoria pueden afectar inicialmente a la lógica de escalado de AWS Batch y a la toma de decisiones. Una vez ejecutadas las instancias, AWS Batch ajusta las asignaciones iniciales.

Ejemplo: reserva de CPU de nodo

El valor de reserva de CPU se calcula en milinúcleos mediante la cantidad total de vCPU disponibles para la instancia.

Número de vCPU	Porcentaje reservado
1	6 %
2	1 %
3-4	0,5%
4 y superior	0,25 %

Si se utilizan los valores anteriores, se aplica lo siguiente:

- El valor de reserva de CPU para una instancia `c5.large` con 2 vCPU es de 70 m. Se calcula de la siguiente manera: $(1 \times 60) + (1 \times 10) = 70$ m.
- El valor de reserva de CPU para una instancia `c5.24xlarge` con 96 vCPU es de 310 m. Se calcula de la siguiente manera: $(1 \times 60) + (1 \times 10) + (2 \times 5) + (92 \times 2.5) = 310$ m.

En este ejemplo, hay 1930 unidades de vCPU milicore (calculadas entre 2000 y 70) disponibles para ejecutar trabajos en una instancia `c5.large`. Supongamos que su trabajo requiere unidades vCPU 2 (2×1000 m), el trabajo no cabe en una sola instancia `c5.large`. Sin embargo, un trabajo que requiere unidades de vCPU 1.75 es adecuado.

Ejemplo: reserva de memoria de nodo

El valor de reserva de memoria se calcula en mebibytes de la siguiente manera:

- Capacidad de la instancia en mebibytes. Por ejemplo, una instancia de 8 GB equivale a 7,748 MiB.
- El valor kubeReserved. El valor kubeReserved es la cantidad de memoria que se debe reservar para los daemons del sistema. El valor kubeReserved se calcula de la siguiente manera: $((11 * \text{número máximo de pods que admite el tipo de instancia}) + 255)$. Para obtener información sobre el número máximo de pods que admite un tipo de instancia, consulte [eni-max-pods.txt](#)
- El valor HardEvictionLimit. Cuando la memoria disponible cae por debajo del valor HardEvictionLimit, la instancia intenta expulsar los pods.

La fórmula para calcular la memoria asignable es la siguiente: $(\text{instance\_capacity\_in\_MiB}) - (11 * (\text{maximum\_number\_of\_pods})) - 255 - (\text{HardEvictionLimit value.})$.

Una instancia c5.large admite hasta 29 pods. Para una instancia c5.large de 8 GB con un valor HardEvictionLimit de 100 MiB, la memoria asignable es 7074 MiB. Esto se calcula de la siguiente manera: $(7748 - (11 * 29) - 255 - 100) = 7074$ MiB. En este ejemplo, un trabajo MiB de 8192 no cabe en esta instancia aunque sea una instancia 8 gibibyte (GiB).

DaemonSets

Cuando utilice DaemonSets, tenga en cuenta lo siguiente:

- Si no se está ejecutando ninguna instancia AWS Batch de Amazon EKS, DaemonSets puede afectar inicialmente a la lógica del escalado y a la toma de decisiones de AWS Batch. AWS Batch inicialmente asigna 0,5 unidades de vCPU y 500 MiB para DaemonSets previsto. Una vez ejecutadas las instancias, AWS Batch ajusta las asignaciones iniciales.
- Si DaemonSet define los límites de vCPU o memoria, los AWS Batch en Amazon EKS tienen menos recursos. Le recomendamos que mantenga el número de DaemonSets asignados a los AWS Batch trabajos lo más bajo posible.

Entornos de computación en Fargate

Fargate es una tecnología que puede utilizar AWS Batch para ejecutar [contenedores](#) sin tener que gestionar servidores o clústeres de instancias de Amazon EC2. Con Fargate, ya no tendrá que aprovisionar, configurar ni escalar clústeres de máquinas virtuales para ejecutar los contenedores. De esta manera, se elimina la necesidad de elegir tipos de servidores, decidir cuándo escalar los clústeres u optimizar conjuntos de clústeres.

Al ejecutar los trabajos con los recursos de Fargate, la aplicación se empaqueta en contenedores, se especifican los requisitos de CPU y de memoria, se definen las políticas de IAM y de redes y se lanza la aplicación. Cada trabajo de Fargate tiene su propio límite de aislamiento y no comparte el kernel subyacente, los recursos de CPU, los recursos de memoria ni la interfaz de red elástica con otro trabajo.

Temas

- [Cuándo usar Fargate](#)
- [Definiciones de trabajo en Fargate](#)
- [Colas de trabajo en Fargate](#)
- [Entornos de computación en Fargate](#)

Cuándo usar Fargate

Recomendamos usar Fargate en la mayoría de los casos. Fargate lanza y escala el cómputo para que se ajuste perfectamente a los requisitos de recursos que especifique para el contenedor. Con Fargate, no es necesario aprovisionar en exceso los servidores ni pagar por ellos. Tampoco tiene que preocuparse por los detalles de los parámetros relacionados con la infraestructura, como el tipo de instancia. Cuando es necesario escalar verticalmente el entorno de cómputo, los trabajos que se ejecutan en los recursos de Fargate pueden comenzar más rápidamente. Por lo general, se tardará unos minutos hasta que se lance la instancia de Amazon EC2. Sin embargo, los trabajos que se ejecutan en Fargate se pueden aprovisionar en unos 30 segundos. El tiempo exacto necesario depende de varios factores, como el tamaño de la imagen del contenedor y la cantidad de trabajos.

No obstante, recomendamos utilizar Amazon EC2 si sus trabajos requieren cualquiera de las siguientes opciones:

- Más de 16 v CPUs
- Más de 120 gibibytes (GiB) de memoria
- Una GPU
- Imagen de máquina de Amazon (AMI) personalizada
- Alguno de los parámetros de [LinuxParameters](#)

Si tiene un gran número de trabajos, le recomendamos utilizar la infraestructura de Amazon EC2. Por ejemplo, si el número de trabajos que se ejecutan simultáneamente supera la limitación de

regulación de Fargate. Esto se debe a que, con EC2, los trabajos se pueden enviar a un ritmo mayor a los recursos de EC2 que a los recursos de Fargate. Además, se pueden ejecutar más trabajos de forma simultánea cuando se utiliza EC2. Para obtener más información, consulte [Cuotas de servicio de Fargate](#) en la Guía para desarrolladores de Amazon Elastic Container Service.

Definiciones de trabajo en Fargate

AWS Batch los trabajos AWS Fargate activados no admiten todos los parámetros de definición de trabajos disponibles. Algunos parámetros no son compatibles, y otros se comportan de forma distinta para trabajos de Fargate.

La siguiente lista describe los parámetros de definición de trabajo que no son válidos o están restringidos de otro modo en los trabajos de Fargate.

platformCapabilities

Debe especificarse como FARGATE.

```
"platformCapabilities": [ "FARGATE" ]
```

type

Debe especificarse como container.

```
"type": "container"
```

Parámetros de containerProperties

executionRoleArn

Debe especificarse para trabajos que se ejecutan en recursos de Fargate. Para obtener más información, consulte [Roles de IAM para tareas](#) en la Guía para desarrolladores de Amazon Elastic Container Service.

```
"executionRoleArn": "arn:aws:iam::123456789012:role/ecsTaskExecutionRole"
```

fargatePlatformConfiguration

(Opcional, solo para las definiciones de trabajo de Fargate). Especifica la versión de la plataforma Fargate o LATEST para una versión reciente de la plataforma. Los valores posibles de platformVersion son 1.3.0, 1.4.0 y LATEST (predeterminado).

```
"fargatePlatformConfiguration": { "platformVersion": "1.4.0" }
```

instanceType, ulimits

No se aplica a trabajos que se ejecutan en recursos de Fargate.

memory, vcpus

Esta configuración debe especificarse en resourceRequirements

privileged

No especifique este parámetro o especifique false.

```
"privileged": false
```

resourceRequirements

Los requisitos de memoria y vCPU deben especificarse mediante los [valores admitidos](#). Los recursos de GPU no son compatibles con los recursos de Fargate.

Si utiliza GuardDuty Runtime Monitoring, el agente de GuardDuty seguridad tiene una pequeña sobrecarga de memoria. Por lo tanto, el límite de memoria debe incluir el tamaño del agente de GuardDuty seguridad. Para obtener información sobre los límites de memoria del agente de GuardDuty seguridad, consulte los [límites de CPU y memoria](#) en la Guía del GuardDuty usuario. Para obtener información sobre las prácticas recomendadas, consulte [Cómo solucionar los errores de falta de memoria en mis tareas de Fargate después de activar Runtime Monitoring](#) en la Guía para desarrolladores de Amazon ECS.

```
"resourceRequirements": [  
  {"type": "MEMORY", "value": "512"},  
  {"type": "VCPU", "value": "0.25"}  
]
```

Parámetros de linuxParameters

devices, maxSwap, sharedMemorySize, swappiness, tmpfs

No se aplica a trabajos que se ejecutan en recursos de Fargate.

Parámetros de logConfiguration

logDriver

Solo se admiten awslogs y splunk. Para obtener más información, consulte [Uso del controlador de registros awslogs](#).

Miembros en networkConfiguration

assignPublicIp

Si la subred privada no tiene una puerta de enlace NAT conectada para enviar tráfico a Internet, [assignPublicIp](#) debe ser "ENABLED". Para obtener más información, consulte [AWS Batch Función de ejecución de IAM](#).

Colas de trabajo en Fargate

AWS Batch las colas de trabajos activas AWS Fargate prácticamente no cambian. La única restricción es que todos los entornos de computación que aparecen en la lista en `computeEnvironmentOrder` deben ser entornos de computación de Fargate (FARGATE o FARGATE_SPOT). Los entornos de computación de EC2 y Fargate no se pueden mezclar.

Entornos de computación en Fargate

AWS Batch los entornos de cómputo activados AWS Fargate no admiten todos los parámetros del entorno de cómputo disponibles. Algunos parámetros no son compatibles. Otros tienen requisitos específicos para Fargate.

La siguiente lista describe los parámetros de entorno de computación que no son válidos o están restringidos de otro modo en los trabajos de Fargate.

type

Este parámetro debe establecerse en MANAGED.

```
"type": "MANAGED"
```

Parámetros del objeto `computeResources`

`allocationStrategy`, `bidPercentage`, `desiredvCpus`, `imageId`, `instanceTypes`, `ec2Configuration`, `ec2KeyPair`, `instanceRole`, `launchTemplate`, `minvCpus`, `placementGroup`, `spotIamFleetRole`

No se aplican a los entornos de computación de Fargate y no se pueden proporcionar.

`subnets`

Si las subredes enumeradas en este parámetro no tienen puertas de enlace NAT conectadas, el parámetro `assignPublicIp` de la definición del trabajo debe estar establecido en `ENABLED`.

`tags`

Esto no se aplica a los entornos de computación de Fargate y no se puede proporcionar. Para especificar etiquetas para los entornos de computación de Fargate, use el parámetro `tags` que no está en el objeto `computeResources`.

`type`

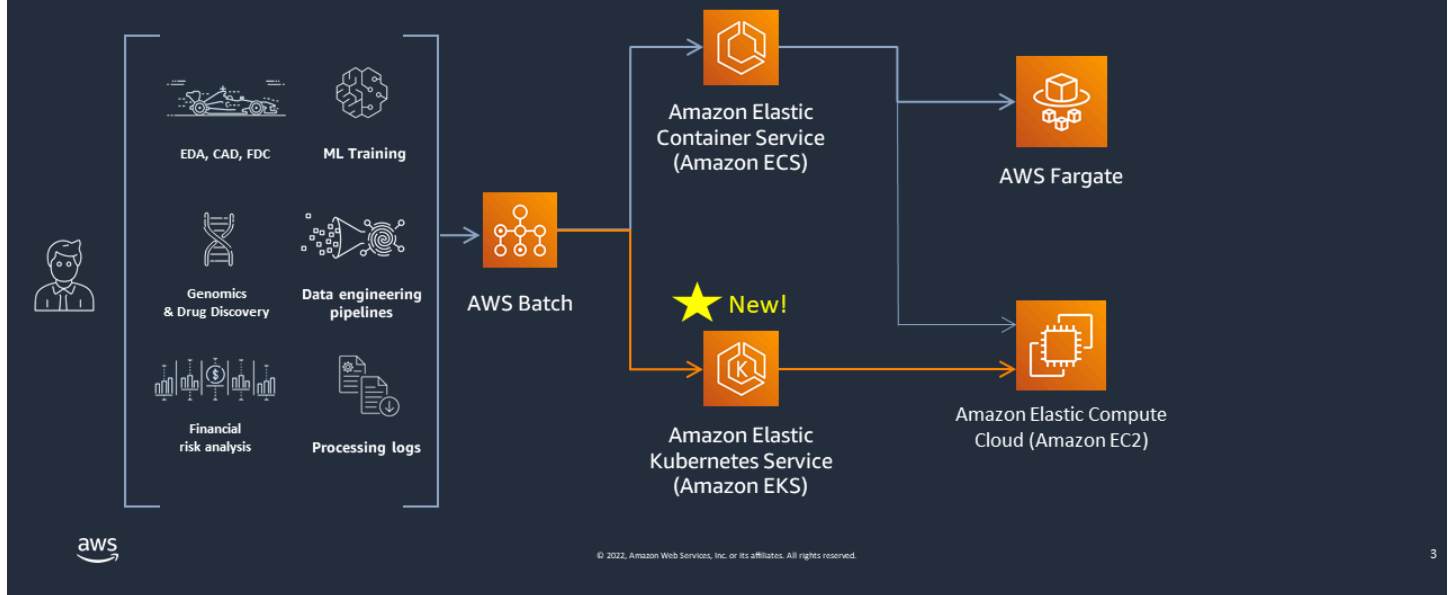
Debe ser `FARGATE` o `FARGATE_SPOT`.

```
"type": "FARGATE_SPOT"
```

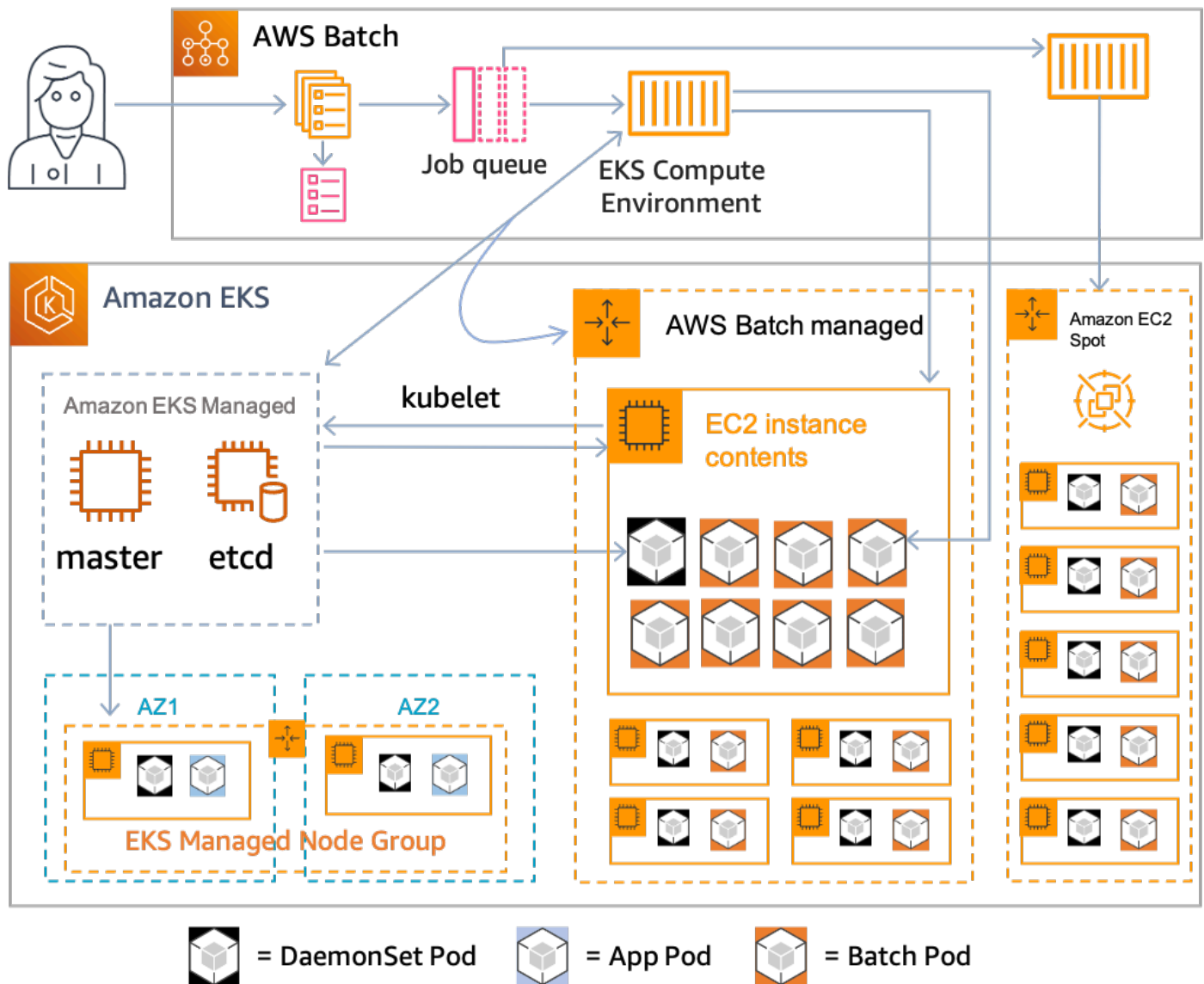
Entornos de computación de Amazon EKS

[Cómo empezar con AWS Batch Amazon EKS](#) proporciona una breve guía para crear entornos de cómputo de EKS. En esta sección se proporcionan más detalles sobre los entornos de computación de Amazon EKS.

AWS Batch for Amazon Elastic Kubernetes Service (Amazon EKS)



AWS Batch simplifica las cargas de trabajo por lotes en los clústeres de Amazon EKS al proporcionar funciones de gestión por lotes. Esto incluye la creación de colas, el seguimiento de las dependencias, la gestión de los reintentos y las prioridades de los trabajos, la gestión de los pods y el escalado de los nodos. AWS Batch puede gestionar varias zonas de disponibilidad y varios tipos y tamaños de instancias de Amazon EC2. AWS Batch integra varias de las prácticas recomendadas de Amazon EC2 Spot para ejecutar sus cargas de trabajo de forma tolerante a errores, lo que permite reducir las interrupciones. Puede utilizar AWS Batch para ejecutar un puñado de trabajos de un día para otro o millones de trabajos de misión crítica con total confianza.



AWS Batch es un servicio gestionado que organiza las cargas de trabajo por lotes en Kubernetes los clústeres gestionados por Amazon Elastic Kubernetes Service (Amazon EKS). AWS Batch lleva a cabo esta organización de forma externa a sus clústeres mediante un modelo de «superposición». Como AWS Batch se trata de un servicio gestionado, no hay Kubernetes componentes (por ejemplo, operadores o recursos personalizados) que instalar o gestionar en el clúster. AWS Batch solo necesita que el clúster esté configurado con controles de acceso basados en roles (RBAC) que permitan AWS Batch comunicarse con el servidor de API. Kubernetes AWS Batch llamadas Kubernetes APIs para crear, monitorear y eliminar Kubernetes pods y nodos.

AWS Batch tiene una lógica de escalado integrada para escalar Kubernetes los nodos en función de la carga de la cola de trabajos con optimizaciones en términos de asignación de la capacidad de trabajo. Cuando la cola de trabajos esté vacía, reduce la AWS Batch escala de los nodos hasta

alcanzar la capacidad mínima que haya establecido, que de forma predeterminada es cero. AWS Batch gestiona todo el ciclo de vida de estos nodos y los decora con etiquetas y manchas. De esta forma, no se colocan otras Kubernetes cargas de trabajo en los nodos gestionados por. AWS Batch La excepción son las DaemonSets que pueden dirigirse a AWS Batch los nodos para proporcionar supervisión y otras funciones necesarias para la correcta ejecución de las tareas. Además, AWS Batch no ejecuta trabajos, específicamente pods, en los nodos del clúster que no administra. De esta forma, puede usar una lógica de escalado y servicios independientes para otras aplicaciones del clúster.

Para enviar trabajos AWS Batch, interactúas directamente con la AWS Batch API. AWS Batch traduce los trabajos en podspecs y, a continuación, crea las solicitudes para colocar pods en los nodos gestionados por AWS Batch su clúster de Amazon EKS. Puede utilizar herramientas como, por ejemplo, `kubectl` para ver los nodos y los pods en ejecución. Cuando un pod ha completado su ejecución, AWS Batch elimina el pod que creó para mantener una carga menor en el Kubernetes sistema.

Puede empezar conectando un clúster de Amazon EKS válido con AWS Batch. A continuación, adjúntele una cola de AWS Batch trabajos y registre una definición de trabajo de Amazon EKS con atributos podspec equivalentes. Por último, envíe los trabajos mediante la operación de [SubmitJob](#) API que hace referencia a la definición del trabajo. Para obtener más información, consulte [Cómo empezar con AWS Batch Amazon EKS](#).

Amazon EKS

Temas

- [AMI predeterminada de Amazon EKS](#)
- [Entornos de AMI mixtos](#)
- [Versiones de Kubernetes compatibles](#)
- [Actualización de la versión Kubernetes del entorno de computación.](#)
- [Responsabilidad compartida de los nodos Kubernetes](#)
- [Ejecute un DaemonSet nodo AWS Batch gestionado](#)
- [Personalización de plantillas de lanzamiento de Amazon EKS](#)
- [¿Cómo actualizar de EKS AL2 a EKS AL2023](#)

AMI predeterminada de Amazon EKS

Al crear un entorno informático Amazon EKS, no necesita especificar una imagen de máquina de Amazon (AMI). AWS Batch selecciona una AMI optimizada para Amazon EKS en función de los tipos de Kubernetes versión e instancia que se especifican en la [CreateComputeEnvironment](#) solicitud. En general, recomendamos que utilice la selección de AMI predeterminada. Para obtener más información sobre Amazon EKS optimizado AMIs, consulte [Amazon Linux optimizado para Amazon EKS AMIs](#) en la Guía del usuario de Amazon EKS.

Important

Amazon Linux 2023 AMIs es la opción predeterminada AWS Batch para Amazon EKS. AWS finalizará el soporte para Amazon EKS AL2 (optimizado y AL2 acelerado) a partir del 26 de noviembre de 2025. Puede seguir utilizando Amazon Linux 2 optimizado para Amazon EKS AWS Batch proporcionado AMIs en sus entornos informáticos de Amazon EKS más allá de la end-of-support fecha del 26 de noviembre del 25 de noviembre; sin embargo, estos entornos informáticos ya no recibirán nuevas actualizaciones de software, parches de seguridad o correcciones de errores. AWS Para obtener más información sobre la actualización de AL2 a AL2023, consulte la Guía del [¿Cómo actualizar de EKS AL2 a EKS AL2023](#) AWS Batch usuario.

Ejecute el siguiente comando para ver qué tipo de AMI AWS Batch ha seleccionado para su entorno informático Amazon EKS. El siguiente ejemplo es un tipo de instancia que no es de GPU.

```
# compute CE example: indicates Batch has chosen the AL2 x86 or ARM EKS 1.32 AMI,
# depending on instance types
$ aws batch describe-compute-environments --compute-environments My-Eks-CE1 \
  | jq '.computeEnvironments[].computeResources.ec2Configuration'
[
  {
    "imageType": "EKS_AL2",
    "imageKubernetesVersion": "1.32"
  }
]
```

El siguiente ejemplo es un tipo de instancia de GPU.

```
# GPU CE example: indicates Batch has choosen the AL2 x86 EKS Accelerated 1.32 AMI
```



```
$ aws batch describe-compute-environments --compute-environments My-Eks-GPU-CE \
  | jq '.computeEnvironments[].computeResources.ec2Configuration'
[
  {
    "imageType": "EKS_AL2_NVIDIA",
    "imageKubernetesVersion": "1.32"
  }
]
```

Entornos de AMI mixtos

Puede utilizar las anulaciones de plantillas de lanzamiento para crear entornos de procesamiento con Amazon Linux 2 (AL2) y Amazon Linux 2023 (AL2023) AMIs. Esto resulta útil si se utilizan diferentes arquitecturas AMIs para distintas arquitecturas o durante los períodos de migración durante la transición de a. AL2 AL2023

Note

AWS finalizará el soporte para Amazon EKS AL2 (optimizado y AL2 acelerado) a partir del 26 de noviembre de AMIs 2025. Si bien puede seguir utilizando Amazon Linux 2 optimizado para Amazon EKS AWS Batch proporcionado AMIs en sus entornos informáticos de Amazon EKS más allá de la end-of-support fecha del 26 de noviembre del 25 de noviembre, estos entornos informáticos ya no recibirán nuevas actualizaciones de software, parches de seguridad o correcciones de errores. AWS Los entornos de AMI mixtos pueden resultar útiles durante el período de transición, ya que le permiten migrar gradualmente las cargas de trabajo y, al AL2023 mismo tiempo, mantener la compatibilidad con las cargas de trabajo AL2 basadas en las existentes.

Ejemplo de configuración con ambos tipos de AMI:

```
{
  "computeResources": {
    "launchTemplate": {
      "launchTemplateId": "TemplateId",
      "version": "1",
      "userDataType": "EKS_BOOTSTRAP_SH",
      "overrides": [
        {
          "instanceType": "c5.large",
```

```
        "imageId": "ami-al2-custom",
        "userDataType": "EKS_B00TSTRAP_SH"
    },
    {
        "instanceType": "c6a.large",
        "imageId": "ami-al2023-custom",
        "userDataType": "EKS_NODEADM"
    }
]
},
"instanceTypes": ["c5.large", "c6a.large"]
}
```

Versiones de Kubernetes compatibles

AWS Batch en Amazon EKS actualmente es compatible con las siguientes Kubernetes versiones:

- 1.34
- 1.33
- 1.32
- 1.31
- 1.30
- 1.29

Es posible que aparezca un mensaje de error similar al siguiente cuando utilice la operación de `CreateComputeEnvironment` API o la operación de `UpdateComputeEnvironment` API para crear o actualizar un entorno de computación. Este problema se produce si especifica una versión de Kubernetes no compatible en `EC2Configuration`.

At least one imageKubernetesVersion in EC2Configuration is not supported.

Para resolver este problema, elimine el entorno de procesamiento y vuelva a crearlo con una versión de Kubernetes compatible.

Puede realizar una actualización de una versión menor en su clúster de Amazon EKS. Por ejemplo, puede actualizar el clúster de 1.xx a 1.yy incluso si la versión secundaria no es compatible.

Sin embargo, es posible que el estado del entorno de computación cambie a `INVALID` después de una actualización de la versión principal. Por ejemplo, si realiza una actualización de una versión principal de `1.xx` a `2.yy`. Si la versión principal no es compatible con AWS Batch, aparecerá un mensaje de error similar al siguiente.

```
reason=CLIENT_ERROR - ... EKS Cluster version [2.yy] is unsupported
```

Actualización de la versión Kubernetes del entorno de computación.

Con AWS Batch, puede actualizar la Kubernetes versión de un entorno informático para que sea compatible con las actualizaciones de clústeres de Amazon EKS. La Kubernetes versión de un entorno de cómputo es la versión AMI de Amazon EKS para los Kubernetes nodos que se AWS Batch lanzan para ejecutar tareas. Puede realizar una actualización de versión de Kubernetes en sus nodos Amazon EKS antes o después de actualizar la versión del plano de control del clúster de Amazon EKS. Le recomendamos que actualice los nodos después de actualizar el plano de control. Para obtener más información, consulte [Actualizar la versión de Kubernetes del clúster de Amazon EKS](#) en la Guía del usuario de Amazon EKS.

Para actualizar la versión de Kubernetes de un entorno de computación, utilice la operación [UpdateComputeEnvironment](#) API.

```
$ aws batch update-compute-environment \
  --compute-environment <compute-environment-name> \
  --compute-resources \
    'ec2Configuration=[{imageType=EKS_AL2,imageKubernetesVersion=1.32}]'
```

Responsabilidad compartida de los nodos Kubernetes

El mantenimiento de los entornos de computación es una responsabilidad compartida.

- No cambies ni elimines AWS Batch nodos, etiquetas, manchas, espacios de nombres, plantillas de lanzamiento ni grupos de escalado automático. No añada información contaminada a los nodos gestionados. AWS Batch Si realiza alguno de estos cambios, su entorno de computación no será compatible y se producirán errores, incluidas las instancias inactivas.
- No dirija sus pods a los nodos AWS Batch gestionados. Si dirige sus pods a los nodos gestionados, se producirán problemas de escalado y las colas de trabajos se atascarán. Ejecuta cargas de trabajo que no se usen AWS Batch en nodos autogestionados o grupos de nodos

gestionados. Para obtener más información, consulte [Grupos de nodos administrados](#) en la Guía del usuario de Amazon EKS.

- Puede seleccionar DaemonSet a para que se ejecute en nodos AWS Batch gestionados. Para obtener más información, consulte [Ejecute un DaemonSet nodo AWS Batch gestionado](#).

AWS Batch no actualiza automáticamente el entorno informático AMIs. Es su responsabilidad actualizarlas. Ejecute el siguiente comando para actualizar AMIs a la última versión de la AMI.

```
$ aws batch update-compute-environment \  
  --compute-environment <compute-environment-name> \  
  --compute-resources 'updateToLatestImageVersion=true'
```

AWS Batch no actualiza automáticamente la Kubernetes versión. Ejecute el siguiente comando para actualizar la Kubernetes versión del entorno de su ordenador a **1.32**.

```
$ aws batch update-compute-environment \  
  --compute-environment <compute-environment-name> \  
  --compute-resources \  
    'ec2Configuration=[{imageType=EKS_AL2,imageKubernetesVersion=1.32}]'
```

Al actualizar a una AMI o a una Kubernetes versión más reciente, puede especificar si desea finalizar los trabajos cuando se actualicen (`terminateJobsOnUpdate`) y cuánto tiempo esperar antes de reemplazar una instancia si los trabajos en ejecución no terminan (`jobExecutionTimeoutMinutes`.) Para obtener más información, consulte [Actualizar un entorno informático en AWS Batch](#) y la política de actualización de la infraestructura ([UpdatePolicy](#)) establecida en la operación [UpdateComputeEnvironment](#) de la API.

Ejecute un DaemonSet nodo AWS Batch gestionado

AWS Batch establece manchas en los nodos gestionados. AWS Batch Kubernetes Puede seleccionar a DaemonSet para que se ejecute en nodos AWS Batch gestionados con lo siguiente `tolerations`.

```
tolerations:  
  - key: "batch.amazonaws.com/batch-node"  
    operator: "Exists"
```

Otra forma de hacerlo es con lo siguiente `tolerations`.

```
tolerations:
- key: "batch.amazonaws.com/batch-node"
  operator: "Exists"
  effect: "NoSchedule"
- key: "batch.amazonaws.com/batch-node"
  operator: "Exists"
  effect: "NoExecute"
```

Personalización de plantillas de lanzamiento de Amazon EKS

AWS Batch en Amazon EKS admite plantillas de lanzamiento. Existen limitaciones en cuanto a lo que puede hacer su plantilla de lanzamiento.

Important

- En el caso de EKS AL2 AMIs, AWS Batch ejecuta `/etc/eks/bootstrap.sh`. No ejecute `/etc/eks/bootstrap.sh` en su plantilla de lanzamiento ni en sus scripts cloud-init user-data. Puede añadir parámetros adicionales además del parámetro `--kubelet-extra-args` a [bootstrap.sh](#). Para ello, defina la variable `AWS_BATCH_KUBELET_EXTRA_ARGS` en el archivo `/etc/aws-batch/batch.config`. Consulte el ejemplo siguiente para obtener información detallada.
- En el caso de EKS AL2023, AWS Batch utiliza el comando [NodeConfigSpec](#) de EKS para hacer que las instancias se unan al clúster de EKS. AWS Batch rellena [ClusterDetails](#) el clúster [NodeConfigSpec](#) de EKS y no es necesario especificarlos.

Note

Le recomendamos que no establezca ninguno de los siguientes [NodeConfigSpec](#) ajustes en la plantilla de lanzamiento, ya que AWS Batch anulará sus valores. Para obtener más información, consulte [Responsabilidad compartida de los nodos Kubernetes](#).

- Taints
- Cluster Name
- apiServerEndpoint
- certificatAuthority
- CIDR

- No cree etiquetas con el prefijo `batch.amazonaws.com/`

Note

Si la plantilla de lanzamiento se cambia después de [CreateComputeEnvironment](#) llamarla, [UpdateComputeEnvironment](#) hay que llamarla para evaluar la versión de la plantilla de lanzamiento que se va a reemplazar.

Temas

- [Adición de argumentos adicionales kubelet](#)
- [Configuración del tiempo de ejecución del contenedor](#)
- [Montaje de un volumen de Amazon EFS](#)
- [IPv6 soporte](#)

Adición de argumentos adicionales **kubelet**

AWS Batch admite la adición de argumentos adicionales al `kubelet` comando. Para ver una lista completa de parámetros que admite [kubelet](#), consulte la Documentación de Kubernetes. En el siguiente ejemplo, para EKS AL2 AMIs, `--node-labels mylabel=helloworld` se agrega a la línea de `kubelet` comandos.

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary=="MYBOUNDARY=="

--MYBOUNDARY--
Content-Type: text/x-shellscript; charset="us-ascii"

#!/bin/bash
mkdir -p /etc/aws-batch

echo AWS_BATCH_KUBELET_EXTRA_ARGS="\--node-labels mylabel=helloworld\" >> /etc/
aws-batch/batch.config

--MYBOUNDARY--
```

Para EKS, AL2023 AMIs el formato de archivo es YAML. Para ver una lista completa de parámetros que admite [NodeConfigSpec](#), consulte la Documentación de Kubernetes. En el siguiente ejemplo, para EKS AL2023 AMIs, `--node-labels mylabel=helloworld` se agrega a la línea de `kubelet` comandos.

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary=="MYBOUNDARY=="

--MYBOUNDARY==
Content-Type: application/node.eks.aws

apiVersion: node.eks.aws/v1alpha1
kind: NodeConfig
spec:
  kubelet:
    flags:
      - --node-labels=mylabel=helloworld

--MYBOUNDARY==
```

Configuración del tiempo de ejecución del contenedor

Puede utilizar la variable de AWS Batch `CONTAINER_RUNTIME` entorno para configurar el tiempo de ejecución del contenedor en un nodo gestionado. En el siguiente ejemplo, se establece el tiempo de ejecución del contenedor en `containerd` cuando se ejecuta `bootstrap.sh`. Para obtener más información, consulte [containerd](#) en la documentación de Kubernetes.

Si utiliza una AMI optimizada de `EKS_AL2023` o una AMI de `EKS_AL2023_NVIDIA`, no necesita especificar el tiempo de ejecución del contenedor, ya que solo se admite `containerd`.

Note

La variable de entorno de `CONTAINER_RUNTIME` equivale a la opción `--container-runtime` de `bootstrap.sh`. Para obtener más información, consulte [Options](#) en la documentación de Kubernetes.

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary=="MYBOUNDARY=="
```

```

---MYBOUNDARY==
Content-Type: text/x-shellscript; charset="us-ascii"

#!/bin/bash
mkdir -p /etc/aws-batch

echo CONTAINER_RUNTIME=containerd >> /etc/aws-batch/batch.config

---MYBOUNDARY===

```

Montaje de un volumen de Amazon EFS

Puede utilizar plantillas de lanzamiento para montar los volúmenes en el nodo. En el siguiente ejemplo, se utilizan los ajustes `cloud-config` `packages` y `runcmd`. Para obtener más información, consulte [ejemplos de configuración de nube](#) en la `cloud-init` documentación.

```

MIME-Version: 1.0
Content-Type: multipart/mixed; boundary==="MYBOUNDARY=="

---MYBOUNDARY==
Content-Type: text/cloud-config; charset="us-ascii"

packages:
- amazon-efs-utils

runcmd:
- file_system_id_01=fs-abcdef123
- efs_directory=/mnt/efs

- mkdir -p ${efs_directory}
- echo "${file_system_id_01}:/ ${efs_directory} efs _netdev,noresvport,tls,iam 0 0"
  >> /etc/fstab
- mount -t efs -o tls ${file_system_id_01}:/ ${efs_directory}

---MYBOUNDARY===

```

Para usar este volumen en el trabajo, debe añadirse al parámetro [eksProperties](#) a. [RegisterJobDefinition](#) El siguiente ejemplo es una gran parte de la definición del trabajo.

```

{
  "jobDefinitionName": "MyJobOnEks_EFS",
  "type": "container",

```



```
"eksProperties": {
  "podProperties": {
    "containers": [
      {
        "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
        "command": ["ls", "-la", "/efs"],
        "resources": {
          "limits": {
            "cpu": "1",
            "memory": "1024Mi"
          }
        },
        "volumeMounts": [
          {
            "name": "efs-volume",
            "mountPath": "/efs"
          }
        ]
      }
    ],
    "volumes": [
      {
        "name": "efs-volume",
        "hostPath": {
          "path": "/mnt/efs"
        }
      }
    ]
  }
}
```

En el nodo, el volumen de Amazon EFS está montado en el directorio `/mnt/efs`. En el contenedor del trabajo de Amazon EKS, el volumen se monta en el directorio `/efs`.

IPv6 soporte

AWS Batch admite clústeres de Amazon EKS que tienen IPv6 direcciones. No se requieren personalizaciones para obtener AWS Batch soporte. Sin embargo, antes de empezar, le recomendamos que revise las consideraciones y condiciones que se describen en [Asignación de IPv6 direcciones a pods y servicios](#) de la Guía del usuario de Amazon EKS.

¿Cómo actualizar de EKS AL2 a EKS AL2023

Los Amazon EKS optimizados AMIs están disponibles en dos familias basadas en Amazon Linux 2 (AL2) y Amazon Linux 2023 (AL2023). AL2023 es un sistema operativo basado en Linux diseñado para proporcionar un entorno seguro, estable y de alto rendimiento para sus aplicaciones en la nube. Para obtener más información sobre las diferencias entre Amazon Linux 2 AL2 y Amazon Linux 2023, AL2023 consulte [Actualización de Amazon Linux 2 a Amazon Linux 2023](#) en la Guía del usuario de Amazon EKS.

Important

AWS finalizará el soporte para Amazon EKS AL2 (optimizado y AL2 acelerado) a partir del 26 de noviembre de 2025. Recomendamos migrar los entornos de cómputo de AWS Batch Amazon EKS a Amazon Linux 2023 antes del 26/11/25 para mantener un rendimiento y una seguridad óptimos. Si bien puede seguir utilizando Amazon Linux 2 optimizado para Amazon EKS AWS Batch proporcionado AMIs en sus entornos informáticos de Amazon EKS más allá de la end-of-support fecha del 26 de noviembre del 25 de noviembre, estos entornos informáticos ya no recibirán nuevas actualizaciones de software, parches de seguridad o correcciones de errores. AWS Posteriormente, es su [responsabilidad mantener](#) estos entornos informáticos en la AMI Amazon Linux 2 optimizada para Amazon EKS end-of-life.

En función de cómo esté configurado su entorno informático, puede utilizar una de las siguientes rutas de actualización de AL2 a AL2023.

Actualice mediante `Ec2Configuration`. `ImageType`

- Si no está utilizando una plantilla de lanzamiento o anulaciones de plantillas de lanzamiento, cambie [Ec2Configuration](#). `ImageType` a `EKS_AL2023` o y, a continuación, `EKS_AL2023_NVIDIA` ejecute. [UpdateComputeEnvironment](#)
- Si especifica una configuración [Ec2](#). `ImageIdOverride` [luego Ec2 Configuration](#). `ImageType` debe coincidir con el tipo de AMI especificado en [Ec2Configuration](#). `ImageIdOverride`.

Si `ImageIdOverride` y `ImageType` no coinciden, el nodo no se unirá al clúster.

Actualización mediante plantillas de lanzamiento

- Si tienes algún argumento `kubelet` adicional definido en una plantilla de lanzamiento o en una modificación de la plantilla de lanzamiento, tendrás que actualizarlo al nuevo [formato de argumentos kubelet adicionales](#).

Si no coincide con el formato de argumentos adicionales `kubelet`, los argumentos adicionales no se aplicarán.

- Porque AL2023 AMIs `containerd` es el único tiempo de ejecución de contenedores compatible. No es necesario especificar el tiempo de ejecución del contenedor en `EKS_AL2023` para la plantilla de lanzamiento.

No puedes especificar un tiempo de ejecución de contenedor personalizado con `EKS_AL2023`.

- Si utiliza una plantilla de lanzamiento o una anulación de la plantilla de lanzamiento que especifique una AMI según `EKS_AL2023`, debe establecer [userDataType](#) en `EKS_NODEADM`.

Si `userDataType` y la AMI no coinciden, el nodo no se unirá al clúster de EKS.

Entornos de servicio para AWS Batch

Los entornos de servicio AWS Batch permiten la integración con la SageMaker IA. Un entorno de servicios contiene los parámetros de configuración específicos de la SageMaker IA necesarios AWS Batch para enviar y gestionar los trabajos de SageMaker formación, al tiempo que proporciona funciones AWS Batch de gestión de colas, programación y prioridades.

Con los entornos de servicio, los científicos de datos y los ingenieros de aprendizaje automático pueden enviar trabajos de SageMaker formación con prioridades a las colas de trabajos de servicio. Esta integración elimina la necesidad de coordinar las cargas de trabajo de machine learning de forma manual, así como evitar gastos excesivos accidentales y mejorar la utilización de los recursos en los flujos de trabajo de machine learning de su organización.

Temas

- [¿En qué se encuentran los entornos de servicio AWS Batch](#)
- [Los estados y el ciclo de vida del entorno de servicio en AWS Batch](#)
- [Cree un entorno de servicios en AWS Batch](#)
- [Actualice un entorno de servicios en AWS Batch](#)
- [Elimine un entorno de servicios en AWS Batch](#)

¿En qué se encuentran los entornos de servicio AWS Batch

Un entorno de servicios es un AWS Batch recurso que contiene los parámetros de configuración necesarios para integrarse AWS Batch con la SageMaker IA. Los entornos de servicio permiten AWS Batch enviar y gestionar los trabajos de SageMaker formación y, al mismo tiempo, proporcionan AWS Batch funciones de gestión de colas, programación y prioridades.

Los entornos de servicio abordan los desafíos comunes a los que se enfrentan los equipos de ciencia de datos a la hora de administrar las cargas de trabajo de machine learning. Las organizaciones suelen limitar la cantidad de instancias disponibles para los modelos de formación a fin de evitar gastos excesivos accidentales, cumplir con las restricciones presupuestarias, ahorrar costos con las instancias reservadas o utilizar tipos de instancias específicos para las cargas de trabajo. Sin embargo, con las instancias asignadas, los científicos de datos desean ejecutar más cargas de trabajo simultáneas de las que es posible, lo que requiere una coordinación manual para decidir qué cargas de trabajo se ejecutan y cuándo.

Este desafío de coordinación afecta a organizaciones de todos los tamaños: desde equipos con solo unos pocos científicos de datos hasta operaciones a gran escala. Así como las organizaciones crecen, también lo hace la complejidad, lo cual significa que se necesitará más tiempo para administrar la coordinación de la carga de trabajo y la participación frecuente del administrador de la infraestructura. Se pierde el tiempo con estos esfuerzos manuales y se reduce la eficiencia de las instancias, lo que genera costos reales para los clientes.

Con los entornos de servicio, los científicos de datos y los ingenieros de aprendizaje automático pueden enviar los trabajos de SageMaker formación con prioridades a colas configurables, lo que garantiza que las cargas de trabajo se ejecuten automáticamente sin intervención en cuanto los recursos estén disponibles. Esta integración aprovecha las amplias capacidades AWS Batch de preparación de colas y programación, lo que permite a los clientes personalizar sus políticas de colas y programación para adaptarlas a los objetivos de su organización.

Cómo funcionan los entornos de servicio con otros componentes AWS Batch

Los entornos de servicio se integran con otros AWS Batch componentes para permitir la creación de colas de trabajos de SageMaker formación:

- Colas de trabajos: los entornos de servicio están asociados a las colas de trabajos para permitir que la cola procese los trabajos de servicio para el trabajo de formación SageMaker
- Trabajos de servicio: al enviar un trabajo de servicio a una cola asociada a un entorno de servicio, AWS Batch utiliza la configuración del entorno para enviar el trabajo de formación correspondiente SageMaker
- Políticas de programación: los entornos de servicio funcionan con políticas AWS Batch de programación para priorizar y gestionar el orden de ejecución de los trabajos de SageMaker formación

Esta integración le permite aprovechar las capacidades avanzadas AWS Batch de preparación de colas y programación y, al mismo tiempo, mantener toda la funcionalidad y flexibilidad de los trabajos de SageMaker formación.

Prácticas recomendadas para los entornos de servicios

Los entornos de servicio ofrecen capacidades para gestionar los trabajos SageMaker de formación a gran escala. Con estas prácticas recomendadas podrá optimizar los costos, el rendimiento y la

eficiencia operativa, mientras que se evitan los problemas de configuración habituales que pueden afectar a los flujos de trabajo de machine learning.

Al planificar la capacidad del entorno de servicios, tenga en cuenta las cuotas y los límites específicos que se aplican a las colas de trabajos de SageMaker formación. Cada entorno de servicio tiene un límite de capacidad máximo expresado en el número de instancias, que controla directamente cuántos trabajos de SageMaker formación se pueden ejecutar simultáneamente. Comprender estos límites ayuda a evitar la contención de recursos y garantiza tiempos de ejecución de los trabajos predecibles.

El rendimiento óptimo del entorno de servicio depende de la comprensión de las características únicas de la programación de los trabajos de SageMaker formación. A diferencia de los trabajos contenerizados tradicionales, los trabajos de servicio pasan de un SCHEDULED estado a otro, mientras que la SageMaker IA adquiere y proporciona las instancias de formación necesarias. Esto significa que las horas de inicio de los trabajos pueden variar considerablemente según la disponibilidad de las instancias y de la capacidad regional.

Important

Los entornos de servicio tienen cuotas específicas que pueden afectar a tu capacidad de escalar las cargas de trabajo de SageMaker formación. Se pueden crear hasta 50 entornos de servicio por cuenta, y cada cola de trabajos solo admite un entorno de servicio asociado. Además, la carga útil de la solicitud de servicio para trabajos individuales tiene un límite de 10 KiB y la API `SubmitServiceJob` tiene un límite de 5 transacciones por segundo por cuenta. La comprensión de estos límites durante la planificación de la capacidad evitará las restricciones de escalamiento inesperadas.

La supervisión eficaz de los entornos de servicio requiere prestar atención tanto AWS Batch a las métricas de los servicios de SageMaker inteligencia artificial como a las métricas de los servicios. [Las transiciones del estado de trabajo](#) ofrecen información valiosa sobre el rendimiento del sistema, sobre todo el tiempo empleado en el estado SCHEDULED, el cual indica los patrones de disponibilidad de la capacidad. Los entornos de servicio mantienen sus propios estados de ciclo de vida, similares a los de los entornos de computación, que pasan por los estados CREATING, VALID, INVALID y DELETING, que deben supervisarse para verificar su estado operativo. A fin de optimizar las configuraciones de los entornos de servicios a lo largo del tiempo, las organizaciones con prácticas de monitoreo consolidadas suelen realizar un seguimiento de la profundidad de las colas, las tasas de finalización de los trabajos y los patrones de utilización de las instancias.

Los estados y el ciclo de vida del entorno de servicio en AWS Batch

Los entornos de servicio mantienen estados del ciclo de vida que indican su estado operativo actual y su preparación para procesar los trabajos de SageMaker formación. Comprender estos estados ayuda a supervisar el estado del entorno de servicio, solucionar problemas de configuración y garantizar un procesamiento fiable de los trabajos. El sistema de gestión estatal sigue los patrones establecidos en los entornos informáticos y, al mismo tiempo, se adapta a los requisitos únicos de la integración laboral de la SageMaker formación.

Los estados del entorno de servicio se administran automáticamente en AWS Batch función de la validación de la configuración, la disponibilidad de los recursos y las comprobaciones del estado operativo. A diferencia de los entornos informáticos que gestionan la infraestructura física, los entornos de servicio se centran en la validación de la configuración y en la preparación para la integración con los servicios de SageMaker IA. Las transiciones de estado permiten comprobar si su entorno de servicio puede enviar y gestionar correctamente los trabajos de SageMaker formación.

Definiciones de estado del entorno de servicios

Los entornos de servicio pueden estar en uno de los cuatro estados posibles que indican su estado operativo actual y su preparación para procesar los trabajos de SageMaker formación. Cada estado representa una fase específica del ciclo de vida del entorno de servicio: desde la creación inicial hasta la preparación operativa y, finalmente, su eliminación. En la siguiente tabla se describe cada estado y su significado:

Estado	Description (Descripción)
CREATING	Estado inicial al momento de crear un entorno de servicios. Durante este estado, AWS Batch valida los parámetros de configuración y establece la integración con los servicios de SageMaker IA. El entorno de servicio no puede procesar los trabajos, y las colas de trabajos asociadas a él no aceptarán el envío de trabajos de servicio. Es normal que en los entornos de servicio correctamente configura

Estado	Description (Descripción)
	dos, el proceso de creación se complete en unos segundos.
VALID	El estado operativo indica que el entorno de servicio ha superado todas las comprobaciones de validación de la configuración y está listo para procesar las tareas de SageMaker formación. Este estado indica que la configuración del entorno de servicio es correcta, que se han establecido todos los permisos necesarios y que AWS Batch puede enviar correctamente los trabajos a SageMaker AI en su nombre. Los entornos de servicio pasan la mayor parte de su ciclo de vida operativo en este estado.
INVALID	Un estado que indica que el entorno de servicio ha detectado un problema de configuración o permisos que le impide procesar los trabajos de SageMaker formación. Las colas de trabajos asociadas a entornos de servicio no válidos no pueden procesar los nuevos envíos de trabajos de servicio hasta que se resuelvan los problemas subyacentes.
DELETING	El estado que se produce cuando se solicita la eliminación de un entorno de servicios. Durante este estado, se AWS Batch asegura de que no haya ningún trabajo de SageMaker formación activo asociado al entorno y realiza las operaciones de limpieza necesarias. Los entornos de servicio en este estado no pueden procesar la presentación de nuevos trabajos, y el proceso de eliminación finaliza una vez que todos los recursos asociados se limpien correctamente.

Transiciones de estado del entorno de servicio

Las transiciones de estado del entorno de servicio se producen automáticamente según los cambios de configuración, los resultados de la validación y la supervisión del estado operativo. El AWS Batch servicio supervisa continuamente el estado del entorno de servicio y actualiza los estados en consecuencia. Comprender estas transiciones ayuda a anticipar cuándo se efectuarán los cambios de configuración y a resolver los problemas que provocan estados no válidos.

Tras una creación y validación correctas, los entornos de servicio pasan de CREATING a VALID. Esta transición confirma que todos los parámetros de configuración son correctos, que los permisos de IAM necesarios están configurados correctamente y que el entorno de servicios se puede integrar correctamente con los servicios de SageMaker IA. Una vez en el estado VALID, las colas de trabajos asociadas pueden empezar a procesar los envíos de trabajos de servicio.

Los entornos de servicio pasan de VALID a INVALID cuando se produce un error en la validación de la configuración o cuando las dependencias dejan de estar disponibles. Esto puede suceder a causa de modificaciones en el rol de IAM, variaciones en los límites de capacidad que infrinjan las cuotas o modificaciones de recursos externos que impacten la capacidad de funcionamiento del entorno de servicios. El campo que indica el motivo del estado entrega detalles específicos sobre la causa del estado no válido.

Los entornos de servicio pueden pasar de VALID a INVALID una vez que se hayan resuelto los problemas subyacentes. Esto puede implicar actualizar los permisos de IAM, corregir las configuraciones de capacidad o restablecer el acceso a los recursos necesarios. AWS Por lo general, la transición se produce automáticamente una vez que AWS Batch detecta que se han solucionado los problemas de configuración.

Cree un entorno de servicios en AWS Batch

Antes de poder ejecutar trabajos de SageMaker formación AWS Batch, debe crear un entorno de servicios. Puede crear un entorno de servicios que contenga los parámetros de configuración necesarios AWS Batch para integrarse con los servicios de SageMaker IA y enviar los trabajos de SageMaker formación en su nombre.

Requisitos previos

Asegúrese disponer de lo siguiente antes de crear un entorno de servicio:

- Permisos de IAM: permisos para crear y administrar entornos de servicios. Para obtener más información, consulte [AWS Batch Políticas, funciones y permisos de IAM](#).

Create a service environment (AWS Console)

Utilice la AWS Batch consola para crear un entorno de servicios a través de la interfaz web.

Cómo crear un entorno de servicio

1. Abra la AWS Batch consola en <https://console.aws.amazon.com/batch/>.
2. En el panel de navegación, elija Entornos.
3. Elija Crear entorno y, a continuación, seleccione Entorno de servicio.
4. Para la configuración del entorno de servicio, elija SageMaker AI.
5. En Nombre, ingrese un nombre único para el entorno de servicios. Caracteres válidos: a-z, A-Z, 0-9, - (guion) y _ (guion bajo).
6. En Cantidad máxima de instancias, ingrese la cantidad máxima de instancias de entrenamiento simultáneas.
7. (Opcional) Para agregar etiquetas, elija Agregar etiqueta e ingrese pares clave-valor.
8. Elija Siguiente.
9. Revise los detalles del nuevo entorno de servicio y elija Crear entorno de servicio.

Create a service environment (AWS CLI)

Utilice el `create-service-environment` comando para crear un entorno de servicio con la AWS CLI.

Cómo crear un entorno de servicio

1. Cree un entorno de servicios con los parámetros básicos necesarios:

```
aws batch create-service-environment \
  --service-environment-name my-sagemaker-service-env \
  --service-environment-type SAGEMAKER_TRAINING \
  --capacity-limits capacityUnit=NUM_INSTANCES,maxCapacity=10
```

2. (Opcional) Cree un entorno de servicio con etiquetas:

```
aws batch create-service-environment \
```

```
--service-environment-name my-sagemaker-service-env \  
--service-environment-type SAGEMAKER_TRAINING \  
--capacity-limits capacityUnit=NUM_INSTANCES,maxCapacity=10 \  
--tags team=data-science,project=ml-training
```

3. Compruebe que el entorno de servicio se haya creado correctamente:

```
aws batch describe-service-environments \  
--service-environment my-sagemaker-service-env
```

El entorno de servicio aparece en la lista de entornos con un estado de CREATING. Cuando la creación se completa correctamente, el estado cambia a VALID y el entorno de servicio está listo para que se le agregue una cola de trabajos de servicio para que el entorno de servicio pueda empezar a procesar los trabajos.

Actualice un entorno de servicios en AWS Batch

Actualice un entorno de servicio para ajustar sus límites de capacidad, cambiar su estado operativo o actualizar las etiquetas asociadas a los recursos. Las actualizaciones del entorno de servicio le permiten ajustar la capacidad a medida que sus requisitos de carga de trabajo de SageMaker formación cambian o modificar la configuración operativa sin volver a crear el entorno. Antes de actualizar un entorno de servicio, comprenda los parámetros que se pueden modificar y el impacto de los cambios en las tareas en ejecución.

Puede cambiar los límites de capacidad, el estado o las etiquetas de un entorno de servicios.

Update a service environment (AWS Console)

Utilice la AWS Batch consola para actualizar un entorno de servicios a través de la interfaz web.

Cómo actualizar un entorno de servicio

1. Abra la AWS Batch consola en <https://console.aws.amazon.com/batch/>.
2. En el panel de navegación, elija Entornos.
3. Seleccione la pestaña Entorno de servicio.
4. Elija el entorno de servicio que desee actualizar.
5. Elija Acciones y luego seleccione cualquiera de las dos opciones:

- Estado: seleccione Habilitar o Deshabilitar para cambiar el estado.
- Límite de capacidad: modifique la cantidad máxima de instancias

6. Seleccione Guardar cambios para aplicar los cambios.

El entorno de servicio se actualiza inmediatamente. Compruebe los detalles del entorno para confirmar que los cambios se aplicaron correctamente. Si ha deshabilitado el entorno de servicio, las colas de trabajos asociadas dejarán de procesar los nuevos envíos de trabajos de servicio hasta que se vuelva a habilitar.

Update a service environment (AWS CLI)

Utilice el `update-service-environment` comando para modificar un entorno de servicio con la AWS CLI.

Cómo actualizar los límites de capacidad del entorno de servicios

1. Actualice el límite de capacidad de un entorno de servicios:

```
aws batch update-service-environment \
  --service-environment my-sagemaker-service-env \
  --capacity-limits capacityUnit=NUM_INSTANCES,maxCapacity=20
```

2. Compruebe que la actualización se haya aplicado correctamente:

```
aws batch describe-service-environments \
  --service-environments my-sagemaker-service-env
```

Cómo actualizar el estado del entorno de servicio

1. Deshabilite un entorno de servicio para dejar de procesar nuevos trabajos:

```
aws batch update-service-environment \
  --service-environment my-sagemaker-service-env \
  --state DISABLED
```

2. Vuelva a habilitar un entorno de servicio para reanudar el procesamiento:

```
aws batch update-service-environment \
  --service-environment my-sagemaker-service-env \
```

```
--state ENABLED
```

Las actualizaciones del entorno de servicio se aplican de inmediato. Supervise el estado del entorno de servicio para garantizar que las actualizaciones se completen correctamente antes de enviar nuevos trabajos.

Elimine un entorno de servicios en AWS Batch

Puede eliminar un entorno de servicios cuando ya no lo necesite para sus trabajos de SageMaker formación. Con la eliminación de un entorno de servicio, se elimina la configuración e impide que se envíen más trabajos. Antes de eliminar un entorno de servicios, asegúrese de que ningún trabajo de SageMaker formación activo dependa de él y de que no haya colas de trabajos asociadas al entorno de servicio.

Important

La eliminación del entorno de servicio es permanente. Una vez eliminado, no se podrá recuperar el entorno de servicio ni su configuración. Si más adelante necesita una funcionalidad similar, deberá crear un nuevo entorno de servicio con la configuración necesaria. Tenga presente deshabilitar el entorno de servicio en lugar de eliminarlo si necesita reactivarlo más adelante.

Note

Al eliminar todos los entornos de servicio de su cuenta, no se elimina automáticamente el rol vinculado al servicio creado para AWS Batch la integración de la IA. SageMaker El rol vinculado al servicio permanece disponible para la creación de futuros entornos de servicios. Si desea eliminar el rol vinculado al servicio, debe eliminarlo por separado mediante IAM después de verificar de que no existan entornos de servicios en su cuenta.

Requisitos previos de eliminación

Antes de poder eliminar un entorno de servicios, debe desasociar cualquier cola de trabajos de servicio y, a continuación, deshabilitar el entorno de servicio.

Antes de eliminar un entorno de servicios:

- Compruebe los trabajos activos: asegúrese de que actualmente no haya ningún trabajo de SageMaker formación en ejecución en el entorno de servicio.
- revise las colas de trabajos: identifique las colas de trabajos asociadas al entorno de servicio y asóciela a un entorno de servicio diferente, o deshabilite y elimine la cola de trabajos.

Administración de colas de trabajos: las colas de trabajos que estaban asociadas a un entorno de servicios eliminado pueden seguir existiendo, pero no pueden procesar trabajos de servicio. Por lo tanto, debe eliminar las colas de trabajos no utilizadas o asociarlas a un entorno de servicio diferente antes de eliminar el entorno de servicio original.

Delete a service environment (AWS Console)

Utilice la AWS Batch consola para eliminar un entorno de servicios a través de la interfaz web.

Cómo eliminar un entorno de servicio

1. Abra la AWS Batch consola en <https://console.aws.amazon.com/batch/>.
2. En el panel de navegación, elija Entornos.
3. Elija la pestaña Entorno de servicio y, a continuación, seleccione un entorno de servicio.
4. Si el entorno de servicios está habilitado, elija Acciones y, a continuación, Deshabilitar.
5. Una vez que el entorno de servicio esté deshabilitado, elija Acciones y, a continuación, Eliminar.
6. En el cuadro de diálogo de confirmación, elija Confirmar.

El entorno de servicio muestra un estado de DELETING mientras se produce la eliminación. Una vez completada la eliminación, el entorno de servicio desaparecerá de la lista de entornos.

Delete a service environment (AWS CLI)

Utilice el `delete-service-environment` comando para eliminar un entorno de servicio con la AWS CLI.

Cómo eliminar un entorno de servicio

1. Compruebe si hay colas de trabajos asociadas al entorno de servicio:

```
aws batch describe-job-queues
```

Si hay colas de trabajos asociadas al entorno de servicio, puede [disociar la cola de trabajos](#) del entorno de servicio y asociarla a un entorno de servicio diferente, o bien eliminar la cola de trabajos.

2. Deshabilite el entorno de servicio:

```
aws batch update-service-environment \  
  --service-environment my-sagemaker-service-env \  
  --state DISABLED
```

3. Elimine el entorno de servicio:

```
aws batch delete-service-environment \  
  --service-environment my-sagemaker-service-env
```

4. Supervise el proceso de eliminación:

```
aws batch describe-service-environments \  
  --service-environment my-sagemaker-service-env
```

El entorno de servicio pasa a estado DELETING durante el proceso de eliminación. Una vez completada la eliminación, el entorno de servicio dejará de figurar en la lista de operaciones descritas. Las colas de trabajos asociadas permanecen allí, pero no pueden procesar los trabajos de servicio hasta que se asocien a un entorno de servicio diferente.

Colas de trabajos

Los trabajos se envían a una cola de trabajos, donde permanecen hasta que pueden programarse para ejecutarse en un entorno de computación. Una AWS cuenta puede tener varias colas de trabajos. Por ejemplo, puede crear una cola que utilice instancias bajo demanda de Amazon EC2 para trabajos de alta prioridad y otra cola que utilice instancias de spot de Amazon EC2 para trabajos de baja prioridad. Las colas de trabajo tienen una prioridad que el programador utiliza para determinar qué cola debe evaluarse primero para su ejecución.

Temas

- [Creación de una cola de trabajos](#)
- [Ver una lista de trabajos en espera en AWS Batch](#)
- [Eliminación de una cola de trabajos en AWS Batch](#)
- [Políticas de programación de reparto justo](#)
- [Programación basada en los recursos](#)
- [Gestión de cuotas](#)

Creación de una cola de trabajos

Antes de enviar trabajos en AWS Batch, es necesario crear una cola de trabajo. Al crear una cola de trabajos, se asocian uno o varios entornos de computación a la cola y se asigna un orden de preferencia.

También se establece la prioridad de la cola de trabajos que determina el orden en que el programador de lotes AWS coloca los trabajos. Esto significa que, si un entorno de computación está asociado a más de una cola de trabajos, se da preferencia a la cola de trabajos con mayor prioridad.

Temas

- [Cómo crear una cola de Amazon EC2](#)
- [Cómo crear una cola de trabajos de Fargate](#)
- [Cómo crear una cola de trabajos de Amazon EKS](#)
- [Creación de una cola de trabajos de entrenamiento de SageMaker en AWS Batch](#)
- [Plantilla de cola de trabajos](#)

Cómo crear una cola de Amazon EC2

Siga estos pasos para crear una cola de trabajos de Amazon Elastic Compute Cloud (Amazon EC2).

Para crear una cola de Amazon EC2

1. Abra la consola AWS Batch en <https://console.aws.amazon.com/batch/>.
2. En la barra de navegación, elija la Región de AWS a utilizar.
3. En el panel de navegación, elija Colas de trabajos.
4. Seleccione Creación de.
5. Para el Tipo de orquestación, seleccione Amazon Elastic Compute Cloud (Amazon EC2).
6. En Nombre, escriba un nombre único para su cola de trabajos. El nombre puede tener una longitud máxima de 128 caracteres y puede contener mayúsculas y minúsculas, números y guiones bajos (_).
7. En Prioridad, introduzca un valor entero para la prioridad de la cola de trabajos. Las colas de trabajo con mayor prioridad se ejecutan antes que con las colas de menor prioridad asociadas con un mismo entorno de computación. La prioridad se determina en orden descendente. Por ejemplo, una cola de trabajos con una prioridad 10, tendrá mayor preferencia de programación que una cola de trabajos cuyo valor de prioridad sea 1.
8. (Opcional) En Política de programación Nombre de recurso de Amazon (ARN), elija una política de programación existente.
9. En Entornos de computación conectados, seleccione uno o varios entornos de computación de la lista para asociarlos a la cola de trabajos. Seleccione los entornos de computación en el orden en que desee que la cola intente colocarlos en la cola de trabajos. El programador de trabajos utiliza el orden en el que selecciona los entornos de computación para determinar qué entorno de computación inicia un trabajo determinado. Antes de asociarlos a una cola de trabajos, los entornos de computación deben estar en estado VALID. Puede asociar hasta tres entornos de computación con una cola de trabajos. Si no tiene un entorno de computación existente, elija Creación de entorno de computación

Note

Todos los entornos de computación asociados a una cola de trabajos deben compartir el mismo modelo de aprovisionamiento. AWS Batch no admite la mezcla de modelos de aprovisionamiento en una sola cola de trabajos.

10. En Orden de entorno de computación, seleccione las flechas arriba y abajo para configurar el orden que desee.
11. Seleccione Crear cola de trabajos para finalizar y crear su cola de trabajos.

Cómo crear una cola de trabajos de Fargate

Siga estos pasos para crear una cola de trabajos de AWS Fargate.

Para crear una cola de trabajos Fargate

1. Abra la consola AWS Batch en <https://console.aws.amazon.com/batch/>.
2. En la barra de navegación, elija la Región de AWS a utilizar.
3. En el panel de navegación, elija Colas de trabajos.
4. Seleccione Creación de.
5. En Tipo de orquestación, elija Fargate.
6. En Nombre, escriba un nombre único para su cola de trabajos. El nombre puede tener una longitud máxima de 128 caracteres y puede contener mayúsculas y minúsculas, números y guiones bajos (_).
7. En Prioridad, introduzca un valor entero para la prioridad de la cola de trabajos. Las colas de trabajo con mayor prioridad se ejecutan antes que con las colas de menor prioridad asociadas con un mismo entorno de computación. La prioridad se determina en orden descendente. Por ejemplo, una cola de trabajos con una prioridad 10, tendrá mayor preferencia de programación que una cola de trabajos cuyo valor de prioridad sea 1.
8. (Opcional) En Política de programación Nombre de recurso de Amazon (ARN), elija una política de programación existente.
9. En Entornos de computación conectados, seleccione uno o varios entornos de computación de la lista para asociarlos a la cola de trabajos. Seleccione los entornos de computación en el orden en que desee que la cola intente colocarlos en la cola de trabajos. El programador de trabajos utiliza el orden en el que selecciona los entornos de computación para determinar qué entorno de computación inicia un trabajo determinado. Antes de asociarlos a una cola de trabajos, los entornos de computación deben estar en estado VALID. Puede asociar hasta tres entornos de computación con una cola de trabajos.

 Note

Todos los entornos de computación asociados a una cola de trabajos deben compartir el mismo modelo de aprovisionamiento. AWS Batch no admite la mezcla de modelos de aprovisionamiento en una sola cola de trabajos.

10. En Orden de entorno de computación, seleccione las flechas arriba y abajo para configurar el orden que desee.
11. Seleccione Crear cola de trabajos para finalizar y crear su cola de trabajos.

Cómo crear una cola de trabajos de Amazon EKS

Siga estos pasos para crear una cola de trabajos de Amazon Elastic Kubernetes Service (Amazon EKS).

Para crear una cola de trabajos de Amazon EKS

1. Abra la consola AWS Batch en <https://console.aws.amazon.com/batch/>.
2. En la barra de navegación, elija la Región de AWS a utilizar.
3. En el panel de navegación, elija Colas de trabajos.
4. Seleccione Creación de.
5. Para el Tipo de orquestación, elija Amazon Elastic Kubernetes Service (Amazon EKS).
6. En Nombre, escriba un nombre único para su cola de trabajos. El nombre puede tener una longitud máxima de 128 caracteres y puede contener mayúsculas y minúsculas, números y guiones bajos (_).
7. En Prioridad, escriba un valor entero para la prioridad de la cola de trabajos. Las colas de trabajo con mayor prioridad se ejecutan antes que con las colas de menor prioridad asociadas con un mismo entorno de computación. La prioridad se determina en orden descendente. Por ejemplo, una cola de trabajos con una prioridad 10, tendrá mayor preferencia de programación que una cola de trabajos cuyo valor de prioridad sea 1.
8. (Opcional) En Política de programación Nombre de recurso de Amazon (ARN), elija una política de programación existente.
9. En Entornos de computación conectados, seleccione uno o varios entornos de computación de la lista para asociarlos a la cola de trabajos. Seleccione los entornos de computación en el orden

en que desee que la cola intente colocarlos en la cola de trabajos. El programador de trabajos utiliza el orden en el que selecciona los entornos de computación para determinar qué entorno de computación inicia un trabajo determinado. Antes de asociarlos a una cola de trabajos, los entornos de computación deben estar en estado `VALID`. Puede asociar hasta tres entornos de computación con una cola de trabajos.

 Note

Todos los entornos de computación asociados a una cola de trabajos deben compartir el mismo modelo de aprovisionamiento. AWS Batch no admite la mezcla de modelos de aprovisionamiento en una sola cola de trabajos.

 Note

Todos los entornos de computación asociados a una cola de trabajos deben compartir la misma arquitectura. AWS Batch no admite la mezcla de tipos de arquitecturas de entorno de computación en una sola cola de trabajos.

10. En Orden de entorno de computación, seleccione las flechas arriba y abajo para configurar el orden que desee.
11. Seleccione Crear cola de trabajos para finalizar y crear su cola de trabajos.

Creación de una cola de trabajos de entrenamiento de SageMaker en AWS Batch

Las colas de trabajo de entrenamiento de SageMaker se integran directamente con el servicio SageMaker AI para ofrecer una programación de trabajos sin servidor sin necesidad de administrar la infraestructura de computación subyacente.

Requisitos previos

Antes de crear una cola de trabajos de entrenamiento de SageMaker, asegúrese de tener:

- Entorno de servicio: entorno de servicio que define los límites de capacidad. Para obtener más información, consulte [Cree un entorno de servicios en AWS Batch](#).

- Permisos de IAM: permisos para crear y administrar colas de trabajos y entornos de servicio de AWS Batch. Para obtener más información, consulte [AWS Batch Políticas, funciones y permisos de IAM](#).

Create a SageMaker Training job queue (AWS Batch console)

1. Abra la consola AWS Batch en <https://console.aws.amazon.com/batch/>.
2. En el panel de navegación, seleccione Colas de trabajo y Crear.
3. En Tipo de orquestación, elija Entrenamiento de SageMaker.
4. En Configuración de la cola de trabajos:
 - a. En Nombre, ingrese el nombre de la cola de trabajos.
 - b. En Prioridad, ingrese un valor entre 0 y 1000. Se da preferencia a una cola de trabajos con mayor prioridad para los entornos de servicio.
 - c. (Opcional) En Política de programación Nombre de recurso de Amazon (ARN), elija una política de programación existente.
 - d. En Entornos de servicios conectados, seleccione un entorno de servicio de la lista para asociarlo a la cola de trabajos.
5. (Opcional) En Límites del estado de trabajo:
 - a. En Configuración incorrecta, elija SERVICE_ENVIRONMENT_MAX_RESOURCE e ingrese el Tiempo máximo de ejecución (segundos).
 - b. En Capacidad, elija INSUFFICIENT_INSTANCE_CAPACITY e ingrese el Tiempo máximo de ejecución (segundos).
6. Elija Crear cola de trabajos.

Create a SageMaker Training job queue (AWS CLI)

Utilice el comando `create-job-queue` para crear una cola de trabajo de entrenamiento de SageMaker.

El siguiente ejemplo crea una cola de trabajo de entrenamiento básica de SageMaker que utiliza un entorno de servicios:

```
aws batch create-job-queue \
  --job-queue-name my-sm-training-fifo-jq \
  --job-queue-type SAGEMAKER_TRAINING \
```

```
--priority 1 \  
--service-environment-order order=1,serviceEnvironment=ExampleServiceEnvironment
```

Sustituya *ExampleServiceEnvironment* por el nombre de su entorno de servicio.

El comando devuelve un resultado similar al siguiente:

```
{  
  "jobQueueName": "my-sm-training-fifo-jq",  
  "jobQueueArn": "arn:aws:batch:region:account:job-queue/my-sm-training-fifo-jq"  
}
```

Tras crear la cola de trabajos, compruebe que se ha creado correctamente y que su estado es válido.

Utilice el comando `describe-job-queues` para ver los detalles de la cola de trabajos:

```
aws batch describe-job-queues --job-queues my-sm-training-fifo-jq
```

El comando devuelve un resultado similar al siguiente:

```
{  
  "jobQueues": [  
    {  
      "jobQueueName": "my-sm-training-fifo-jq",  
      "jobQueueArn": "arn:aws:batch:region:account:job-queue/my-sm-training-fifo-jq",  
      "state": "ENABLED",  
      "status": "VALID",  
      "statusReason": "JobQueue Healthy",  
      "priority": 1,  
      "computeEnvironmentOrder": [],  
      "serviceEnvironmentOrder": [  
        {  
          "order": 1,  
          "serviceEnvironment": "arn:aws:batch:region:account:service-environment/ExampleServiceEnvironment"  
        }  
      ],  
      "jobQueueType": "SAGEMAKER_TRAINING",  
      "tags": {},  
      "jobStateTimeLimitActions": []  
    }  
  ]  
}
```

```
]
}
```

Asegúrese de que:

- El valor de `state` es `ENABLED`
- El valor de `status` es `VALID`
- El valor de `statusReason` es `JobQueue Healthy`
- El valor de `jobQueueType` es `SAGEMAKER_TRAINING`
- `serviceEnvironmentOrder` hace referencia al entorno de servicio

Plantilla de cola de trabajos

A continuación se muestra una plantilla de cola de trabajos vacía. Puede utilizar esta plantilla para crear su cola de trabajos. A continuación, puede guardar esta cola de trabajos en un archivo y utilizarla con la opción `--cli-input-json` de AWS CLI. Para obtener más información sobre estos parámetros, consulte [CreateJobQueue](#) en la Referencia de la API AWS Batch.

Note

Puede generar una plantilla de cola de trabajos con el siguiente comando de la AWS CLI.

```
$ aws batch create-job-queue --generate-cli-skeleton
```

```
{
  "computeEnvironmentOrder": [
    {
      "computeEnvironment": "",
      "order": 0
    }
  ],
  "jobQueueName": "",
  "jobStateTimeLimitActions": [
    {
      "state": "RUNNABLE",
      "action": "CANCEL",
      "maxTimeSeconds": 0,

```

```
        "reason": ""
    },
    ],
    "priority": 0,
    "schedulingPolicyArn": "",
    "state": "ENABLED",
    "tags": {
        "KeyName": ""
    }
}
```

Ver una lista de trabajos en espera en AWS Batch

Tras crear una cola de trabajos y enviar los trabajos, es importante supervisar su progreso. Puede utilizar la página Detalles del trabajo para revisar, gestionar y supervisar su cola de trabajos.

Visualización de información de la cola de trabajos

En la AWS Batch consola, seleccione Colas de trabajos en el panel de navegación y elija la cola de trabajos que desee para ver sus detalles. En esta página puede revisar y administrar la cola de trabajos o ver información adicional sobre las operaciones de la cola de trabajos, como la instantánea de la cola de trabajos, los límites de estado de los trabajos, el orden del entorno, las etiquetas y el código JSON de la cola de trabajos.

Detalles de la cola de trabajos

En esta sección se muestran información general y opciones de mantenimiento para la cola de trabajos. Tenga en cuenta que puede buscar el nombre de recurso de Amazon (ARN) en esta sección.

Para buscar esta información mediante AWS Command Line Interface, utilice la [DescribeJobQueues](#) operación junto con el nombre de la cola de trabajos o el ARN correspondiente.

Recursos compartidos activos

En el caso de las colas de trabajos que utilizan la programación de reparto equitativo, AWS Batch proporciona visibilidad de la forma en que los diferentes identificadores de acciones consumen capacidad. Esta información le ayuda a entender la distribución de los recursos y a identificar las participaciones que pueden necesitar ajustes.

Note

La pestaña Acciones activas solo está presente cuando el algoritmo de programación de la cola de trabajos es Fair-share.

La sección de las 20 acciones más activas muestra los identificadores de acciones cuyos trabajos están programados, iniciales y en ejecución. Esta vista incluye:

- Nombre del identificador del recurso compartido: el identificador único del recurso compartido.

Los identificadores de acciones son etiquetas que agrupan los trabajos para una programación equitativa. Cuando envía trabajos con el mismo identificador de recursos, los AWS Batch considera parte de la misma carga de trabajo a efectos de asignación de recursos. Los identificadores de recursos compartidos ayudan a garantizar una distribución equitativa de la capacidad informática entre los diferentes equipos, proyectos o tipos de carga de trabajo. Para obtener más información, consulte [Uso de identificadores de reparto para identificar las cargas de trabajo](#).

- Utilización de la capacidad: la cantidad de recursos para los que está configurado el trabajo. Esto se mide en `instancescpu`, o en vCPU función del entorno.
- Ver acción de empleo: un enlace para ver todos los trabajos de ese recurso compartido.

Puede ver esta información en formato de lista y gráfico:

- Vista de lista: pantalla tabular con números de capacidad exactos
- Vista de gráfico: gráfico de barras visual que muestra la utilización relativa

Instantánea de la cola de trabajos

Esta sección proporciona una lista estática de los primeros 100 RUNNABLE trabajos que están en cola. Puede utilizar el campo de búsqueda para buscar información en cualquier columna de la sección de resultados y reducir la lista. Los trabajos del área de resultados de instantánea se ordenan según la estrategia de ejecución de la cola de trabajos. En el first-in-first-out caso de las colas de trabajos (FIFO), el orden de los trabajos se basa en el tiempo de envío. En el caso de las colas de trabajos de la [programación de reparto justo](#), el orden de los trabajos se basa en su prioridad y en el uso de recursos compartidos. El campo Capacidad requerida muestra la cantidad de recursos que el trabajo está configurado para usar.

Como los resultados son una instantánea de la cola de trabajos, la lista de resultados no se actualiza automáticamente. Para actualizar la lista, seleccione Actualizar en la parte superior de la sección. Elija el hipervínculo del nombre del trabajo para ir a Detalles del trabajo y ver el estado u otra información relacionada.

Para buscar esta información mediante AWS CLI, utilice la [GetJobQueueSnapshot](#) operación junto con el nombre de la cola de trabajos o el ARN correspondiente.

```
aws batch get-job-queue-snapshot --job-queue my-sm-training-fifo-jq
```

Límites del estado de trabajo

Utilice esta pestaña para revisar la información de configuración sobre el tiempo que un trabajo puede tener el estado RUNNABLE antes de que se cancele.

Para buscar esta información mediante AWS CLI, utilice la [DescribeJobQueues](#) operación junto con el nombre de la cola de trabajos o el ARN correspondiente.

Orden de entornos

Si su cola de trabajos se ejecuta en varios entornos, esta pestaña muestra el orden y una descripción general.

Para buscar esta información mediante AWS CLI, utilice la [DescribeJobQueues](#) operación junto con el nombre de la cola de trabajos o el ARN correspondiente.

Tags

Utilice esta pestaña para revisar y administrar las etiquetas asociadas a esta cola de trabajos.

JSON

Utilice esta pestaña para copiar el código JSON asociado a esta cola de trabajos. A continuación, puede reutilizar el JSON para AWS CloudFormation plantillas y AWS CLI scripts.

Eliminación de una cola de trabajos en AWS Batch

Puede deshabilitar y eliminar la cola de trabajos cuando ya no la necesite.

Delete a job queue (AWS Batch console)

1. Abra la consola AWS Batch en <https://console.aws.amazon.com/batch/>.

2. En el panel de navegación, seleccione Colas de trabajos y luego elija una cola de trabajos.
3. Elija Acciones y luego Deshabilitar.
4. Cuando el estado de la cola de trabajos sea Deshabilitado, seleccione Acciones y, a continuación, Eliminar.
5. En la ventana modal, seleccione Eliminar cola de trabajos.

Delete a job queue (AWS CLI)

1. Deshabilite la cola de trabajos para evitar que se envíen nuevos trabajos:

```
aws batch update-job-queue \  
  --job-queue my-sm-training-fifo-jq \  
  --state DISABLED
```

2. Espere a que se completen los trabajos en ejecución y, a continuación, elimine la cola de trabajos:

```
aws batch delete-job-queue \  
  --job-queue my-sm-training-fifo-jq
```

Políticas de programación de reparto justo

El AWS Batch programador evalúa cuándo, dónde y cómo ejecutar los trabajos que se envían a una cola de trabajos. Si no especifica una política de programación al crear una cola de trabajos, el programador de trabajos utilizará de forma predeterminada la AWS Batch estrategia de «primero en entrar, primero en salir» (FIFO). Una estrategia FIFO puede provocar que los trabajos importantes se queden “atascados” con respecto a los trabajos que se enviaron anteriormente. Al especificar una política de programación diferente, puede asignar los recursos de computación de acuerdo con sus necesidades específicas.

Note

Si desea programar el orden específico en el que se ejecutan los trabajos, utilice el [dependsOn](#) parámetro in [SubmitJob](#) para especificar las dependencias de cada trabajo.

Si crea una política de programación y la adjunta a una cola de trabajos, se activará la programación de reparto justo. Si la cola de trabajos tiene una política de programación, la política de programación determina el orden en que se ejecutan los trabajos. Para obtener más información, consulte [Uso de políticas de programación de reparto justo para asignar identificadores de participación equitativa](#).

Temas

- [Uso de identificadores de reparto para identificar las cargas de trabajo](#)
- [Uso de políticas de programación de reparto justo para asignar identificadores de participación equitativa](#)
- [Uso de una programación de reparto justo para ayudar a programar los trabajos](#)
- [Tutorial: Cree una política de programación de reparto justo](#)
- [Referencia: plantilla de política de programación de reparto equitativo](#)

Uso de identificadores de reparto para identificar las cargas de trabajo

Puede usar los identificadores de recursos compartidos para etiquetar los trabajos y diferenciar entre usuarios y cargas de trabajo. El AWS Batch planificador realiza un seguimiento del uso de cada identificador de recurso compartido mediante la $(T * weightFactor)$ fórmula, donde T es el uso de la vCPU a lo largo del tiempo. El programador selecciona los trabajos con el menor uso a partir del identificador de recursos compartidos. Puede usar un identificador de reparto sin anularlo.

Note

Los identificadores de acciones son únicos dentro de una cola de trabajos y no se agregan entre las colas de trabajos.

Puede establecer la prioridad de programación de reparto justo para configurar el orden en que se ejecutan los trabajos en un identificador de recursos compartidos. Los trabajos con una prioridad de programación más alta se programan primero. Si no especifica una política de programación de reparto justo, todos los trabajos que se envían a la cola de trabajos se programan en orden FIFO. Al enviar trabajos, no puede especificar identificadores de cuota ni prioridades de programación de reparto justo.

 Note

Los recursos de computación adjuntos se asignan por igual entre todos los identificadores de recursos compartidos, a menos que se anulen de forma explícita.

Uso de políticas de programación de reparto justo para asignar identificadores de participación equitativa

Puede utilizar políticas de programación equitativas para configurar cómo se asignan los recursos informáticos de una cola de trabajos entre los usuarios o las cargas de trabajo. Al utilizar políticas de programación equitativas, puede asignar distintos identificadores de reparto a las cargas de trabajo o a los usuarios. AWS Batch asigna a cada identificador de recurso compartido un porcentaje del total de recursos que están disponibles durante un período de tiempo.

El porcentaje de reparto justo se calcula utilizando los valores `shareDecaySeconds` y `shareDistribution`. Puede añadir tiempo al análisis del reparto justo asignando un tiempo de caída de participación de las acciones a la política. Agregar tiempo da más peso al tiempo y menos al peso definido. Si especifica una reserva informática, puede reservar los recursos de computación para los identificadores de reparto que no estén activos. Para obtener más información, consulte [SchedulingPolicyDetail](#).

Uso de una programación de reparto justo para ayudar a programar los trabajos

La programación de reparto justo proporciona un conjunto de controles para ayudar a programar los trabajos. Para obtener más información acerca de los parámetros de política de programación, consulte [SchedulingPolicyDetail](#).

- Segundos de disminución de las acciones: período de tiempo (en segundos) que el AWS Batch programador utiliza para calcular un porcentaje de participación equitativa para cada identificador de acciones. Un valor de cero indica que solo se mide el uso actual. Un tiempo de decaimiento más largo da más peso al tiempo.

 Note

El período de tiempo de descomposición se calcula de la siguiente manera:
 $shareDecaySeconds + OrderMinutes$ donde *OrderMinutes* es el tiempo del pedido en minutos.

- Reserva de computación: evita que los trabajos de un único identificador de recurso compartido consuman todos los recursos adjuntos a la cola de trabajos. La proporción reservada es $(computeReservation/100)^{ActiveFairShares}$, donde *ActiveFairShares* es el número de identificadores de reparto activos.

 Note

Si un identificador de recursos compartidos tiene trabajos en un estado SUBMITTED, PENDING, RUNNABLE, STARTING, o RUNNING se considera un identificador de recurso compartido activo. Una vez transcurrido el período de inactividad, el identificador de acciones se considera inactivo.

- Factor de ponderación: el factor de ponderación para el identificador de cuota justa. El valor predeterminado es 1. Un valor inferior permite ejecutar los trabajos del identificador de recurso compartido o proporciona tiempo de ejecución adicional al identificador de recurso compartido. Por ejemplo, los trabajos que utilizan un identificador de recurso compartido con un factor de peso de 0,125 (1/8) obtienen ocho veces los recursos de computación de los trabajos que utilizan un identificador de recurso compartido con un factor de ponderación de 1.

 Note

Solo necesita definir este atributo cuando necesite actualizar el factor de ponderación predeterminado de 1.

Cuando la cola de trabajos esté activa y procese trabajos, podrá revisar una lista de los primeros 100 trabajos de RUNNABLE mediante la instantánea de la cola de trabajos. Para obtener más información, consulte [Ver una lista de trabajos en espera en AWS Batch](#).

Tutorial: Cree una política de programación de reparto justo

Antes de crear una cola de trabajos con una política de programación, debe crear una política de programación. Al crear una política de programación, se asocian uno o más identificadores de reparto o prefijos de identificador de reparto a las ponderaciones de la cola y, si lo prefiere, se le asigna un período de reducción y se calcula la reserva a la política.

Para crear una política de programación equitativa

1. Abre la AWS Batch consola en. <https://console.aws.amazon.com/batch/>
2. En la barra de navegación, seleccione la región a utilizar.
3. En el panel de navegación, elija Políticas de programación, Creación de.
4. Para Nombre, escriba un nombre único para la política de programación. Se admiten hasta 128 letras (mayúsculas y minúsculas), números, guiones y caracteres de subrayado.
5. (Opcional) En Segundos de decaimiento de cuota, introduzca un valor entero para el tiempo de decaimiento de cuota de la política de programación de reparto justo. Un tiempo de decaimiento de la cuota más largo utilizará muchos más recursos de computación durante un tiempo más largo al programar los trabajos. Esto puede permitir que los trabajos que utilizan un identificador de reparto utilicen temporalmente más recursos de computación de los que permitiría el peso de ese identificador si ese identificador de reparto no hubiera utilizado recursos de computación recientemente.
6. (Opcional) En Reserva de computación, introduzca un valor entero para la reserva de cálculo de la política de programación de reparto justo. La reserva de cómputo guardará una v CPUs en reserva para utilizarla como identificadores de acciones que no estén activos actualmente.

La proporción reservada es $(computeReservation/100)^{ActiveFairShares}$, donde *ActiveFairShares* es el número de identificadores de reparto activos.

Por ejemplo, un *computeReservation* valor de 50 indica que se AWS Batch debe reservar el 50% de la CPU virtual máxima disponible si solo hay un identificador de recurso, el 25% si hay dos identificadores de recurso y el 12,5% si hay tres identificadores de recurso. Un valor *computeReservation* de 25 indica que AWS Batch debe reservar el 25 % de la VCPU máxima disponible si solo hay un identificador de reparto, el 6,25 % si hay dos identificadores de reparto y el 1,56 % si hay tres identificadores de reparto.

7. En la sección Atributos de cuota, puede especificar el identificador de reparto y el peso de cada identificador de reparto para asociarlos a la política de programación de reparto justo.

- a. Seleccione Añadir identificador de reparto.
 - b. En Identificador de reparto, especifique el identificador de reparto. Si la cadena termina con “*”, se convierte en un prefijo identificador de reparto que se utiliza para hacer coincidir los identificadores de reparto de los trabajos. Todos los identificadores de reparto y los prefijos de los identificadores de reparto de una política de programación deben ser únicos y no pueden superponerse. Por ejemplo, los identificadores de reparto no pueden tener el prefijo “UserA*” y el identificador de reparto “userA1” en la misma política de programación de reparto justo.
 - c. En Factor de ponderación, especifique la ponderación relativa para el identificador de reparto. El valor predeterminado es 1.0. Un valor más bajo tiene una mayor prioridad para los recursos de computación. Si se utiliza un prefijo identificador de reparto, los trabajos con identificadores de reparto que comiencen con ese prefijo compartirán el factor de ponderación. De hecho, esto aumenta el factor de ponderación de esos puestos de trabajo, lo que reduce su prioridad individual, a la vez que mantiene el mismo factor de ponderación para el prefijo identificador de reparto.
8. (Opcional) En la sección Etiquetas, puede especificar la clave y el valor de cada etiqueta que desea asociar a la política de programación. Para obtener más información, consulte [Etiquetar los recursos de AWS Batch](#).
 9. Seleccione Enviar para terminar y crear su política de programación.

Referencia: plantilla de política de programación de reparto equitativo

A continuación se muestra una plantilla vacía de política de programación de reparto justo. Puede usar esta plantilla para crear su política de programación, que luego se puede guardar en un archivo y usar con la AWS CLI `--cli-input-json` opción. Para obtener más información sobre estos parámetros, consulte [CreateSchedulingPolicy](#) la referencia de la AWS Batch API.

Note

Puede generar una plantilla de cola de trabajos con el siguiente AWS CLI comando.

```
$ aws batch create-scheduling-policy --generate-cli-skeleton
```

```
{
```



```
{
  "name": "",
  "fairsharePolicy": {
    "shareDecaySeconds": 0,
    "computeReservation": 0,
    "shareDistribution": [
      {
        "shareIdentifier": "",
        "weightFactor": 0.0
      }
    ]
  },
  "tags": {
    "KeyName": ""
  }
}
```

Programación basada en los recursos

AWS Batch programa los trabajos en función de la vCPU, la GPU y la disponibilidad de memoria en el entorno informático (CE) asociado a la cola de trabajos (JQ). Sin embargo, a veces, la mera disponibilidad de estos recursos del CE no garantiza que el trabajo se ejecute correctamente, ya que puede depender de otros recursos. Por lo tanto, estos trabajos se deben cancelar o finalizar. Esto da lugar a un uso ineficiente de los recursos de computación. Para resolver este problema, la programación basada en los recursos puede comprobar la disponibilidad de los recursos dependientes y ajenos al CE antes de programar el trabajo para que se ejecute en un CE.

AWS Batch La programación basada en los recursos te permite programar los trabajos en función de los recursos consumibles que se necesitan para ejecutarlos: identificadores de licencia de terceros, ancho de banda de acceso a la base de datos, necesidad de limitar las llamadas a una API de terceros, etc. Estos recursos consumibles se deben especificar para ejecutar un trabajo, por lo tanto, Batch contempla dichas dependencias de recursos cuando realiza la programación de uno. Puede evitar realizar intervenciones manuales para eliminar fallas en el trabajo y las largas esperas derivadas de la escasez de recursos consumibles. La reducción de la infrautilización de los recursos de computación se lleva a cabo con la asignación de los trabajos que contienen todos los recursos necesarios.

La programación basada en los recursos está disponible para las políticas de programación FIFO y de reparto justo, y se puede utilizar con todas las plataformas de computación compatibles con Batch, como EKS, ECS y Fargate. Así como con trabajos de matriz, trabajos paralelos de varios nodos (MNP) y trabajos habituales de Batch.

Para configurar la programación basada en los recursos, primero debe especificar todos los recursos consumibles necesarios para ejecutar los trabajos, junto con el recuento total disponible de cada recurso. A continuación, para cada trabajo que requiera un recurso consumible, especifique el nombre y la cantidad necesaria para cada recurso. Batch realiza un seguimiento de la cantidad de recursos consumibles disponibles para los trabajos de sus colas de trabajos y se asegura de que un trabajo esté programado para ejecutarse solo cuando todos los recursos consumibles necesarios estén disponibles para que el trabajo se ejecute correctamente.

Temas

- [Creación de recursos consumibles](#)
- [Cómo especificar los recursos necesarios para ejecutar un trabajo](#)
- [Cómo comprobar la cantidad de recursos disponibles y en uso](#)
- [Cómo actualizar la cantidad de un recurso mientras los trabajos lo utilizan](#)
- [Cómo buscar trabajos que requieren un recurso consumible específico](#)
- [Cómo eliminar un recurso consumible](#)

Creación de recursos consumibles

Primero, debe crear recursos consumibles que representen los recursos no relacionados con el CE que se consumen al ejecutar un trabajo y que solo están disponibles en cantidades limitadas. Cada recurso consumible tiene:

- un nombre del recurso (`consumableResourceName`) que debe ser único a nivel de cuenta,
- (opcional) un tipo de recurso (`resourceType`) que indica si el recurso está disponible para su uso una vez finalizado un trabajo. Puede ser cualquiera de los dos:
 - `REPLENISHABLE` (predeterminado)
 - `NON_REPLENISHABLE`
- una cantidad total (`totalQuantity`) que especifica el monto total del recurso consumible disponible.

La cantidad máxima de recursos consumibles por cuenta es de 50 000.

Consola:

1. En el panel de navegación izquierdo de la [consola de AWS Batch](#), seleccione Recursos consumibles.
2. Seleccione Crear recurso consumible.
3. Ingrese un Nombre de recurso único, la Cantidad total de recursos y seleccione si el Tipo de recurso es de Reposición o de No reposición.
4. Seleccione Crear recurso consumible.

API:

Utilice la [API de CreateConsumableResource](#) para definir los recursos deseados.

Cómo especificar los recursos necesarios para ejecutar un trabajo

En el registro de un trabajo se puede especificar el nombre de uno o más recursos creados (consumableResource) y la cantidad de ese recurso que requiere cada instancia del trabajo (quantity).

Batch realiza un seguimiento de las unidades disponibles de cada recurso en todo momento. Para cada trabajo de la cola de trabajos, el programador de Batch garantiza que el trabajo se ejecute solo cuando las dependencias de recursos especificadas estén disponibles.

Si un recurso consumible para el trabajo no está disponible cuando el trabajo llegue al principio de la cola, el trabajo esperará en estado RUNNABLE hasta que estén disponibles todos los recursos necesarios o se alcance el límite de tiempo del estado del trabajo (consulte [Ver una lista de trabajos en espera en AWS Batch](#)) Una vez que Batch haya validado que todos los recursos estén disponibles, el trabajo pasa al estado STARTING y después a RUNNING. Los recursos se bloquean una vez que el trabajo se traslada a STARTING y luego, se desbloquean cuando el trabajo se traslada a SUCCEEDED o FAILED.

También puede actualizar la cantidad de un recurso necesaria para un trabajo específico al enviar el trabajo.

Consola:

Cómo especificar los recursos y las cantidades necesarios para definir un trabajo:

1. Defina un trabajo mediante el asistente de definición de trabajos de la [consola de AWS Batch](#) (Definiciones de trabajos -> Crear).

2. En el paso 4 del asistente: Configurar los contenedores, en Recurso consumible, seleccione el Nombre del recurso necesario de la lista. En el campo Valor solicitado, ingrese la cantidad de este recurso que una instancia de este trabajo necesita y, a continuación, seleccione Agregar recurso consumible.
3. Repita el paso anterior para todos los recursos consumibles que requiere el trabajo. Puede especificar hasta 5 recursos para cada trabajo que defina.
4. Verá una lista de los recursos consumibles que ha creado después de completar el asistente de definición de trabajos y antes de elegir Crear definición de trabajos.

Cómo actualizar las cantidades de recursos necesarias al enviar un trabajo:

1. En el panel de navegación izquierdo de la [consola de AWS Batch](#), elija Trabajos y luego Enviar trabajo nuevo.
2. En el paso 2 del asistente: Configurar las anulaciones, en Anulaciones de recursos consumibles, ingrese un nuevo Valor solicitado para cualquier recurso consumible cuya cantidad necesaria desee anular para el trabajo.
3. Cuando haya completado todas las anulaciones que desee realizar para este trabajo, seleccione Siguiente para continuar en Revisar y enviar.

API:

Cuando registre un trabajo en la [API RegisterJobDefinition](#), utilice `consumableResourceList` en la parte `consumableResourceProperties` de la solicitud para especificar los recursos consumibles necesarios para ejecutar una instancia del trabajo y la cantidad de cada uno de ellos.

Al enviar un trabajo con la [API SubmitJob](#), puede anular la lista de recursos consumibles y la cantidad de cada uno utilizando la parte `consumableResourcePropertiesOverride` de la solicitud. Ten presente que esto solo anula la cantidad de recursos que necesita cada instancia del trabajo, no la cantidad total disponible.

Cómo comprobar la cantidad de recursos disponibles y en uso

Batch permite consultar la cantidad de recursos disponibles (`availableQuantity`), la cantidad de recursos en uso (`inUseQuantity`) y el total de recursos (`totalQuantity`) en cada momento.

Una vez que un trabajo pasa al estado `STARTING`, los recursos consumidos se restarán de la cantidad disponible de ese recurso. Si el recurso es `REPLENISHABLE`, la cantidad de recursos consumidos se volverá a sumar a la cantidad disponible en cuanto la tarea pase al estado `SUCCEED` (correcto) o `FAILED` (fallido), y la cantidad total seguirá siendo la misma. Si el recurso es `NON_REPLENISHABLE`, la cantidad de recursos consumidos se resta de las cantidades totales y disponibles, y no se volverá a sumar si la tarea pasa al estado `SUCCEED` o `FAILED`.

 Note

Esta información puede demorarse hasta 30 segundos.

Consola:

1. En el panel de navegación izquierdo de la [consola de AWS Batch](#), seleccione Recursos consumibles.
2. Seleccione la pestaña Reposición o No reposición para ver los recursos de ese tipo que ha creado.
3. Para cada recurso de Reposición, la consola muestra el Nombre, la cantidad Total del recurso, el número En uso actual y el número que aún está Disponible, junto con un cálculo de la Utilización (el número de recursos en uso dividido por la cantidad total de ese recurso).

Para cada recurso de No reposición, la consola muestra el Nombre, el número En uso actual y cuántos están todavía Disponibles.

También puede ver la información actual sobre los recursos consumibles en la página de detalles del trabajo en la consola.

1. En el panel de navegación izquierdo de la [consola de AWS Batch](#), elija Trabajos y, a continuación, seleccione el nombre de un trabajo para abrir la página de detalles de ese trabajo.
2. La información sobre los Recursos de reposición y los Recursos que no se pueden reponer está disponible en caso de que el trabajo los requiera. Para ambos tipos, la consola muestra el Nombre del recurso, la cantidad Solicitada para el trabajo, cuántos están todavía Disponibles, el número En uso actual y la cantidad Total del recurso, junto con un cálculo de la Utilización actual (el número de recursos que utiliza el trabajo dividido por la cantidad total de ese recurso).

API:

Utilice la [API DescribeConsumableResource](#) que devuelve la siguiente información:

```
{
  "availableQuantity": number,
  "consumableResourceArn": "string",
  "consumableResourceName": "string",
  "createdAt": number,
  "inUseQuantity": number,
  "resourceType": "string",
  "tags": {
    "string" : "string"
  },
  "totalQuantity": number
}
```

Además, la [API de ListConsumableResources](#) informa de la cantidad de recursos en uso (`inUseQuantity`) y la cantidad total de recursos disponibles actualmente (`totalQuantity`) como parte de su lista de todos los recursos consumibles que ha creado en su cuenta. Esta API también permite filtrar la consulta de la lista de recursos consumibles según el nombre del recurso consumible.

Cómo actualizar la cantidad de un recurso mientras los trabajos lo utilizan

La cantidad total de un recurso se puede sumar, restar o establecer a un nuevo valor.

Si la nueva cantidad total especificada es mayor a la anterior, Batch programa más trabajos en consecuencia. Si la nueva cantidad total es menor a la anterior y no hay unidades de este recurso en uso, Batch simplemente reduce la cantidad total (o disponible). Si hay unidades en uso, Batch reduce la cantidad disponible de inmediato y, cuando finalizan los trabajos, Batch reduce la cantidad total (disponible) para llegar finalmente al nuevo número.

Consola:

1. En el panel de navegación izquierdo de la [consola de AWS Batch](#), seleccione Recursos consumibles.
2. Seleccione la pestaña Reposición o No reposición para ver los recursos de ese tipo que ha creado.
3. En los recursos de Reposición:

1. Elija el recurso que desee actualizar y, a continuación, seleccione Acciones y elija Establecer recursos, Agregar recursos o Eliminar recursos.
2. Aparece una ventana emergente en la que puede Establecer el valor total, Agregar recursos o Eliminar recursos según la acción que haya elegido en el paso anterior. Ingrese la cantidad que desee establecer como el nuevo valor total, la que desee agregar a la cantidad total o la que desee restar de la cantidad total y, a continuación, seleccione Aceptar.

En los recursos de No reposición:

1. Elija el recurso que desee actualizar y, a continuación, seleccione Acciones y elija Establecer recursos, Agregar recursos o Eliminar recursos.
2. Aparece una ventana emergente en la que puede Establecer el valor disponible, Agregar recursos o Eliminar recursos según la acción que haya elegido en el paso anterior. Ingrese la cantidad que desee establecer como nuevo valor disponible, que desee agregar a la cantidad disponible o que desee restar de la cantidad disponible y, a continuación, seleccione Aceptar.

API:

Utilice la [API UpdateConsumableResource](#) para establecer una nueva cantidad total para el recurso o para aumentarla o reducirla.

Cómo buscar trabajos que requieren un recurso consumible específico

Batch permite recuperar una lista de trabajos que requieren un recurso consumible específico.

Consola:

1. En el panel de navegación izquierdo de la [consola de AWS Batch](#), seleccione Recursos consumibles.
2. En la lista, seleccione el nombre del recurso consumible. Se abrirá la página de detalles para ese recurso.
3. En Buscar trabajos, ingrese los filtros que desee aplicar a la lista de trabajos. Puede filtrar por el Nombre del trabajo (“es igual a”, “empieza por”), el Intervalo de fechas (cuando se creó el trabajo) y por Criterios adicionales (“cola de trabajos”, “definición del trabajo”, “identificador de trabajo compartido”). Para cada tipo de filtro que desee aplicar, seleccione una de las opciones disponibles en la lista desplegable e ingrese la información adicional solicitada.

Elija Buscar.

4. Se muestra una lista (filtrada) de los trabajos que requieren el recurso consumible, que incluye el nombre del trabajo, el estado, el número de unidades solicitadas del recurso consumible, otros recursos consumibles necesarios, etc. Con esta lista, puede seleccionar uno o más trabajos para Cancelarlos o Finalizarlos. También puede seleccionar el nombre de un trabajo para abrir la página de detalles de ese trabajo.
5. En Buscar trabajos, ahora puede Actualizar los resultados o Borrar la búsqueda y empezar de nuevo.

API:

Con la [API ListJobsByConsumableResource](#), puede obtener una lista de trabajos que utilizan un recurso consumible específico. Esta API también permite filtrar la consulta de la lista de trabajos según el estado o nombre del trabajo.

Cómo eliminar un recurso consumible

Un recurso consumible se puede eliminar en cualquier momento, incluso cuando los trabajos que requieren el recurso estén en ejecución. Tras la eliminación de un recurso consumible, puede haber un intervalo entre el momento en que se recibe el comando de eliminación y el momento en que el programador de tareas acepta la eliminación, por lo tanto, es posible que los trabajos que consumen el recurso se programen inmediatamente después de la llamada de eliminación. Si el recurso consumible eliminado tiene el tipo de recurso (`resourceType`) `REPLENISHABLE`, se ignorará cuando se completen los trabajos. Si elimina un recurso consumible y lo vuelve a crear con el mismo nombre, se considerará que es el mismo recurso y los trabajos `RUNNABLE` lo pueden utilizar.

Consola:

1. En el panel de navegación izquierdo de la [consola de AWS Batch](#), seleccione Recursos consumibles.
2. Seleccione la pestaña Reposición o No reposición para ver los recursos de ese tipo que ha creado.
3. Seleccione cada recurso que desea eliminar y, a continuación, elija Eliminar. Aparecerá la ventana emergente Eliminar recurso consumible. Para confirmar la eliminación, elija Eliminar.

También puede eliminar un recurso consumible de su página de detalles en la consola.

1. En el panel de navegación izquierdo de la [consola de AWS Batch](#), seleccione Recursos consumibles.
2. Seleccione la pestaña Reposición o No reposición para ver los recursos de ese tipo que ha creado.
3. Elija el nombre del recurso que desea eliminar. Aparecerá la página de detalles del recurso consumible. Elija Eliminar. Aparecerá la ventana emergente Eliminar recurso consumible. Para confirmar la eliminación, elija Eliminar.

API:

Utilice la [API DeleteConsumableResource](#) para eliminar un recurso consumible.

Gestión de cuotas

AWS Batch programa los trabajos en función de los recursos que requiere el trabajo y la capacidad disponible en el entorno de servicios (SE) conectado, lo que permite un alto uso de los recursos informáticos. Con la administración de cuotas, los administradores también pueden controlar cuántos recursos puede consumir un equipo o proyecto mediante asignaciones de recursos más precisas.

La administración de cuotas permite a los administradores asignar de manera eficiente los recursos informáticos compartidos entre equipos y proyectos mediante la definición de las cuotas asociadas, AWS Batch recursos que incluyen las cuotas informáticas (límites de capacidad) y las estrategias de uso compartido de recursos para la computación inactiva. Cada cuota compartida funciona como una cola virtual anidada dentro de una cola de trabajos asociada. Al programar trabajos para una cola de trabajos, AWS Batch recorrerá todas las cuotas compartidas adjuntas.

Los administradores pueden habilitar el intercambio de recursos entre cuotas con total confianza, ya que la preferencia permitirá que cualquier cuota recupere los recursos que haya prestado a otros cuando los necesite. Dentro de una cuota compartida, puede elegir si las tareas en ejecución deben sustituirse por las de mayor prioridad o dejar que finalicen las tareas en ejecución. Las prioridades de trabajo se pueden establecer en el momento de la presentación y actualizarse más adelante; las prioridades actualizadas se tienen en cuenta a la hora de tomar decisiones preferentes. La utilización de la capacidad se puede monitorear en la cola, la cuota de cuotas y la granularidad a nivel de trabajo.

La administración de cuotas solo se admite para las colas de trabajos conectadas a un entorno de servicios. SAGEMAKER_TRAINING

Temas

- [Cuotas compartidas](#)
- [Preferencia](#)
- [Crear recursos de administración de cuotas](#)
- [Creación de cuotas compartidas](#)
- [Someter los trabajos a una cuota de cuota](#)

Cuotas compartidas

Las cuotas compartidas son colas virtuales anidadas debajo de una cola de trabajos. Se pueden adjuntar un máximo de 20 a cualquier cola de trabajos. Las cuotas compartidas permiten asignar una cuota de cómputo a un equipo o proyecto mediante el uso de límites de capacidad; una cuota compartida requiere que se proporcione al menos un límite de capacidad y admite un máximo de 5 límites de capacidad. Cada límite de capacidad debe expresarse como límites de instancias para los tipos de instancias de trabajo de SageMaker formación compatibles.

Cuota: estrategia de reparto de recursos

Las cuotas compartidas también tienen una configuración explícita para compartir recursos.

- Si el cómputo inactivo de una cuota compartida debe reservarse únicamente para sus trabajos, seleccione `RESERVE`.
- Si el cómputo inactivo de una cuota se puede prestar a otras cuotas compartidas, seleccione `LEND`.
- Si el cómputo inactivo de una cuota se puede prestar a otras cuotas y los trabajos de esa cuota deberían poder utilizar el cómputo inactivo, seleccione una opción `LEND_AND_BORROW` con un límite de préstamo configurado.

Restauración de la capacidad mediante preferencia

AWS Batch realizará una acción preferente entre acciones para devolver la capacidad prestada a una cuota de cuota cuando lleguen los puestos de trabajo. Al elegir qué trabajos preferir, AWS Batch tiene en cuenta el número y el tipo de instancias que un trabajo está tomando prestado, las prioridades relativas de los trabajos y la duración del trabajo, y aplica una heurística personalizada.

Los administradores que reduzcan los límites de capacidad de una cuota compartida también pueden hacer que SCHEDULEDSTARTING, o los RUNNING trabajos que se encuentren dentro de esa cuota, puedan optar a la preferencia, incluso si la estrategia de uso compartido de recursos RESERVE se seleccionó LEND o se seleccionó como estrategia de intercambio de recursos, si la capacidad ya consumida por esa cuota de cuota supera los límites de capacidad reducidos.

Preferencia

La administración de cuotas es el único algoritmo de AWS Batch programación que incorpora la preferencia, es AWS Batch decir SCHEDULEDSTARTING, detener un RUNNING trabajo o crear capacidad para un RUNNABLE trabajo.

Intercambio preferente

La administración de cuotas utiliza la preferencia entre acciones para devolver la capacidad prestada a una cuota de cuota cuando llegan los puestos de trabajo.

Los administradores que reduzcan los límites de capacidad de una cuota de cuota también pueden hacer que SCHEDULEDSTARTING, o los RUNNING trabajos que se encuentren dentro de esa cuota, puedan optar a la exclusión si la capacidad consumida por esa cuota ahora supera los límites de capacidad configurados.

Prioridad en el uso compartido

Las cuotas compartidas se pueden configurar para habilitar la preferencia dentro de las acciones, lo que permite que los trabajos de mayor prioridad activen la preferencia de los RUNNABLE trabajos de menor prioridad dentro de la misma cuota de cuota que se ingresó, o. SCHEDULED STARTING
RUNNING

Algoritmo de selección preferente

Al elegir qué trabajos preferir, AWS Batch tiene en cuenta el número y el tipo de instancias que toma prestadas, las prioridades relativas de los trabajos y la duración de los trabajos, y aplica una heurística personalizada. Puede utilizar la llamada a la [UpdateServiceJob](#) API para actualizar un trabajo después schedulingPriority de enviarlo. Esto puede resultar útil para reducir la prioridad de un RUNNING trabajo (lo que aumenta la probabilidad de que se dé prioridad) o para aumentar la prioridad de un RUNNABLE trabajo dentro de una cuota con la preferencia dentro del recurso compartido habilitada, lo que permite que el trabajo prevalezca sobre un trabajo que ya esté en ejecución.

Reintentos preferentes

La opción predeterminada es que los trabajos prioritarios se vuelvan a poner en cola sin límite. `RUNNABLE` Para limitar el número de preferencias que puede tener un trabajo, defina el envío de trabajos. `preemptionRetriesBeforeTermination` Si `preemptionRetriesBeforeTermination` se establece en 0, los trabajos se asignan a `FAILED` su primera prioridad.

En el trabajo se guarda una ventana deslizante con los intentos de prevención recientes, que se puede ver a través de ella. [DescribeServiceJob](#)

Crear recursos de administración de cuotas

La administración de cuotas requiere ajustes específicos al crear una política de programación, un entorno de servicio y una cola de trabajos asociados.

Requisitos previos

Antes de crear los recursos de administración de cuotas, asegúrese de tener:

- Permisos de IAM: permisos para crear y gestionar colas de AWS Batch trabajos, políticas de programación y entornos de servicio. Para obtener más información, consulte [AWS Batch Políticas, funciones y permisos de IAM](#).

Configure quota management resources (AWS Batch console)

La AWS Batch consola proporciona un flujo de trabajo integrado para crear todos los recursos necesarios para la gestión de las cuotas. El flujo de trabajo de creación de colas de trabajos de administración de cuotas también crea políticas de programación y entornos de servicio compatibles con la administración de cuotas.

1. Abra la AWS Batch consola en. <https://console.aws.amazon.com/batch/>
2. En el panel de navegación, elija Colas de trabajos y, a continuación, Crear.
3. Para el tipo de orquestación, elija SageMaker Entrenamiento.
4. En Configuración de la cola de trabajos:
 - a. En Nombre, introduzca el nombre de la cola de trabajos.
 - b. En Prioridad, introduzca un valor entre 0 y 1000. En los entornos de servicio, se da preferencia a las colas de trabajos con mayor prioridad.

5. Para la programación:
 - a. Para el algoritmo de programación, elija Administración de cuotas.
 - b. Para el ARN de la política de programación:
 - Si ya existe una política de programación que especifique la administración de cuotas, selecciónela en el menú desplegable.
 - De lo contrario, elija Crear política de programación.
 - i. Se abre una barra lateral para configurar la política de programación de la administración de cuotas.
 - ii. Proporcione un nombre para la política de programación.
 - iii. Seleccione Crear. El campo ARN de la política de programación ahora está relleno.
6. Para la configuración del entorno de servicio, en Entorno de servicio conectado:

 Note

Los entornos de servicio habilitados para la administración de cuotas solo se pueden conectar a una única cola de trabajos habilitada para la administración de cuotas.

- a. Si ya se ha creado un entorno de servicios compatible con la administración de cuotas y aún no está conectado a una cola de trabajos habilitada para la administración de cuotas, selecciónelo en el menú desplegable.
 - b. De lo contrario, selecciona Crear un entorno de servicios. Se abre una barra lateral para configurar el entorno de servicio.
 - i. Proporcione un nombre para el entorno de servicio.
 - ii. Proporcione al menos un límite de capacidad (y como máximo 5). Para cada límite de capacidad, elija un tipo de instancia en el menú desplegable y un número máximo de instancias.
7. (Opcional) En Límites del estado de trabajo:
 - a. En Configuración incorrecta, elija una de estas opciones `SERVICE_ENVIRONMENT_MAX_RESOURCE` e introduzca el tiempo máximo de ejecución (segundos).

- b. En Capacidad, elija `INSUFFICIENT_INSTANCE_CAPACITY` e ingrese el Tiempo máximo de ejecución (segundos).
8. Seleccione Crear cola de trabajos.

Configure quota management resources (AWS CLI)

Para configurar la administración de cuotas mediante la AWS CLI, cree una política de programación, un entorno de servicio y una cola de trabajos. Tanto la política de programación como el entorno de servicios deben ser compatibles con la administración de cuotas y deben crearse antes de crear la cola de trabajos.

Cree una política de programación

Utilice el `create-scheduling-policy` comando para crear una política de programación compatible con la administración de cuotas. Proporcione una política de cuotas compartidas durante la creación:

```
aws batch create-scheduling-policy \  
  --name my-qm-sagemaker-scheduling-policy \  
  --quota-share-policy idleResourceAssignmentStrategy="FIFO"
```

Compruebe que la política de programación se haya creado correctamente:

```
aws batch describe-scheduling-policies \  
  --arns arn-for-my-qm-sagemaker-scheduling-policy
```

Cree un entorno de servicio

Utilice el `create-service-environment` comando para crear un entorno de servicio habilitado para la administración de cuotas. Asegúrese de que los límites de capacidad utilicen tipos de instancias que acepten los trabajos de SageMaker formación, como `m1.g6.xlarge` o `m1.p4d.24xlarge`.

```
aws batch create-service-environment \  
  --service-environment-name my-qm-sagemaker-service-env \  
  --service-environment-type SAGEMAKER_TRAINING \  
  --capacity-limits capacityUnit=instance_type,maxCapacity=instance_count
```

Compruebe que el entorno de servicio se haya creado correctamente:

```
aws batch describe-service-environments \  
  --service-environments my-qm-sagemaker-service-env
```

Cree una cola de trabajos

Utilice el `create-job-queue` comando para crear una cola de trabajos habilitada para la administración de cuotas. Deben cumplirse los siguientes criterios:

- Debe proporcionarse un entorno de SAGEMAKER_TRAINING servicio único que no esté conectado actualmente a otra cola de trabajos.
- El entorno de servicio debe expresar los límites de capacidad en términos de tipos de instancias, por ejemplo `m1.m6i.xlarge`, en lugar de `NUM_INSTANCES`.
- Debe estar conectada una política de programación que contenga un `quotaSharePolicy`.
- El `jobQueueType` debe ser `SAGEMAKER_TRAINING`.

```
aws batch create-job-queue \  
  --job-queue-name my-qm-sagemaker-jq \  
  --job-queue-type SAGEMAKER_TRAINING \  
  --priority 1 \  
  --service-environment-order order=1,serviceEnvironment=my-qm-sagemaker-service-env \  
  --scheduling-policy-arn arn-for-my-qm-sagemaker-scheduling-policy
```

Compruebe que la cola de trabajos se haya creado correctamente:

```
aws batch describe-job-queues \  
  --job-queues my-qm-sagemaker-jq
```

Asegúrese de que:

- El `state` es `ENABLED`
- El `status` es `VALID`
- El `statusReason` es `JobQueue Healthy`

Creación de cuotas compartidas

Las cuotas compartidas funcionan como colas virtuales dentro de la cola de trabajos asociada entre las que el AWS Batch planificador itera al programar. Permiten a los administradores asignar una cuota de cómputo a un equipo o proyecto mediante una configuración explícita para la `capacity limits` estrategia de uso compartido de recursos.

Requisitos previos

Antes de crear cuotas compartidas, asegúrese de tener:

- Política de programación de la administración de cuotas, entorno de servicio y cola de trabajos: política de programación de administración de cuotas, entorno de servicio y cola de trabajos con la administración de cuotas habilitada. Para obtener más información, consulte [Crear recursos de administración de cuotas](#).
- Permisos de IAM: permisos para crear y gestionar AWS Batch cuotas compartidas. Para obtener más información, consulte [AWS Batch Políticas, funciones y permisos de IAM](#).

Configure quota shares (AWS Batch console)

1. Abra la AWS Batch consola en. <https://console.aws.amazon.com/batch/>
2. En el panel de navegación, elija Colas de trabajos y seleccione una cola de trabajos habilitada para la administración de cuotas de la lista. Elija el enlace con el nombre de la cola de trabajos.
3. En la sección Cuotas compartidas, selecciona Crear cuota compartida.
4. Proporcione un nombre para la cuota compartida.
5. En Límites de capacidad, selecciona Añadir límites de capacidad. Debe especificarse al menos un límite de capacidad.
 - a. Seleccione un tipo de instancia en el menú desplegable y establezca el número máximo de instancias que esta cuota le ha asignado.
 - b. (Opcional) Seleccione Añadir límites de capacidad y repita el procedimiento para adjuntar un máximo de cinco límites de capacidad.
6. Para compartir la capacidad, elija cómo esta cuota compartida comparte su capacidad con otras cuotas de la misma cola de trabajos:
 - Seleccione Reservar si la cuota no debe prestar o tomar prestado el cómputo inactivo.

- Seleccione Prestar si la cuota puede transferir el cómputo inactivo a otras cuotas compartidas.
 - Seleccione Prestar y tomar prestado si la cuota de cuota puede prestar y tomar prestado el cómputo inactivo, y el cómputo prestado se recuperará mediante la preferencia entre particiones cuando llegue el trabajo.
7. (Opcional) Para la preferencia entre acciones, elige si deseas habilitar o deshabilitar la preferencia entre acciones. Al habilitar la preferencia entre acciones, los trabajos de mayor prioridad se anteponen a los trabajos de menor prioridad que ya están en el mercado o en el estado. SCHEDULED STARTING RUNNING Si se desactiva la preferencia en el reparto, los trabajos de mayor prioridad esperarán a que haya capacidad disponible.
 8. Selecciona Crear cuota compartida.

Configure quota shares (AWS CLI)

Utilice el `create-quota-share` comando para crear una cuota compartida. Debe elegir una estrategia para compartir recursos y si desea habilitar la preferencia entre recursos compartidos.

Ejemplo de préstamo y préstamo

El siguiente ejemplo crea una cuota de cuota que permite prestar y tomar en préstamo capacidad ociosa, con un límite de préstamo del 100% de sus límites de capacidad configurados. También permite la priorización de recursos compartidos, de modo que los trabajos de mayor prioridad no tienen que esperar a que se completen los de menor prioridad que se hayan programado en SageMaker IA.

```
aws batch create-quota-share \  
  --quota-share-name lend_and_borrow_qs \  
  --job-queue my-qm-sagemaker-jq \  
  --capacity-limits maxCapacity=5,capacityUnit=ml.m6i.large \  
  --resource-sharing-configuration strategy=LEND_AND_BORROW,borrowLimit=100 \  
  --preemption-configuration inSharePreemption=ENABLED
```

Pongamos solo un ejemplo

Las cuotas compartidas se pueden configurar para que solo presten capacidad ociosa, pero no la tomen prestada por sí mismas. El siguiente ejemplo se combina LEND con la desactivación de la preferencia dentro de los recursos compartidos.

```
aws batch create-quota-share \  
  --quota-share-name lend_and_borrow_qs \  
  --job-queue my-qm-sagemaker-jq \  
  --capacity-limits maxCapacity=5,capacityUnit=ml.m6i.large \  
  --resource-sharing-configuration strategy=LEND,borrowLimit=100 \  
  --preemption-configuration inSharePreemption=DISABLED
```

```
--quota-share-name lend_qs \  
--job-queue my-qm-sagemaker-jq \  
--capacity-limits maxCapacity=8,capacityUnit=ml.m6i.large \  
--resource-sharing-configuration strategy=LEND \  
--preemption-configuration inSharePreemption=DISABLED
```

Ejemplo de reserva

Las cuotas compartidas también se pueden configurar para reservar la capacidad inactiva. Los trabajos recién enviados cuando una cuota compartida tiene capacidad inactiva pueden empezar antes, pero la utilización general de las colas será menor si una cuota compartida no tiene trabajos.

```
aws batch create-quota-share \  
--quota-share-name reserved_qs \  
--job-queue my-qm-sagemaker-jq \  
--capacity-limits maxCapacity=2,capacityUnit=ml.m6i.large \  
--resource-sharing-configuration strategy=RESERVE \  
--preemption-configuration inSharePreemption=DISABLED
```

Someter los trabajos a una cuota de cuota

Las colas de trabajos de administración de cuotas requieren que todos los trabajos especifiquen una cuota de cuota al enviar el trabajo. Para enviar los trabajos a una cuota compartida, especifique la entrada `quotaShareName`. [SubmitServiceJob](#) Opcionalmente, se `preemptionConfiguration` puede proporcionar una para limitar el número de intentos de prioridad antes de que se produzca un intento de trabajo. `FAILED` Para limitar el número de preferencias que puede tener un trabajo, establézcalo en el momento de la presentación `preemptionRetriesBeforeTermination` del [ServiceJobPreemptionConfiguration](#) trabajo.

Envíe un trabajo mediante el AWS CLI

En el siguiente ejemplo, se utiliza el `submit-service-job` comando para enviar un trabajo a una cuota compartida.

```
aws batch submit-service-job \  
--job-name "my-sagemaker-training-job" \  
--job-queue "my-sagemaker-job-queue" \  
--service-job-type "SAGEMAKER_TRAINING" \  
--quota-share-name lend_qs \  
--job-queue my-qm-sagemaker-jq \  
--capacity-limits maxCapacity=8,capacityUnit=ml.m6i.large \  
--resource-sharing-configuration strategy=LEND \  
--preemption-configuration inSharePreemption=DISABLED
```

```
--quota-share-name "my_quota_share" \  
--timeout-config '{"attemptDurationSeconds":3600}' \  
--scheduling-priority 5 \  
--service-request-payload '{\"TrainingJobName\": \"sagemaker-training-job-  
example\", \"AlgorithmSpecification\": {\"TrainingImage\": \"123456789012.dkr.ecr.us-  
east-1.amazonaws.com/pytorch-inference:1.8.0-cpu-py3\", \"TrainingInputMode  
\": \"File\", \"ContainerEntrypoint\": [\"sleep\", \"1\"]}, \"RoleArn\":  
\"arn:aws:iam::123456789012:role/SageMakerExecutionRole\", \"OutputDataConfig\":  
{\"S3OutputPath\": \"s3://example-bucket/model-output/\"}, \"ResourceConfig\":  
{\"InstanceType\": \"ml.m5.large\", \"InstanceCount\": 1, \"VolumeSizeInGB\": 1}}'
```

Definiciones de trabajo

AWS Batch Las definiciones de trabajos especifican cómo se van a ejecutar los trabajos. Si bien cada trabajo debe hacer referencia a una definición de trabajo, muchos de los parámetros especificados en dicha definición pueden ser ignorados en tiempo de ejecución.

Algunos de los atributos especificados en una definición de trabajo son:

- La imagen de Docker que se debe utilizar con el contenedor en el trabajo.
- Cuántos v CPUs y cuánta memoria usar con el contenedor.
- El comando que el contenedor debe ejecutar al iniciarse.
- Las variables de entorno que se deben transmitirse al contenedor cuando se inicia (si las hay).
- Los volúmenes de datos que deben utilizarse con el contenedor.
- Qué función de IAM (si la hay) debe utilizar su trabajo para obtener AWS permisos.

Contenido

- [Creación de una definición de trabajo de un solo nodo](#)
- [Creación de una definición de trabajo paralelo de varios nodos](#)
- [Plantilla de definición de trabajo que utiliza ContainerProperties](#)
- [Cree definiciones de trabajo mediante EcsProperties](#)
- [Uso del controlador de registros awslogs](#)
- [Especificación de información confidencial](#)
- [Autenticación de registros privados para trabajos](#)
- [Volúmenes de Amazon EFS](#)
- [Ejemplos de definición de trabajo](#)

Creación de una definición de trabajo de un solo nodo

Antes de poder ejecutar trabajos AWS Batch, debe crear una definición de trabajo. Este proceso varía ligeramente para los trabajos paralelos entre un solo nodo y de varios nodos. Este tema trata específicamente sobre cómo crear una definición de trabajo para un AWS Batch trabajo que no es un trabajo paralelo de varios nodos (también conocido como programación de bandas).

Puede crear una definición de trabajo paralelo de varios nodos en los recursos de Amazon Elastic Container Service. Para obtener más información, consulte [the section called “Creación de una definición de trabajo paralelo de varios nodos”](#).

Temas

- [Creación de una definición de trabajo de un solo nodo en los recursos de Amazon EC2](#)
- [Creación de una definición de trabajo de un solo nodo en los recursos de Fargate](#)
- [Creación de una definición de trabajo de un solo nodo en los recursos de Amazon EKS](#)
- [Creación de una definición de trabajo de un solo nodo con varios contenedores en los recursos de Amazon EC2](#)

Creación de una definición de trabajo de un solo nodo en los recursos de Amazon EC2

Siga estos pasos para crear una definición de trabajo de un solo nodo en recursos de Amazon Elastic Compute Cloud (Amazon EC2).

Para crear una nueva definición de trabajo en los recursos de Amazon EC2:

1. Abra la AWS Batch consola en. <https://console.aws.amazon.com/batch/>
2. En la barra de navegación, elija la Región de AWS que desee utilizar.
3. En el panel de navegación izquierdo, seleccione Definiciones de trabajos.
4. Seleccione Creación de.
5. Para el Tipo de orquestación, seleccione Amazon Elastic Compute Cloud (Amazon EC2).
6. Para la Configuración de la plataforma EC2, desactive Habilitar el procesamiento paralelo de varios nodos.
7. En Nombre, escriba un nombre único para la definición de trabajo. El nombre puede tener una longitud máxima de 128 caracteres. Puede contener letras mayúsculas y minúsculas, números, guiones (-) y guiones bajos (_).
8. (Opcional) En Tiempo de espera de ejecución, introduzca el valor del tiempo de espera (en segundos). El tiempo de espera de ejecución es el tiempo que transcurre antes de que finalice un trabajo pendiente. Si un intento supera la duración del tiempo de espera, el intento se detiene y pasa a un estado FAILED. Para obtener más información, consulte [Tiempos de espera de trabajo](#). El valor mínimo es de 60 segundos.

9. (Opcional) Activa la Prioridad de programación. Introduzca un valor de prioridad de programación entre 0 y 100. Los valores más altos tienen mayor prioridad.
10. (Opcional) En Intentos de trabajo, introduzca el número de veces que AWS Batch intenta mover el trabajo al estado RUNNABLE. Ingrese un número entero entre 1 y 10.
11. (Opcional) Para las Condiciones de la estrategia de reintento, seleccione Agregar evaluación al salir. Introduzca al menos un valor de parámetro y, a continuación, elija una Acción. Para cada conjunto de condiciones, la Acción debe estar configurada como Reintentar o Salir. Estas acciones significan lo siguiente:
 - Reintentar: AWS Batch se vuelve a intentar hasta alcanzar el número de intentos de trabajo que especificó.
 - Salir: AWS Batch deja de volver a intentar el trabajo.

 Important

Si elige Agregar evaluar al salir, debe configurar al menos un parámetro y elegir una Acción o Eliminar evaluar al salir.

12. (Opcional) Amplíe Etiquetas y, a continuación, elija Agregar etiqueta para agregar etiquetas al recurso. Elija Agregar nueva etiqueta e introduzca la clave y el valor opcional.
13. (Opcional) Active Propagar etiquetas para propagar etiquetas desde el trabajo y la definición del trabajo a la tarea de Amazon ECS.
14. Seleccione Página siguiente.
15. En la sección Configuración del contenedor:
 - a. En Imagen, elija la imagen Docker que desea utilizar para su trabajo. De manera predeterminada, las imágenes del registro de Docker Hub están disponibles. También es posible especificar otros repositorios con *repository-url/image:tag*. El nombre puede tener una longitud máxima de 225 caracteres. Puede contener letras mayúsculas y minúsculas, números, guiones medios (-), guiones bajos (_), dos puntos (:), barras inclinadas (/) y signos numéricos (#). Este parámetro se asigna a Image en la sección [Creación de un contenedor](#) de la [API remota de Docker](#) y el parámetro IMAGE de [docker run](#).

 Note

La arquitectura de la imagen de Docker debe coincidir con la arquitectura del procesador de los recursos informáticos en las que estén programadas. Por ejemplo, las imágenes de Docker basadas en ARM solo se pueden ejecutar en recursos informáticos basados en ARM.

- Las imágenes de los repositorios públicos de Amazon ECR utilizan las convenciones de nomenclatura completa `registry/repository[:tag]` o `registry/repository[@digest]` (por ejemplo, `public.ecr.aws/registry_alias/my-web-app:latest`).
 - Las imágenes de los repositorios de Amazon ECR utilizan la convención de nomenclatura completa `registry/repository[:tag]` (por ejemplo, `aws_account_id.dkr.ecr.region.amazonaws.com/my-web-app:latest`).
 - Las imágenes de los repositorios oficiales de Docker Hub utilizan un solo nombre (por ejemplo, `ubuntu` o `mongo`).
 - Las imágenes de otros repositorios de Docker Hub se clasifican con un nombre de organización (por ejemplo, `amazon/amazon-ecs-agent`).
 - Las imágenes de otros repositorios online se cualifican más con un nombre de dominio (por ejemplo, `quay.io/assemblyline/ubuntu`).
- b. En Comando, ingrese los comandos en el campo como su equivalente de matriz de cadenas JSON.

Este parámetro se asigna a `Cmd` en la sección [Creación de un contenedor](#) de la [API remota de Docker](#) y el parámetro `COMMAND` se corresponde con `docker run`. [Para obtener más información sobre el DockerCMD parámetro, consulta https://docs.docker.com/engine/reference/builder/#cmd.](#)

 Note

También puede usar valores predeterminados para la sustitución de parámetros y marcadores de posición en el comando. Para obtener más información, consulte [Parameters](#).

- c. (Opcional) En la función de ejecución, especifique una función de IAM que conceda permiso a los agentes de contenedores de Amazon ECS para realizar llamadas a la AWS API en su nombre. Esta característica utiliza roles de IAM de Amazon ECS para las tareas. Para obtener más información, consulte [Roles de IAM de ejecución de tareas de Amazon ECS](#) en la Guía para desarrolladores de Amazon Elastic Container Service.
- d. Para configurar el rol de trabajo, elija un rol de IAM que tenga permisos para. AWS APIs. Esta característica utiliza roles de IAM de Amazon ECS para las tareas. Para obtener más información, consulte [Roles de IAM para tareas](#) en la Guía para desarrolladores de Amazon Elastic Container Service.

 Note

Aquí solo se muestran roles con la relación de confianza del Rol de tarea de servicio de Amazon Elastic Container. Para obtener más información sobre cómo crear un rol de IAM para sus AWS Batch trabajos, consulte [Creación de un rol y una política de IAM para sus tareas en la Guía para](#) desarrolladores de Amazon Elastic Container Service.

- 16. En Parámetros, elija Agregar parámetros para añadir marcadores de sustitución de parámetros como pares Clave y Valores opcionales.
- 17. En la sección de Configuración del entorno:
 - a. Para v CPUs, introduzca el número de v CPUs que desea reservar para el contenedor. Este parámetro se corresponde con CpuShares en la sección [Creación de un contenedor](#) de la [API remota de Docker](#) y con la opción `--cpu-shares` de [docker run](#). Cada vCPU es equivalente a 1 024 cuotas de CPU. Debe especificar al menos una vCPU.
 - b. En Memoria, introduzca el límite de memoria disponible para el contenedor. Si su contenedor intenta superar la cantidad de memoria que ha especificado aquí, se detiene el contenedor. Este parámetro se corresponde con Memory en la sección [Creación de un contenedor](#) de la [API remota de Docker](#) y con la opción `--memory` de [docker run](#). Debe especificar al menos 4 MiB de memoria para un trabajo.

 Note

Para maximizar el uso de los recursos, priorice la memoria para los trabajos de un tipo de instancia específico. Para obtener más información, consulte [Administración de la memoria de recurso de computación](#).

- c. En Número de GPUs, elija el número de GPUs que desea reservar para el contenedor.
 - d. (Opcional) En el caso de Variables de entorno, seleccione Agregar variable de entorno para añadir variables de entorno como pares de nombre-valor. Estas variables se transfieren al contenedor.
 - e. (Opcional) En Secretos, seleccione Agregar secreto para añadir los secretos como pares de nombre-valor. Estos secretos están expuestos en el contenedor. [Para obtener más información, consulte: LogConfiguration Opciones secretas](#).
18. Seleccione Página siguiente.
19. En la sección de Configuración de Linux:
- a. En Usuario, introduzca el nombre de usuario a utilizar dentro del contenedor. Este parámetro se corresponde con `User` en la sección [Creación de un contenedor](#) de la [API remota de Docker](#) y con la opción `--user` de [docker run](#).
 - b. (Opcional) Para otorgar al contenedor de su trabajo permisos elevados en la instancia host (similares a los del usuario de `root`), arrastre el control deslizante Privilegiado hacia la derecha. Este parámetro se corresponde con `Privileged` en la sección [Creación de un contenedor](#) de la [API remota de Docker](#) y con la opción `--privileged` de [docker run](#).
 - c. (Opcional) Active la opción Habilitar el proceso `init` para ejecutar un proceso `init` dentro del contenedor. Este proceso reenvía señales y recoge procesos.
20. (Opcional) En la sección de Configuración de Filesystem:
- a. Active la opción Habilitar el sistema de archivos de solo lectura para eliminar el acceso de escritura al volumen.
 - b. En Tamaño de memoria compartida, introduzca el tamaño (en MiB) del `/dev/shm` volumen de .
 - c. En Tamaño de intercambio máximo, introduzca la cantidad total de memoria de intercambio (en MiB) que puede utilizar el contenedor.
 - d. En Intercambio, introduzca un valor entre 0 y 100 para indicar el comportamiento de intercambio del contenedor. Si no especifica un valor y el intercambio está activado,

el valor predeterminado es 60. [Para obtener más información, consulte:swappiness.LinuxParameters](#)

- e. (Opcional) Expandir Configuración adicional.
- f. (Opcional) En el caso de Tmpfs, seleccione Agregar tmpfs para añadir una montura tmpfs.
- g. (Opcional) En el caso de los Dispositivos, seleccione Agregar dispositivo para añadir un dispositivo:
 - i. En Container path (Ruta del contenedor), especifique la ruta de la instancia del contenedor que va a exponer el dispositivo asignado a la instancia del host. Si lo deja en blanco, se utiliza la ruta del host en el contenedor.
 - ii. En Host path (Ruta de host), especifique la ruta de un dispositivo de la instancia del host.
 - iii. En la página Permisos, haga clic en uno o varios permisos para aplicarlos al dispositivo. Los permisos disponibles son READ, WRITE y MKNOD.
- h. (Opcional) En Configuración de volúmenes, seleccione Agregar volumen para crear una lista de volúmenes que se transferirán al contenedor. Introduzca el Nombre y la Ruta de origen del volumen y, a continuación, seleccione Agregar volumen. También puede optar por activar Habilitar EFS.
- i. (Opcional) En Puntos de montaje, elija la configuración Agregar puntos de montaje para agregar puntos de montaje a los volúmenes de datos. Debe especificar el volumen de origen y la ruta del contenedor. Estos puntos de montaje se transfieren al Docker daemon de una instancia de contenedor. También puede elegir que el volumen sea de Solo lectura.
- j. (Opcional) En Configuración de Ulimits, seleccione Agregar ulimit para agregar un ulimits valor al contenedor. Introduzca los valores de Nombre, Límite flexible y Límite invariable y, a continuación, elija Agregar límite máximo.

21. En la sección de Propiedades de tareas:

- a. En Función de ejecución (condicional), elija una función que permita a los agentes de Amazon ECS realizar llamadas a la AWS API en su nombre. Para obtener más información sobre la creación de un Rol de ejecución, consulte [Tutorial: creación del rol de ejecución de IAM](#).
- b. Elija Habilitar el comando de ejecución ECS para permitir el acceso directo al shell del contenedor de Amazon ECS y omitir el sistema operativo del host. Debe elegir un Rol de tarea.

 Important

El comando Ejecución de ECS requiere que el sistema de archivos sea modificable.

- c. Para el rol de tarea, elija un rol de Amazon ECS Identity and Access Management (IAM) para permitir que el contenedor realice llamadas a la AWS API en su nombre. Para obtener más información, consulte [Rol de IAM de las tareas de Amazon ECS](#) en la Guía para desarrolladores de Amazon Elastic Container Service.

22. (Opcional) En la sección de Configuración de registro:

- a. En el Controlador de registro, elija el controlador de registro que desee utilizar. [Para obtener más información sobre los controladores de registro disponibles, consulte LogConfiguration: LogDriver.](#)

 Note

Se utiliza el controlador de registro `awslogs` de forma predeterminada.

- b. En Opciones, elija Agregar opción para agregar una opción. Introduzca un par nombre-valor y, a continuación, elija Agregar opción.
- c. En Secretos, seleccione Agregar secreto. Introduzca un par nombre-valor y, a continuación, seleccione Agregar secreto para añadir un secreto.

 Tip

[Para obtener más información, consulte: Opciones secretas. LogConfiguration](#)

23. Seleccione Página siguiente.

- 24. Para la Revisión de definición de trabajo, revise los pasos de configuración. Si necesita realizar cambios, elija Editar. Cuando haya terminado, seleccione Creación de definición de trabajo.

Creación de una definición de trabajo de un solo nodo en los recursos de Fargate

Complete los siguientes pasos para crear una definición de trabajo de un solo nodo en los recursos. AWS Fargate

Para crear una nueva definición de trabajo en los recursos de Fargate:

1. Abra la AWS Batch consola en. <https://console.aws.amazon.com/batch/>
2. En la barra de navegación superior, selecciona la Región de AWS que desees usar.
3. En el panel de navegación izquierdo, seleccione Definiciones de trabajos.
4. Seleccione Creación de.
5. En Tipo de orquestación, elija Fargate. Para obtener más información, consulte [Entornos de computación en Fargate](#).
6. En Nombre, escriba un nombre único para la definición de trabajo. El nombre puede tener una longitud máxima de 128 caracteres. Puede contener letras mayúsculas y minúsculas, números, guiones (-) y guiones bajos (_).
7. (Opcional) En Tiempo de espera de ejecución, introduzca el valor del tiempo de espera (en segundos). El tiempo de espera de ejecución es el tiempo que transcurre antes de que finalice un trabajo pendiente. Si un intento supera la duración del tiempo de espera, el intento se detiene y pasa a un estado FAILED. Para obtener más información, consulte [Tiempos de espera de trabajo](#). El valor mínimo es de 60 segundos.
8. (Opcional) Activa la Prioridad de programación. Introduzca un valor de prioridad de programación entre 0 y 100. Los valores más altos tienen mayor prioridad respecto a los valores más bajos.
9. (Opcional) Amplíe Etiquetas y, a continuación, elija Agregar etiqueta para agregar etiquetas al recurso. Active Propagar etiquetas para propagar etiquetas desde el trabajo y la definición del trabajo.
10. En la sección de Configuración de la plataforma Fargate:
 - a. Para la Plataforma de tiempo de ejecución, elija la arquitectura del entorno de computación.
 - b. En Familia de sistemas operativos, seleccione el sistema operativo para el entorno de computación.
 - c. En Arquitectura CPU, elija la arquitectura vCPU.
 - d. Para la versión de la plataforma Fargate, introduzca LATEST o una versión específica del entorno del tiempo de ejecución.
 - e. (Opcional) Active Asignar IP pública para asignar una dirección IP a una interfaz de redes de trabajo Fargate. Para que un trabajo que se ejecuta en una subred privada envíe tráfico saliente a Internet, la subred privada requiere que se conecte una puerta de enlace NAT para enrutar las solicitudes a Internet. Es posible que desee hacer esto para poder extraer imágenes de contenedores. Para obtener más información, consulte [Amazon](#)

[ECS task networking](#) (Integración en red de las tareas de Amazon ECS) en la Guía para desarrolladores de Amazon Elastic Container Service.

- f. (Opcional) En Almacenamiento efímero, introduzca la cantidad de almacenamiento efímero que se va a asignar a la tarea. La cantidad de almacenamiento efímero debe estar entre 21 GiB y 200 GiB. De forma predeterminada, se asignan 20 GiB de almacenamiento efímero si no ingresa un valor.

 Note

El almacenamiento efímero requiere la versión de la plataforma 1.4 Fargate o una posterior.

- g. En la función de ejecución, especifique una función de IAM que conceda permiso al contenedor de Amazon ECS y a los agentes de Fargate para AWS realizar llamadas a la API en su nombre. Esta característica utiliza roles de IAM de Amazon ECS para otorgarle funcionalidad a la tarea. Para obtener más información, incluidos los requisitos previos de configuración, consulte [Roles de IAM para tareas de ejecución de Amazon ECS](#) en la Guía para desarrolladores de Amazon Elastic Container Service.
- h. Para los Intentos de trabajo, introduzca el número de veces que AWS Batch intenta mover el trabajo al estado RUNNABLE. Ingrese un número entero entre 1 y 10.
- i. (Opcional) Para las Condiciones de la estrategia de reintento, seleccione Agregar evaluación al salir. Introduzca al menos un valor de parámetro y, a continuación, elija una Acción. Para cada conjunto de condiciones, la Acción debe estar configurada como Reintentar o Salir. Estas acciones significan lo siguiente:
- Reintentar: AWS Batch vuelve a intentarlo hasta alcanzar el número de intentos de trabajo que especificó.
 - Salir: AWS Batch deja de volver a intentar el trabajo.

 Important

Si elige Agregar evaluar al salir, debe configurar al menos un parámetro y elegir una Acción o Eliminar evaluar al salir.

11. Seleccione Página siguiente.

12. En la sección Configuración del contenedor:

- a. En Imagen, elija la imagen de Docker que desea utilizar para su trabajo. Por defecto, las imágenes del registro de Docker Hub están disponibles. También es posible especificar otros repositorios con *repository-url/image:tag*. El nombre puede tener una longitud máxima de 225 caracteres. Puede contener letras mayúsculas y minúsculas, números, guiones medios (-), guiones bajos (_), dos puntos (:), puntos (.), barras inclinadas (/) y signos numéricos (#). Este parámetro se asigna a Image en la sección [Creación de un contenedor](#) de la [API remota de Docker](#) y el parámetro IMAGE de [docker run](#).

 Note

La arquitectura de la imagen de Docker debe coincidir con la arquitectura del procesador de los recursos informáticos en las que estén programadas. Por ejemplo, las imágenes de Docker basadas en ARM solo se pueden ejecutar en recursos informáticos basados en ARM.

- Las imágenes de los repositorios públicos de Amazon ECR utilizan las convenciones de nomenclatura completa `registry/repository[:tag]` o `registry/repository[@digest]` (por ejemplo, `public.ecr.aws/registry_alias/my-web-app:latest`).
 - Las imágenes de los repositorios de Amazon ECR utilizan la convención de nomenclatura completa `registry/repository[:tag]` (por ejemplo, `aws_account_id.dkr.ecr.region.amazonaws.com/my-web-app:latest`).
 - Las imágenes de los repositorios oficiales de Docker Hub utilizan un solo nombre (por ejemplo, `ubuntu` o `mongo`).
 - Las imágenes de otros repositorios de Docker Hub se clasifican con un nombre de organización (por ejemplo, `amazon/amazon-ecs-agent`).
 - Las imágenes de otros repositorios online se cualifican más con un nombre de dominio (por ejemplo, `quay.io/assemblyline/ubuntu`).
- b. En Comando, ingrese los comandos en el campo como su equivalente de matriz de cadenas JSON.

Este parámetro se asigna a Cmd en la sección [Creación de un contenedor](#) de la [API remota de Docker](#) y el parámetro COMMAND se corresponde con [docker run](#). [Para obtener más](#)

información sobre el DockerCMD parámetro, consulta <https://docs.docker.com/engine/reference/builder/#cmd>.

 Note

También puede usar valores predeterminados para la sustitución de parámetros y marcadores de posición en el comando. Para obtener más información, consulte [Parameters](#).

- c. (Opcional) Añada parámetros a la definición del trabajo como asignaciones de nombre-valor para anular los valores predeterminados de la definición del trabajo. Para añadir un parámetro:
- En Parámetros, elija Agregar parámetros, introduzca un par de nombre-valor y, a continuación, elija Agregar parámetro.

 Important

Si elige Agregar parámetro, debe configurar al menos un parámetro o elegir Eliminar parámetro

- d. En la sección de Configuración del entorno:
- i. Para configurar el rol de Job, elija un rol de IAM que tenga permisos para. AWS APIs. Esta característica utiliza roles de IAM de Amazon ECS para otorgarle funcionalidad a la tarea. Para obtener más información, consulte [Roles de IAM para tareas](#) en la Guía para desarrolladores de Amazon Elastic Container Service.

 Note

Aquí solo se muestran roles con la relación de confianza del Rol de tarea de servicio de Amazon Elastic Container. Para obtener más información sobre cómo crear un rol de IAM para sus AWS Batch trabajos, consulte [Creación de un rol y una política de IAM para sus tareas en la Guía para desarrolladores de Amazon Elastic Container Service](#).

- ii. Para v CPUs, introduzca el número de v CPUs que desea reservar para el contenedor. Este parámetro se corresponde con CpuShares en la sección [Creación de un](#)

[contenedor](#) de la [API remota de Docker](#) y con la opción `--cpu-shares` de [docker run](#). Cada vCPU es equivalente a 1 024 cuotas de CPU. Debe especificar al menos una vCPU.

- iii. En Memoria, introduzca el límite de memoria disponible para el contenedor. Si su contenedor intenta superar la memoria especificada aquí, el contenedor se detiene. Este parámetro se corresponde con Memory en la sección [Creación de un contenedor](#) de la [API remota de Docker](#) y con la opción `--memory` de [docker run](#). Debe especificar al menos 4 MiB de memoria para un trabajo.

Si utiliza GuardDuty Runtime Monitoring, existe una ligera sobrecarga de memoria para el agente GuardDuty de seguridad. Por lo tanto, el límite de memoria debe incluir el tamaño del agente de GuardDuty seguridad. Para obtener información sobre los límites de memoria del agente de GuardDuty seguridad, consulte los [límites de CPU y memoria](#) en la Guía del GuardDuty usuario. Para obtener información sobre las prácticas recomendadas, consulte [Cómo solucionar los errores de falta de memoria en mis tareas de Fargate después de activar Runtime Monitoring](#) en la Guía para desarrolladores de Amazon ECS.

 Note

Para maximizar el uso de los recursos, priorice la memoria para los trabajos de un tipo de instancia específico. Para obtener más información, consulte [Administración de la memoria de recurso de computación](#).

- e. (Opcional) En el caso de Variables de entorno, seleccione Agregar variable de entorno para añadir variables de entorno como pares de nombre-valor. Estas variables se transfieren al contenedor.
 - f. (Opcional) En Secretos, seleccione Agregar secreto para añadir los secretos como pares de nombre-valor. Estos secretos están expuestos en el contenedor. [Para obtener más información, consulte: LogConfiguration Opciones secretas](#).
 - g. Seleccione Página siguiente.
13. (Opcional) En la sección de Configuración de Linux:
- a. En Usuario, introduzca el nombre de usuario a utilizar dentro del contenedor.
 - b. Active la opción Habilitar el proceso para ejecutar un proceso dentro del contenedor. Este proceso reenvía señales y recoge procesos.

- c. Active la opción Habilitar el sistema de archivos de solo lectura para eliminar el acceso de escritura al volumen.
 - d. (Opcional) Expandir Configuración adicional.
 - e. Para la Configuración de puntos de montaje, elija la configuración Agregar puntos de montaje para agregar puntos de montaje a los volúmenes de datos. Debe especificar el volumen de origen y la ruta del contenedor. Estos puntos de montaje se transfieren al Docker daemon de una instancia de contenedor.
 - f. Para la Configuración de volúmenes, seleccione Agregar volumen para crear una lista de volúmenes que se transferirán al contenedor. Introduzca el nombre y la ruta de origen del volumen y, a continuación, seleccione Agregar volumen.
 - g. En la sección de Propiedades de tareas:
 - i. En Función de ejecución (condicional), elija una función que permita a los agentes de Amazon ECS realizar llamadas a la AWS API en su nombre. Para obtener más información sobre la creación de un Rol de ejecución, consulte [Tutorial: creación del rol de ejecución de IAM](#).
 - ii. Elija Habilitar el comando de ejecución ECS para permitir el acceso directo al shell del contenedor de Amazon ECS y omitir el sistema operativo del host. Debe elegir un Rol de tarea.
-  **Important**

El comando Ejecución de ECS requiere que el sistema de archivos sea modificable.
- iii. Para el rol de tarea, elija un rol de Amazon ECS Identity and Access Management (IAM) para permitir que el contenedor realice llamadas a la AWS API en su nombre. Para obtener más información, consulte [Rol de IAM de las tareas de Amazon ECS](#) en la Guía para desarrolladores de Amazon Elastic Container Service.
 - h. En la sección de Configuración de registro:
 - i. (Opcional) En el Controlador de registro, elija el controlador de registro que desee utilizar. [Para obtener más información sobre los controladores de registro disponibles, consulte LogConfiguration: LogDriver.](#)

Note

Se utiliza el controlador de registro `awslogs` de forma predeterminada.

- ii. (Opcional) En Opciones, elija Agregar opción para agregar una opción. Introduzca un par nombre-valor y, a continuación, elija Agregar opción.
- iii. (Opcional) En Secretos, seleccione Agregar secreto para añadir un secreto. A continuación, introduzca un par nombre-valor y seleccione Agregar secreto.

Tip

[Para obtener más información, consulte: Opciones secretas. LogConfiguration](#)

- 14. Seleccione Página siguiente.
- 15. Para la Revisión de definición de trabajo, revise los pasos de configuración. Si necesita realizar cambios, elija Editar. Cuando haya terminado, seleccione Creación de definición de trabajo.

Creación de una definición de trabajo de un solo nodo en los recursos de Amazon EKS

Siga estos pasos para crear una definición de trabajo de un solo nodo en recursos de Amazon Elastic Kubernetes Service (Amazon EKS).

Para crear una nueva definición de trabajo en los recursos de Amazon EKS:

- 1. Abra la AWS Batch consola en. <https://console.aws.amazon.com/batch/>
- 2. En la barra de navegación superior, selecciona la Región de AWS que desees usar.
- 3. En el panel de navegación izquierdo, seleccione Definiciones de trabajos.
- 4. Seleccione Creación de.
- 5. Para el tipo de orquestación, elija Elastic Kubernetes Service (EKS).
- 6. En Nombre, escriba un nombre único para la definición de trabajo. El nombre puede tener una longitud máxima de 128 caracteres. Puede contener letras mayúsculas y minúsculas, números, guiones (-) y guiones bajos (_).
- 7. (Opcional) En Tiempo de espera de ejecución, introduzca el valor del tiempo de espera (en segundos). El tiempo de espera de ejecución es el tiempo que transcurre antes de que finalice

un trabajo pendiente. Si un intento supera la duración del tiempo de espera, el intento se detiene y pasa a un estado FAILED. Para obtener más información, consulte [Tiempos de espera de trabajo](#). El valor mínimo es de 60 segundos.

8. (Opcional) Activa la Prioridad de programación. Introduzca un valor de prioridad de programación entre 0 y 100. Los valores más altos tienen mayor prioridad respecto a los valores más bajos.
9. (Opcional) Amplíe Etiquetas y, a continuación, elija Agregar etiqueta para agregar etiquetas al recurso.
10. Seleccione Página siguiente.
11. En la sección de propiedades de pod EKS:
 - a. En Nombre de cuenta de servicio, introduzca una cuenta que proporcione una identidad para los procesos que se ejecutan en un pod.
 - b. Active la Red de host para usar el modelo de red Kubernetes pod y abra un puerto de escucha para las conexiones entrantes. Desactive esta configuración solo para las comunicaciones salientes.
 - c. Para la política DNS, elija una de las siguientes opciones:
 - Sin valor (nulo): pod ignora la configuración de DNS del entorno Kubernetes.
 - Predeterminado: pod hereda la configuración de resolución de nombres del nodo en el que se ejecuta.

 Note

Si no se especifica una política de DNS, la política de DNS Valor predeterminado no es la política de DNS predeterminada. En su lugar, ClusterFirstse usa.

- ClusterFirst— Cualquier consulta de DNS que no coincida con el sufijo del dominio del clúster configurado se reenvía al servidor de nombres ascendente que se hereda del nodo.
 - ClusterFirstWithHostNet— Úselo si la red host está activada.
- d. (Opcional) En Volúmenes, seleccione Agregar volumen y, a continuación:
 - i. Agregue un Nombre para su volumen.
 - ii. (Opcional) Agregue la Ruta del host al directorio del host.

- iii. (Opcional) Agregue un límite Medio y de Tamaño para configurar un [emptyDir de Kubernetes](#).
 - iv. (Opcional) Proporcione un Nombre del secreto para el pod e indique si el secreto es Opcional.
 - v. (Opcional) Defina un Nombre de recuperación para adjuntar una [Recuperación de volumen persistente](#) de Kubernetes al pod e indique si es de Solo lectura.
- e. (Opcional) En Etiquetas de los pods, seleccione Agregar etiquetas de pod y, a continuación, introduzca un par de nombre-valor.

 Important

El prefijo de una etiqueta de pod no puede contener `kubernetes.io/`, `k8s.io/` o `batch.amazonaws.com/`.

- f. (Opcional) Para las Anotaciones del pod, seleccione Agregar anotaciones y, a continuación, ingrese un par nombre-valor.

 Important

El prefijo de una anotación de pod no puede contener `kubernetes.io/`, `k8s.io/` o `batch.amazonaws.com/`.

- g. Seleccione Página siguiente.
- h. En la sección Configuración del contenedor:
- i. Para el Nombre, escriba un nombre único para el contenedor. El nombre debe empezar por una letra o un número y puede tener un máximo de 63 caracteres. Puede contener letras mayúsculas y minúsculas, números y guiones (-).
 - ii. En Imagen, elija la imagen Docker que desea utilizar para su trabajo. Por defecto, las imágenes del registro de Docker Hub están disponibles. También es posible especificar otros repositorios con *repository-url/image:tag*. El nombre puede tener una longitud máxima de 225 caracteres. Puede contener letras mayúsculas y minúsculas, números, guiones medios (-), guiones bajos (_), dos puntos (:), puntos (.), barras inclinadas (/) y signos numéricos (#). Este parámetro se asigna a Image en la sección [Creación de un contenedor](#) de la [API remota de Docker](#) y el parámetro IMAGE de [docker run](#)

 Note

La arquitectura de la imagen de Docker debe coincidir con la arquitectura del procesador de los recursos informáticos en las que estén programadas. Por ejemplo, las imágenes de Docker basadas en ARM solo se pueden ejecutar en recursos informáticos basados en ARM.

- Las imágenes de los repositorios públicos de Amazon ECR utilizan las convenciones de nomenclatura completa `registry/repository[:tag]` o `registry/repository[@digest]` (por ejemplo, `public.ecr.aws/registry_alias/my-web-app:latest`).
 - Las imágenes de los repositorios de Amazon ECR utilizan la convención de nomenclatura completa `registry/repository[:tag]` (por ejemplo, `aws_account_id.dkr.ecr.region.amazonaws.com/my-web-app:latest`).
 - Las imágenes de los repositorios oficiales de Docker Hub utilizan un solo nombre (por ejemplo, `ubuntu` o `mongo`).
 - Las imágenes de otros repositorios de Docker Hub se clasifican con un nombre de organización (por ejemplo, `amazon/amazon-ecs-agent`).
 - Las imágenes de otros repositorios online se cualifican más con un nombre de dominio (por ejemplo, `quay.io/assemblyline/ubuntu`).
- iii. (Opcional) En Política de extracción de imágenes, elija cuándo se extraerán las imágenes.
 - iv. (Opcional) En Comando, ingrese un comando JSON para pasarlo al contenedor.
 - v. (Opcional) En Argumentos, introduzca los argumentos para pasar al contenedor. Si no se proporciona un argumento, se utiliza el comando contenedor de imágenes.
- i. (Opcional) Puede añadir parámetros a la definición del trabajo como asignaciones de nombre-valor para anular los valores predeterminados de la definición del trabajo. Para añadir un parámetro:
 - En Parámetros, introduzca un par de nombre-valor y, a continuación, elija Agregar parámetro.

 Important

Si elige Agregar parámetro, debe configurar al menos un parámetro o elegir Eliminar parámetro

- j. En la sección de Configuración del entorno:
 - i. Para v CPUs, introduce el número de v que deseas reservar CPUs para el contenedor. Este parámetro se corresponde con CpuShares en la sección [Creación de un contenedor](#) de la [API remota de Docker](#) y con la opción `--cpu-shares` de [docker run](#). Cada vCPU es equivalente a 1 024 cuotas de CPU. Debe especificar al menos una vCPU.
 - ii. En Memoria, introduzca el límite de memoria disponible para el contenedor. Si su contenedor intenta superar la memoria especificada aquí, el contenedor se detiene. Este parámetro se corresponde con Memory en la sección [Creación de un contenedor](#) de la [API remota de Docker](#) y con la opción `--memory` de [docker run](#). Debe especificar al menos 4 MiB de memoria para un trabajo.

 Note

Para maximizar el uso de los recursos, priorice la memoria para los trabajos de un tipo de instancia específico. Para obtener más información, consulte [Administración de la memoria de recurso de computación](#).

- k. (Opcional) En el caso de Variables de entorno, seleccione Agregar variable de entorno para añadir variables de entorno como pares de nombre-valor. Estas variables se transfieren al contenedor.
- l. (Opcional) En Montaje del volumen:
 - i. Seleccione Agregar montaje del volumen.
 - ii. Introduzca un Nombre y, a continuación, introduzca una Ruta de montaje en el contenedor donde está montado el volumen. Introduzca SubPatha para especificar una subruta dentro del volumen al que se hace referencia en lugar de su raíz.
 - iii. Seleccione Solo lectura para eliminar los permisos de escritura del volumen.
 - iv. Seleccione Agregar montaje del volumen.

- m. (Opcional) En Ejecutar como usuario, introduzca un ID de usuario para ejecutar el proceso del contenedor.

 Note

El ID de usuario debe existir en la imagen para que se ejecute el contenedor.

- n. (Opcional) En Ejecutar como grupo, introduzca un ID de grupo para ejecutar el proceso del contenedor en tiempo de ejecución.

 Note

El ID de grupo debe existir en la imagen para que se ejecute el contenedor.

- o. (Opcional) Para otorgar al contenedor de su trabajo permisos elevados en la instancia host (similares a los del usuario de root), arrastre el control deslizante Privilegiado hacia la derecha. Este parámetro se corresponde con Privileged en la sección [Creación de un contenedor](#) de la [API remota de Docker](#) y con la opción `--privileged` de [docker run](#).
- p. (Opcional) Active el sistema de Archivos raíz de solo lectura para eliminar el acceso de escritura al sistema de archivos raíz.
- q. (Opcional) Active la opción Ejecutar como usuario no raíz para ejecutar los contenedores en el pod como usuario no raíz.

 Note

Si la opción Ejecutar como no raíz está activada, kubelet valida la imagen en tiempo de ejecución para comprobar que la imagen no se ejecuta como UID 0.

- r. Seleccione Página siguiente.
12. Para la Revisión de definición de trabajo, revise los pasos de configuración. Si necesita realizar cambios, elija Editar. Cuando haya terminado, seleccione Creación de definición de trabajo.

Creación de una definición de trabajo de un solo nodo con varios contenedores en los recursos de Amazon EC2

Siga estos pasos para crear una definición de trabajo de un solo nodo con varios contenedores en los recursos de Amazon Elastic Compute Cloud (Amazon EC2).

Para crear una nueva definición de trabajo en los recursos de Amazon EC2:

1. Abra la AWS Batch consola en. <https://console.aws.amazon.com/batch/>
2. En la barra de navegación, elija la Región de AWS que desee utilizar.
3. En el panel de navegación izquierdo, seleccione Definiciones de trabajos.
4. Seleccione Creación de.
5. Para el Tipo de orquestación, seleccione Amazon Elastic Compute Cloud (Amazon EC2).
6. En Estructura de definición de trabajos, desactive Uso del procesamiento de estructura de containerProperties heredado.
7. Para la Configuración de la plataforma EC2, desactive Habilitar el procesamiento paralelo de varios nodos.
8. Elija Siguiente.
9. En la sección Configuración general, ingrese lo siguiente:
 - a. En Nombre, escriba un nombre único para la definición de trabajo. El nombre puede tener una longitud máxima de 128 caracteres. Puede contener letras mayúsculas y minúsculas, números, guiones (-) y guiones bajos (_).
 - b. En Tiempo de espera de ejecución: opcional, ingrese el valor del tiempo de espera (en segundos). El tiempo de espera de ejecución es el tiempo que transcurre antes de que finalice un trabajo pendiente. Si un intento supera la duración del tiempo de espera, el intento se detiene y pasa a un estado FAILED. Para obtener más información, consulte [Tiempos de espera de trabajo](#). El valor mínimo es de 60 segundos.
 - c. Activa la Prioridad de programación: opcional. Introduzca un valor de prioridad de programación entre 0 y 100. Los valores más altos tienen mayor prioridad.
 - d. Amplíe Etiquetas: opcional y, a continuación, elija Agregar etiqueta para agregar etiquetas al recurso. Elija Agregar nueva etiqueta e introduzca la clave y el valor opcional.
 - e. Active Propagar etiquetas para propagar etiquetas desde el trabajo y la definición del trabajo a la tarea de Amazon ECS.
10. En la sección Estrategia de reintento: opcional, ingrese lo siguiente:
 - a. En Intentos de trabajo, ingrese el número de veces que AWS Batch intenta mover el trabajo al estado RUNNABLE. Ingrese un número entero entre 1 y 10.
 - b. En Condiciones de la estrategia de reintento, seleccione Agregar evaluación al salir. Introduzca al menos un valor de parámetro y, a continuación, elija una Acción. Para cada

conjunto de condiciones, la Acción debe estar configurada como Reintentar o Salir. Estas acciones significan lo siguiente:

- Reintentar: AWS Batch se vuelve a intentar hasta alcanzar el número de intentos de trabajo que especificó.
- Salir: AWS Batch deja de volver a intentar el trabajo.

 Important

Si elige Agregar evaluar al salir, debe configurar al menos un parámetro y elegir una Acción o Eliminar evaluar al salir.

11. En la sección Propiedades de la tarea, ingrese lo siguiente:

- a. En Función de ejecución (condicional), elija una función que permita a los agentes de Amazon ECS realizar llamadas a la AWS API en su nombre. Para obtener más información sobre la creación de un Rol de ejecución, consulte [Tutorial: creación del rol de ejecución de IAM](#).
- b. Elija Habilitar el comando de ejecución ECS para permitir el acceso directo al shell del contenedor de Amazon ECS y omitir el sistema operativo del host. Debe elegir un Rol de tarea.

 Important

El comando Ejecución de ECS requiere que el sistema de archivos sea modificable.

- c. Para el rol de tarea, elija un rol de Amazon ECS Identity and Access Management (IAM) para permitir que el contenedor realice llamadas a la AWS API en su nombre. Para obtener más información, consulte [Rol de IAM de las tareas de Amazon ECS](#) en la Guía para desarrolladores de Amazon Elastic Container Service.
- d. En Modo IPC, seleccione host, task o none. Si se especifica host, todos los contenedores que están dentro de las tareas que tienen especificado el modo de IPC del host en la misma instancia de contenedor comparten los mismos recursos de IPC con la instancia Amazon EC2 del host. Si se especifica la tarea, todos los contenedores que están dentro de la tarea especificada comparten los mismos recursos de IPC. Si no se especifica nada, los recursos de IPC dentro de los contenedores de una tarea son privados y no se comparten con otros contenedores en una tarea o en la instancia de contenedor. Si no

se especifica ningún valor, el uso compartido del espacio de nombre de recursos de IPC depende de la configuración del daemon de Docker en la instancia de contenedor.

- e. En Modo PID, elija `host` o `task`. Por ejemplo, la supervisión de los archivos sidecar puede necesitar `pidMode` para acceder a información sobre otros contenedores que se ejecutan en la misma tarea. Si se especifica `host`, todos los contenedores dentro de las tareas que tienen especificado el modo de PID del `host` en la misma instancia de contenedor comparten el mismo espacio de nombres del proceso con la instancia Amazon EC2 del `host`. Si se especifica `task`, todos los contenedores dentro de la tarea especificada comparten el mismo espacio de nombres del proceso. Si no se especifica ningún valor, el valor predeterminado es un espacio de nombre privado para cada contenedor.

12. En la sección Recursos consumibles, ingrese lo siguiente:

- a. Ingrese un Nombre único y el Valor solicitado.
- b. Puede agregar más recursos consumibles seleccionando Agregar recurso consumible.

13. En la sección Almacenamiento, ingrese lo siguiente:

- a. Ingrese el Nombre y la Ruta de origen del volumen y, a continuación, seleccione Agregar volumen. También puede optar por activar Habilitar EFS.
- b. Puede agregar más volúmenes seleccionando Agregar volumen.

14. En Parámetros, elija Agregar parámetros para añadir marcadores de sustitución de parámetros como pares Clave y Valores opcionales.

15. Seleccione Página siguiente.

16. En la sección Configuración del contenedor:

- a. En Name (Nombre), escriba un nombre para el contenedor.
- b. En Contenedor esencial, actívelo si el contenedor es esencial.
- c. En Imagen, elija la imagen Docker que desea utilizar para su trabajo. De manera predeterminada, las imágenes del registro de Docker Hub están disponibles. También es posible especificar otros repositorios con `repository-url/image:tag`. El nombre puede tener una longitud máxima de 225 caracteres. Puede contener letras mayúsculas y minúsculas, números, guiones medios (-), guiones bajos (_), dos puntos (:), barras inclinadas (/) y signos numéricos (#). Este parámetro se asigna a `Image` en la sección [Creación de un contenedor](#) de la [API remota de Docker](#) y el parámetro `IMAGE` de [docker run](#).

 Note

La arquitectura de la imagen de Docker debe coincidir con la arquitectura del procesador de los recursos informáticos en las que estén programadas. Por ejemplo, las imágenes de Docker basadas en ARM solo se pueden ejecutar en recursos informáticos basados en ARM.

- Las imágenes de los repositorios públicos de Amazon ECR utilizan las convenciones de nomenclatura completa `registry/repository[:tag]` o `registry/repository[@digest]` (por ejemplo, `public.ecr.aws/registry_alias/my-web-app:latest`).
 - Las imágenes de los repositorios de Amazon ECR utilizan la convención de nomenclatura completa `registry/repository[:tag]` (por ejemplo, `aws_account_id.dkr.ecr.region.amazonaws.com/my-web-app:latest`).
 - Las imágenes de los repositorios oficiales de Docker Hub utilizan un solo nombre (por ejemplo, `ubuntu` o `mongo`).
 - Las imágenes de otros repositorios de Docker Hub se clasifican con un nombre de organización (por ejemplo, `amazon/amazon-ecs-agent`).
 - Las imágenes de otros repositorios online se cualifican más con un nombre de dominio (por ejemplo, `quay.io/assemblyline/ubuntu`).
- d. En Requisitos de recursos, configure cada una de las siguientes opciones:
- i. Para v CPUs, elige el número de CPUs para el contenedor.
 - ii. En Memoria, elija la cantidad de memoria para el contenedor.
 - iii. Para GPU (opcional), elige el número de GPUs para el contenedor.
- e. En Usuario, introduzca el nombre de usuario a utilizar dentro del contenedor.
- f. Active la opción Habilitar el sistema de archivos de solo lectura para eliminar el acceso de escritura al volumen.
- g. Active Privilegiado para otorgar al contenedor de su trabajo permisos elevados en la instancia host, similares a los del usuario raíz.
- h. En Comando, ingrese los comandos en el campo como su equivalente de matriz de cadenas JSON.

Este parámetro se asigna a Cmd en la sección [Creación de un contenedor](#) de la [API remota de Docker](#) y el parámetro COMMAND se corresponde con [docker run](#). Para obtener más información sobre el Docker CMD parámetro, consulta <https://docs.docker.com/engine/reference/builder/#cmd>.

 Note

También puede usar valores predeterminados para la sustitución de parámetros y marcadores de posición en el comando. Para obtener más información, consulte [Parameters](#).

- i. En Credenciales del repositorio: opcional, ingrese el ARN del secreto que contiene sus credenciales.
- j. En Variables de entorno: opcional, seleccione Agregar variables de entorno para agregar variables de entorno y pasarlas al contenedor.
- k. En la sección Parámetros de Linux: opcional:
 - i. Active la opción Habilitar el proceso para ejecutar un proceso dentro del contenedor.
 - ii. En Tamaño de memoria compartida, ingrese el tamaño (en MiB) del volumen de /dev/shm.
 - iii. En Tamaño de intercambio máximo, introduzca la cantidad total de memoria de intercambio (en MiB) que puede utilizar el contenedor.
 - iv. En Intercambio, introduzca un valor entre 0 y 100 para indicar el comportamiento de intercambio del contenedor. Si no especifica un valor y el intercambio está activado, el valor predeterminado es 60.
 - v. En el caso de los Dispositivos, seleccione Agregar dispositivo para añadir un dispositivo:
 - A. En Container path (Ruta del contenedor), especifique la ruta de la instancia del contenedor que va a exponer el dispositivo asignado a la instancia del host. Si lo deja en blanco, se utiliza la ruta del host en el contenedor.
 - B. En Host path (Ruta de host), especifique la ruta de un dispositivo de la instancia del host.
 - C. En la página Permisos, haga clic en uno o varios permisos para aplicarlos al dispositivo. Los permisos disponibles son READ, WRITE y MKNOD.

vi. En el caso de Tmpfs, seleccione Agregar tmpfs para añadir una montura tmpfs.

I.

 Note

El registro de FireLens debe realizarse en un contenedor dedicado. Cómo configurar el registro de FireLens:

- En todos los contenedores, excepto en el contenedor dedicado a FireLens, configure el Controlador de registro en `awsfirelens`.
- En el contenedor de FireLens, configure la Configuración de FireLens: opcional y la Configuración de registro: opcional para el destino del registro.

En la sección Configuración de FireLens: opcional:

 Important

AWS Batch aplica el modo de host red en trabajos Amazon ECS que no son de MNP ni de Fargate. El [Usuario raíz es obligatorio](#) para Amazon ECS FireLens. Para [mayor seguridad](#), cuando ejecute tareas utilizando el modo de red host, Amazon ECS recomienda no ejecutar contenedores con el usuario raíz (UID 0). Por lo tanto, todos los trabajos de ECS que no sean de MNP ni de Fargate y que registren FireLens no cumplirán con las prácticas recomendadas de seguridad.

i. En Tipo, elija `fluentd` o `fluentbit`.

ii. En Opciones, introduzca el par de opciones. `name/value` Puede agregar más Opciones mediante la Opción agregada.

m. En la sección de Configuración de registro: opcional:

i. En el Controlador de registro, elija el controlador de registro que desee utilizar. [Para obtener más información sobre los controladores de registro disponibles, consulte: LogConfiguration LogDriver.](#)

 Note

Se utiliza el controlador de registro `awslogs` de forma predeterminada.

- ii. En Opciones, elija Agregar opción para agregar una opción. Introduzca un par nombre-valor y, a continuación, elija Agregar opción.
- iii. En Secretos, seleccione Agregar secreto. Introduzca un par nombre-valor y, a continuación, seleccione Agregar secreto para añadir un secreto.

 Tip

[Para obtener más información, consulte: Opciones secretas. LogConfiguration](#)

- n. En Puntos de montaje: opcional, elija la configuración Agregar puntos de montaje para agregar puntos de montaje a los volúmenes de datos. Debe especificar el volumen de origen y la ruta del contenedor.
- o. En Secretos: opcional, seleccione Agregar secreto para añadir un secreto. A continuación, introduzca un par nombre-valor y seleccione Agregar secreto.

 Tip

[Para obtener más información, consulte: Opciones secretas. LogConfiguration](#)

- p. En Unlimits: opcional, seleccione Agregar ulimit para agregar un valor de `ulimits` al contenedor. Introduzca los valores de Nombre, Límite flexible y Límite invariable y, a continuación, elija Agregar límite máximo.
 - q. En Dependencias: opcional, elija Agregar dependencias de contenedor. Elija el nombre del contenedor y su estado para determinar cuándo comienza.
17. Si solo tiene un contenedor configurado, debe elegir Agregar contenedor y completar la configuración del nuevo contenedor. De lo contrario, elija Siguiente para revisarla.

Creación de una definición de trabajo paralelo de varios nodos

Antes de poder ejecutar trabajos AWS Batch, debe crear una definición de trabajo. Este proceso varía ligeramente para los trabajos paralelos entre un solo nodo y de varios nodos. Este tema trata específicamente sobre cómo crear una definición de trabajo para un trabajo paralelo de AWS Batch varios nodos (también conocido como programación de bandas). Para obtener más información, consulte [Trabajos paralelos de varios nodos](#).

 Note

AWS Fargate no admite trabajos paralelos de varios nodos.

Contenido

- [Tutorial: creación de una definición de trabajo paralelo de varios nodos en los recursos de Amazon EC2](#)

Tutorial: creación de una definición de trabajo paralelo de varios nodos en los recursos de Amazon EC2

Para crear una definición de trabajo paralelo de varios nodos en recursos de Amazon Elastic Compute Cloud (Amazon EC2).

 Note

Para crear una definición de trabajo de un solo nodo, consulte [Creación de una definición de trabajo de un solo nodo en los recursos de Amazon EC2](#).

Para crear una definición de trabajo paralelo de varios nodos en los recursos de Amazon EC2:

1. Abra la AWS Batch consola en. <https://console.aws.amazon.com/batch/>
2. En la barra de navegación, seleccione la Región de AWS que desee utilizar.
3. En el panel de navegación, seleccione Definiciones de trabajo.
4. Seleccione Creación de.
5. Para el Tipo de orquestación, seleccione Amazon Elastic Compute Cloud (Amazon EC2).
6. En Habilitar el paralelo de varios nodos, active el paralelo de varios nodos.
7. En Nombre, escriba un nombre único para la definición de trabajo. El nombre puede tener una longitud máxima de 128 caracteres y puede contener mayúsculas y minúsculas, números, guiones (-) y guiones bajos (_).
8. (Opcional) En Tiempo de espera de ejecución, especifique el número máximo de segundos que desea permitir que el trabajo intente ejecutarse. Si un intento supera la duración del tiempo

de espera, el intento se detiene y pasa a un estado FAILED. Para obtener más información, consulte [Tiempos de espera de trabajo](#).

9. (Opcional) Activa la Prioridad de programación. Introduzca un valor de prioridad de programación entre 0 y 100. Los valores más altos tienen mayor prioridad respecto a los valores más bajos.
10. (Opcional) En Intentos de trabajo, introduzca el número de veces que AWS Batch intenta mover el trabajo al estado RUNNABLE. Ingrese un número entero entre 1 y 10.
11. (Opcional) Para las Condiciones de la estrategia de reintento, seleccione Agregar evaluación al salir. Introduzca al menos un valor de parámetro y, a continuación, elija una Acción. Para cada conjunto de condiciones, la Acción debe estar configurada como Reintentar o Salir. Estas acciones significan lo siguiente:
 - Reintentar: AWS Batch se vuelve a intentar hasta alcanzar el número de intentos de trabajo que especificó.
 - Salir: AWS Batch deja de volver a intentar el trabajo.

 Important

Si elige Agregar evaluar al salir, debe configurar al menos un parámetro y elegir una Acción o Eliminar evaluar al salir.

12. (Opcional) Amplíe Etiquetas y, a continuación, elija Agregar etiqueta para agregar etiquetas al recurso. Elija Agregar nueva etiqueta e introduzca la clave y el valor opcional. También puede activar Propagar etiquetas para propagar etiquetas desde el trabajo y la definición del trabajo a la tarea de Amazon ECS.
13. Seleccione Página siguiente.
14. En Number of nodes (Número de nodos), introduzca el número total de nodos que desea utilizar en el trabajo.
15. En Main node (Nodo principal), introduzca el índice de nodo que desea utilizar para el nodo principal. El índice de nodo principal predeterminado es 0.
16. En Tipo de instancia, elija un tipo de instancia.

 Note

El tipo de instancia que elija se aplica a todos los nodos.

17. En Parámetros, elija Agregar parámetros para añadir marcadores de sustitución de parámetros como pares Clave y Valores opcionales.
18. En la sección Rangos de nodos:
 - a. Seleccione Agregar rango de nodos. Esto crea una sección de Rango de nodos.
 - b. En Target nodes (Nodos de destino), especifique el rango del grupo de nodos utilizando la notación *range_start:range_end*.

Puede crear hasta cinco rangos de nodos para los nodos que ha especificado para el trabajo. Los rangos de nodos utilizan el valor de índice para un nodo, y el índice de nodo comienza a partir de 0. Asegúrese de que el valor del índice final del rango del grupo de nodos final sea uno menos que el número de nodos que especificó. Por ejemplo, supongamos que ha especificado 10 nodos y desea utilizar un único grupo de nodos. Entonces, el rango final será 9.

- c. En Imagen, elija la imagen Docker que desea utilizar para su trabajo. De manera predeterminada, las imágenes del registro de Docker Hub están disponibles. También es posible especificar otros repositorios con *repository-url/image:tag*. El nombre puede tener una longitud máxima de 225 caracteres. Puede contener letras mayúsculas y minúsculas, números, guiones medios (-), guiones bajos (_), dos puntos (:), barras inclinadas (/) y signos numéricos (#). Este parámetro se asigna a Image en la sección [Creación de un contenedor](#) de la [API remota de Docker](#) y el parámetro IMAGE de [docker run](#).

 Note

La arquitectura de la imagen de Docker debe coincidir con la arquitectura del procesador de los recursos informáticos en las que estén programadas. Por ejemplo, las imágenes de Docker basadas en ARM solo se pueden ejecutar en recursos informáticos basados en ARM.

- Las imágenes de los repositorios públicos de Amazon ECR utilizan las convenciones de nomenclatura completa `registry/repository[:tag]` o `registry/repository[@digest]` (por ejemplo, `public.ecr.aws/registry_alias/my-web-app:latest`).

- Las imágenes de los repositorios de Amazon ECR utilizan la convención de nomenclatura completa `registry/repository[:tag]`. Por ejemplo, `aws_account_id.dkr.ecr.region.amazonaws.com/my-web-app:latest`
 - Las imágenes de los repositorios oficiales de Docker Hub utilizan un solo nombre (por ejemplo, `ubuntu` o `mongo`).
 - Las imágenes de otros repositorios de Docker Hub se clasifican con un nombre de organización (por ejemplo, `amazon/amazon-ecs-agent`).
 - Las imágenes de otros repositorios online se cualifican más con un nombre de dominio (por ejemplo, `quay.io/assemblyline/ubuntu`).
- d. En Comando, ingrese los comandos en el campo como su equivalente de matriz de cadenas JSON.

Este parámetro se asigna a `Cmd` en la sección [Creación de un contenedor](#) de la [API remota de Docker](#) y el parámetro `COMMAND` se corresponde con `docker run`. [Para obtener más información sobre el DockerCMD parámetro, consulta https://docs.docker.com/engine/reference/builder/#cmd.](#)

 Note

También puede usar valores predeterminados para la sustitución de parámetros y marcadores de posición en el comando. Para obtener más información, consulte [Parameters](#).

- e. Para `v` CPUs, especifique el número de `v` que desea reservar CPUs para el contenedor. Este parámetro se corresponde con `CpuShares` en la sección [Creación de un contenedor](#) de la [API remota de Docker](#) y con la opción `--cpu-shares` de `docker run`. Cada vCPU es equivalente a 1 024 cuotas de CPU. Debe especificar al menos una vCPU.
- f. En Memoria, especifique límite máximo (en MiB) de memoria que quiera presentarle al contenedor del trabajo. Si su contenedor intenta superar la memoria especificada aquí, el contenedor se detiene. Este parámetro se corresponde con `Memory` en la sección [Creación de un contenedor](#) de la [API remota de Docker](#) y con la opción `--memory` de `docker run`. Debe especificar al menos 4 MiB de memoria para un trabajo.

 Note

Para maximizar la utilización de los recursos, puede proporcionar a los trabajos la mayor cantidad de memoria posible para un tipo de instancia determinado. Para obtener más información, consulte [Administración de la memoria de recurso de computación](#).

- g. (Opcional) En Número de GPUs, especifique el número de usos de GPUs su trabajo. El trabajo se ejecuta en un contenedor y el número especificado de ellos GPUs están anclados a ese contenedor.
- h. (Opcional) Para el rol Job, puede especificar un rol de IAM que proporcione al contenedor de su trabajo permisos para usar el AWS APIs. Esta característica utiliza roles de IAM de Amazon ECS para otorgarle funcionalidad a la tarea. Para obtener más información, incluidos los requisitos previos de configuración, consulte [Roles de IAM para tareas](#) en la Guía para desarrolladores de Amazon Elastic Container Service.

 Note

Para trabajos que se ejecutan en recursos de Fargate, se requiere un puesto de trabajo.

 Note

Aquí solo se muestran roles con la relación de confianza del Rol de tarea de servicio de Amazon Elastic Container. Para obtener más información sobre cómo crear un rol de IAM para sus AWS Batch trabajos, consulte [Creación de un rol y una política de IAM para sus tareas en la Guía para desarrolladores de Amazon Elastic Container Service](#).

- i. (Opcional) En la función de ejecución, especifique una función de IAM que conceda permiso a los agentes de contenedores de Amazon ECS para realizar llamadas a la AWS API en su nombre. Esta característica utiliza roles de IAM de Amazon ECS para otorgarle funcionalidad a la tarea. Para obtener más información, consulte [Roles de IAM de ejecución de tareas de Amazon ECS](#) en la Guía para desarrolladores de Amazon Elastic Container Service.

19. (Opcional) Expandir Configuración adicional:

- a. En el caso de Variables de entorno, seleccione Agregar variable de entorno para añadir variables de entorno como pares de nombre-valor. Estas variables se transfieren al contenedor.
- b. Para la configuración del rol de trabajo, puede especificar un rol de IAM que proporcione al contenedor de su trabajo permisos para usar el AWS APIs. Esta característica utiliza roles de IAM de Amazon ECS para otorgarle funcionalidad a la tarea. Para obtener más información, incluidos los requisitos previos de configuración, consulte [Roles de IAM para tareas](#) en la Guía para desarrolladores de Amazon Elastic Container Service.

Note

Para trabajos que se ejecutan en recursos de Fargate, se requiere un puesto de trabajo.

Note

Aquí solo se muestran roles con la relación de confianza del Rol de tarea de servicio de Amazon Elastic Container. Para obtener más información sobre cómo crear un rol de IAM para trabajos de AWS Batch , consulte [Creación de un rol de IAM y una política para sus tareas](#) en la Guía para desarrolladores de Amazon Elastic Container Service.

- c. En Función de ejecución, especifique una función de IAM que conceda permiso a los agentes de contenedor de Amazon ECS para realizar llamadas a la AWS API en su nombre. Esta característica utiliza roles de IAM de Amazon ECS para otorgarle funcionalidad a la tarea. Para obtener más información, consulte [Roles de IAM de ejecución de tareas de Amazon ECS](#) en la Guía para desarrolladores de Amazon Elastic Container Service.

20. En la sección Configuración de seguridad:

- a. (Opcional) Para concederle privilegios elevados al contenedor del trabajo en la instancia del host (similares a los de un usuario root), active Privilegiado. Este parámetro se corresponde con Privileged en la sección [Creación de un contenedor](#) de la [API remota de Docker](#) y con la opción `--privileged` de [docker run](#).

- b. (Opcional) En Usuario, introduzca el nombre de usuario a utilizar dentro del contenedor. Este parámetro se corresponde con `User` en la sección [Creación de un contenedor](#) de la [API remota de Docker](#) y con la opción `--user` de [docker run](#).
- c. (Opcional) En Secretos, seleccione Agregar secreto para añadir los secretos como pares de nombre-valor. Estos secretos están expuestos en el contenedor. [Para obtener más información, consulte LogConfiguration: Opciones secretas.](#)

21. En la sección de Configuración de Linux:

- a. Active la opción Habilitar el sistema de archivos de solo lectura para eliminar el acceso de escritura al volumen.
- b. (Opcional) Active la opción Habilitar el proceso `init` para ejecutar un proceso `init` dentro del contenedor. Este proceso reenvía señales y recoge procesos.
- c. En Tamaño de memoria compartida, introduzca el tamaño (en MiB) del `/dev/shm` volumen de .
- d. En Tamaño de intercambio máximo, introduzca la cantidad total de memoria de intercambio (en MiB) que puede utilizar el contenedor.
- e. En Intercambio, introduzca un valor entre 0 y 100 para indicar el comportamiento de intercambio del contenedor. Si no especifica un valor y el intercambio está habilitado, el valor predeterminado es 60. [Para obtener más información, consulte:swappiness. LinuxParameters](#)
- f. (Opcional) En el caso de los Dispositivos, seleccione Agregar dispositivo para añadir un dispositivo:
 - i. En Container path (Ruta del contenedor), especifique la ruta de la instancia del contenedor que va a exponer el dispositivo asignado a la instancia del host. Si lo deja en blanco, se utiliza la ruta del host en el contenedor.
 - ii. En Host path (Ruta de host), especifique la ruta de un dispositivo de la instancia del host.
 - iii. En la página Permisos, haga clic en uno o varios permisos para aplicarlos al dispositivo. Los permisos disponibles son `READ`, `WRITE` y `MKNOD`.

22. (Opcional) En Puntos de montaje, elija la configuración Agregar puntos de montaje para agregar puntos de montaje a los volúmenes de datos. Debe especificar el volumen de origen y la ruta del contenedor. Estos puntos de montaje se transfieren al daemon Docker de una instancia de contenedor. También puede elegir que el volumen sea de Solo lectura.

23. (Opcional) En Configuración de Ulimits, seleccione Agregar ulimit para agregar un `ulimits` valor al contenedor. Introduzca los valores de Nombre, Límite flexible y Límite invariable y, a continuación, elija Agregar límite máximo.
 24. (Opcional) En Configuración de volúmenes, seleccione Agregar volumen para crear una lista de volúmenes que se transferirán al contenedor. Introduzca el Nombre y la Ruta de origen del volumen y, a continuación, seleccione Agregar volumen. También puede optar por activar Habilitar EFS.
 25. (Opcional) En el caso de Tmpfs, seleccione Agregar tmpfs para añadir una montura tmpfs.
 26. En la sección de Propiedades de tareas:
 - a. En Función de ejecución (condicional), elija una función que permita a los agentes de Amazon ECS realizar llamadas a la AWS API en su nombre. Para obtener más información sobre la creación de un Rol de ejecución, consulte [Tutorial: creación del rol de ejecución de IAM](#).
 - b.**Important**

Para utilizar el comando de ejecución de ECS, su entorno de computación debe cumplir con las [consideraciones del entorno de computación para los trabajos paralelos de varios nodos](#).
- Elija Habilitar el comando de ejecución ECS para permitir el acceso directo al shell del contenedor de Amazon ECS y omitir el sistema operativo del host. Debe elegir un Rol de tarea.
-
- Important**
- El comando Ejecución de ECS requiere que el sistema de archivos sea modificable.
- c. Para el rol de tarea, elija un rol de Amazon ECS Identity and Access Management (IAM) para permitir que el contenedor realice llamadas a la AWS API en su nombre. Para obtener más información, consulte [Rol de IAM de las tareas de Amazon ECS](#) en la Guía para desarrolladores de Amazon Elastic Container Service.
27. (Opcional) En la sección de Configuración de registro:

- a. En el Controlador de registro, elija el controlador de registro que desee utilizar. [Para obtener más información sobre los controladores de registro disponibles, consulte LogConfiguration: LogDriver.](#)

 Note

Se utiliza el controlador de registro `awslogs` de forma predeterminada.

- b. En Opciones, elija Agregar opción para agregar una opción. Introduzca un par nombre-valor y, a continuación, elija Agregar opción.
- c. En Secretos, seleccione Agregar secreto. Introduzca un par nombre-valor y, a continuación, seleccione Agregar secreto para añadir un secreto.

 Tip

[Para obtener más información, consulte: Opciones secretas. LogConfiguration](#)

28. Seleccione Página siguiente.

29. Para la Revisión de definición de trabajo, revise los pasos de configuración. Si necesita realizar cambios, elija Editar. Cuando haya terminado, seleccione Creación de definición de trabajo.

Plantilla de definición de trabajo que utiliza ContainerProperties

A continuación, se muestra una plantilla de definición de trabajo vacía que incluye un único contenedor. Puede utilizar esta plantilla para crear la definición del trabajo, que posteriormente se puede guardar en un archivo y utilizarse con la opción de la AWS CLI `--cli-input-json`. Para obtener más información sobre estos parámetros, consulte [JobDefinition](#).

 Note

Puede generar la plantilla de definición de trabajo de un único contenedor con el siguiente comando de AWS CLI:

```
$ aws batch register-job-definition --generate-cli-skeleton
```

```
{
  "jobDefinitionName": "",
  "type": "container",
  "parameters": {
    "KeyName": ""
  },
  "schedulingPriority": 0,
  "containerProperties": {
    "image": "",
    "vcpus": 0,
    "memory": 0,
    "command": [
      ""
    ],
    "jobRoleArn": "",
    "executionRoleArn": "",
    "volumes": [
      {
        "host": {
          "sourcePath": ""
        },
        "name": "",
        "efsVolumeConfiguration": {
          "fileSystemId": "",
          "rootDirectory": "",
          "transitEncryption": "ENABLED",
          "transitEncryptionPort": 0,
          "authorizationConfig": {
            "accessPointId": "",
            "iam": "DISABLED"
          }
        }
      }
    ],
    "environment": [
      {
        "name": "",
        "value": ""
      }
    ],
    "mountPoints": [
      {
        "containerPath": "",
```



```

        "readOnly": true,
        "sourceVolume": ""
    }
],
"readonlyRootFilesystem": true,
"privileged": true,
"ulimits": [
    {
        "hardLimit": 0,
        "name": "",
        "softLimit": 0
    }
],
"user": "",
"instanceType": "",
"resourceRequirements": [
    {
        "value": "",
        "type": "MEMORY"
    }
],
"linuxParameters": {
    "devices": [
        {
            "hostPath": "",
            "containerPath": "",
            "permissions": [
                "WRITE"
            ]
        }
    ],
    "initProcessEnabled": true,
    "sharedMemorySize": 0,
    "tmpfs": [
        {
            "containerPath": "",
            "size": 0,
            "mountOptions": [
                ""
            ]
        }
    ],
    "maxSwap": 0,
    "swappiness": 0
}

```

```

    },
    "logConfiguration": {
      "logDriver": "syslog",
      "options": {
        "KeyName": ""
      },
      "secretOptions": [
        {
          "name": "",
          "valueFrom": ""
        }
      ]
    },
    "secrets": [
      {
        "name": "",
        "valueFrom": ""
      }
    ],
    "networkConfiguration": {
      "assignPublicIp": "DISABLED"
    },
    "fargatePlatformConfiguration": {
      "platformVersion": ""
    }
  },
  "nodeProperties": {
    "numNodes": 0,
    "mainNode": 0,
    "nodeRangeProperties": [
      {
        "targetNodes": "",
        "container": {
          "image": "",
          "vcpus": 0,
          "memory": 0,
          "command": [
            ""
          ],
          "jobRoleArn": "",
          "executionRoleArn": "",
          "volumes": [
            {
              "host": {

```

```

        "sourcePath": ""
    },
    "name": "",
    "efsVolumeConfiguration": {
        "fileSystemId": "",
        "rootDirectory": "",
        "transitEncryption": "DISABLED",
        "transitEncryptionPort": 0,
        "authorizationConfig": {
            "accessPointId": "",
            "iam": "ENABLED"
        }
    }
},
"environment": [
    {
        "name": "",
        "value": ""
    }
],
"mountPoints": [
    {
        "containerPath": "",
        "readOnly": true,
        "sourceVolume": ""
    }
],
"readonlyRootFilesystem": true,
"privileged": true,
"ulimits": [
    {
        "hardLimit": 0,
        "name": "",
        "softLimit": 0
    }
],
"user": "",
"instanceType": "",
"resourceRequirements": [
    {
        "value": "",
        "type": "MEMORY"
    }
]

```

```
    ],
    "linuxParameters": {
      "devices": [
        {
          "hostPath": "",
          "containerPath": "",
          "permissions": [
            "WRITE"
          ]
        }
      ],
      "initProcessEnabled": true,
      "sharedMemorySize": 0,
      "tmpfs": [
        {
          "containerPath": "",
          "size": 0,
          "mountOptions": [
            ""
          ]
        }
      ],
      "maxSwap": 0,
      "swappiness": 0
    },
    "logConfiguration": {
      "logDriver": "awslogs",
      "options": {
        "KeyName": ""
      },
      "secretOptions": [
        {
          "name": "",
          "valueFrom": ""
        }
      ]
    },
    "secrets": [
      {
        "name": "",
        "valueFrom": ""
      }
    ],
    "networkConfiguration": {
```

```

        "assignPublicIp": "DISABLED"
      },
      "fargatePlatformConfiguration": {
        "platformVersion": ""
      }
    }
  ],
  "retryStrategy": {
    "attempts": 0,
    "evaluateOnExit": [
      {
        "onStatusReason": "",
        "onReason": "",
        "onExitCode": "",
        "action": "RETRY"
      }
    ]
  },
  "propagateTags": true,
  "timeout": {
    "attemptDurationSeconds": 0
  },
  "tags": {
    "KeyName": ""
  },
  "platformCapabilities": [
    "EC2"
  ],
  "eksProperties": {
    "podProperties": {
      "serviceAccountName": "",
      "hostNetwork": true,
      "dnsPolicy": "",
      "containers": [
        {
          "name": "",
          "image": "",
          "imagePullPolicy": "",
          "command": [
            ""
          ],
          "args": [

```

```

        ""
    ],
    "env": [
        {
            "name": "",
            "value": ""
        }
    ],
    "resources": {
        "limits": {
            "KeyName": ""
        },
        "requests": {
            "KeyName": ""
        }
    },
    "volumeMounts": [
        {
            "name": "",
            "mountPath": "",
            "readOnly": true
        }
    ],
    "securityContext": {
        "runAsUser": 0,
        "runAsGroup": 0,
        "privileged": true,
        "readOnlyRootFilesystem": true,
        "runAsNonRoot": true
    }
},
"volumes": [
    {
        "name": "",
        "hostPath": {
            "path": ""
        },
        "emptyDir": {
            "medium": "",
            "sizeLimit": ""
        },
        "secret": {
            "secretName": "",

```

```
    "optional": true
  }
}
]
```

Parámetros de definición de trabajo para ContainerProperties

Las definiciones de trabajo que usan [ContainerProperties](#) se dividen en varias partes:

- El nombre de la definición de trabajo
- El tipo de definición de trabajo
- Los valores predeterminados de marcador de sustitución de parámetros
- Las propiedades del contenedor para el trabajo
- Las propiedades de Amazon EKS para la definición del trabajo que son necesarias para los trabajos que se ejecutan en los recursos de Amazon EKS
- Las propiedades de nodo necesarias para un trabajo paralelo de varios nodos
- Las capacidades de la plataforma que son necesarias para los trabajos que se ejecutan en los recursos de Fargate
- Los detalles de propagación de etiquetas predeterminados de la definición del trabajo
- La estrategia de reintento predeterminada para la definición del trabajo
- La prioridad de programación predeterminada para la definición del trabajo
- Las etiquetas predeterminadas para la definición del trabajo
- El tiempo de espera predeterminado de la definición del trabajo

Contenido

- [Nombre de la definición de trabajo](#)
- [Tipo](#)
- [Parameters](#)
- [Propiedades de contenedor](#)
- [Propiedades de Amazon EKS](#)
- [Capacidades de plataforma](#)

- [Propagar etiquetas](#)
- [Propiedades del nodo](#)
- [Estrategia de reintento](#)
- [Prioridad de programación](#)
- [Etiquetas](#)
- [Tiempo de espera](#)

Nombre de la definición de trabajo

`jobDefinitionName`

Al registrar una definición de trabajo, es necesario especificar un nombre. El nombre puede tener una longitud máxima de 128 caracteres. Puede contener letras mayúsculas y minúsculas, números, guiones (-) y guiones bajos (_). A la primera definición de trabajo registrado con ese nombre se le da asigna la revisión 1. A las definiciones de trabajo posteriores registradas con el mismo nombre se les asignan números de revisión en orden ascendente.

Tipo: cadena

Obligatorio: sí

Tipo

`type`

Al registrar una definición de trabajo, es necesario especificar el tipo de trabajo. Si el trabajo se ejecuta en recursos de Fargate, entonces `multinode` no es compatible. Para obtener más información sobre los trabajos paralelos de varios nodos, consulte [Creación de una definición de trabajo paralelo de varios nodos](#).

Tipo: cadena

Valores válidos: `container` | `multinode`

Obligatorio: sí

Parameters

parameters

Al enviar un trabajo, es posible especificar parámetros que sustituyan los marcadores de posición o ignoren los parámetros de definición de trabajo predeterminados. Los parámetros de las solicitudes de envío de trabajo prevalecen sobre los predeterminados de la definición de trabajo. Esto significa que puede usar la misma definición de trabajo para varios trabajos que usen el mismo formato. También puede cambiar mediante programación los valores del comando en el momento del envío.

Tipo: mapa de cadena a cadena

Obligatorio: no

Al registrar una definición de trabajo, es posible utilizar marcadores de posición de sustitución de parámetros en el campo `command` de las propiedades de un contenedor de trabajo. La sintaxis es la siguiente.

```
"command": [  
    "ffmpeg",  
    "-i",  
    "Ref::inputfile",  
    "-c",  
    "Ref::codec",  
    "-o",  
    "Ref::outputfile"  
]
```

En el ejemplo anterior, hay marcadores de posición de sustitución de parámetros `Ref::inputfile`, `Ref::codec` y `Ref::outputfile` en el comando. Puede usar el objeto `parameters` de la definición de trabajo para establecer valores predeterminados para estos marcadores de posición. Por ejemplo, para definir un valor predeterminado para el marcador de posición `Ref::codec`, especifique lo siguiente en la definición del trabajo:

```
"parameters" : {"codec" : "mp4"}
```

Cuando esta definición de trabajo se envíe para su ejecución, el argumento `Ref::codec` del comando para el contenedor se sustituirá por el valor predeterminado, `mp4`.

Propiedades de contenedor

Al registrar una definición de trabajo, especifique una lista de las propiedades de contenedor que se transmiten al daemon de Docker en una instancia de contenedor cuando se coloca el trabajo. Las siguientes propiedades de contenedor se permiten en una definición de trabajo. Para los trabajos de un solo nodo, estas propiedades de contenedor se establecen en el nivel de definición de trabajo. En los trabajos paralelos de varios nodos, las propiedades de contenedor se establecen en el nivel [Propiedades del nodo](#) para cada grupo de nodos.

command

El comando que se transfiere al contenedor. Este parámetro se asigna a Cmd en la sección [Creación de un contenedor](#) de la [API remota de Docker](#) y el parámetro COMMAND se corresponde con [docker run](#). Para obtener más información sobre el parámetro CMD de Docker, consulte <https://docs.docker.com/engine/reference/builder/#cmd>.

```
"command": ["string", ...]
```

Tipo: matriz de cadenas

Obligatorio: no

environment

Las variables de entorno a transferir a un contenedor. Este parámetro se corresponde con Env en la sección [Creación de un contenedor](#) de la [API remota de Docker](#) y con la opción --env de [docker run](#).

Important

No es recomendable usar variables del entorno en texto sin formato para información confidencial, como los datos de la credencial.

Note

Las variables de entorno no pueden empezar por AWS_BATCH. Esta convención de nomenclatura está reservada para las variables que establece el servicio de AWS Batch.

Tipo: matriz de pares clave-valor

Obligatorio: no

name

El nombre de la variable de entorno.

Tipo: cadena

Obligatorio: sí, si se utiliza `environment`.

value

El valor de la variable de entorno.

Tipo: cadena

Obligatorio: sí, si se utiliza `environment`.

```
"environment" : [  
  { "name" : "envName1", "value" : "envValue1" },  
  { "name" : "envName2", "value" : "envValue2" }  
]
```

executionRoleArn

Al registrar una definición de trabajo, es posible especificar un rol de IAM. El rol le concede al agente de contenedor de Amazon ECS permisos para llamar a las acciones de la API que se especifican en sus políticas asociadas, en nombre de quien registra la definición de trabajo. Para los trabajos que se ejecutan en recursos de Fargate, debe proporcionar un rol de ejecución. Para obtener más información, consulte [AWS Batch Función de ejecución de IAM](#).

Tipo: string

Obligatorio: no

fargatePlatformConfiguration

La configuración de la plataforma para trabajos que se ejecutan en recursos de Fargate. Los trabajos que se ejecutan en los recursos de EC2 no deben especificar este parámetro.

Tipo: objeto [FargatePlatformConfiguration](#)

Obligatorio: no

platformVersion

La versión de la plataforma de AWS Fargate utilizada para los trabajos o LATEST para usar una versión reciente y aprobada de la plataforma de AWS Fargate.

Tipo: cadena

Valor predeterminado: LATEST

Obligatorio: no

image

La imagen que se utiliza para iniciar un trabajo. Esta cadena se transfiere directamente al daemon de Docker. Las imágenes del registro de Docker Hub están disponibles de forma predeterminada. También es posible especificar otros repositorios con *repository-url/image:tag*. Se permiten hasta 255 letras (mayúsculas y minúsculas), números, guiones, caracteres de subrayado, comas, puntos, barras diagonales y signos numéricos. Este parámetro se asigna a Image en la sección [Creación de un contenedor](#) de la [API remota de Docker](#) y el parámetro IMAGE de [docker run](#).

 Note

La arquitectura de la imagen de Docker debe coincidir con la arquitectura del procesador de los recursos informáticos en las que estén programadas. Por ejemplo, las imágenes de Docker basadas en ARM solo se pueden ejecutar en recursos informáticos basados en ARM.

- Las imágenes de los repositorios públicos de Amazon ECR utilizan las convenciones de nomenclatura completa `registry/repository[:tag]` o `registry/repository[@digest]` (por ejemplo, `public.ecr.aws/registry_alias/my-web-app:latest`).
- Las imágenes de los repositorios de Amazon ECR utilizan la convención de nomenclatura completa `registry/repository[:tag]`. Por ejemplo, `aws_account_id.dkr.ecr.region.amazonaws.com/my-web-app:latest`.
- Las imágenes de los repositorios oficiales de Docker Hub utilizan un solo nombre (por ejemplo, `ubuntu` o `mongo`).

- Las imágenes de otros repositorios de Docker Hub se identifican con un nombre de organización (por ejemplo, amazon/amazon-ecs-agent).
- Las imágenes de otros repositorios online se cualifican más con un nombre de dominio (por ejemplo, quay.io/assemblyline/ubuntu).

Tipo: cadena

Obligatorio: sí

instanceType

El tipo de instancia que se va a utilizar para un trabajo en paralelo de varios nodos. Todos los grupos de nodos de un trabajo paralelo de varios nodos deben utilizar el mismo tipo de instancia. Este parámetro no es aplicable a trabajos de contenedor de un solo nodo o para trabajos que se ejecutan en recursos de Fargate.

Tipo: cadena

Obligatorio: no

jobRoleArn

Al registrar una definición de trabajo, es posible especificar un rol de IAM. El rol le concede al contenedor de trabajo permisos para llamar a las acciones de la API que se especifican en sus políticas asociadas, en nombre de quien registra la definición de trabajo. Para obtener más información, consulte [Roles de IAM para tareas](#) en la Guía para desarrolladores de Amazon Elastic Container Service.

Tipo: cadena

Obligatorio: no

linuxParameters

Modificaciones específicas de Linux que se aplican al contenedor, como los detalles de asignación de dispositivos.

```
"linuxParameters": {
  "devices": [
    {
      "hostPath": "string",
      "containerPath": "string",
      "permissions": [
        "READ", "WRITE", "MKNOD"
```

```

    ]
  }
],
"initProcessEnabled": true/false,
"sharedMemorySize": 0,
"tmpfs": [
  {
    "containerPath": "string",
    "size": integer,
    "mountOptions": [
      "string"
    ]
  }
],
"maxSwap": integer,
"swappiness": integer
}

```

Tipo: objeto [LinuxParameters](#)

Obligatorio: no

devices

Lista de dispositivos asignados en el contenedor. Este parámetro se corresponde con Devices en la sección [Creación de un contenedor](#) de la [API remota de Docker](#) y la opción --device se corresponde con [docker run](#).

Note

Este parámetro no es aplicable a trabajos que se ejecutan en recursos de Fargate.

Tipo: matriz de objetos [Dispositivo](#)

Obligatorio: no

hostPath

Ruta en la que se encuentra el dispositivo disponible en la instancia de contenedor del host.

Tipo: cadena

Obligatorio: sí

`containerPath`

Ruta en la que queda expuesto el dispositivo en el contenedor. Si no se especifica, el dispositivo se expone en la misma ruta que la ruta del host.

Tipo: cadena

Obligatorio: no

`permissions`

Permisos del dispositivo en el contenedor. Si no se especifica, los permisos se establecen en READ, WRITE y MKNOD.

Tipo: matriz de cadenas

Obligatorio: no

Valores válidos: READ | WRITE | MKNOD

`initProcessEnabled`

Si es "true" (verdadero), ejecute un proceso de `init` dentro del contenedor que reenvíe señales y aproveche procesos. Este parámetro se corresponde con la opción `--init` de [docker run](#). Este parámetro requiere la versión 1.25 de la API remota de Docker o superior en su instancia de contenedor. Para comprobar la versión de la API remota de Docker en su instancia de contenedor, inicie sesión en su instancia de contenedor y ejecute el comando siguiente: `sudo docker version | grep "Server API version"`

Tipo: Booleano

Obligatorio: no

`maxSwap`

La cantidad total de memoria de intercambio (en MiB) que puede utilizar un trabajo. Este parámetro se traduce en la opción `--memory-swap` de [docker run](#) donde el valor es la suma de la memoria del contenedor más el valor de `maxSwap`. Para obtener más información, consulte [Detalles de --memory-swap](#) en la documentación de Docker.

Si se especifica un valor `maxSwap` para 0, el contenedor no utiliza el intercambio. Los valores aceptados son 0 o cualquier entero positivo. Si se omite el parámetro `maxSwap`, el

contenedor utiliza la configuración de intercambio de la instancia de contenedor en la que se está ejecutando. Debe establecerse un valor de `maxSwap` para el parámetro `swappiness`.

 Note

Este parámetro no es aplicable a trabajos que se ejecutan en recursos de Fargate.

Tipo: entero

Obligatorio: no

`sharedMemorySize`

El valor del tamaño (en MiB) del volumen `/dev/shm`. Este parámetro se corresponde con la opción `--shm-size` de [docker run](#).

 Note

Este parámetro no es aplicable a trabajos que se ejecutan en recursos de Fargate.

Tipo: entero

Obligatorio: no

`swappiness`

Puede utilizar esta opción para ajustar el comportamiento de intercambio de memoria de un contenedor. Un valor `swappiness` de 0 impide que ocurra el intercambio a menos que sea absolutamente necesario. Un valor `swappiness` de 100 aumenta al máximo el intercambio de páginas. Los valores aceptados son números enteros comprendidos entre 0 y 100. Si no se especifica el parámetro `swappiness`, se utiliza el valor predeterminado de 60. Si no se especifica ningún valor para `maxSwap`, este parámetro se omite. Si `maxSwap` se establece en 0, el contenedor no utiliza intercambio. Este parámetro se corresponde con la opción `--memory-swappiness` de [docker run](#).

Tenga en cuenta lo siguiente cuando utilice una configuración de intercambio por contenedor.

- El espacio de intercambio debe estar habilitado y asignado a la instancia de contenedor para que los contenedores lo utilicen.

 Note

Las AML optimizadas de Amazon ECS no tienen habilitado el intercambio de forma predeterminada. Debe habilitar el intercambio en la instancia para utilizar esta característica. Para obtener más información, consulte [Volúmenes de intercambio de almacén de instancias](#) en la Guía del usuario de Amazon EC2 o [¿Cómo asigno memoria para que funcione como espacio de intercambio en una instancia de Amazon EC2 mediante un archivo de intercambio?](#).

- Los parámetros de espacio de intercambio solo se admiten para definiciones de trabajos que utilizan recursos de EC2.
- Si en una definición de trabajo se omiten los parámetros `maxSwap` y `swappiness`, cada contenedor tendrá un valor `swappiness` predeterminado de 60. El uso total de intercambio se limita al doble de la reserva de memoria del contenedor.

 Note

Este parámetro no es aplicable a trabajos que se ejecutan en recursos de Fargate.

Tipo: entero

Obligatorio: no

`tmpfs`

La ruta del contenedor, las opciones de montaje y el tamaño del montaje `tmpfs`.

Tipo: matriz de objetos [Tmpfs](#)

 Note

Este parámetro no es aplicable a trabajos que se ejecutan en recursos de Fargate.

Obligatorio: no

`containerPath`

La ruta de archivo absoluta en el contenedor donde se monta el volumen `tmpfs`.

Tipo: cadena

Obligatorio: sí

`mountOptions`

La lista de opciones de montaje del volumen tmpfs.

Valores válidos: "defaults" | "ro" | "rw" | "suid" | "nosuid" | "dev" | "nodev" | "exec" | "noexec" | "sync" | "async" | "dirsync" | "remount" | "mand" | "nomand" | "atime" | "noatime" | "diratime" | "nodiratime" | "bind" | "rbind" | "unbindable" | "runbindable" | "private" | "rprivate" | "shared" | "rshared" | "slave" | "rslave" | "relatime" | "norelatime" | "strictatime" | "nostrictatime" | "mode" | "uid" | "gid" | "nr_inodes" | "nr_blocks" | "mpol"

Tipo: matriz de cadenas

Obligatorio: no

`size`

El tamaño (en MiB) del volumen tmpfs.

Tipo: número entero

Obligatorio: sí

`logConfiguration`

La especificación de configuración de registro para el trabajo.

Este parámetro se corresponde con LogConfig en la sección [Creación de un contenedor](#) de la [API remota de Docker](#) y la opción `--log-driver` se corresponde con [docker run](#). De forma predeterminada, los contenedores usan el mismo controlador de registro que el daemon de Docker. No obstante, el contenedor puede usar un controlador de registro distinto del daemon de Docker especificando un controlador de registro con este parámetro en la definición de contenedor. Para utilizar un controlador de registro distinto para un contenedor, el sistema de registro se debe configurar correctamente en la instancia de contenedor o en un servidor de registro distinto para opciones de registro remotas. Para obtener más información acerca de las opciones para los distintos controladores de registro admitidos, consulte [Configurar controladores de registro](#) en la documentación de Docker.

 Note

AWS Batch admite actualmente un subconjunto de los controladores de registro disponibles para el daemon de Docker (se muestra en el tipo de datos [LogConfiguration](#)).

Este parámetro requiere la versión 1.18 de la API remota de Docker o superior en su instancia de contenedor. Para comprobar la versión de la API remota de Docker en su instancia de contenedor, inicie sesión en su instancia de contenedor y ejecute el comando siguiente: `sudo docker version | grep "Server API version"`

```
"logConfiguration": {
  "devices": [
    {
      "logDriver": "string",
      "options": {
        "optionName1" : "optionValue1",
        "optionName2" : "optionValue2"
      }
      "secretOptions": [
        {
          "name" : "secretOptionName1",
          "valueFrom" : "secretOptionArn1"
        },
        {
          "name" : "secretOptionName2",
          "valueFrom" : "secretOptionArn2"
        }
      ]
    }
  ]
}
```

Tipo: objeto [LogConfiguration](#)

Obligatorio: no

logDriver

El controlador de registro que utilizar para el trabajo. De forma predeterminada, AWS Batch habilita el controlador de registro `awslogs`. Los valores válidos que se enumeran para

este parámetro son controladores de registro con los que el agente del contenedor de Amazon ECS se puede comunicar de forma predeterminada.

Este parámetro se corresponde con LogConfig en la sección [Creación de un contenedor](#) de la [API remota de Docker](#) y la opción `--log-driver` se corresponde con [docker run](#). De forma predeterminada, los trabajos usan el mismo controlador de registro que el daemon de Docker. No obstante, el trabajo puede usar un controlador de registro distinto del daemon de Docker especificando un controlador de registro con este parámetro en la definición de contenedor. Si desea especificar otro controlador de registro para un trabajo, el sistema de registro se debe configurar en la instancia de contenedor en el entorno de computación. O bien, configúrelo en otro servidor de registro para proporcionar opciones de registro remoto. Para obtener más información acerca de las opciones para los distintos controladores de registro admitidos, consulte [Configurar controladores de registro](#) en la documentación de Docker.

 Note

AWS Batch actualmente admite un subconjunto de los controladores de registro disponibles para el daemon de Docker. Es posible que haya controladores de registro adicionales en versiones futuras del agente de contenedor de Amazon ECS.

Los controladores de registro admitidos son `awslogs`, `fluentd`, `gelf`, `json-file`, `journald`, `logentries`, `syslog` y `splunk`.

 Note

Los trabajos que se ejecutan en recursos de Fargate están restringidos a los controladores de registro `awslogs` y `splunk`.

Este parámetro requiere la versión 1.18 de la API remota de Docker o superior en su instancia de contenedor. Para comprobar la versión de la API remota de Docker en su instancia de contenedor, inicie sesión en su instancia de contenedor y ejecute el comando siguiente: `sudo docker version | grep "Server API version"`

 Note

El agente de contenedor de Amazon ECS que se ejecuta en una instancia de contenedor debe registrar los controladores de registro disponibles en dicha instancia con la variable de entorno `ECS_AVAILABLE_LOGGING_DRIVERS`. De lo contrario, los contenedores ubicados en esa instancia no pueden usar estas opciones de configuración de registro. Para obtener más información, consulte [Configuración del agente de contenedor de Amazon ECS](#) en la Guía para desarrolladores de Amazon Elastic Container Service.

awslogs

Especifica el controlador de registro de los Registros de Amazon CloudWatch. Para obtener más información, consulte [Uso del controlador de registros awslogs](#) y [Uso del controlador de Registros de Amazon CloudWatch](#) en la documentación de Docker.

fluentd

Especifica el controlador de registro Fluentd. Para obtener más información, incluidos el uso y las opciones, consulte [Controlador de registro Fluentd](#) en la documentación de Docker.

gelf

Especifica el controlador de registro Graylog Extended Format (GELF). Para obtener más información, incluidos el uso y las opciones, consulte [Controlador de registro Graylog Extended Format](#) en la documentación de Docker.

journald

Especifica el controlador de registro journald. Para obtener más información, incluidos el uso y las opciones, consulte [Controlador de registro Journald](#) en la documentación de Docker.

json-file

Especifica el controlador de registro de archivos JSON. Para obtener más información, incluidos el uso y las opciones, consulte [Controlador de registro de archivos JSON](#) en la documentación de Docker.

splunk

Especifica el controlador de registro Splunk. Para obtener más información, incluidos el uso y las opciones, consulte [Controlador de registro Splunk](#) en la documentación de Docker.

syslog

Especifica el controlador de registro syslog. Para obtener más información, incluidos el uso y las opciones, consulte [Controlador de registro Syslog](#) en la documentación de Docker.

Tipo: cadena

Obligatorio: sí

Valores válidos: awslogs | fluentd | gelf | journald | json-file | splunk | syslog

Note

Si tiene un controlador personalizado que no se ha mostrado anteriormente y que desea utilizar con el agente de contenedor de Amazon ECS, puede bifurcar el proyecto del agente de contenedor de Amazon ECS [disponible en GitHub](#) y personalizarlo para que funcione con dicho controlador. Le recomendamos enviar solicitudes de inserción para los cambios que desea que incluyamos. No obstante, Amazon Web Services actualmente no permite solicitudes que ejecutan copias modificadas de este software.

options

Opciones de configuración de registro que enviar a un controlador de registro para el trabajo.

Este parámetro requiere la versión 1.19 de la API remota de Docker o superior en su instancia de contenedor.

Tipo: mapa de cadena a cadena

Obligatorio: no

secretOptions

Un objeto que representa el secreto que transferir a la configuración de registro. Para obtener más información, consulte [Especificación de información confidencial](#).

Tipo: matriz de objetos

Obligatorio: no

name

El nombre de la opción del controlador de registro que se va a configurar en el trabajo.

Tipo: cadena

Obligatorio: sí

valueFrom

El nombre de recurso de Amazon (ARN) del secreto que se va a exponer en la configuración de registro del contenedor. Los valores admitidos son el ARN completo del secreto del Secrets Manager o el ARN completo del parámetro en el almacén de parámetros SSM.

 Note

Si el parámetro del almacén de parámetros SSM está en la misma Región de AWS que la tarea que se va a lanzar, se puede utilizar el ARN completo o el nombre del parámetro. Si el parámetro existe en una región distinta, el ARN completo debe especificarse.

Tipo: cadena

Obligatorio: sí

memory

Este parámetro ha quedado obsoleto, utilice [resourceRequirements](#) en su lugar.

El número de MiB de memoria que reservar para el trabajo.

Como ejemplo de cómo usar [resourceRequirements](#), si la definición de su trabajo contiene una sintaxis similar a la siguiente.

```
"containerProperties": {  
  "memory": 512
```

```
}
```

La sintaxis equivalente a través de [resourceRequirements](#) es de la siguiente manera.

```
"containerProperties": {  
  "resourceRequirements": [  
    {  
      "type": "MEMORY",  
      "value": "512"  
    }  
  ]  
}
```

Tipo: número entero

Obligatorio: sí

mountPoints

Los puntos de montaje para los volúmenes de datos del contenedor. Este parámetro se corresponde con Volumes en la sección [Creación de un contenedor](#) de la [API remota de Docker](#) y con la opción `--volume` de [docker run](#).

```
"mountPoints": [  
  {  
    "sourceVolume": "string",  
    "containerPath": "string",  
    "readOnly": true/false  
  }  
]
```

Tipo: matriz de objetos

Obligatorio: no

sourceVolume

El nombre del volumen a montar.

Tipo: cadena

Obligatorio: sí, si se utiliza mountPoints.

containerPath

La ruta en el contenedor en la que montar el volumen de host.

Tipo: cadena

Obligatorio: sí, si se utiliza mountPoints.

readOnly

Si este valor es true, el acceso del contenedor al volumen es de solo lectura. Si este valor es false, el contenedor puede escribir en el volumen.

Tipo: Booleano

Obligatorio: no

Valor predeterminado: False

networkConfiguration

La configuración de red para los trabajos que se ejecutan en recursos de Fargate. Los trabajos que se ejecutan en los recursos de EC2 no deben especificar este parámetro.

```
"networkConfiguration": {  
  "assignPublicIp": "string"  
}
```

Tipo: matriz de objetos

Obligatorio: no

assignPublicIp

Indica si el trabajo tiene una dirección IP pública. Esto es obligatorio si el trabajo requiere acceso a la red saliente.

Tipo: cadena

Valores válidos: ENABLED | DISABLED

Obligatorio: no

Valor predeterminado: DISABLED

privileged

Cuando este parámetro es verdadero, al contenedor se le conceden permisos elevados en la instancia de contenedor de host, similares a los de un usuario root. Este parámetro se corresponde con Privileged en la sección [Creación de un contenedor](#) de la [API remota de Docker](#) y con la opción `--privileged` de [docker run](#). Este parámetro no es aplicable a trabajos que se ejecutan en recursos de Fargate. No lo proporcione ni lo especifique como falso.

```
"privileged": true/false
```

Tipo: Booleano

Obligatorio: no

readonlyRootFilesystem

Cuando este parámetro es verdadero, al contenedor se le concede acceso de solo lectura a su sistema de archivos raíz. Este parámetro se corresponde con ReadonlyRootfs en la sección [Creación de un contenedor](#) de la [API remota de Docker](#) y con la opción `--read-only` de [docker run](#).

```
"readonlyRootFilesystem": true/false
```

Tipo: Booleano

Obligatorio: no

resourceRequirements

El tipo y la cantidad de un recurso para asignar a un contenedor. Los recursos admitidos incluyen GPU, MEMORY, y VCPU.

```
"resourceRequirements" : [  
  {  
    "type": "GPU",  
    "value": "number"  
  }  
]
```

Tipo: matriz de objetos

Obligatorio: no

type

El tipo de recurso para asignar a un contenedor. Los recursos admitidos incluyen GPU, MEMORY, y VCPU.

Tipo: cadena

Obligatorio: sí, si se utiliza `resourceRequirements`.

value

La cantidad del recurso especificado que se va a reservar para el contenedor. Los valores varían en función del type especificado.

type="GPU"

El número de unidades GPU físicas que reservar para el contenedor. El número de unidades GPU reservadas para todos los contenedores de una tarea no puede superar el número de GPU disponibles en el recurso de computación en el que se lanza la tarea.

type="MEMORY"

El límite máximo (en MiB) de memoria a presentar al contenedor. Si su contenedor intenta superar la memoria especificada aquí, el contenedor se cancela. Este parámetro se corresponde con `Memory` en la sección [Create a container](#) (Creación de un contenedor) de la [API remota de Docker](#) y la opción `--memory` se corresponde con [docker run](#). Debe especificar al menos 4 MiB de memoria para un trabajo. Es obligatorio, pero se puede especificar en varios lugares para trabajos paralelos de varios nodos (MNP). Debe especificarse para cada nodo al menos una vez. Este parámetro se corresponde con `Memory` en la sección [Create a container](#) (Creación de un contenedor) de la [API remota de Docker](#) y la opción `--memory` se corresponde con [docker run](#).

 Note

Si está intentando maximizar la utilización de los recursos proporcionando a las tareas la mayor cantidad de memoria posible para un tipo de instancia determinado, consulte [Administración de la memoria de recurso de computación](#).

Para los trabajos que se ejecutan en recursos de Fargate, `value` debe coincidir con uno de los valores admitidos. Además, el valor de VCPU debe ser uno de los valores compatibles con el valor de memoria correspondiente.

VCPU	MEMORY
0,25 vCPU	512, 1024 y 2048 MiB
0,5 vCPU	1024-4096 MiB en incrementos de 1024 MiB
1 vCPU	2048-8192 MiB en incrementos de 1024 MiB
2 vCPU	4096-16 384 MiB en incrementos de 1024 MiB
4 vCPU	8192-30 720 MiB en incrementos de 1024 MiB
8 vCPU	16 384-61 440 MiB en incrementos de 4096 MiB
16 vCPU	32 768-122 880 MiB en incrementos de 8192 MiB

`type="VCPU"`

El número de vCPU reservado para el trabajo. Este parámetro se corresponde con `CpuShares` en la sección [Create a container](#) (Creación de un contenedor) de la [API remota de Docker](#) y la opción `--cpu-shares` se corresponde con [docker run](#). Cada vCPU es equivalente a 1 024 cuotas de CPU. Para los trabajos que se ejecutan en recursos EC2, debe especificar al menos una vCPU. Es obligatorio, pero se puede especificar en varias ubicaciones. Debe especificarse para cada nodo al menos una vez.

Para los trabajos que se ejecutan en recursos de Fargate, `value` debe coincidir con uno de los valores admitidos, y los valores de MEMORY deben ser uno de los valores admitidos para ese valor de vCPU. Los valores admitidos son 0.25, 0.5, 1, 2, 4, 8 y 16.

El valor predeterminado para la cuota de recuento de recursos de vCPU bajo demanda de Fargate es de 6 vCPU. Para más información sobre las cuotas de Fargate, consulte [AWS Cuotas de Fargate](#) en Referencia general de Amazon Web Services.

Tipo: cadena

Obligatorio: sí, si se utiliza `resourceRequirements`.

secrets

Los secretos del trabajo que se exponen como variables de entorno. Para obtener más información, consulte [Especificación de información confidencial](#).

```
"secrets": [  
  {  
    "name": "secretName1",  
    "valueFrom": "secretArn1"  
  },  
  {  
    "name": "secretName2",  
    "valueFrom": "secretArn2"  
  }  
  ...  
]
```

Tipo: matriz de objetos

Obligatorio: no

name

El nombre de la variable de entorno que contiene el secreto.

Tipo: cadena

Obligatorio: sí, si se utiliza secrets.

valueFrom

El secreto para exponer en el contenedor. Los valores admitidos son el nombre de recurso de Amazon (ARN) completo del secreto del Secrets Manager o el ARN completo del parámetro en el almacén de parámetros SSM.

Note

Si el parámetro del almacén de parámetros SSM está en la misma Región de AWS que el trabajo que se va a lanzar, se puede utilizar el ARN completo o el nombre del parámetro. Si el parámetro existe en una región distinta, el ARN completo debe especificarse.

Tipo: cadena

Obligatorio: sí, si se utiliza secrets.

ulimits

Una lista de valores para ulimits a definir en el contenedor. Este parámetro se corresponde con Ulimits en la sección [Creación de un contenedor](#) de la [API remota de Docker](#) y con la opción `--ulimit` de [docker run](#).

```
"ulimits": [  
  {  
    "name": string,  
    "softLimit": integer,  
    "hardLimit": integer  
  }  
  ...  
]
```

Tipo: matriz de objetos

Obligatorio: no

name

El valor type de ulimit.

Tipo: cadena

Obligatorio: sí, si se utiliza ulimits.

hardLimit

El límite máximo para el tipo de ulimit.

Tipo: número entero

Obligatorio: sí, si se utiliza ulimits.

softLimit

El límite flexible para el tipo de ulimit.

Tipo: número entero

Obligatorio: sí, si se utiliza ulimits.

user

El nombre de usuario que utilizar dentro del contenedor. Este parámetro se corresponde con User en la sección [Creación de un contenedor](#) de la [API remota de Docker](#) y con la opción --user de [docker run](#).

```
"user": "string"
```

Tipo: cadena

Obligatorio: no

vcpus

Este parámetro ha quedado obsoleto, utilice [resourceRequirements](#) en su lugar.

El número de vCPU reservado para el contenedor.

Como ejemplo sobre el uso de resourceRequirements, si la definición de su trabajo contiene líneas similares a las siguientes:

```
"containerProperties": {  
  "vcpus": 2  
}
```

Las líneas equivalentes que utilizan [resourceRequirements](#) son las siguientes.

```
"containerProperties": {  
  "resourceRequirements": [  
    {  
      "type": "VCPU",  
      "value": "2"  
    }  
  ]  
}
```

Tipo: número entero

Obligatorio: sí

volumes

Al registrar una definición de trabajos, puede especificar una lista de los volúmenes que se pasan al daemon de Docker en una instancia de contenedor. Los siguientes parámetros están permitidos en las propiedades de contenedor:

```
"volumes": [  
  {  
    "name": "string",  
    "host": {  
      "sourcePath": "string"  
    },  
    "efsVolumeConfiguration": {  
      "authorizationConfig": {  
        "accessPointId": "string",  
        "iam": "string"  
      },  
      "fileSystemId": "string",  
      "rootDirectory": "string",  
      "transitEncryption": "string",  
      "transitEncryptionPort": number  
    }  
  }  
]
```

name

El nombre del volumen. Se admiten hasta 255 letras (mayúsculas y minúsculas), números, guiones y caracteres de subrayado. Se hace referencia a este nombre en el parámetro `sourceVolume` de la definición de contenedor `mountPoints`.

Tipo: cadena

Obligatorio: no

host

El contenido del parámetro `host` determina si el volumen de datos persiste en la instancia de contenedor del `host` y dónde se almacena. Si el parámetro `host` está vacío, el daemon de Docker le asigna una ruta de `host` al volumen de datos. Sin embargo, no se garantiza que los datos persistan después de que el contenedor asociado deje de funcionar.

 Note

Este parámetro no es aplicable a trabajos que se ejecutan en recursos de Fargate.

Tipo: objeto

Obligatorio: no

`sourcePath`

La ruta de la instancia de contenedor del host que se le presenta al contenedor. Si este parámetro está vacío, el daemon de Docker asigna una ruta de host.

Si el parámetro `host` contiene una ubicación de archivos `sourcePath`, el volumen de datos persiste en la ubicación especificada en la instancia de contenedor del host hasta que la elimine manualmente. Si el valor `sourcePath` no existe en la instancia de contenedor del host, el daemon de Docker la crea. Si la ubicación existe, el contenido de la carpeta de la ruta de origen se exporta.

Tipo: cadena

Requerido: no

`efsVolumeConfiguration`

Este parámetro se especifica cuando se utiliza un sistema de archivos de Amazon Elastic File System para el almacenamiento de tareas. Para obtener más información, consulte [Volúmenes de Amazon EFS](#).

Tipo: objeto

Obligatorio: no

`authorizationConfig`

Los detalles de configuración de autorización en el sistema de archivos de Amazon EFS.

Tipo: cadena

Requerido: no

accessPointId

El ID del punto de acceso de Amazon EFS que se va a utilizar. Si se especifica un punto de acceso, el valor del directorio raíz que se especifica en `EFSVolumeConfiguration` debe omitirse o establecerse en `/`. Esto impone la ruta establecida en el punto de acceso de EFS. Si se utiliza un punto de acceso, el cifrado de tránsito debe estar habilitado en el `EFSVolumeConfiguration`. Para obtener más información, consulte [Trabajo con puntos de acceso de Amazon EFS](#) en la Guía del usuario de Amazon Elastic File System.

Tipo: cadena

Requerido: no

iam

Determina si se utiliza o no el rol de IAM de trabajos de AWS Batch establecido en una definición de trabajo al montar el sistema de archivos de Amazon EFS. Si está habilitado, el cifrado de tránsito debe estar habilitado en el `EFSVolumeConfiguration`. Si se omite este parámetro, se usa el valor predeterminado de `DISABLED`. Para obtener más información, consulte [Uso de puntos de acceso de Amazon EFS](#).

Tipo: cadena

Valores válidos: `ENABLED` | `DISABLED`

Obligatorio: no

fileSystemId

El ID del sistema de archivos de Amazon EFS que se va a usar.

Tipo: cadena

Obligatorio: no

rootDirectory

Directorio del sistema de archivos de Amazon EFS que se va a montar como directorio raíz dentro del host. Si se omite este parámetro, se utiliza la raíz del volumen de Amazon EFS. Si especifica `/`, se obtiene el mismo efecto que si se omite este parámetro. La longitud máxima es de 4096 caracteres.

 Important

Si se especifica un punto de acceso de EFS en el `authorizationConfig`, se debe omitir el parámetro del directorio raíz o establecerlo en `/`. Esto impone la ruta establecida en el punto de acceso de Amazon EFS.

Tipo: cadena

Obligatorio: no

`transitEncryption`

Determina si se habilita el cifrado de los datos en tránsito de Amazon EFS entre el host de Amazon ECS y el servidor de Amazon EFS. El cifrado en tránsito debe estar habilitado si se utiliza la autorización de IAM en Amazon EFS. Si se omite este parámetro, se usa el valor predeterminado de `DISABLED`. Para obtener más información, consulte [Cifrado de datos en tránsito](#) en la Guía del usuario de Amazon Elastic File System.

Tipo: cadena

Valores válidos: `ENABLED` | `DISABLED`

Obligatorio: no

`transitEncryptionPort`

El puerto que se utilizará al enviar datos cifrados entre el host de Amazon ECS y el servidor de Amazon EFS. Si no se especifica un puerto de cifrado en tránsito, se emplea la estrategia de selección de puertos que utiliza el ayudante de montaje de Amazon EFS. Este valor debe estar entre 0 y 65 535. Para obtener más información, consulte [Ayudante de montaje de EFS](#) en la Guía del usuario de Amazon Elastic File System.

Tipo: entero

Obligatorio: no

Propiedades de Amazon EKS

Un objeto con diversas propiedades específicas para los trabajos basados en Amazon EKS. No se puede especificar para definiciones de trabajo basadas en Amazon ECS.

podProperties

Las propiedades de los recursos del pod de Kubernetes de un trabajo.

Tipo: objeto [EksPodProperties](#)

Obligatorio: no

containers

Las propiedades del contenedor que se utiliza en el pod de Amazon EKS.

Tipo: objeto [EksContainer](#)

Obligatorio: no

args

Una matriz de argumentos para el punto de entrada. Si no se especifica, se utiliza el CMD de la imagen de contenedor. Esto corresponde al miembro args en la parte [Punto de entrada](#) del [Pod](#) en Kubernetes. Las referencias de variables de entorno se expanden mediante el entorno del contenedor.

Si la variable de entorno a la que se hace referencia no existe, la referencia en el comando no cambia. Por ejemplo, si la referencia es “\$(NAME1)” y la variable de entorno NAME1 no existe, la cadena de comando seguirá siendo “\$(NAME1)”. \$\$ se reemplaza por \$, y la cadena resultante no se expande. Por ejemplo, \$\$ (VAR_NAME) se pasa como \$ (VAR_NAME), ya sea que existe o no la variable de entorno VAR_NAME. Para obtener más información, consulte [CMD](#) en la Referencia de Dockerfile y [Definición de un comando y los argumentos para un pod](#) en la documentación de Kubernetes.

Tipo: matriz de cadenas

Obligatorio: no

command

El punto de entrada para el contenedor. Esto no se ejecuta dentro de un shell. Si no se especifica, se utiliza el ENTRYPOINT de la imagen de contenedor. Las referencias de variables de entorno se expanden mediante el entorno del contenedor.

Si la variable de entorno a la que se hace referencia no existe, la referencia en el comando no cambia. Por ejemplo, si la referencia es “\$(NAME1)” y la variable de entorno NAME1 no existe, la cadena de comando seguirá siendo “\$(NAME1)”. \$\$ se reemplaza por \$,

y la cadena resultante no se expande. Por ejemplo, `$$ (VAR_NAME)` se pasará como `$(VAR_NAME)`, ya sea que existe o no la variable de entorno `VAR_NAME`. El punto de entrada no se puede actualizar. Para obtener más información, consulte [PUNTO DE ENTRADA](#) en la Referencia de Dockerfile y [Definición de un comando y los argumentos para un contenedor](#) y [Punto de entrada](#) en la documentación de Kubernetes.

Tipo: matriz de cadenas

Obligatorio: no

`env`

Las variables de entorno a transferir a un contenedor.

 Note

Las variables de entorno no pueden empezar por “AWS_BATCH”. Esta convención de nomenclatura está reservada para las variables que establece AWS Batch.

Tipo: matriz de objetos [EksContainerEnvironmentVariable](#)

Obligatorio: no

`name`

El nombre de la variable de entorno.

Tipo: cadena

Obligatorio: sí

`value`

El valor de la variable de entorno.

Tipo: cadena

Obligatorio: no

`image`

La imagen de Docker que se utiliza para iniciar el contenedor.

Tipo: cadena

Obligatorio: sí

`imagePullPolicy`

La política de extracción de imágenes para el contenedor. Los valores admitidos son `Always`, `IfNotPresent` y `Never`. El valor predeterminado de este parámetro es `IfNotPresent`. Sin embargo, si se especifica la etiqueta `:latest`, el valor predeterminado es `Always`. Para obtener más información, consulte [Actualización de imágenes](#) en la documentación de Kubernetes.

Tipo: cadena

Obligatorio: no

`name`

El nombre del contenedor. Si no se especifica el nombre, el nombre predeterminado es `Default`. Cada contenedor de un pod debe tener un nombre único.

Tipo: cadena

Obligatorio: no

`resources`

El tipo y la cantidad de recursos para asignar a un contenedor. Los recursos admitidos incluyen `memory`, `cpu`, y `nvidia.com/gpu`. Para obtener más información, consulte [Administración de recursos para pods y contenedores](#) en la documentación de Kubernetes.

Tipo: objeto [EksContainerResourceRequirements](#)

Obligatorio: no

`limits`

El tipo y la cantidad de recursos que se reservan para el contenedor. Los valores varían en función del `name` que se especifica. Los recursos se pueden solicitar mediante los `limits` o los objetos `requests`.

`memoria`

El límite máximo de memoria (en MiB) para el contenedor, mediante números enteros, con un sufijo `"Mi"`. Si su contenedor intenta superar la memoria especificada aquí, se cancela el contenedor. Debe especificar al menos 4 MiB de memoria para un trabajo. `memory` se puede especificar en `limits`, `requests` o en

ambos. Si `memory` se especifica en ambos lugares, el valor especificado en `limits` debe ser igual al valor especificado en `requests`.

 Note

Para maximizar el uso de los recursos, proporcione a los trabajos la mayor cantidad de memoria posible para el tipo de instancia específico que se está utilizando. Para aprender a hacerlo, consulte [Administración de la memoria de recurso de computación](#).

`cpu`

La cantidad de CPU reservadas para el contenedor. Los valores deben ser múltiplos pares de 0.25. `cpu` se puede especificar en `limits`, `requests` o en ambos. Si `cpu` se especifica en ambos lugares, el valor especificado en `limits` debe ser tan grande como el valor especificado en `requests`.

`nvidia.com/gpu`

La cantidad de GPU reservadas para el contenedor. Los valores deben ser números enteros. `memory` se puede especificar en `limits`, `requests` o en ambos. Si `memory` se especifica en ambos lugares, el valor especificado en `limits` debe ser igual al valor especificado en `requests`.

Tipo: mapa de cadena a cadena

Limitaciones de longitud de los valores: longitud mínima de 1. La longitud máxima es de 256 caracteres.

Obligatorio: no

`requests`

El tipo y la cantidad de recursos que se solicitan para el contenedor. Los valores varían en función del `name` que se especifica. Los recursos se pueden solicitar mediante los `limits` o los objetos `requests`.

`memory`

El límite máximo de memoria (en MiB) para el contenedor, mediante números enteros, con un sufijo “Mi”. Si su contenedor intenta superar la memoria

especificada aquí, se cancela el contenedor. Debe especificar al menos 4 MiB de memoria para un trabajo. `memory` se puede especificar en `limits`, `requests` o en ambos. Si `memory` se especifica en ambos, el valor especificado en `limits` debe ser igual al valor especificado en `requests`.

 Note

Si está intentando maximizar la utilización de los recursos proporcionando a las tareas la mayor cantidad de memoria posible para un tipo de instancia determinado, consulte [Administración de la memoria de recurso de computación](#).

`cpu`

La cantidad de CPU reservadas para el contenedor. Los valores deben ser múltiplos pares de 0.25. `cpu` se puede especificar en `limits`, `requests` o en ambos. Si `cpu` se especifica en ambos, el valor especificado en `limits` debe ser tan grande como el valor especificado en `requests`.

`nvidia.com/gpu`

La cantidad de GPU reservadas para el contenedor. Los valores deben ser números enteros. `nvidia.com/gpu` se puede especificar en `limits`, `requests` o en ambos. Si `nvidia.com/gpu` se especifica en ambos, el valor especificado en `limits` debe ser igual al valor especificado en `requests`.

Tipo: mapa de cadena a cadena

Limitaciones de longitud de los valores: longitud mínima de 1. La longitud máxima es de 256 caracteres.

Obligatorio: no

`securityContext`

El contexto de seguridad para un trabajo. Para obtener más información, consulte [Configuración de un contexto de seguridad para un pod o un contenedor](#) en la documentación de Kubernetes.

Tipo: objeto [EksContainerSecurityContext](#)

Obligatorio: no
`privileged`

Cuando este parámetro es `true`, al contenedor se le conceden permisos elevados en la instancia de contenedor de host. El nivel de permisos es similar al de los permisos de usuario `root`. El valor predeterminado es `false`. Este parámetro se asigna a la política `privileged` de [Políticas de seguridad de pod privilegiados](#) en la documentación de Kubernetes.

Tipo: Booleano

Obligatorio: no

`readOnlyRootFilesystem`

Cuando este parámetro es `true`, al contenedor se le concede acceso de solo lectura al sistema de archivos raíz. El valor predeterminado es `false`. Este parámetro se asigna a la política `ReadOnlyRootFilesystem` de [Políticas de seguridad de pods de sistemas de archivos y volúmenes](#) en la documentación de Kubernetes.

Tipo: Booleano

Obligatorio: no

`runAsGroup`

Cuando se especifica este parámetro, el contenedor se ejecuta como el ID del grupo especificado (`gid`). Si no se especifica este parámetro, el valor predeterminado es el grupo que se especifica en los metadatos de la imagen. Este parámetro se asigna a las políticas `RunAsGroup` y `MustRunAs` de [Políticas de seguridad de pods de grupos y usuarios](#) en la documentación de Kubernetes.

Tipo: Long

Obligatorio: no

`runAsNonRoot`

Cuando se especifica este parámetro, el contenedor se ejecuta como un usuario con un `uid` distinto de 0. Si no se especifica este parámetro, se aplica la regla mencionada. Este parámetro se asigna a las políticas `RunAsUser` y `MustRunAsNonRoot` de

[Políticas de seguridad de pods de grupos y usuarios](#) en la documentación de Kubernetes.

Tipo: Long

Obligatorio: no

`runAsUser`

Cuando se especifica este parámetro, el contenedor se ejecuta como el ID del usuario especificado (`uid`). Si no se especifica este parámetro, el valor predeterminado es el usuario que se especifica en los metadatos de la imagen. Este parámetro se asigna a las políticas `RunAsUser` y `MustRunAs` de [Políticas de seguridad de pods de grupos y usuarios](#) en la documentación de Kubernetes.

Tipo: Long

Obligatorio: no

`volumeMounts`

El volumen se monta para un contenedor para un trabajo de Amazon EKS. Para obtener más información sobre los volúmenes y los montajes de volúmenes en Kubernetes, consulte [Volúmenes](#) en la documentación de Kubernetes.

Tipo: matriz de objetos [EksContainerVolumeMount](#)

Obligatorio: no

`mountPath`

La ruta en el contenedor donde se monta el volumen.

Tipo: cadena

Obligatorio: no

`name`

El nombre del montaje de volumen. Debe coincidir con el nombre de uno de los volúmenes del pod.

Tipo: cadena

Obligatorio: no

`readOnly`

Si este valor es `true`, el acceso del contenedor al volumen es de solo lectura. De lo contrario, el contenedor puede escribir en el volumen. El valor predeterminado es `false`.

Tipo: Booleano

Obligatorio: no

`dnsPolicy`

La política de DNS del pod. El valor predeterminado es `ClusterFirst`. Si no se especifica el parámetro `hostNetwork`, el valor predeterminado es `ClusterFirstWithHostNet`. `ClusterFirst` indica que cualquier consulta de DNS que no coincida con el sufijo de dominio del clúster configurado se reenvía al servidor de nombres ascendente heredado del nodo. Si no se especificó ningún valor para `dnsPolicy` en la operación de API [RegisterJobDefinition](#), las operaciones de API [DescribeJobDefinitions](#) o [DescribeJobs](#) no devuelven ningún valor para `dnsPolicy`. La configuración de las especificaciones del pod contendrá `ClusterFirst` o `ClusterFirstWithHostNet`, según el valor del parámetro `hostNetwork`. Para obtener más información, consulte [Política de DNS del pod](#) en la documentación de Kubernetes.

Valores válidos: `Default` | `ClusterFirst` | `ClusterFirstWithHostNet`

Tipo: cadena

Obligatorio: no

`hostNetwork`

Indica si el pod utiliza la dirección IP de red de los hosts. El valor predeterminado es `true`. Cuando se establece en `false`, se activa el modelo de red de pods de Kubernetes. La mayoría de las cargas de trabajo de AWS Batch son únicamente de salida y no requieren la sobrecarga de asignación de IP para cada pod para las conexiones entrantes. Para obtener más información, consulte [Espacios de nombres de host](#) y [Redes de pod](#) en la documentación de Kubernetes.

Tipo: Booleano

Obligatorio: no

`serviceAccountName`

El nombre de la cuenta de servicio que se utiliza para ejecutar el pod. Para obtener más información, consulte [Cuentas de servicio de Kubernetes](#) y [Configuración de una cuenta de servicio de Kubernetes para asumir un rol de IAM](#) en la Guía del usuario de Amazon EKS y [Configuración de cuentas de servicio para pods](#) en la documentación de Kubernetes.

Tipo: cadena

Obligatorio: no

`volumes`

Especifica los volúmenes para una definición de trabajo que utiliza recursos de Amazon EKS.

Tipo: matriz de objetos [EksVolume](#)

Obligatorio: no

`emptyDir`

Especifica la configuración de un volumen de Kubernetes `emptyDir`. Un volumen `emptyDir` se crea por primera vez cuando se asigna un pod a un nodo. Existe mientras ese pod se ejecuta en ese nodo. Al principio, el volumen `emptyDir` está vacío. Todos los contenedores del pod pueden leer y escribir los archivos del volumen `emptyDir`. Sin embargo, el volumen `emptyDir` se puede montar en la misma ruta o en rutas diferentes en cada contenedor. Cuando por algún motivo se elimina un pod de un nodo, los datos de `emptyDir` se eliminan de manera permanente. Para obtener más información, consulte [emptyDir](#) en la documentación de Kubernetes.

Tipo: objeto [EksEmptyDir](#)

Obligatorio: no

`medium`

El medio para almacenar el volumen. El valor predeterminado es una cadena vacía, que utiliza el almacenamiento del nodo.

""

(Predeterminado) Utilice el almacenamiento en disco del nodo.

“Memoria”

Utilice el volumen `tmpfs` respaldado por la RAM del nodo. El contenido del volumen se pierde cuando el nodo se reinicia, y cualquier almacenamiento en el volumen cuenta respecto del límite de memoria del contenedor.

Tipo: cadena

Obligatorio: no

`sizeLimit`

El tamaño máximo del volumen. De forma predeterminada, no hay un tamaño máximo definido.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 256 caracteres.

Obligatorio: no

`hostPath`

Especifica la configuración de un volumen de Kubernetes `hostPath`. Un volumen `hostPath` monta un archivo o un directorio existente del sistema de archivos del nodo `host` en el pod. Para obtener más información, consulte [hostPath](#) en la documentación de Kubernetes.

Tipo: objeto `EksHostPath`

Obligatorio: no

`path`

La ruta del archivo o del directorio en el `host` para montar en contenedores en el pod.

Tipo: cadena

Obligatorio: no

`nombre`

El nombre del volumen. El nombre debe estar permitido como nombre de subdominio de DNS. Para obtener más información, consulte [Nombres de subdominio de DNS](#) en la documentación de Kubernetes.

Tipo: cadena

Obligatorio: sí

secreta

Especifica la configuración de un volumen de Kubernetes `secret`. Para obtener más información, consulte [secreto](#) en la documentación de Kubernetes.

Tipo: objeto [EksSecret](#)

Obligatorio: no

opcional

Especifica si se deben definir el secreto o las claves del secreto.

Tipo: Booleano

Obligatorio: no

secretName

El nombre del secreto. El nombre debe estar permitido como nombre de subdominio de DNS. Para obtener más información, consulte [Nombres de subdominio de DNS](#) en la documentación de Kubernetes.

Tipo: cadena

Obligatorio: sí

Capacidades de plataforma

platformCapabilities

Las capacidades de plataforma requeridas por la definición de trabajo. Si no se especifica ningún valor, el valor predeterminado es EC2. Para los trabajos que se ejecutan en recursos de Fargate, se especifica FARGATE.

Note

Si el trabajo se ejecuta en recursos de Amazon EKS, entonces no debe especificar `platformCapabilities`.

Tipo: cadena

Valores válidos: EC2 | FARGATE

Obligatorio: no

Propagar etiquetas

`propagateTags`

Especifica si se van a propagar las etiquetas desde el trabajo o la definición del trabajo a la tarea de Amazon ECS correspondiente. Si no se especifica ningún valor, las etiquetas no se propagan. Las etiquetas solo se pueden propagar a las tareas cuando se crea la tarea. En el caso de etiquetas con el mismo nombre, las etiquetas de trabajo tienen prioridad sobre las etiquetas de definiciones de trabajo. Si el número total de etiquetas combinadas del trabajo y la definición del trabajo es superior a 50, el trabajo se mueve al estado FAILED.

Note

Si el trabajo se ejecuta en recursos de Amazon EKS, entonces no debe especificar `propagateTags`.

Tipo: Booleano

Obligatorio: no

Propiedades del nodo

`nodeProperties`

Al registrar una definición de trabajo paralelo de varios nodos, debe especificar una lista de propiedades de nodos. Estas propiedades de los nodos definen el número de nodos que se van a utilizar en el trabajo, el índice de nodos principales y los diferentes rangos de nodos que se van a utilizar. Si el trabajo se ejecuta en recursos de Fargate, no puede especificar `nodeProperties`. En su lugar, utilice `containerProperties`. En una definición de trabajo, se permiten las siguientes propiedades de nodo. Para obtener más información, consulte [Trabajos paralelos de varios nodos](#).

 Note

Si el trabajo se ejecuta en recursos de Amazon EKS, entonces no debe especificar `nodeProperties`.

Tipo: objeto [NodeProperties](#)

Obligatorio: no

`mainNode`

Especifica el índice de nodo del nodo principal de un trabajo paralelo de varios nodos. Este valor de índice de nodo debe ser más pequeño que la cantidad de nodos.

Tipo: número entero

Obligatorio: sí

`numNodes`

La cantidad de nodos que están asociados a un trabajo paralelo de varios nodos.

Tipo: número entero

Obligatorio: sí

`nodeRangeProperties`

Una lista de rangos de nodos y sus propiedades que están asociados a un trabajo paralelo de varios nodos.

 Note

Un grupo de nodos es un conjunto de nodos de trabajo idénticos que comparten las mismas propiedades de contenedor. Puede utilizar AWS Batch para especificar hasta cinco grupos de nodos distintos para cada trabajo.

Tipo: matriz de objetos [NodeRangeProperty](#)

Obligatorio: sí

targetNodes

El rango de nodos, utilizando valores de índice de nodo. Un rango de 0:3 indica nodos con valores de índice de 0 a 3. Si se omite el valor inicial del rango (:n), se utiliza 0 para iniciar el rango. Si se omite el valor final del rango (n:), se utiliza el índice de nodo más alto posible para finalizar el rango. Los rangos de nodos acumulativos deben tener en cuenta todos los nodos (0:n). Puede anidar rangos de nodos, por ejemplo, 0:10 y 4:5. Para este caso, las propiedades del rango 4:5 anulan las propiedades de 0:10.

Tipo: cadena

Obligatorio: no

container

Los detalles del contenedor del rango de nodos. Para obtener más información, consulte [Propiedades de contenedor](#).

Tipo: objeto [ContainerProperties](#)

Obligatorio: no

Estrategia de reintento

retryStrategy

Al registrar una definición de trabajo, se puede especificar una estrategia de reintento para aplicarla a los trabajos fallidos que se envíen con esta definición de trabajo. Cualquier estrategia de reintento que se especifique durante una operación [SubmitJob](#) anula la estrategia de reintento definida aquí. De forma predeterminada, cada trabajo se intenta una vez. Si especifica más de un intento, el trabajo se vuelve a intentar si falla. Algunos ejemplos de intentos fallidos son que el trabajo devuelva un código de salida distinto de cero o que la instancia de contenedor se termine. Para obtener más información, consulte [Reintentos automáticos de trabajo](#).

Tipo: objeto [RetryStrategy](#)

Obligatorio: no

attempts

La cantidad de veces que toma pasar un trabajo al estado `RUNNABLE`. Puede especificar entre 1 y 10 intentos. Si `attempts` es mayor que uno y el envío del trabajo falla, se reintenta tantas veces como se haya especificado hasta que pase a `RUNNABLE`.

```
"attempts": integer
```

Tipo: entero

Obligatorio: no

evaluateOnExit

Matriz de hasta 5 objetos que especifican las condiciones según las que el trabajo se vuelve a intentar o falla. Si se especifica este parámetro, entonces también debe especificarse el parámetro `attempts`. Si `evaluateOnExit` se especifica pero ninguna de las entradas coincide, se vuelve a intentar el trabajo.

```
"evaluateOnExit": [  
  {  
    "action": "string",  
    "onExitCode": "string",  
    "onReason": "string",  
    "onStatusReason": "string"  
  }  
]
```

Tipo: matriz de objetos [EvaluateOnExit](#)

Obligatorio: no

action

Especifica la acción que se debe realizar si se cumplen todas las condiciones especificadas (`onStatusReason`, `onReason` y `onExitCode`). Los valores no distinguen entre mayúsculas y minúsculas.

Tipo: cadena

Obligatorio: sí

Valores válidos: `RETRY` | `EXIT`

onExitCode

Contiene un patrón global para que coincida con la representación decimal del `ExitCode` devuelto para un trabajo. El patrón puede tener 512 caracteres como máximo. Solo puede contener números. No puede contener letras ni caracteres especiales. Opcionalmente, puede terminar con un asterisco (*) para que solo el inicio de la cadena tenga que ser una coincidencia exacta.

Tipo: cadena

Obligatorio: no

onReason

Contiene un patrón global para que coincida con el `Reason` devuelto para un trabajo. El patrón puede tener 512 caracteres como máximo. Puede tener letras, números, puntos (.), dos puntos (:) y espacios en blanco (espacios, pestañas). Opcionalmente, puede terminar con un asterisco (*) para que solo el inicio de la cadena tenga que ser una coincidencia exacta.

Tipo: cadena

Obligatorio: no

onStatusReason

Contiene un patrón global para que coincida con el `StatusReason` devuelto para un trabajo. El patrón puede tener 512 caracteres como máximo. Puede tener letras, números, puntos (.), dos puntos (:) y espacios en blanco (espacios, pestañas). Opcionalmente, puede terminar con un asterisco (*) para que solo el inicio de la cadena tenga que ser una coincidencia exacta.

Tipo: cadena

Obligatorio: no

Prioridad de programación

schedulingPriority

La prioridad de programación de los trabajos que se enviaron con esta definición de trabajo. Esto solo afecta a las colas de trabajo con una política de reparto justo. Los trabajos con una

prioridad de programación más alta se programan antes que los trabajos con una prioridad de programación más baja.

El valor mínimo admitido es 0 y el valor máximo admitido es 9999.

Tipo: entero

Obligatorio: no

Etiquetas

tags

Etiquetas de pares clave-valor para asociarlas a la definición del trabajo. Para obtener más información, consulte [Etiquetar los recursos de AWS Batch](#).

Tipo: mapa de cadena a cadena

Obligatorio: no

Tiempo de espera

timeout

Puede configurar una duración de tiempo de espera para sus trabajos de modo que si un trabajo se ejecuta durante más tiempo, AWS Batch lo finaliza. Para obtener más información, consulte [Tiempos de espera de trabajo](#). Si un trabajo se termina debido a que se ha agotado el tiempo de espera, no se vuelve a intentar. Cualquier configuración de tiempo de espera que se especifique durante una operación [SubmitJob](#) sustituye la configuración de tiempo de espera definida aquí. Para obtener más información, consulte [Tiempos de espera de trabajo](#).

Tipo: objeto [JobTimeout](#)

Obligatorio: no

attemptDurationSeconds

La duración en segundos (medida a partir de la marca de tiempo `startedAt` del intento de trabajo) después de la cual AWS Batch termina los trabajos pendientes. El valor mínimo del tiempo de espera es 60 segundos.

En el caso de los trabajos de matriz, el tiempo de espera se aplica a los trabajos secundarios, no al trabajo de matriz principal.

En el caso de los trabajos paralelos de varios nodos (MNP), el tiempo de espera se aplica a todo el trabajo, no a los nodos individuales.

Tipo: entero

Obligatorio: no

Cree definiciones de trabajo mediante EcsProperties

Al utilizar las definiciones de AWS Batch trabajo [EcsProperties](#), puede modelar el hardware, los sensores, los entornos 3D y otras simulaciones en contenedores separados. Puede utilizar esta característica para organizar de forma lógica los componentes de la carga de trabajo y separarlos de la aplicación principal. Esta función se puede utilizar AWS Batch en Amazon Elastic Container Service (Amazon ECS), Amazon Elastic Kubernetes Service (Amazon EKS) y. AWS Fargate

ContainerProperties frente a las definiciones de trabajo de EcsProperties

Puede utilizar las definiciones de trabajo [ContainerProperties](#) o [EcsProperties](#) según lo requiera su caso de uso. A un nivel superior, ejecutar AWS Batch tareas con EcsProperties es similar a ejecutar tareas con un. ContainerProperties

La estructura de definición de trabajos heredada, que utiliza ContainerProperties, sigue siendo compatible. Si actualmente tiene flujos de trabajo que utilizan esta estructura, puede seguir ejecutándolos.

La principal diferencia es que se ha añadido un objeto nuevo a la definición del trabajo para dar cabida a las definiciones basadas en EcsProperties.

Por ejemplo, una definición de trabajo que usa ContainerProperties en Amazon ECS y Fargate tiene la siguiente estructura:

```
{
  "containerProperties": {
    ...
    "image": "my_ecr_image1",
    ...
  }
}
```

```
},  
...  
}
```

Una definición de trabajo que usa `EcsProperties` en Amazon ECS y Fargate tiene la siguiente estructura:

```
{  
  "ecsProperties": {  
    "taskProperties": [{  
      "containers": [  
        {  
          ...  
          "image": "my_ecr_image1",  
          ...  
        },  
        {  
          ...  
          "image": "my_ecr_image2",  
          ...  
        },  
      ],  
    },  
  ],  
}
```

Cambios generales en el AWS Batch APIs

A continuación, se describen más detalladamente algunas de las principales diferencias entre el uso de `ContainerProperties` y los tipos de datos de la API de `EcsProperties`:

- Muchos de los parámetros que se utilizan en `ContainerProperties` aparecen en `TaskContainerProperties`. Algunos ejemplos incluyen `command`, `image`, `privileged`, `secrets` y `users`. Todos se pueden encontrar en su interior [TaskContainerProperties](#).
- Algunos de los parámetros de `TaskContainerProperties` no tienen equivalentes funcionales en la estructura heredada. Algunos ejemplos incluyen `dependsOn`, `essential`, `name`, `ipcMode` y `pidMode`. Para obtener más información, consulte [EcsTaskDetails](#) y [TaskContainerProperties](#).

Además, algunos parámetros de `ContainerProperties` no tienen equivalentes ni aplicación en la estructura de `EcsProperties`. En [taskProperties](#), `container` se ha sustituido por `containers` para que el nuevo objeto pueda aceptar hasta diez elementos. [Para obtener más información, consulte: ContainerProperties y: containersRegisterJobDefinition. EcsTaskProperties](#)

- La funcionalidad de `taskRoleArn` es equivalente a `jobRoleArn`. [Para obtener más información, consulte: y: EcsTaskProperties taskRoleArn ContainerProperties jobRoleArn](#)

- Puede incluir de uno a diez contenedores en la estructura de `EcsProperties`. [Para obtener más información, consulte: `contenedoresEcsTaskProperties`](#).
- Los objetos `taskProperties` e `instanceTypes` son matrices, pero actualmente solo aceptan un elemento. [Por ejemplo, `:taskProperties` y `:instanceTypesEcsProperties`. `NodeRangeProperty`](#)

Definiciones de trabajo en varios contenedores para Amazon ECS

Para adaptarse a la estructura de varios contenedores de Amazon ECS, algunos de los tipos de datos de API son diferentes. Por ejemplo:

- [`ecsProperties`](#) tiene el mismo nivel que `containerProperties` en la definición de contenedor único. Para obtener más información, consulte [EcsProperties](#) en la Guía de referencia de la API de AWS Batch .
- [`taskProperties`](#) contiene las propiedades definidas para la tarea de Amazon ECS. Para obtener más información, consulte [EcsProperties](#) en la Guía de referencia de la API de AWS Batch .
- [`containers`](#) incluye información similar a la de `containerProperties` en la definición de contenedor único. La principal diferencia es que `containers` permite definir hasta diez contenedores. [Para obtener más información, consulta `Properties:Containers` en la Guía de referencia de la API. `ECSTask` AWS Batch](#)
- El parámetro [`essential`](#) indica cómo afecta el contenedor al trabajo. Todos los contenedores esenciales deben completarse correctamente (salir como 0) para que el trabajo progrese. Si un contenedor marcado como esencial falla (sale con un valor distinto de 0), el trabajo fallará.

El valor predeterminado es `true`, y al menos un contenedor debe estar marcado como `essential`. Para obtener más información, consulte [essential](#) en la Guía de referencia de la API de AWS Batch .

- Con el parámetro [`dependsOn`](#), puede definir una lista de dependencias de contenedores. Para obtener más información, consulte [dependsOn](#) en la Guía de referencia de la API de AWS Batch .

Note

La complejidad de la lista de `dependsOn` y el tiempo de ejecución del contenedor asociado pueden afectar a la hora de inicio del trabajo. Si las dependencias tardan mucho en ejecutarse, el trabajo permanecerá en el estado `STARTING` hasta que se complete.

[Para obtener más información sobre la estructura `ecsProperties` y, consulte la sintaxis de `RegisterJobDefinition` solicitud de `ECSProperties`.](#)

Definiciones de trabajos con varios contenedores para Amazon EKS

Para adaptarse a la estructura de varios contenedores de Amazon EKS, algunos de los tipos de datos de API son diferentes. Por ejemplo:

- [name](#) es un único identificador para el contenedor. Este objeto no es necesario para un único contenedor, pero sí para definir varios contenedores en un pod. Si name no está definido para contenedores individuales, se aplica el nombre predeterminado, `default`.
- [initContainers](#) se definen dentro del tipo de [eksPodProperties](#) datos. Se ejecutan antes que los contenedores de aplicaciones, siempre hasta completarse, y deben completarse correctamente antes de que se inicie el siguiente contenedor.

Estos contenedores están registrados en el agente Amazon EKS Connector y conservan la información de registro en el almacén de datos de backend de Amazon Elastic Kubernetes Service. El objeto `initContainers` puede aceptar hasta diez elementos. Para obtener más información, consulte [Init Containers](#) en la documentación de Kubernetes.

Note

El objeto `initContainers` puede afectar a la hora de inicio del trabajo. Si los `initContainers` tardan mucho en ejecutarse, el trabajo mantendrá el estado `STARTING` hasta que se complete.

- [shareProcessNamespace](#) indica si los contenedores del pod pueden compartir el mismo espacio de nombres del proceso. El valor predeterminado es `false`. Configúrelo en `true` para permitir que los contenedores vean y señalen los procesos de otros contenedores que se encuentran en el mismo pod.
- Todos los contenedores son importantes. Todos los contenedores deben completarse correctamente (salir como 0) para que el trabajo se complete correctamente. Si un contenedor falla (sale con un valor distinto de 0), se produce un error en el trabajo.

Para obtener más información sobre la estructura `eksProperties` y, consulte la sintaxis de [RegisterJobDefinition](#) solicitud de [eksProperties](#).

Referencia: escenarios de trabajo de AWS Batch que utilizan EcsProperties

Para ilustrar cómo se pueden estructurar las definiciones de trabajo de AWS Batch que utilizan `EcsProperties` en función de sus necesidades, en este tema se presentan las siguientes cargas útiles de [RegisterJobDefinition](#). Puede copiar estos ejemplos en un archivo, personalizarlos según sus necesidades y, a continuación, usar AWS Command Line Interface (AWS CLI) para llamar a `RegisterJobDefinition`.

Trabajo de AWS Batch para Amazon ECS en Amazon EC2

A continuación, se muestra el ejemplo de un trabajo de AWS Batch para Amazon Elastic Container Service en Amazon Elastic Compute Cloud:

```
{
  "jobDefinitionName": "multicontainer-ecs-ec2",
  "type": "container",
  "ecsProperties": {
    "taskProperties": [
      {
        "containers": [
          {
            "name": "c1",
            "essential": false,
            "command": [
              "echo",
              "hello world"
            ],
            "image": "public.ecr.aws/amazonlinux/amazonlinux:latest",
            "resourceRequirements": [
              {
                "type": "VCPU",
                "value": "2"
              },
              {
                "type": "MEMORY",
                "value": "4096"
              }
            ]
          },
          {
            "name": "c2",
            "essential": false,
```

```

        "command": [
            "echo",
            "hello world"
        ],
        "image": "public.ecr.aws/amazonlinux/amazonlinux:latest",
        "resourceRequirements": [
            {
                "type": "VCPU",
                "value": "2"
            },
            {
                "type": "MEMORY",
                "value": "4096"
            }
        ]
    },
    {
        "name": "c3",
        "essential": true,
        "command": [
            "echo",
            "hello world"
        ],
        "image": "public.ecr.aws/amazonlinux/amazonlinux:latest",
        "firelensConfiguration": {
            "type": "fluentbit",
            "options": {
                "enable-ecs-log-metadata": "true"
            }
        },
        "resourceRequirements": [
            {
                "type": "VCPU",
                "value": "6"
            },
            {
                "type": "MEMORY",
                "value": "12288"
            }
        ]
    }
]
}
]

```

```
}  
}
```

Trabajo de AWS Batch para Amazon ECS alojado en Fargate

A continuación, se muestra el ejemplo de un trabajo de AWS Batch para Amazon Elastic Container Service en AWS Fargate:

```
{  
  "jobDefinitionName": "multicontainer-ecs-fargate",  
  "type": "container",  
  "platformCapabilities": [  
    "FARGATE"  
  ],  
  "ecsProperties": {  
    "taskProperties": [  
      {  
        "containers": [  
          {  
            "name": "c1",  
            "command": [  
              "echo",  
              "hello world"  
            ],  
            "image": "public.ecr.aws/amazonlinux/amazonlinux:latest",  
            "resourceRequirements": [  
              {  
                "type": "VCPU",  
                "value": "2"  
              },  
              {  
                "type": "MEMORY",  
                "value": "4096"  
              }  
            ]  
          },  
          {  
            "name": "c2",  
            "essential": true,  
            "command": [  
              "echo",  
              "hello world"  
            ],  
            "resourceRequirements": [  
              {  
                "type": "VCPU",  
                "value": "2"  
              },  
              {  
                "type": "MEMORY",  
                "value": "4096"  
              }  
            ]  
          }  
        ]  
      }  
    ]  
  }  
}
```

```

        "image": "public.ecr.aws/amazonlinux/amazonlinux:latest",
        "resourceRequirements": [
            {
                "type": "VCPU",
                "value": "6"
            },
            {
                "type": "MEMORY",
                "value": "12288"
            }
        ]
    },
    "executionRoleArn": "arn:aws:iam::1112223333:role/ecsTaskExecutionRole"
}
]
}
}

```

Trabajo de AWS Batch para Amazon EKS

A continuación, se muestra el ejemplo de un trabajo de AWS Batch para Amazon Elastic Kubernetes Service:

```

{
  "jobDefinitionName": "multicontainer-eks",
  "type": "container",
  "eksProperties": {
    "podProperties": {
      "shareProcessNamespace": true,
      "initContainers": [
        {
          "name": "init-container",
          "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
          "command": [
            "echo"
          ],
          "args": [
            "hello world"
          ],
          "resources": {
            "requests": {
              "cpu": "1",

```

```

        "memory": "512Mi"
      }
    },
    {
      "name": "init-container-2",
      "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
      "command": [
        "echo",
        "my second init container"
      ],
      "resources": {
        "requests": {
          "cpu": "1",
          "memory": "512Mi"
        }
      }
    }
  ],
  "containers": [
    {
      "name": "c1",
      "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
      "command": [
        "echo world"
      ],
      "resources": {
        "requests": {
          "cpu": "1",
          "memory": "512Mi"
        }
      }
    },
    {
      "name": "sleep-container",
      "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
      "command": [
        "sleep",
        "20"
      ],
      "resources": {
        "requests": {
          "cpu": "1",
          "memory": "512Mi"

```

```
}
}
}
}
}
}
```

Trabajo de AWS Batch MNP con varios contenedores por nodo

A continuación, se muestra el ejemplo de un trabajo de AWS Batch paralelo a varios nodos (MNP) con varios contenedores por nodo:

```
{
  "jobDefinitionName": "multicontainer-mnp",
  "type": "multinode",
  "nodeProperties": {
    "numNodes": 6,
    "mainNode": 0,
    "nodeRangeProperties": [
      {
        "targetNodes": "0:5",
        "ecsProperties": {
          "taskProperties": [
            {
              "containers": [
                {
                  "name": "range05-c1",
                  "command": [
                    "echo",
                    "hello world"
                  ],
                  "image": "public.ecr.aws/amazonlinux/amazonlinux:latest",
                  "resourceRequirements": [
                    {
                      "type": "VCPU",
                      "value": "2"
                    },
                    {
                      "type": "MEMORY",
                      "value": "4096"
                    }
                  ]
                }
              ]
            }
          ]
        }
      }
    ]
  }
}
```

```

    },
    {
      "name": "range05-c2",
      "command": [
        "echo",
        "hello world"
      ],
      "image": "public.ecr.aws/amazonlinux/amazonlinux:latest",
      "resourceRequirements": [
        {
          "type": "VCPU",
          "value": "2"
        },
        {
          "type": "MEMORY",
          "value": "4096"
        }
      ]
    }
  ]
}

```

Uso del controlador de registros awslogs

De forma predeterminada, AWS Batch permite que el controlador de `awslogs` registro envíe información de registro a CloudWatch Logs. Puede usar esta característica para ver distintos registros desde sus contenedores en una ubicación cómoda y evitar que los registros de contenedor ocupen espacio en disco en sus instancias de contenedor. Este tema le ayuda a configurar el controlador de registro `awslogs` en sus definiciones de trabajo.

Note

En la AWS Batch consola, puede configurar el controlador de `awslogs` registro en la sección Configuración de registros al crear una definición de trabajo.

 Note

El tipo de información que registran los contenedores del trabajo depende en gran medida del comando `ENTRYPOINT`. De forma predeterminada, los registros que se capturan muestran el resultado del comando que normalmente se ve en una terminal interactiva si se ejecuta el contenedor de forma local, que son las `STDERR` I/O transmisiones `STDOUT` y. El controlador de `awslogs` registros simplemente pasa estos registros de Docker a CloudWatch Logs. Para obtener más información acerca de cómo se procesan los registros de Docker, incluidas formas alternativas de capturar diferentes datos de archivo o flujos, consulte la página sobre cómo [Ver los registros de un contenedor o servicio](#) en la documentación de Docker.

Para enviar los registros del sistema de sus instancias de contenedor a CloudWatch Logs, consulte [Uso CloudWatch de registros con AWS Batch](#). Para obtener más información sobre CloudWatch los registros, consulte [Supervisión de los archivos de registro](#) y [las cuotas de CloudWatch registros](#) en la Guía del usuario de Amazon CloudWatch Logs.

opciones del controlador de registro de `awslogs` en el tipo de datos AWS Batch `JobDefinition`

El controlador de `awslogs` registro admite las siguientes opciones en las definiciones de AWS Batch trabajos. Para obtener más información, consulte el [controlador de registro de CloudWatch registros](#) en la documentación de Docker.

`awslogs-region`

Obligatorio: no

Especifique la región a la que el controlador de registros `awslogs` debe enviarle sus registros de Docker. De forma predeterminada, la región que se utiliza es la misma que la del trabajo. Puede optar por enviar todos los registros de los trabajos de distintas regiones a una sola región en CloudWatch los registros. De este modo, podrá verlos todos desde una misma ubicación. Como alternativa, puede separarlos por región para obtener un enfoque más detallado. Sin embargo, cuando elija esta opción, asegúrese de que los grupos de registro especificados existan en la región que especifique.

awslogs-group

Obligatorio: opcional

Con la opción de `awslogs-group`, puede especificar el grupo de registro al que el controlador de registro `awslogs` envía sus flujos de registro. Si este campo no está especificado, se utiliza `aws/batch/job`.

awslogs-stream-prefix

Obligatorio: opcional

Con `awslogs-stream-prefix` esta opción, puede asociar un flujo de registro al prefijo especificado y al ID de tarea de Amazon ECS del AWS Batch trabajo al que pertenece el contenedor. Si especifica un prefijo con esta opción, entonces el flujo de registros adopta el siguiente formato:

```
prefix-name/default/ecs-task-id
```

awslogs-datetime-format

Obligatorio: no

Esta opción define un patrón de inicio de varias líneas en formato `strftime` de Python. Un mensaje de registro consta de una línea que coincide con el patrón y de líneas siguientes que no coinciden con el patrón. Por tanto, la línea coincidente es el delimitador entre los mensajes de registro.

Un ejemplo de caso de uso de este formato es para analizar la salida como un volcado de pila, que de lo contrario podría registrarse en varias entradas. El patrón correcto permite capturarla en una sola entrada.

Para obtener más información, consulte [awslogs-datetime-format](#).

Esta opción siempre tiene prioridad si ambos `awslogs-datetime-format` y `awslogs-multiline-pattern` están configurados.

Note

El registro de varias líneas realiza análisis de las expresiones regulares y correspondencia de todos los mensajes de registro. Esto puede tener un impacto negativo sobre el rendimiento de registro.

awslogs-multiline-pattern

Obligatorio: no

Esta opción define un patrón de inicio de varias líneas con una expresión regular. Un mensaje de registro consta de una línea que coincide con el patrón y de líneas siguientes que no coinciden con el patrón. Por tanto, la línea coincidente es el delimitador entre los mensajes de registro.

Para obtener más información, consulte [awslogs-multiline-pattern](#) la documentación de Docker.

Esta opción se pasa por alto si también se ha configurado `awslogs-datetime-format`.

Note

El registro de varias líneas realiza análisis de las expresiones regulares y correspondencia de todos los mensajes de registro. Esto puede tener un impacto negativo sobre el rendimiento de registro.

awslogs-create-group

Obligatorio: no

Especifique si desea que el grupo de registros se cree automáticamente. Si no se especifica esta opción, el valor predeterminado es `false`.

Warning

No se recomienda esta opción. Le recomendamos que cree el grupo de CloudWatch registros con antelación mediante la acción de la [CreateLogGroup](#) API Logs, ya que cada trabajo intenta crear el grupo de registros, lo que aumenta las probabilidades de que el trabajo falle.

Note

Su política de IAM para la ejecución de rol debe incluir el permiso `logs:CreateLogGroup` antes de intentar utilizar `awslogs-create-group`.

Especificación de una configuración de registro en la definición de trabajo

De forma predeterminada, AWS Batch habilita el controlador de `awslogs` registro. En esta sección, se describe cómo personalizar la configuración de registro `awslogs` de un trabajo. Para obtener más información, consulte [Creación de una definición de trabajo de un solo nodo](#).

Los siguientes fragmentos de JSON de configuración de registro tienen un objeto `logConfiguration` especificado para cada trabajo. Una es para un WordPress trabajo que envía registros a un grupo de registros llamado `awslogs-wordpress` y otra es para un contenedor MySQL que envía registros a un grupo de registros llamado `awslogs-mysql`. Ambos contenedores utilizan el prefijo de flujo de registros `awslogs-example`.

```
"logConfiguration": {
  "logDriver": "awslogs",
  "options": {
    "awslogs-group": "awslogs-wordpress",
    "awslogs-stream-prefix": "awslogs-example"
  }
}
```

```
"logConfiguration": {
  "logDriver": "awslogs",
  "options": {
    "awslogs-group": "awslogs-mysql",
    "awslogs-stream-prefix": "awslogs-example"
  }
}
```

En la AWS Batch consola, se especifica la configuración de registro para la definición del `wordpress` trabajo, como se muestra en la imagen siguiente.

Log configuration

Log driver

awslogs ▼

Options

Name	Value	
awslogs-group ▼	awslogs-wordpress	Remove option
awslogs-stream-prefix ▼	awslogs-example	Remove option

Add option

Secrets

Add secret

Tras registrar una definición de tarea con el controlador de `awslogs` registro en una configuración de registro de definición de trabajos, puede enviar un trabajo con esa definición de trabajo para empezar a enviar CloudWatch registros a Logs. Para obtener más información, consulte [Tutorial: envío de un trabajo](#).

Especificación de información confidencial

Con AWS Batch ella, puede introducir datos confidenciales en sus trabajos almacenándolos en AWS Secrets Manager secretos o AWS Systems Manager en parámetros del almacén de parámetros y, a continuación, hacer referencia a ellos en la definición de su trabajo.

Los secretos se pueden exponer a un trabajo de las siguientes formas:

- Para inyectar datos sensibles en sus contenedores como variables de entorno, utilice el parámetro de definición de trabajo `secrets`.
- Para hacer referencia a información sensible en la configuración del registro de un trabajo, utilice el parámetro de definición del trabajo `secretOptions`.

Temas

- [Especificación de datos confidenciales con Secrets Manager](#)
- [Especificación de información confidencial con el almacén de parámetros de Systems Manager](#)

Especificación de datos confidenciales con Secrets Manager

Con AWS Batch ella, puede introducir datos confidenciales en sus trabajos almacenándolos en AWS Secrets Manager secreto y luego haciendo referencia a ellos en su definición de trabajo. La información confidencial almacenada en secretos de Secrets Manager se puede exponer a un trabajo como variables de entorno o como parte de la configuración del registro.

Cuando inyecta un secreto como variable de entorno, puede especificar una clave JSON o versión de un secreto para inyectar. Este proceso le ayuda a controlar la información confidencial expuesta al trabajo. Para obtener más información acerca del control de versiones de los secretos, consulte los [Términos y conceptos clave de AWS Secrets Manager](#) en la Guía del usuario de AWS Secrets Manager .

Consideraciones para especificar información confidencial mediante Secrets Manager

Cuando se utilice Secrets Manager para especificar información confidencial para trabajos, se debe tener en cuenta lo siguiente.

- Para inyectar un secreto utilizando una clave JSON específica o una versión de un secreto, la instancia de contenedor de su entorno de computación debe tener instalada la versión 1.37.0 o posterior del agente de contenedores de Amazon ECS. Sin embargo, le recomendamos que utilice la versión más reciente del agente contenedor. Para obtener información sobre cómo comprobar la versión del agente y actualizarlo a la versión más reciente, consulte [Actualización del agente de contenedor de Amazon ECS](#) en la Guía para desarrolladores de Amazon Elastic Container Service.

Para inyectar el contenido completo de un secreto como variable de entorno o para inyectar un secreto en una configuración de registro, la instancia de contenedor debe tener la versión 1.23.0 o posterior del agente de contenedor.

- Solo se admiten los secretos que almacenan datos de texto, que son secretos creados con el `SecretString` parámetro de la [CreateSecret](#) API. No se admiten los secretos que almacenan datos binarios, que son secretos creados con el `SecretBinary` parámetro de la [CreateSecret](#) API.
- Cuando utilice una definición de trabajos que haga referencia a secretos de Secrets Manager para recuperar información confidencial de los trabajos, si también está utilizando puntos de conexión de VPC de la interfaz, debe crear los puntos de conexión de VPC de la interfaz para Secrets

Manager. Para obtener más información, consulte [Utilización de Secrets Manager con puntos de enlace de la VPC](#) en la Guía del usuario de AWS Secrets Manager .

- Los datos confidenciales se inyectan en el contenedor al iniciar el trabajo. Si el secreto se actualiza posteriormente o se rota, el trabajo no recibe automáticamente el valor actualizado. Debe lanzar un nuevo trabajo para obligar al servicio a lanzar uno nuevo con el valor secreto actualizado.

Permisos de IAM necesarios para los secretos AWS Batch

Para utilizar esta característica, debe tener la función de ejecución y hacer referencia a ella en la definición del trabajo. Esto permite que el agente de contenedor extraiga los recursos necesarios de Secrets Manager. Para obtener más información, consulte [AWS Batch Función de ejecución de IAM](#).

Para proporcionar acceso a los secretos de Secrets Manager que cree, agregue manualmente los siguientes permisos como una política insertada al rol de ejecución. Para obtener más información, consulte [Adición y eliminación de políticas de IAM](#) en la Guía del usuario de IAM.

- `secretsmanager:GetSecretValue`: obligatorio si se referencia un secreto de Secrets Manager.
- `kms:Decrypt`: obligatorio solo si el secreto utiliza una clave de KMS personalizada y no la clave predeterminada. El ARN de su clave personalizada debe añadirse como un recurso.

La siguiente política insertada de ejemplo agrega los permisos necesarios.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:GetSecretValue",
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:secretsmanager:us-east-2:777777777777:secret:<secret_name>",
        "arn:aws:kms:us-east-2:777777777777:key:<key_id>"
      ]
    }
  ]
}
```

```
}  
]  
}
```

Introducción de información confidencial como variable de entorno

Puede especificar los siguientes elementos en la definición del trabajo:

- El objeto `secrets` que contiene el nombre de la variable de entorno que se va a establecer en el trabajo
- Nombre de recurso de Amazon (ARN) del secreto de Secrets Manager
- Parámetros adicionales que contienen información confidencial que se debe presentar al trabajo

En el ejemplo siguiente, se muestra la sintaxis completa que se debe especificar para el secreto de Secrets Manager.

```
arn:aws:secretsmanager:region:aws_account_id:secret:secret-name:json-key:version-  
stage:version-id
```

En la siguiente sección se describen los parámetros adicionales. Algunos parámetros son opcionales. Sin embargo, si no los utiliza, debe incluir los dos puntos `:` para utilizar los valores por defecto. A continuación se ofrecen ejemplos para obtener más contexto.

`json-key`

Especifica el nombre de la clave en un par clave-valor con el valor que desea establecer como valor de variable de entorno. Solo se admiten valores en formato JSON. Si no especifica una clave JSON, se usa el contenido completo del secreto.

`version-stage`

Especifica la etiqueta de ensayo de la versión de un secreto que desea utilizar. Si se especifica una etiqueta de ensayo de versión, no se puede especificar un ID de versión. Si no se especifica ninguna etapa de versión, el comportamiento predeterminado consiste en recuperar el secreto con la etiqueta de ensayo `AWSCURRENT`.

Las etiquetas de ensayo se utilizan para realizar un seguimiento de las distintas versiones de un secreto cuando se actualizan o rotan. Cada versión de un secreto tiene una o varias etiquetas

de ensayo y un ID. Para obtener más información, consulte [Términos y conceptos clave de AWS Secrets Manager](#) en la Guía del AWS Secrets Manager usuario.

version-id

Especifica el identificador único de la versión del secreto que desea utilizar. Si se especifica un ID de versión, no se puede especificar una etiqueta de ensayo de versión. Si no se especifica ningún ID de versión, el comportamiento predeterminado consiste en recuperar el secreto con la etiqueta de ensayo AWSCURRENT.

Las versiones se IDs utilizan para realizar un seguimiento de las diferentes versiones de un secreto cuando se actualizan o se rotan. Cada versión de un secreto tiene un ID. Para obtener más información, consulte [Términos y conceptos clave de AWS Secrets Manager](#) en la Guía del AWS Secrets Manager usuario.

Definiciones de contenedor de ejemplo

En los siguientes ejemplos, se muestran las formas en las que se pueden referenciar secretos de Secrets Manager en las definiciones de contenedor.

Example hacer referencia a un secreto completo

A continuación, se incluye un fragmento de código de una definición de tarea que muestra el formato cuando se referencia el texto completo de un secreto de Secrets Manager.

```
{
  "containerProperties": [{
    "secrets": [{
      "name": "environment_variable_name",
      "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:secret_name-AbCdEf"
    }]
  }]
}
```

Example hacer referencia a una clave específica dentro de un secreto

A continuación, se muestra un ejemplo de resultado de un `get-secret-value` comando $\geq$ que muestra el contenido de un secreto junto con la etiqueta provisional de la versión y el identificador de versión asociados al mismo.


```
{
  "ARN": "arn:aws:secretsmanager:region:aws_account_id:secret:appauthexample-AbCdEf",
  "Name": "appauthexample",
  "VersionId": "871d9eca-18aa-46a9-8785-981dd39ab30c",
  "SecretString": "{\"username1\":\"password1\",\"username2\":\"password2\",\\\"username3\\\":\\\"password3\\\"}",
  "VersionStages": [
    "AWSCURRENT"
  ],
  "CreateDate": 1581968848.921
}
```

Haga referencia a una clave específica de la salida anterior en una definición de contenedor especificando el nombre de clave al final del ARN.

```
{
  "containerProperties": [{
    "secrets": [{
      "name": "environment_variable_name",
      "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:appauthexample-AbCdEf:username1::"
    }]
  }]
}
```

Example hacer referencia a una versión de secreto específica

A continuación se muestra una salida de ejemplo de un comando [>describe-secret](#) que muestra el contenido sin cifrar de un secreto junto con los metadatos de todas las versiones del secreto.

```
{
  "ARN": "arn:aws:secretsmanager:region:aws_account_id:secret:appauthexample-AbCdEf",
  "Name": "appauthexample",
  "Description": "Example of a secret containing application authorization data.",
  "RotationEnabled": false,
  "LastChangedDate": 1581968848.926,
  "LastAccessedDate": 1581897600.0,
  "Tags": [],
  "VersionIdsToStages": {
    "871d9eca-18aa-46a9-8785-981dd39ab30c": [
      "AWSCURRENT"
    ],
  },
}
```

```

        "9d4cb84b-ad69-40c0-a0ab-cead36b967e8": [
            "AWSPREVIOUS"
        ]
    }
}

```

Haga referencia a una etiqueta de ensayo de versión específica de la salida anterior en una definición de contenedor especificando el nombre de clave al final del ARN.

```

{
  "containerProperties": [{
    "secrets": [{
      "name": "environment_variable_name",
      "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:appauthexample-AbCdEf::AWSPREVIOUS:"
    }]
  }]
}

```

Haga referencia a un ID de versión específico de la salida anterior en una definición de contenedor especificando el nombre de clave al final del ARN.

```

{
  "containerProperties": [{
    "secrets": [{
      "name": "environment_variable_name",
      "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:appauthexample-AbCdEf::9d4cb84b-ad69-40c0-a0ab-cead36b967e8"
    }]
  }]
}

```

Example hacer referencia a una clave específica y una etiqueta de ensayo de versión de un secreto

A continuación se muestra cómo hacer referencia tanto a una clave específica dentro de un secreto como a una etiqueta de ensayo de versión específica.

```

{
  "containerProperties": [{
    "secrets": [{
      "name": "environment_variable_name",

```

```

    "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:appauthexample-AbCdEf:username1:AWSPREVIOUS:"
  }}
}}
}

```

Para especificar una clave y un ID de versión específicos, utilice la sintaxis siguiente.

```

{
  "containerProperties": [{
    "secrets": [{
      "name": "environment_variable_name",
      "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:appauthexample-AbCdEf:username1::9d4cb84b-ad69-40c0-a0ab-cead36b967e8"
    }]
  }]
}

```

Introducción de información confidencial en una configuración de registro

Al especificar una `logConfiguration` en la definición de trabajo, puede especificar `secretOptions` con el nombre de la opción del controlador de registros para definir el contenedor y el ARN completo del secreto de Secrets Manager que contiene la información confidencial que se va a presentar al contenedor.

A continuación, se incluye un fragmento de código de una definición de trabajo que muestra el formato cuando se referencia un secreto de Secrets Manager.

```

{
  "containerProperties": [{
    "logConfiguration": [{
      "logDriver": "splunk",
      "options": {
        "splunk-url": "https://cloud.splunk.com:8080"
      },
      "secretOptions": [{
        "name": "splunk-token",
        "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:secret_name-AbCdEf"
      }]
    }]
  }]
}

```

```
}]  
}
```

Crea un secreto AWS Secrets Manager

Puede utilizar la consola de Secrets Manager para crear un secreto con su información confidencial. Para obtener más información, consulte [Creación de un secreto básico](#) en la Guía del usuario de AWS Secrets Manager .

Para crear un secreto básico

Utilice Secrets Manager para crear un secreto con su información confidencial.

1. Abra la consola de Secrets Manager en <https://console.aws.amazon.com/secretsmanager/>.
2. Elija Almacenar un secreto nuevo.
3. En Select secret type (Seleccionar tipo de secreto), elija Other type of secrets (Otro tipo de secretos).
4. Especifique los detalles de su secreto personalizado como pares de clave y valor. Por ejemplo, puede especificar una clave `UserName` y, a continuación, proporcionar el nombre de usuario como su valor. Agregue una segunda clave con el nombre `Password` y el texto de la contraseña como su valor. También puede añadir entradas para el nombre de una base de datos, la dirección del servidor o el puerto TCP. Puede añadir tantas parejas como sea necesario para almacenar la información que necesite.

Como opción, puede elegir la pestaña Plaintext (Texto no cifrado) y escribir el valor del secreto del modo que prefiera.

5. Elija la clave de AWS KMS cifrado que desee utilizar para cifrar el texto protegido del secreto. Si no elige ninguna, Secrets Manager comprueba si existe una clave predeterminada para la cuenta y la utiliza si existe. Si no existe una clave predeterminada, Secrets Manager crea una automáticamente. También puede elegir Add new key (Añadir nueva clave) para crear una clave de KMS personalizada específicamente para este secreto. Para crear su propia clave de KMS, debe tener permisos para crear claves de KMS en su cuenta.
6. Elija Siguiente.
7. En Secret name (Nombre del secreto), escriba una ruta y un nombre opcionales, tales como **production/MyAwesomeAppSecret** o **development/TestSecret** y elija Next (Siguiente). Opcionalmente, puede añadir una descripción para ayudarle a recordar el objetivo de este secreto más adelante.

El nombre del secreto debe estar formado por letras ASCII, dígitos o cualquiera de los siguientes caracteres: /_+=.@-

8. (Opcional) En este punto puede configurar la rotación para su secreto. Para este procedimiento, deje seleccionado **Disable automatic rotation** (Deshabilitar la rotación automática) y, a continuación, elija **Next** (Siguiente).

Para obtener información sobre cómo configurar la rotación de datos secretos nuevos o existentes, consulte [Rotación de sus datos AWS Secrets Manager secretos](#).

9. Revise la configuración y, a continuación, elija **Store secret** (Almacenar secreto) para guardar todo lo que ingresó como nuevo secreto en Secrets Manager.

Especificación de información confidencial con el almacén de parámetros de Systems Manager

Con él AWS Batch, puede introducir datos confidenciales en sus contenedores almacenándolos en los parámetros del almacén de parámetros y, a continuación, haciendo referencia a ellos en la definición de su contenedor. AWS Systems Manager

Temas

- [Consideraciones para especificar información confidencial mediante el Parameter Store de Systems Manager](#)
- [Permisos de IAM necesarios para los secretos AWS Batch](#)
- [Introducción de información confidencial como variable de entorno](#)
- [Introducción de información confidencial en una configuración de registro](#)
- [Cree un AWS Systems Manager parámetro de almacén de parámetros](#)

Consideraciones para especificar información confidencial mediante el Parameter Store de Systems Manager

Cuando se especifica información confidencial para contenedores que utilizan parámetros del Parameter Store de Systems Manager, se debe tener en cuenta lo siguiente.

- Esta característica requiere que la instancia de contenedor tenga una versión 1.23.0 o posterior del agente de contenedor. Sin embargo, le recomendamos que utilice la versión más reciente del

agente de contenedores. Para obtener información sobre cómo comprobar la versión del agente y actualizarlo a la versión más reciente, consulte [Actualización del agente de contenedor de Amazon ECS](#) en la Guía para desarrolladores de Amazon Elastic Container Service.

- Los datos confidenciales se inyectan en el contenedor para su trabajo al iniciar el contenedor. Si el secreto o el parámetro de Parameter Store se actualizan posteriormente o se rotan, el contenedor no recibe automáticamente el valor actualizado. Debe lanzar un nuevo trabajo para obligar el lanzamiento de uno nuevo con los secretos actualizados.

Permisos de IAM necesarios para los secretos AWS Batch

Para utilizar esta característica, debe tener la función de ejecución y hacer referencia a ella en la definición del trabajo. Esto permite al agente de contenedores de Amazon ECS obtener los AWS Systems Manager recursos necesarios. Para obtener más información, consulte [AWS Batch Función de ejecución de IAM](#).

Para proporcionar acceso a los AWS Systems Manager parámetros del almacén de parámetros que cree, añada manualmente los siguientes permisos como política integrada a la función de ejecución. Para obtener más información, consulte [Adición y eliminación de políticas de IAM](#) en la Guía del usuario de IAM.

- `ssm:GetParameters`: obligatorio si se hace referencia a un parámetro del Parameter Store de Systems Manager en una definición de tareas.
- `secretsmanager:GetSecretValue`: obligatorio si se hace referencia a un secreto de Secrets Manager directamente o si el parámetro del Parameter Store de Systems Manager hace referencia a un secreto de Secrets Manager en una definición de tareas.
- `kms:Decrypt`: obligatorio solo si el secreto utiliza una clave de KMS personalizada y no la clave predeterminada. El ARN de su clave personalizada debe añadirse como un recurso.

A continuación, mostramos un ejemplo de política insertada que añade los permisos necesarios.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
```

```

        "Action": [
            "ssm:GetParameters",
            "secretsmanager:GetSecretValue",
            "kms:Decrypt"
        ],
        "Resource": [
            "arn:aws:ssm:us-east-2:999999999999:parameter/<parameter_name>",
            "arn:aws:secretsmanager:us-east-2:999999999999:secret:<secret_name>",
            "arn:aws:kms:us-east-2:999999999999:key/<key_id>"
        ]
    }
}

```

Introducción de información confidencial como variable de entorno

En la definición de contenedor, especifique `secrets` con el nombre de la variable de entorno que se va a establecer en el contenedor y el ARN completo del parámetro del Parameter Store de Systems Manager que contiene la información confidencial que se va a presentar al contenedor.

A continuación, se incluye un fragmento de código de una definición de tarea que muestra el formato cuando se hace referencia a un parámetro del almacén de parámetros de Systems Manager. Si el parámetro del almacén de parámetros de Systems Manager está en la misma región que la tarea que se va a lanzar, se puede utilizar el ARN completo o el nombre del parámetro. Si el parámetro existe en una región distinta, el ARN completo debe especificarse.

```

{
  "containerProperties": [{
    "secrets": [{
      "name": "environment_variable_name",
      "valueFrom": "arn:aws:ssm:region:aws_account_id:parameter/parameter_name"
    }]
  }]
}

```

Introducción de información confidencial en una configuración de registro

En la definición de contenedor, al especificar una `logConfiguration`, puede especificar `secretOptions` con el nombre de la opción del controlador de registros que va a definir en el

contenedor y el ARN completo del parámetro del Parameter Store de Systems Manager que contiene la información confidencial que se va a presentar al contenedor.

Important

Si el parámetro del almacén de parámetros de Systems Manager está en la misma región que la tarea que se va a lanzar, se puede utilizar el ARN completo o el nombre del parámetro. Si el parámetro existe en una región distinta, el ARN completo debe especificarse.

A continuación, se incluye un fragmento de código de una definición de tarea que muestra el formato cuando se hace referencia a un parámetro del almacén de parámetros de Systems Manager.

```
{
  "containerProperties": [{
    "logConfiguration": [{
      "logDriver": "fluentd",
      "options": {
        "tag": "fluentd demo"
      },
      "secretOptions": [{
        "name": "fluentd-address",
        "valueFrom": "arn:aws:ssm:region:aws_account_id:parameter/parameter_name"
      }]
    }]
  }]
}
```

Cree un AWS Systems Manager parámetro de almacén de parámetros

Puede utilizar la AWS Systems Manager consola para crear un parámetro del almacén de parámetros de Systems Manager para sus datos confidenciales. Para obtener más información, consulte [Explicación: Creación y utilización de un parámetro en un comando \(consola\)](#) en la Guía del usuario de AWS Systems Manager .

Para crear un parámetro con Parameter Store

1. Abra la AWS Systems Manager consola en <https://console.aws.amazon.com/systems-manager/>.
2. En el panel de navegación, elija Parameter Store, Create parameter (Creación de parámetro).

3. En Name (Nombre), escriba una jerarquía y un nombre de parámetro. Por ejemplo, escriba `test/database_password`.
4. En Description (Descripción), escriba una descripción opcional.
5. En Tipo, elija Cadena StringList, o SecureString.

 Note

- Si lo elige SecureString, aparece el campo ID de clave de KMS. Si no proporciona un ID de clave de KMS, un ARN de clave de KMS, un nombre de alias o un ARN de alias, el sistema utiliza `alias/aws/ssm`. Esta es la clave de KMS predeterminada para Systems Manager. Para evitar utilizar esta clave, elija una clave personalizada. Para obtener más información, consulte [Utilización de parámetros de cadena segura](#) en la Guía del usuario de AWS Systems Manager .
- Si crea un parámetro de cadena segura en la consola mediante el parámetro `key-id` con un nombre de alias de clave de KMS o un ARN de alias personalizado, debe especificar el prefijo `alias/` antes del alias. A continuación se muestra un ejemplo de ARN:

```
arn:aws:kms:us-east-2:123456789012:alias/MyAliasName
```

El siguiente es un ejemplo de un nombre de alias:

```
alias/MyAliasName
```

6. En Value (Valor), escriba un valor. Por ejemplo, `MyFirstParameter`. Si lo elige SecureString, el valor se enmascara exactamente como lo ingresó.
7. Elija Create parameter.

Autenticación de registros privados para trabajos

La autenticación del registro privado para los trabajos AWS Secrets Manager le permite almacenar sus credenciales de forma segura y luego hacer referencia a ellas en su definición de trabajo. Esto proporciona una forma de hacer referencia a las imágenes de contenedores que existen en registros privados y AWS que no requieren autenticación en las definiciones de trabajo. Esta característica se admite para trabajos alojados en instancias de Amazon EC2 y Fargate.

 Important

Si la definición del trabajo hace referencia a una imagen que está almacenada en Amazon ECR, este tema no es válido. Para obtener más información, consulte [Utilización de imágenes de Amazon ECR con Amazon ECS](#) en la Guía del usuario de Amazon Elastic Container Registry.

Para los trabajos alojados en instancias de Amazon EC2, esta característica requiere la versión 1.19.0 del agente de contenedor o una posterior. Sin embargo, recomendamos utilizar la versión más reciente del agente contenedor. Para obtener información sobre cómo comprobar la versión del agente y actualizarlo a la versión más reciente, consulte [Actualización del agente de contenedor de Amazon ECS](#) en la Guía para desarrolladores de Amazon Elastic Container Service.

Para los trabajos alojados en Fargate, esta característica requiere la versión de la plataforma 1.2.0 o posterior. Para obtener más información, consulte [Versiones de la plataforma de AWS Fargate Linux](#) en la Guía para desarrolladores de Amazon Elastic Container Service.

En la definición de contenedor, especifique el objeto `repositoryCredentials` con los detalles del secreto que ha creado. El secreto al que haga referencia puede provenir de una cuenta distinta Región de AWS o distinta de la del trabajo que lo utiliza.

 Note

Al usar la AWS Batch API o el AWS SDK, si el secreto existe en el Región de AWS mismo lugar que el trabajo que está lanzando, puede usar el ARN completo o el nombre del secreto. AWS CLI Si el secreto existe en otra cuenta, debe especificarse el ARN completo del secreto. Al utilizar el Consola de administración de AWS, se debe especificar siempre el ARN completo del secreto.

A continuación, se incluye un fragmento de definición de trabajos que muestra los parámetros necesarios:

```
"containerProperties": [  
  {  
    "image": "private-repo/private-image",  
    "repositoryCredentials": {
```

```
    "credentialsParameter":  
      "arn:aws:secretsmanager:region:123456789012:secret:secret_name"  
    }  
  }  
]
```

Permisos de IAM requeridos para la autenticación de registros privados

Se requiere la función de ejecución para utilizar esta característica. Esto permite que el agente de contenedor extraiga la imagen del contenedor. Para obtener más información, consulte [AWS Batch Función de ejecución de IAM](#).

Para proporcionar acceso a los secretos que cree, agregue los siguientes permisos como una política insertada al rol de ejecución. Para obtener más información, consulte [Adición y eliminación de políticas de IAM](#).

- `secretsmanager:GetSecretValue`
- `kms:Decrypt`: solo se requiere si la clave utiliza una clave de KMS personalizada y no la clave de KMS predeterminada. Se debe agregar el nombre de recurso de Amazon (ARN) de la clave de personalizada como un recurso.

Es siguiente es un ejemplo de política insertada que agrega los permisos.

JSON

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "kms:Decrypt",  
        "secretsmanager:GetSecretValue"  
      ],  
      "Resource": [  
        "arn:aws:secretsmanager:us-east-1:123456789012:secret:secret_name",  
        "arn:aws:kms:us-east-1:123456789012:key/key_id"  
      ]  
    }  
  ]  
}
```

```
}  
]
```

Tutorial: creación de un secreto para la autenticación de registros privados

Complete los siguientes pasos para crear un secreto para sus credenciales de registro privado con AWS Secrets Manager.

Creación de un secreto básico

1. Abra la AWS Secrets Manager consola en <https://console.aws.amazon.com/secretsmanager/>.
2. Elija Almacenar un secreto nuevo.
3. En Select secret type (Seleccionar tipo de secreto), elija Other type of secrets (Otro tipo de secretos).
4. Seleccione Plaintext (Texto no cifrado) y especifique sus credenciales de registros privados con el siguiente formato:

```
{  
  "username" : "privateRegistryUsername",  
  "password" : "privateRegistryPassword"  
}
```

5. Elija Siguiente.
6. En Secret name (Nombre del secreto), ingrese una ruta y un nombre opcionales, como **production/MyAwesomeAppSecret** o **development/TestSecret**, y elija Next (Siguiente). Opcionalmente, puede añadir una descripción para ayudarle a recordar el objetivo de este secreto más adelante.

El nombre del secreto debe estar formado por letras ASCII, dígitos o alguno de los siguientes caracteres: /_+=.@-.

7. (Opcional) En este punto puede configurar la rotación para su secreto. Para este procedimiento, deje seleccionado Disable automatic rotation (Deshabilitar la rotación automática) y, a continuación, elija Next (Siguiente).

Para obtener instrucciones sobre cómo configurar la rotación de secretos nuevos o existentes, consulte Cómo [rotar sus AWS Secrets Manager secretos](#).

8. Revise la configuración y, a continuación, elija Store secret (Almacenar secreto) para guardar todo lo que ingresó como secreto nuevo en Secrets Manager.

Registre una definición de trabajo y, en Registro privado, active Autenticación del registro privado. A continuación, en nombre o ARN de Secrets Manager, introduzca el nombre de recurso de Amazon (ARN) del secreto. Para obtener más información, consulte [Permisos de IAM requeridos para la autenticación de registros privados](#).

Volúmenes de Amazon EFS

Amazon Elastic File System (Amazon EFS) proporciona almacenamiento de archivos sencillo y escalable para usarlo con sus trabajos AWS Batch. Con Amazon EFS, la capacidad de almacenamiento es elástica. Escala automáticamente a medida que se agregan o eliminan archivos. Las aplicaciones disponen del almacenamiento que necesitan, cuando lo necesitan.

Puede utilizar sistemas de archivos de Amazon EFS con AWS Batch para exportar los datos del sistema de archivos a través de la flota de instancias de contenedor. De esta forma, sus trabajos tienen acceso al mismo almacenamiento persistente. No obstante, debe configurar su AMI de instancia de contenedor para montar el sistema de archivos de Amazon EFS antes de que se inicie el daemon de Docker. Además, sus definiciones de trabajo deben hacer referencia a montajes de volúmenes en la instancia de contenedor para utilizar el sistema de archivos. Las secciones siguientes le ayudarán a comenzar a utilizar Amazon EFS con AWS Batch.

Consideraciones acerca de volúmenes de Amazon EFS

Al utilizar volúmenes de Amazon EFS, se debe tener en cuenta lo siguiente:

- Para los trabajos que utilizan recursos EC2, se ha agregado compatibilidad con el sistema de archivos de Amazon EFS como vista previa pública en la versión AMI optimizada para Amazon ECS 20191212 con el agente de contenedor versión 1.35.0. Sin embargo, el sistema de archivos de Amazon EFS está disponible con carácter general a partir de la versión 20200319 de la AMI optimizada para Amazon ECS con el agente de contenedor versión 1.38.0, que contenía las características de punto de acceso de Amazon EFS y autorización de IAM. Recomendamos utilizar la versión AMI optimizada para Amazon ECS 20200319 o una posterior para aprovechar estas características. Para obtener más información, consulte la sección sobre [Versiones AMI Linux optimizada para Amazon ECS](#) en la Guía para desarrolladores de Amazon Elastic Container Service.

 Note

Si crea su propia AML, debe usar el agente de contenedor 1.38.0 o posterior, versión `ecs-init` 1.38.0-1 o una posterior, y ejecutar los siguientes comandos en su instancia de Amazon EC2. Todo esto es para habilitar el complemento de volumen Amazon ECS. Los comandos dependen de si utiliza Amazon Linux 2 o Amazon Linux como imagen base.

Amazon Linux 2

```
$ yum install amazon-efs-utils
systemctl enable --now amazon-ecs-volume-plugin
```

Amazon Linux

```
$ yum install amazon-efs-utils
sudo shutdown -r now
```

- Para los trabajos que utilizan recursos de Fargate, se agregó la compatibilidad con el sistema de archivos de Amazon EFS al usar la versión de la plataforma 1.4.0 o posterior. Para obtener más información, consulte [Versiones de la plataforma de AWS Fargate](#) en la Guía para desarrolladores de Amazon Elastic Container Service.
- Cuando se utilizan volúmenes de Amazon EFS en trabajos que usan recursos de Fargate, Fargate crea un contenedor supervisor responsable de administrar el volumen de Amazon EFS. El contenedor supervisor utiliza una pequeña cantidad de la memoria del trabajo. El contenedor supervisor puede verse al consultar la versión 4 del punto de conexión de metadatos de la tarea. Para obtener más información, consulte [Versión 4 del punto de conexión de metadatos de tareas](#) en la Guía del usuario de Amazon Elastic Container Service para AWS Fargate.

Uso de puntos de acceso de Amazon EFS

Los puntos de acceso de Amazon EFS son puntos de entrada específicos de la aplicación a un sistema de archivos de EFS que le ayudan a administrar el acceso de la aplicación a conjuntos de datos compartidos. Para obtener más información acerca de los puntos de acceso de Amazon EFS y cómo controla el acceso a ellos, consulte [Uso de puntos de acceso de Amazon EFS](#) en la Guía del usuario de Amazon Elastic File System.

Los puntos de acceso pueden imponer una identidad de usuario, incluidos los grupos POSIX del usuario, para todas las solicitudes del sistema de archivos que se realizan a través de ellos. Los puntos de acceso también pueden imponer un directorio raíz diferente para el sistema de archivos, de modo que los clientes solo puedan acceder a los datos del directorio especificado o de sus subdirectorios.

Note

Cuando se crea un punto de acceso EFS, se especifica una ruta en el sistema de archivos para que sirva como directorio raíz. Cuando se hace referencia al sistema de archivos de EFS con un ID de punto de acceso en la definición de trabajo de AWS Batch, el directorio raíz se debe omitir o establecer en /, lo que aplicará la ruta establecida en el punto de acceso de EFS.

Puede utilizar un rol de IAM con el trabajo de AWS Batch para exigir que determinadas aplicaciones utilicen un punto de acceso específico. Al combinar políticas de IAM con puntos de acceso, puede proporcionar fácilmente acceso seguro a conjuntos de datos específicos para sus aplicaciones. Esta característica utiliza roles de IAM de Amazon ECS para otorgarle funcionalidad a la tarea. Para obtener más información, consulte [Roles de IAM para tareas](#) en la Guía para desarrolladores de Amazon Elastic Container Service.

Indique un sistema de archivos de Amazon EFS en la definición del trabajo

Para usar volúmenes del sistema de archivos de Amazon EFS para sus contenedores, debe especificar las configuraciones de volumen y punto de montaje en su definición de trabajo. El siguiente fragmento de JSON de definición de trabajo muestra la sintaxis de los objetos `volumes` y `mountPoints` para un contenedor:

```
{
  "containerProperties": [
    {
      "image": "amazonlinux:2",
      "command": [
        "ls",
        "-la",
        "/mount/efs"
      ],
      "mountPoints": [
```

```
{
  {
    "sourceVolume": "myEfsVolume",
    "containerPath": "/mount/efs",
    "readOnly": true
  }
],
"volumes": [
  {
    "name": "myEfsVolume",
    "efsVolumeConfiguration": {
      "fileSystemId": "fs-12345678",
      "rootDirectory": "/path/to/my/data",
      "transitEncryption": "ENABLED",
      "transitEncryptionPort": integer,
      "authorizationConfig": {
        "accessPointId": "fsap-1234567890abcdef1",
        "iam": "ENABLED"
      }
    }
  }
]
}
```

efsVolumeConfiguration

Tipo: objeto

Obligatorio: no

Este parámetro se especifica cuando se usan volúmenes de Amazon EFS.

fileSystemId

Tipo: cadena

Obligatorio: sí

El ID del sistema de archivos de Amazon EFS que se va a usar.

rootDirectory

Tipo: cadena

Obligatorio: no

Directorio del sistema de archivos de Amazon EFS que se va a montar como directorio raíz dentro del host. Si se omite este parámetro, se utiliza la raíz del volumen de Amazon EFS. Si se especifica /, se obtiene el mismo efecto que si se omite este parámetro. Puede tener hasta 4 096 caracteres de longitud.

 Important

Si se especifica un punto de acceso de EFS en el `authorizationConfig`, se debe omitir el parámetro del directorio raíz o establecerlo en /. Esto impone la ruta establecida en el punto de acceso de EFS.

`transitEncryption`

Tipo: cadena

Valores válidos: ENABLED | DISABLED

Obligatorio: no

Determina si se habilita el cifrado de los datos en tránsito de Amazon EFS entre el host de AWS Batch y el servidor de Amazon EFS. El cifrado en tránsito debe estar habilitado si se utiliza la autorización de IAM en Amazon EFS. Si se omite este parámetro, se usa el valor predeterminado de DISABLED. Para obtener más información, consulte [Cifrado de datos en tránsito](#) en la Guía del usuario de Amazon Elastic File System.

`transitEncryptionPort`

Tipo: entero

Obligatorio: no

El puerto que se utilizará al enviar datos cifrados entre el host de AWS Batch y el servidor de Amazon EFS. Si no se especifica un puerto de cifrado en tránsito, se emplea la estrategia de selección de puertos que utiliza el ayudante de montaje de Amazon EFS. Este valor debe estar entre 0 y 65 535. Para obtener más información, consulte [Ayudante de montaje de EFS](#) en la Guía del usuario de Amazon Elastic File System.

`authorizationConfig`

Tipo: objeto

Obligatorio: no

Los detalles de configuración de autorización en el sistema de archivos de Amazon EFS.

`accessPointId`

Tipo: cadena

Requerido: no

ID de punto de acceso que se va a utilizar. Si se especifica un punto de acceso, el valor del directorio raíz en `efsVolumeConfiguration` se debe omitir o establecer en `/`.

Esto impone la ruta establecida en el punto de acceso de EFS. Si se utiliza un punto de acceso, el cifrado de tránsito debe estar habilitado en el `EFSVolumeConfiguration`.

Para obtener más información, consulte [Trabajo con puntos de acceso de Amazon EFS](#) en la Guía del usuario de Amazon Elastic File System.

`iam`

Tipo: cadena

Valores válidos: ENABLED | DISABLED

Obligatorio: no

Determina si se debe utilizar el rol de IAM del trabajo AWS Batch definido en una definición de trabajo al montar el sistema de archivos de Amazon EFS. Si está habilitado, el cifrado de tránsito debe estar habilitado en el `EFSVolumeConfiguration`. Si se omite este parámetro, se usa el valor predeterminado de DISABLED. Para obtener más información sobre el rol de ejecución de IAM, consulte [AWS Batch Función de ejecución de IAM](#).

Ejemplos de definición de trabajo

Los ejemplos de definición de trabajo a continuación ilustran cómo utilizar patrones comunes, como las variables de entorno, las sustituciones de parámetros y los montajes de volúmenes.

Contenido

- [Variables de entorno](#)
- [Sustitución de parámetros](#)
- [Funcionalidad de GPU de prueba](#)
- [Trabajo paralelo de varios nodos](#)

Variables de entorno

En el siguiente ejemplo, la definición de trabajo utiliza variables de entorno para especificar un tipo de archivo y un URL de Amazon S3. Este ejemplo se ha extraído del artículo de blog de informática titulado [Creating a Simple "Fetch & Run" AWS Batch Job](#). El script [fetch_and_run.sh](#) que se describe en el artículo utiliza estas variables de entorno para descargar el script `myjob.sh` de S3 y declarar su tipo de archivo.

Aunque en este ejemplo el comando y las variables de entorno están codificadas de forma rígida en la definición de trabajo, puede especificar sustituciones de comandos y de variables de entorno para que la definición de trabajo más versátil.

```
{
  "jobDefinitionName": "fetch_and_run",
  "type": "container",
  "containerProperties": {
    "image": "123456789012.dkr.ecr.us-east-1.amazonaws.com/fetch_and_run",
    "resourceRequirements": [
      {
        "type": "MEMORY",
        "value": "2000"
      },
      {
        "type": "VCPU",
        "value": "2"
      }
    ],
    "command": [
      "myjob.sh",
      "60"
    ],
    "jobRoleArn": "arn:aws:iam::123456789012:role/AWSBatchS3ReadOnly",
    "environment": [
      {
        "name": "BATCH_FILE_S3_URL",
        "value": "s3://amzn-s3-demo-source-bucket/myjob.sh"
      },
      {
        "name": "BATCH_FILE_TYPE",
        "value": "script"
      }
    ]
  },
}
```

```
    "user": "nobody"
  }
}
```

Sustitución de parámetros

El siguiente ejemplo de definición de trabajo ilustra cómo permitir la sustitución de parámetros y cómo establecer valores predeterminados.

Las declaraciones `Ref::` de la sección `command` se utilizan para definir marcadores de posición para sustituir parámetros. Al enviar un trabajo con esta definición de trabajo, se especifican las sustituciones de parámetros que ocuparán dichos valores, como `inputfile` y `outputfile`. La sección `parameters` a continuación establece un valor predeterminado para `codec`, pero es posible invalidar ese parámetro si fuera necesario.

Para obtener más información, consulte [Parameters](#).

```
{
  "jobDefinitionName": "ffmpeg_parameters",
  "type": "container",
  "parameters": {"codec": "mp4"},
  "containerProperties": {
    "image": "my_repo/ffmpeg",
    "resourceRequirements": [
      {
        "type": "MEMORY",
        "value": "2000"
      },
      {
        "type": "VCPU",
        "value": "2"
      }
    ],
    "command": [
      "ffmpeg",
      "-i",
      "Ref::inputfile",
      "-c",
      "Ref::codec",
      "-o",
      "Ref::outputfile"
    ],
  },
}
```

```

    "jobRoleArn": "arn:aws:iam::123456789012:role/ECSTask-S3FullAccess",
    "user": "nobody"
  }
}

```

Funcionalidad de GPU de prueba

En el siguiente ejemplo, la definición de trabajo prueba si la AMI de carga de trabajo de GPU descrita en [Uso de una AMI de carga de trabajo de GPU](#) se ha configurado correctamente. Esta definición de trabajo de ejemplo ejecuta el [ejemplo](#) del clasificador TensorFlow deep MNIST de GitHub.

```

{
  "containerProperties": {
    "image": "tensorflow/tensorflow:1.8.0-devel-gpu",
    "resourceRequirements": [
      {
        "type": "MEMORY",
        "value": "32000"
      },
      {
        "type": "VCPU",
        "value": "8"
      }
    ],
    "command": [
      "sh",
      "-c",
      "cd /tensorflow/tensorflow/examples/tutorials/mnist; python mnist_deep.py"
    ]
  },
  "type": "container",
  "jobDefinitionName": "tensorflow_mnist_deep"
}

```

Es posible crear un archivo con el texto JSON indicado arriba, que se denomine `tensorflow_mnist_deep.json` y, a continuación, registrar una definición de trabajo de AWS Batch con el siguiente comando:

```
aws batch register-job-definition --cli-input-json file://tensorflow_mnist_deep.json
```

Trabajo paralelo de varios nodos

La definición de trabajo de ejemplo siguiente muestra un trabajo paralelo de varios nodos.

Para obtener más información, consulte [Creación de un flujo de trabajo de dinámica molecular estrechamente relacionado con trabajos paralelos de múltiples nodos en AWS Batch](#) en el blog de AWS Compute.

```
{
  "jobDefinitionName": "gromacs-jobdef",
  "jobDefinitionArn": "arn:aws:batch:us-east-2:123456789012:job-definition/gromacs-jobdef:1",
  "revision": 6,
  "status": "ACTIVE",
  "type": "multinode",
  "parameters": {},
  "nodeProperties": {
    "numNodes": 2,
    "mainNode": 0,
    "nodeRangeProperties": [
      {
        "targetNodes": "0:1",
        "container": {
          "image": "123456789012.dkr.ecr.us-east-2.amazonaws.com/gromacs_mpi:latest",
          "resourceRequirements": [
            {
              "type": "MEMORY",
              "value": "24000"
            },
            {
              "type": "VCPU",
              "value": "8"
            }
          ],
          "command": [],
          "jobRoleArn": "arn:aws:iam::123456789012:role/ecsTaskExecutionRole",
          "ulimits": [],
          "instanceType": "p3.2xlarge"
        }
      }
    ]
  }
}
```

Tareas

Los trabajos son la unidad de trabajo con la que se empieza. AWS Batch Los trabajos se pueden invocar como aplicaciones en contenedores que se ejecutan en instancias de contenedor de Amazon ECS dentro de un clúster de ECS.

Los trabajos en contenedores pueden hacer referencia a una imagen, comando o parámetros de contenedor. Para obtener más información, consulte [JobDefinition](#).

Puede enviar una gran cantidad de trabajos sencillos e independientes.

Temas

- [Tutorial: envío de un trabajo](#)
- [Trabajos de servicio en AWS Batch](#)
- [Estados de trabajo](#)
- [AWS Batch variables de entorno laboral](#)
- [Reintentos automáticos de trabajo](#)
- [Dependencias de trabajos](#)
- [Tiempos de espera de trabajo](#)
- [Trabajos de Amazon EKS](#)
- [Trabajos paralelos de varios nodos](#)
- [Trabajos paralelos de varios nodos en Amazon EKS](#)
- [Trabajos de matrices](#)
- [Ejecución de trabajos de GPU](#)
- [Vea los AWS Batch trabajos en una cola de trabajos](#)
- [AWS Batch Busque trabajos en una cola de trabajos](#)
- [Modos de red para AWS Batch trabajos](#)
- [Vea los registros de AWS Batch trabajos en CloudWatch los registros](#)
- [Revise la información del AWS Batch trabajo](#)

Tutorial: envío de un trabajo

Después de registrar una definición de trabajo, puede enviarla como trabajo a una cola de AWS Batch trabajos. Muchos de los parámetros que se especifican en la definición de tareas pueden ignorarse en tiempo de ejecución.

Envío de un trabajo

1. Abra la AWS Batch consola en. <https://console.aws.amazon.com/batch/>
2. En la barra de navegación, seleccione la Región de AWS que desee utilizar.
3. En el panel de navegación, seleccione Trabajos.
4. Seleccione Enviar el trabajo.
5. En Nombre, escriba un nombre único para la definición de trabajo. El nombre puede tener una longitud máxima de 128 caracteres. Puede contener letras mayúsculas y minúsculas, números, guiones (-) y guiones bajos (_).
6. En Definición de trabajo, seleccione una definición de trabajo creada anteriormente. Para obtener más información, consulte [Creación de una definición de trabajo de un solo nodo](#).
7. En Cola de trabajos, elija una cola de trabajos existente. Para obtener más información, consulte [Creación de una cola de trabajos](#).
8. En Dependencias de trabajos, elija Añadir dependencias de trabajos.
 - En ID de trabajo, introduzca el ID de trabajo de cualquier dependencia. A continuación, seleccione Añadir dependencias de trabajos. Un trabajo puede tener hasta 20 dependencias. Para obtener más información, consulte [Dependencias de trabajos](#).
9. (Solo trabajos de matrices) En Tamaño de matriz, especifique un tamaño de matriz comprendido entre 2 y 10 000.
10. (Opcional) Amplíe Etiquetas y, a continuación, elija Agregar etiqueta para agregar etiquetas al recurso. Elija Agregar nueva etiqueta e introduzca la clave y el valor opcional.
11. Seleccione Página siguiente.
12. En la sección Anulaciones de trabajos:
 - a. (Opcional) En Prioridad de programación, introduzca un valor de prioridad de programación entre 0 y 100. Los valores más altos tienen mayor prioridad.

- b. (Opcional) En Intentos de trabajo, introduzca el número máximo de veces que AWS Batch intenta mover el trabajo a un estado `RUNNABLE`. Puede especificar un número comprendido entre 1 y 10. Para obtener más información, consulte [Reintentos automáticos de trabajo](#).
- c. (Opcional) En Tiempo de espera de ejecución, introduzca el valor del tiempo de espera (en segundos). El tiempo de espera de ejecución es el tiempo que transcurre antes de que finalice un trabajo pendiente. Si un intento supera el tiempo de espera, se detiene y el estado cambia a `FAILED`. Para obtener más información, consulte [Tiempos de espera de trabajo](#). El valor mínimo es de 60 segundos.

 Important

No confíe en que los trabajos que se ejecuten con los recursos de Fargate duren más de 14 días. Después de 14 días, es posible que los recursos de Fargate ya no estén disponibles y es probable que el trabajo se finalice.

- d. (Opcional) Active Propagar etiquetas para propagar etiquetas desde el trabajo y la definición del trabajo a la tarea de Amazon ECS.

13. Expanda Configuración adicional.

14. (Opcional) Para las Condiciones de la estrategia de reintento, seleccione Agregar evaluación al salir. Introduzca al menos un valor de parámetro y, a continuación, elija una Acción. Para cada conjunto de condiciones, la Acción debe estar configurada como Reintentar o Salir. Estas acciones significan lo siguiente:

- Reintentar: AWS Batch se vuelve a intentar hasta alcanzar el número de intentos de trabajo que especificó.
- Salir: AWS Batch deja de volver a intentar el trabajo.

 Important

Si elige Añadir evaluación al salir, configure al menos un parámetro y elija una Acción o elija Eliminar evaluación al salir.

15. En Parámetros, seleccione Añadir parámetros para añadir marcadores de sustitución de parámetros. Ingrese una clave y un valor opcional.

16. En la sección Anulaciones de contenedores:

- a. En Comando, ingrese los comandos en el campo como su equivalente de matriz de cadenas JSON.

Este parámetro se asigna a Cmd en la sección [Creación de un contenedor](#) de la [API remota de Docker](#) y el parámetro COMMAND se corresponde con [docker run](#). Para obtener más información sobre el CMD parámetro de Docker, consulte <https://docs.docker.com/engine/reference/builder/#cmd>.

 Note

Este parámetro no puede contener una cadena vacía.

- b. Para v CPUs, introduce el número de v CPUs que desees reservar para el contenedor. Este parámetro se corresponde con CpuShares en la sección [Creación de un contenedor](#) de la [API remota de Docker](#) y con la opción --cpu-shares de [docker run](#). Cada vCPU es equivalente a 1 024 cuotas de CPU. Debe especificar al menos una vCPU.
- c. En Memoria, introduzca el límite de memoria disponible para el contenedor. Si su contenedor intenta superar la memoria especificada aquí, el contenedor se detiene. Este parámetro se corresponde con Memory en la sección [Creación de un contenedor](#) de la [API remota de Docker](#) y con la opción --memory de [docker run](#). Debe especificar al menos 4 MiB de memoria para un trabajo.

 Note

Para maximizar el uso de los recursos, priorice la memoria para los trabajos de un tipo de instancia específico. Para obtener más información, consulte [Administración de la memoria de recurso de computación](#).

- d. (Opcional) En Número de GPUs, elija el número de GPUs que desea reservar para el contenedor.
- e. (Opcional) En el caso de Variables de entorno, seleccione Agregar variable de entorno para añadir variables de entorno como pares de nombre-valor. Estas variables se transfieren al contenedor.
- f. Seleccione Página siguiente.
- g. En Revisión del trabajo, revise los pasos de configuración. Si necesita realizar cambios, elija Editar. Cuando haya terminado, seleccione Creación de definición de trabajo.

Trabajos de servicio en AWS Batch

AWS Batch los trabajos de servicio le permiten enviar solicitudes a AWS los servicios a través de colas de AWS Batch trabajos. Actualmente, AWS Batch admite trabajos SageMaker de formación como trabajos de servicio. A diferencia de los trabajos contenerizados, en los que se AWS Batch gestiona la ejecución subyacente del contenedor, los trabajos de servicio AWS Batch permiten programar los trabajos y hacer colas, mientras que el AWS servicio objetivo (como la SageMaker IA) se encarga de la ejecución real de los trabajos.

AWS Batch for SageMaker Training Jobs permite a los científicos de datos enviar los trabajos de formación con prioridades a colas configurables, lo que garantiza que las cargas de trabajo se ejecuten sin intervención en cuanto los recursos estén disponibles. Esta capacidad aborda desafíos comunes, como la coordinación de los recursos, la prevención de gastos excesivos accidentales, el cumplimiento de las restricciones presupuestarias, la optimización de los costos con instancias reservadas y la eliminación de la necesidad de coordinación manual entre los miembros del equipo.

Los trabajos de servicio se diferencian de los trabajos contenerizados en varios aspectos clave:

- Envío de trabajos: los trabajos de servicio deben enviarse mediante la [SubmitServiceJob](#) API. Los trabajos de servicio no se pueden enviar a través de la consola de AWS Batch .
- Ejecución de trabajos: AWS Batch programa y pone en cola los trabajos de servicio, pero el AWS servicio de destino ejecuta la carga de trabajo real.
- Identificadores de recursos: los trabajos de servicio utilizan la ARNs palabra «trabajo de servicio» en lugar de «trabajo» para distinguirlos de los trabajos en contenedores.

Para empezar a utilizar los trabajos de AWS Batch servicio de formación, consulte. SageMaker [the section called “¿Cómo empezar con AWS Batch la SageMaker IA”](#)

Temas

- [Cargas útiles de trabajos de servicio en AWS Batch](#)
- [Envíe un trabajo de servicio en AWS Batch](#)
- [Asignar el estado del trabajo del AWS Batch servicio al estado de la SageMaker IA](#)
- [Estrategias de reintento de trabajos de servicio en AWS Batch](#)
- [Supervise los trabajos de servicio en una cola AWS Batch](#)
- [Finalización de los trabajos de servicio](#)

Cargas útiles de trabajos de servicio en AWS Batch

Cuando envía trabajos de servicio utilizando [SubmitServiceJob](#), proporciona dos parámetros clave que definen el trabajo: `serviceJobType`, y `serviceRequestPayload`.

- `serviceJobType` Especifica qué AWS servicio ejecutará el trabajo. Para los trabajos de SageMaker formación, este valor es `SAGEMAKER_TRAINING`.
- `serviceRequestPayload` es una cadena codificada en JSON que contiene la solicitud completa que normalmente se envía directamente al servicio de destino. Para los trabajos de SageMaker formación, esta carga útil contiene los mismos parámetros que utilizaría con la [CreateTrainingJob](#) API de SageMaker IA.

Para obtener una lista completa de todos los parámetros disponibles y sus descripciones, consulta la referencia de la [CreateTrainingJob](#) API de SageMaker IA. Todos los parámetros que sean compatibles con `CreateTrainingJob` se pueden incluir en la carga útil del trabajo de servicio.

Para ver más ejemplos de configuraciones de trabajos de formación [APIs, consulte la CLI y SDKs](#) la [Guía para desarrolladores de SageMaker IA](#).

Recomendamos utilizar PySDK para la creación de trabajos de servicio, ya que PySDK tiene clases y utilidades auxiliares. Para ver un ejemplo del uso de PySDK, consulta los [ejemplos de SageMaker IA](#) en GitHub.

Ejemplo de carga útil de un trabajo de servicio

El siguiente ejemplo muestra una carga útil de trabajo de servicio sencilla para un trabajo de SageMaker formación que ejecuta un guion de formación que dice «hola mundo»:

Esta carga útil se pasará como una cadena JSON al parámetro `serviceRequestPayload` cuando se llame `SubmitServiceJob`.

```
{
  "TrainingJobName": "my-simple-training-job",
  "RoleArn": "arn:aws:iam::123456789012:role/SageMakerExecutionRole",
  "AlgorithmSpecification": {
    "TrainingInputMode": "File",
    "TrainingImage": "763104351884.dkr.ecr.us-west-2.amazonaws.com/pytorch-
training:2.0.0-cpu-py310",
    "ContainerEntrypoint": [
      "echo",
```

```
    "hello world"
  ]
},
"ResourceConfig": {
  "InstanceType": "m1.c5.xlarge",
  "InstanceCount": 1,
  "VolumeSizeInGB": 1
},
"OutputDataConfig": {
  "S3OutputPath": "s3://your-output-bucket/output"
},
"StoppingCondition": {
  "MaxRuntimeInSeconds": 30
}
}
```

Envíe un trabajo de servicio en AWS Batch

Para enviar trabajos de servicio a AWS Batch, usa la [SubmitServiceJob](#) API. Puede enviar trabajos mediante el SDK AWS CLI o el SDK.

Si aún no tiene un rol de ejecución, debe crear uno antes de enviar el trabajo de servicio. Para crear la función de ejecución de SageMaker IA, consulte [Cómo utilizar las funciones de ejecución de SageMaker IA](#) en la [guía para desarrolladores de SageMaker IA](#).

Flujo de trabajo para el envío del trabajo de servicio

Cuando envíes un trabajo de servicio, AWS Batch sigue este flujo de trabajo:

1. AWS Batch recibe su [SubmitServiceJob](#) solicitud y valida los parámetros AWS Batch específicos. `serviceRequestPayload` se pasa sin validación.
2. El trabajo entra en el estado SUBMITTED y se coloca en la cola de trabajos especificada
3. AWS Batch evalúa si hay capacidad disponible en el entorno de servicio para los RUNNABLE trabajos que se encuentran al principio de la cola
4. Si hay capacidad disponible, el trabajo se traslada a AI SCHEDULED y se ha transferido a AI SageMaker
5. Cuando se haya adquirido capacidad y la SageMaker IA haya descargado los datos del trabajo de servicio, el trabajo de servicio comenzará a inicializarse y el trabajo cambiará a STARTING.
6. Cuando la SageMaker IA comience a ejecutar el trabajo, su estado cambiará a RUNNING.

7. Mientras la SageMaker IA ejecuta la tarea, AWS Batch supervisa su progreso y asigna los estados del servicio a los estados de las AWS Batch tareas. Para obtener más información sobre cómo se asignan los estados de las tareas de servicio, consulte [???](#).
8. Cuando se completa el trabajo de servicio, pasa a SUCCEEDED y cualquier salida estará lista para descargarse.

Requisitos previos

Antes de enviar un trabajo de servicio, asegúrese de disponer de lo siguiente:

- Entorno de servicio: entorno de servicio que define los límites de capacidad. Para obtener más información, consulte [Cree un entorno de servicios en AWS Batch](#).
- SageMaker cola de trabajos: cola de SageMaker trabajos que permite programar los trabajos. Para obtener más información, consulte [Creación de una cola de trabajos de entrenamiento de SageMaker en AWS Batch](#).
- Permisos de IAM: permisos para crear y administrar colas de trabajos y entornos de servicio de AWS Batch . Para obtener más información, consulte [AWS Batch Políticas, funciones y permisos de IAM](#).

Envíe un trabajo de servicio con la AWS CLI

A continuación, se muestra cómo enviar un trabajo de servicio mediante la AWS CLI:

```
aws batch submit-service-job \
  --job-name "my-sagemaker-training-job" \
  --job-queue "my-sagemaker-job-queue" \
  --service-job-type "SAGEMAKER_TRAINING" \
  --service-request-payload '{"TrainingJobName\": \"sagemaker-training-job-example\", \"AlgorithmSpecification\": {\"TrainingImage\": \"123456789012.dkr.ecr.us-east-1.amazonaws.com/pytorch-inference:1.8.0-cpu-py3\", \"TrainingInputMode\": \"File\", \"ContainerEntrypoint\": [\"sleep\", \"1\"]}, \"RoleArn\": \"arn:aws:iam::123456789012:role/SageMakerExecutionRole\", \"OutputDataConfig\": {\"S3OutputPath\": \"s3://example-bucket/model-output/\"}, \"ResourceConfig\": {\"InstanceType\": \"ml.m5.large\", \"InstanceCount\": 1, \"VolumeSizeInGB\": 1}}'
  --client-token "unique-token-12345"
```

Para obtener más información acerca de los parámetros `serviceRequestPayload`, consulte [the section called “Cargas útiles de trabajos de servicio”](#).

Asignar el estado del trabajo del AWS Batch servicio al estado de la SageMaker IA

Al enviar trabajos a una cola de SageMaker trabajos utilizando [SubmitServiceJob](#), AWS Batch gestiona el ciclo de vida de los trabajos y asigna los estados de los AWS Batch [trabajos a estados](#) de trabajo de SageMaker formación equivalentes. Los trabajos de servicio, como los de SageMaker formación, siguen un ciclo de vida estatal diferente al de los trabajos en contenedores tradicionales. Así como los trabajos de servicio comparten la mayoría de los estados con los trabajos de contenedores, también presentan el estado de SCHEDULED y muestran diferentes comportamientos de reintento, sobre todo al momento de gestionar errores de capacidad insuficiente del servicio de destino.

La siguiente tabla muestra el estado del AWS Batch trabajo y el SageMaker estado/ correspondiente: SecondaryStatus

Estado del lote	SageMaker Estado principal de la IA	SageMaker Estado secundario de IA	Description (Descripción)
SUBMITTED	N/A	N/A	Trabajo enviado a la cola, a la espera de la evaluación del programador.
RUNNABLE	N/A	N/A	El trabajo está en cola y listo para su programación. Los trabajos en este estado se inician tan pronto como haya suficientes recursos disponibles en el entorno de servicio. Los trabajos pueden permanecer en este estado de forma indefinida si los recursos suficientes no están disponibles.
SCHEDULED	InProgress	Pending	El trabajo de servicio se envió correctamente a SageMaker AI
STARTING	InProgress	Downloading	SageMaker Trabajo de formación descargando datos e imágenes. Se adquirió la capacidad de trabajo de

Estado del lote	SageMaker Estado principal de la IA	SageMaker Estado secundario de IA	Description (Descripción)
			entrenamiento y comienza la inicialización del trabajo.
RUNNING	InProgress	Training	SageMaker Algoritmo de ejecución de tareas de entrenamiento
RUNNING	InProgress	Uploading	SageMaker Trabajo de entrenamiento cargando artefactos de salida después de completar el entrenamiento
SUCCEEDED	Completed	Completed	SageMaker El trabajo de formación se completó correctamente. Los artefactos de salida terminaron de cargarse.
FAILED	Failed	Failed	SageMaker El trabajo de formación detectó un error irreparable.
FAILED	Stopped	Stopped	SageMaker El trabajo de entrenamiento se dejó de utilizar manualmente. StopTrainingJob

Estrategias de reintento de trabajos de servicio en AWS Batch

Las estrategias de reintento de trabajos de servicio permiten AWS Batch reintentar automáticamente los trabajos de servicio fallidos en condiciones específicas.

Los trabajos de servicio pueden requerir varios intentos por diversos motivos:

- Problemas de servicio temporales: los errores de servicio internos, las limitaciones o las interrupciones temporales pueden provocar que los trabajos fallen durante el envío o la ejecución.
- Fallos en la inicialización del entrenamiento: los problemas durante el inicio del trabajo, como los problemas de extracción de imágenes o los errores de inicialización, pueden resolverse intentándolo de nuevo.

Con la configuración de las estrategias de reintentos adecuadas, se mejoran las tasas de éxito en el trabajo y se reduce la necesidad de intervención manual, especialmente en el caso de las cargas de trabajo de formación prolongadas.

Note

Los trabajos de servicio reintentan automáticamente ciertos tipos de errores, como los de capacidad insuficiente, sin consumir los reintentos configurados. La estrategia de reintentos se ocupa principalmente de otros tipos de errores, como los errores de algoritmo o los problemas de servicio.

Configuración de estrategias de reintento

Las estrategias de reintentos de los trabajos de servicio se configuran mediante [ServiceJobRetryStrategy](#), lo que admite tanto el recuento simple de reintentos como la lógica de reintentos condicional.

Configuración de reintentos

La estrategia de reintentos más sencilla especifica el número de reintentos que se deben realizar si se produce un error en un trabajo de servicio:

```
{
  "retryStrategy": {
    "attempts": 3
  }
}
```

Esta configuración permite volver a intentar el trabajo de servicio hasta 3 veces si se produce un error.

Important

El valor `attempts` representa la cantidad total de veces que se puede colocar el trabajo en el estado `RUNNABLE`, como el intento inicial. Un valor de 3 significa que, al principio, se intentará realizar el trabajo una vez y, si se produce un error, se volverá a intentar hasta 2 veces más.

Vuelva a intentar la configuración con evaluateOnExit

Puede utilizar el parámetro `evaluateOnExit` para especificar las condiciones en las que se deben volver a intentar los trabajos o permitir que fallen. Esto resulta útil cuando distintos tipos de errores requieren un tratamiento diferente.

La matriz `evaluateOnExit` puede contener hasta 5 estrategias de reintento, cada una de las cuales especifica una acción (RETRY o EXIT) y condiciones según los motivos del estado:

```
{
  "retryStrategy": {
    "attempts": 5,
    "evaluateOnExit": [
      {
        "action": "RETRY",
        "onStatusReason": "Received status from SageMaker: InternalServerError*"
      },
      {
        "action": "EXIT",
        "onStatusReason": "Received status from SageMaker: ValidationException*"
      },
      {
        "action": "EXIT",
        "onStatusReason": "*"
      }
    ]
  }
}
```

Esta configuración:

- Vuelve a intentar los trabajos que fallan debido a errores internos del servidor de la SageMaker IA
- marca como fallidos los trabajos de forma inmediata que encuentran excepciones de validación (errores del cliente que no se resuelven mediante reintentos),
- incluye una regla general para la salida en caso de cualquier otro tipo de fallo.

Patrón de coincidencias de motivos de estado

El parámetro `onStatusReason` admite el patrón de coincidencias con hasta 512 caracteres. Los patrones pueden usar caracteres comodín (*) y coincidir con los motivos de estado devueltos por SageMaker la IA.

En el caso de los trabajos de servicio, los mensajes de estado de la SageMaker IA llevan el prefijo «Estado recibido de SageMaker:» para distinguirlos de los mensajes generados por la AWS Batch IA. Los patrones comunes incluyen lo siguiente:

- `Received status from SageMaker: InternalServerError*`: Coincide con errores internos del servicio
- `Received status from SageMaker: ValidationException*`: Coincide con errores de validación del cliente
- `Received status from SageMaker: ResourceLimitExceeded*`: Coincide con errores del límite de recursos
- `*CapacityError*`: Coincide con los fallos relacionados con la capacidad

Tip

Utilice una coincidencia de patrones específica para gestionar los diferentes tipos de errores de forma adecuada. Por ejemplo, reintentar los errores internos del servidor, pero fallar inmediatamente si se trata de errores de validación que indiquen problemas con los parámetros del trabajo.

Supervise los trabajos de servicio en una cola AWS Batch

Puede supervisar el estado de los trabajos de su cola de trabajos de SageMaker formación utilizando `list-service-jobs`, y `get-job-queue-snapshot`.

Cómo ver los trabajos en ejecución en la cola:

```
aws batch list-service-jobs \  
  --job-queue my-sm-training-fifo-jq \  
  --job-status RUNNING
```

Cómo ver los trabajos en espera en la cola:

```
aws batch list-service-jobs \  
  --job-queue my-sm-training-fifo-jq \  
  --job-status RUNNABLE
```

Vea los trabajos que se han enviado SageMaker pero que aún no están en ejecución:

```
aws batch list-service-jobs \  
  --job-queue my-sm-training-fifo-jq \  
  --job-status SCHEDULED
```

Cómo obtener una instantánea de los trabajos que están al comienzo de su lista:

```
aws batch get-job-queue-snapshot --job-queue my-sm-training-fifo-jq
```

Este comando muestra el orden de los próximos trabajos de servicio en la cola.

Obtención de información detallada sobre el trabajo de servicio

Utilice la operación [DescribeServiceJob](#) para obtener información completa sobre un trabajo de servicio específico, como su estado actual, los identificadores de recursos de servicio y la información detallada sobre los intentos.

Cómo ver la información detallada sobre un trabajo específico:

```
aws batch describe-service-job \  
  --job-id a4d6c728-8ee8-4c65-8e2a-9a5e8f4b7c3d
```

Este comando devuelve información completa sobre el trabajo, como:

- el ARN del trabajo y estado actual,
- Identificadores de recursos de servicio (como el ARN del trabajo de SageMaker formación)
- la configuración de la prioridad de programación y reintentos,
- la carga útil de solicitud de servicio que contiene los parámetros de servicio originales,
- la información detallada sobre los intentos con las horas de inicio y finalización,
- los mensajes de estado del servicio de destino.

SageMaker Supervise los trabajos de formación

Al supervisar los trabajos de SageMaker formación AWS Batch, puede acceder tanto a la información del AWS Batch puesto como a los detalles subyacentes del puesto de SageMaker formación.

El identificador de recursos de servicio en los detalles del trabajo contiene el ARN del trabajo de SageMaker formación:

```
{
  "latestAttempt": {
    "serviceResourceId": {
      "name": "TrainingJobArn",
      "value": "arn:aws:sagemaker:us-east-1:123456789012:training-job/my-training-job"
    }
  }
}
```

Puedes usar este ARN para obtener detalles adicionales directamente de: SageMaker

```
aws sagemaker describe-training-job \
  --training-job-name my-training-job
```

Supervise el progreso del trabajo comprobando tanto AWS Batch el estado como el estado del trabajo de SageMaker formación. El estado del AWS Batch puesto muestra el ciclo de vida general del trabajo, mientras que el estado del puesto de SageMaker formación proporciona detalles específicos del servicio sobre el proceso de formación.

Finalización de los trabajos de servicio

Utilice la operación [TerminateServiceJob](#) para detener un trabajo de servicio en ejecución.

Cómo finalizar un trabajo de servicio específico:

```
aws batch terminate-service-job \
  --job-id a4d6c728-8ee8-4c65-8e2a-9a5e8f4b7c3d \
  --reason "Job terminated by user request"
```

Cuando se termina un trabajo de servicio, AWS Batch se detiene el trabajo y se lo notifica al servicio de destino. En el SageMaker caso de los trabajos de formación, esto también detendrá el trabajo de formación en SageMaker IA.

Estados de trabajo

Al enviar un trabajo a una cola de trabajos, el AWS Batch trabajo entra en ese estado. SUBMITTED A continuación, pasa por los estados siguientes hasta que termina de ejecutarse correctamente (finaliza con el código 0) o no (finaliza con un código distinto de cero). Los trabajos de AWS Batch pueden tener los siguientes estados:

SUBMITTED

Un trabajo que se ha enviado a la cola, y que aún no ha sido evaluado por el programador. El programador evalúa el trabajo para determinar si tiene alguna dependencia pendiente de la correcta finalización de cualquier otro trabajo. Si hay dependencias, el trabajo pasa a PENDING. Si no hay dependencias, el trabajo pasa a RUNNABLE.

PENDING

Un trabajo que está en la cola y que aún no se ha podido ejecutar debido a una dependencia de otro trabajo o recurso. Una vez se cumplan las dependencias, el trabajo pasa a RUNNABLE.

Note

Los padres de los trabajos de matriz se actualizan PENDING cuando se actualiza un trabajo secundario RUNNABLE y permanecen en PENDING su estado mientras los trabajos secundarios se están ejecutando. Para ver estos trabajos, filtre por PENDING estado hasta que todos los trabajos secundarios alcancen un estado terminal.

RUNNABLE

Un trabajo que está en la cola, que no tiene dependencias pendientes y, por tanto, que está listo para ser programado para un host. Los trabajos en este estado se inician tan pronto como haya recursos suficientes disponibles en alguno de los entornos de computación que se asignan a la cola del trabajo. Sin embargo, un trabajo puede permanecer en este estado de forma indefinida si los recursos suficientes no están disponibles.

Note

Si los trabajos no progresan a STARTING, consulte [Trabajos bloqueados en estado RUNNABLE](#) en la sección de resolución de problemas.

STARTING

Estos trabajos se han programado para un host y las operaciones de inicio de contenedor pertinentes están en curso. Después de que la imagen de contenedor se extraiga y el contenedor esté en marcha, el trabajo pasa a RUNNING.

La duración de la extracción de imágenes, de la finalización de Amazon EKS InitContainer y de la resolución de Amazon ECS ContainerDependency ocurren con el estado STARTING. El tiempo que se tarda en extraer una imagen para un trabajo equivale al tiempo que el trabajo mostrará el estado STARTING.

Por ejemplo, si se tardan tres minutos en extraer la imagen de un trabajo, este mostrará el estado STARTING durante tres minutos. Si initContainers tarda un total de diez minutos en completarse, el trabajo de Amazon EKS mostrará STARTING durante diez minutos. Si tiene conjuntos containerDependencies de Amazon ECS en un trabajo de Amazon ECS, este mostrará STARTING hasta que se resuelvan todas las dependencias de contenedor (tiempo de ejecución). STARTING no está incluido en los tiempos de espera; la duración comienza cuando se ejecuta. Para obtener más información, consulte [Estados de trabajo](#).

RUNNING

El trabajo se ejecuta como un trabajo de contenedor en una instancia de contenedor de Amazon ECS dentro de un entorno de computación. Cuando el contenedor del trabajo se cierra, el código del proceso de salida determina si el trabajo ha finalizado correctamente o si ha fallado. Un código de salida 0 indica una correcta ejecución y cualquier código de salida distinto de cero indica error. Si al trabajo asociado a un intento fallido le quedan intentos en su configuración de estrategia de reintento opcional, el trabajo vuelve a pasar al estado RUNNABLE. Para obtener más información, consulte [Reintentos automáticos de trabajo](#).

Note

Los registros de los RUNNING trabajos están disponibles en CloudWatch Registros. El grupo de registros es /aws/batch/job, y el formato del nombre del flujo de registro es el siguiente: *first200CharsOfJobDefinitionName/default/ecs_task_id*. Sin embargo, esto puede cambiar en el futuro.

Cuando un trabajo alcanza el RUNNING estado, puedes recuperar mediante programación su nombre de flujo de registro con la operación de la [DescribeJobs](#) API. Para obtener más información, consulte [Ver los datos de registro enviados a CloudWatch los registros](#) en la Guía del usuario de Amazon CloudWatch Logs. De forma predeterminada, estos registros nunca caducan. Puede modificar el periodo de retención de copia de seguridad. Para obtener más información, consulte [Cambiar la retención de datos de registro en CloudWatch los registros](#) en la Guía del usuario de Amazon CloudWatch Logs.

SUCCEEDED

El trabajo se ha completado correctamente con un código de salida 0. El estado de los SUCCEEDED trabajos se mantiene AWS Batch durante al menos 7 días.

Note

Los registros de los SUCCEEDED trabajos están disponibles en CloudWatch Registros. El grupo de registros es /aws/batch/job, y el formato del nombre del flujo de registro es el siguiente: *first200CharsOfJobDefinitionName/default/ecs_task_id*. Este formato puede cambiar en el futuro.

Cuando un trabajo alcanza el RUNNING estado, puedes recuperar mediante programación su nombre de flujo de registro con la operación de la [DescribeJobsAPI](#). Para obtener más información, consulte [Ver los datos de registro enviados a CloudWatch los registros](#) en la Guía del usuario de Amazon CloudWatch Logs. De forma predeterminada, estos registros nunca caducan. Puede modificar el periodo de retención de copia de seguridad. Para obtener más información, consulte [Cambiar la retención de datos de registro en CloudWatch los registros](#) en la Guía del usuario de Amazon CloudWatch Logs.

FAILED

El trabajo ha fallado en todos los intentos disponibles. El estado de trabajos FAILED persiste en AWS Batch durante al menos 7 días.

Note

Los registros de los FAILED trabajos están disponibles en CloudWatch Logs. El grupo de registros es /aws/batch/job, y el formato del nombre del flujo de registro es el siguiente: *first200CharsOfJobDefinitionName/default/ecs_task_id*. Este formato puede cambiar en el futuro.

Una vez que un trabajo alcanza el RUNNING estado, puedes recuperar su flujo de registros mediante programación con la operación de la [DescribeJobsAPI](#). Para obtener más información, consulte [Ver los datos de registro enviados a CloudWatch los registros](#) en la Guía del usuario de Amazon CloudWatch Logs. De forma predeterminada, estos registros nunca caducan. Puede modificar el periodo de retención de copia de seguridad. Para obtener más información, consulte [Cambiar la retención de datos de registro en CloudWatch los registros](#) en la Guía del usuario de Amazon CloudWatch Logs.

AWS Batch variables de entorno laboral

AWS Batch establece variables de entorno específicas en los trabajos de contenedores. Estas variables de entorno proporcionan una visión introspectiva de los contenedores que se encuentran dentro de los trabajos. Puede usar los valores de estas variables en la lógica de sus aplicaciones. Todas las variables que AWS Batch se configuran comienzan con el `AWS_BATCH_` prefijo. Se trata de un prefijo de variable de entorno protegido. No puede usar este prefijo para sus propias variables en las definiciones o anulaciones de trabajos.

Las variables de entorno siguientes están disponibles en los contenedores de trabajos:

`AWS_BATCH_CE_NAME`

Esta variable se establece en el nombre del entorno de computación en el que está situado el trabajo.

`AWS_BATCH_JOB_ARRAY_INDEX`

Esta variable solo se establece en los trabajos de matrices secundarios. El índice de trabajo de matriz empieza en 0, y cada trabajo secundario recibe un número de índice único. Por ejemplo, un trabajo de matriz con 10 elementos secundarios tiene valores de índice comprendidos entre 0 y 9. Puede utilizar este valor de índice para controlar la forma en la que se diferencian los elementos secundarios del trabajo de matriz. Para obtener más información, consulte [Uso del índice de trabajo de matriz para controlar la diferenciación de trabajos](#).

`AWS_BATCH_JOB_ARRAY_SIZE`

Esta variable se establece en el tamaño del trabajo de matriz principal. El tamaño del trabajo de matriz principal se pasa al trabajo de matriz secundario en esta variable.

`AWS_BATCH_JOB_ATTEMPT`

Esta variable se establece en el número de reintentos de trabajo. Al primer intento se le asigna el valor 1. Para obtener más información, consulte [Reintentos automáticos de trabajo](#).

`AWS_BATCH_JOB_ID`

Esta variable se establece en el ID del AWS Batch trabajo.

`AWS_BATCH_JOB_KUBERNETES_NODE_UID`

Esta variable se establece como el UID de Kubernetes del objeto de nodo que se encuentra en el clúster de Kubernetes en el que se ejecuta el pod. Esta variable solo se establece para

los trabajos que se ejecutan en los recursos de Amazon EKS. Para obtener más información, consulte [UIDs](#) en la documentación de Kubernetes.

`AWS_BATCH_JOB_MAIN_NODE_INDEX`

Esta variable solo se establece en los trabajos paralelos de varios nodos. Esta variable se establece en el número de índice del nodo principal del trabajo. El código de la aplicación puede comparar `AWS_BATCH_JOB_MAIN_NODE_INDEX` con `AWS_BATCH_JOB_NODE_INDEX` en un nodo individual para determinar si es el nodo principal.

`AWS_BATCH_JOB_MAIN_NODE_PRIVATE_IPV4_ADDRESS`

Esta variable solo se establece en los trabajos paralelos de varios nodos. Esta variable no está presente en el nodo principal, pero está configurada en la IPv4 dirección privada del nodo principal del trabajo. El código de la aplicación del nodo secundario puede utilizar esta dirección para comunicarse con el nodo principal.

`AWS_BATCH_JOB_NODE_INDEX`

Esta variable solo se establece en los trabajos paralelos de varios nodos. Esta variable se establece en el número de índice del nodo. El índice del nodo comienza a partir de 0 y cada nodo recibe un número de índice único. Por ejemplo, un trabajo paralelo de varios nodos con 10 elementos secundarios tiene valores de índice comprendidos entre 0 y 9.

`AWS_BATCH_JOB_NUM_NODES`

Esta variable solo se establece en los trabajos paralelos de varios nodos. Esta variable se establece en el número de nodos solicitados para el trabajo en paralelo de varios nodos.

`AWS_BATCH_JQ_NAME`

Esta variable se establece en el nombre de la cola de trabajos a la que se ha enviado el trabajo.

Reintentos automáticos de trabajo

Puede aplicar una estrategia de reintento a los trabajos y las definiciones de trabajo que permita reintentar automáticamente la ejecución de los trabajos con errores. Entre las posibles situaciones de error se incluyen:

- Cualquier código de salida distinto de cero a partir de un trabajo de contenedor
- Una finalización o error de la instancia de Amazon EC2
- Error o AWS interrupción del servicio interno

El acto de enviar un trabajo a una cola de trabajos y que pase al estado RUNNING se considera un intento. De forma predeterminada, a cada trabajo se le concede un intento para pasar al estado SUCCEEDED o FAILED. Sin embargo, la definición de trabajo y los flujos de envíos de trabajo permiten especificar una estrategia de entre 1 y 10 reintentos. Si [evaluateOnExit](#) se especifica, puede contener hasta 5 estrategias de reintentos. Si [evaluateOnExit](#) se especifica, pero ninguna de las estrategias de reintento coincide, se vuelve a intentar el trabajo. En el caso de los trabajos que no coincidan con la salida, añada una última entrada que salga por cualquier motivo. Por ejemplo, este objeto `evaluateOnExit` tiene dos entradas con acciones de RETRY y una última entrada con una acción de EXIT.

```
"evaluateOnExit": [  
  {  
    "action": "RETRY",  
    "onReason": "AGENT"  
  },  
  {  
    "action": "RETRY",  
    "onStatusReason": "Task failed to start"  
  },  
  {  
    "action": "EXIT",  
    "onReason": "*"   
  }  
]
```

En la ejecución, la variable de entorno `AWS_BATCH_JOB_ATTEMPT` se ajusta a la cantidad de reintentos de trabajo de contenedor correspondiente. Al primer intento se le asigna el 1, y los intentos posteriores se enumeran en orden ascendente (por ejemplo, 2, 3, 4, etc.).

Por ejemplo, supongamos que un intento de trabajo falla por cualquier motivo y que el número de intentos especificado en la configuración de reintentos es superior al número de `AWS_BATCH_JOB_ATTEMPT`. A continuación, el trabajo se vuelve a colocar en el estado RUNNABLE. Para obtener más información, consulte [Estados de trabajo](#).

 Note

Los trabajos que se cancelan o se terminan no se reintentan. De la misma manera, los trabajos que fallan debido a una definición de trabajo no válida tampoco se reintentan.

Para obtener más información, consulte [Estrategia de reintento](#), [Creación de una definición de trabajo de un solo nodo](#), [Tutorial: envío de un trabajo](#) y [Códigos de error de tareas detenidas](#).

Dependencias de trabajos

Al enviar un AWS Batch trabajo, puede especificar el trabajo del IDs que depende. Al hacerlo, el AWS Batch programador se asegura de que el trabajo se ejecute solo después de que las dependencias especificadas se hayan completado correctamente. Una vez que estas terminen correctamente, el trabajo dependiente pasa del estado PENDING al estado RUNNABLE y, a continuación, a STARTING y a RUNNING. Si alguna de las dependencias del trabajo produce un error, el trabajo dependiente pasa automáticamente de PENDING a FAILED.

Por ejemplo, Job A puede expresar una dependencia de hasta 20 trabajos distintos que deben completarse correctamente para que pueda ejecutarse. A continuación, puede enviar trabajos adicionales que dependan de Job A y de hasta otros 19 trabajos.

En los trabajos de matrices, puede especificar una dependencia de tipo SEQUENTIAL sin especificar un ID de trabajo para que cada trabajo de matriz secundario se complete de forma secuencial, comenzando a partir del índice 0. También puede especificar una dependencia de tipo N_TO_N con un ID de trabajo. De esta forma, cada índice secundario de este trabajo debe esperar a que se complete el índice secundario correspondiente de cada dependencia antes de comenzar. Para obtener más información, consulte [Trabajos de matrices](#).

Para enviar un AWS Batch trabajo con dependencias, consulte. [Tutorial: envío de un trabajo](#)

[Programación basada en los recursos](#) permite programar los trabajos según los recursos consumibles requeridos para su ejecución. Estos recursos consumibles se deben especificar para ejecutar un trabajo, por lo tanto, Batch contempla dichas dependencias de recursos cuando realiza la programación de uno. La reducción de la infrautilización de los recursos de computación se lleva a cabo con la asignación de los trabajos que contienen todos los recursos necesarios. La programación basada en los recursos está disponible para las políticas de programación FIFO y de reparto justo, y se puede utilizar con todas las plataformas de computación compatibles con Batch, como EKS, ECS y Fargate. Así como con trabajos de matriz, trabajos paralelos de varios nodos (MNP) y trabajos habituales de Batch.

Tiempos de espera de trabajo

Puede configurar una duración de tiempo de espera para sus trabajos de modo que si un trabajo se ejecuta durante más tiempo, AWS Batch lo finaliza. Por ejemplo, es posible que tenga un trabajo

que sabe que solo debería tardar 15 minutos en completarse. A veces la aplicación se bloquea en un bucle y se ejecuta para siempre, por lo que puede establecer un tiempo de espera de 30 minutos para terminar el trabajo bloqueado.

 Important

De forma predeterminada, AWS Batch no tiene un tiempo de espera para el trabajo. Si no define un tiempo de espera para el trabajo, el trabajo se ejecutará hasta que salga el contenedor.

Especifique un parámetro `attemptDurationSeconds`, que debe tener al menos de 60 segundos, en la definición del trabajo o al enviarlo. Cuando haya transcurrido este número de segundos después de la `startedAt` marca de tiempo del intento de trabajo, AWS Batch finaliza el trabajo. En el recurso de computación, el contenedor del trabajo recibe una señal `SIGTERM` para dar a la aplicación la posibilidad de que se apague correctamente. Si el contenedor se sigue ejecutando al cabo de 30 segundos, se envía una señal `SIGKILL` para forzar su cierre.

Las finalizaciones por haberse agotado el tiempo de espera se realizan en la medida que es posible. No espere que se produzcan exactamente en el momento en que se agota el tiempo de espera del intento de trabajo (pueden tardar algunos segundos más). Si su aplicación necesita una ejecución de tiempo de espera precisa, debe implementar esa lógica en la aplicación. Si tiene una gran cantidad de trabajos cuyo tiempo de espera se agota simultáneamente, las terminaciones por tiempo de espera se deben comportar como una cola del tipo "primero en entrar, primero en salir", donde los trabajos se terminen por lotes.

 Note

No hay un valor de tiempo de espera máximo para un trabajo. AWS Batch

Si un trabajo se termina por superar la duración del tiempo de espera, no se vuelve a intentar. Si se produce un error en un intento de trabajo, se puede reintentar si se han habilitado los reintentos y la cuenta atrás del tiempo de espera comienza para el nuevo intento.

 Important

Los trabajos que se ejecutan con recursos Fargate no pueden esperar funcionar durante más de 14 días. Si el tiempo de espera supera los 14 días, es posible que los recursos de Fargate ya no estén disponibles y el trabajo se cancelará.

En el caso de los trabajos de matriz, los trabajos secundarios tienen la misma configuración de tiempo de espera que el trabajo principal.

Para obtener información sobre el envío de un AWS Batch trabajo con una configuración de tiempo de espera, consulte. [Tutorial: envío de un trabajo](#)

Trabajos de Amazon EKS

Un trabajo es la unidad de trabajo más pequeña de AWS Batch. Un AWS Batch trabajo en Amazon EKS consiste en one-to-one mapear un Kubernetes pod. Una definición de AWS Batch trabajo es una plantilla para un AWS Batch trabajo. Cuando envía un AWS Batch trabajo, hace referencia a una definición de trabajo, selecciona una cola de trabajos y proporciona un nombre para el trabajo. En la definición de trabajo de un AWS Batch trabajo en Amazon EKS, el parámetro [eksProperties](#) define el conjunto de parámetros que admite un trabajo en AWS Batch Amazon EKS. En una [SubmitJob](#) solicitud, el [eksPropertiesOverride](#) parámetro permite anular algunos parámetros comunes. De esta forma, puede utilizar plantillas de definiciones de trabajos para varios trabajos. Cuando se envía un trabajo a su clúster de Amazon EKS, lo AWS Batch transforma en un podspec (Kind: Pod). podspecUtiliza algunos AWS Batch parámetros adicionales para garantizar que los trabajos se escalen y programen correctamente. AWS Batch combina etiquetas e imitaciones para garantizar que los trabajos se ejecuten únicamente en los nodos AWS Batch gestionados y que otros módulos no se ejecuten en esos nodos.

 Important

- Si el `hostNetwork` parámetro no está establecido de forma explícita en una definición de trabajo de Amazon EKS, el modo de red del pod AWS Batch pasa por defecto al modo `host`. Más específicamente, se aplican los siguientes ajustes: `hostNetwork=true` y `dnsPolicy=ClusterFirstWithHostNet`.
- AWS Batch limpia los módulos de trabajos poco después de que un pod complete su trabajo. Para ver los registros de las aplicaciones del pod, configure un servicio de registro

para su clúster. Para obtener más información, consulte [Utilice CloudWatch los registros para supervisar AWS Batch los trabajos de Amazon EKS](#).

Temas

- [Tutorial: asignación de un trabajo en ejecución a un pod y un nodo](#)
- [Tutorial: reasignación de un pod en ejecución a su función](#)

Tutorial: asignación de un trabajo en ejecución a un pod y un nodo

Los `podProperties` de un trabajo en ejecución tienen los parámetros `podName` y `nodeName` establecidos para el intento de trabajo actual. Utilice la operación [DescribeJobs](#) de la API para ver estos parámetros.

A continuación, se muestra un ejemplo del resultado.

```
$ aws batch describe-jobs --job 2d044787-c663-4ce6-a6fe-f2baf7e51b04
{
  "jobs": [
    {
      "status": "RUNNING",
      "jobArn": "arn:aws:batch:us-east-1:123456789012:job/2d044787-c663-4ce6-a6fe-f2baf7e51b04",
      "jobDefinition": "arn:aws:batch:us-east-1:123456789012:job-definition/MyJobOnEks_SleepWithRequestsOnly:1",
      "jobQueue": "arn:aws:batch:us-east-1:123456789012:job-queue/My-Eks-JQ1",
      "jobId": "2d044787-c663-4ce6-a6fe-f2baf7e51b04",
      "eksProperties": {
        "podProperties": {
          "nodeName": "ip-192-168-55-175.ec2.internal",
          "containers": [
            {
              "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
              "resources": {
                "requests": {
                  "cpu": "1",
                  "memory": "1024Mi"
                }
              }
            }
          ]
        }
      }
    }
  ]
}
```

```

    ],
    "podName": "aws-batch.b0aca953-ba8f-3791-83e2-ed13af39428c"
  }
}
]
}

```

En el caso de un trabajo con los reintentos habilitados, el podName y nodeName de cada intento completado aparece en el parámetro de eksAttempts lista de la operación de [DescribeJobs](#) API. El podName y el nodeName del intento en ejecución actual se encuentra en el objeto podProperties.

Tutorial: reasignación de un pod en ejecución a su función

Un pod tiene etiquetas que indican la dirección jobId y el entorno uuid de cómputo al que pertenece. AWS Batch inyecta variables de entorno para que el tiempo de ejecución del trabajo pueda hacer referencia a la información del trabajo. Para obtener más información, consulte [AWS Batch variables de entorno laboral](#). Puede ver esta información al ejecutar el siguiente comando. El resultado es el siguiente.

```

$ kubectl describe pod aws-batch.14638eb9-d218-372d-ba5c-1c9ab9c7f2a1 -n my-aws-batch-namespace
Name:          aws-batch.14638eb9-d218-372d-ba5c-1c9ab9c7f2a1
Namespace:     my-aws-batch-namespace
Priority:       0
Node:          ip-192-168-45-88.ec2.internal/192.168.45.88
Start Time:    Wed, 26 Oct 2022 00:30:48 +0000
Labels:        batch.amazonaws.com/compute-environment-uuid=5c19160b-d450-31c9-8454-86cf5b30548f
               batch.amazonaws.com/job-id=f980f2cf-6309-4c77-a2b2-d83fbba0e9f0
               batch.amazonaws.com/node-uid=a4be5c1d-9881-4524-b967-587789094647
...
Status:        Running
IP:            192.168.45.88
IPs:
  IP: 192.168.45.88
Containers:
  default:
    Image:      public.ecr.aws/amazonlinux/amazonlinux:2
    ...
  Environment:
    AWS_BATCH_JOB_KUBERNETES_NODE_UID:  a4be5c1d-9881-4524-b967-587789094647

```


AWS_BATCH_JOB_ID:	f980f2cf-6309-4c77-a2b2-d83fbbba0e9f0
AWS_BATCH_JOB_NAME:	My-Eks-JQ1
AWS_BATCH_JOB_ATTEMPT:	1
AWS_BATCH_CE_NAME:	My-Eks-CE1

...

Características compatibles con AWS Batch Amazon EKS Jobs

Estas son las características AWS Batch específicas que también son comunes a los Kubernetes trabajos que se ejecutan en Amazon EKS:

- [Dependencias de trabajos](#)
- [Trabajos de matrices](#)
- [Tiempos de espera de trabajo](#)
- [Reintentos automáticos de trabajo](#)
- [Uso de una programación de reparto justo para ayudar a programar los trabajos](#)

Kubernetes y **SecretsServiceAccounts**

AWS Batch admite referencias Kubernetes Secrets y. ServiceAccounts Puede configurar los pods para utilizar los roles de IAM de Amazon EKS para las cuentas de servicio. Para obtener más información, consulte [Configurar los pods para usar una cuenta de servicio de Kubernetes](#) en la [Guía del usuario de Amazon EKS](#).

Documentos relacionados

- [Consideraciones sobre memoria y vCPU para AWS Batch en Amazon EKS](#)
- [Ejecución de trabajos de GPU](#)
- [Trabajos bloqueados en estado RUNNABLE](#)

Trabajos paralelos de varios nodos

Los trabajos paralelos de varios nodos le permiten ejecutar trabajos individuales que abarcan varias instancias de Amazon EC2. Los trabajos paralelos de varios nodos de AWS Batch (también denominados programación por grupos) le permiten ejecutar aplicaciones informáticas de alto

rendimiento a gran escala estrechamente acopladas y entrenar modelos de GPU distribuida sin necesidad de lanzar, configurar y administrar los recursos de Amazon EC2 directamente. Un trabajo paralelo de AWS Batch varios nodos es compatible con cualquier marco que admita la comunicación entre nodos basada en IP. Algunos ejemplos son Apache MXNet TensorFlow, Caffe2 o Message Passing Interface (MPI).

Los trabajos paralelos de varios nodos se envían como un único trabajo. Sin embargo, la definición de trabajo (o las anulaciones de nodos de envío de trabajos) especifica el número de nodos que se deben crear para el trabajo y qué grupos de nodos deben crearse. Cada trabajo paralelo de varios nodos contiene un nodo principal, que se lanza en primer lugar. Una vez que el nodo principal está en marcha, se lanzan e inician los nodos secundarios. El trabajo finaliza solo si sale el nodo principal. A continuación, se detienen todos los nodos secundarios. Para obtener más información, consulte [Grupos de nodos](#).

Los nodos de trabajo paralelos de varios nodos son de un solo inquilino. Esto significa que solo se ejecuta un contenedor de trabajos en cada instancia de Amazon EC2.

El estado final del trabajo (SUCCEEDED o FAILED) lo determina el estado final del trabajo del nodo principal. Para obtener el estado de un trabajo paralelo de varios nodos, puede describir el trabajo utilizando el ID de trabajo obtenido al enviar el trabajo. Si necesita los detalles de los nodos secundarios, deberá describir cada nodo secundario por separado. Puede direccionar los nodos mediante la notación `#N` (empezando por 0). Por ejemplo, para acceder a los detalles del segundo nodo de un trabajo, describe `aws_batch_job_id #1` mediante la operación de la AWS Batch [DescribeJobs](#) API. La información `started`, `stoppedAt`, `statusReason` y `exit` de un trabajo paralelo de varios nodos se rellena desde el nodo principal.

Si especifica los reintentos de trabajo, un error en el nodo principal provocará otro intento. Los errores en los nodos secundarios no provocan que se produzcan más intentos. Cada nuevo intento de un trabajo paralelo de varios nodos actualiza el intento correspondiente de sus nodos secundarios asociados.

Para ejecutar trabajos paralelos de varios nodos AWS Batch, el código de la aplicación debe contener los marcos y las bibliotecas necesarios para la comunicación distribuida.

Temas

- [Variables de entorno](#)
- [Grupos de nodos](#)
- [Ciclo de vida laboral para trabajos de MNP](#)

- [Consideraciones sobre el entorno de cómputo para MNP con AWS Batch](#)

Variables de entorno

En tiempo de ejecución, cada nodo tiene configuradas las variables de entorno estándar que reciben todos los AWS Batch trabajos. Además, los nodos se configuran con las siguientes variables de entorno que son específicas para los trabajos paralelos de varios nodos:

`AWS_BATCH_JOB_MAIN_NODE_INDEX`

Esta variable se establece en el número de índice del nodo principal del trabajo. El código de la aplicación puede comparar `AWS_BATCH_JOB_MAIN_NODE_INDEX` con `AWS_BATCH_JOB_NODE_INDEX` en un nodo individual para determinar si es el nodo principal.

`AWS_BATCH_JOB_MAIN_NODE_PRIVATE_IPV4_ADDRESS`

Esta variable solo se establece en los trabajos paralelos de varios nodos. Esta variable no está presente en el nodo principal. Esta variable se establece en la IPv4 dirección privada del nodo principal del trabajo. El código de la aplicación del nodo secundario puede utilizar esta dirección para comunicarse con el nodo principal.

`AWS_BATCH_JOB_NODE_INDEX`

Esta variable se establece en el número de índice del nodo. El índice del nodo comienza a partir de 0 y cada nodo recibe un número de índice único. Por ejemplo, un trabajo paralelo de varios nodos con 10 elementos secundarios tiene valores de índice comprendidos entre 0 y 9.

`AWS_BATCH_JOB_NUM_NODES`

Esta variable se establece en el número de nodos solicitados para el trabajo en paralelo de varios nodos.

Grupos de nodos

Un grupo de nodos es un conjunto de nodos de trabajo idénticos que comparten las mismas propiedades de contenedor. Se puede utilizar AWS Batch para especificar hasta cinco grupos de nodos distintos para cada trabajo.

Cada grupo puede tener sus propias imágenes de contenedor, comandos, variables de entorno, etc. Por ejemplo, puede enviar un trabajo que requiera una sola instancia `c5.xlarge` para el nodo

principal y cinco nodos secundarios de la instancia `c5.xlarge`. Cada uno de estos grupos de nodos distintos puede especificar diferentes imágenes de contenedor o comandos para ejecutarlos en cada trabajo.

Como alternativa, todos los nodos de su trabajo pueden usar un único grupo de nodos. Además, el código de su aplicación puede diferenciar las funciones de los nodos, como el nodo principal y el nodo secundario. Para ello, compara la variable de entorno `AWS_BATCH_JOB_MAIN_NODE_INDEX` con su propio valor para `AWS_BATCH_JOB_NODE_INDEX`. Puede tener un máximo de 1000 nodos en un solo trabajo. Este es el límite predeterminado de las instancias de un clúster de Amazon ECS. Puede [solicitar un aumento de este límite](#).

Note

Actualmente, todos los grupos de nodos de un trabajo paralelo de varios nodos deben usar el mismo tipo de instancia.

Ciclo de vida laboral para trabajos de MNP

Al enviar un trabajo paralelo de varios nodos, el trabajo entra en el estado `SUBMITTED`. A continuación, el trabajo espera a que finalicen todas las dependencias del trabajo. El trabajo también pasa al estado `RUNNABLE`. Por último, AWS Batch aprovisiona la capacidad de instancia necesaria para ejecutar el trabajo y lanza estas instancias.

Cada trabajo paralelo de varios nodos contiene un nodo principal. El nodo principal es una subtaska única que AWS Batch supervisa para determinar el resultado del trabajo de varios nodos enviado. El nodo principal se lanza en primer lugar y pasa a tener el estado `STARTING`. El valor de tiempo de espera especificado en el parámetro `attemptDurationSeconds` se aplica a todo el trabajo y no a los nodos.

Cuando el nodo principal alcanza el estado `RUNNING` (después de que el contenedor del nodo se esté ejecutando), los nodos secundarios se lanzan y pasan al estado `STARTING`. Los nodos secundarios aparecen en orden aleatorio. No hay ninguna garantía sobre la sincronización o el orden del lanzamiento de los nodos secundarios. Para asegurarse de que todos los nodos de los trabajos están en el estado `RUNNING` después de que el contenedor del nodo se está ejecutando, el código de la aplicación puede consultar la API de AWS Batch para obtener información sobre el nodo principal y los nodos secundarios. Como alternativa, el código de la aplicación puede esperar hasta que todos los nodos estén en línea antes de iniciar cualquier tarea de procesamiento

distribuido. La dirección IP privada del nodo principal está disponible en la variable de entorno `AWS_BATCH_JOB_MAIN_NODE_PRIVATE_IPV4_ADDRESS` de cada nodo secundario. El código de la aplicación puede utilizar esta información para coordinar y comunicar datos entre cada una de las tareas.

A medida que finalizan los nodos individuales, pasan al estado `SUCCEEDED` o `FAILED`, en función de su código de salida. Si finaliza el nodo principal, se considera que el trabajo ha finalizado y todos los nodos secundarios se detienen. Si un nodo secundario muere, AWS Batch no realiza ninguna acción en los demás nodos del trabajo. Si no desea que el trabajo continúe con una cantidad reducida de nodos, debe tenerlo en cuenta en el código de la aplicación. De este modo, se termina o se cancela el trabajo.

Consideraciones sobre el entorno de cómputo para MNP con AWS Batch

Hay varios aspectos que es preciso tener en cuenta al configurar entornos de computación para ejecutar trabajos paralelos de varios nodos con AWS Batch.

- Los trabajos paralelos de varios nodos no se admiten en entornos de computación `UNMANAGED`.
- Si va a enviar trabajos paralelos de varios nodos a un entorno de computación, considere la posibilidad de crear un grupo con ubicación en clúster en una única zona de disponibilidad y asociarlo a los recursos de computación. Esto mantiene los trabajos paralelos de varios nodos en una agrupación lógica de instancias muy próxima al alto potencial de flujo de la red. Para obtener más información, consulte [Grupos de ubicación](#) en la Guía del usuario de Amazon EC2.
- Los trabajos paralelos de varios nodos no se admiten en entornos de computación que utilicen instancias de spot.
- AWS Batch los trabajos paralelos de varios nodos utilizan el modo de `awsvpc` red Amazon ECS, que proporciona a los contenedores de trabajos paralelos de varios nodos las mismas propiedades de red que las instancias de Amazon EC2. Cada contenedor de trabajos paralelos de varios nodos obtiene su propia interfaz de red elástica, una dirección IP privada principal y un nombre de host DNS interno. La interfaz de red se crea en la misma subred de VPC como su recurso de computación de host.
- El entorno de computación no puede tener asociados más de cinco grupos de seguridad. Las interfaces de red elásticas que se crean y se adjuntan a una tarea de MNP utilizarán los grupos de seguridad especificados en el entorno de computación. Si no desea especificar un grupo de seguridad, se utilizará el predeterminado de la VPC.
- El modo de red `awsvpc` no proporciona las interfaces de red elásticas para los trabajos paralelos de varios nodos con direcciones IP públicas. Para obtener acceso a Internet, los recursos de

computación deben lanzarse en una subred privada configurada para utilizar una puerta de enlace NAT. Para obtener más información, consulte [Gateways NAT](#) en la Guía del usuario de Amazon VPC. La comunicación entre nodos debe utilizar la dirección IP privada o el nombre de host DNS para el nodo. Los trabajos paralelos de varios nodos que se ejecutan en recursos de computación dentro de subredes públicas no disponen de acceso de red saliente. Para crear una VPC con subredes privadas y una gateway NAT, consulte [Creación de una nube virtual privada](#).

- Las interfaces de red elásticas que se crean y asocian a los recursos de computación no se pueden separar manualmente ni la cuenta puede modificarlas. De este modo, se evita la eliminación accidental de una interfaz de red elástica que esté asociada a un trabajo en ejecución. Para liberar las interfaces de red elásticas de una tarea, termine el trabajo.
- Su entorno de cómputo debe tener un máximo de v suficiente CPUs para soportar su trabajo paralelo de varios nodos.
- Su cuota de instancias de Amazon EC2 incluye la cantidad de instancias necesarias para ejecutar su trabajo. Por ejemplo, si el trabajo requiere 30 instancias, pero la cuenta solo puede ejecutar 20 en una región. Entonces, su trabajo se quedará estancado en el estado RUNNABLE.
- Si especifica un tipo de instancia para un grupo de nodos en un trabajo paralelo de varios nodos, el entorno de computación debe ser capaz de lanzar ese tipo de instancia.

Trabajos paralelos de varios nodos en Amazon EKS

Puede utilizarlos AWS Batch en Amazon Elastic Kubernetes Service para ejecutar trabajos paralelos (MNP) de varios nodos (también conocidos como programación de bandas) en los clústeres gestionados. Kubernetes Esta opción se suele utilizar para trabajos grandes, estrechamente acoplados y de alto rendimiento que no se pueden ejecutar en una sola instancia de Amazon Elastic Compute Cloud. Para obtener más información, consulte [Trabajos paralelos de varios nodos](#).

Puede utilizar esta característica para ejecutar aplicaciones informáticas de alto rendimiento específicas de Kubernetes y gestionadas por Amazon EKS, entrenamiento del modelo lingüístico de gran tamaño y otros trabajos de inteligencia artificial (IA) o machine learning (ML).

Temas

- [Ejecución de trabajos de MNP](#)
- [Creación de una definición de trabajo de MNP de Amazon EKS](#)
- [Envío de un trabajo de Amazon EKS MNP](#)
- [Anulación de una definición de trabajo de MNP de Amazon EKS](#)

Ejecución de trabajos de MNP

AWS Batch admite trabajos de MNP en Amazon Elastic Container Service y Amazon EKS mediante Amazon EC2. A continuación, se proporcionan más detalles sobre los parámetros de instancia y contenedor de la característica.

Cuotas de instancias para MNP en Amazon EKS

- Se pueden utilizar hasta 1000 instancias para un solo trabajo de MNP.
- Se pueden unir hasta 5000 instancias a un único clúster de Amazon EKS.
- Se pueden agrupar en clústeres y adjuntar hasta 5 entornos de computación a una cola de trabajos.

Por ejemplo, puede escalar verticalmente hasta 5 entornos de computación en clústeres agrupados en una cola de trabajos y 1000 instancias en cada uno de ellos.

Además de los parámetros de la instancia, tenga en cuenta que no puede usar Fargate para trabajos de MNP a través de ninguno de los dos servicios.

Solo puede utilizar un tipo de instancia en cada trabajo de MNP. Para cambiar el tipo de instancia, actualice el entorno de computación o defina uno nuevo. También puede especificar el tipo de instancia y proporcionar los requisitos de vCPU o memoria al crear la definición del trabajo.

Cuotas de contenedor para MNP en Amazon EKS

- Un trabajo paralelo de varios nodos admite un pod por nodo.
- Hasta 10 contenedores (o 10 contenedores init). Para obtener más información, consulte [Init Containers](#) (en la documentación de Kubernetes) en cada pod.
- Hasta 5 rangos de nodos en cada trabajo de MNP.
- Hasta 10 imágenes de contenedores distintas en cada rango de nodos.

Por ejemplo, puede ejecutar hasta un máximo de 10 000 contenedores en un solo trabajo de MNP que contenga 5 rangos de nodos y un total de 50 imágenes únicas.

Ejecución de trabajos de MNP en una VPC de Amazon privada y un clúster de Amazon EKS

Los trabajos de MNP se pueden ejecutar en cualquier clúster de Amazon EKS, tanto si tiene Internet público como si no. Cuando utilice un clúster de Amazon EKS con acceso exclusivo a la red privada, asegúrese de que AWS Batch puede acceder al plano de control de Amazon EKS y al servidor de Kubernetes API gestionado. Puede conceder el acceso necesario a través de los puntos de conexión de Amazon Virtual Private Cloud. Para obtener más información, consulte [Configurar un servicio de punto de conexión](#).

Los pods de clústeres de Amazon EKS no pueden descargar una imagen de una fuente pública porque la VPC privada no tiene acceso a Internet. El clúster de Amazon EKS debe extraer imágenes de un registro de contenedores que esté en su Amazon VPC. Puede crear un [Amazon Elastic Container Registry \(Amazon ECR\)](#) en su Amazon VPC y copiar las imágenes del contenedor para acceder a los nodos.

También puede crear una regla de caché de extracción con Amazon ECR. Una vez creada una regla de caché de extracción para un registro público externo, basta con extraer una imagen de ese registro público externo mediante la URI de su registro privado de Amazon ECR. A continuación, Amazon ECR crea un repositorio y almacena la imagen en la caché. Cuando se extrae una imagen en caché mediante el URI del registro privado de Amazon ECR, Amazon ECR verifica el registro remoto para ver si hay una nueva versión de la imagen y actualizará su registro privado hasta una vez cada 24 horas. Para obtener más información, consulte [Creación de una regla de caché de extracción en Amazon ECR](#).

Notificación de error

Si tus trabajos de MNP están bloqueados, puedes recibir notificaciones a través de Amazon Consola de administración de AWS y Amazon EventBridge. Por ejemplo, si un trabajo de MNP está atascado en primer lugar en la cola, puede recibir una notificación sobre el problema junto con información sobre la causa para tomar medidas inmediatas y desbloquear la cola de trabajos. También puede finalizar automáticamente el trabajo de MNP si no se realiza ninguna acción en un período de tiempo determinado, que se puede definir en la plantilla de cola de trabajos. Para obtener más información, consulte [Eventos bloqueados en la cola de trabajos](#)

Creación de una definición de trabajo de MNP de Amazon EKS

Para definir y ejecutar trabajos de MNP en Amazon EKS, hay nuevos parámetros en las operaciones de API de [SubmitJob](#) y [RegisterJobDefinition](#).

- Use [eksProperties](#) en la sección [nodeProperties](#) para establecer una definición del trabajo de MNP.
- Use [eksPropertiesOverride](#) en la sección [nodePropertyOverrides](#) para anular los parámetros establecidos en la definición del trabajo al enviar un trabajo de MNP.

Estas acciones se pueden definir mediante las operaciones de API y Consola de administración de AWS.

Referencia: registre la carga útil de la solicitud de definición de trabajo de Amazon EKS MNP

En el siguiente ejemplo, se muestra cómo puede registrar una definición de trabajo de Amazon EKS MNP con dos nodos.

```
{
  "jobDefinitionName": "MyEksMnpJobDefinition",
  "type": "multinode",
  "nodeProperties": {
    "numNodes": 2,
    "mainNode": 0,
    "nodeRangeProperties": [
      {
        "targetNodes" : "0:",
        "eksProperties": {
          "podProperties": {
            "containers": [
              {
                "name": "test-eks-container-1",
                "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
                "command": [
                  "sleep",
                  "60"
                ],
                "resources": {
                  "limits": {
                    "cpu": "1",
                    "memory": "1024Mi"
                  }
                },
                "securityContext": {
                  "runAsUser": 1000,
```

```
        "runAsGroup":3000,
        "privileged":true,
        "readOnlyRootFilesystem":true,
        "runAsNonRoot":true
    }
}
],
"initContainers": [
    {
        "name":"init-ekscontainer",
        "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
        "command": [
            "echo",
            "helloWorld"
        ],
        "resources": {
            "limits": {
                "cpu": "1",
                "memory": "1024Mi"
            }
        }
    }
],
"metadata": {
    "labels": {
        "environment" : "test"
    }
}
}
}
}
```

Para registrar la definición de trabajo mediante el AWS CLI, copia la definición en un archivo local denominado `MyEksMnpJobDefinition.json` y ejecuta el siguiente comando.

```
aws batch register-job-definition --cli-input-json file://MyEksMnpJobDefinition.json
```

Recibirá la siguiente respuesta JSON.

 $\{$

```
{
  "jobDefinitionName": "MyEksMnpJobDefinition",
  "jobDefinitionArn": "arn:aws:batch:us-east-1:0123456789:job-definition/
MyEksMnpJobDefinition:1",
  "revision": 1
}
```

Envío de un trabajo de Amazon EKS MNP

Para enviar un trabajo con la definición de trabajo registrada, introduzca el siguiente comando. Sustituya el valor de `<EKS_JOB_QUEUE_NAME>` por el nombre o el ARN de una cola de trabajos preexistente asociada a un entorno de computación de Amazon EKS.

```
aws batch submit-job --job-queue <EKS_JOB_QUEUE_NAME> \
  --job-definition MyEksMnpJobDefinition \
  --job-name myFirstEksMnpJob
```

Recibirá la siguiente respuesta JSON.

```
{
  "jobArn": "arn:aws:batch:region:account:job/9b979cce-9da0-446d-90e2-ffa16d52af68",
  "jobName": "myFirstEksMnpJob",
  "jobId": "<JOB_ID>"
}
```

Puede comprobar el estado del trabajo con el ID de trabajo devuelto mediante el siguiente comando.

```
aws batch describe-jobs --jobs <JOB_ID>
```

Anulación de una definición de trabajo de MNP de Amazon EKS

Si lo desea, puede anular los detalles de la definición del trabajo (por ejemplo, cambiar el tamaño del trabajo de MNP o los detalles del trabajo secundario). A continuación, se proporciona un ejemplo de carga útil de solicitud de JSON para enviar un trabajo MNP de cinco nodos y cambios en el comando del contenedor `test-eks-container-1`.

```
{
  "numNodes": 5,
  "nodePropertyOverrides": [
    {
      "targetNodes": "0:",
```

```
"eksPropertiesOverride": {
  "podProperties": {
    "containers": [
      {
        "name": "test-eks-container-1",
        "command": [
          "sleep",
          "150"
        ]
      }
    ]
  }
}
```

Para enviar un trabajo con estas anulaciones, guarda el ejemplo en un archivo local, `eks-mnp-job-nodeoverride.json`, y úsalo AWS CLI para enviar el trabajo con las anulaciones.

Trabajos de matrices

Un trabajo de matriz es un trabajo que comparte parámetros comunes, como la definición del trabajo CPUs, v y la memoria. Se ejecuta como un conjunto de trabajos básicos relacionados, pero independientes, que se pueden distribuir en varios hosts y pueden ejecutarse de forma simultánea. Los trabajos de matrices son la manera más eficiente de ejecutar trabajos que ocurren todos en paralelo, como simulaciones Monte Carlo, barridos paramétricos o grandes trabajos de representación.

AWS Batch Los trabajos de matriz se envían igual que los trabajos normales. Sin embargo, es necesario especificar un tamaño de matriz (entre 2 y 10 000) para definir la cantidad de trabajos secundarios que deberían ejecutarse en la matriz. Si envía un trabajo con un tamaño de matriz de 1 000, se ejecuta un solo trabajo que genera 1 000 trabajos secundarios. El trabajo de matriz es una referencia o un puntero para administrar todos los trabajos secundarios. Esto permite enviar grandes cargas de trabajo con una sola consulta. El tiempo de espera especificado en el parámetro `attemptDurationSeconds` se aplica a cada trabajo secundario. El trabajo de matriz principal no tiene tiempo de espera.

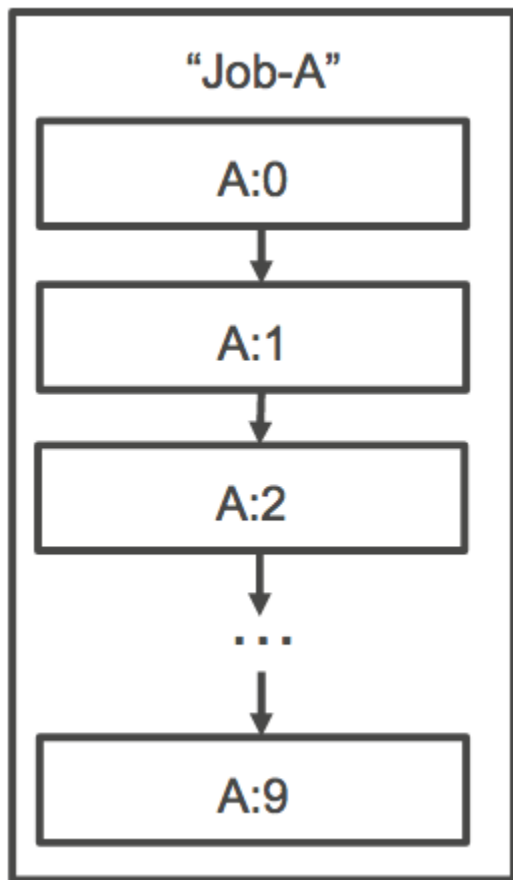
Al enviar un trabajo de matriz, el trabajo de matriz principal recibe un ID de AWS Batch trabajo normal. Cada trabajo secundario tiene el mismo ID base. Cada trabajo secundario tiene el

mismo ID de base, pero su índice de matriz se añade al final del ID principal, por ejemplo, *example_job_ID:0* para el primer trabajo secundario de la matriz.

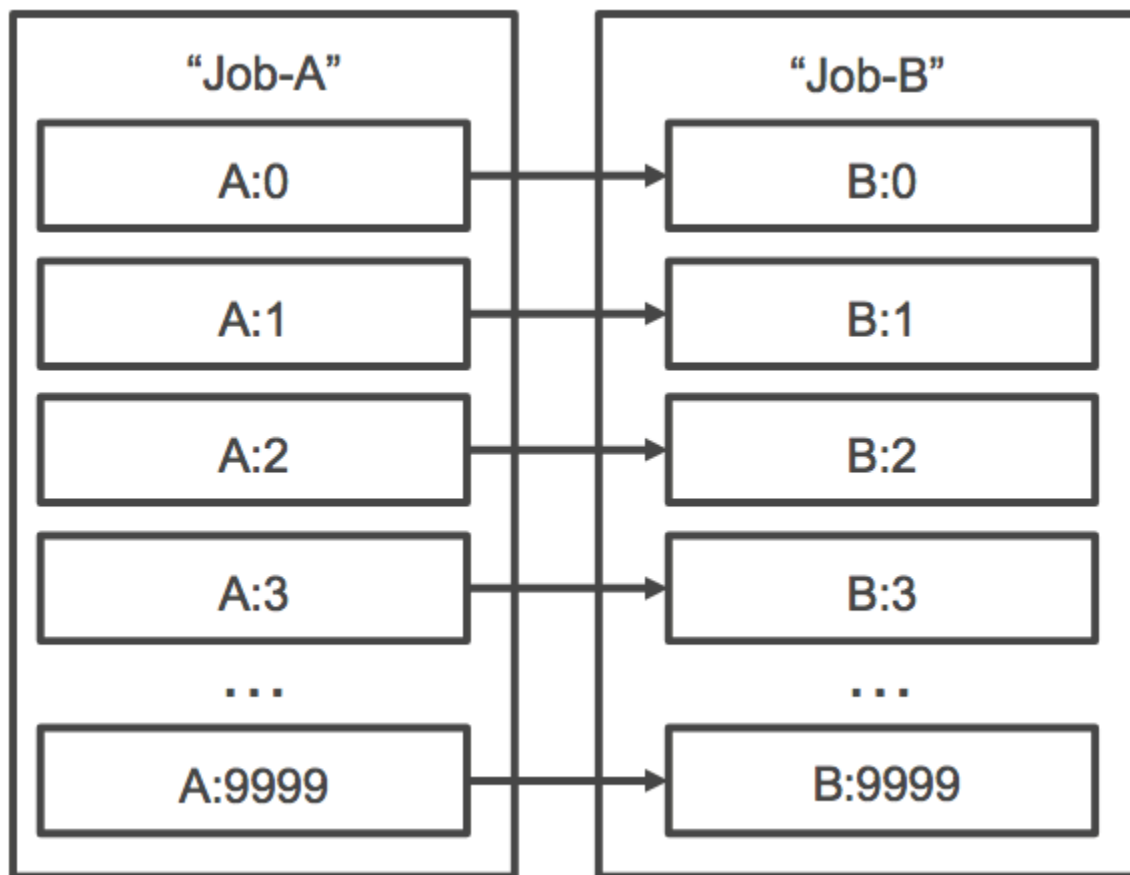
El trabajo de matriz principal puede introducir un estado SUBMITTED, PENDING, FAILED o SUCCEEDED. El trabajo de matriz principal se actualiza a PENDING cuando se actualiza cualquier trabajo secundario a RUNNABLE. Para obtener más información acerca de estas dependencias, consulte [Dependencias de trabajos](#).

Durante el tiempo de ejecución, la variable de entorno AWS_BATCH_JOB_ARRAY_INDEX se establece en el número de índice del trabajo de matriz correspondiente del contenedor. Al primer índice de trabajo de matriz se le asigna el número 0, y los trabajos secundarios posteriores se numeran en orden ascendente (por ejemplo, 1, 2, 3, etc.). Puede utilizar este valor de índice para controlar la forma en la que se diferencian los elementos secundarios del trabajo de matriz. Para obtener más información, consulte [Uso del índice de trabajo de matriz para controlar la diferenciación de trabajos](#).

En las dependencias de trabajo de matriz, puede especificar un tipo para la dependencia, como, por ejemplo, SEQUENTIAL o N_TO_N. Puede especificar una dependencia de tipo SEQUENTIAL (sin especificar un ID de trabajo) para que cada trabajo de matriz secundario se complete de forma secuencial, comenzando a partir del índice 0. Por ejemplo, si envía un trabajo de matriz con un tamaño de matriz de 100, y especifica una dependencia de tipo SEQUENTIAL, se generan 100 trabajos secundarios de forma secuencial, cada uno de los cuales debe completarse correctamente para que comience el siguiente. En la ilustración siguiente se muestra Job A, un trabajo de matriz con un tamaño de matriz de 10. Cada trabajo del índice secundario de Job A depende del trabajo secundario anterior. Job A:1 no puede comenzar hasta que termine Job A:0.



También puede especificar una dependencia de tipo N_TO_N con un ID de trabajo para los trabajos de matriz. De esta forma, cada índice secundario de este trabajo debe esperar a que se complete el índice secundario correspondiente de cada dependencia antes de comenzar. En la ilustración siguiente se muestran Job A y Job B, dos trabajos de matrices con un tamaño de matriz de 10 000. Cada trabajo del índice secundario de Job B depende del índice correspondiente de Job A. Job B:1 no puede comenzar hasta que termine Job A:1.



Si cancela o termina un trabajo de matriz principal, todos los trabajos secundarios se cancelan o terminan con él. Puede cancelar o terminar trabajos secundarios por separado (con lo que adoptarán el estado FAILED) sin afectar al resto de trabajos secundarios. Sin embargo, si un trabajo de matriz secundario produce un error (durante su ejecución o por una cancelación/terminación manual), el trabajo principal también producirá un error. En este escenario, el trabajo principal cambia a FAILED cuando se completan todos los trabajos secundarios.

Para obtener más información sobre la búsqueda y el filtrado de trabajos de matriz, consulte [Buscar para trabajos en una cola de trabajos](#).

Temas

- [Ejemplo de un flujo de trabajos de matriz](#)
- [Uso del índice de trabajo de matriz para controlar la diferenciación de trabajos](#)

Ejemplo de un flujo de trabajos de matriz

Un flujo de trabajo habitual para AWS Batch los clientes consiste en ejecutar un trabajo de configuración previo, ejecutar una serie de comandos en un gran número de tareas de entrada y, a continuación, concluir con un trabajo que agregue los resultados y escriba datos resumidos en Amazon S3, DynamoDB, Amazon Redshift o Aurora.

Por ejemplo:

- JobA: un trabajo estándar que no es de matriz, que realiza un listado rápido y una validación de metadatos de los objetos de un bucket BucketA de Amazon S3. La sintaxis de [SubmitJobJSON](#) es la siguiente.

```
{
  "jobName": "JobA",
  "jobQueue": "ProdQueue",
  "jobDefinition": "JobA-list-and-validate:1"
}
```

- JobB: un trabajo de matriz con 10 000 copias que depende de JobA, que ejecuta comandos con un uso intensivo de la CPU en cada objeto de BucketA y que carga los resultados en BucketB. La sintaxis de [SubmitJobJSON](#) es la siguiente.

```
{
  "jobName": "JobB",
  "jobQueue": "ProdQueue",
  "jobDefinition": "JobB-CPU-Intensive-Processing:1",
  "containerOverrides": {
    "resourceRequirements": [
      {
        "type": "MEMORY",
        "value": "4096"
      },
      {
        "type": "VCPU",
        "value": "32"
      }
    ]
  }
  "arrayProperties": {
    "size": 10000
  },
}
```



```

    "dependsOn": [
      {
        "jobId": "JobA_job_ID"
      }
    ]
  }
}

```

- JobC: otro trabajo de matriz de 10 000 copias que depende de JobB con un modelo de dependencia N_TO_N, que ejecuta comandos con un uso intensivo de la memoria en cada elemento de BucketB, escribe los metadatos en DynamoDB y carga la salida obtenida en BucketC . La sintaxis de [SubmitJobJSON](#) es la siguiente.

```

{
  "jobName": "JobC",
  "jobQueue": "ProdQueue",
  "jobDefinition": "JobC-Memory-Intensive-Processing:1",
  "containerOverrides": {
    "resourceRequirements": [
      {
        "type": "MEMORY",
        "value": "32768"
      },
      {
        "type": "VCPU",
        "value": "1"
      }
    ]
  }
  "arrayProperties": {
    "size": 10000
  },
  "dependsOn": [
    {
      "jobId": "JobB_job_ID",
      "type": "N_TO_N"
    }
  ]
}

```

- JobD: un trabajo de matriz que realiza 10 pasos de validación que requieren consultar DynamoDB y que pueden interactuar con cualquiera de los buckets de Amazon S3 anteriores. Cada uno de los pasos de JobD ejecuta el mismo comando. Sin embargo, el comportamiento es diferente en

función del valor de la variable de entorno `AWS_BATCH_JOB_ARRAY_INDEX` del contenedor del trabajo. Estos pasos de validación se ejecutan de forma secuencial (por ejemplo `JobD:0` y, a continuación, `JobD:1`). La sintaxis de [SubmitJob](#)JSON es la siguiente.

```
{
  "jobName": "JobD",
  "jobQueue": "ProdQueue",
  "jobDefinition": "JobD-Sequential-Validation:1",
  "containerOverrides": {
    "resourceRequirements": [
      {
        "type": "MEMORY",
        "value": "32768"
      },
      {
        "type": "VCPU",
        "value": "1"
      }
    ]
  },
  "arrayProperties": {
    "size": 10
  },
  "dependsOn": [
    {
      "jobId": "JobC_job_ID"
    },
    {
      "type": "SEQUENTIAL"
    }
  ]
}
```

- JobE: un trabajo final que no es de matriz, que realiza varias operaciones de limpieza sencillas, y que envía una notificación de Amazon SNS con un mensaje que indica que la canalización se ha completado y que incluye un enlace con la URL de salida. La sintaxis de [SubmitJob](#)JSON es la siguiente.

```
{
  "jobName": "JobE",
  "jobQueue": "ProdQueue",
```

```
"jobDefinition": "JobE-Cleanup-and-Notification:1",
"parameters": {
  "SourceBucket": "s3://amzn-s3-demo-source-bucket",
  "Recipient": "pipeline-notifications@mycompany.com"
},
"dependsOn": [
  {
    "jobId": "JobD_job_ID"
  }
]
}
```

Uso del índice de trabajo de matriz para controlar la diferenciación de trabajos

En este tutorial se describe cómo utilizar la variable de entorno de `AWS_BATCH_JOB_ARRAY_INDEX` para diferenciar los trabajos secundarios. Cada trabajo secundario se asigna a esta variable. El ejemplo utiliza el número de índice del trabajo secundario para leer una línea específica de un archivo. A continuación, sustituye el parámetro asociado a ese número de línea por un comando incluido en el contenedor del trabajo. El resultado es que puede tener varios AWS Batch trabajos que ejecuten la misma imagen de Docker y los mismos argumentos de comando. Sin embargo, los resultados son diferentes porque el índice de trabajos de la matriz se usa como modificador.

En este tutorial, creará un archivo de texto que tiene todos los colores del arco iris, cada uno en su propia línea. A continuación, creará un script de punto de entrada para un contenedor de Docker que convierte el índice en un valor que se puede utilizar como número de línea en el archivo de colores. El índice comienza en cero, pero los números de línea comienzan en uno. Creará un Dockerfile que copia los archivos de colores y de índice en la imagen del contenedor y establece el valor `ENTRYPOINT` de la imagen en el script de punto de entrada. El Dockerfile y los recursos se compilarán en una imagen de Docker que se enviará a Amazon ECR. A continuación, registras una definición de trabajo que usa tu nueva imagen de contenedor, envías un trabajo de AWS Batch matriz con esa definición de trabajo y ves los resultados.

Temas

- [Requisitos previos](#)
- [Creación de una imagen de contenedor](#)
- [Enviar la imagen a Amazon ECR](#)

- [Crear y registrar una definición de trabajo](#)
- [Envíe un trabajo AWS Batch de matriz](#)
- [Ver los registros del trabajo de matriz](#)

Requisitos previos

Este flujo de trabajo de tutorial tiene los requisitos previos siguientes:

- Un entorno AWS Batch informático. Para obtener más información, consulte [Creación de un entorno de computación](#).
- Una cola de AWS Batch trabajos y un entorno informático asociado. Para obtener más información, consulte [Creación de una cola de trabajos](#).
- Están AWS CLI instalados en su sistema local. Para obtener más información, consulte [>Instalación de AWS Command Line Interface](#) en la Guía del usuario de AWS Command Line Interface .
- Docker debe estar instalado en el sistema local. Para obtener más información, consulte [About Docker CE](#) en la documentación de Docker.

Creación de una imagen de contenedor

Puede utilizar el `AWS_BATCH_JOB_ARRAY_INDEX` en una definición de trabajo en el parámetro de comando. Sin embargo, se recomienda crear una imagen contenedor que utilice la variable en un script de punto de entrada en su lugar. En esta sección, se describe cómo crear dicha imagen de contenedor.

Para compilar la imagen de contenedor de Docker

1. Cree un directorio nuevo para utilizarlo como espacio de trabajo de la imagen de Docker y desplácese a él.
2. Cree un archivo denominado `colors.txt` en el directorio del espacio de trabajo y pegue en él el contenido siguiente.

```
red
orange
yellow
green
blue
```

```
indigo
violet
```

3. Cree un archivo denominado `print-color.sh` en el directorio del espacio de trabajo y pegue en él el contenido siguiente.

Note

La variable `LINE` se establece en `AWS_BATCH_JOB_ARRAY_INDEX + 1` debido a que el índice de matriz empieza en 0, pero los números de línea empiezan en 1. La variable `COLOR` se establece en el color de `colors.txt` que está asociado a su número de línea.

```
#!/bin/sh
LINE=$((AWS_BATCH_JOB_ARRAY_INDEX + 1))
COLOR=$(sed -n ${LINE}p /tmp/colors.txt)
echo My favorite color of the rainbow is $COLOR.
```

4. Cree un archivo denominado `Dockerfile` en el directorio del espacio de trabajo y pegue en él el contenido siguiente. Este `Dockerfile` copia los archivos anteriores en el contenedor y configura el script de punto de entrada para que se ejecute al iniciarse el contenedor.

```
FROM busybox
COPY print-color.sh /tmp/print-color.sh
COPY colors.txt /tmp/colors.txt
RUN chmod +x /tmp/print-color.sh
ENTRYPOINT /tmp/print-color.sh
```

5. Compile la imagen de Docker.

```
$ docker build -t print-color .
```

6. Pruebe el contenedor con el script siguiente. Este script establece la variable `AWS_BATCH_JOB_ARRAY_INDEX` en 0 localmente y, a continuación, incrementa su valor para simular lo que haría un trabajo de matriz con siete elementos secundarios.

```
$ AWS_BATCH_JOB_ARRAY_INDEX=0
while [ $AWS_BATCH_JOB_ARRAY_INDEX -le 6 ]
do
```

```
docker run -e AWS_BATCH_JOB_ARRAY_INDEX=$AWS_BATCH_JOB_ARRAY_INDEX print-color
AWS_BATCH_JOB_ARRAY_INDEX=$((AWS_BATCH_JOB_ARRAY_INDEX + 1))
done
```

Se genera la siguiente salida.

```
My favorite color of the rainbow is red.
My favorite color of the rainbow is orange.
My favorite color of the rainbow is yellow.
My favorite color of the rainbow is green.
My favorite color of the rainbow is blue.
My favorite color of the rainbow is indigo.
My favorite color of the rainbow is violet.
```

Enviar la imagen a Amazon ECR

Ahora que ha compilado y probado el contenedor de Docker, debe enviarlo a un repositorio de imágenes. En este ejemplo se utiliza Amazon ECR, pero puede utilizar otro registro, como DockerHub.

1. Cree un repositorio de imágenes de Amazon ECR para almacenar la imagen de contenedor. En este ejemplo solo se utiliza el AWS CLI, pero también se puede utilizar el Consola de administración de AWS. Para obtener más información, consulte [Creación de un repositorio](#) en la Guía del usuario de Amazon Elastic Container Registry.

```
$ aws ecr create-repository --repository-name print-color
```

2. Etiquete la imagen `print-color` con el URI del repositorio de Amazon ECR que se obtuvo del paso anterior.

```
$ docker tag print-color aws_account_id.dkr.ecr.region.amazonaws.com/print-color
```

3. Inicie sesión en su registro Amazon ECR. Para obtener más información, consulte [Autenticación de registros](#) en la Guía del usuario de Amazon Elastic Container Registry.

```
$ aws ecr get-login-password \
  --region region | docker login \
  --username AWS \
  --password-stdin aws_account_id.dkr.ecr.region.amazonaws.com
```

4. Envíe la imagen a Amazon ECR.

```
$ docker push aws_account_id.dkr.ecr.region.amazonaws.com/print-color
```

Crear y registrar una definición de trabajo

Ahora que su imagen de Docker está en un registro de imágenes, puede especificarla en una definición de AWS Batch trabajo. Posteriormente, puede utilizarlo para ejecutar un trabajo de matriz. En este ejemplo se utiliza la API de AWS CLI. Sin embargo, también puede utilizar la Consola de administración de AWS. Para obtener más información, consulte [Creación de una definición de trabajo de un solo nodo](#).

Para crear una definición de trabajo

1. Cree un archivo denominado `print-color-job-def.json` en el directorio del espacio de trabajo y pegue en él el contenido siguiente. Reemplace el URI del repositorio de imágenes por el URI de su propia imagen.

```
{
  "jobDefinitionName": "print-color",
  "type": "container",
  "containerProperties": {
    "image": "aws_account_id.dkr.ecr.region.amazonaws.com/print-color",
    "resourceRequirements": [
      {
        "type": "MEMORY",
        "value": "250"
      },
      {
        "type": "VCPU",
        "value": "1"
      }
    ]
  }
}
```

2. Registre la definición del trabajo con AWS Batch.

```
$ aws batch register-job-definition --cli-input-json file://print-color-job-def.json
```

Envíe un trabajo AWS Batch de matriz

Después de registrar la definición de su trabajo, puede enviar un trabajo de AWS Batch matriz que utilice su nueva imagen de contenedor.

Para enviar un trabajo AWS Batch de matriz

1. Cree un archivo denominado `print-color-job.json` en el directorio del espacio de trabajo y pegue en él el contenido siguiente.

Note

En este ejemplo, se utiliza la cola de trabajos mencionada en la sección [the section called “Requisitos previos”](#).

```
{
  "jobName": "print-color",
  "jobQueue": "existing-job-queue",
  "arrayProperties": {
    "size": 7
  },
  "jobDefinition": "print-color"
}
```

2. Envíe el trabajo a su lista de AWS Batch trabajos. Anote el ID de trabajo que se devuelve como resultado.

```
$ aws batch submit-job --cli-input-json file://print-color-job.json
```

3. Describa el estado del trabajo y espere a que el trabajo adopte el valor SUCCEEDED.

Ver los registros del trabajo de matriz

Cuando su trabajo alcance el SUCCEEDED estado, podrá ver los CloudWatch registros desde el contenedor del trabajo.

Para ver los registros de su trabajo en CloudWatch Registros

1. Abra la AWS Batch consola en <https://console.aws.amazon.com/batch/>.

2. En el panel de navegación izquierdo, elija Jobs (Trabajos).
3. En Job queue (Cola de trabajos), seleccione una cola.
4. En la sección Status (Estado), elija succeeded (realizado correctamente).
5. Para mostrar todos los trabajos secundarios del trabajo de matriz, seleccione el ID de trabajo que se ha obtenido en la sección anterior.
6. Para ver los logs del contenedor del trabajo, seleccione uno de los trabajos secundarios y elija View logs (Ver logs).

Filter events	
Time (UTC +00:00)	Message
2018-07-13	
No older events found at the moment. Retry.	
▶ 20:16:20	My favorite color of the rainbow is red.
No newer events found at the moment. Retry.	

7. Vea los logs del otro trabajo secundario. Cada trabajo devuelve un color diferente del arco iris.

Ejecución de trabajos de GPU

Los trabajos de GPU le ayudan a ejecutar trabajos que utilizan los de una instancia. GPUs

Se admiten los siguientes tipos de instancias de Amazon EC2 basadas en GPU. Para obtener más información, consulte [instancias G3 de Amazon EC2](#), [instancias G4 de Amazon EC2](#), [instancias G5 de Amazon EC2](#), [instancias G6 de Amazon EC2](#), [instancias P2 de Amazon EC2](#), [instancias P3 de Amazon EC2](#), [instancias P4d de Amazon EC2](#), [instancias P5 de Amazon EC2](#), [instancias P6 de Amazon EC2](#), [instancias Trn1 de Amazon EC2](#), [instancias Trn2 de Amazon EC2](#), [instancias de Inf1 de Amazon EC2](#), [instancias Inf2 de Amazon EC2](#), [instancias DI1 de Amazon EC2](#) e [instancias DI2 de Amazon EC2](#).

Tipo de instancia	GPUs	Memoria de GPU	v CPUs	Memoria	Ancho de banda de la red
g3s.xlarge	1	8 GiB	4	30,5 GiB	10 Gbps
g3.4xlarge	1	8 GiB	16	122 GiB	Hasta 10 Gbps

Tipo de instancia	GPUs	Memoria de GPU	v CPUs	Memoria	Ancho de banda de la red
g3.8xlarge	2	16 GiB	32	244 GiB	10 Gbps
g3.16xlarge	4	32 GiB	64	488 GiB	25 Gbps
g4dn.xlarge	1	16 GiB	4	16 GiB	Hasta 25 Gbps.
g4dn.2xlarge	1	16 GiB	8	32 GiB	Hasta 25 Gbps.
g4dn.4xlarge	1	16 GiB	16	64 GiB	Hasta 25 Gbps.
g4dn.8xlarge	1	16 GiB	32	128 GiB	50 Gbps
g4dn.12xlarge	4	64 GiB	48	192 GiB	50 Gbps
g4dn.16xlarge	1	16 GiB	64	256 GiB	50 Gbps
g5.xlarge	1	24 GiB	4	16 GiB	Hasta 10 Gbps
g5.2xlarge	1	24 GiB	8	32 GiB	Hasta 10 Gbps
g5.4xlarge	1	24 GiB	16	64 GiB	Hasta 25 Gbps.
g5.8xlarge	1	24 GiB	32	128 GiB	25 Gbps
g5.16xlarge	1	24 GiB	64	256 GiB	25 Gbps
g5.12xlarge	4	96 GiB	48	192 GiB	40 Gbps
g5.24xlarge	4	96 GiB	96	384 GiB	50 Gbps
g5.48xlarge	8	192 GiB	192	768 GiB	100 Gbps
g5g.xlarge	1	16 GiB	4	8 GiB	Hasta 10 Gbps
g5g.2xlarge	1	16 GiB	8	16 GiB	Hasta 10 Gbps
g5g.4xlarge	1	16 GiB	16	32 GiB	Hasta 10 Gbps
g5g.8xlarge	1	16 GiB	32	64 GiB	12 Gbps

Tipo de instancia	GPUs	Memoria de GPU	v CPUs	Memoria	Ancho de banda de la red
g5g.16xlarge	2	32 GiB	64	128 GiB	25 Gbps
g5g.metal	2	32 GiB	64	128 GiB	25 Gbps
g6.xlarge	1	24 GiB	4	16 GiB	Hasta 10 Gbps
g6.2xlarge	1	24 GiB	8	32 GiB	Hasta 10 Gbps
g6.4xlarge	1	24 GiB	16	64 GiB	Hasta 25 Gbps.
g6.8xlarge	1	24 GiB	32	128 GiB	25 Gbps
g6.16xlarge	1	24 GiB	64	256 GiB	25 Gbps
g6.12xlarge	4	96 GiB	48	192 GiB	40 Gbps
g6.24xlarge	4	96 GiB	96	384 GiB	50 Gbps
g6.48xlarge	8	192 GiB	192	768 GiB	100 Gbps
g6e.xlarge	1	48 GiB	4	32 GiB	Hasta 20 Gbps
g6e.2xlarge	1	48 GiB	8	64 GiB	Hasta 20 Gbps
g6e.4xlarge	1	48 GiB	16	128 GiB	20 Gbps
g6g.8xlarge	1	48 GiB	32	256 GiB	25 Gbps
g6e.16xlarge	1	48 GiB	64	512 GiB	35 Gbps
g6e.12xlarge	4	192 GiB	48	384 GiB	100 Gbps
g6e.24xlarge	4	192 GiB	96	768 GiB	200 Gbps
g6e.48xlarge	8	384 GiB	192	1536 GiB	400 Gbps
gr6.4xlarge	1	24 GiB	16	128 GiB	Hasta 25 Gbps.
gr6.8xlarge	1	24 GiB	32	256 GiB	25 Gbps

Tipo de instancia	GPUs	Memoria de GPU	v CPUs	Memoria	Ancho de banda de la red
p2.xlarge	1	12 GiB	4	61 GiB	Alto
p2.8xlarge	8	96 GiB	32	488 GiB	10 Gbps
p2.16xlarge	16	192 GiB	64	732 GiB	20 Gbps
p3.2xlarge	1	16 GiB	8	61 GiB	Hasta 10 Gbps
p3.8xlarge	4	64 GiB	32	244 GiB	10 Gbps
p3.16xlarge	8	128 GiB	64	488 GiB	25 Gbps
p3dn.24xlarge	8	256 GiB	96	768 GiB	100 Gbps
p4d.24xlarge	8	320 GiB	96	1152 GiB	400 Gbps
p4de.24xlarge	8	640 GiB	96	1152 GiB	400 Gbps
p5.48xlarge	8	640 GiB	192	2 TiB	3200 Gbps
p5e.48xlarge	8	1128 GiB	192	2 TiB	3200 Gbps
p5en.48xlarge	8	1128 GiB	192	2 TiB	3200 Gbps
p6-b200.48xlarge	8	1440 GiB	192	2 TiB	100 Gbps
trn1.2xlarge	1	32 GiB	8	32 GiB	Hasta 12,5 Gbps
trn1.32xlarge	16	512 GiB	128	512 GiB	800 Gbps
trn1n.32xlarge	16	512 GiB	128	512 GiB	1600 Gbps
trn2.48xlarge	16	1,5 TiB	192	2 TiB	3,2 Tbps
inf1.xlarge	1	8 GiB	4	8 GiB	Hasta 25 Gbps.
inf1.2xlarge	1	8 GiB	8	16 GiB	Hasta 25 Gbps.
inf1.6xlarge	4	32 GiB	24	48 GiB	25 Gbps

Tipo de instancia	GPUs	Memoria de GPU	v CPUs	Memoria	Ancho de banda de la red
inf1.24xlarge	16	128 GiB	96	192 GiB	100 Gbps
inf2.xlarge	1	32 GiB	4	16 GiB	Hasta 15 Gbps
inf2.8xlarge	1	32 GiB	32	128 GiB	Hasta 25 Gbps.
inf2.24xlarge	6	192 GiB	96	384 GiB	50 Gbps
inf2.48xlarge	12	384 GiB	192	768 GiB	100 Gbps
dl1.24xlarge	8	256 GiB	96	768 GiB	400 Gbps
dl2q.24xlarge	8	128 GiB	96	768 GiB	100 Gbps

Note

Para trabajos de GPU, AWS Batch solo admite tipos de instancias que tengan NVIDIA GPUs. Por ejemplo, la familia de [G4ad](#) no es compatible con la programación de GPU. Puede seguir [G4ad](#) utilizándolo AWS Batch definiendo solo los requisitos de vcpu y memoria en la definición del trabajo y, a continuación, accediendo a las GPU del host directamente mediante la personalización en una [plantilla de lanzamiento de Amazon EC2, los datos de usuario](#) con una AMI optimizada para cómputo de Amazon ECS o Amazon EKS, o una AMI personalizada para usar AMD. GPUs

Los tipos de instancias que utilizan una ARM64 arquitectura son compatibles con los trabajos de GPU con datos de usuario personalizados AMIs proporcionados a Amazon EC2 AWS Batch o para acceder a ellos GPUs mediante códigos y configuraciones personalizados. Por ejemplo, las familias de instancias de [G5g](#).

El parámetro [ResourceRequirements](#) de la definición del trabajo especifica el número de tareas que se van GPUs a fijar al contenedor. Este número de no GPUs está disponible para ningún otro trabajo que se ejecute en esa instancia durante ese trabajo. Todos los tipos de instancia de un entorno de computación que ejecutan los trabajos de GPU deben pertenecer a las familias de instancias de p3, p4, p5, p6, g3, g3s, g4, g5 o g6. Si no lo hace así, un trabajo de GPU podría quedar bloqueado en el estado RUNNABLE.

Los trabajos que no utilizan el se GPUs pueden ejecutar en instancias de GPU. Sin embargo, es posible que su ejecución en las instancias de GPU cueste más que en instancias similares que no sean de GPU. En función de la vCPU específica, la memoria y el tiempo necesario, estos trabajos sin GPU pueden bloquear la ejecución de los trabajos de GPU.

Temas

- [Creación de un clúster de Kubernetes basado en GPU en Amazon EKS](#)
- [Creación de una definición de trabajo de GPU de Amazon EKS](#)
- [Ejecución de un trabajo de GPU en el clúster de Amazon EKS](#)

Creación de un clúster de Kubernetes basado en GPU en Amazon EKS

Antes de crear un clúster de Kubernetes basado en GPU en Amazon EKS, debe haber completado los pasos que se indican en [Cómo empezar con AWS Batch Amazon EKS](#). Además, tenga en cuenta lo siguiente:

- AWS Batch admite tipos de instancias con NVIDIA GPUs.
- De forma predeterminada, AWS Batch selecciona la AML acelerada de Amazon EKS con la Kubernetes versión que coincide con la versión del plano de control del clúster de Amazon EKS.

```
$ cat <<EOF > ./batch-eks-gpu-ce.json
{
  "computeEnvironmentName": "My-Eks-GPU-CE1",
  "type": "MANAGED",
  "state": "ENABLED",
  "eksConfiguration": {
    "eksClusterArn": "arn:aws:eks:<region>:<account>:cluster/<cluster-name>",
    "kubernetesNamespace": "my-aws-batch-namespace"
  },
  "computeResources": {
    "type": "EC2",
    "allocationStrategy": "BEST_FIT_PROGRESSIVE",
    "minvCpus": 0,
    "maxvCpus": 1024,
    "instanceTypes": [
      "p3dn.24xlarge",
      "p4d.24xlarge"
    ],
    "subnets": [
```

```

    "<eks-cluster-subnets-with-access-to-internet-for-image-pull>"
  ],
  "securityGroupIds": [
    "<eks-cluster-sg>"
  ],
  "instanceRole": "<eks-instance-profile>"
}
}
EOF

$ aws batch create-compute-environment --cli-input-json file:///./batch-eks-gpu-ce.json

```

AWS Batch no administra el complemento del dispositivo NVIDIA GPU en su nombre. Debe instalar este complemento en su clúster de Amazon EKS y permitir que se dirija a los AWS Batch nodos. Para obtener más información, consulte [Habilitar GPU Support en Kubernetes](#) on GitHub.

Para configurar el complemento del NVIDIA dispositivo (DaemonSet) para que se dirija a los AWS Batch nodos, ejecute los siguientes comandos.

```

# pull nvidia daemonset spec
$ curl -O https://raw.githubusercontent.com/NVIDIA/k8s-device-plugin/v0.12.2/nvidia-device-plugin.yml
# using your favorite editor, add Batch node toleration
# this will allow the DaemonSet to run on Batch nodes
- key: "batch.amazonaws.com/batch-node"
  operator: "Exists"

$ kubectl apply -f nvidia-device-plugin.yml

```

No se recomienda mezclar cargas de trabajo informáticas (CPU y memoria) con cargas de trabajo basadas en GPU en las mismas combinaciones de entorno de computación y cola de tareas. Esto se debe a que las tareas informáticas pueden consumir la capacidad de la GPU.

Para adjuntar colas de trabajos, ejecute los siguientes comandos.

```

$ cat <<EOF > ./batch-eks-gpu-jq.json
{
  "jobQueueName": "My-Eks-GPU-JQ1",
  "priority": 10,
  "computeEnvironmentOrder": [
    {
      "order": 1,

```

```

        "computeEnvironment": "My-Eks-GPU-CE1"
    }
]
}
EOF

$ aws batch create-job-queue --cli-input-json file:///./batch-eks-gpu-jq.json

```

Creación de una definición de trabajo de GPU de Amazon EKS

Solo `nvidia.com/gpu` se admite en este momento y el valor del recurso que establezca debe ser un número entero. No puede usar fracciones de GPU. Para obtener más información, consulte [Programación GPUs](#) en la Kubernetes documentación.

Para registrar una definición de trabajo de GPU para Amazon EKS, ejecute los siguientes comandos.

```

$ cat <<EOF > ./batch-eks-gpu-jd.json
{
  "jobDefinitionName": "MyGPUJobOnEks_Smi",
  "type": "container",
  "eksProperties": {
    "podProperties": {
      "hostNetwork": true,
      "containers": [
        {
          "image": "nvcr.io/nvidia/cuda:10.2-runtime-centos7",
          "command": ["nvidia-smi"],
          "resources": {
            "limits": {
              "cpu": "1",
              "memory": "1024Mi",
              "nvidia.com/gpu": "1"
            }
          }
        }
      ]
    }
  }
}
EOF

$ aws batch register-job-definition --cli-input-json file:///./batch-eks-gpu-jd.json

```


Ejecución de un trabajo de GPU en el clúster de Amazon EKS

El recurso de la GPU no se puede comprimir. AWS Batch crea una especificación de módulo para los trabajos de GPU en la que el valor de la solicitud es igual al valor de los límites. Se trata de un requisito de Kubernetes.

Para reiniciar un trabajo de GPU, ejecute los siguientes comandos.

```
$ aws batch submit-job --job-queue My-Eks-GPU-JQ1 --job-definition MyGPUJob0nEks_Smi --
job-name My-Eks-GPU-Job

# locate information that can help debug or find logs (if using Amazon CloudWatch Logs
with Fluent Bit)
$ aws batch describe-jobs --job <job-id> | jq '.jobs[].eksProperties.podProperties |
{podName, nodeName}'
{
  "podName": "aws-batch.f3d697c4-3bb5-3955-aa6c-977fcf1cb0ca",
  "nodeName": "ip-192-168-59-101.ec2.internal"
}
```

Vea los AWS Batch trabajos en una cola de trabajos

Vea y filtre los trabajos de AWS Batch. Esta característica ofrece una opción para visualizar una cola de trabajos existente y filtrar trabajos según una de las tres opciones.

La búsqueda y el filtrado tienen la capacidad de recuperar trabajos que no están en un estado terminal (SUCCEEDED o FAILED). Una vez que el estado de un trabajo sea SUCCEEDED o FAILED, podrá recuperarlo durante un máximo de siete días. Aún puedes ver los EventBridge registros de un trabajo CloudWatch o de Amazon.

Utilice este procedimiento para enumerar todos los trabajos de una cola de trabajos en la AWS Batch consola. Si lo desea, utilice el campo Filtrar results para restringir los resultados en función de los criterios que especifique.

1. Vaya a la [consola de AWS Batch](#).
2. En el panel de navegación, seleccione Trabajos.
3. Expanda la lista desplegable Cola de trabajos y elija la cola de trabajos en la que desea buscar.

 Note

Puede buscar trabajos en una sola cola de trabajos a la vez.

4. En el campo Filtrar resultados, introduzca las palabras clave que desee incluir en los resultados. Puede usar este campo para filtrar por nombre, estado o ID del trabajo. Según la propiedad, es posible que deba definir operadores adicionales, como igual a (=) o que contengan (:), que debe definir.

 Note

SageMaker Las colas de trabajos de formación solo admiten el filtrado por nombre e ID de trabajo

5. Elija Buscar.

AWS Batch Busque trabajos en una cola de trabajos

Puede buscar y filtrar sus trabajos AWS Batch mediante la búsqueda de empleo. Esta característica ofrece una opción para buscar en una cola de trabajos existente y filtrar sus trabajos.

La búsqueda y el filtrado tienen la capacidad de recuperar trabajos que no están en un estado terminal (SUCCEEDED o FAILED). Una vez que el estado de un trabajo sea SUCCEEDED o FAILED, podrá recuperarlo durante un máximo de siete días. Aún puedes ver los EventBridge registros de un trabajo CloudWatch o de Amazon.

Para buscar con varios criterios al mismo tiempo, utilice la característica Búsqueda avanzada. Por ejemplo, puede incluir uno o todos de los siguientes filtros: Estado, Intervalo de fechas y Criterios adicionales (como nombre del trabajo, definición del trabajo o ID del trabajo).

Buscar AWS Batch trabajos (AWS consola)

Utilice este procedimiento para buscar los trabajos en una cola de trabajos en la AWS Batch consola.

1. Vaya a la [consola de AWS Batch](#).
2. En el panel de navegación, seleccione Trabajos.
3. Active la Búsqueda avanzada.

4. Expanda la lista desplegable Cola de trabajos y elija la cola de trabajos en la que desea buscar.

 Note

Puede buscar trabajos en una sola cola de trabajos a la vez.

5. En las opciones de búsqueda:
 - a. En la lista desplegable de Estados, puedes elegir uno o más estados por los que filtrar. Para obtener más información, consulte [Estados de trabajo](#) y [Estado del trabajo de servicio](#).

 Note

Los padres de los trabajos de matriz se actualizan PENDING cuando se actualiza un trabajo secundario RUNNABLE y permanecen en PENDING su estado mientras los trabajos secundarios se están ejecutando. Para ver estos trabajos, filtre por PENDING estado hasta que todos los trabajos secundarios alcancen un estado terminal.

- b. (Opcional) Elija un Intervalo de fechas para filtrar los resultados en función de un rango de fechas y horas.
 - Seleccione el Modo relativo para buscar trabajos que tengan una fecha de creación en un intervalo de tiempo que cuente hacia atrás desde la fecha y la hora actuales.
 - Seleccione el Modo absoluto para buscar trabajos que tengan una fecha de creación en el intervalo de fechas y horas que especifique.
 - c. En el campo Criterios adicionales, introduzca las palabras clave que desee incluir en los resultados de búsqueda. Por ejemplo, puede usar este campo para buscar por nombre de trabajo, definición de trabajo, ID de trabajo o identificador de recurso compartido. Según la propiedad, es posible que deba definir operadores adicionales, como igual a (=) o que contengan (:), que debe definir.

 Note

SageMaker Las colas de trabajos de formación solo admiten el filtrado por nombre e ID de trabajo

 Note

Al filtrar por identificador de recurso compartido, también puede especificar el estado de un trabajo. Se trata de una excepción a la restricción, en la que otros filtros excluyen el filtrado del estado del trabajo.

6. Elija Buscar.

Buscar y filtrar AWS Batch trabajos (AWS CLI)

Use este procedimiento para enumerar todos los trabajos de una cola de trabajos con AWS CLI. Si lo desea, utilice el parámetro `-filters` para restringir los resultados en función de los criterios que especifique.

Search job queue (AWS CLI)

Puede utilizar el comando [list-jobs](#) para buscar y filtrar una cola de trabajos.

Por ejemplo, puede buscar una cola de trabajos según el nombre del trabajo:

```
aws batch list-jobs \  
  --job-queue my-job-queue \  
  --filters name=JOB_NAME,values="my-job"
```

Filtrar trabajos por identificador de recursos compartidos:

```
aws batch list-jobs \  
  --job-queue my-job-queue \  
  --filters name=SHARE_IDENTIFIER,values="my-share"
```

Al filtrar por identificador de recurso compartido, puede incluir el estado del trabajo:

```
aws batch list-jobs \  
  --job-queue my-job-queue \  
  --job-status RUNNING \  
  --filters name=SHARE_IDENTIFIER,values="my-share"
```

En el comando anterior, realice los siguientes cambios:

- *my-job-queue* Sustitúyalo por el nombre de la cola de trabajos.
- *my-job* Sustitúyalo por el nombre de tu trabajo.
- *my-share* Sustitúyalo por el identificador de recurso compartido por el que quieras filtrar.

Search service job queue (AWS CLI)

Puede usar el [list-service-jobs](#) comando para buscar y filtrar una cola de trabajos de servicio.

Por ejemplo, puede buscar una cola de trabajos de servicio según el nombre del trabajo:

```
aws batch list-service-jobs \
  --job-queue my-sm-queue \
  --filters name=JOB_NAME,values="my-sm-job"
```

Filtre los trabajos de servicio por identificador de recurso compartido:

```
aws batch list-service-jobs \
  --job-queue my-sm-queue \
  --filters name=SHARE_IDENTIFIER,values="my-share"
```

En el comando anterior, realice los siguientes cambios:

- *my-sm-queue* Sustitúyalo por el nombre de la cola de trabajos de servicio.
- *my-sm-job* Sustitúyalo por el nombre de tu trabajo de servicio.
- *my-share* Sustitúyalo por el identificador de recurso compartido por el que quieras filtrar.

Modos de red para AWS Batch trabajos

En la siguiente tabla se describen los modos de red y el uso típico de los tipos de AWS Batch trabajo. Consulte los enlaces de la columna “Tipo de trabajo” para obtener más información sobre las consideraciones y los comportamientos.

Tipo de trabajo	Modos de redes compatibles	Uso normal
Trabajo sencillo ECS-EC2	host	Se utiliza para las cargas de trabajo por lotes, fácilmente paralelas y escalables, que

Tipo de trabajo	Modos de redes compatibles	Uso normal
		solo requieren la salida a la VPC definida en el entorno de computación.
Trabajo paralelo de varios nodos ECS-EC2	awsvpc	Se utiliza para cargas de trabajo distribuidas entre varios host (nodos) estrechamente acopladas y modeladas como una sola tarea con comunicaciones coordinadas entre los nodos de tareas.
Trabajo sencillo ECS-Fargate	awsvpc	Tecnología sin servidor para cargas de trabajo por lotes fácilmente paralelas. Suele ser el modelo de trabajo con el menor costo total de propiedad y el mayor aislamiento de contenedores.
Trabajo sencillo EKS-EC2	host y pod	Se utiliza para las cargas de trabajo por lotes, fácilmente paralelas y escalables, que solo requieren la salida a la VPC definida en el entorno de computación. El valor predeterminado es la red de host.

Tipo de trabajo	Modos de redes compatibles	Uso normal
Trabajo paralelo de varios nodos EKS-EC2	host y pod	Se utiliza para cargas de trabajo distribuidas entre varios host (nodos) estrechamente acopladas y modeladas como una sola tarea con comunicaciones coordinadas entre los nodos de pods. El valor predeterminado es la red de host.

Vea los registros de AWS Batch trabajos en CloudWatch los registros

Puede [configurar sus AWS Batch trabajos](#) para enviar información de registro a Amazon CloudWatch Logs. De esta manera, puede ver diferentes registros de sus trabajos en una ubicación práctica. Para obtener más información, consulte [Uso CloudWatch de registros con AWS Batch](#).

También puede usar los registros de tareas de la AWS Batch consola para supervisar o solucionar problemas de una AWS Batch tarea.

1. Abra la [consola de AWS Batch](#).
2. Elija Jobs (Trabajos). Para obtener información más detallada sobre cómo ordenar y filtrar los trabajos de la cola de trabajos, consulte [Vea los AWS Batch trabajos en una cola de trabajos](#) y [Buscar para trabajos en una cola de trabajos](#)
3. En Cola de trabajos, elija la cola de trabajos que desee.

Tip

Si hay varios trabajos en la cola de trabajos, puede activar la Búsqueda y filtrado para encontrar un trabajo más rápido. Para obtener más información, consulte [AWS Batch Busque trabajos en una cola de trabajos](#).

4. En Estado, elija el estado del trabajo que desee.
5. Seleccione el trabajo que desee y se abrirá la página de Detalles.

6. En la página de Detalles, desplácese hacia abajo hasta el Nombre del flujo de registro y haga clic en el enlace. El enlace abre la página de Amazon CloudWatch Logs del trabajo.
7. (Opcional) Si es la primera vez que ve los registros, es posible que le pidan autorización.

Si se requiere autorización, introduce **yOK**, a continuación, selecciona Autorizar para aceptar CloudWatch los cargos de Amazon.

 Note

Para revocar tu autorización de CloudWatch cobro:

1. En el panel de navegación de la izquierda, elija Permisos.
2. Para Registros de trabajos, seleccione Editar.
3. Desactive la CloudWatch casilla Autorizar el uso de Batch.
4. Seleccione Save changes (Guardar cambios).

Revise la información del AWS Batch trabajo

Puede revisar la información del trabajo AWS Batch , como el estado, la definición del trabajo y la información del contenedor.

1. Abra la [consola de AWS Batch](#).
2. Elija Jobs (Trabajos).
3. En Cola de trabajos, elija la cola de trabajos que desee.

 Tip

Si hay varios trabajos en la cola de trabajos, puede activar Buscar y filtrar para encontrar un trabajo más rápido. Para obtener más información, consulte [AWS Batch Busque trabajos en una cola de trabajos](#).

4. Elija el trabajo que desee.

 Note

También puede usar AWS Command Line Interface (AWS CLI) para ver los detalles de un AWS Batch trabajo. Para obtener más información, consulte [describe-jobs](#) en la [AWS CLI Referencia de comandos](#).

Seguridad en AWS Batch

La seguridad en la nube AWS es la máxima prioridad. Como AWS cliente, usted se beneficia de los centros de datos y las arquitecturas de red diseñados 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 Nube de AWS. 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 Batch, consulte [AWS Servicios incluidos en el ámbito de aplicación por programa de conformidad y AWS servicios incluidos](#) .
- Seguridad en la nube: su responsabilidad viene determinada por el servicio de AWS que utilice. También es responsable de otros factores, incluida la confidencialidad de los datos, los requisitos de la empresa y la legislación y los reglamentos vigentes.

Esta documentación le ayuda a comprender cómo aplicar el modelo de responsabilidad compartida cuando se utiliza AWS Batch. Los siguientes temas muestran cómo configurarlo AWS Batch para cumplir sus objetivos de seguridad y conformidad. También aprenderá a utilizar otros AWS servicios que le ayudan a supervisar y proteger sus AWS Batch recursos.

Temas

- [Identity and Access Management para AWS Batch](#)
- [AWS Batch Políticas, funciones y permisos de IAM](#)
- [AWS Batch Función de ejecución de IAM](#)
- [Creación de una nube virtual privada](#)
- [Utilice un punto final de interfaz para acceder AWS Batch](#)
- [Validación de conformidad para AWS Batch](#)
- [Seguridad de la infraestructura en AWS Batch](#)
- [Prevención de la sustitución confusa entre servicios](#)
- [Registrar las llamadas a la AWS Batch API con AWS CloudTrail](#)

- [Solucionar problemas de IAM AWS Batch](#)

Identity and Access Management para AWS Batch

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 Batch La IAM es una Servicio de AWS opción que puede utilizar sin coste adicional.

Temas

- [Público](#)
- [Autenticación con identidades](#)
- [Administración del acceso con políticas](#)
- [¿Cómo AWS Batch funciona con IAM](#)
- [Ejemplos de políticas basadas en la identidad para AWS Batch](#)
- [AWS políticas gestionadas para AWS Batch](#)

Público

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 [Solucionar problemas de IAM AWS Batch](#)).
- Administrador del servicio: determine el acceso de los usuarios y envíe las solicitudes de permiso (consulte [¿Cómo AWS Batch funciona con IAM](#)).
- Administrador de IAM: escribe las políticas para administrar el acceso (consulte [Ejemplos de políticas basadas en la identidad para AWS Batch](#)).

Autenticación con identidades

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 las tareas que requieren credenciales de usuario raíz, consulte [Tareas que requieren credenciales de usuario raíz](#) en la Guía del usuario de IAM.

Identidad federada

Como práctica recomendada, exija a los usuarios humanos que utilicen la federación con un proveedor de identidades para acceder Servicios de AWS mediante credenciales temporales.

Una identidad federada es un usuario del directorio empresarial, del proveedor de identidades web o al Directory Service que se accede Servicios de AWS mediante credenciales de una fuente de identidad. Las identidades federadas asumen roles que proporcionan credenciales temporales.

Para una administración de acceso centralizada, se recomienda AWS IAM Identity Center. Para obtener más información, consulte [¿Qué es el Centro de identidades de IAM?](#) en la Guía del usuario de AWS IAM Identity Center .

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.

Administración del acceso con políticas

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.

Otros tipos de políticas

AWS admite tipos de políticas adicionales que pueden establecer los permisos máximos que conceden 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.

¿Cómo AWS Batch funciona con IAM

Antes de usar IAM para administrar el acceso AWS Batch, infórmese sobre las funciones de IAM disponibles para su uso. AWS Batch

Funciones de IAM que puede utilizar con AWS Batch

Característica de IAM	AWS Batch soporte
Políticas basadas en identidades	Sí
Políticas basadas en recursos	No
Acciones de políticas	Sí
Recursos de políticas	Sí
Claves de condición de política	Sí
ACLs	No
ABAC (etiquetas en políticas)	Sí
Credenciales temporales	Sí
Permisos de entidades principales	Sí
Roles de servicio	Sí
Roles vinculados al servicio	Sí

Para obtener una visión general de cómo AWS Batch funcionan otros AWS servicios con la mayoría de las funciones de IAM, consulte [AWS los servicios que funcionan con IAM](#) en la Guía del usuario de IAM.

Políticas basadas en la identidad para AWS Batch

Compatibilidad con las políticas basadas en identidad: sí

Las políticas basadas en identidad son documentos de políticas de permisos JSON que puede asociar a una identidad, como un usuario de IAM, un grupo de usuarios o un rol. Estas políticas

controlan qué acciones pueden realizar los usuarios y los roles, 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.

Con las políticas basadas en identidades de IAM, puede especificar las acciones y los recursos permitidos o denegados, así como las condiciones en las que se permiten o deniegan las acciones. Para obtener más información sobre los elementos que puede utilizar en una política de JSON, consulte [Referencia de los elementos de la política de JSON de IAM](#) en la Guía del usuario de IAM.

Ejemplos de políticas basadas en la identidad para AWS Batch

Para ver ejemplos de políticas AWS Batch basadas en la identidad, consulte. [Ejemplos de políticas basadas en la identidad para AWS Batch](#)

Acciones políticas para AWS Batch

Compatibilidad con las acciones de políticas: sí

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.

Para ver una lista de AWS Batch acciones, consulte [las acciones definidas por AWS Batch](#) en la referencia de autorización del servicio.

Las acciones políticas AWS Batch utilizan el siguiente prefijo antes de la acción:

```
batch
```

Para especificar varias acciones en una única instrucción, sepárelas con comas.

```
"Action": [  
    "batch:action1",  
    "batch: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": "batch:Describe*"
```

Para ver ejemplos de políticas AWS Batch basadas en la identidad, consulte [Ejemplos de políticas basadas en la identidad para AWS Batch](#)

Recursos de políticas para AWS Batch

Compatibilidad con los recursos de políticas: sí

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": "*"
```

Para ver una lista de los tipos de AWS Batch recursos y sus tipos ARNs, consulte [los recursos definidos por AWS Batch](#) en la Referencia de autorización del servicio. Para obtener información sobre las acciones con las que puede especificar el ARN de cada recurso, consulte [Acciones definidas por AWS Batch](#).

Claves de condición de la política para AWS Batch

Compatibilidad con claves de condición de políticas específicas del servicio: sí

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.

Para ver una lista de claves de AWS Batch condición, consulte las [claves de condición AWS Batch en la](#) Referencia de autorización de servicio. Para saber con qué acciones y recursos puede utilizar una clave de condición, consulte [Acciones definidas por AWS Batch](#).

Control de acceso basado en atributos (ABAC) con AWS Batch

Admite ABAC (etiquetas en las políticas): sí

El control de acceso basado en atributos (ABAC) es una estrategia de autorización que define permisos en función de atributos denominados etiquetas. Puede adjuntar etiquetas a las entidades y AWS los recursos de IAM y, a continuación, diseñar políticas de ABAC para permitir las operaciones cuando la etiqueta del principal coincida con la etiqueta del recurso.

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`.

Si un servicio admite las tres claves de condición para cada tipo de recurso, el valor es Sí para el servicio. Si un servicio admite las tres claves de condición solo para algunos tipos de recursos, el valor es Parcial.

Para obtener más información sobre ABAC, consulte [Definición de permisos con la autorización de ABAC](#) en la Guía del usuario de IAM. Para ver un tutorial con los pasos para configurar ABAC, consulte [Uso del control de acceso basado en atributos \(ABAC\)](#) en la Guía del usuario de IAM.

Utilice credenciales temporales con AWS Batch

Compatibilidad con credenciales temporales: sí

Las credenciales temporales proporcionan acceso a AWS los recursos a corto plazo y se crean automáticamente al utilizar la federación o al cambiar de función. AWS recomienda generar credenciales temporales de forma dinámica en lugar de utilizar claves de acceso a largo plazo. Para obtener más información, consulte [Credenciales de seguridad temporales en IAM](#) y [Servicios de AWS que funcionan con IAM](#) en la Guía del usuario de IAM.

Permisos principales entre servicios para AWS Batch

Admite sesiones de acceso directo (FAS): sí

Las sesiones de acceso directo (FAS) utilizan los permisos de la persona principal que llama Servicio de AWS, junto con la solicitud, Servicio de AWS para realizar solicitudes a los servicios

descendientes. Para obtener información sobre las políticas a la hora de realizar solicitudes de FAS, consulte [Sesiones de acceso directo](#).

Funciones de servicio para AWS Batch

Compatible con roles de servicio: sí

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.

Warning

Cambiar los permisos de un rol de servicio podría interrumpir AWS Batch la funcionalidad. Edite los roles de servicio sólo cuando AWS Batch proporcione orientación para hacerlo.

Funciones vinculadas al servicio para AWS Batch

Compatible con roles vinculados al servicio: sí

Un rol vinculado a un servicio es un tipo de rol de servicio que está vinculado a un. Servicio de AWS El servicio puede asumir el rol para realizar una acción en su nombre. Los roles vinculados al servicio aparecen en usted Cuenta de AWS y son propiedad del servicio. Un administrador de IAM puede ver, pero no editar, los permisos de los roles vinculados a servicios.

Para más información sobre cómo crear o administrar roles vinculados a servicios, consulta [Servicios de AWS que funcionan con IAM](#). Busque un servicio en la tabla que incluya Yes en la columna Rol vinculado a un servicio. Seleccione el vínculo Sí para ver la documentación acerca del rol vinculado a servicios para ese servicio.

Ejemplos de políticas basadas en la identidad para AWS Batch

De forma predeterminada, los usuarios y roles no tienen permiso para crear, ver ni modificar recursos de AWS Batch . Un administrador de IAM puede crear políticas de IAM para conceder permisos a los usuarios para realizar acciones en los recursos que necesitan.

Para obtener información acerca de cómo crear una política basada en identidades de IAM mediante el uso de estos documentos de políticas JSON de ejemplo, consulte [Creación de políticas de IAM \(consola\)](#) en la Guía del usuario de IAM.

Para obtener más información sobre las acciones y los tipos de recursos definidos AWS Batch, incluido el formato de cada uno de los tipos de recursos, consulte [las claves de condición, recursos y acciones de la Referencia AWS Batch](#) de autorización de servicios. ARNs

Temas

- [Prácticas recomendadas sobre las políticas](#)
- [Uso de la consola AWS Batch](#)
- [Cómo permitir a los usuarios consultar sus propios permisos](#)

Prácticas recomendadas sobre las políticas

Las políticas basadas en la identidad determinan si alguien puede crear AWS Batch 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.

Uso de la consola AWS Batch

Para acceder a la AWS Batch consola, debe tener un conjunto mínimo de permisos. Estos permisos deben permitirle enumerar y ver detalles sobre los AWS Batch recursos de su cuenta Cuenta de AWS. Si crea una política basada en identidades que sea más restrictiva que el mínimo de permisos necesarios, la consola no funcionará del modo esperado para las entidades (usuarios o roles) que tengan esa política.

No es necesario que concedas permisos mínimos de consola a los usuarios que solo realicen llamadas a la API AWS CLI o a la AWS API. En su lugar, permita el acceso únicamente a las acciones que coincidan con la operación de API que intentan realizar.

Para garantizar que los usuarios y los roles puedan seguir utilizando la AWS Batch consola, adjunte también la política `ReadOnlyAWS` gestionada AWS Batch `ConsoleAccess` o la política gestionada a las entidades. Para obtener más información, consulte [Adición de permisos a un usuario](#) en la Guía del usuario de IAM:

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:ListGroupsForUser",
        "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": "*"
    }
  ]
}
```

AWS políticas gestionadas para AWS Batch

Puedes usar políticas AWS administradas para simplificar la administración del acceso a la identidad de tu equipo y de AWS los recursos aprovisionados. AWS las políticas gestionadas cubren una variedad de casos de uso comunes, están disponibles de forma predeterminada en tu AWS cuenta y se mantienen y actualizan en tu nombre. No puedes cambiar los permisos en las políticas AWS gestionadas. Si necesita una mayor flexibilidad, también puede optar por crear políticas de

IAM administradas por el cliente. De esta forma, puede proporcionar a su equipo los recursos aprovisionados solo con los permisos exactos que necesitan.

Para obtener más información sobre las políticas AWS administradas, consulte las [políticas AWS administradas](#) en la Guía del usuario de IAM.

AWS los servicios mantienen y AWS actualizan las políticas gestionadas en su nombre. Periódicamente, AWS los servicios añaden permisos adicionales a una política AWS gestionada. AWS Lo más probable es que las políticas gestionadas se actualicen cuando se lance una nueva función o cuando esté disponible una nueva operación. Este tipo de actualización afecta a todas las identidades (usuarios, grupos y roles) donde se asocia la política. Sin embargo, no eliminan los permisos ni anulan los permisos existentes.

Además, AWS admite políticas administradas 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 y descripciones de las políticas de funciones de trabajo, consulte [Políticas administradas de AWS para funciones de trabajo](#) en la Guía del usuario de IAM.

AWS política gestionada: `BatchServiceRolePolicy`

El `BatchServiceRolePolicy` vinculado al [AWSBatchServiceRoleForBatch](#) servicio utiliza la política de IAM gestionada. Esto le permite AWS Batch realizar acciones en su nombre. No puede adjuntar esta política a sus entidades de IAM. Para obtener más información, consulte [Utilizar funciones vinculadas al servicio para AWS Batch](#).

Esta política permite AWS Batch realizar las siguientes acciones en recursos específicos:

- `autoscaling`— Permite AWS Batch crear y gestionar los recursos de Amazon EC2 Auto Scaling. AWS Batch crea y administra grupos de Amazon EC2 Auto Scaling para la mayoría de los entornos informáticos.
- `ec2`— Permite AWS Batch controlar el ciclo de vida de las instancias de Amazon EC2, así como crear y gestionar plantillas y etiquetas de lanzamiento. AWS Batch crea y gestiona las solicitudes de EC2 Spot Fleet para algunos entornos de computación puntual de EC2.

- `ecs`- Permite AWS Batch crear y gestionar clústeres de Amazon ECS, definiciones de tareas y tareas para la ejecución de trabajos.
- `eks`- Permite AWS Batch describir el recurso del clúster Amazon EKS para las validaciones.
- `iam`- Permite AWS Batch validar y transferir las funciones proporcionadas por el propietario a Amazon EC2, Amazon EC2 Auto Scaling y Amazon ECS.
- `logs`— Permite AWS Batch crear y gestionar grupos de registros y flujos de registros para AWS Batch trabajos.

Para ver el JSON de la política, consulte [BatchServiceRolePolicy](#) la [Guía de referencia de políticas AWS administradas](#).

AWS política gestionada: `AWSBatchServiceRolePolicyForSageMaker`

[AWSBatchServiceRoleForAWSBatchWithSagemaker](#) permite AWS Batch realizar acciones en su nombre. No puede adjuntar esta política a sus entidades de IAM. Para obtener más información, consulte [Utilizar funciones vinculadas al servicio para AWS Batch](#).

Esta política permite AWS Batch realizar las siguientes acciones en recursos específicos:

- `sagemaker`— Permite AWS Batch gestionar los trabajos de formación en SageMaker IA y otros recursos de SageMaker IA.
- `iam:PassRole`— Permite AWS Batch transferir las funciones de ejecución definidas por el cliente a la SageMaker IA para la ejecución de las tareas. La limitación de recursos permite transferir funciones a SageMaker los servicios de IA.

Para ver el JSON de la política, consulta la [Guía AWSBatchServiceRolePolicyForSageMakerde referencia de políticas AWS gestionadas](#).

AWS política gestionada: `AWSBatchServiceRole` política

La política de permisos de roles denominada `AWSBatchServiceRole` AWS Batch permite realizar las siguientes acciones en recursos específicos:

La política de IAM `AWSBatchServiceRole` gestionada suele ser utilizada por un rol denominado `AWSBatchServiceRole` incluye los siguientes permisos. Siguiendo el consejo de seguridad estándar de concesión de privilegios mínimos, la política administrada `AWSBatchServiceRole` se puede utilizar como guía. Si alguno de los permisos que se conceden en la política administrada no resulta necesario para su caso de uso, cree una política personalizada y agregue solo los permisos que

necesite. Esta política y esta función AWS Batch gestionadas se pueden utilizar con la mayoría de los tipos de entornos informáticos, pero se prefiere el uso de funciones vinculadas al servicio para disfrutar de una experiencia de gestión menos propensa a errores, con mejor alcance y mejor gestionada.

- **autoscaling**— Permite AWS Batch crear y gestionar los recursos de Amazon EC2 Auto Scaling. AWS Batch crea y administra grupos de Amazon EC2 Auto Scaling para la mayoría de los entornos informáticos.
- **ec2**— Permite AWS Batch gestionar el ciclo de vida de las instancias de Amazon EC2, así como crear y gestionar plantillas y etiquetas de lanzamiento. AWS Batch crea y gestiona las solicitudes de EC2 Spot Fleet para algunos entornos de computación puntual de EC2.
- **ecs**- Permite AWS Batch crear y gestionar clústeres de Amazon ECS, definiciones de tareas y tareas para la ejecución de trabajos.
- **iam**- Permite AWS Batch validar y transferir las funciones proporcionadas por el propietario a Amazon EC2, Amazon EC2 Auto Scaling y Amazon ECS.
- **logs**— Permite AWS Batch crear y gestionar grupos de registros y flujos de registros para AWS Batch trabajos.

Para ver el JSON de la política, consulte [AWSBatchServiceRole](#) la [Guía de referencia de políticas AWS administradas](#).

AWS política gestionada: AWSBatchFullAccess

La AWSBatchFullAccesspolítica otorga a AWS Batch las acciones pleno acceso a AWS Batch los recursos. También otorga acceso a las acciones de descripción y lista para los servicios de Amazon EC2, Amazon ECS CloudWatch, Amazon EKS e IAM. Esto permite que las identidades de IAM, ya sean usuarios o roles, puedan ver los recursos AWS Batch administrados que se crearon en su nombre. Por último, esta política también permite transferir determinados roles de IAM a esos servicios.

Puede adjuntarlos AWSBatchFullAccessa sus entidades de IAM. AWS Batch también vincula esta política a un rol de servicio que le permite AWS Batch realizar acciones en su nombre.

Para ver el JSON de la política, consulta la [Guía AWSBatchFullAccessde referencia de políticas AWS administradas](#).

AWS Batch actualizaciones de las políticas AWS gestionadas

Vea los detalles sobre las actualizaciones de las políticas AWS administradas AWS Batch desde que este servicio comenzó a rastrear estos cambios. Para recibir alertas automáticas sobre los cambios en esta página, suscríbase a la fuente RSS de la página del historial del AWS Batch documento.

Cambio	Descripción	Fecha
AWSBatchServiceRolePolicyForSageMaker política agregada	Se ha añadido una nueva política AWS gestionada para la función <code>AWSBatchServiceRolePolicyForSageMaker</code> vinculada al servicio que permite AWS Batch gestionar la SageMaker IA en tu nombre.	31 de julio de 2025
BatchServiceRolePolicy política actualizada	Se ha actualizado para agregar asistencia al describir el historial de solicitudes y las actividades Amazon EC2 Auto Scaling de la flota de spot.	5 de diciembre de 2023
AWSBatchServiceRole política agregada	Se actualizó para agregar una declaración IDs, otorgar AWS Batch permisos <code>ec2:DescribeSpotFleetRequestsHistory</code> y <code>autoscaling:DescribeScalingActivities</code> .	5 de diciembre de 2023
BatchServiceRolePolicy política actualizada	Se actualizó para añadir compatibilidad con la descripción de los clústeres de Amazon EKS.	20 de octubre de 2022
AWSBatchFullAccess política actualizada	Se actualizó para añadir compatibilidad con el listado y la descripción de los clústeres de Amazon EKS.	20 de octubre de 2022
BatchServiceRolePolicy política actualizada	Se actualizó para añadir compatibilidad con los grupos de reserva de capacidad de Amazon EC2 gestionados por Grupos de recursos de AWS. Para obtener más información, consulte Trabajar con grupos	18 de mayo de 2022

Cambio	Descripción	Fecha
	de reservas de capacidad en la Guía del usuario de Amazon EC2.	
BatchServiceRolePolicy y AWSBatchServiceRole políticas actualizadas	Se ha actualizado para añadir compatibilidad con la descripción del estado de las instancias AWS Batch gestionadas en Amazon EC2, de forma que se sustituyan las instancias en mal estado.	6 de diciembre de 2021
BatchServiceRolePolicy política actualizada	Se actualizó para añadir compatibilidad con los recursos de grupos de ubicación, reserva de capacidad, GPU elástica y Elastic Inference en Amazon EC2.	26 de marzo de 2021
BatchServiceRolePolicy política agregada	Con la política BatchServiceRolePolicy administrada para el rol AWSServiceRoleForBatch vinculado al servicio, puede usar un rol vinculado al servicio administrado por AWS Batch. Con esta política, no necesita mantener su propio rol para usarlo en sus entornos de computación.	10 de marzo de 2021
AWSBatchFullAccess - añadir permiso para añadir un rol vinculado al servicio	Añada permisos de IAM para poder añadir el rol AWSServiceRoleForBatch vinculado al servicio a la cuenta.	10 de marzo de 2021
AWS Batch comenzó a rastrear los cambios	AWS Batch comenzó a realizar un seguimiento de los cambios de sus políticas AWS gestionadas.	10 de marzo de 2021

AWS Batch Políticas, funciones y permisos de IAM

De forma predeterminada, los usuarios no tienen permiso para crear o modificar AWS Batch recursos ni para realizar tareas mediante la AWS Batch API, la AWS Batch consola o la AWS CLI. Para permitir a los usuarios trabajar con estos recursos, debe crear políticas de IAM que concedan

permisos para utilizar recursos específicos y acciones de la API. A continuación, asocie esas políticas a los usuarios o grupos que necesiten esos permisos.

Cuando se adjunta una política a un usuario o grupo de usuarios, la política permite o deniega los permisos para realizar tareas específicas en recursos específicos. Para obtener más información, consulte [Permisos y políticas](#) en la Guía del usuario de IAM. Para obtener más información sobre cómo crear y administrar políticas personalizadas de IAM, consulte [Administración de políticas de IAM](#).

AWS Batch hace llamadas a otras personas Servicios de AWS en tu nombre. Como resultado, AWS Batch debe autenticarse con sus credenciales. Más específicamente, AWS Batch se autentica mediante la creación de una función y una política de IAM que proporcionen estos permisos. A continuación, asocia el rol a sus entornos de computación al crearlos. Para obtener más información, consulte Funciones de [IAMRol de instancia de Amazon ECS, Uso de funciones vinculadas a servicios](#) y [Creación de una función para delegar permisos a un AWS servicio](#) en la Guía del usuario de IAM.

Temas

- [Estructura de la política de IAM](#)
- [Recurso: ejemplos de políticas para AWS Batch](#)
- [Recurso: política AWS Batch gestionada](#)

Estructura de la política de IAM

En los siguientes temas se explica la estructura de una política de IAM.

Temas

- [Sintaxis de la política](#)
- [Acciones de API para AWS Batch](#)
- [Nombres de recursos de Amazon para AWS Batch](#)
- [Comprobación de que los usuarios tienen los permisos necesarios](#)

Sintaxis de la política

Una política de IAM es un documento JSON que contiene una o varias instrucciones. Cada instrucción tiene la estructura siguiente.

```
{
  "Statement": [{
    "Effect": "effect",
    "Action": "action",
    "Resource": "arn",
    "Condition": {
      "condition": {
        "key": "value"
      }
    }
  ]
}
```

Una instrucción está compuesta por cuatro elementos principales:

- **Effect:** el valor de effect puede ser Allow o Deny. Por defecto, los usuarios no tienen permiso para utilizar recursos y acciones de la API. De este modo, se deniegan todas las solicitudes. Si se concede un permiso explícito se anula el valor predeterminado. Una denegación explícita invalida cualquier permiso concedido.
- **Acción:** La acción es la acción específica de la API para la que está concediendo o denegando el permiso. Para obtener instrucciones acerca de cómo especificar la acción, consulte [Acciones de API para AWS Batch](#).
- **Resource:** el recurso al que afecta la acción. Con algunas acciones de la API de AWS Batch , puede incluir recursos específicos en su política que pueden ser creados o modificados por la acción. Para especificar un recurso en la instrucción se utiliza el nombre de recurso de Amazon (ARN). Para obtener más información, consulte [Se admiten permisos a nivel de recursos para AWS Batch las acciones de la API](#) y [Nombres de recursos de Amazon para AWS Batch](#). Si la operación de la AWS Batch API actualmente no admite permisos a nivel de recursos, incluye un comodín (*) para especificar que la acción puede afectar a todos los recursos.
- **Condition:** las condiciones son opcionales. Se pueden usar para controlar cuándo está en vigor la política.

Para obtener más información sobre ejemplos de declaraciones de política de IAM, consulte. AWS Batch [Recurso: ejemplos de políticas para AWS Batch](#)

Acciones de API para AWS Batch

En una instrucción de política de IAM, puede especificar cualquier acción de API de cualquier servicio que sea compatible con IAM. Para AWS Batch, utilice el siguiente prefijo con el nombre de la acción de la API: `batch:` (por ejemplo, `batch:SubmitJob` y `ybatch>CreateComputeEnvironment`).

Para especificar varias acciones en una única sentencia, separe cada acción con una coma.

```
"Action": ["batch:action1", "batch:action2"]
```

También puede especificar varias acciones mediante un comodín (*). Por ejemplo, puede especificar todas las acciones cuyo nombre empiece por la palabra “Describe”.

```
"Action": "batch:Describe*"
```

Para especificar todas las acciones AWS Batch de la API, incluye un comodín (*).

```
"Action": "batch:*"
```

Para ver una lista de AWS Batch acciones, consulta la sección [Acciones](#) en la referencia de la AWS Batch API.

Nombres de recursos de Amazon para AWS Batch

Cada declaración de política de IAM se aplica a los recursos que especifique mediante sus nombres de recursos de Amazon (ARNs).

Un Nombre de recurso de Amazon (ARN) tiene la siguiente sintaxis general:

```
arn:aws:[service]:[region]:[account]:resourceType/resourcePath
```

service

El servicio (por ejemplo, `batch`).

region

La del Región de AWS recurso (por ejemplo, `us-east-2`).

account

El Cuenta de AWS ID, sin guiones (por ejemplo,123456789012).

resourceType

El tipo de recurso (por ejemplo, compute-environment).

resourcePath

Una ruta que identifica al recurso. Puede utilizar un comodín (*) en sus rutas.

AWS Batch En la actualidad, las operaciones de la API admiten permisos a nivel de recursos en varias operaciones de la API. Para obtener más información, consulte [Se admiten permisos a nivel de recursos para AWS Batch las acciones de la API](#). Para especificar todos los recursos, o si una acción específica de la API no es compatible ARNs, incluye un comodín (*) en el elemento.

Resource

```
"Resource": "*"
```

Comprobación de que los usuarios tienen los permisos necesarios

Antes de poner una política de IAM en producción, asegúrese de que concede a los usuarios los permisos para utilizar las acciones y recursos específicos de la API que necesitan.

Para ello, cree primero un usuario de prueba y adjúntele la política de IAM. A continuación, realice una solicitud como usuario de prueba. Puede realizar solicitudes de prueba en la consola o con la AWS CLI.

Note

También puede probar las políticas con el [Simulador de política de IAM](#). Para obtener más información sobre el simulador de políticas, consulte [Trabajar con el simulador de política de IAM](#) en la Guía del usuario de IAM.

Si la política no concede los permisos previstos al usuario o es demasiado permisiva, puede ajustarla según sea necesario. Repita las pruebas hasta obtener el resultado deseado.

⚠ Important

Puede que los cambios en la política tarden varios minutos en propagarse y surtir efecto. Por lo tanto, le recomendamos que deje pasar al menos cinco minutos antes de probar las actualizaciones de sus políticas.

Si se produce un error en la comprobación de autorización, la solicitud devuelve un mensaje codificado con información de diagnóstico. Puede decodificar el mensaje usando la acción `DecodeAuthorizationMessage`. Para obtener más información, consulta [DecodeAuthorizationMessage](#) la Referencia de la AWS Security Token Service API y [decode-authorization-message](#) la Referencia de AWS CLI comandos.

Recurso: ejemplos de políticas para AWS Batch

Puede crear políticas de IAM específicas para restringir las llamadas y los recursos a los que tienen acceso los usuarios de su cuenta. A continuación, puede asignar esas políticas a los usuarios.

Cuando se adjunta una política a un usuario o grupo de usuarios, esta permite o deniega a los usuarios el permiso para tareas específicas en recursos específicos. Para obtener más información, consulte [Permisos y políticas](#) en la Guía del usuario de IAM. Para obtener instrucciones sobre cómo gestionar y crear políticas de IAM personalizadas, consulte [Gestión de políticas de IAM](#).

Los siguientes ejemplos muestran instrucciones de política que puede utilizar para controlar los permisos que los usuarios de tienen en AWS Batch.

Ejemplos

- [Recurso: Acceso de solo lectura para AWS Batch](#)
- [Recurso: restricción del envío de un trabajo según el usuario de POSIX, la imagen de Docker el nivel de privilegios y el rol en el envío del trabajo](#)
- [Recurso: restricción del envío de un trabajo según el prefijo de la definición de trabajo](#)
- [Recurso: restricción de una cola de trabajos](#)
- [Denegación de la acción cuando todas las condiciones coincidan con las cadenas](#)
- [Recurso: denegación de la acción cuando alguna clave de condición coincida con una cadena](#)
- [Recurso: uso de la clave de condición `batch:ShareIdentifier`](#)
- [Gestione los recursos de SageMaker IA con AWS Batch](#)

- [Recurso: restrinja el envío de trabajos mediante etiquetas de recursos en la definición del trabajo y en la cola de trabajos](#)

Recurso: Acceso de solo lectura para AWS Batch

La siguiente política otorga a los usuarios permisos para usar todas las acciones de la AWS Batch API cuyo nombre comience por `y`. `Describe List`

A menos que otra instrucción les otorgue permiso para hacerlo, los usuarios no tienen permiso para llevar a cabo ninguna acción en los recursos. De forma predeterminada, se les niega el permiso para usar las acciones de la API.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "batch:Describe*",
        "batch:List*",
        "batch:Get*"
      ],
      "Resource": "*"
    }
  ]
}
```

Recurso: restricción del envío de un trabajo según el usuario de POSIX, la imagen de Docker el nivel de privilegios y el rol en el envío del trabajo

La siguiente política permite a un usuario POSIX administrar su propio conjunto de definiciones de trabajo restringidas.

Utilice la primera y la segunda sentencia para registrar y anular el registro de cualquier nombre de definición de trabajo cuyo nombre lleve el prefijo. *JobDefA_*

La primera instrucción también utiliza claves de contexto de condición para restringir los valores de usuario de POSIX, estado de privilegio e imagen de contenedor en las `containerProperties` de una definición de trabajo. Para obtener más información, consulta [RegisterJobDefinition](#) en la AWS Batch Referencia de la API de . En este ejemplo, las definiciones de trabajo solo se pueden registrar cuando el usuario POSIX está configurado en `nobody`. El indicador privilegiado está establecido en `false`. Por ejemplo, el siguiente comando muestra las herramientas `myImage` de un repositorio de Amazon ECR.

Important

Docker resuelve el parámetro `user` de ese usuario `uid` desde la imagen del contenedor. En la mayoría de los casos, se encuentra en el archivo `/etc/passwd` de la imagen del contenedor. Esta resolución de nombres se puede evitar si usan valores de `uid` directos tanto en la definición de trabajo como en las políticas de IAM asociadas. Tanto las operaciones de API de AWS Batch como las claves condicionales de IAM `batch:User` admiten valores numéricos.

Uso de la tercera instrucción para restringir únicamente un rol específico a una definición de trabajo.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "batch:RegisterJobDefinition"
      ],
      "Resource": [
        "arn:aws:batch:us-east-2:999999999999:job-definition/JobDefA_*"
      ],
      "Condition": {
        "StringEquals": {
          "batch:User": [
            "nobody"
          ],
          "batch:Image": [
            "999999999999.dkr.ecr.us-east-2.amazonaws.com/myImage"
          ]
        }
      }
    }
  ]
}
```

```

        ]
      },
      "Bool": {
        "batch:Privileged": "false"
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "batch:DeregisterJobDefinition"
      ],
      "Resource": [
        "arn:aws:batch:us-east-2:999999999999:job-definition/JobDefA_*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iam:PassRole"
      ],
      "Resource": [
        "arn:aws:iam::999999999999:role/MyBatchJobRole"
      ]
    }
  ]
}

```

Recurso: restricción del envío de un trabajo según el prefijo de la definición de trabajo

Utilice la siguiente política para enviar trabajos a cualquier cola de trabajos con cualquier nombre de definición de trabajo que comience por. *JobDefA*

Important

Al determinar el ámbito del acceso en el nivel de recursos para el envío de trabajos, debe proporcionar los tipos de recursos de cola de trabajos y de definición de trabajo.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "batch:SubmitJob"
      ],
      "Resource": [
        "arn:aws:batch:us-east-2:111122223333:job-definition/JobDefA_*",
        "arn:aws:batch:us-east-2:111122223333:job-queue/*"
      ]
    }
  ]
}
```

Recurso: restricción de una cola de trabajos

Uso de la siguiente política para enviar trabajos a una cola de trabajos denominada queue1 con cualquier nombre de definición de trabajo.

 Important

Al determinar el ámbito del acceso en el nivel de recursos para el envío de trabajos, debe proporcionar los tipos de recursos de cola de trabajos y de definición de trabajo.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "batch:SubmitJob"
      ],
```

```

        "Resource": [
            "arn:aws:batch:us-east-2:888888888888:job-definition/*",
            "arn:aws:batch:us-east-2:888888888888:job-queue/queue1"
        ]
    }
}

```

Denegación de la acción cuando todas las condiciones coincidan con las cadenas

La siguiente política deniega el acceso a la operación de la [RegisterJobDefinition](#) API cuando la clave de condición `batch:Image` (ID de imagen del contenedor) es "*string1*" y la clave de condición `batch:LogDriver` (controlador del registro del contenedor) es "*string2*". AWS Batch evalúa las claves de condición de cada contenedor. Cuando un trabajo abarca varios contenedores, como un trabajo paralelo de varios nodos, es posible que los contenedores tengan configuraciones diferentes. Si se evalúan varias claves de condición en una instrucción, se combinan mediante la lógica AND. Por lo tanto, si alguna de las múltiples claves de condición no coincide con un contenedor, el efecto Deny no se aplica a ese contenedor. Por el contrario, es posible que se deniegue un contenedor diferente en el mismo trabajo.

Para ver la lista de claves de condición AWS Batch, consulte las [claves de condición AWS Batch en la](#) Referencia de autorización de servicio. A excepción de `batch:ShareIdentifier`, todas las claves de condición `batch` se pueden usar de esta manera. La clave de condición `batch:ShareIdentifier` se define para un trabajo, no para una definición de trabajo.

JSON

```

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

```

```
{
  "Effect": "Deny",
  "Action": "batch:RegisterJobDefinition",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "batch:Image": "string1",
      "batch:LogDriver": "string2"
    }
  }
}
```

Recurso: denegación de la acción cuando alguna clave de condición coincida con una cadena

La siguiente política deniega el acceso a la operación de la [RegisterJobDefinition](#) API cuando la clave de condición `batch:Image` (ID de imagen del contenedor) es "*string1*" o la clave de condición `batch:LogDriver` (controlador del registro del contenedor) es "*string2*". Cuando un trabajo abarca varios contenedores, como un trabajo paralelo de varios nodos, es posible que los contenedores tengan configuraciones diferentes. Si se evalúan varias claves de condición en una instrucción, se combinan mediante la lógica AND. Por lo tanto, si alguna de las múltiples claves de condición no coincide con un contenedor, el efecto Deny no se aplica a ese contenedor. Por el contrario, es posible que se deniegue un contenedor diferente en el mismo trabajo.

Para ver la lista de claves de condición AWS Batch, consulte [Claves de condición AWS Batch en la Referencia](#) de autorización de servicios. A excepción de `batch:ShareIdentifier`, todas las claves de condición `batch` se pueden usar de esta manera. (La clave de condición `batch:ShareIdentifier` se define para un trabajo, no para una definición de trabajo).

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
```

```
    "batch:RegisterJobDefinition"
  ],
  "Resource": [
    "*"
  ]
},
{
  "Effect": "Deny",
  "Action": [
    "batch:RegisterJobDefinition"
  ],
  "Resource": [
    "*"
  ],
  "Condition": {
    "StringEquals": {
      "batch:Image": [
        "string1"
      ]
    }
  }
},
{
  "Effect": "Deny",
  "Action": [
    "batch:RegisterJobDefinition"
  ],
  "Resource": [
    "*"
  ],
  "Condition": {
    "StringEquals": {
      "batch:LogDriver": [
        "string2"
      ]
    }
  }
}
]
```

Recurso: uso de la clave de condición **batch:ShareIdentifier**

Utilice la siguiente política para enviar los trabajos que utilizan la definición de trabajo jobDefA con la cola de trabajos jobqueue1 con el identificador compartido lowCpu.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "batch:SubmitJob"
      ],
      "Resource": [
        "arn:aws:batch:us-east-2:555555555555:job-definition/JobDefA",
        "arn:aws:batch:us-east-2:555555555555:job-queue/jobqueue1"
      ],
      "Condition": {
        "StringEquals": {
          "batch:ShareIdentifier": [
            "lowCpu"
          ]
        }
      }
    }
  ]
}
```

Gestione los recursos de SageMaker IA con AWS Batch

Esta política permite AWS Batch gestionar los recursos de SageMaker IA.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```



```

        "Effect": "Allow",
        "Action": [
            "batch:*"
        ],
        "Resource": "*"
    },
    {
        "Effect": "Allow",
        "Action": [
            "iam:CreateServiceLinkedRole"
        ],
        "Resource": "arn:aws:iam::*:role/*AWSServiceRoleForAWSBatchWithSagemaker",
        "Condition": {
            "StringEquals": {
                "iam:AWSServiceName": "sagemaker-queuing.batch.amazonaws.com"
            }
        }
    },
    {
        "Effect": "Allow",
        "Action": "iam:PassRole",
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "iam:PassedToService": [
                    "sagemaker.amazonaws.com"
                ]
            }
        }
    }
]
}

```

Recurso: restrinja el envío de trabajos mediante etiquetas de recursos en la definición del trabajo y en la cola de trabajos

Utilice la siguiente política para enviar trabajos solo cuando la cola de trabajos tenga la etiqueta `Environment=dev` y la definición de trabajos la tenga. `Project=calc` Esta política muestra cómo

utilizar las etiquetas de recursos para controlar el acceso a AWS Batch los recursos durante el envío de los trabajos.

 Important

Al enviar trabajos con políticas que evalúen las etiquetas de recursos de las definiciones de trabajos, debe enviar los trabajos utilizando el formato de revisión de la definición de trabajos (`job-definition:revision`). Si envía trabajos sin especificar una revisión, las etiquetas de definición de trabajos no se evaluarán, lo que podría eludir los controles de acceso previstos. El `*:*` patrón del ARN del recurso exige que las presentaciones incluyan una revisión, lo que garantiza que las políticas de etiquetas se apliquen siempre de manera efectiva.

Esta política usa dos declaraciones independientes porque aplica diferentes condiciones de etiqueta a diferentes tipos de recursos. Al determinar el ámbito del acceso en el nivel de recursos para el envío de trabajos, debe proporcionar los tipos de recursos de cola de trabajos y de definición de trabajo.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "batch:SubmitJob",
      "Resource": "arn:aws:batch:*:*:job-queue/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Environment": "dev"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": "batch:SubmitJob",
      "Resource": "arn:aws:batch:*:*:job-definition/*:*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Project": "calc"
        }
      }
    }
  ]
}
```

```
}  
]  
}
```

Recurso: política AWS Batch gestionada

AWS Batch proporciona una política gestionada que puede adjuntar a los usuarios. Esta política otorga permiso para usar AWS Batch los recursos y las operaciones de la API. Puede aplicar esta política directamente o utilizarla como punto de partida para crear sus propias políticas. Para obtener más información sobre cada operación de la API mencionada en estas políticas, consulte [Acciones](#) en la Referencia de la API AWS Batch .

AWSBatchFullAccess

Esta política permite el acceso total del administrador a AWS Batch.

Para ver el JSON de la política, consulte [AWSBatchFullAccess](#) la [Guía de referencia de políticas AWS administradas](#).

AWS Batch Función de ejecución de IAM

La función de ejecución otorga al contenedor y a los AWS Fargate agentes de Amazon ECS permiso para realizar llamadas a la AWS API en su nombre.

Note

El rol de ejecución de tareas es compatible con la versión 1.16.0 y posteriores del agente de contenedor de Amazon ECS.

El rol de ejecución de IAM es necesario en función de los requisitos de la tarea. Puede tener varios roles de ejecución de tareas asociados a su cuenta para distintos fines y servicios.

Note

Para obtener más información sobre el rol de instancia de Amazon ECS, consulte [Rol de instancia de Amazon ECS](#). Para obtener más información acerca de los roles de servicio, consulte [¿Cómo AWS Batch funciona con IAM](#).

Amazon ECS proporciona la política administrada AmazonECSTaskExecutionRolePolicy. Esta política contiene los permisos necesarios para los casos de uso comunes descritos anteriormente. Puede ser necesario agregar políticas en línea al rol de ejecución de tareas para los casos de uso especiales que se describen a continuación.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecr:GetAuthorizationToken",
        "ecr:BatchCheckLayerAvailability",
        "ecr:GetDownloadUrlForLayer",
        "ecr:BatchGetImage",
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": "*"
    }
  ]
}
```

Se admiten permisos a nivel de recursos para AWS Batch las acciones de la API

El término permisos a nivel de recursos se refiere a la capacidad de especificar los recursos en los que los usuarios pueden realizar acciones. AWS Batch admite parcialmente los permisos a nivel de recursos. En el caso de algunas AWS Batch acciones, puede controlar cuándo se permite a los usuarios utilizarlas en función de las condiciones que deben cumplirse. También puede controlar en función de los recursos específicos que los usuarios pueden usar. Por ejemplo, puede conceder a los usuarios permisos para enviar trabajos, pero solo a una cola de trabajos específica y únicamente con una definición de trabajo determinada.

Para obtener más información sobre las acciones y los tipos de recursos definidos por AWS Batch, incluido el ARNs formato de cada uno de los tipos de recursos, consulte Acciones, recursos y claves de condición [AWS Batch](#) en la Referencia de autorización de servicio.

Tutorial: creación del rol de ejecución de IAM

Si su cuenta aún no tiene un rol de ejecución de IAM, siga estos pasos para crear el rol.

1. Abra la consola de IAM en <https://console.aws.amazon.com/iam/>.
2. Seleccione Roles en el panel de navegación.
3. Elija Creación de rol.
4. En Tipo de entidad de confianza, elija Servicio de AWS.
5. En Servicio o caso de uso, elija Elastic Container Service. Luego elija nuevamente Tarea de Elastic Container Service.
6. Elija Siguiente.
7. Para ver las políticas de permisos, busca Amazon ECSTask ExecutionRolePolicy.
8. Selecciona la casilla de verificación situada a la izquierda de la ECSTask ExecutionRolePolicy política de Amazon y, a continuación, selecciona Siguiente.
9. En Nombre de rol, escriba ecsTaskExecutionRole y luego elija Creación de rol.

Tutorial: comprobación del rol de ejecución de IAM

Utilice el procedimiento siguiente para comprobar si la cuenta ya dispone del rol de ejecución de IAM y para asociar la política de IAM gestionada en caso necesario.

1. Abra la consola de IAM en <https://console.aws.amazon.com/iam/>.
2. Seleccione Roles en el panel de navegación.
3. En la lista de roles, busque ecsTaskExecutionRole. Si no puede encontrar el rol, consulte [Tutorial: creación del rol de ejecución de IAM](#). Si el rol existe, selecciónelo para ver sus políticas asociadas.
4. En la pestaña Permisos, comprueba que la política ECSTask ExecutionRolePolicy gestionada por Amazon esté asociada al rol. Si la política se ha asociado, el rol de ejecución de la tarea está configurado correctamente. En caso contrario, siga los pasos derivados a continuación para asociar la política.
 - a. Elija Agregar permisos y luego Adjuntar políticas.
 - b. Busca Amazon ECSTask ExecutionRolePolicy.
 - c. Marca la casilla situada a la izquierda de la ECSTask ExecutionRolePolicy política de Amazon y selecciona Adjuntar políticas.

5. Seleccione Trust Relationships.
6. Verifique que la relación de confianza contiene la siguiente política. Si la relación de confianza coincide con la política siguiente, el rol está configurado correctamente. Si la relación de confianza no coincide, elija Editar la política de confianza, introduzca lo siguiente y después elija Actualizar política.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "ecs-tasks.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Utilizar funciones vinculadas al servicio para AWS Batch

AWS Batch [usa roles vinculados al AWS Identity and Access Management servicio \(IAM\)](#). Un rol vinculado a un servicio es un tipo único de rol de IAM al que se vincula directamente. AWS Batch Los roles vinculados al servicio están predefinidos AWS Batch e incluyen todos los permisos que el servicio requiere para llamar a otros AWS servicios en su nombre.

AWS Batch utiliza dos funciones diferentes vinculadas al servicio:

- [AWSServiceRoleForBatch](#)- Para AWS Batch operaciones que incluyen entornos de cómputo.
- [AWSServiceRoleForAWSBatchWithSagemaker](#)- Para la gestión de cargas de trabajo y la creación de colas mediante SageMaker IA.

Temas

- [Uso de roles para AWS Batch](#)

- [Uso de roles para la IA AWS Batch SageMaker](#)

Uso de roles para AWS Batch

AWS Batch [usa roles vinculados al AWS Identity and Access Management servicio \(IAM\)](#). Un rol vinculado a un servicio es un tipo único de rol de IAM al que se vincula directamente. AWS Batch Los roles vinculados al servicio están predefinidos AWS Batch e incluyen todos los permisos que el servicio requiere para llamar a otros AWS servicios en su nombre.

Un rol vinculado a un servicio facilita la configuración AWS Batch , ya que no es necesario añadir manualmente los permisos necesarios. AWS Batch define los permisos de sus funciones vinculadas al servicio y, a menos que se defina lo contrario, solo AWS Batch puede asumir sus funciones. Los permisos definidos incluyen las políticas de confianza y de permisos, y que la política de permisos no se pueda asociar a ninguna otra entidad de IAM.

Note

Realice una de las siguientes acciones para especificar una función de servicio para un entorno AWS Batch informático.

- Utilice una cadena vacía para el rol de servicio. Esto permite AWS Batch crear el rol de servicio.
- Especifique el rol de servicio en el siguiente formato:
`arn:aws:iam::account_number:role/aws-service-role/
batch.amazonaws.com/AWSServiceRoleForBatch.`

Para obtener más información, consulte [Nombre de rol o ARN incorrectos](#) la Guía AWS Batch del usuario.

Solo es posible eliminar un rol vinculado a un servicio después de eliminar sus recursos relacionados. Esto protege sus AWS Batch 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 Funciones vinculadas a servicios. Elija una opción Sí con un enlace para ver la documentación acerca del rol vinculado al servicio en cuestión.

Permisos de roles vinculados al servicio para AWS Batch

AWS Batch utiliza el rol vinculado al servicio denominado `AWSServiceRoleForBatch`: permite AWS Batch crear y administrar AWS recursos en su nombre.

El rol `AWSServiceRoleForBatch` vinculado al servicio confía en los siguientes servicios para asumir el rol:

- `batch.amazonaws.com`

La política de permisos de roles denominada [BatchServiceRolePolicy](#) permite AWS Batch realizar las siguientes acciones en los recursos especificados:

- `autoscaling`— Permite AWS Batch crear y gestionar los recursos de Amazon EC2 Auto Scaling. AWS Batch crea y administra grupos de Amazon EC2 Auto Scaling para la mayoría de los entornos informáticos.
- `ec2`— Permite AWS Batch controlar el ciclo de vida de las instancias de Amazon EC2, así como crear y gestionar plantillas y etiquetas de lanzamiento. AWS Batch crea y gestiona las solicitudes de EC2 Spot Fleet para algunos entornos de computación puntual de EC2.
- `ecs`- Permite AWS Batch crear y gestionar clústeres de Amazon ECS, definiciones de tareas y tareas para la ejecución de trabajos.
- `eks`- Permite AWS Batch describir el recurso del clúster Amazon EKS para las validaciones.
- `iam`- Permite AWS Batch validar y transferir las funciones proporcionadas por el propietario a Amazon EC2, Amazon EC2 Auto Scaling y Amazon ECS.
- `logs`— Permite AWS Batch crear y gestionar grupos de registros y flujos de registros para AWS Batch trabajos.

Debe configurar los permisos para permitir a sus usuarios, grupos o funciones, crear, editar o eliminar la descripción de un rol vinculado al servicio. Para obtener más información, consulte [Permisos de roles vinculados a servicios](#) en la Guía del usuario de IAM.

Crear un rol vinculado a un servicio para AWS Batch

No necesita crear manualmente un rol vinculado a servicios. Al crear un entorno informático en la Consola de administración de AWS, la o la AWS API AWS CLI, se AWS Batch crea automáticamente el rol vinculado al servicio.

 Important

Este rol vinculado a servicios puede aparecer en su cuenta si se ha completado una acción en otro servicio que utilice las características compatibles con este rol. Si utilizabas el AWS Batch servicio antes del 10 de marzo de 2021, cuando comenzó a admitir funciones vinculadas al servicio, entonces AWS Batch creaste la AWSService RoleForBatch función en tu cuenta. Para obtener más información, consulte [Apareció un nuevo puesto en mi](#). Cuenta de AWS

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 crear un entorno informático, vuelve a AWS Batch crear el rol vinculado al servicio para usted.

Edición de un rol vinculado a un servicio para AWS Batch

AWS Batch no permite editar el rol vinculado al AWSService RoleForBatch servicio. Después de crear un rol vinculado al servicio, no podrá cambiar el nombre del rol, ya que varias entidades podrían hacer referencia al rol. Sin embargo, sí puede editar la descripción del rol con IAM. Para obtener más información, consulte [Editar un rol vinculado a servicios](#) en la Guía del usuario de IAM.

Para permitir que una entidad de IAM edite la descripción del rol vinculado al servicio AWSService RoleForBatch

Añada la siguiente instrucción a la política de permisos. Esto permite a una entidad de IAM editar la descripción del rol vinculado a servicio.

```
{
  "Effect": "Allow",
  "Action": [
    "iam:UpdateRoleDescription"
  ],
  "Resource": "arn:aws:iam::*:role/aws-service-role/batch.amazonaws.com/
  AWSServiceRoleForBatch",
  "Condition": {"StringLike": {"iam:AWSServiceName": "batch.amazonaws.com"}}
}
```

Eliminar un rol vinculado a un servicio para AWS Batch

Si ya no necesita utilizar una característica o servicio que requiere un rol vinculado a un servicio, le recomendamos que elimine dicho rol. De esta forma no tiene una entidad no utilizada que no se

monitoree ni mantenga de forma activa. Sin embargo, debe limpiar el rol vinculado a servicios antes de eliminarlo manualmente.

Para permitir que una entidad de IAM elimine el rol vinculado al servicio AWSService RoleForBatch

Añada la siguiente instrucción a la política de permisos. Esto permite a una entidad de IAM eliminar un rol vinculado a un servicio.

```
{
  "Effect": "Allow",
  "Action": [
    "iam:DeleteServiceLinkedRole",
    "iam:GetServiceLinkedRoleDeletionStatus"
  ],
  "Resource": "arn:aws:iam::*:role/aws-service-role/batch.amazonaws.com/
  AWSServiceRoleForBatch",
  "Condition": {"StringLike": {"iam:AWSServiceName": "batch.amazonaws.com"}}
}
```

Saneamiento de un rol vinculado a servicios

Antes de poder usar IAM para eliminar un rol vinculado a un servicio, primero debe confirmar que el rol no tiene sesiones activas y eliminar todos los entornos de AWS Batch procesamiento que utilizan el rol en todas AWS las regiones en una sola partición.

Para comprobar si el rol vinculado a servicio tiene una sesión activa

1. Abra la consola de IAM en <https://console.aws.amazon.com/iam/>.
2. En el panel de navegación, elija Funciones y, a continuación, el AWSService RoleForBatch nombre (no la casilla de verificación).
3. En la página Summary, elija Access Advisor y revise la actividad reciente del rol vinculado a servicio.

Note

Si no sabe si AWS Batch está utilizando el AWSService RoleForBatch rol, puede intentar eliminarlo. Si el servicio está utilizando el rol, este no podrá eliminarse. Puede ver las regiones en las que se está utilizando el rol. Si el rol se está utilizando, debe esperar que la sesión finalice para poder eliminarlo. No se puede revocar la sesión de un rol vinculado a un servicio.

Para eliminar AWS Batch los recursos utilizados por el rol AWSService RoleForBatch vinculado al servicio

Debe eliminar todos los entornos AWS Batch informáticos que utilizan el AWSService RoleForBatch rol en todas AWS las regiones antes de poder eliminar el AWSService RoleForBatch rol.

1. Abra la AWS Batch consola en <https://console.aws.amazon.com/batch/>.
2. En la barra de navegación, seleccione la región a utilizar.
3. En el panel de navegación, elija Entornos de computación.
4. Seleccione el entorno de computación.
5. Elija Deshabilitar. Espere a que Estado cambie a DESHABILITADO.
6. Seleccione el entorno de computación.
7. Elija Eliminar. Confirme que desea eliminar el entorno de computación mediante la elección de Eliminar entorno de computación.
8. Repita los pasos 1 a 7 para todos los entornos de computación que utilizan la función vinculada al servicio en todas las regiones.

Eliminación de un rol vinculado a un servicio en IAM (Consola)

Puede utilizar la consola de IAM para eliminar un rol vinculado a un servicio.

Para eliminar un rol vinculado a un servicio (consola)

1. Inicie sesión en la consola de IAM Consola de administración de AWS y ábrala en <https://console.aws.amazon.com/iam/>.
2. En el panel de navegación de la consola de IAM, elija Roles. A continuación, active la casilla de verificación situada junto a ella AWSServiceRoleForBatch, no el nombre o la fila en sí.
3. Elija Eliminar rol.
4. En el cuadro de diálogo de confirmación, revise los datos del último acceso al servicio, que muestra cuándo cada uno de los roles seleccionados tuvo acceso a un Servicio de AWS por última vez. Esto lo ayuda a confirmar si el rol está actualmente activo. Si desea continuar, seleccione Yes, Delete para enviar la solicitud de eliminación del rol vinculado al servicio.
5. Consulte las notificaciones de la consola de IAM para monitorear el progreso de la eliminación del rol vinculado al servicio. Como el proceso de eliminación del rol vinculado al servicio de IAM es asíncrono, dicha tarea puede realizarse correctamente o fallar después de que envía la solicitud de eliminación.

- Si la tarea se realiza correctamente, el rol se elimina de la lista y aparece una notificación informando de ello en la parte superior de la página.
- Si la tarea no se realiza correctamente, puede seleccionar View details (Ver detalles) o View Resources (Ver recursos) desde las notificaciones para obtener información sobre el motivo por el que no se pudo eliminar el rol. Si la eliminación no pudo producirse porque el rol está utilizando los recursos del servicio, la notificación incluye una lista de dichos recursos si el servicio proporciona dicha información. Tras conocer esa información, podrá [limpiar los recursos](#) y volver a enviar la solicitud de eliminación.

 Note

Es posible que tenga que repetir este proceso varias veces, en función de la información que devuelva el servicio. Por ejemplo, el rol vinculado al servicio podría estar utilizando seis recursos y el servicio podría estar devolviendo información solo acerca de cinco de ellos. Si limpia los cinco recursos y envía la solicitud de eliminación del rol de nuevo, se producirá un error y el servicio informará del recurso restante. Un servicio podría informar de todos los recursos, algunos o ninguno.

- Si se produce un error en la tarea y la notificación no incluye una lista de los recursos, el servicio no podría devolver dicha información. Para obtener información acerca de cómo limpiar los recursos de ese servicio, consulte [Servicios de AWS que funcionan con IAM](#). Identifique su servicio en la tabla y haga clic en el enlace Yes (Sí) para consultar la documentación relacionada con los roles vinculados a dicho servicio.

Eliminación de un rol vinculado a un servicio en IAM (AWS CLI)

Puede utilizar los comandos de IAM de AWS Command Line Interface para eliminar un rol vinculado a un servicio.

Para eliminar un rol vinculado a un servicio (CLI)

1. Como los roles vinculados a servicios no se pueden eliminar si están en uso o tienen recursos asociados, debe enviar una solicitud de eliminación. Esta solicitud puede denegarse si no se cumplen estas condiciones. Debe apuntar el valor `deletion-task-id` de la respuesta para comprobar el estado de la tarea de eliminación. Ingrese el siguiente comando para enviar una solicitud de eliminación de un rol vinculado a un servicio:

```
$ aws iam delete-service-linked-role --role-name AWSServiceRoleForBatch
```

2. Utilice el siguiente comando para comprobar el estado de la tarea de eliminación:

```
$ aws iam get-service-linked-role-deletion-status --deletion-task-id deletion-task-id
```

El estado de la tarea de eliminación puede ser NOT_STARTED, IN_PROGRESS, SUCCEEDED o FAILED. Si ocurre un error durante la eliminación, la llamada devuelve el motivo del error para que pueda resolver el problema. Si la eliminación no pudo producirse porque el rol está utilizando los recursos del servicio, la notificación incluye una lista de dichos recursos si el servicio proporciona dicha información. Tras conocer esa información, podrá [limpiar los recursos](#) y volver a enviar la solicitud de eliminación.

Note

Es posible que tenga que repetir este proceso varias veces, en función de la información que devuelva el servicio. Por ejemplo, el rol vinculado al servicio podría estar utilizando seis recursos y el servicio podría estar devolviendo información solo acerca de cinco de ellos. Si limpia los cinco recursos y envía la solicitud de eliminación del rol de nuevo, se producirá un error y el servicio informará del recurso restante. Un servicio podría devolver todos los recursos, algunos de ellos. O bien, podría no reportar ningún recurso. Para obtener información acerca de cómo limpiar los recursos de un servicio que no está informando de ningún recurso, consulte los [Servicios de AWS que funcionan con IAM](#). Identifique su servicio en la tabla y haga clic en el enlace Yes (Sí) para consultar la documentación relacionada con los roles vinculados a dicho servicio.

Eliminación de un rol vinculado a servicio en IAM (API de AWS)

Puede utilizar la API de IAM para eliminar un rol vinculado a un servicio.

Para eliminar un rol vinculado a un servicio (API)

1. Para enviar una solicitud de eliminación de un rol vinculado a un servicio, llame a [DeleteServiceLinkedRole](#). En la solicitud, especifique el nombre del AWSService RoleForBatch rol.

Como los roles vinculados a servicios no se puede eliminar si están en uso o tienen recursos asociados, debe enviar una solicitud de eliminación. Esta solicitud puede denegarse si no se cumplen estas condiciones. Debe apuntar el valor `DeletionTaskId` de la respuesta para comprobar el estado de la tarea de eliminación.

2. Para comprobar el estado de la tarea de eliminación, realice una llamada a [GetServiceLinkedRoleDeletionStatus](#). En la solicitud, especifique el valor de `DeletionTaskId`.

El estado de la tarea de eliminación puede ser `NOT_STARTED`, `IN_PROGRESS`, `SUCCEEDED` o `FAILED`. Si ocurre un error durante la eliminación, la llamada devuelve el motivo del error para que pueda resolver el problema. Si la eliminación no pudo producirse porque el rol está utilizando los recursos del servicio, la notificación incluye una lista de dichos recursos si el servicio proporciona dicha información. Tras conocer esa información, podrá [limpiar los recursos](#) y volver a enviar la solicitud de eliminación.

Note

Es posible que tenga que repetir este proceso varias veces, en función de la información que devuelva el servicio. Por ejemplo, el rol vinculado al servicio podría estar utilizando seis recursos y el servicio podría estar devolviendo información solo acerca de cinco de ellos. Si limpia los cinco recursos y envía la solicitud de eliminación del rol de nuevo, se producirá un error y el servicio informará del recurso restante. Un servicio podría informar de todos los recursos, algunos o ninguno. Para obtener información acerca de cómo limpiar los recursos de un servicio que no está informando de ningún recurso, consulte [Servicios de AWS que funcionan con IAM](#). Identifique su servicio en la tabla y haga clic en el enlace Yes (Sí) para consultar la documentación relacionada con los roles vinculados a dicho servicio.

Regiones compatibles para los roles AWS Batch vinculados al servicio

AWS Batch 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 enlace de AWS Batch](#).

Uso de roles para la IA AWS Batch SageMaker

AWS Batch [usa roles vinculados al AWS Identity and Access Management servicio \(IAM\)](#). Un rol vinculado a un servicio es un tipo único de rol de IAM al que se vincula directamente. AWS Batch

Los roles vinculados al servicio están predefinidos AWS Batch e incluyen todos los permisos que el servicio requiere para llamar a otros AWS servicios en su nombre.

Un rol vinculado a un servicio facilita la configuración AWS Batch , ya que no es necesario añadir manualmente los permisos necesarios. AWS Batch define los permisos de sus funciones vinculadas al servicio y, a menos que se defina lo contrario, solo AWS Batch puede asumir sus funciones. Los permisos definidos incluyen las políticas de confianza y de permisos, y que la política de permisos no se pueda adjuntar 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 Batch 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 Funciones vinculadas a servicios. Elija una opción Sí con un enlace para ver la documentación acerca del rol vinculado al servicio en cuestión.

Permisos de roles vinculados al servicio para AWS Batch

AWS Batch utiliza el rol vinculado al servicio denominado `AWSServiceRoleForAWSBatchWithSagemaker`: permite poner en cola y gestionar AWS Batch los trabajos de SageMaker formación en su nombre.

El rol `AWSService RoleFor AWSBatch WithSagemaker` vinculado al servicio confía en los siguientes servicios para asumir el rol:

- `sagemaker-queuing.batch.amazonaws.com`

La política de permisos del rol permite AWS Batch realizar las siguientes acciones en los recursos especificados:

- `sagemaker`— Permite AWS Batch gestionar los trabajos de SageMaker formación, transformar los puestos de trabajo y otros recursos de SageMaker IA.
- `iam:PassRole`— Permite AWS Batch transferir las funciones de ejecución definidas por el cliente a la SageMaker IA para la ejecución de las tareas. La limitación de recursos permite transferir funciones a SageMaker los servicios de IA.

Debe configurar los permisos para permitir a sus usuarios, grupos o funciones, crear, editar o eliminar la descripción de un rol vinculado al servicio. Para obtener más información, consulte [Permisos de roles vinculados a servicios](#) en la Guía del usuario de IAM.

Crear un rol vinculado a un servicio para AWS Batch

No necesita crear manualmente un rol vinculado a servicios. Al crear un entorno de servicios mediante `CreateServiceEnvironment` la Consola de administración de AWS, la o la AWS API AWS CLI, se AWS Batch crea automáticamente el rol vinculado al servicio.

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. Cuando crea un entorno de servicios mediante `CreateServiceEnvironment`, AWS Batch vuelve a crear el rol vinculado al servicio para usted.

Para ver el JSON de la política, consulta la Guía [AWSBatchServiceRolePolicyForSageMaker](#) de [referencia de políticas AWS administradas](#).

Edición de un rol vinculado a un servicio para AWS Batch

AWS Batch no permite editar el rol vinculado al `AWSServiceRoleForAWSBatchWithSageMaker` servicio. Después de crear un rol vinculado al servicio, no podrá cambiar el nombre del rol, ya que varias entidades podrían hacer referencia al rol. Sin embargo, sí puede editar la descripción del rol con IAM. Para obtener más información, consulte [Editar un rol vinculado a servicios](#) en la Guía del usuario de IAM.

Eliminar un rol vinculado a un servicio para AWS Batch

Si ya no necesita utilizar una característica o servicio que requiere un rol vinculado a un servicio, le recomendamos que elimine dicho rol. De esta forma no tiene una entidad no utilizada que no se monitoree ni mantenga de forma activa. Sin embargo, debe limpiar el rol vinculado a servicios antes de eliminarlo manualmente.

Limpiar un rol vinculado a servicios

Antes de poder usar IAM para eliminar un rol vinculado a un servicio, primero debe confirmar que el rol no tiene sesiones activas y eliminar todos los entornos de servicio que utilizan el rol en todas AWS las regiones en una sola partición.

Para comprobar si el rol vinculado a servicio tiene una sesión activa

1. Abra la consola de IAM en <https://console.aws.amazon.com/iam/>.

2. En el panel de navegación, elija Funciones y, a continuación, el AWSService RoleFor AWSBatch WithSagemaker nombre (no la casilla de verificación).
3. En la página Summary, elija Access Advisor y revise la actividad reciente del rol vinculado a servicio.

 Note

Si no sabe si AWS Batch está utilizando el AWSService RoleFor AWSBatch WithSagemaker rol, puede intentar eliminarlo. Si el servicio está utilizando el rol, este no podrá eliminarse. Puede ver las regiones en las que se está utilizando el rol. Si el rol se está utilizando, debe esperar que la sesión finalice para poder eliminarlo. No se puede revocar la sesión de un rol vinculado a un servicio.

Para eliminar AWS Batch los recursos utilizados por el rol AWSService RoleFor AWSBatch WithSagemaker vinculado al servicio

Debe disociar todas las colas de trabajos de todos los entornos de servicio y, a continuación, eliminar todos los entornos de servicio que utilizan el AWSService RoleFor AWSBatch WithSagemaker rol en todas AWS las regiones antes de poder eliminar el rol. AWSService RoleFor AWSBatch WithSagemaker

1. Abra la consola en AWS Batch . <https://console.aws.amazon.com/batch/>
2. En la barra de navegación, seleccione la región a utilizar.
3. En el panel de navegación, elija Entornos y, a continuación, elija Entornos de servicios.
4. Seleccione todos los Entornos de servicio.
5. Elija Deshabilitar. Espere a que Estado cambie a DESHABILITADO.
6. Seleccione el entorno de servicio.
7. Elija Eliminar. Confirme que desea eliminar el entorno de servicio mediante la elección de Eliminar entorno de servicio.
8. Repita los pasos 1 a 7 para todos los entornos de servicio que utilizan el rol vinculado al servicio en todas las regiones.

Eliminación de un rol vinculado a un servicio en IAM (Consola)

Puede utilizar la consola de IAM para eliminar un rol vinculado a un servicio.

Para eliminar un rol vinculado a un servicio (consola)

1. Inicie sesión en la consola de IAM Consola de administración de AWS y ábrala en <https://console.aws.amazon.com/iam/>.
2. En el panel de navegación de la consola de IAM, elija Roles. A continuación, active la casilla de verificación situada junto a ella AWSService RoleFor AWSBatchWithSagemaker, no el nombre o la fila en sí.
3. Elija Eliminar rol.
4. En el cuadro de diálogo de confirmación, revise los datos del último acceso al servicio, que muestra cuándo cada uno de los roles seleccionados tuvo acceso a un Servicio de AWS por última vez. Esto lo ayuda a confirmar si el rol está actualmente activo. Si desea continuar, seleccione Yes, Delete para enviar la solicitud de eliminación del rol vinculado al servicio.
5. Consulte las notificaciones de la consola de IAM para monitorear el progreso de la eliminación del rol vinculado al servicio. Como el proceso de eliminación del rol vinculado al servicio de IAM es asíncrono, dicha tarea puede realizarse correctamente o fallar después de que envía la solicitud de eliminación.
 - Si la tarea se realiza correctamente, el rol se elimina de la lista y aparece una notificación informando de ello en la parte superior de la página.
 - Si la tarea no se realiza correctamente, puede seleccionar View details (Ver detalles) o View Resources (Ver recursos) desde las notificaciones para obtener información sobre el motivo por el que no se pudo eliminar el rol. Si la eliminación no pudo producirse porque el rol está utilizando los recursos del servicio, la notificación incluye una lista de dichos recursos si el servicio proporciona dicha información. Tras conocer esa información, podrá [limpiar los recursos](#) y volver a enviar la solicitud de eliminación.

Note

Es posible que tenga que repetir este proceso varias veces, en función de la información que devuelva el servicio. Por ejemplo, el rol vinculado al servicio podría estar utilizando seis recursos y el servicio podría estar devolviendo información solo acerca de cinco de ellos. Si limpia los cinco recursos y envía la solicitud de eliminación del rol de nuevo, se producirá un error y el servicio informará del recurso restante. Un servicio podría informar de todos los recursos, algunos o ninguno.

- Si se produce un error en la tarea y la notificación no incluye una lista de los recursos, el servicio no podría devolver dicha información. Para obtener información acerca de

cómo limpiar los recursos de ese servicio, consulte [Servicios de AWS que funcionan con IAM](#). Identifique su servicio en la tabla y haga clic en el enlace Yes (Sí) para consultar la documentación relacionada con los roles vinculados a dicho servicio.

Eliminación de un rol vinculado a un servicio en IAM (AWS CLI)

Puede utilizar los comandos de IAM de AWS Command Line Interface para eliminar un rol vinculado a un servicio.

Para eliminar un rol vinculado a un servicio (CLI)

1. Como los roles vinculados a servicios no se pueden eliminar si están en uso o tienen recursos asociados, debe enviar una solicitud de eliminación. Esta solicitud puede denegarse si no se cumplen estas condiciones. Debe apuntar el valor `deletion-task-id` de la respuesta para comprobar el estado de la tarea de eliminación. Ingrese el siguiente comando para enviar una solicitud de eliminación de un rol vinculado a un servicio:

```
$ aws iam delete-service-linked-role --role-name
AWSServiceRoleForAWSBatchWithSagemaker
```

2. Utilice el siguiente comando para comprobar el estado de la tarea de eliminación:

```
$ aws iam get-service-linked-role-deletion-status --deletion-task-id deletion-task-id
```

El estado de la tarea de eliminación puede ser `NOT_STARTED`, `IN_PROGRESS`, `SUCCEEDED` o `FAILED`. Si ocurre un error durante la eliminación, la llamada devuelve el motivo del error para que pueda resolver el problema. Si la eliminación no pudo producirse porque el rol está utilizando los recursos del servicio, la notificación incluye una lista de dichos recursos si el servicio proporciona dicha información. Tras conocer esa información, podrá [limpiar los recursos](#) y volver a enviar la solicitud de eliminación.

Note

Es posible que tenga que repetir este proceso varias veces, en función de la información que devuelva el servicio. Por ejemplo, el rol vinculado al servicio podría estar utilizando seis recursos y el servicio podría estar devolviendo información solo acerca de cinco de ellos. Si limpia los cinco recursos y envía la solicitud de eliminación del rol de nuevo,

se producirá un error y el servicio informará del recurso restante. Un servicio podría devolver todos los recursos, algunos de ellos. O bien, podría no reportar ningún recurso. Para obtener información acerca de cómo limpiar los recursos de un servicio que no está informando de ningún recurso, consulte los [Servicios de AWS que funcionan con IAM](#). Identifique su servicio en la tabla y haga clic en el enlace Yes (Sí) para consultar la documentación relacionada con los roles vinculados a dicho servicio.

Eliminación de un rol vinculado a servicio en IAM (API de AWS)

Puede utilizar la API de IAM para eliminar un rol vinculado a un servicio.

Para eliminar un rol vinculado a un servicio (API)

1. Para enviar una solicitud de eliminación de un rol vinculado a un servicio, llame a [DeleteServiceLinkedRole](#). En la solicitud, especifique el nombre del AWSService RoleFor AWSBatch WithSagemaker rol.

Como los roles vinculados a servicios no se puede eliminar si están en uso o tienen recursos asociados, debe enviar una solicitud de eliminación. Esta solicitud puede denegarse si no se cumplen estas condiciones. Debe apuntar el valor DeletionTaskId de la respuesta para comprobar el estado de la tarea de eliminación.

2. Para comprobar el estado de la tarea de eliminación, realice una llamada a [GetServiceLinkedRoleDeletionStatus](#). En la solicitud, especifique el valor de DeletionTaskId.

El estado de la tarea de eliminación puede ser NOT_STARTED, IN_PROGRESS, SUCCEEDED o FAILED. Si ocurre un error durante la eliminación, la llamada devuelve el motivo del error para que pueda resolver el problema. Si la eliminación no pudo producirse porque el rol está utilizando los recursos del servicio, la notificación incluye una lista de dichos recursos si el servicio proporciona dicha información. Tras conocer esa información, podrá [limpiar los recursos](#) y volver a enviar la solicitud de eliminación.

Note

Es posible que tenga que repetir este proceso varias veces, en función de la información que devuelva el servicio. Por ejemplo, el rol vinculado al servicio podría estar utilizando seis recursos y el servicio podría estar devolviendo información solo acerca de cinco de ellos. Si limpia los cinco recursos y envía la solicitud de eliminación del rol de nuevo,

se producirá un error y el servicio informará del recurso restante. Un servicio podría informar de todos los recursos, algunos o ninguno. Para obtener información acerca de cómo limpiar los recursos de un servicio que no está informando de ningún recurso, consulte [Servicios de AWS que funcionan con IAM](#). Identifique su servicio en la tabla y haga clic en el enlace Yes (Sí) para consultar la documentación relacionada con los roles vinculados a dicho servicio.

Regiones compatibles para los roles AWS Batch vinculados al servicio

AWS Batch 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 enlace de AWS Batch](#).

Rol de instancia de Amazon ECS

AWS Batch los entornos de cómputo se rellenan con instancias de contenedor de Amazon ECS. Ejecutan el agente de contenedor de Amazon ECS de forma local. El agente de contenedores de Amazon ECS realiza llamadas a varias operaciones de AWS API en su nombre. Por lo tanto, las instancias de contenedor que ejecutan el agente requieren una política de IAM y un rol para que estos servicios reconozcan que el agente le pertenece. Debe crear un rol de IAM y un perfil de instancia para que las instancias de contenedor los usen cuando se lancen. De lo contrario, no puede crear un entorno de computación y lanzar instancias de contenedor en él. Este requisito se aplica a las instancias de contenedor lanzadas con o sin la AMI optimizada de Amazon ECS proporcionada por Amazon. Para obtener más información, consulte [Rol de instancia de Amazon ECS](#) en la Guía para desarrolladores de Amazon Elastic Container Service.

Temas

- [Comprobación del rol de instancia de Amazon ECS en su cuenta](#)

Comprobación del rol de instancia de Amazon ECS en su cuenta

El perfil de instancia y el rol de instancia de Amazon ECS se crean automáticamente en la experiencia de primer uso de la consola. Sin embargo, puede seguir estos pasos para verificar si la cuenta ya dispone del rol y el perfil de instancia de Amazon ECS. En los siguientes pasos también se explica cómo adjuntar la política de IAM administrada.

Tutorial: comprobación del **ecsInstanceRole** en la consola de IAM

1. Abra la consola de IAM en <https://console.aws.amazon.com/iam/>.
2. Seleccione Roles en el panel de navegación.
3. En la lista de roles, busque `ecsInstanceRole`. Si el rol no existe, siga los siguientes pasos para crearlo.
 - a. Seleccione Creación de rol.
 - b. En Tipo de entidad de confianza, elija Servicio de AWS.
 - c. En Casos de uso comunes, elija EC2.
 - d. Elija Siguiente.
 - e. Para ver las políticas de permisos, busca Amazon EC2 ContainerServicefor EC2 Role.
 - f. Selecciona la casilla de verificación situada junto a Amazon EC2 ContainerServicefor EC2 Role y, a continuación, selecciona Siguiente.
 - g. En Nombre de rol, escriba `ecsInstanceRole` y elija Crear rol.

Note

Si utiliza el Consola de administración de AWS para crear un rol para Amazon EC2, la consola crea un perfil de instancia con el mismo nombre que el rol.

Como alternativa, puede usar el AWS CLI para crear el rol de `ecsInstanceRole` IAM. En el siguiente ejemplo, se crea un rol de IAM con una política de confianza y una política AWS gestionada.

Tutorial: creación de un rol de IAM y un perfil de instancias (AWS CLI)

1. Cree la siguiente política de confianza y guárdela en un archivo de texto que se denomina `ecsInstanceRole-role-trust-policy.json`.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```

```

    "Effect": "Allow",
    "Principal": { "Service": "ec2.amazonaws.com"},
    "Action": "sts:AssumeRole"
  }
]
}

```

- Utilice el comando [create-role](#) para crear el rol de servicio `ecsInstanceRole`. Especifique la ubicación del archivo de política de confianza en el parámetro `assume-role-policy-document`.

```

$ aws iam create-role \
  --role-name ecsInstanceRole \
  --assume-role-policy-document file://ecsInstanceRole-role-trust-policy.json

```

- Usa el [create-instance-profile](#) comando para crear un perfil de instancia con un nombre `ecsInstanceRole`.

Note

Debes crear roles y perfiles de instancia como acciones independientes en la AWS API AWS CLI y.

```

$ aws iam create-instance-profile --instance-profile-name ecsInstanceRole

```

A continuación, se muestra un ejemplo de respuesta.

```

{
  "InstanceProfile": {
    "Path": "/",
    "InstanceProfileName": "ecsInstanceRole",
    "InstanceProfileId": "AIPAT46P5RDITREXAMPLE",
    "Arn": "arn:aws:iam::123456789012:instance-profile/ecsInstanceRole",
    "CreateDate": "2022-06-30T23:53:34.093Z",
    "Roles": [],
  }
}

```

- Usa el comando [add-role-to-instance-profile](#) para agregar el `ecsInstanceRole` rol al perfil de la `ecsInstanceRole` instancia.

```
aws iam add-role-to-instance-profile \  
    --role-name ecsInstanceRole --instance-profile-name ecsInstanceRole
```

5. Usa el [attach-role-policy](#) comando para adjuntar la política AmazonEC2ContainerServiceforEC2Role AWS administrada al ecsInstanceRole rol.

```
$ aws iam attach-role-policy \  
    --policy-arn arn:aws:iam::aws:policy/service-role/  
AmazonEC2ContainerServiceforEC2Role \  
    --role-name ecsInstanceRole
```

Rol de flota de spot de Amazon EC2

Si crea un entorno de computación administrado que utiliza las instancias de flota de spot Amazon EC2, debe crear la política AmazonEC2SpotFleetTaggingRole. Esta política concede a la flota de spot permiso para lanzar, etiquetar y finalizar instancias en su nombre. Especifique el rol en la solicitud de flota de spot. También debe tener las funciones AWSServiceRoleForEC2Spot y AWSServiceRoleForEC2SpotFleet vinculadas a servicios para Amazon EC2 Spot y Spot Fleet. Utilice la siguiente instrucción para crear todos estos roles. Para obtener más información, consulte [Uso de roles vinculados a servicios](#) y [Creación de un rol para delegar permisos a un AWS servicio](#) en la Guía del usuario de IAM.

Temas

- [Cree funciones de flota puntual de Amazon EC2 en el Consola de administración de AWS](#)
- [Cree funciones de flota puntual en Amazon EC2 con AWS CLI](#)

Cree funciones de flota puntual de Amazon EC2 en el Consola de administración de AWS

Para crear el rol vinculado a servicio de IAM denominado **AmazonEC2SpotFleetTaggingRole** para la flota de spot de Amazon EC2

1. Abra la consola de IAM en <https://console.aws.amazon.com/iam/>.
2. En Gestión de acceso, seleccione Roles,
3. En Roles, seleccione Creación de rol.

4. En Seleccione una entidad de confianza para Tipo de entidad de confianza, elija Servicio de AWS.
5. Para otros casos de uso Servicios de AWS, elija EC2 y, a continuación, EC2 - Spot Fleet Tagging.
6. Elija Siguiente.
7. En Políticas de permisos para el Nombre de la política, compruebe AmazonEC2SpotFleetTaggingRole.
8. Elija Siguiente.
9. Para Nombrar, revisar y crear:
 - a. En Nombre de rol, escriba un nombre para identificar el rol.
 - b. En Descripción, introduzca una breve explicación de la política.
 - c. (Opcional) En Paso 1: seleccione entidades de confianza, elija Editar para modificar el código.
 - d. (Opcional) En el Paso 2: agregar permisos, seleccione Editar para modificar el código.
 - e. (Opcional) En Agregar etiquetas, elija Agregar etiqueta para agregar etiquetas al recurso.
 - f. Elija Creación de rol.

Note

En el pasado, había dos políticas administradas para el rol de la flota de spot de Amazon EC2.

- Amazon EC2 SpotFleetRole: Esta es la política gestionada original para el rol de Spot Fleet. Sin embargo, ya no recomendamos que lo utilice con AWS Batch. Esta política no admite el etiquetado de instancias de spot en entornos de computación, que es obligatorio para utilizar la función vinculada al servicio AWSServiceRoleForBatch. Si ya había creado un rol de flota de spot con esta política, aplique la nueva política recomendada a ese rol. Para obtener más información, consulte [Instancias de spot no etiquetadas en el momento de su creación](#).
- Amazon EC2 SpotFleetTaggingRole: Esta función proporciona todos los permisos necesarios para etiquetar las instancias puntuales de Amazon EC2. Utilice esta función para permitir el etiquetado de instancias puntuales en sus entornos AWS Batch informáticos.

Cree funciones de flota puntual en Amazon EC2 con AWS CLI

Para crear el rol de Amazon EC2 SpotFleetTaggingRole IAM para sus entornos de cómputo de Spot Fleet

1. Ejecute el siguiente comando con la AWS CLI.

```
$ aws iam create-role --role-name AmazonEC2SpotFleetTaggingRole \
  --assume-role-policy-document '{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "spotfleet.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}'
```

2. Para adjuntar la política de IAM EC2 SpotFleetTaggingRole gestionada por Amazon a tu EC2 SpotFleetTaggingRole rol de Amazon, ejecuta el siguiente comando con. AWS CLI

```
$ aws iam attach-role-policy \
  --policy-arn \
    arn:aws:iam::aws:policy/service-role/AmazonEC2SpotFleetTaggingRole \
  --role-name \
    AmazonEC2SpotFleetTaggingRole
```

Para crear el rol vinculado a servicio de IAM denominado **AWSServiceRoleForEC2Spot** para el spot de Amazon EC2

Note

Si el rol vinculado al servicio de IAM denominado **AWSServiceRoleForEC2Spot** ya existe, aparecerá un mensaje de error similar al siguiente.

An error occurred (InvalidInput) when calling the CreateServiceLinkedRole operation:
Service role name AWSServiceRoleForEC2Spot has been taken in this account, please try a different suffix.

- Ejecute el siguiente comando con la AWS CLI.

```
$ aws iam create-service-linked-role --aws-service-name spot.amazonaws.com
```

Para crear el rol vinculado a servicio de IAM denominado **AWSServiceRoleForEC2SpotFleet** para la flota de spot de Amazon EC2

Note

Si el rol vinculado al servicio de IAM denominado **AWSServiceRoleForEC2SpotFleet** ya existe, aparecerá un mensaje de error similar al siguiente.

An error occurred (InvalidInput) when calling the CreateServiceLinkedRole operation:
Service role name AWSServiceRoleForEC2SpotFleet has been taken in this account, please try a different suffix.

- Ejecute el siguiente comando con la AWS CLI.

```
$ aws iam create-service-linked-role --aws-service-name spotfleet.amazonaws.com
```

EventBridge Rol de IAM

Amazon EventBridge ofrece un flujo casi en tiempo real de los eventos del sistema que describen los cambios en los AWS recursos. AWS Batch los trabajos están disponibles como EventBridge objetivos. Mediante reglas sencillas que puede configurar rápidamente, puede asignar eventos y enviar trabajos de AWS Batch como respuesta a ellos. Antes de poder enviar AWS Batch trabajos

con EventBridge reglas y objetivos, EventBridge debe tener permisos para ejecutar los AWS Batch trabajos en su nombre.

 Note

Si crea una regla en la EventBridge consola que especifica una AWS Batch cola como destino, puede crear este rol. Para ver un tutorial de ejemplo, consulte [Trabajos de AWS Batch como destinos de EventBridge](#). Puede crear el EventBridge rol manualmente mediante la consola de IAM. Para obtener instrucciones, consulte [Creación de un rol mediante políticas de confianza personalizadas \(consola\)](#) en la Guía del usuario de IAM.

La relación de confianza para su función de EventBridge IAM debe proporcionar al director del `events.amazonaws.com` servicio la capacidad de asumir la función.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "events.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Asegúrese de que la política asociada a su función de EventBridge IAM permita `batch:SubmitJob` conceder permisos a sus recursos. En el siguiente ejemplo, AWS Batch proporciona la política administrada de `AWSBatchServiceEventTargetRole` para conceder estos permisos.

JSON

```
{
```

```
"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "batch:SubmitJob"
    ],
    "Resource": "*"
  }
]
```

Creación de una nube virtual privada

Los recursos de computación de sus entornos de computación necesitan acceso de red externo para comunicarse con AWS Batch y el punto de conexión del servicio de Amazon ECS. Sin embargo, es posible que tenga trabajos que desee ejecutar en subredes privadas. La creación de una VPC con subredes públicas y privadas le proporciona la flexibilidad necesaria para ejecutar trabajos en una subred pública o privada.

Puede usar Amazon Virtual Private Cloud (Amazon VPC) para lanzar AWS recursos en una red virtual que usted defina. En este tema se proporciona un enlace al asistente de Amazon VPC y una lista de las opciones que se pueden seleccionar.

Creación de una VPC

Para obtener información sobre cómo crear una VPC, consulte [Creación de una VPC únicamente](#) en la Guía del usuario de Amazon VPC y utilice la siguiente tabla para determinar qué opciones seleccionar.

Opción	Valor	
Recursos para crear	VPC solo	
Name	De manera opcional, indique un nombre para su VPC.	
IPv4 Bloque CIDR	IPv4 Entrada manual de CIDR	

Opción	Valor	
	El bloque de CIDR debe ser de un tamaño de entre /16 y /28.	
IPv6 Bloque CIDR	Sin bloqueo IPv6 CIDR	
Propiedad	Predeterminado	

Para obtener más información acerca de Amazon VPC, consulte [¿Qué es Amazon VPC?](#) en la Guía del usuario de Amazon VPC.

Siguientes pasos

Después de crear la VPC, se recomiendan los siguientes pasos:

- Cree grupos de seguridad para los recursos públicos y privados si necesitan acceso de red de entrada. Para obtener más información, consulte [Uso de grupos de seguridad](#) en la Guía del usuario de Amazon VPC.
- Cree un entorno informático AWS Batch gestionado que lance recursos informáticos a su nueva VPC. Para obtener más información, consulte [Creación de un entorno de computación](#). Si utilizas el asistente de creación de entornos de cómputo de la AWS Batch consola, puedes especificar la VPC que acabas de crear y las subredes públicas o privadas en las que quieres lanzar tus instancias.
- Crea una cola de AWS Batch trabajos que se asigne a tu nuevo entorno de cómputo. Para obtener más información, consulte [Creación de una cola de trabajos](#).
- Cree una definición de trabajo para ejecutar sus trabajos. Para obtener más información, consulte [Creación de una definición de trabajo de un solo nodo](#).
- Envíe un trabajo con su definición de trabajo a la nueva cola de trabajos. Este trabajo llegará al entorno de computación que creó con su nueva VPC y sus subredes. Para obtener más información, consulte [Tutorial: envío de un trabajo](#).

Utilice un punto final de interfaz para acceder AWS Batch

Puede usarlo AWS PrivateLink para crear una conexión privada entre su VPC y. AWS Batch Puede acceder a AWS Batch como si estuviera en su VPC, sin el uso de una puerta de enlace de Internet, un dispositivo NAT, una conexión VPN o una conexión Direct Connect . Las instancias de la VPC no necesitan direcciones IP públicas para acceder a AWS Batch.

Esta conexión privada se establece mediante la creación de un punto de conexión de interfaz alimentado por AWS PrivateLink. Creamos una interfaz de red de punto de conexión en cada subred habilitada para el punto de conexión de interfaz. Se trata de interfaces de red administradas por el solicitante que sirven como punto de entrada para el tráfico destinado a AWS Batch.

Para obtener más información, consulte [Puntos de conexión de VPC](#) en la AWS PrivateLink guía.

Consideraciones para AWS Batch

Antes de configurar un punto final de la interfaz AWS Batch, consulte las [propiedades y limitaciones del punto final de la interfaz](#) en la AWS PrivateLink Guía.

AWS Batch permite realizar llamadas a todas sus acciones de API a través del punto final de la interfaz.

Antes de configurar los puntos finales de la VPC de la interfaz AWS Batch, tenga en cuenta las siguientes consideraciones:

- Los trabajos que utilizan el tipo de lanzamiento de recursos de Fargate no requieren los puntos de enlace de la VPC de la interfaz para Amazon ECS, pero es posible que necesite los puntos de enlace de la VPC de la interfaz para Amazon AWS Batch ECR, Secrets Manager o Amazon Logs, que se describen en los puntos siguientes. CloudWatch
- Para ejecutar trabajos, debe crear los puntos de conexión de VPC de interfaz para Amazon ECS. Para obtener más información, consulte [Puntos de conexión de VPC de tipo interfaz \(AWS PrivateLink PrivateLink\)](#) en la Guía para desarrolladores de Amazon Elastic Container Service.
- Para permitir que sus tareas extraigan imágenes privadas de Amazon ECR, debe crear los puntos de conexión de VPC de interfaz de Amazon ECR. Para obtener más información, consulte [Puntos de enlace de la VPC de interfaz \(AWS PrivateLink\)](#) en la Guía del usuario de Amazon Elastic Container Registry.
- Para permitir que sus tareas extraigan información confidencial de Secrets Manager, debe crear los puntos de conexión de VPC de interfaz de Secrets Manager. Para obtener más información,

consulte [Utilización de Secrets Manager con puntos de enlace de la VPC](#) en la Guía del usuario de AWS Secrets Manager .

- Si su VPC no tiene una puerta de enlace a Internet y sus trabajos utilizan el controlador de `awslogs` registro para enviar la información de registro a CloudWatch Logs, debe crear un punto de enlace de VPC de interfaz para Logs. CloudWatch Para obtener más información, consulte [Uso de CloudWatch registros con puntos de enlace de VPC de interfaz](#) en la Guía del usuario de Amazon CloudWatch Logs.
- Las tareas que utilizan el tipo de lanzamiento de EC2 requieren que las instancias de contenedor en las que se lanzan ejecuten la versión 1.25.1 o posterior del agente de contenedor de Amazon ECS. Para obtener más información, consulte [Versiones del agente de contenedores de Linux en Amazon ECS](#) en la Guía para desarrolladores de Amazon Elastic Container Service.
- Los puntos de enlace de la VPC no admiten las solicitudes entre regiones. Asegúrese de crear su punto de conexión en la misma región en la que tiene previsto enviar llamadas a la API de AWS Batch.
- Los puntos de conexión de VPC solo admiten DNS proporcionadas por Amazon a través de Amazon Route 53. Si desea utilizar su propio DNS, puede utilizar el enrutamiento de DNS condicional. Para obtener más información, consulte [Conjuntos de opciones de DHCP](#) en la Guía del usuario de Amazon VPC.
- El grupo de seguridad asociado al punto de conexión de la VPC debe permitir las conexiones entrantes en el puerto 443 desde la subred privada de la VPC.
- AWS Batch no admite los puntos finales de la interfaz de VPC en los siguientes casos: Regiones de AWS
 - Asia-Pacífico (Osaka) (`ap-northeast-3`)
 - Asia-Pacífico (Yakarta) (`ap-southeast-3`)

Cree un punto final de interfaz para AWS Batch

Puede crear un punto final de interfaz para AWS Batch usar la consola de Amazon VPC o AWS Command Line Interface (AWS CLI). Para obtener más información, consulte [Creación de un punto de conexión de interfaz](#) en la Guía de AWS PrivateLink .

Cree un punto final de interfaz para AWS Batch utilizar los siguientes nombres de servicio:

- `com.amazonaws. region.batch`

- `com.amazonaws.region.batch-fips` (para ver puntos de enlace compatibles con FIPS, consulte puntos de enlace y cuotas) [AWS Batch](#)

Por ejemplo:

```
com.amazonaws.us-east-2.batch
```

```
com.amazonaws.us-east-2.batch-fips
```

En la partición `aws-cn`, el formato es diferente:

```
cn.com.amazonaws.region.batch
```

Por ejemplo:

```
cn.com.amazonaws.cn-northwest-1.batch
```

Nombres de DNS privados para los AWS Batch puntos finales de la interfaz

Si habilita el DNS privado para el punto final de la interfaz, puede usar nombres de DNS específicos para AWS Batch conectarse. Ofrecemos estas opciones:

- `lote.region.amazonaws.com`
- `lote.region.api.aws`

Para puntos finales compatibles con FIPS:

- `embarque por lotes.region.api.aws`
- `fips.batch.region.amazonaws.com` no es compatible

Para obtener más información, consulte [Acceso a un servicio a través de un punto de conexión de interfaz](#) en la Guía AWS PrivateLink .

Creación de una política de punto de conexión para el punto de conexión de interfaz

Una política de punto de conexión es un recurso de IAM que puede adjuntar al punto de conexión de su interfaz. La política de puntos finales predeterminada permite el acceso total a AWS Batch través del punto final de la interfaz. Para controlar el acceso permitido a AWS Batch desde la VPC, adjunte una política de puntos de conexión personalizada al punto de conexión de interfaz.

Una política de punto de conexión especifica la siguiente información:

- Las entidades principales (Cuentas de AWS, usuarios y roles de IAM) que puede realizar acciones
- Las acciones que se pueden realizar.
- El recurso en el que se pueden realizar las acciones.

Para obtener más información, consulte [Control del acceso a los servicios con políticas de punto de conexión](#) en la Guía del usuario de AWS PrivateLink .

Ejemplo: política de puntos finales de VPC para acciones AWS Batch

El siguiente es un ejemplo de una política de un punto de conexión personalizado. Al adjuntar esta política al punto final de la interfaz, se concede acceso a las AWS Batch acciones enumeradas a todos los principales de todos los recursos.

```
{
  "Statement": [
    {
      "Principal": "*",
      "Effect": "Allow",
      "Action": [
        "batch:SubmitJob",
        "batch:ListJobs",
        "batch:DescribeJobs"
      ],
      "Resource": "*"
    }
  ]
}
```

Validación de conformidad para AWS Batch

Para saber si uno Servicio de AWS está dentro del ámbito de aplicación de programas de cumplimiento específicos, consulte [Servicios de AWS Alcance por programa de cumplimiento](#) [Servicios de AWS](#) de cumplimiento y elija el programa de cumplimiento que le interese. 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 [Descarga de informes en AWS Artifact](#) .

Su responsabilidad de cumplimiento al Servicios de AWS utilizarlos viene determinada por la confidencialidad de sus datos, los objetivos de cumplimiento de su empresa y las leyes y reglamentos aplicables. Para obtener más información sobre su responsabilidad de conformidad al utilizarlos Servicios de AWS, consulte [AWS la documentación de seguridad](#).

Seguridad de la infraestructura en AWS Batch

Como servicio gestionado, AWS Batch 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 Batch 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.

Puede llamar a estas operaciones de API desde cualquier ubicación de la red, pero AWS Batch admite políticas de acceso basadas en los recursos, que pueden incluir restricciones basadas en la dirección IP de origen. También puede utilizar AWS Batch políticas para controlar el acceso desde puntos de enlace específicos o específicos de Amazon Virtual Private Cloud (Amazon VPC). VPCs En efecto, esto aísla el acceso a la red a un AWS Batch recurso determinado únicamente de la VPC específica de la red. AWS

Prevención de la sustitución confusa entre servicios

El problema del suplente confuso es una cuestión de seguridad en la 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 lo 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:SourceArn](#) las claves de contexto en las políticas de recursos para limitar los permisos que se AWS Batch otorgan a otro servicio al recurso. Si el valor de `aws:SourceArn` no contiene el ID de cuenta, como un 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 que se AWS Batch almacena.

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 está especificando varios recursos, utilice la clave de condición de contexto global `aws:SourceArn` con caracteres comodines (*) para las partes desconocidas del ARN. Por ejemplo, `arn:aws:service:*:123456789012:*`.

En los siguientes ejemplos se muestra cómo utilizar las claves de contexto de condición `aws:SourceAccount` global `aws:SourceArn` y las claves contextuales AWS Batch para evitar el confuso problema de los diputados.

Ejemplo: rol para acceder a un solo entorno de computación

El siguiente rol solo se puede usar para acceder a un entorno de computación. El nombre del trabajo debe especificarse como `*` porque la cola de trabajos se puede asociar a varios entornos de computación.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "batch.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        },
        "ArnLike": {
          "aws:SourceArn": [
            "arn:aws:batch:us-east-1:123456789012:compute-environment/testCE",
            "arn:aws:batch:us-east-1:123456789012:job/*"
          ]
        }
      }
    }
  ]
}
```

Ejemplo: rol para acceder a varios entornos de computación

El siguiente rol se puede utilizar para tener acceso a varios entornos de computación. El nombre del trabajo debe especificarse como `*` porque la cola de trabajos se puede asociar a varios entornos de computación.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "batch.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        },
        "ArnLike": {
          "aws:SourceArn": [
            "arn:aws:batch:us-east-1:123456789012:compute-environment/*",
            "arn:aws:batch:us-east-1:123456789012:job/*"
          ]
        }
      }
    }
  ]
}
```

Registrar las llamadas a la AWS Batch API con AWS CloudTrail

AWS Batch 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 Batch. CloudTrail captura todas las llamadas a la API AWS Batch como eventos. Las llamadas capturadas incluyen llamadas desde la AWS Batch consola y llamadas en código a las operaciones de la AWS Batch API. Si crea una ruta, puede habilitar la entrega continua de CloudTrail eventos a un bucket de Amazon S3, incluidos los eventos para AWS Batch. 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 Batch, la dirección IP desde la que se realizó la solicitud, quién la realizó, cuándo se realizó y detalles adicionales.

Para obtener más información CloudTrail, consulte la [Guía AWS CloudTrail del usuario](#).

Temas

- [AWS Batch información en CloudTrail](#)
- [Referencia: Descripción de las entradas de los archivos de AWS Batch registro](#)

AWS Batch información en CloudTrail

CloudTrail está habilitada en su AWS cuenta al crear la cuenta. Cuando se produce una actividad en AWS Batch, 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, consulte [Visualización de 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 Batch, crea una ruta. Un rastro permite CloudTrail entregar archivos de registro a un bucket de Amazon S3. De forma predeterminada, cuando crea una ruta en la consola, la ruta se aplica a todas AWS las regiones. 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 Integraciones y servicios 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](#)

Todas AWS Batch las acciones se registran CloudTrail y se documentan en la [https://docs.aws.amazon.com/batch/ última versión APIReference //](https://docs.aws.amazon.com/batch/última versión APIReference //). Por ejemplo, las llamadas a las secciones [SubmitJob](#), [ListJobs](#) y [DescribeJobs](#) generan entradas en los archivos de registro de CloudTrail .

Cada entrada de registro o evento contiene información sobre quién generó la solicitud. La información de identidad del usuario lo ayuda a determinar lo siguiente:

- Si la solicitud se realizó con las credenciales raíz o del usuario de 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 servicio. AWS

Para obtener más información, consulte el [Elemento `userIdentity` de CloudTrail](#).

Referencia: Descripción de las entradas de los archivos de AWS Batch 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 única solicitud de cualquier origen e incluye información sobre la acción solicitada, la fecha y la hora de la acción, los parámetros de la solicitud, etcétera. Los archivos de registro de CloudTrail no son un rastro de la pila ordenada de las llamadas a la API públicas, por lo que no aparecen en ningún orden específico.

El siguiente ejemplo muestra una entrada de CloudTrail registro que demuestra la [CreateComputeEnvironment](#) acción.

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AIDACKCEVSQ6C2EXAMPLE:admin",
    "arn": "arn:aws:sts::012345678910:assumed-role/Admin/admin",
    "accountId": "012345678910",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2017-12-20T00:48:46Z"
      },
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE",
        "arn": "arn:aws:iam::012345678910:role/Admin",
        "accountId": "012345678910",
        "userName": "Admin"
      }
    }
  }
}
```



```

},
"eventTime": "2017-12-20T00:48:46Z",
"eventSource": "batch.amazonaws.com",
"eventName": "CreateComputeEnvironment",
"awsRegion": "us-east-1",
"sourceIPAddress": "203.0.113.1",
"userAgent": "aws-cli/1.11.167 Python/2.7.10 Darwin/16.7.0 botocore/1.7.25",
"requestParameters": {
  "computeResources": {
    "subnets": [
      "subnet-5eda8e04"
    ],
    "tags": {
      "testBatchTags": "CLI testing CE"
    },
    "desiredvCpus": 0,
    "minvCpus": 0,
    "instanceTypes": [
      "optimal"
    ],
    "securityGroupIds": [
      "sg-aba9e8db"
    ],
    "instanceRole": "ecsInstanceRole",
    "maxvCpus": 128,
    "type": "EC2"
  },
  "state": "ENABLED",
  "type": "MANAGED",
  "computeEnvironmentName": "Test"
},
"responseElements": {
  "computeEnvironmentName": "Test",
  "computeEnvironmentArn": "arn:aws:batch:us-east-1:012345678910:compute-environment/
Test"
},
"requestID": "890b8639-e51f-11e7-b038-EXAMPLE",
"eventID": "874f89fa-70fc-4798-bc00-EXAMPLE",
"readOnly": false,
"eventType": "AwsApiCall",
"recipientAccountId": "012345678910"
}

```

Solucionar problemas de IAM AWS Batch

Utilice la siguiente información como ayuda para diagnosticar y solucionar problemas comunes que pueden surgir al trabajar con AWS Batch un IAM.

Temas

- [No estoy autorizado a realizar ninguna acción en AWS Batch](#)
- [No estoy autorizado a realizar iam: PassRole](#)
- [Quiero permitir que personas ajenas a mi AWS cuenta accedan a mis AWS Batch recursos](#)

No estoy autorizado a realizar ninguna acción en AWS Batch

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.

En el siguiente ejemplo, el error se produce cuando el usuario mateojackson intenta utilizar la consola para consultar los detalles acerca de un recurso ficticio *my-example-widget*, pero no tiene los permisos ficticios batch:*GetWidget*.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:  
batch:GetWidget on resource: my-example-widget
```

En este caso, Mateo pide a su administrador que actualice sus políticas de forma que pueda obtener acceso al recurso *my-example-widget* mediante la acción batch:*GetWidget*. Para obtener más información sobre la concesión de permisos para transferir un rol, consulte [Conceder permisos a un usuario para transferir un rol a un AWS servicio](#).

No estoy autorizado a 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 Batch.

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 Batch. 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 Batch 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 Batch es compatible con estas funciones, consulte. [¿Cómo AWS Batch 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.

AWS Step Functions

Puede utilizar la consola de AWS Batch para ver detalles de las máquinas de estado de Step Functions y las funciones que utilizan.

Secciones

- [Tutorial: visualización de detalles de las máquinas de estado](#)
- [Tutorial: edición de una máquina de estado](#)
- [Tutorial: ejecución de una máquina de estado](#)

Tutorial: visualización de detalles de las máquinas de estado

La consola de AWS Batch muestra una lista de las máquinas de estado en la Región de AWS actual que contiene al menos un paso de flujo de trabajo que envía un trabajo AWS Batch.

Elija una máquina de estado para ver una representación gráfica del flujo de trabajo. Los pasos resaltados en azul representan trabajos AWS Batch. Utilice los controles del gráfico para acercar, alejar y centrar el gráfico.

Note

Cuando se hace [referencia dinámicamente con JsonPath](#) a un trabajo AWS Batch en la definición de máquina de estado, los detalles del trabajo no se pueden mostrar en la consola de AWS Batch. En su lugar, el nombre del trabajo se muestra como una referencia dinámica y los pasos correspondientes del gráfico aparecen atenuados.

Para ver los detalles de una máquina de estado

1. Abra la consola de AWS Batch de la [Página de orquestación del flujo de trabajo con Step Functions](#).
2. Creación de una máquina de estado.

<result>

La consola de AWS Batch abre la página Detalles.

</result>

Para obtener más información, consulte [Step Functions](#) en la Guía para desarrolladores de AWS Step Functions.

Tutorial: edición de una máquina de estado

Cuando desee editar una máquina de estado, AWS Batch abre la página Editar definición de la consola de Step Functions.

Para editar una máquina de estado

1. Abra la consola de AWS Batch de la [Página de orquestación del flujo de trabajo con Step Functions](#).
2. Creación de una máquina de estado.
3. Elija Editar.

La consola de Step Functions abre la página Editar definición.

4. Edite la máquina de estado y elija Guardar.

Para obtener más información acerca de la edición de máquinas de estado, consulte [Lenguaje de máquinas de estado de Step Functions](#) en la Guía para desarrolladores de AWS Step Functions.

Tutorial: ejecución de una máquina de estado

Cuando desee ejecutar una máquina de estado, AWS Batch abre la página Nueva ejecución de la consola de Step Functions.

Para ejecutar una máquina de estado

1. Abra la consola de AWS Batch de la [Página de orquestación del flujo de trabajo con Step Functions](#).
2. Creación de una máquina de estado.
3. Elija Ejecutar.

La consola de Step Functions abre la página Nueva ejecución.

4. (Opcional) Edite la máquina de estado y elija Iniciar ejecución.

Para obtener más información acerca de cómo ejecutar máquinas de estado, consulte [Conceptos de ejecución de máquinas de estado de Step Functions](#) en la Guía para desarrolladores de AWS Step Functions.

Transmisión de eventos de AWS Batch para Amazon EventBridge

Puede utilizar el flujo de eventos de AWS Batch de Amazon EventBridge para recibir notificaciones casi en tiempo real sobre el estado actual de los trabajos en sus colas de trabajo.

Puede usar EventBridge para obtener más información sobre su servicio de AWS Batch. Más específicamente, puede usarlo para comprobar el progreso de los trabajos, crear flujos de trabajo personalizados de AWS Batch, generar informes o métricas de uso o crear sus propios paneles. Con AWS Batch y EventBridge, puede eliminar el código de programación y monitorización que sondea continuamente los trabajos de AWS Batch para detectar cambios en su estado. En su lugar, puede gestionar los cambios de estado de los trabajos AWS Batch de forma asíncrona mediante una variedad de destinos de Amazon EventBridge. Entre ellos se incluyen AWS Lambda, Amazon Simple Queue Service, Amazon Simple Notification Service o Amazon Kinesis Data Streams.

Se garantiza que los eventos de la secuencia de eventos de AWS Batch se entregan al menos una vez. En caso de que se envíen eventos duplicados, el evento ofrece información suficiente para identificar duplicados. De esta forma, puede comparar la marca horaria del evento y el estado del trabajo.

Los trabajos AWS Batch están disponibles como destinos de EventBridge. Mediante reglas sencillas, puede asignar eventos y enviar trabajos de AWS Batch como respuesta a ellos. Para obtener más información, consulte [¿Qué es EventBridge?](#) en la Guía del usuario de Amazon EventBridge. También puede utilizar EventBridge para programar acciones automatizadas que se activen automáticamente en determinados momentos a través de cron o de expresiones de frecuencia. Para obtener más información, consulte [Creación de reglas de Amazon EventBridge que se ejecutan en un horario](#) en la Guía del usuario de Amazon EventBridge. Para ver un tutorial de ejemplo, consulte [Trabajos de AWS Batch como destinos de EventBridge](#). Para obtener información sobre el uso del programador de EventBridge, consulte [Configuración del Programador de Amazon EventBridge](#) en la Guía del usuario de Amazon EventBridge.

Temas

- [AWS Batch eventos](#)
- [Tutorial: Usa las notificaciones AWS de usuario con AWS Batch](#)
- [Trabajos de AWS Batch como destinos de EventBridge](#)
- [Tutorial: escucha de los eventos de trabajo de AWS Batch con EventBridge](#)

- [Tutorial: envío de alertas de Amazon Simple Notification Service para eventos de trabajos fallidos](#)

AWS Batch eventos

AWS Batch envía los eventos de cambio de estado del trabajo a EventBridge. AWS Batch rastrea el estado de sus trabajos. Si el estado de un trabajo enviado anteriormente cambia, se invoca un evento. Por ejemplo, si un trabajo en el estado `RUNNING` pasa a al estado `FAILED`. Estos eventos se clasifican como eventos de cambio en el estado de los trabajos.

Note

AWS Batch podría añadir otros tipos de eventos, fuentes y detalles en el futuro. Si va a deserializar datos JSON de eventos mediante programación, asegúrese de que la aplicación esté preparada para tratar propiedades desconocidas. Esto es para evitar problemas si se agregan estas propiedades adicionales y cuando se agregan.

Eventos de cambio de estado de los trabajos

Cada vez que un trabajo existente (enviado previamente) cambia de estado, se crea un evento. Para obtener más información acerca de los estados de los trabajos de AWS Batch, consulte [Estados de trabajo](#).

Note

No se crean eventos para el envío inicial de los trabajos.

Example Evento de cambio de estado de los trabajos

Los eventos de cambio de estado de trabajo se entregan en el siguiente formato. La sección `detail` se asemeja al objeto [JobDetail](#) que se devuelve de una operación de la API [DescribeJobs](#) en la Referencia de la API AWS Batch. Para obtener más información acerca de los parámetros de EventBridge, consulte [Eventos y patrones de eventos](#) en la Guía del usuario de Amazon EventBridge.

```
{  
  "version": "0",
```



```

    "id": "c8f9c4b5-76e5-d76a-f980-7011e206042b",
    "detail-type": "Batch Job State Change",
    "source": "aws.batch",
    "account": "123456789012",
    "time": "2022-01-11T23:36:40Z",
    "region": "us-east-1",
    "resources": [
      "arn:aws:batch:us-east-1:123456789012:job/4c7599ae-0a82-49aa-ba5a-4727fcce14a8"
    ],
    "detail": {
      "jobArn": "arn:aws:batch:us-east-1:123456789012:job/4c7599ae-0a82-49aa-ba5a-4727fcce14a8",
      "jobName": "event-test",
      "jobId": "4c7599ae-0a82-49aa-ba5a-4727fcce14a8",
      "jobQueue": "arn:aws:batch:us-east-1:123456789012:job-queue/PexjEHappyPathCanary2JobQueue",
      "status": "RUNNABLE",
      "attempts": [],
      "createdAt": 1641944200058,
      "retryStrategy": {
        "attempts": 2,
        "evaluateOnExit": []
      },
      "dependsOn": [],
      "jobDefinition": "arn:aws:batch:us-east-1:123456789012:job-definition/first-run-job-definition:1",
      "parameters": {},
      "container": {
        "image": "137112412989.dkr.ecr.us-east-1.amazonaws.com/amazonlinux:latest",
        "command": [
          "sleep",
          "600"
        ],
        "volumes": [],
        "environment": [],
        "mountPoints": [],
        "ulimits": [],
        "networkInterfaces": [],
        "resourceRequirements": [
          {
            "value": "2",
            "type": "VCPU"
          }, {
            "value": "256",

```

```

        "type": "MEMORY"
      }
    ],
    "secrets": [],
  },
  "propagateTags": false,
  "platformCapabilities": []
}

```

Eventos bloqueados en la cola de trabajos

Cada vez que se AWS Batch detecta un trabajo en el `RUNNABLE` estado y, por lo tanto, se bloquea una cola, se crea un evento en Amazon CloudWatch Events. Para obtener más información sobre las causas de las colas de bloqueo admitidas, consulte [Trabajos bloqueados en estado `RUNNABLE`](#). El mismo motivo también está disponible en el campo `statusReason` de la acción de API [DescribeJobs](#).

Example Evento bloqueado en la cola de trabajos

Los eventos bloqueados en la cola de trabajos se entregan en el siguiente formato. La `detail` sección se parece al [JobDetail](#) objeto devuelto por una operación de [DescribeJobs](#) API en la Referencia de AWS Batch API. Para obtener más información sobre EventBridge los parámetros, consulte [Eventos y patrones de eventos](#) en la Guía del EventBridge usuario de Amazon.

```

{
  "version": "0",
  "id": "c8f9c4b5-76e5-d76a-f980-7011e206042b",
  "detail-type": "Batch Job Queue Blocked",
  "source": "aws.batch",
  "account": "123456789012",
  "time": "2022-01-11T23:36:40Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:batch:us-east-1:123456789012:job/4c7599ae-0a82-49aa-ba5a-4727fcce14a8",
    "arn:aws:batch:us-east-1:123456789012:job-queue/PexjEHappyPathCanary2JobQueue"
  ],
  "detail": {
    "jobArn": "arn:aws:batch:us-east-1:123456789012:job/4c7599ae-0a82-49aa-ba5a-4727fcce14a8",
    "jobName": "event-test",

```

```

    "jobId": "4c7599ae-0a82-49aa-ba5a-4727fcce14a8",
    "jobQueue": "arn:aws:batch:us-east-1:123456789012:job-queue/
PexjEHappyPathCanary2JobQueue",
    "status": "RUNNABLE",
    "statusReason": "blocked-reason",
    "attempts": [],
    "createdAt": 1641944200058,
    "retryStrategy": {
        "attempts": 2,
        "evaluateOnExit": []
    },
    "dependsOn": [],
    "jobDefinition": "arn:aws:batch:us-east-1:123456789012:job-definition/first-
run-job-definition:1",
    "parameters": {},
    "container": {
        "image": "137112412989.dkr.ecr.us-east-1.amazonaws.com/amazonlinux:latest",
        "command": [
            "sleep",
            "600"
        ],
        "volumes": [],
        "environment": [],
        "mountPoints": [],
        "ulimits": [],
        "networkInterfaces": [],
        "resourceRequirements": [
            {
                "value": "2",
                "type": "VCPU"
            }, {
                "value": "256",
                "type": "MEMORY"
            }
        ],
        "secrets": []
    },
    "propagateTags": false,
    "platformCapabilities": []
}

```

Eventos de cambio de estado de los trabajos

Cada vez que un trabajo de servicio existente cambia de estado, se crea un evento. Para obtener más información sobre los estados de los trabajos de servicio, consulte [Asignar el estado del trabajo del AWS Batch servicio al estado de la SageMaker IA](#).

Note

No se crean eventos para el envío inicial de los trabajos.

Example Evento de cambio de estado de los trabajos de servicio

Los eventos de cambio de estado de trabajo de servicio se entregan en el siguiente formato. La sección detail se asemeja a la respuesta que se devuelve de una operación de la API [DescribeServiceJob](#) en la Referencia de la API de AWS Batch. Para obtener más información acerca de los parámetros de EventBridge, consulte [Eventos y patrones de eventos](#) en la Guía del usuario de Amazon EventBridge.

Note

Los campos tags y serviceRequestPayload no se incluyen en el evento detail.

```
{
  "version": "0",
  "id": "c8f9c4b5-76e5-d76a-f980-7011e206042b",
  "detail-type": "Batch Service Job State Change",
  "source": "aws.batch",
  "account": "123456789012",
  "time": "2022-01-11T23:36:40Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:batch:us-east-1:123456789012:service-job/4c7599ae-0a82-49aa-ba5a-4727fcce14a8"
  ],
  "detail": {
    "attempts": [
      {
        "serviceResourceId": {
```

```

        "name": "TrainingJobArn",
        "value": "arn:aws:sagemaker:us-east-1:123456789012:training-job/AWSBatchmy-
training-job88b610a69aa8380ca5b0a7aba3f81cb8"
    },
    "startedAt": 1641944300058,
    "stoppedAt": 1641944400058,
    "statusReason": "Received status from SageMaker: Training job completed"
}
],
"createdAt": 1641944200058,
"jobArn": "arn:aws:batch:us-east-1:123456789012:service-job/4c7599ae-0a82-49aa-
ba5a-4727fcce14a8",
"jobId": "0bb17543-ece6-4480-b1a7-a556d344746b",
"jobName": "event-test",
"jobQueue": "arn:aws:batch:us-east-1:123456789012:job-queue/HappyPathJobQueue",
"latestAttempt": {
    "serviceResourceId": {
        "name": "TrainingJobArn",
        "value": "arn:aws:sagemaker:us-east-1:123456789012:training-job/AWSBatchmy-
training-job88b610a69aa8380ca5b0a7aba3f81cb8"
    }
},
"serviceJobType": "SAGEMAKER_TRAINING",
"startedAt": 1641944300058,
"status": "SUCCEEDED",
"statusReason": "Received status from SageMaker: Training job completed",
"stoppedAt": 1641944400058,
"timeoutConfig": {
    "attemptDurationSeconds": 60
}
}
}

```

Eventos bloqueados en el trabajo de servicio

Cada vez que se AWS Batch detecta una cola bloqueada, se crea un evento en Amazon CloudWatch Events. El motivo de la cola bloqueada está disponible en el campo `statusReason` de la acción de API [DescribeServiceJob](#).

Example Evento bloqueado del servicio de la cola de trabajos

El servicio de eventos bloqueados en la cola de trabajos se entrega en el siguiente formato. La `detail` sección se parece a la respuesta que devuelve la operación de la [DescribeServiceJob](#) API

en la Referencia de la AWS Batch API. Para obtener más información sobre EventBridge los parámetros, consulte [Eventos y patrones de eventos](#) en la Guía del EventBridge usuario de Amazon.

Note

Los campos `tags` y `serviceRequestPayload` no se incluyen en el evento `detail`.

```
{
  "version": "0",
  "id": "c8f9c4b5-76e5-d76a-f980-7011e206042b",
  "detail-type": "Batch Service Job Queue Blocked",
  "source": "aws.batch",
  "account": "123456789012",
  "time": "2022-01-11T23:36:40Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:batch:us-east-1:123456789012:service-job/4c7599ae-0a82-49aa-ba5a-4727fcce14a8"
  ],
  "detail": {
    "attempts": [],
    "createdAt": 1641944200058,
    "jobArn": "arn:aws:batch:us-east-1:123456789012:service-job/4c7599ae-0a82-49aa-ba5a-4727fcce14a8",
    "jobId": "6271dfdf-d8a7-41b1-a4d2-55a2224f5375",
    "jobName": "event-test",
    "jobQueue": "arn:aws:batch:us-east-1:123456789012:job-queue/HappyPathJobQueue",
    "serviceJobType": "SAGEMAKER_TRAINING",
    "status": "RUNNABLE",
    "statusReason": "blocked-reason",
    "timeoutConfig": {
      "attemptDurationSeconds": 60
    }
  }
}
```

Eventos de prevención de tareas de servicio

Los eventos de preferencia se publican cada vez que se inicia o finaliza una secuencia de preferencia. Un evento iniciado se crea cuando se activa por primera vez la preferencia para

recuperar la capacidad prestada. Un evento completado se crea cuando finaliza la secuencia de preferencia e incluye un resumen de los intentos evitados.

Example Evento iniciado por Service Job Preemption

Los eventos iniciados por la prevención de tareas de servicio se entregan en el siguiente formato. Para obtener más información sobre EventBridge los parámetros, consulte [Eventos y patrones de eventos](#) en la Guía del EventBridge usuario de Amazon.

```
{
  "version": "0",
  "id": "1b4a511e-2737-226a-a1e7-fc97f1cd9681",
  "detail-type": "Batch Service Job Preemption Started",
  "source": "aws.batch",
  "account": "123456789012",
  "time": "2026-03-16T19:57:27Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:batch:us-east-1:123456789012:service-job/51f245a9-2995-4a53-bced-7b3c00028f84"
  ],
  "detail": {
    "statusReason": "PREEMPTION_IN_PROGRESS: Cross-share preemption triggered to reclaim borrowed capacity",
    "preemptedJobArn": "arn:aws:batch:us-east-1:123456789012:service-job/51f245a9-2995-4a53-bced-7b3c00028f84",
    "status": "RUNNING"
  }
}
```

Example Evento Service Job Preemption completado

Los eventos de prioridad de trabajos de servicio finalizados se entregan en el siguiente formato. El `preemptionSummary` campo contiene detalles sobre los intentos prevenidos, incluido el recuento y la información sobre los intentos prevenidos más recientes.

```
{
  "version": "0",
  "id": "2b1c6151-c166-edf5-822c-211b3bfd46b2",
  "detail-type": "Batch Service Job Preemption Completed",
  "source": "aws.batch",
  "account": "123456789012",
  "time": "2026-03-16T19:57:47Z",
```

```

"region": "us-east-1",
"resources": [
  "arn:aws:batch:us-east-1:123456789012:service-job/51f245a9-2995-4a53-
bcded-7b3c00028f84"
],
"detail": {
  "preemptedJobArn": "arn:aws:batch:us-east-1:123456789012:service-
job/51f245a9-2995-4a53-bced-7b3c00028f84",
  "preemptionSummary": {
    "preemptedAttemptCount": 1,
    "recentPreemptedAttempts": [
      {
        "serviceResourceId": {
          "name": "TrainingJobArn",
          "value": "arn:aws:sagemaker:us-east-1:123456789012:training-job/AWSBatchqm-
training-job9b2f08f911cf3dd794c1b3e72ae7ca5f"
        },
        "startedAt": 1773690923359,
        "stoppedAt": 1773691064669,
        "statusReason": "Cross-share preemption triggered to reclaim borrowed
capacity"
      }
    ]
  },
  "status": "RUNNABLE"
}
}

```

Tutorial: Usa las notificaciones AWS de usuario con AWS Batch

Puede usar [notificaciones de usuario de AWS](#) para configurar los canales de entrega a fin de recibir notificaciones sobre los eventos de AWS Batch . Recibirá una notificación cuando un evento coincida con una regla que especifique. Puede recibir notificaciones de eventos a través de varios canales, como correo electrónico, notificaciones por chat de [Amazon Q Developer en aplicaciones de chat](#) o notificaciones push de [AWS Console Mobile Application](#). También puede ver las notificaciones en el [Centro de notificaciones de la consola](#). Las notificaciones de usuario admiten la agregación, lo que puede reducir el número de notificaciones que recibe durante eventos específicos.

Para configurar las notificaciones de usuario en AWS Batch:

1. Abra la [consola de AWS Batch](#).

2. Elija Dashesboard (Panel).
3. Seleccione Configurar notificaciones.
4. En Notificaciones de AWS usuario, elija Crear configuración de notificaciones.

Para obtener más información sobre cómo configurar y ver las notificaciones de usuario, consulte [Introducción a las notificaciones AWS de usuario](#).

Trabajos de AWS Batch como destinos de EventBridge

Amazon EventBridge entrega una secuencia de eventos de sistema casi en tiempo real que describe los cambios producidos en los recursos de Amazon Web Services. Por lo general, AWS Batch en Amazon Elastic Container Service, Amazon Elastic Kubernetes Service y los trabajos de AWS Fargate están disponibles como destinos de EventBridge. Mediante reglas sencillas, puede asignar eventos y enviar trabajos de AWS Batch como respuesta a ellos. Para obtener más información, consulte [¿Qué es EventBridge?](#) en la Guía del usuario de Amazon EventBridge.

También puede utilizar EventBridge para programar acciones automatizadas que se activen automáticamente en determinados momentos a través cron o expresiones de frecuencia. Para obtener más información, consulte [Creación de reglas de Amazon EventBridge que se ejecutan en un horario](#) en la Guía del usuario de Amazon EventBridge.

Para obtener información acerca de cómo crear reglas de eventos en la consola de EventBridge, consulte [Creación de reglas de Amazon EventBridge que reaccionan a los eventos](#) en la Guía del usuario de Amazon EventBridge.

Los casos de uso más comunes de los trabajos de AWS Batch como objetivo de EventBridge incluyen los siguientes casos de uso:

- Un trabajo programado se produce a intervalos de tiempo regulares. Por ejemplo, un trabajo de cron solo se realiza durante las horas de bajo uso, cuando las instancias puntuales de spot de Amazon EC2 son menos costosas.
- Se ejecuta un trabajo de AWS Batch en respuesta a una operación de API registrada en CloudTrail. Por ejemplo, se envía un trabajo cada vez que se carga un objeto en un bucket de Amazon S3 específico. Cada vez que esto ocurre, el transformador de entrada de EventBridge pasa el bucket y el nombre clave del objeto a los parámetros de AWS Batch.

 Note

En este escenario, todos los recursos AWS relacionados deben estar en la misma región. Esto incluye recursos como el bucket de Amazon S3, la regla de EventBridge y los registros de CloudTrail.

Antes de poder enviar los trabajos de AWS Batch con las reglas y los destinos de EventBridge, el servicio EventBridge requiere varios permisos para ejecutar los trabajos de AWS Batch. Cuando cree una regla en la consola de EventBridge que especifique un trabajo de AWS Batch como destino, se le ofrecerá la oportunidad de crear este rol. Para obtener más información sobre la entidad principal de servicio y los permisos de IAM necesarios para este rol, consulte [EventBridge Rol de IAM](#).

Temas

- [Tutorial: creación de un trabajo de AWS Batch programado](#)
- [Tutorial: creación de una regla con un patrón de evento](#)
- [Tutorial: Pase la información del evento a un AWS Batch objetivo según un cronograma utilizando el transformador EventBridge de entrada](#)

Tutorial: creación de un trabajo de AWS Batch programado

En el siguiente procedimiento se explica cómo crear un trabajo de AWS Batch programado y el rol de IAM de EventBridge necesario.

Creación de un trabajo de AWS Batch programado con EventBridge

 Note

Este procedimiento funciona para todos los trabajos de AWS Batch en Amazon ECS, Amazon EKS y AWS Fargate.

1. Abra la consola de Amazon EventBridge en <https://console.aws.amazon.com/events/>.
2. En la barra de navegación, elija la Región de AWS a utilizar.
3. En el panel de navegación, elija Reglas.
4. Elija Creación de regla.

5. En Nombre, especifique un nombre único para el entorno de computación. El nombre puede contener hasta 64 caracteres. Puede contener letras mayúsculas y minúsculas, números, guiones (-) y guiones bajos (_).

 Note

Una regla no puede tener el mismo nombre que otra regla de la misma región y del mismo bus de eventos.

6. (Opcional) En Descripción, introduzca una descripción para la regla.
7. En Bus de eventos, elija el bus de eventos que desea asociar a esta regla. Si desea que esta regla coincida con eventos procedentes de su cuenta, elija predeterminado. Cuando un Servicio de AWS en su cuenta emite un evento, siempre va al bus de eventos predeterminado de su cuenta.
8. (Opcional) Desactive la regla en el bus seleccionado si no desea ejecutarla inmediatamente.
9. En Tipo de regla, elija Programación.
10. Seleccione Continuar para crear la regla o Siguiente.
11. En Programar patrón, realice una de las siguientes acciones:
 - Elija Un horario detallado que se ejecute a una hora específica, como las 8:00 a. m. PST el primer lunes de cada mes y después introduzca la expresión cron. Para obtener más información, consulte [Expresiones cron](#) en la Guía del usuario de Amazon EventBridge.
 - Elija un horario que se ejecute a un ritmo regular, por ejemplo, cada 10 minutos. y, a continuación, introduzca una expresión de frecuencia.
12. Elija Siguiente.
13. En Tipos de destino, elija Servicio de AWS.
14. En Seleccione un objetivo, elija Cola de trabajos por lotes. A continuación, configure lo siguiente:
 - Cola de trabajos: introduzca el nombre de recurso de Amazon (ARN) de la cola de trabajos en la que va a programar el trabajo.
 - Definición de trabajo: introduzca el nombre y la revisión o el ARN completo de la definición de trabajo que va a utilizar para el trabajo.
 - Nombre de trabajo: introduzca un nombre para el trabajo.
 - Tamaño de la matriz: (opcional) introduzca un tamaño de matriz para que el trabajo ejecute más de una copia. Para obtener más información, consulte [Trabajos de matrices](#).

- Intentos de trabajo: (opcional) introduzca el número de veces que se debe reintentar el trabajo si se produce un error. Para obtener más información, consulte [Reintentos automáticos de trabajo](#).
15. En los tipos de destino de la opción Cola de trabajos por lotes, EventBridge necesita permiso para enviar eventos al destino. EventBridge puede crear el rol de IAM necesario para que se ejecute la regla. Realice una de las siguientes acciones:
 - Para crear un rol de IAM automáticamente, elija Creación de un nuevo rol para este recurso específico.
 - Para utilizar un rol de IAM que haya creado antes, elija Uso de rol existente.
 16. (Opcional) Amplíe Configuración adicional.
 - a. En Configurar la entrada de destino, elija cómo se procesa el texto de un evento antes de pasarlo al destino.
 - b. En Antigüedad máxima del evento, especifique el intervalo de tiempo durante el que se guardan los eventos sin procesar.
 - c. En Intentos de reintento, introduzca el número de veces que se volverá a intentar un evento.
 - d. En Cola de mensajes fallidos, elija una opción para gestionar los eventos no procesados. Si es necesario, especifique la cola de Amazon SQS que se utilizará como la cola de mensajes fallidos.
 17. (Opcional) Elija Agregar otro destino para agregar otro destino para esta regla.
 18. Elija Siguiente.
 19. (Opcional) En Etiquetas, elija Añadir nueva etiqueta para añadir una etiqueta de recurso a la regla. Para obtener más información, consulte [Etiquetas de Amazon EventBridge](#).
 20. Elija Siguiente.
 21. En Revisar y crear, revise los pasos de configuración. Si necesita realizar cambios, elija Editar. Cuando haya terminado, seleccione Creación de regla.

Para obtener más información sobre cómo crear reglas, consulte [Creación de reglas de EventBridge que reaccionan a eventos](#) en la Guía del usuario de Amazon EventBridge.

Tutorial: creación de una regla con un patrón de evento

El siguiente procedimiento explica cómo crear una regla con un patrón de evento.

Cómo crear una regla que envíe el evento a un destino cuando el evento coincida con un patrón definido

Note

Este procedimiento funciona para todos los trabajos de AWS Batch en Amazon ECS, Amazon EKS y AWS Fargate.

1. Abra la consola de Amazon EventBridge en <https://console.aws.amazon.com/events/>.
2. En la barra de navegación, elija la Región de AWS a utilizar.
3. En el panel de navegación, elija Reglas.
4. Elija Creación de regla.
5. En Nombre, especifique un nombre único para el entorno de computación. El nombre puede contener hasta 64 caracteres. Puede contener letras mayúsculas y minúsculas, números, guiones (-) y guiones bajos (_).

Note

Una regla no puede tener el mismo nombre que otra regla de la misma región y del mismo bus de eventos.

6. (Opcional) En Descripción, introduzca una descripción para la regla.
7. En Bus de eventos, elija el bus de eventos que desea asociar a esta regla. Si desea que esta regla coincida con eventos procedentes de su cuenta, elija predeterminado. Cuando un Servicio de AWS en su cuenta emite un evento, siempre va al bus de eventos predeterminado de su cuenta.
8. (Opcional) Desactive la regla en el bus seleccionado si no desea ejecutarla inmediatamente.
9. En Tipo de regla, elija Regla con un patrón de evento.
10. Seleccione Siguiente.
11. En Origen del evento, elija AWS Eventos o eventos de socios de EventBridge.
12. (Opcional) En Ejemplo de evento:
 - a. En Ejemplo de tipo de evento, elija eventos de AWS.
 - b. En Eventos de muestra, elija Cambiar estado de trabajo por lotes.

13. En Método de creación, elija Uso de forma de patrón.
14. En Patrón de eventos:
 - a. En Origen del evento, elija Servicios de AWS.
 - b. En Servicio de AWS, elija Lote.
 - c. En Tipo de evento, elija Cambio de estado de trabajo por lotes.
15. Seleccione Siguiente.
16. En Tipos de destino, elija Servicio de AWS.
17. En Elegir un tipo de destino, elija un tipo de destino. Por ejemplo, elija Cola de trabajos por lotes. A continuación, especifique lo siguiente:
 - Cola de trabajos: introduzca el nombre de recurso de Amazon (ARN) de la cola de trabajos en la que va a programar el trabajo.
 - Definición de trabajo: introduzca el nombre y la revisión o el ARN completo de la definición de trabajo que va a utilizar para el trabajo.
 - Nombre de trabajo: introduzca un nombre para el trabajo.
 - Tamaño de la matriz: (opcional) introduzca un tamaño de matriz para que el trabajo ejecute más de una copia. Para obtener más información, consulte [Trabajos de matrices](#).
 - Intentos de trabajo: (opcional) introduzca el número de veces que se debe reintentar el trabajo si se produce un error. Para obtener más información, consulte [Reintentos automáticos de trabajo](#).
18. En los tipos de destino de la opción Cola de trabajos por lotes, EventBridge necesita permiso para enviar eventos al destino. EventBridge puede crear el rol de IAM necesario para que se ejecute la regla. Realice una de las siguientes acciones:
 - Para crear un rol de IAM automáticamente, elija Creación de un nuevo rol para este recurso específico.
 - Para utilizar un rol de IAM que haya creado antes, elija Uso de función existente.
19. (Opcional) Amplíe Configuración adicional.
 - a. En Configurar la entrada de destino, elija cómo se procesa el texto de un evento.
 - b. En Antigüedad máxima del evento, especifique el intervalo de tiempo durante el que se guardan los eventos sin procesar.
 - c. En Intentos de reintento, introduzca el número de veces que se volverá a intentar un evento.

- d. En Cola de mensajes fallidos, elija una opción para gestionar los eventos no procesados. Si es necesario, especifique la cola de Amazon SQS que se utilizará como la cola de mensajes fallidos.
- 20. (Opcional) Elija Agregar otro destino para agregar otro destino para esta regla.
- 21. Elija Siguiente.
- 22. (Opcional) En Etiquetas, elija Añadir nueva etiqueta para añadir una etiqueta de recurso. Para obtener más información, consulte [Etiquetas de Amazon EventBridge](#) en la Guía del usuario de Amazon EventBridge.
- 23. Elija Siguiente.
- 24. En Revisar y crear, revise los pasos de configuración. Si necesita realizar cambios, elija Editar. Cuando haya terminado, elija Creación de regla.

Para obtener más información sobre cómo crear la regla, consulte [Creación de reglas de EventBridge que reaccionan a eventos](#) en la Guía del usuario de Amazon EventBridge.

Tutorial: Pase la información del evento a un AWS Batch objetivo según un cronograma utilizando el transformador EventBridge de entrada

Puede usar el transformador EventBridge de entrada para pasar la información del evento al AWS Batch enviar un trabajo. Esto puede resultar especialmente valioso si se invocan trabajos como resultado de otra información sobre eventos de AWS . Un ejemplo es la carga de un objeto en un bucket de Amazon S3. También puede usar una definición de trabajo con valores de sustitución de parámetros en el comando del contenedor. El transformador EventBridge de entrada puede proporcionar los valores de los parámetros en función de los datos del evento.

A continuación, se crea un objetivo de AWS Batch evento que analiza la información del evento que lo inicia y la transforma en un `parameters` objeto. Cuando se ejecuta el trabajo, los parámetros del evento que se desencadena se pasan al comando del contenedor del trabajo.

Note

En este escenario, todos los AWS recursos (como los depósitos, EventBridge las reglas y los CloudTrail registros de Amazon S3) deben estar en la misma región.

Para crear un AWS Batch objetivo que utilice el transformador de entrada

1. Abre la EventBridge consola de Amazon en <https://console.aws.amazon.com/events/>.
2. En la barra de navegación, selecciona la Región de AWS que deseas usar.
3. En el panel de navegación, seleccione Reglas.
4. Elija Creación de regla.
5. En Nombre, especifique un nombre único para el entorno de computación. El nombre puede contener hasta 64 caracteres. Puede contener letras mayúsculas y minúsculas, números, guiones (-) y guiones bajos (_).

 Note

Una regla no puede tener el mismo nombre que otra regla en el mismo bus de eventos Región de AWS y en el mismo.

6. (Opcional) En Descripción, introduzca una descripción para la regla.
7. En Bus de eventos, elija el bus de eventos que desea asociar a esta regla. Si desea que esta regla coincida con eventos procedentes de su cuenta, seleccione predeterminado. Cuando un Servicio de AWS elemento de tu cuenta emite un evento, siempre va al bus de eventos predeterminado de tu cuenta.
8. (Opcional) Desactive la regla en el bus seleccionado si no desea ejecutarla inmediatamente.
9. En Tipo de regla, elija Programación.
10. Seleccione Continuar para crear la regla o Siguiente.
11. En Programar patrón, realice una de las siguientes acciones:
 - Elija Un horario detallado que se ejecute a una hora específica, como las 8:00 a. m. PST el primer lunes de cada mes y después introduzca la expresión cron. Para obtener más información, consulte [Cron Expressions](#) en la Guía del EventBridge usuario de Amazon.
 - Elija un horario que se ejecute a un ritmo regular, por ejemplo, cada 10 minutos. y, a continuación, introduzca una expresión de frecuencia.
12. Elija Siguiente.
13. En Tipos de destino, elija Servicio de AWS.
14. En Seleccione un objetivo, elija Cola de trabajos por lotes. A continuación, configure lo siguiente:

- Cola de trabajos: introduzca el nombre de recurso de Amazon (ARN) de la cola de trabajos en la que va a programar el trabajo.
 - Definición de trabajo: introduzca el nombre y la revisión o el ARN completo de la definición de trabajo que va a utilizar para el trabajo.
 - Nombre de trabajo: introduzca un nombre para el trabajo.
 - Tamaño de la matriz: (opcional) introduzca un tamaño de matriz para que el trabajo ejecute más de una copia. Para obtener más información, consulte [Trabajos de matrices](#).
 - Intentos de trabajo: (opcional) introduzca el número de veces que se debe reintentar el trabajo si se produce un error. Para obtener más información, consulte [Reintentos automáticos de trabajo](#).
15. Para los tipos de objetivos de cola de trabajos por lotes, EventBridge necesita permiso para enviar eventos al destino. EventBridge puede crear la función de IAM necesaria para que se ejecute la regla. Realice una de las siguientes acciones:
 - Para crear un rol de IAM automáticamente, elija Creación de un nuevo rol para este recurso específico.
 - Para utilizar un rol de IAM que haya creado antes, elija Uso de rol existente.
 16. (Opcional) Amplíe Configuración adicional.
 17. En la sección Ajustes adicionales, en Configurar entrada de destino, elija Transformador de entrada.
 18. Elija Configurar transformador de entrada.
 19. (Opcional) En Ejemplo de evento:
 - a. En Ejemplo de tipo de evento, elija eventos de AWS .
 - b. En Eventos de muestra, elija Cambiar estado de trabajo por lotes.
 20. En la sección Transformador de entrada de destino en Ruta de entrada, especifique los valores que se van a analizar del evento que se desencadena. Por ejemplo, para analizar el evento Cambiar estado de trabajo por lotes, utilice el siguiente formato JSON.

```
{
  "instance": "$.detail.jobId",
  "state": "$.detail.status"
}
```

21. En Plantilla, introduzca lo siguiente.

```
{
  "instance": <jobId> ,
  "status": <status>
}
```

22. Elija Confirmar.
23. En Antigüedad máxima del evento, especifique el intervalo de tiempo durante el que se guardan los eventos sin procesar.
24. En Intentos de reintento, introduzca el número de veces que se volverá a intentar un evento.
25. En Cola de mensajes fallidos, elija una opción para gestionar los eventos no procesados. Si es necesario, especifique la cola de Amazon SQS que se utilizará como la cola de mensajes fallidos.
26. (Opcional) Elija Agregar otro destino para agregar otro destino para esta regla.
27. Elija Siguiente.
28. (Opcional) En Etiquetas, elija Añadir nueva etiqueta para añadir una etiqueta de recurso. Para obtener más información, consulta las [EventBridge etiquetas de Amazon](#) en la Guía del EventBridge usuario de Amazon.
29. Elija Siguiente.
30. En Revisar y crear, revise los pasos de configuración. Si necesita realizar cambios, elija Editar. Cuando haya terminado, elija Creación de regla.

Tutorial: escucha de los eventos de trabajo de AWS Batch con EventBridge

En este tutorial, se configura una función simple AWS Lambda que escucha eventos de trabajo AWS Batch y los escribe en un flujo de registro de los Registros de CloudWatch.

Requisitos previos

En este tutorial se supone que tiene un entorno de computación y una cola de trabajos que están listos para aceptar trabajos. Si no dispone de un entorno de computación y una cola de trabajos para capturar eventos, siga los pasos de [Cómo empezar con AWS Batch los tutoriales](#) para crearlos. Al final de este tutorial, puede enviar opcionalmente un trabajo a esta cola de trabajos para comprobar que ha configurado correctamente su función de Lambda.

Temas

- [Tutorial: creación de la función de Lambda](#)
- [Tutorial: registro de la regla de eventos](#)
- [Tutorial: prueba de la configuración](#)

Tutorial: creación de la función de Lambda

En este procedimiento, creará una función de Lambda sencilla que servirá como destino para los mensajes del flujo de eventos de AWS Batch.

Para crear una función de Lambda de destino

1. Abra la consola de AWS Lambda en <https://console.aws.amazon.com/lambda/>.
2. Elija Create function (Creación de función) y, a continuación, elija Author from scratch (Creación de desde cero).
3. En Nombre de función, introduzca batch-event-stream-handler.
4. En Tiempo de ejecución, seleccione Python 3.8.
5. Seleccione Creación de función.
6. En la sección Código fuente, edite el código de muestra para que coincida con el siguiente ejemplo:

```
import json

def lambda_handler(event, _context):
    # _context is not used
    del _context
    if event["source"] != "aws.batch":
        raise ValueError("Function only supports input from events with a source
type of: aws.batch")

    print(json.dumps(event))
```

Se trata de una función simple de Python 3.8 que imprime los eventos enviados por AWS Batch. Si todo está configurado correctamente, al final de este tutorial verá los detalles de los eventos del flujo de registros de los Registros de CloudWatch asociados a esta función de Lambda.

7. Seleccione Implementar.

Tutorial: registro de la regla de eventos

En esta sección, se crea una regla de evento de EventBridge que captura eventos de trabajos procedentes de sus recursos de AWS Batch. Esta regla captura todos los eventos procedentes de AWS Batch dentro de la cuenta donde se define. Los propios mensajes de trabajo contienen información sobre el origen del evento, incluida la cola de trabajo en la que se envió. Puede utilizar esta información para filtrar y ordenar eventos de manera programática.

Note

Si utiliza el Consola de administración de AWS para crear una regla de evento, la consola añade automáticamente los permisos IAM para EventBridge para llamar a su función de Lambda. Sin embargo, si crea una regla de evento utilizando la AWS CLI, tiene que otorgar permisos explícitamente. Para obtener más información, consulte [Eventos y patrones de eventos en EventBridge](#) en la Guía del usuario de Amazon EventBridge.

Para crear su regla EventBridge

1. Abra la consola de Amazon EventBridge en <https://console.aws.amazon.com/events/>.
2. En el panel de navegación, seleccione Reglas.
3. Elija Creación de regla.
4. Escriba un nombre y una descripción para la regla.

Una regla no puede tener el mismo nombre que otra regla de la misma región y del mismo bus de eventos.

5. En Bus de eventos, seleccione el bus de eventos que desea asociar a esta regla. Si desea que esta regla coincida con eventos procedentes de su cuenta, seleccione Bus de eventos predeterminado de AWS. Cuando un servicio de AWS en su cuenta emite un evento, siempre va al bus de eventos predeterminado de su cuenta.
6. En Tipo de regla, elija Regla con un patrón de evento.
7. Seleccione Siguiente.
8. En Origen del evento, seleccione Otro.
9. En Patrón de eventos, seleccione Patrones personalizados (editor de JSON).
10. Pegue el siguiente patrón de eventos en el área de texto.

```
{
  "source": [
    "aws.batch"
  ]
}
```

Esta regla se aplica a todos los grupos de AWS Batch y a todos los eventos de AWS Batch. También puede crear una regla más específica para filtrar algunos resultados.

11. Elija Siguiente.
12. En Target types (Tipos de destino), elija AWS service.
13. En Seleccionar un objetivo, elija Función de Lambda y seleccione su función de Lambda.
14. (Opcional) En Configuración adicional, haga lo siguiente:
 - a. En Antigüedad máxima del evento, indique un valor entre un minuto (00:01) y 24 horas (24:00).
 - b. En Cantidad de reintentos, indique un número entre 0 y 185.
 - c. En Cola de mensajes fallidos, seleccione si desea utilizar una cola de Amazon SQS estándar como cola de mensajes fallidos. EventBridge envía eventos que coincidan con esta regla a la cola de mensajes fallidos si no se entregan correctamente al destino. Realice una de las siguientes acciones:
 - Seleccione Ninguna para no usar una cola de mensajes fallidos.
 - Elija Seleccionar una cola de Amazon SQS en la cuenta de AWS actual para usarla como cola de mensajes fallidos y luego seleccione de la lista desplegable la cola que quiera usar.
 - Elija Seleccionar una cola de Amazon SQS en otra cuenta de AWS como cola de mensajes fallidos y luego ingrese el ARN de la cola que quiera usar. Debe asociar una política basada en recursos a la cola que conceda permiso a EventBridge para enviarle mensajes. Para obtener más información, consulte [Concesión de permisos a la cola de mensajes fallidos](#) en la Guía del usuario de Amazon EventBridge.
15. Elija Siguiente.
16. (Opcional) Introduzca una o varias etiquetas para la regla. Para obtener más información, consulte [Etiquetas de Amazon EventBridge](#) en la Guía del usuario de Amazon EventBridge.
17. Elija Siguiente.
18. Revise los detalles de la regla y seleccione Creación de regla.

Tutorial: prueba de la configuración

Ahora puede probar la configuración de EventBridge enviando una tarea a la cola de tareas. Si todo está configurado correctamente, su función de Lambda se activa y escribe los datos de eventos en un flujo de registro de los Registros de CloudWatch para la función.

Para probar la configuración

1. Abra la consola AWS Batch en <https://console.aws.amazon.com/batch/>.
2. Envíe un nuevo trabajo de AWS Batch. Para obtener más información, consulte [Tutorial: envío de un trabajo](#).
3. Abra la consola de CloudWatch en <https://console.aws.amazon.com/cloudwatch/>.
4. En el panel de navegación, elija Logs (Registros) y seleccione el grupo de registros para la función de Lambda (por ejemplo, `/aws/lambda/my-function`).
5. Seleccione una secuencia de registro para ver los datos de los eventos.

Tutorial: envío de alertas de Amazon Simple Notification Service para eventos de trabajos fallidos

En este tutorial, usted configura una regla de evento Amazon EventBridge que sólo captura eventos de trabajo donde el trabajo se ha movido a un estado FAILED. Al final de este tutorial, si lo desea, también puede enviar un trabajo a esta cola de trabajos. Esto sirve para comprobar que ha configurado correctamente las alertas de Amazon SNS.

Requisitos previos

En este tutorial se supone que tiene un entorno de computación y una cola de trabajos que están listos para aceptar trabajos. Si no dispone de un entorno de computación y una cola de trabajos para capturar eventos, siga los pasos de [Cómo empezar con AWS Batch los tutoriales](#) para crearlos.

Temas

- [Tutorial: creación de un tema de Amazon SNS y suscribirse a él](#)
- [Tutorial: registro de una regla de eventos](#)
- [Tutorial: prueba de la regla](#)
- [Regla alternativa: cola de trabajos por lotes bloqueada](#)

Tutorial: creación de un tema de Amazon SNS y suscribirse a él

Para este tutorial, se configura un tema de Amazon SNS para utilizarse como destino de eventos para la nueva regla de eventos.

Para crear un tema de Amazon SNS

1. Abra la consola de Amazon SNS en <https://console.aws.amazon.com/sns/v3/home>.
2. Seleccione Topics (Temas), Create topic (Creación de tema).
3. En Tipo, seleccione Estándar.
4. Para Nombre, escriba **JobFailedAlert** y, a continuación, elija Creación de tema.
5. En la pantalla JobFailedAlert, elija Creación de suscripción.
6. En Protocolo, seleccione Correo electrónico.
7. En Endpoint (Punto de conexión), ingrese una dirección de email a la que actualmente tenga acceso y elija Create subscription (Creación de suscripción).
8. Consulte su cuenta de correo electrónico y espere para recibir un mensaje de correo electrónico de confirmación de la suscripción. Cuando lo reciba, seleccione Confirmar suscripción.

Tutorial: registro de una regla de eventos

A continuación, registre una regla de eventos que solo capture los eventos de trabajos con error.

Para registrar su regla de EventBridge

1. Abra la consola de Amazon EventBridge en <https://console.aws.amazon.com/events/>.
2. En el panel de navegación, seleccione Reglas.
3. Elija Creación de regla.
4. Escriba un nombre y una descripción para la regla.

Una regla no puede tener el mismo nombre que otra regla de la misma región y del mismo bus de eventos.

5. En Bus de eventos, seleccione el bus de eventos que desea asociar a esta regla. Si desea que esta regla coincida con eventos procedentes de su cuenta, seleccione Bus de eventos predeterminado de AWS. Cuando un servicio de AWS en su cuenta emite un evento, siempre va al bus de eventos predeterminado de su cuenta.

6. En Tipo de regla, elija Regla con un patrón de evento.
7. Seleccione Siguiente.
8. En Origen del evento, seleccione Otro.
9. En Patrón de eventos, seleccione Patrones personalizados (editor de JSON).
10. Pegue el siguiente patrón de eventos en el área de texto.

```
{
  "detail-type": [
    "Batch Job State Change"
  ],
  "source": [
    "aws.batch"
  ],
  "detail": {
    "status": [
      "FAILED"
    ]
  }
}
```

Este código define una regla de eventos de EventBridge que coincide con cualquier evento en el que el estado del trabajo sea FAILED. Para obtener más información sobre los patrones de eventos, consulte [Eventos y patrones de eventos](#) en la Guía del usuario de Amazon EventBridge.

11. Elija Siguiente.
12. En Tipos de destino, seleccione Servicio de AWS.
13. En Seleccione un destino, elija un tema de SNS y, en Tema, elija JobFailedAlert.
14. (Opcional) En Configuración adicional, haga lo siguiente:
 - a. En Antigüedad máxima del evento, indique un valor entre un minuto (00:01) y 24 horas (24:00).
 - b. En Cantidad de reintentos, indique un número entre 0 y 185.
 - c. En Cola de mensajes fallidos, seleccione si desea utilizar una cola de Amazon SQS estándar como cola de mensajes fallidos. EventBridge envía eventos que coincidan con esta regla a la cola de mensajes fallidos si no se entregan correctamente al destino. Realice una de las siguientes acciones:
 - Seleccione Ninguna para no usar una cola de mensajes fallidos.

- Elija Seleccionar una cola de Amazon SQS en la cuenta de AWS actual para usarla como cola de mensajes fallidos y luego seleccione de la lista desplegable la cola que quiera usar.
- Elija Seleccionar una cola de Amazon SQS en otra cuenta de AWS como cola de mensajes fallidos y luego ingrese el ARN de la cola que quiera usar. Debe asociar una política basada en recursos a la cola que conceda permiso a EventBridge para enviarle mensajes. Para obtener más información, consulte [Concesión de permisos a la cola de mensajes fallidos](#) en la Guía del usuario de Amazon EventBridge.

15. Elija Siguiente.

16. (Opcional) Introduzca una o varias etiquetas para la regla. Para obtener más información, consulte [Etiquetas de Amazon EventBridge](#) en la Guía del usuario de Amazon EventBridge.

17. Elija Siguiente.

18. Revise los detalles de la regla y seleccione Creación de regla.

Tutorial: prueba de la regla

Para probar la regla, envíe un trabajo que se cierre poco después de que comience y tenga un código de salida distinto de cero. Si su regla de eventos está configurada correctamente, debería recibir un mensaje de correo electrónico en unos minutos con el texto del evento.

Para probar una regla

1. Abra la consola AWS Batch en <https://console.aws.amazon.com/batch/>.
2. Envíe un nuevo trabajo de AWS Batch. Para obtener más información, consulte [Tutorial: envío de un trabajo](#). Para el comando del trabajo, sustituya este comando para salir del contenedor con un código de salida de 1.

```
/bin/sh, -c, 'exit 1'
```

3. Compruebe su correo electrónico para confirmar que ha recibido una alerta de correo electrónico para la notificación de trabajo fallido.

Regla alternativa: cola de trabajos por lotes bloqueada

Para crear una regla de eventos que supervise si la cola de trabajos por lotes está bloqueada, repita estos tutoriales con las siguientes modificaciones:

1. En [Tutorial: creación de un tema de Amazon SNS y suscribirse a él](#), utilice *BlockedJobQueue* como nombre del tema.
2. En [Tutorial: registro de una regla de eventos](#), utilice el siguiente patrón en el editor JSON:

```
{
  "detail-type": [
    "Batch Job Queue Blocked"
  ],
  "source": [
    "aws.batch"
  ]
}
```

Elastic Fabric Adapter

Un adaptador Elastic Fabric Adapter (EFA) es un dispositivo de red que acelera las aplicaciones de informática de alto rendimiento (HPC). AWS Batch es compatible con las aplicaciones que utilizan EFA si se cumplen las siguientes condiciones.

- Para obtener una lista de los tipos de instancia que admiten los EFA, consulte [Tipos de instancias compatibles](#) en la Guía del usuario de Amazon EC2.

Tip

Para ver una lista de los tipos de instancia que admiten las EFA en un Región de AWS, ejecute el siguiente comando. Luego, cruce la lista que se muestra con la lista de tipos de instancias disponibles en la consola AWS Batch.

```
$ aws ec2 describe-instance-types --region us-east-1 --filters Name=network-info.efa-supported,Values=true --query "InstanceTypes[*].[InstanceType]" --output text | sort
```

- Para obtener una lista de los sistemas operativos compatibles con EFA, consulte [Sistemas operativos compatibles con](#).
- La AMI tiene cargado el controlador de EFA.
- El grupo de seguridad del adaptador EFA debe permitir todo el tráfico de entrada y salida del propio grupo de seguridad.
- Todas las instancias que utilizan un adaptador EFA deben estar en el mismo grupo con ubicación en clúster.
- La definición del trabajo debe incluir un miembro `devices` con `hostPath` establecido en `/dev/infiniband/uverbs0` para permitir que el dispositivo EFA se transfiera a través del contenedor. Si se especifica `containerPath`, también debe establecerse en `/dev/infiniband/uverbs0`. Si se configura `permissions`, debe establecerse en `READ | WRITE | MKNOD`.

La ubicación de los miembros [LinuxParameters](#) será diferente en los trabajos paralelos de varios nodos y los trabajos de contenedor de un solo nodo. Los ejemplos siguientes ilustran las diferencias, pero faltan algunos valores necesarios.

Example Ejemplo de un trabajo paralelo con varios nodos

```
{
  "jobDefinitionName": "EFA-MNP-JobDef",
  "type": "multinode",
  "nodeProperties": {
    ...
    "nodeRangeProperties": [
      {
        ...
        "container": {
          ...
          "linuxParameters": {
            "devices": [
              {
                "hostPath": "/dev/infiniband/uverbs0",
                "containerPath": "/dev/infiniband/uverbs0",
                "permissions": [
                  "READ", "WRITE", "MKNOD"
                ]
              },
            ],
          },
        },
      },
    ],
  },
},
}
```

Example Ejemplo de un trabajo de contenedor con un solo nodo

```
{
  "jobDefinitionName": "EFA-Container-JobDef",
  "type": "container",
  ...
  "containerProperties": {
    ...
    "linuxParameters": {
      "devices": [
        {
          "hostPath": "/dev/infiniband/uverbs0",
        },
      ],
    },
  },
}
```

```
    ],  
    },  
  },  
}
```

Para obtener más información sobre EFA, consulte [Elastic Fabric Adapter](#) en la Guía del usuario de Amazon EC2.

Supervisar AWS Batch

El monitoreo es una parte importante del mantenimiento de la confiabilidad, la disponibilidad y el rendimiento de AWS Batch su AWS solución.

Le recomendamos encarecidamente que recopile datos de supervisión de todas las partes de su AWS solución para facilitar la depuración de un fallo multipunto, en caso de que se produzca. Comience por crear un plan de monitoreo que responda a las siguientes preguntas. Si no estás seguro de cómo responder a estas preguntas, puedes seguir utilizando Amazon CloudWatch Logs para establecer tus niveles de referencia de rendimiento.

- ¿Cuáles son los objetivos de la monitorización?
- ¿Qué recursos va a monitorizar?
- ¿Con qué frecuencia va a monitorizar estos recursos?
- ¿Qué herramientas de monitorización va a utilizar?
- ¿Quién se encargará de realizar las tareas de supervisión?
- ¿Quién debería recibir una notificación cuando surjan problemas?

El siguiente paso es establecer una línea base de AWS Batch rendimiento normal en su entorno midiendo el rendimiento en distintos momentos y bajo diferentes condiciones de carga. Mientras monitorea AWS Batch, guarde los datos de monitoreo históricos para poder compararlos con los datos de rendimiento actuales. Esto le ayudará a identificar patrones de rendimiento normales y anomalías de rendimiento, así como a idear métodos para abordarlos.

Los temas de esta sección pueden ayudarlo a iniciar el registro y el monitoreo de AWS Batch.

Temas

- [Uso CloudWatch de registros con AWS Batch](#)
- [AWS Batch Información de contenedores de CloudWatch](#)
- [Utilice CloudWatch los registros para supervisar AWS Batch los trabajos de Amazon EKS](#)

Uso CloudWatch de registros con AWS Batch

Puede configurar sus AWS Batch trabajos en los recursos de EC2 para enviar información de registro y métricas detalladas a CloudWatch los registros. De esta forma, puede ver distintos

registros desde sus trabajos en una ubicación cómoda. Para obtener más información sobre CloudWatch los registros, consulta [¿Qué es Amazon CloudWatch Logs?](#) en la Guía del CloudWatch usuario de Amazon.

 Note

De forma predeterminada, CloudWatch los registros están activados para los AWS contenedores Fargate.

Para activar y personalizar el registro de CloudWatch registros, revisa las siguientes tareas de configuración únicas:

- Para los entornos AWS Batch informáticos que se basan en recursos de EC2, añada una política de IAM a la `ecsInstanceRole` función. Para obtener más información, consulte [the section called “Tutorial: Añadir una política de CloudWatch IAM de registros”](#).
- Cree una plantilla de lanzamiento de Amazon EC2 que incluya una CloudWatch supervisión detallada y, a continuación, especifique la plantilla al crear su entorno AWS Batch informático. También puede instalar el CloudWatch agente en una imagen existente y, a continuación, especificar la imagen en el asistente de AWS Batch primera ejecución.
- (Opcional) Configure el controlador `awslogs`. Puede agregar parámetros que cambien el comportamiento predeterminado en los recursos de EC2 y Fargate. Para obtener más información, consulte [the section called “Uso del controlador de registros `awslogs`”](#).

Temas

- [Tutorial: Añadir una política de CloudWatch IAM de registros](#)
- [Instale y configure el CloudWatch agente](#)
- [Tutorial: Ver CloudWatch registros](#)

Tutorial: Añadir una política de CloudWatch IAM de registros

Antes de que sus trabajos puedan enviar datos de registro y métricas detalladas a CloudWatch Logs, debe crear una política de IAM que utilice los CloudWatch registros. APIs Después de crear la política de IAM, asóciela al rol `ecsInstanceRole`.

Note

Si la ECS-CloudWatchLogs política no está asociada a la `ecsInstanceRole` función, las métricas básicas se pueden seguir enviando a CloudWatch Logs. Sin embargo, las métricas básicas no incluyen datos de registro ni métricas detalladas, como el espacio libre en disco.

AWS Batch los entornos de cómputo utilizan recursos de Amazon EC2. Al crear un entorno informático mediante el asistente de AWS Batch primera ejecución, AWS Batch crea el `ecsInstanceRole` rol y configura el entorno con él.

Si no utiliza el asistente de primera ejecución, puede especificar la `ecsInstanceRole` función al crear un entorno informático en la AWS Command Line Interface API o. AWS Batch Para obtener más información, consulte [Referencia de comandos AWS CLI](#) o [Referencia de la API AWS Batch](#).

Para crear la política de IAM **ECS-CloudWatchLogs**

1. Abra la consola de IAM en <https://console.aws.amazon.com/iam/>.
2. En el panel de navegación, seleccione Políticas.
3. Elija Creación de política.
4. Elija JSON y, a continuación, escriba la política siguiente:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "logs:DescribeLogStreams"
      ],
      "Resource": [
        "arn:aws:logs:*:*:*"
      ]
    }
  ]
}
```



```
]
}
```

5. Elija Siguiente: etiquetas.
6. (Opcional) En Añadir etiquetas, elija Añadir etiqueta para añadir una etiqueta a la política.
7. Elija Siguiente: revisar.
8. En la página Revisar política, en Nombre, escriba **ECS-CloudWatchLogs**, y después introduzca una Descripción opcional.
9. Elija Creación de política.

Asociación de la política **ECS-CloudWatchLogs** a **ecsInstanceRole**

1. Abra la consola de IAM en <https://console.aws.amazon.com/iam/>.
2. Seleccione Roles en el panel de navegación.
3. Elija ecsInstanceRole. Si el rol no existe, siga los procedimientos que se indican en [Rol de instancia de Amazon ECS](#) para crear el rol.
4. Elija Agregar permisos y luego Adjuntar políticas.
5. Elija la CloudWatchLogs política ECS y, a continuación, elija Adjuntar política.

Instale y configure el CloudWatch agente

Puede crear una plantilla de lanzamiento de Amazon EC2 que incluya CloudWatch la supervisión. Para obtener más información, consulte, [Lanzar una instancia desde una plantilla de lanzamiento y Detalles avanzados](#) en la Guía del usuario de Amazon EC2.

También puede instalar el CloudWatch agente en una AMI de Amazon EC2 existente y, a continuación, especificar la imagen en el asistente de AWS Batch primera ejecución. Para obtener más información, consulte [Instalación del CloudWatch agente y. Cómo empezar con AWS Batch los tutoriales](#)

Note

Los AWS Fargate recursos no admiten plantillas de lanzamiento.

Tutorial: Ver CloudWatch registros

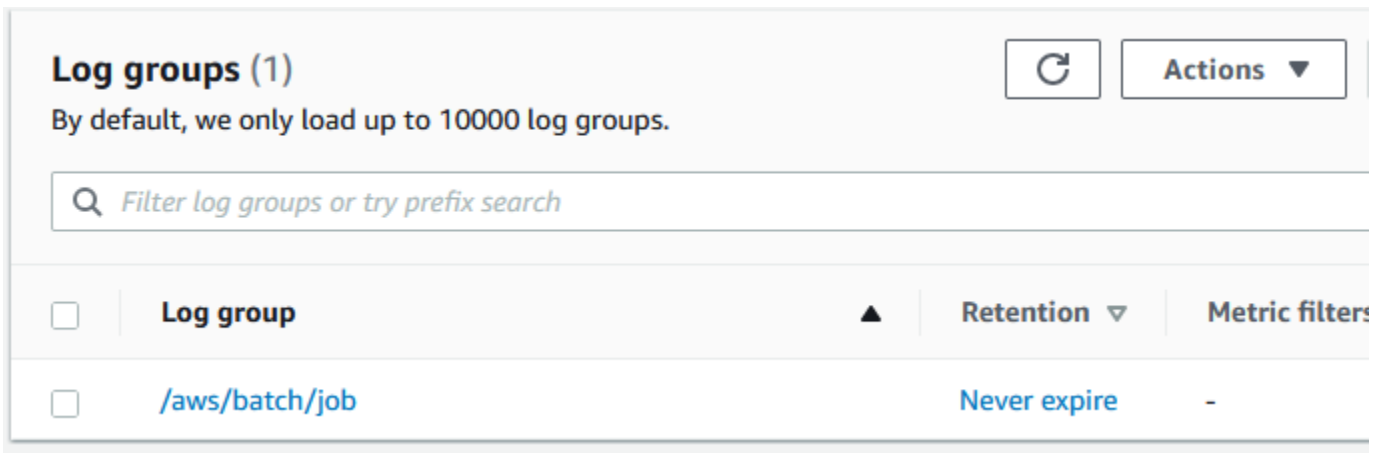
Puede ver y buscar CloudWatch registros en el Consola de administración de AWS.

Note

Los datos pueden tardar unos minutos en mostrarse en CloudWatch los registros.

Para ver los datos CloudWatch de tus registros

1. Abre la CloudWatch consola en <https://console.aws.amazon.com/cloudwatch/>.
2. En el panel de navegación izquierdo, seleccione Registros y, a continuación, Grupos de registros.



3. Seleccione un grupo de registros que desea ver.

Log streams (9)			Delete	Create log stream	Search all
Filter log streams or try prefix search		< 1 >			
☐	Log stream	Last event time			
☐	Test-jd/default/6622fe43-b2a3-4805-a0a6-3828329cc32b	2020-08-18T19:50:19.311Z			
☐	first-run-job-definition/default/86ed75ac-4f3f-4044-8fb0-dfd9c85ae6b2	2020-08-18T02:07:42.738Z			
☐	Test-jd/default/48f4a9dd-be07-4b43-8696-f0995eefe28b	2020-08-14T00:18:19.395Z			
☐	first-run-job-definition/default/d7d5ccf4-a0a0-44f1-bf36-35f2b3632912	2020-08-13T22:39:06.936Z			
☐	gpuJD/default/6ecf8ffb-ee03-4041-aa18-ab5e7a6dff0d	2019-03-26T08:48:39.637Z			

4. Elija el flujo de registros que desea ver. De forma predeterminada, las transmisiones se identifican con los primeros 200 caracteres del nombre del trabajo y el ID de la tarea de Amazon ECS.

Tip

Para descargar los datos del flujo de registro, elija Acciones.

Log events

Actions

Create Metric Filter

Filter events

Clear

1m

30m

1h

12h

Custom

▶

Timestamp

Message

▶

2020-08-17T19:07:42.738-07:00...

There are older events to load. [Load more.](#)

'hello world'

No newer events at this moment. *Auto retry paused.* [Resume](#)

AWS Batch Información de contenedores de CloudWatch

CloudWatch Container Insights recopila, agrega y resume métricas y registros de sus trabajos y entornos de computación de AWS Batch. Las métricas incluyen la utilización de recursos como CPU, memoria, disco y red. Puede añadir estas métricas a los paneles de CloudWatch.

Los datos operativos se recopilan como eventos de registro de rendimiento. Son entradas que utilizan un esquema JSON estructurado que permite incorporar y almacenar datos de cardinalidad alta a escala. A partir de estos datos, CloudWatch crea métricas agregadas de nivel superior a nivel de entorno de computación y de trabajo como métricas CloudWatch. Para obtener más información, consulte [Registros estructurados de Container Insights para Amazon ECS](#) en la Guía del usuario de Amazon CloudWatch.

Important

Las métricas recopiladas por CloudWatch Container Insights se cobran como métricas personalizadas. Para obtener más información, consulte los [precios de los Eventos de Amazon CloudWatch](#).

Temas

- [Active Container Insights](#)

Active Container Insights

Siga estos pasos para activar Container Insights en los entornos de computación de AWS Batch.

1. Abra la [consola de AWS Batch](#).
2. Seleccione Environments (Entornos).
3. Elija el entorno de desarrollo que quiera.
4. En la pestaña Información de contenedores, habilite Información de contenedores para el entorno de computación.

Tip

Puede seleccionar un intervalo predeterminado para agregar las métricas o crear un intervalo personalizado.

De forma predeterminada, se muestran las siguientes métricas. Para obtener una lista completa de las métricas de Amazon ECS Container Insights, consulte [Métricas de Amazon ECS Container Insights](#) en la Guía del usuario de Amazon CloudWatch.

- **JobCount** – El número de trabajos que se ejecutan en el entorno de computación.
- **ContainerInstanceCount** – El número de instancias de Amazon Elastic Compute Cloud que ejecutan el agente de Amazon ECS y están registradas en el entorno de computación.
- **MemoryReserved** – La memoria que se reserva para los trabajos del entorno de computación. Esta métrica se recopila únicamente para las tareas que tienen una reserva de memoria definida en su definición de tarea.
- **MemoryUtilized** – La memoria que están utilizando los trabajos del entorno de computación. Esta métrica se recopila únicamente para las tareas que tienen una reserva de memoria definida en su definición de tarea.
- **CpuReserved** – Las unidades CPU que se reservan para los trabajos del entorno de computación. Esta métrica se recopila únicamente para las tareas que tienen una reserva definida de la CPU en la definición de contenedor.
- **CpuUtilized** – Las unidades CPU utilizadas por los trabajos en el entorno de computación. Esta métrica se recopila únicamente para las tareas que tienen una reserva definida de la CPU en la definición de contenedor.
- **NetworkRxBytes** - El número de bytes que se reciben. Esta métrica solo está disponible para los contenedores en tareas que utilizan awsvpc o los modos de red puente.
- **NetworkTxBytes** – El número de bytes que se transmiten. Esta métrica solo está disponible para los contenedores en tareas que utilizan awsvpc o los modos de red puente.
- **StorageReadBytes** – El número de bytes que se leen del almacenamiento.
- **StorageWriteBytes** – El número de bytes que se escriben en el almacenamiento.

Utilice CloudWatch los registros para supervisar AWS Batch los trabajos de Amazon EKS

Puedes usar Amazon CloudWatch Logs para monitorear, almacenar y ver todos tus archivos de registro en un solo lugar. Con CloudWatch Logs, puede buscar, filtrar y analizar los datos de registro de varias fuentes.

Puede descargar una Fluent Bit imagen de AWS formulario que incluye un complemento para supervisar AWS Batch los trabajos de Amazon EKS en CloudWatch Logs. Fluent Bites un

procesador y reenviador de registros de código abierto que es compatible con Docker y es compatible. Kubernetes Le recomendamos que utilice Fluent Bit como router de registro porque consume menos recursos que Fluentd. Para obtener más información, consulte [Instalar el CloudWatch agente con el complemento Amazon CloudWatch Observability EKS o el gráfico Helm](#).

Requisitos previos

- Adjunte la CloudWatchAgentServerPolicy política a la AWS Identity and Access Management política de sus nodos trabajadores. Para obtener más información, consulte [Verificación de los requisitos previos](#).

Instalación del complemento

Para obtener instrucciones sobre cómo instalar AWS Fluent Bit y crear los CloudWatch grupos, consulte [Instalar el CloudWatch agente con el complemento Amazon CloudWatch Observability EKS o el diagrama de Helm](#).

Debe ingresar los siguientes [datos de configuración adicionales](#) al instalar el complemento:

- Si instala el complemento con el, Consola de administración de AWS debe proporcionar las siguientes tolerancias en los valores de configuración:

```
{
  "tolerations": [
    {
      "key": "batch.amazonaws.com/batch-node",
      "operator": "Exists"
    }
  ]
}
```

- Si instala el complemento con AWS CLI , añada los siguientes argumentos:

```
--configuration-values '{"tolerations":[{"key":"batch.amazonaws.com/batch-node","operator":"Exists"}]}'
```

 Tip

Recuerde que Fluent Bit utiliza 0,5 CPU y 100 MB de memoria en los nodos AWS Batch . Esto reduce la capacidad total disponible para los AWS Batch trabajos. Tenga esto en cuenta a la hora de dimensionar sus trabajos.

Etiquetar los recursos de AWS Batch

Para ayudarle a administrar sus recursos de AWS Batch, puede asignar sus propios metadatos a cada recurso en forma de etiquetas. En este tema se describe qué son las etiquetas y cómo crearlas.

Temas

- [Conceptos básicos de etiquetas](#)
- [Etiquetar los recursos](#)
- [Restricciones de las etiquetas](#)
- [Tutorial: administración de etiquetas con la consola](#)
- [Administración de etiquetas mediante el uso de CLI o API](#)

Conceptos básicos de etiquetas

Una etiqueta es una marca que se asigna a un recurso de AWS. Cada etiqueta está formada por una clave y un valor opcional, ambos definidos por el usuario.

Las etiquetas le permiten clasificar los recursos de AWS según, por ejemplo, su finalidad, propietario o entorno. Cuando tenga muchos recursos del mismo tipo, puede identificar rápidamente un recurso específico en función de las etiquetas que le haya asignado. Por ejemplo, puede definir un conjunto de etiquetas para los servicios de AWS Batch para ayudarle a realizar un seguimiento del propietario y del nivel de pila de cada servicio. Le recomendamos que diseñe un conjunto coherente de claves de etiqueta para cada tipo de recurso.

Además, las etiquetas no se asignan a los recursos automáticamente. Después de agregar una etiqueta, puede editar las claves y los valores de las etiquetas o eliminar etiquetas de un recurso en cualquier momento. Si elimina un recurso, también se eliminará cualquier etiqueta asignada a dicho recurso.

Las etiquetas no tienen ningún significado semántico para AWS Batch, por lo que se interpretan estrictamente como cadenas de caracteres. Puede establecer el valor de una etiqueta como una cadena vacía, pero no puede asignarle un valor nulo. Si añade una etiqueta con la misma clave que una etiqueta existente en ese recurso, el nuevo valor sobrescribirá al antiguo.

Puede trabajar con etiquetas utilizando la Consola de administración de AWS, la AWS CLI y la API de AWS Batch.

Si utiliza AWS Identity and Access Management (IAM), puede controlar qué usuarios de su cuenta de AWS tienen permiso para crear, editar o eliminar etiquetas.

Etiquetar los recursos

Puede etiquetar entornos de computación de AWS Batch, trabajos, definiciones de trabajos, colas de trabajos y políticas de programación nuevos o existentes.

Si utiliza la consola de AWS Batch, puede aplicar etiquetas a los recursos de nueva creación o a los recursos existentes cuando lo desee mediante la pestaña Etiquetas en la página de recursos en cuestión.

Si utiliza la API de AWS Batch, la AWS CLI o un SDK de AWS, puede aplicar etiquetas a los recursos nuevos mediante el parámetro de `tags` en la acción de la API pertinente o utilizar la acción de la API de `TagResource` para aplicar etiquetas a los recursos existentes. Para obtener más información, consulte [TagResource](#).

Además, algunas acciones de creación de recursos le permiten especificar etiquetas para un recurso al crearlo. Si no se pueden aplicar etiquetas durante la creación del recurso, el proceso de creación de recursos falla. Esto garantiza que los recursos que pretendía etiquetar en el momento de su creación se creen con etiquetas específicas o no se creen en absoluto. Si etiqueta recursos en el momento de su creación, no es necesario ejecutar scripts de etiquetado personalizados después de la creación del recurso.

En la tabla siguiente se describen los recursos de AWS Batch que se pueden etiquetar y aquellos que se pueden etiquetar en el momento de su creación.

Asistencia de etiquetas para recursos de AWS Batch

Recurso	Admite etiquetas	Admite la propagación de etiquetas	Admite el etiquetado durante la creación (API de AWS Batch, AWS CLI y SDK de AWS)
Entornos de computación AWS Batch	Sí	No. Las etiquetas del entorno de computación no se propagan a ningún otro recurso. Las etiquetas de los	Sí

Recurso	Admite etiquetas	Admite la propagación de etiquetas	Admite el etiquetado o durante la creación (API de AWS Batch, AWS CLI y SDK de AWS)
		recursos se especifican en el miembro tags del objeto ComputeResources pasado en la operación de API CreateComputeEnvironment .	
AWS Batch Trabajos de	Sí	Sí	Sí
Definiciones de trabajo de AWS Batch	Sí	No	Sí
Colas de trabajo de AWS Batch	Sí	No	Sí
Políticas de programación de AWS Batch	Sí	No	Sí

Restricciones de las etiquetas

Se aplican las siguientes restricciones básicas a las etiquetas:

- Número máximo de etiquetas por recurso: 50
- Para cada recurso, cada clave de etiqueta debe ser única y solo puede tener un valor.
- Longitud máxima de la clave: 128 caracteres Unicode en UTF-8
- Longitud máxima del valor: 256 caracteres Unicode en UTF-8
- Si se utiliza su esquema de etiquetado en múltiples servicios y recursos de AWS, recuerde que otros servicios pueden tener restricciones sobre caracteres permitidos. Los caracteres permitidos

generalmente son: letras, números y espacios representables en UTF-8, además de los siguientes caracteres: + - = . _ : / @.

- Las claves y los valores de las etiquetas distinguen entre mayúsculas y minúsculas.
- No utilice `aws :`, `AWS :` ni ninguna combinación de mayúsculas o minúsculas del mismo como prefijo para claves o valores, ya que está reservado para uso de AWS. Las claves y valores de etiquetas que tienen este prefijo no se pueden editar. Las etiquetas que tengan este prefijo no cuentan para el límite de etiquetas por recurso.

Tutorial: administración de etiquetas con la consola

Con la AWS Batch consola, puede administrar las etiquetas asociadas a los entornos informáticos, los trabajos, las definiciones de trabajos y las colas de trabajos nuevos o existentes.

Adición de etiquetas a un recurso individual durante su creación

Puede añadir etiquetas a los entornos AWS Batch informáticos, los trabajos, las definiciones de trabajos, las colas de trabajos y las políticas de programación al crearlos.

Agregar y eliminar etiquetas en un recurso individual

AWS Batch le permite añadir o eliminar las etiquetas asociadas a sus clústeres directamente desde la página del recurso.

Para agregar o eliminar una etiqueta en un recurso individual

1. Abra la AWS Batch consola en <https://console.aws.amazon.com/batch/>.
2. En la barra de navegación, elija la región a utilizar.
3. En el panel de navegación, elija un tipo de recurso (por ejemplo, Colas de trabajos).
4. Elija un recurso específico y, a continuación, elija Editar etiquetas.
5. Agregue o elimine sus etiquetas según sea necesario.
 - Para añadir una etiqueta, especifique la clave y el valor en los cuadros de texto vacíos al final de la lista.
 - Para eliminar una etiqueta, seleccione el botón

Delete icon
junto a la etiqueta.

- Repita este proceso para cada etiqueta que desee agregar o eliminar y, a continuación, elija Editar etiquetas para finalizar.

Administración de etiquetas mediante el uso de CLI o API

Utilice los siguientes comandos de AWS CLI u operaciones de la API de AWS Batch para agregar, actualizar, enumerar y eliminar las etiquetas de sus recursos.

Asistencia de etiquetas para recursos de AWS Batch

Tarea	Acción de la API	AWS CLI	AWS Tools for Windows PowerShell
Agregar o sobrescribir una o varias etiquetas.	TagResource	tag-resource	Add-BATResourceTag
Eliminar una o varias etiquetas.	UntagResource	untag-resource	Remove-BATResourceTag
Enumera las etiquetas de un recurso	ListTagsForResource	list-tags-for-resource	Get-BATResourceTag

Los siguientes ejemplos muestran cómo agregar o quitar etiquetas a los recursos mediante la AWS CLI.

Ejemplo 1: Etiquetar un recurso existente

El siguiente comando etiqueta un recurso existente.

```
aws batch tag-resource --resource-arn resource_ARN --tags team=devs
```

Ejemplo 2: Eliminar la etiqueta de un recurso existente

El siguiente comando elimina una etiqueta de un recurso existente.

```
aws batch untag-resource --resource-arn resource_ARN --tag-keys tag_key
```

Ejemplo 3: enumerar etiquetas de un recurso

El siguiente comando enumera las etiquetas asociadas a un recurso existente.

```
aws batch list-tags-for-resource --resource-arn resource_ARN
```

Algunas acciones de creación de recursos le permiten especificar etiquetas al crear el recurso. Las siguientes acciones admiten etiquetado durante la creación.

Tarea	Acción de la API	AWS CLI	AWS Tools for Windows PowerShell
Creación de un entorno de computación	CreateComputeEnvironment	create-compute-environment	New-BATComputeEnvironment
Creación de una cola de trabajos	CreateJobQueue	create-job-queue	New-BATJobQueue
Creación de una política de programación	CreateSchedulingPolicy	create-scheduling-policy	New-BATSchedulingPolicy
Registrar una definición de trabajo	RegisterJobDefinition	register-job-definition	Register-BATJobDefinition
Enviar un trabajo	SubmitJob	submit-job	Submit-BAT Job
Creación del recurso consumible	CreateConsumableResource	create-consumable-resource	Create-BATConsumableResource

Prácticas recomendadas de AWS Batch

Puede utilizar AWS Batch para ejecutar diversas cargas de trabajo computacionales exigentes a escala sin administrar una arquitectura compleja. Los trabajos AWS Batch se pueden utilizar en una amplia gama de casos de uso en áreas como la epidemiología, los juegos y el machine learning.

En este tema se describen las prácticas recomendadas que se deben tener en cuenta a la hora de utilizar AWS Batch y las directrices sobre cómo ejecutar y optimizar las cargas de trabajo al utilizar AWS Batch.

Temas

- [Cuándo se debe usar AWS Batch](#)
- [Lista de comprobación para ejecutarla a escala](#)
- [Optimice los contenedores y las AMI](#)
- [Elija el recurso de entorno de computación adecuado](#)
- [Amazon EC2 bajo demanda o Amazon EC2 Spot](#)
- [Prácticas recomendadas para instancias de spot de Amazon EC2 para AWS Batch](#)
- [Errores comunes y solución de problemas](#)

Cuándo se debe usar AWS Batch

AWS Batch ejecuta trabajos a escala y a bajo costo, y proporciona servicios de colas y escalado optimizados en función de los costos. Sin embargo, no todas las cargas de trabajo son adecuadas para su ejecución utilizando AWS Batch.

- **Trabajos cortos:** si un trabajo se ejecuta solo durante unos segundos, la sobrecarga necesaria para programar el trabajo por lotes puede llevar más tiempo que el tiempo de ejecución del trabajo en sí. Como solución alternativa, binpack sus tareas antes de enviarlas en AWS Batch. A continuación, configure los trabajos AWS Batch para que se repitan sobre las tareas. Por ejemplo, coloque los argumentos de cada tarea en una tabla de Amazon DynamoDB o en un archivo en un bucket de Amazon S3. Considere la posibilidad de agrupar las tareas de forma que las tareas se ejecuten durante 3 a 5 minutos cada una. Después de binpack los trabajos, repase los grupos de tareas dentro de su trabajo AWS Batch.
- **Trabajos que deben ejecutarse inmediatamente:** AWS Batch puede procesar trabajos rápidamente. Sin embargo, AWS Batch es un programador y optimiza buscando la rentabilidad, la prioridad

del trabajo y el rendimiento. AWS Batch podría requerir tiempo para procesar sus solicitudes. Si necesita una respuesta en menos de unos segundos, lo más adecuado es un enfoque basado en servicios que utilice Amazon ECS o Amazon EKS.

Lista de comprobación para ejecutarla a escala

Antes de ejecutar una gran carga de trabajo en 50 000 o más vCPU, tenga en cuenta la siguiente lista de comprobación.

Note

Si planea ejecutar una gran carga de trabajo en un millón o más de vCPU o necesita orientación para ejecutar a gran escala, póngase en contacto con su equipo de AWS.

- Compruebe sus cuotas de Amazon EC2: compruebe su cuota de servicio de Amazon EC2 (también conocidas como límites) en el panel Service Quotas de la Consola de administración de AWS. Si es necesario, solicite un aumento de cuota para el número máximo de instancias de Amazon EC2. Recuerde que las instancias de Amazon EC2 Spot y las instancias bajo demanda de Amazon tienen cuotas distintas. Para obtener más información, consulte [Introducción a Service Quotas](#).
- Verifique su cuota de Amazon Elastic Block Store para cada región: cada instancia utiliza un volumen GP2 o GP3 para el sistema operativo. De forma predeterminada, la cuota de cada Región de AWS es de 300 TiB. Sin embargo, cada instancia utiliza conteos como parte de esta cuota. Por lo tanto, asegúrese de tener esto en cuenta cuando verifique su cuota de Amazon Elastic Block Store para cada región. Si alcanza su cuota, no podrá crear más instancias. Para obtener más información, consulte [Puntos de conexión y cuotas de Amazon Elastic Block Store](#).
- Utilice Amazon S3 para el almacenamiento: Amazon S3 ofrece un alto rendimiento y ayuda a eliminar las conjeturas sobre la cantidad de almacenamiento que se debe aprovisionar en función del número de trabajos e instancias de cada zona de disponibilidad. Para obtener más información, consulte [Patrones de diseño de prácticas recomendadas: optimización del rendimiento de Amazon S3](#).
- Amplíe gradualmente para identificar los cuellos de botella en una fase temprana: en el caso de un trabajo que se ejecute en un millón o más de vCPU, comience con un nivel más bajo y aumente gradualmente para poder identificar los cuellos de botella en una fase temprana. Por ejemplo, comience por ejecutar en 50 mil vCPU. A continuación, aumente el conteo a 200 mil

vCPU y, después, a 500 mil vCPU, y así sucesivamente. En otras palabras, continúe aumentando gradualmente el número de vCPU hasta alcanzar la cantidad deseada de vCPU.

- Supervise para identificar los posibles problemas con antelación: para evitar posibles interrupciones y problemas al ejecutar a gran escala, asegúrese de supervisar tanto la aplicación como la arquitectura. Es posible que se produzcan interrupciones incluso al escalar de 1000 a 5000 vCPUs. Puede utilizar Registros de Amazon CloudWatch para revisar los datos de registro o utilizar CloudWatch Embedded Metrics mediante una biblioteca cliente. Para obtener más información, consulte [Referencia del agente de los Registros de CloudWatch](#) y [aws-embedded-metrics](#).

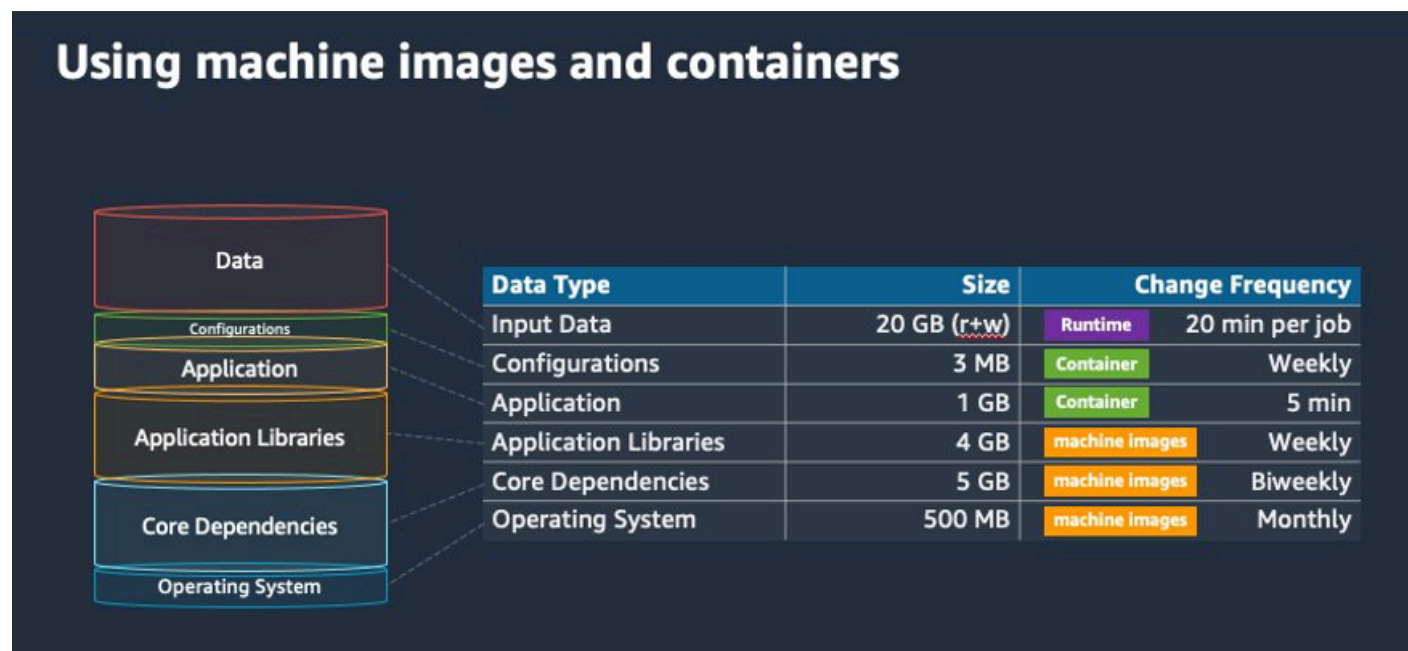
Optimice los contenedores y las AMI

El tamaño y la estructura del contenedor son importantes para la primera serie de trabajos que ejecute. Esto es especialmente cierto si el contenedor tiene más de 4 GB. Las imágenes del contenedor están integradas en capas. Docker recupera las capas en paralelo mediante tres subprocesos simultáneos. Puede aumentar el número de subprocesos simultáneos mediante el parámetro `max-concurrent-downloads`. Para obtener más información, consulte la [documentación de Docker](#).

Aunque puede utilizar contenedores más grandes, le recomendamos que optimice la estructura y el tamaño de los contenedores para acelerar los tiempos de startup.

- Los contenedores más pequeños se obtienen más rápido: los contenedores más pequeños pueden acelerar los tiempos de inicio de las aplicaciones. Para reducir el tamaño del contenedor, descargue las bibliotecas o los archivos que se actualizan con poca frecuencia en la Imagen de máquina de Amazon (AMI). También puede utilizar soportes de encuadernación para dar acceso a sus contenedores. Para obtener más información, consulte [Montajes de enlace](#).
- Cree capas que tengan un tamaño uniforme y divida las capas grandes: cada capa se recupera mediante un hilo. Por lo tanto, una capa grande podría afectar significativamente el tiempo de startup de su trabajo. Recomendamos un tamaño máximo de capa de 2 GB como compensación entre un tamaño de contenedor más grande y tiempos de startup más rápidos. Puede ejecutar el comando `docker history your_image_id` para comprobar la estructura de la imagen del contenedor y el tamaño de la capa. Para obtener más información, consulte la [documentación de Docker](#).
- Utilice Amazon Elastic Container Registry como repositorio de contenedores: cuando ejecuta miles de trabajos en paralelo, un repositorio autogestionado puede fallar o reducir el rendimiento.

Amazon ECR funciona a escala y puede gestionar cargas de trabajo con hasta más de un millón de vCPU.



Elija el recurso de entorno de computación adecuado

AWS Fargate requiere menos instalación y configuración iniciales que Amazon EC2 y probablemente sea más fácil de usar, especialmente si es la primera vez. Con Fargate, no necesita administrar servidores, gestionar la planificación de la capacidad ni aislar las cargas de trabajo de contenedores por seguridad.

Si tiene los siguientes requisitos, le recomendamos que utilice las instancias de Fargate:

- Sus trabajos deben comenzar rápidamente, específicamente en menos de 30 segundos.
- Los requisitos de sus trabajos son 16 vCPU o menos, ninguna GPU y 120 GiB de memoria o menos.

Para obtener más información, consulte [Cuándo usar Fargate](#).

Si tiene los siguientes requisitos, le recomendamos que utilice las instancias de Amazon EC2:

- Necesita un mayor control sobre la selección de instancias o necesita utilizar tipos de instancias específicos.

- Sus trabajos requieren recursos que AWS Fargate no puede proporcionar, como GPU, más memoria, una AMI personalizada o el adaptador Amazon Elastic Fabric.
- Necesita un alto nivel de rendimiento o simultaneidad.
- Debe personalizar la AMI, la plantilla de lanzamiento de Amazon EC2 o el acceso a parámetros especiales de Linux.

Con Amazon EC2, puede ajustar con mayor precisión su carga de trabajo a sus requisitos específicos y ejecutar a escala si es necesario.

Amazon EC2 bajo demanda o Amazon EC2 Spot

La mayoría de clientes de AWS Batch utilizan las instancias de spot de Amazon EC2 porque ahorran en comparación con las instancias bajo demanda. Sin embargo, si su carga de trabajo dura varias horas y no puede interrumpirse, las instancias bajo demanda podrían ser más adecuadas para usted. Siempre puede probar primero las instancias de spot y cambiar a las instancias bajo demanda si es necesario.

Si tiene los siguientes requisitos y expectativas, utilice las instancias bajo demanda de Amazon EC2:

- El tiempo de ejecución de sus trabajos es de más de una hora y no puede tolerar las interrupciones de su carga de trabajo.
- Tiene un SLO (objetivo de nivel de servicio) estricto para su carga de trabajo general y no puede aumentar el tiempo computacional.
- Las instancias que necesita tienen más probabilidades de sufrir interrupciones.

Si tiene los siguientes requisitos y expectativas, utilice las instancias de spot de Amazon EC2:

- El tiempo de ejecución de sus trabajos suele ser de 30 minutos o menos.
- Puede tolerar posibles interrupciones y la reprogramación de los trabajos como parte de su carga de trabajo. Para obtener más información, consulte [Asistente de instancia de spot](#).
- Los trabajos de larga duración se pueden reiniciar desde un punto de control si se interrumpen.

Puede combinar ambos modelos de compra enviándolos primero en una instancia de spot y, a continuación, utilizando la instancia bajo demanda como opción alternativa. Por ejemplo, envíe sus trabajos en una cola que esté conectada a entornos de computación que se ejecuten en instancias de spot de Amazon EC2. Si se interrumpe un trabajo, busca el evento en Amazon EventBridge y

correlaciónalo con una recuperación de una instancia de spot. A continuación, vuelva a enviar el trabajo a una cola bajo demanda mediante una función AWS Lambda o AWS Step Functions. Para obtener más información, consulte [Tutorial: envío de alertas de Amazon Simple Notification Service para eventos de trabajos fallidos](#), [Prácticas recomendadas para gestionar las interrupciones de las instancias de spot de Amazon EC2](#) y [Gestionar AWS Batch con Step Functions](#).

 Important

Utilice distintos tipos de instancias, tamaños y zonas de disponibilidad para su entorno de computación bajo demanda a fin de mantener la disponibilidad del grupo de instancias de spot de Amazon EC2 y reducir la tasa de interrupciones.

Prácticas recomendadas para instancias de spot de Amazon EC2 para AWS Batch

Si elige las instancias de spot de Amazon Elastic Compute Cloud (EC2), es probable que pueda optimizar su flujo de trabajo para ahorrar costos, a veces de forma significativa. Para obtener más información, consulte [Prácticas recomendadas de seguridad para Amazon EC2 Spot](#).

Para optimizar su flujo de trabajo y ahorrar costos, tenga en cuenta las siguientes prácticas recomendadas de Amazon EC2 Spot para AWS Batch:

- Elija la estrategia de asignación de **SPOT_CAPACITY_OPTIMIZED**: AWS Batch elige las instancias de Amazon EC2 de los grupos de capacidad de Amazon EC2 Spot más amplios. Si le preocupan las interrupciones, esta es una opción adecuada. Para obtener más información, consulte [Estrategias de asignación de tipos de instancia para AWS Batch](#).
- Diversifique los tipos de instancias: para diversificar sus tipos de instancias, considere tamaños y familias compatibles y, a continuación, deje que AWS Batch elija en función del precio o la disponibilidad. Por ejemplo, considere c5.24xlarge como una alternativa a c5.12xlarge o a las familias c5a, c5n, c5d, m5 y m5d. Para más información, consulte [Ser flexible con respecto a los tipos de instancia y las zonas de disponibilidad](#).
- Reduzca el tiempo de ejecución o el punto de control de los trabajos: recomendamos no ejecutar trabajos que tarden una hora o más cuando se utilizan instancias de spot de Amazon EC2 para evitar interrupciones. Si divide o pone puntos de control a sus trabajos en partes más pequeñas de 30 minutos o menos, puede reducir considerablemente la posibilidad de interrupciones.

- Utilice reintentos automatizados: para evitar interrupciones en los trabajos AWS Batch, configure los reintentos automatizados para los trabajos. Los trabajos por lotes pueden interrumpirse por cualquiera de los siguientes motivos: se devuelve un código de salida distinto de cero, se produce un error de servicio o se produce la recuperación de una instancia. Puede configurar hasta 10 reintentos automáticos. Para empezar, le recomendamos que establezca al menos de 1 a 3 reintentos automatizados. Para obtener información sobre el seguimiento de las interrupciones puntuales de Amazon EC2, consulte [Avisos de interrupción de spot](#).

Para AWS Batch, si establece el parámetro de reintento, el trabajo se coloca al principio de la cola de trabajos. Es decir, se da prioridad al trabajo. Al crear la definición de trabajo o al enviar el trabajo en la AWS CLI, puede configurar una estrategia de reintento. Para obtener más información, consulte [Enviar un trabajo](#).

```
$ aws batch submit-job --job-name MyJob \  
  --job-queue MyJQ \  
  --job-definition MyJD \  
  --retry-strategy attempts=2
```

- Utilice reintentos personalizados: puede configurar una estrategia de reintento de trabajo para el código de salida de una aplicación específica o la recuperación de instancias. En el siguiente ejemplo, si el host provoca el error, se puede volver a intentar el trabajo hasta cinco veces. Sin embargo, si el trabajo falla por un motivo diferente, el trabajo se cierra y el estado se establece en FAILED.

```
"retryStrategy": {  
  "attempts": 5,  
  "evaluateOnExit":  
    [{  
      "onStatusReason" : "Host EC2*",  
      "action": "RETRY"  
    }, {  
      "onReason" : "*",  
      "action": "EXIT"  
    }]  
}
```

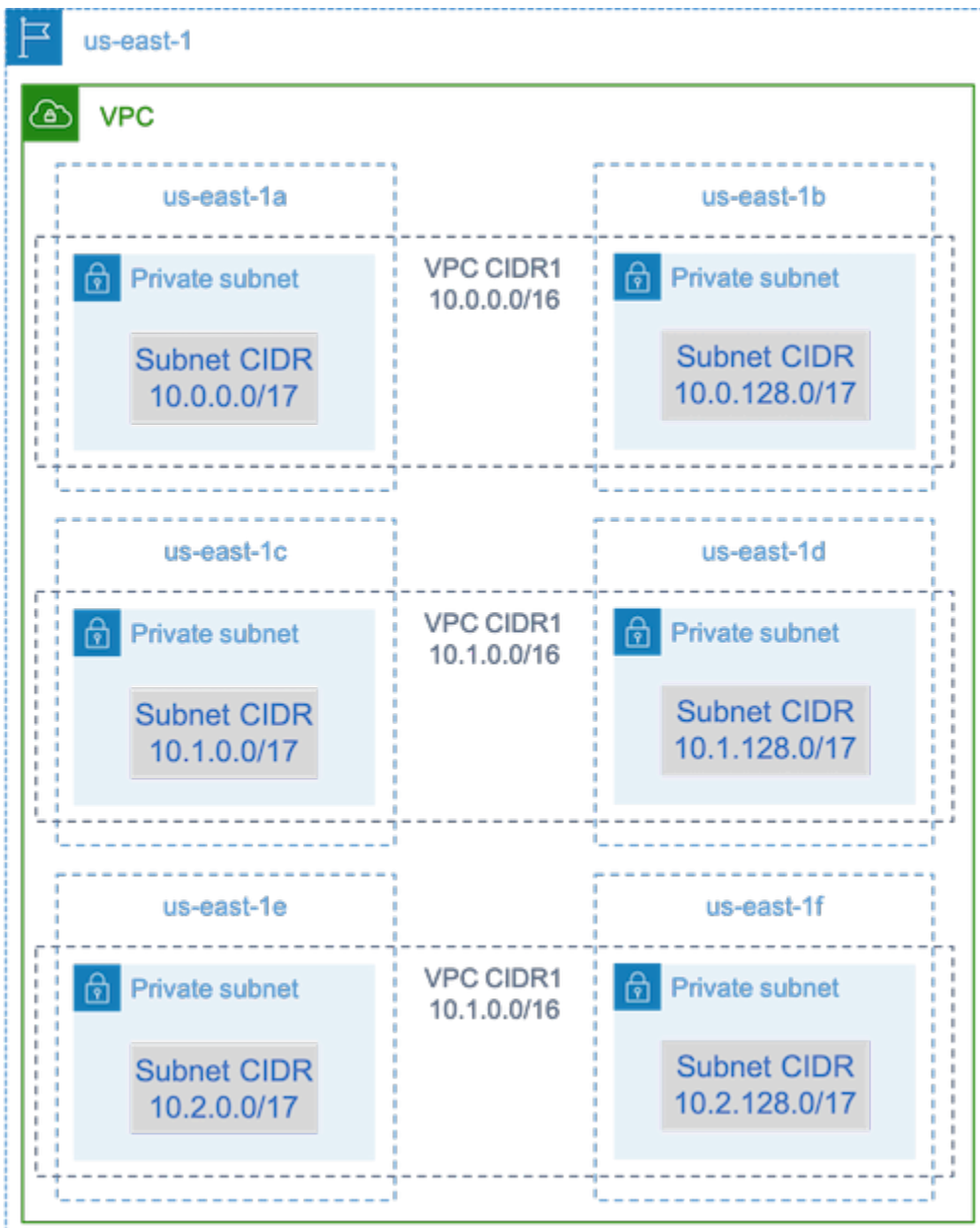
- Utilice el panel de control de interrupciones puntuales: puede utilizar el panel de control de interrupciones puntuales para realizar un seguimiento de las interrupciones puntuales. La aplicación proporciona métricas sobre las instancias de spot de Amazon EC2 que se recuperan

y las zonas de disponibilidad en las que se encuentran las instancias de spot. Para obtener más información, consulte [Interrupciones de instancias de spot](#).

Errores comunes y solución de problemas

Los errores en AWS Batch suelen producirse en el nivel de la aplicación o se deben a configuraciones de instancias que no cumplen los requisitos específicos de su trabajo. Otros problemas incluyen que los trabajos se atasquen en un estado `RUNNABLE` o que los entornos de computación se queden atascados en un estado `INVALID`. Para obtener más información sobre la solución de problemas de los trabajos que se quedan atascados en el estado `RUNNABLE`, consulte [Trabajos bloqueados en estado RUNNABLE](#). Para obtener información sobre cómo solucionar problemas de entornos de computación en un estado `INVALID`, consulte [Entorno de computación de INVALID](#).

- Compruebe las cuotas de vCPU puntuales de Amazon EC2: compruebe que sus cuotas de servicio actuales cumplen los requisitos del trabajo. Por ejemplo, supongamos que su cuota de servicio actual es de 256 vCPU y que el trabajo requiere 10 000 vCPU. Entonces, la cuota de servicio no cumple con el requisito del trabajo. Para obtener más información e instrucciones de solución de problemas, consulte [Service quotas de Amazon EC2](#) y [¿Cómo puedo aumentar la cuota de servicio de mis recursos de Amazon EC2?](#).
- Los trabajos fallan antes de que se ejecute la aplicación: algunos trabajos pueden fallar debido a un error `DockerTimeoutError` o a un error `CannotPullContainerError`. Para obtener información sobre la solución de problemas, consulte [¿Cómo se resuelve el error “DockerTimeoutError” en AWS Batch?](#).
- Direcciones IP insuficientes: la cantidad de direcciones IP de la VPC y las subredes puede limitar la cantidad de instancias que se pueden crear. Utilice enrutamiento entre dominios sin clases (CIDR) para proporcionar más direcciones IP de las necesarias para ejecutar sus cargas de trabajo. Si es necesario, también puede crear una VPC dedicada con un gran espacio de direcciones. Por ejemplo, puede crear una VPC con varios CIDR en `10.x.0.0/16` y una subred en cada zona de disponibilidad con un CIDR de `10.x.y.0/17`. En este ejemplo, x está entre 1 y 4 e y es 0 o 128. Esta configuración proporciona 36 000 direcciones IP en cada subred.



- Compruebe que las instancias estén registradas en Amazon EC2: si ve sus instancias en la consola de Amazon EC2, pero no ve ninguna instancia de contenedor de Amazon Elastic Container Service en su clúster de Amazon ECS, es posible que el agente de Amazon ECS no esté instalado en una Imagen de máquina de Amazon (AMI). Es posible que el agente de Amazon ECS, los datos de Amazon EC2 de su AMI o la plantilla de lanzamiento tampoco estén configurados correctamente. Para aislar la causa raíz, cree una instancia Amazon EC2 independiente o conéctese a una instancia existente mediante SSH. Para obtener más información, consulte [Configuración del agente contenedor de Amazon ECS](#), [Ubicaciones de los archivos de registro de Amazon ECS](#) y [AMI de recursos de computación](#).

- Revise el panel de AWS: revise el panel de AWS para comprobar el estado del trabajo esperado y que el entorno de computación se escala según lo esperado. También puede revisar los registros de trabajos en CloudWatch.
- Compruebe que la instancia esté creada: si se crea una instancia, significa que su entorno de computación se ha escalado según lo previsto. Si sus instancias no se crean, busque las subredes asociadas en su entorno de computación para cambiarlas. Para obtener más información, consulte [Verificar una actividad de escalado para un grupo de escalado automático](#).

También le recomendamos que verifique que sus instancias pueden cumplir con los requisitos de trabajo relacionados. Por ejemplo, un trabajo puede requerir 1 TiB de memoria, pero el entorno de computación usa un tipo de instancia C5 que está limitado a 192 GB de memoria.

- Verifique que sus instancias estén siendo solicitadas por AWS Batch: consulte el historial del grupo de escalado automático para verificar que sus instancias estén siendo solicitadas por AWS Batch. Esto indica cómo Amazon EC2 intenta adquirir instancias. Si recibe un error que indica que Amazon EC2 Spot no puede adquirir una instancia en una zona de disponibilidad específica, es posible que la zona de disponibilidad no ofrezca una familia de instancias específica.
- Compruebe que las instancias se registren en Amazon ECS: si ve instancias en la consola de Amazon EC2, pero no hay instancias de contenedor de Amazon ECS en su clúster de Amazon ECS, es posible que el agente de Amazon ECS no esté instalado en la Imagen de máquina de Amazon (AMI). Además, es posible que el agente de Amazon ECS, los datos de Amazon EC2 de su AMI o la plantilla de lanzamiento tampoco estén configurados correctamente. Para aislar la causa raíz, cree una instancia Amazon EC2 independiente o conéctese a una instancia existente mediante SSH. Para obtener más información, consulte [Archivo de configuración del agente de CloudWatch: sección de registros](#), [Ubicaciones de archivos de registro de Amazon ECS](#) y [AMI de recursos de computación](#).
- Abra un ticket de soporte: si aún tiene problemas después de solucionar algunos problemas y tiene un plan de soporte, abra un ticket de soporte. En el ticket de soporte, asegúrese de incluir información sobre el problema, las especificaciones de la carga de trabajo, la configuración y los resultados de las pruebas. Para obtener más información, consulte [Comparar Soporte planes](#).
- Revise los foros de AWS Batch y HPC: para obtener más información, consulte los foros de [AWS Batch](#) y [HPC](#).
- Revise el panel de monitoreo del tiempo de ejecución de AWS Batch: este panel utiliza una arquitectura sin servidor para capturar eventos de Amazon ECS AWS Batch, y Amazon EC2 a fin de proporcionar información sobre los trabajos y las instancias. Para obtener más información, consulte [Solución de paneles de monitoreo de tiempo de ejecución de AWS Batch](#).

Solución de problemas AWS Batch

Es posible que necesite solucionar problemas relacionados con sus entornos de computación, colas de trabajos, definiciones de trabajos o trabajos. En este capítulo se describe cómo solucionar y resolver estos problemas en su entorno. AWS Batch

AWS Batch utiliza políticas, funciones y permisos de IAM y se ejecuta en la infraestructura de Amazon EC2, Amazon ECS y Amazon AWS Fargate Elastic Kubernetes Service. Para solucionar problemas relacionados con estos servicios, consulte lo siguiente:

- [Solución de problemas de IAM](#) en la Guía del usuario de IAM
- [Solución de problemas de Amazon ECS](#) en la Guía para desarrolladores de Amazon Elastic Container Service
- [Solución de problemas de Amazon EKS](#) en la Guía del usuario de Amazon EKS
- [Solución de problemas para instancias de EC2](#) en la Guía del usuario de Amazon EC2

Contenido

- [AWS Batch](#)
 - [Configuración óptima del tipo de instancia para recibir actualizaciones automáticas de la familia de instancias](#)
 - [Entorno de computación de INVALID](#)
 - [Nombre de rol o ARN incorrectos](#)
 - [Reparación de un entorno de computación de INVALID](#)
 - [Trabajos bloqueados en estado RUNNABLE](#)
 - [Instancias de spot no etiquetadas en el momento de su creación](#)
 - [Las instancias de spot no se están reduciendo verticalmente](#)
 - [Adjunta la política EC2 SpotFleetTaggingRole gestionada por Amazon a tu función de Spot Fleet en el Consola de administración de AWS](#)
 - [Adjunta la política EC2 SpotFleetTaggingRole gestionada por Amazon a tu puesto de Spot Fleet con el AWS CLI](#)
 - [No puedo recuperar los secretos de Secrets Manager](#)
 - [No se pueden anular los requisitos de recursos de la definición del trabajo](#)
 - [Mensaje de error al actualizar la configuración de desiredvCpus](#)

- [AWS Batch en Amazon EKS](#)
 - [Entorno de computación de INVALID](#)
 - [La versión del Kubernetes no es compatible](#)
 - [El perfil de instancia no existe](#)
 - [Espacio de nombres de Kubernetes no válido](#)
 - [Entorno de computación eliminado](#)
 - [Los nodos no se unen al clúster de Amazon EKS](#)
 - [AWS Batch en Amazon EKS, el trabajo está RUNNABLE estancado](#)
 - [AWS Batch en Amazon EKS, el trabajo está STARTING estancado](#)
 - [Situación: Error al adjuntar o montar una recuperación de volumen persistente](#)
 - [Compruebe que aws-auth ConfigMap se ha configurado correctamente](#)
 - [Los permisos o enlaces de RBAC no están configurados correctamente](#)

AWS Batch

Consulte los siguientes temas para ver procesos de revisión y posibles soluciones a los problemas comunes que pueden surgir al usar AWS Batch.

Temas

- [Configuración óptima del tipo de instancia para recibir actualizaciones automáticas de la familia de instancias](#)
- [Entorno de computación de INVALID](#)
- [Trabajos bloqueados en estado RUNNABLE](#)
- [Instancias de spot no etiquetadas en el momento de su creación](#)
- [Las instancias de spot no se están reduciendo verticalmente](#)
- [No puedo recuperar los secretos de Secrets Manager](#)
- [No se pueden anular los requisitos de recursos de la definición del trabajo](#)
- [Mensaje de error al actualizar la configuración de desiredvCpus](#)

Configuración óptima del tipo de instancia para recibir actualizaciones automáticas de la familia de instancias

Note

Desde el 1 de noviembre de 2025, el comportamiento de `optimal` se cambiará para que coincida con `default_x86_64`. Las familias de instancias podrían actualizarse a una generación más nueva durante el proceso del cambio. No es necesario realizar ninguna acción para realizar la actualización.

AWS Batch admitía una sola opción en `InstanceTypes` `optimal` para adaptarse a la demanda de sus colas de trabajos. Integramos dos nuevas opciones de tipos de instancias: `default_x86_64` y `default_arm64`. Si no se selecciona el tipo de instancia, se utilizará `default_x86_64`. Estas nuevas opciones seleccionarán automáticamente los tipos de instancias rentables para diferentes familias y generaciones según los requisitos de la cola de trabajo, lo que permitirá ejecutar rápidamente las cargas de trabajo.

A medida que haya una capacidad suficiente de nuevos tipos de instancias disponibles en una Región de AWS, el grupo predeterminado correspondiente se actualizará automáticamente con el nuevo tipo de instancia. La opción `optimal` existente seguirá siendo compatible y no quedará obsoleta, ya que será compatible con los grupos predeterminados subyacentes para luego brindar instancias actualizadas. Si está utilizando `optimal`, no tiene que hacer nada.

Sin embargo, ten en cuenta que solo `ENABLED` los entornos de `VALID` cómputo (CEs) se actualizarán automáticamente con los nuevos tipos de instancias. Si tienes `DISABLED` alguna `INVALID` CEs, recibirá las actualizaciones una vez que se vuelva a habilitar y se establezca en un `VALID` estado.

Entorno de computación de **INVALID**

Es posible que haya configurado incorrectamente un entorno de computación gestionado. Si lo ha hecho, el entorno de computación entra en un estado `INVALID` y no puede aceptar trabajos para ubicarlos. En las siguientes secciones se describen las posibles causas y cómo solucionar los problemas en función de la causa.

 Important

AWS Batch crea y administra varios AWS recursos en su nombre y dentro de su cuenta, incluidas las plantillas de lanzamiento de Amazon EC2, los grupos de Auto Scaling de Amazon EC2, las flotas puntuales de Amazon EC2 y los clústeres de Amazon ECS. Estos recursos administrados están configurados específicamente para garantizar un funcionamiento óptimo de AWS Batch . La modificación manual de estos recursos administrados por lotes, a menos que se indique explícitamente en la documentación de AWS Batch , puede provocar un comportamiento inesperado dando lugar a un entorno de computación INVALID, a un comportamiento de escalado de instancias subóptimo, a un retraso en el procesamiento de la carga de trabajo o a costos inesperados. El servicio no puede admitir estas modificaciones manuales de forma determinista. AWS Batch Utilice siempre el Batch compatible APIs o la consola Batch para gestionar sus entornos informáticos.

Nombre de rol o ARN incorrectos

La causa más común por la que un entorno informático entra en un INVALID estado es que el rol de AWS Batch servicio o el rol de Amazon EC2 Spot Fleet tienen un nombre o un nombre de recurso de Amazon (ARN) incorrectos. Esto es más común en los entornos informáticos que se crean con AWS CLI o. AWS SDKs Cuando crea un entorno informático en Consola de administración de AWS, le AWS Batch ayuda a elegir el servicio correcto o las funciones de Spot Fleet. Sin embargo, supongamos que introduce manualmente el nombre o el ARN y los introduce de forma incorrecta. Luego, el entorno de computación resultante también es INVALID.

Sin embargo, supongamos que introduce manualmente el nombre o el ARN de un recurso de IAM en un comando de AWS CLI o en el código del SDK. En este caso, no AWS Batch se puede validar la cadena. En su lugar, AWS Batch debe aceptar el valor incorrecto e intentar crear el entorno. Si AWS Batch no se crea el entorno, éste pasa a un INVALID estado y aparecen los siguientes errores.

Si se trata de un rol de servicio inválido:

```
CLIENT_ERROR - Not authorized to perform sts:AssumeRole (Service:
AWSSecurityTokenService; Status Code: 403; Error Code: AccessDenied;
Request ID: dc0e2d28-2e99-11e7-b372-7fcc6fb65fe7)
```

Si se trata de un rol de flota de spot no válido:

```
CLIENT_ERROR - Parameter: SpotFleetRequestConfig.IamFleetRole
is invalid. (Service: AmazonEC2; Status Code: 400; Error Code:
InvalidSpotFleetRequestConfig; Request ID: 331205f0-5ae3-4cea-
bac4-897769639f8d) Parameter: SpotFleetRequestConfig.IamFleetRole is
invalid
```

Una causa común de este problema es el siguiente escenario. Solo se especifica el nombre de una función de IAM cuando se utiliza el AWS CLI o el AWS SDKs, en lugar del nombre completo del recurso de Amazon (ARN). En función de cómo se haya creado el rol de servicio, el ARN puede contener un prefijo de ruta `aws-service-role`. Por ejemplo, si crea manualmente el rol de servicio de AWS Batch utilizando los procedimientos de [Utilizar funciones vinculadas al servicio para AWS Batch](#), el ARN del rol de servicio debería ser similar al siguiente.

```
arn:aws:iam::123456789012:role/AWSBatchServiceRole
```

Sin embargo, si creó el rol de servicio como parte del asistente de primera ejecución de la consola hoy, el ARN de su rol de servicio podría tener el siguiente aspecto.

```
arn:aws:iam::123456789012:role/aws-service-role/AWSBatchServiceRole
```

Este problema también puede producirse si adjuntas la política de AWS Batch nivel de servicio (AWSBatchServiceRole) a una función que no sea de servicio. Por ejemplo, es posible que reciba un mensaje de error similar al siguiente en este escenario:

```
CLIENT_ERROR - User: arn:aws:sts::account_number:assumed-role/batch-replacement-role/
aws-batch is not
authorized to perform: action on resource ...
```

Para resolver este problema, siga uno de estos pasos.

- Utilice una cadena vacía para la función de servicio al crear el AWS Batch entorno de procesamiento.
- Especifique el rol de servicio en el siguiente formato: `arn:aws:iam::account_number:role/aws-service-role/batch.amazonaws.com/AWSServiceRoleForBatch`.

Si solo especificas el nombre de un rol de IAM al usar el AWS CLI o el AWS SDKs, se AWS Batch asume que tu ARN no usa `aws-service-role` el prefijo de ruta. Por eso se recomienda especificar el ARN completo para los roles de IAM al crear entornos de computación.

Para reparar un entorno de computación configurado de forma incorrecta, consulte [Reparación de un entorno de computación de INVALID](#).

Reparación de un entorno de computación de **INVALID**

Si el entorno de computación se encuentra en estado INVALID, actualícelo para reparar el parámetro no válido. Para un [Nombre de rol o ARN incorrectos](#), actualice el entorno de computación con el rol de servicio correcto.

Para reparar un entorno de computación mal configurado

1. Abra la consola en AWS Batch . <https://console.aws.amazon.com/batch/>
2. En la barra de navegación, seleccione la Región de AWS que desee utilizar.
3. En el panel de navegación, elija Entornos de computación.
4. En la página Entornos de computación, marque el botón de opción que se encuentre junto al entorno de computación a editar y, a continuación, seleccione Editar.
5. En la página Actualizar entornos de computación, en Rol de servicio, elija el rol de IAM a utilizar con su entorno de computación. La consola AWS Batch solo muestra roles con la relación de confianza adecuada para entornos de computación.

Tip

Para obtener instrucciones sobre cómo crear un rol vinculado al servicio, consulte [Uso de roles para AWS Batch](#).

6. Seleccione Guardar para actualizar el entorno de computación.

Trabajos bloqueados en estado **RUNNABLE**

Suponga que su entorno de computación contiene recursos de computación, pero sus trabajos no progresan más allá del estado RUNNABLE. Seguramente, algo impida que los trabajos se coloquen en un recurso informático y se bloqueen las colas de trabajos. A continuación, se explica cómo saber si un trabajo está esperando su turno o se ha atascado y bloqueado la cola.

Si AWS Batch detecta que tienes un RUNNABLE trabajo a la cabeza y bloquea la cola, recibirás un [Eventos bloqueados en la cola de trabajos](#) evento de Amazon CloudWatch Events con el motivo. El mismo motivo también se actualiza en el campo statusReason como parte de las llamadas a las API [ListJobs](#) y [DescribeJobs](#).

Si lo desea, puede configurar el parámetro `jobStateTimeLimitActions` mediante [CreateJobQueue](#) y las acciones de la API [UpdateJobQueue](#).

 Note

Actualmente, en el caso de las colas de trabajos conectadas a entornos informáticos de Amazon ECS, Amazon EKS o Fargate, la única acción que puede realizar `jobStateLimitActions.action` es cancelar un trabajo.

El `jobStateTimeLimitActions` parámetro se usa para especificar un conjunto de acciones que se AWS Batch realizan en trabajos en un estado específico. Puede establecer un umbral de tiempo en segundos en el campo `maxTimeSeconds`.

Cuando un trabajo ha estado en un `RUNNABLE` estado con el `definidoStatusReason`, AWS Batch realiza la acción especificada una `maxTimeSeconds` vez transcurrido.

Por ejemplo, puede configurar el parámetro `jobStateTimeLimitActions` para que espere hasta 4 horas en cualquier trabajo con el estado `RUNNABLE` que esté esperando a que haya suficiente capacidad disponible. `statusReason` Para ello, establezca el valor entre `CAPACITY:INSUFFICIENT_INSTANCE_CAPACITY` y `maxTimeSeconds 14400` antes de cancelar el trabajo y permitir que el siguiente pase a ocupar el primer lugar de la lista de trabajos.

Los siguientes son los motivos que se utilizan AWS Batch cuando detecta que una cola de trabajos está bloqueada. Esta lista proporciona los mensajes devueltos por las acciones de `DescribeJobs` y las acciones de API `ListJobs`. También son los mismos valores que puede definir para el parámetro `jobStateLimitActions.statusReason`.

1. Motivo: todos los entornos de computación conectados tienen errores de capacidad insuficiente. Cuando se solicita, AWS Batch detecta las instancias de Amazon EC2 que experimentan errores de capacidad insuficiente. La cancelación manual del trabajo permitirá que el trabajo siguiente pase al principio de la cola.
 - Mensaje de **statusReason** mientras el trabajo está atascado:
`CAPACITY:INSUFFICIENT_INSTANCE_CAPACITY - Service cannot fulfill the capacity requested for instance type [instanceTypeName]`
 - **reason** utilizado para **jobStateTimeLimitActions**:
`CAPACITY:INSUFFICIENT_INSTANCE_CAPACITY`

- Mensaje de **statusReason** después de cancelar el trabajo:
 Canceled by JobStateTimeLimit action due to reason:
 CAPACITY:INSUFFICIENT_INSTANCE_CAPACITY

Nota:

- a. El rol AWS Batch de servicio requiere `autoscaling:DescribeScalingActivities` permiso para que esta detección funcione. Si utiliza el rol vinculado al servicio (SLR) [Utilizar funciones vinculadas al servicio para AWS Batch](#) o la política administrada [AWS política gestionada: AWSBatchServiceRolepolítica](#), no tiene que realizar ninguna acción porque sus políticas de permisos están actualizadas.
 - b. Si utiliza el SLR o la política gestionada, debe añadir los permisos `autoscaling:DescribeScalingActivities` y `ec2:DescribeSpotFleetRequestHistory` para poder recibir los eventos de las colas de trabajos bloqueadas y el estado actualizado de los trabajos cuando esté en `RUNNABLE`. Además, AWS Batch necesita estos permisos para realizar acciones de `cancellation` a través del parámetro `jobStateTimeLimitActions`, incluso si se han configurado en la cola de trabajos.
 - c. En el caso de un trabajo paralelo de varios nodos (MNP), si el entorno de computación de Amazon EC2 de alta prioridad adjunto experimenta errores `insufficient capacity`, bloquea la cola aunque otro entorno de computación de prioridad inferior sufra este error.
2. Motivo: todos los entornos de computación tienen un parámetro [maxvCpus](#) inferior a los requisitos del trabajo. Si se cancela el trabajo, ya sea manualmente o configurando el parámetro `jobStateTimeLimitActions` en `statusReason`, se permite que el trabajo siguiente pase al primer lugar en la cola. Si lo desea, puede aumentar el parámetro `maxvCpus` del entorno de computación principal para satisfacer las necesidades del trabajo bloqueado.
 - Mensaje de **statusReason** mientras el trabajo está atascado:
 MISCONFIGURATION:COMPUTE_ENVIRONMENT_MAX_RESOURCE - CE(s) associated with the job queue cannot meet the CPU requirement of the job.
 - **reason** utilizado para **jobStateTimeLimitActions**:
 MISCONFIGURATION:COMPUTE_ENVIRONMENT_MAX_RESOURCE
 - Mensaje de **statusReason** después de cancelar el trabajo:
 Canceled by JobStateTimeLimit action due to reason:
 MISCONFIGURATION:COMPUTE_ENVIRONMENT_MAX_RESOURCE
 3. Motivo: ninguno de los entornos de procesamiento tiene instancias que cumplan con los requisitos del trabajo. Cuando un trabajo solicita recursos, AWS Batch detecta que ningún

entorno informático conectado puede alojar el trabajo entrante. Si se cancela el trabajo, ya sea manualmente o configurando el parámetro `jobStateTimeLimitActions` en `statusReason`, se permite que el trabajo siguiente pase al primer lugar en la cola. Si lo desea, puede redefinir los tipos de instancias permitidos en el entorno de computación para añadir los recursos de trabajo necesarios.

- Mensaje de **statusReason** mientras el trabajo está atascado:
`MISCONFIGURATION:JOB_RESOURCE_REQUIREMENT - The job resource requirement (vCPU/memory/GPU) is higher than that can be met by the CE(s) attached to the job queue.`
- **reason** utilizado para **jobStateTimeLimitActions**:
`MISCONFIGURATION:JOB_RESOURCE_REQUIREMENT`
- Mensaje de **statusReason** después de cancelar el trabajo:
`Canceled by JobStateTimeLimit action due to reason:`
`MISCONFIGURATION:JOB_RESOURCE_REQUIREMENT`

4. Motivo: todos los entornos de computación tienen problemas con los roles de servicio. Para resolver este problema, compare los permisos de su rol de servicio con [AWS políticas gestionadas para AWS Batch](#) y solucione las brechas. Nota: No se puede configurar una acción programable a través del parámetro `jobStateTimeLimitActions` para resolver este error.

Se recomienda utilizar [Utilizar funciones vinculadas al servicio para AWS Batch](#) para evitar errores similares.

Si se cancela el trabajo, ya sea manualmente o configurando el parámetro `jobStateTimeLimitActions` en `statusReason`, se permite que el trabajo siguiente pase al primer lugar en la cola. Si no se resuelven los problemas relacionados con los roles de servicio, es probable que también se bloquee el siguiente trabajo. Lo mejor es investigar y resolver este problema manualmente.

- Mensaje de **statusReason** mientras el trabajo está atascado:
`MISCONFIGURATION:SERVICE_ROLE_PERMISSIONS - Batch service role has a permission issue.`
5. Motivo: Su entorno de computación tiene una configuración que no es compatible con ese tipo de instancia. Esto puede ocurrir cuando los tipos de instancias no están disponibles en las zonas de disponibilidad seleccionadas o cuando la plantilla o configuración de lanzamiento contiene ajustes incompatibles con los tipos de instancias especificados. Para resolver este problema, compruebe que los tipos de instancia sean compatibles con las zonas de disponibilidad Región de AWS y especificadas, compruebe que la configuración de la plantilla de lanzamiento sea compatible con

los tipos de instancias y considere la posibilidad de actualizarlos a tipos de instancias de nueva generación. Para obtener más información sobre la búsqueda de tipos de instancias compatibles, consulte [Búsqueda de un tipo de instancia de Amazon EC2](#) en la Guía del usuario de Amazon EC2.

- Mensaje de **statusReason** mientras el trabajo está atascado:
MISCONFIGURATION:EC2_INSTANCE_CONFIGURATION_UNSUPPORTED - Your compute environment associated with this job queue has an unsupported instance type configuration.
6. Motivo: todos los entornos de computación no son válidos. Para obtener más información, consulte [Entorno de computación de INVALID](#). Nota: No se puede configurar una acción programable a través del parámetro `jobStateTimeLimitActions` para resolver este error.
- Mensaje de **statusReason** mientras el trabajo está atascado: ACTION_REQUIRED - CE(s) associated with the job queue are invalid.
7. Motivo: AWS Batch ha detectado una cola bloqueada, pero no ha podido determinar el motivo. Nota: No se puede configurar una acción programable a través del parámetro `jobStateTimeLimitActions` para resolver este error. Para obtener más información sobre la solución de problemas, consulte [¿Por qué mi trabajo de AWS Batch está atascado como RUNNABLE en AWS?](#) en re:Post.
- Mensaje de **statusReason** mientras el trabajo está atascado: UNDETERMINED - Batch job is blocked, root cause is undetermined.

En caso de que no hayas recibido un evento de CloudWatch Events o hayas recibido un evento con un motivo desconocido, estas son algunas de las causas más comunes de este problema.

El controlador de registros **awslogs** no está configurado en los recursos de computación

AWS Batch los trabajos envían su información de registro a CloudWatch Logs. Para activarlo, debe configurar sus recursos de computación para utilizar el controlador de registro `awslogs`. Supongamos que basa la AMI de sus recursos de computación en la AMI optimizada para Amazon ECS (o Amazon Linux). A continuación, este controlador se registra de forma predeterminada en el paquete `ecs-init`. Ahora supongamos que usa una AMI base diferente. Luego, debe verificar que el controlador de registro `awslogs` esté especificado como un controlador de registro disponible con la variable de entorno `ECS_AVAILABLE_LOGGING_DRIVERS` cuando se inicia el agente de contenedor de Amazon ECS. Para obtener más información, consulte [Especificaciones de AMI de recursos de computación](#) y [Tutorial: creación de una AMI de recursos de computación](#).

Recursos insuficientes

Si sus definiciones de trabajo especifican más recursos de memoria o CPU de lo que pueden asignar los recursos de computación, los trabajos no se asignarán. Por ejemplo, supongamos que su trabajo especifica 4 GiB de memoria y que sus recursos de computación tienen menos de los disponibles. Entonces, se da el caso de que el trabajo no se puede asignar a esos recursos de computación. En ese caso, debe reducir la memoria especificada en la definición del trabajo o añadir más recursos de computación en su entorno. Una parte de la memoria se reserva para el agente de contenedor de Amazon ECS y otros procesos críticos del sistema. Para obtener más información, consulte [Administración de la memoria de recurso de computación](#).

No hay acceso a Internet para los recursos de computación

Los recursos de computación de las deben obtener acceso para comunicarse con el punto de conexión del servicio de Amazon ECS. Esto puede ser a través de un punto de conexión de VPC de la interfaz o a través de recursos de computación de las con direcciones IP públicas.

Para obtener más información acerca de los puntos de enlace de la VPC de la interfaz, consulte [Puntos de enlace de la VPC de la interfaz de Amazon ECS \(AWS PrivateLink\)](#) en la Guía para desarrolladores de Amazon Elastic Container Service.

Si no tiene configurado un punto de conexión de la VPC de la interfaz y los recursos de computación de las no tienen direcciones IP públicas, deberán utilizar traducción de direcciones de red (NAT) para proporcionar este acceso. Para obtener más información, consulte [Gateways NAT](#) en la Guía del usuario de Amazon VPC. Para obtener más información, consulte [the section called “Creación de una VPC”](#).

Se ha alcanzado el límite de instancias de Amazon EC2

La cantidad de instancias de Amazon EC2 en las que puede lanzar su cuenta Región de AWS viene determinada por su cuota de instancias EC2. Algunos tipos de instancias también tienen una per-instance-type cuota. Para obtener más información sobre la cuota de instancias de Amazon EC2 de su cuenta, incluida la forma de solicitar un aumento del límite, consulte los [Límites de servicio de Amazon EC2](#) en la Guía del usuario de Amazon EC2.

El agente de contenedor de Amazon ECS no está instalado

El agente contenedor Amazon ECS debe estar instalado en la Imagen de máquina de Amazon (AMI) para permitir a AWS Batch ejecutar los trabajos. El agente de contenedores Amazon ECS se instala de forma predeterminada en Amazon ECS Optimized AMIs. Para obtener más información sobre el agente de contenedor de Amazon ECS, consulte [Agente de contenedor de Amazon ECS](#) en la Guía para desarrolladores de Amazon Elastic Container Service.

Para obtener más información, consulte [¿Por qué mi AWS Batch trabajo está RUNNABLE estancado?](#) en Re:post.

Instancias de spot no etiquetadas en el momento de su creación

El etiquetado de instancias de spot para los recursos de computación de AWS Batch se admite desde el 25 de octubre de 2017. Antes de esa fecha, la política administrada por IAM recomendada (AmazonEC2SpotFleetRole) para el rol de flota de spot de Amazon EC2 no contenía permisos para etiquetar las instancias de spot en el momento del lanzamiento. La nueva política administrada de IAM recomendada se denomina AmazonEC2SpotFleetTaggingRole. Admite el etiquetado de instancias de spot en el momento del lanzamiento.

Para corregir el etiquetado de las instancias de spot al crearlas, siga el siguiente procedimiento para aplicar la política gestionada de IAM actualmente recomendada a su rol de flota de spot de Amazon EC2. De esta forma, todas las instancias de spot futuras que se creen con ese rol tendrán permisos para aplicar etiquetas de instancia cuando se creen.

Para aplicar la política administrada de IAM actual al rol de flota de spot de Amazon EC2

1. Abra la consola de IAM en <https://console.aws.amazon.com/iam/>.
2. Elija Roles y elija el rol de flota de spot de Amazon EC2.
3. Elija Asociar política.
4. Seleccione AmazonEC2SpotFleetTaggingRole y elija Asociar política.
5. Elija de nuevo el rol de flota de spot de Amazon EC2 para quitar la política anterior.
6. Seleccione la x que hay a la derecha de la política AmazonEC2SpotFleetRole y elija Separar.

Las instancias de spot no se están reduciendo verticalmente

AWS Batch introdujo la función AWSServiceRoleForBatch vinculada al servicio el 10 de marzo de 2021. Si no se especifica ningún rol en el parámetro del entorno de computación de `serviceRole`, este rol vinculado al servicio se usa como rol de servicio. Sin embargo, supongamos que el rol vinculado al servicio se usa en un entorno de cómputo de EC2 Spot, pero el rol Spot utilizado no incluye la política gestionada por Amazon EC2 SpotFleetTaggingRole. Entonces, la instancia de spot no se reduce verticalmente. Como resultado, recibirá un error con el siguiente mensaje: “No está autorizado a realizar esta operación”. Siga estos pasos para actualizar la función de flota de spot que utiliza en el parámetro `spotIamFleetRole`. Para obtener más información, consulte [Uso de roles](#)

[vinculados a servicios](#) y [Creación de un rol para delegar permisos a un AWS servicio](#) en la Guía del usuario de IAM.

Temas

- [Adjunta la política EC2 SpotFleetTaggingRole gestionada por Amazon a tu función de Spot Fleet en el Consola de administración de AWS](#)
- [Adjunta la política EC2 SpotFleetTaggingRole gestionada por Amazon a tu puesto de Spot Fleet con el AWS CLI](#)

Adjunta la política EC2 SpotFleetTaggingRole gestionada por Amazon a tu función de Spot Fleet en el Consola de administración de AWS

Para aplicar la política administrada de IAM actual al rol de flota de spot de Amazon EC2

1. Abra la consola de IAM en <https://console.aws.amazon.com/iam/>.
2. Elija Roles y elija el rol de flota de spot de Amazon EC2.
3. Elija Asociar política.
4. Selecciona Amazon EC2 SpotFleetTaggingRole y elige Adjuntar política.
5. Elija de nuevo el rol de flota de spot de Amazon EC2 para quitar la política anterior.
6. Selecciona la x situada a la derecha de la EC2 SpotFleetRole política de Amazon y selecciona Separar.

Adjunta la política EC2 SpotFleetTaggingRole gestionada por Amazon a tu puesto de Spot Fleet con el AWS CLI

Los comandos de ejemplo asumen que su rol de Amazon EC2 Spot Fleet tiene un nombre.

AmazonEC2SpotFleetRole Si su rol usa un nombre diferente, ajuste los comandos para que coincidan.

Para adjuntar la política EC2 SpotFleetTaggingRole gestionada por Amazon a su función de Spot Fleet

1. Para adjuntar la política de IAM EC2 SpotFleetTaggingRole gestionada por Amazon a tu *AmazonEC2SpotFleetRole* rol, ejecuta el siguiente comando mediante AWS CLI

```
$ aws iam attach-role-policy \
```

```
--policy-arn arn:aws:iam::aws:policy/service-role/AmazonEC2SpotFleetTaggingRole \
--role-name AmazonEC2SpotFleetRole
```

2. Para separar la política de IAM EC2 SpotFleetRole gestionada por Amazon de tu *AmazonEC2SpotFleetRole* función, ejecuta el siguiente comando mediante AWS CLI

```
$ aws iam detach-role-policy \
--policy-arn arn:aws:iam::aws:policy/service-role/AmazonEC2SpotFleetRole \
--role-name AmazonEC2SpotFleetRole
```

No puedo recuperar los secretos de Secrets Manager

Si utiliza una AMI con un agente de Amazon ECS anterior a la versión 1.16.0-1, debe utilizar la variable de configuración del agente de Amazon ECS `ECS_ENABLE_AWSLOGS_EXECUTIONROLE_OVERRIDE=true` para utilizar esta característica. Puede agregarlo al archivo `./etc/ecs/ecs.config` a una nueva instancia de contenedor al crear esa instancia. O bien, puede añadirlo a una instancia existente. Si lo agrega a una instancia existente, debe reiniciar el agente de ECS después de agregarlo. Para obtener más información, consulte [Configuración del agente de contenedor de Amazon ECS](#) en la Guía para desarrolladores de Amazon Elastic Container Service.

No se pueden anular los requisitos de recursos de la definición del trabajo

[Las anulaciones de memoria y vCPU que se especifican en `vcpus` los miembros `memory` y de la estructura `ContainerOverrides`, que se pasan a `SubmitJob`, no pueden anular los requisitos de memoria y vCPU que se especifican en la estructura `ResourceRequirements` de la definición del trabajo.](#)

Si intenta anular estos requisitos de recursos, puede aparecer el siguiente mensaje de error:

“Este valor se envió en una clave obsoleta y puede entrar en conflicto con el valor proporcionado por los requisitos de recursos de la definición del trabajo”.

Para corregir esto, especifique los requisitos de memoria y vCPU en el miembro `ResourceRequirements` de `ContainerOverrides`. Por ejemplo, si las anulaciones de memoria y vCPU se especifican en las siguientes líneas.

```
"containerOverrides": {
```

```
"memory": 8192,  
"vcpus": 4  
}
```

Cámbielas a lo siguiente:

```
"containerOverrides": {  
  "resourceRequirements": [  
    {  
      "type": "MEMORY",  
      "value": "8192"  
    },  
    {  
      "type": "VCPU",  
      "value": "4"  
    }  
  ],  
}
```

Realice el mismo cambio en los requisitos de memoria y vCPU que se especifican en el objeto [ContainerProperties](#) de la definición del trabajo. Por ejemplo, si los requisitos de memoria y vCPU se especifican en las siguientes líneas.

```
{  
  "containerProperties": {  
    "memory": 4096,  
    "vcpus": 2,  
  }  
}
```

Cámbielos a lo siguiente:

```
"containerProperties": {  
  "resourceRequirements": [  
    {  
      "type": "MEMORY",  
      "value": "4096"  
    },  
    {  
      "type": "VCPU",  
      "value": "2"  
    }  
  ]  
}
```

```
] ,  
}
```

Mensaje de error al actualizar la configuración de **desiredvCpus**

Aparece el siguiente mensaje de error cuando utiliza la AWS Batch API para actualizar la configuración `v()` deseada. CPUs `desiredvCpus`

Manually scaling down compute environment is not supported. Disconnecting job queues from compute environment will cause it to scale-down to `minvCpus`.

Este problema se produce si el valor actualizado `desiredvCpus` es inferior al valor actual `desiredvCpus`. Al actualizar el valor `desiredvCpus`, deben cumplirse las dos condiciones siguientes:

- El valor `desiredvCpus` debe estar entre los valores `minvCpus` y `maxvCpus`.
- El valor actualizado `desiredvCpus` debe ser igual o mayor que el valor actual `desiredvCpus`.

AWS Batch en Amazon EKS

Temas

- [Entorno de computación de **INVALID**](#)
- [AWS Batch en Amazon EKS, el trabajo está **RUNNABLE** estancado](#)
- [AWS Batch en Amazon EKS, el trabajo está **STARTING** estancado](#)
- [Compruebe que `aws-auth` ConfigMap se ha configurado correctamente](#)
- [Los permisos o enlaces de RBAC no están configurados correctamente](#)

Consulte los siguientes temas para conocer los procesos de revisión y las posibles soluciones a los problemas comunes que pueden surgir al utilizar AWS Batch Amazon Elastic Kubernetes Service.

Entorno de computación de **INVALID**

Es posible que haya configurado incorrectamente un entorno de computación gestionado. Si lo ha hecho, el entorno de computación entra en un estado `INVALID` y no puede aceptar trabajos

para ubicarlos. En las siguientes secciones se describen las posibles causas y cómo solucionar los problemas en función de la causa.

La versión del Kubernetes no es compatible

Es posible que aparezca un mensaje de error similar al siguiente cuando utilice la operación de `CreateComputeEnvironment` API o la operación de `UpdateComputeEnvironment` API para crear o actualizar un entorno de computación. Este problema se produce si especifica una versión de Kubernetes no compatible en `EC2Configuration`.

```
At least one imageKubernetesVersion in EC2Configuration is not supported.
```

Para resolver este problema, elimine el entorno de procesamiento y vuelva a crearlo con una versión de Kubernetes compatible.

Puede realizar una actualización de una versión menor en su clúster de Amazon EKS. Por ejemplo, puede actualizar el clúster de 1.x.x a 1.y.y incluso si la versión secundaria no es compatible.

Sin embargo, es posible que el estado del entorno de computación cambie a `INVALID` después de una actualización de la versión principal. Por ejemplo, si realiza una actualización de una versión principal de 1.x.x a 2.y.y. Si la versión principal no es compatible con AWS Batch, aparecerá un mensaje de error similar al siguiente.

```
reason=CLIENT_ERROR - ... EKS Cluster version [2.yy] is unsupported
```

Para resolver este problema, especifique una versión Kubernetes compatible cuando utilice una operación de API para crear o actualizar un entorno de computación.

AWS Batch en Amazon EKS actualmente es compatible con las siguientes Kubernetes versiones:

- 1.34
- 1.33
- 1.32
- 1.31
- 1.30
- 1.29

El perfil de instancia no existe

Si el perfil de instancia especificado no existe, el estado del entorno AWS Batch de cómputo de Amazon EKS cambia a `INVALID`. Aparece un error establecido en el parámetro `statusReason` similar al siguiente.

```
CLIENT_ERROR - Instance profile arn:aws:iam::...:instance-profile/<name> does not exist
```

Para resolver este problema, especifique o cree un perfil de instancia que funcione. Para obtener más información, consulte [Rol de IAM de nodo Amazon EKS](#) en la Guía del usuario de Amazon EKS.

Espacio de nombres de Kubernetes no válido

Si AWS Batch en Amazon EKS no puede validar el espacio de nombres del entorno de cómputo, el estado del entorno de cómputo cambia a `INVALID`. Por ejemplo, este problema puede producirse si el espacio de nombres no existe.

Aparece un mensaje de error establecido en el parámetro `statusReason` similar al siguiente.

```
CLIENT_ERROR - Unable to validate Kubernetes Namespace
```

Este problema puede producirse si se cumple cualquiera de las siguientes condiciones:

- La cadena de espacio Kubernetes de nombres de la llamada `CreateComputeEnvironment` no existe. Para obtener más información, consulte [CreateComputeEnvironment](#).
- Los permisos de control de acceso basado en roles (RBAC) necesarios para administrar el espacio de nombres no están configurados correctamente.
- AWS Batch no tiene acceso al punto final del servidor de la Kubernetes API Amazon EKS.

Para resolver este problema, consulte [Compruebe que aws-auth ConfigMap se ha configurado correctamente](#). Para obtener más información, consulte [Cómo empezar con AWS Batch Amazon EKS](#).

Entorno de computación eliminado

Suponga que elimina un clúster de Amazon EKS antes de eliminar el entorno informático adjunto AWS Batch en Amazon EKS. A continuación, el estado del entorno de computación cambia a

INVALID. En este escenario, el entorno de computación no funciona correctamente si se vuelve a crear el clúster de Amazon EKS con el mismo nombre.

Para resolver este problema, elimine y, a continuación, vuelva a crear AWS Batch el entorno informático de Amazon EKS.

Los nodos no se unen al clúster de Amazon EKS

AWS Batch en Amazon EKS reduce la escala de un entorno informático si determina que no todos los nodos se han unido al clúster de Amazon EKS. Cuando AWS Batch en Amazon EKS reduce la escala del entorno de cómputo, el estado del entorno de cómputo cambia aINVALID.

Note

AWS Batch no cambia el estado del entorno de cómputo de forma inmediata para que pueda solucionar el problema.

Aparece un mensaje de error establecido en el parámetro `statusReason` similar a uno de los siguientes:

Your compute environment has been INVALIDATED and scaled down because none of the instances joined the underlying ECS Cluster. Common issues preventing instances joining are the following: VPC/Subnet configuration preventing communication to ECS, incorrect Instance Profile policy preventing authorization to ECS, or customized AMI or LaunchTemplate configurations affecting ECS agent.

Your compute environment has been INVALIDATED and scaled down because none of the nodes joined the underlying Amazon EKS Cluster. Common issues preventing nodes joining are the following: networking configuration preventing communication to Amazon EKS Cluster, incorrect Amazon EKS Instance Profile or Kubernetes RBAC policy preventing authorization to Amazon EKS Cluster, customized AMI or LaunchTemplate configurations affecting Amazon EKS/Kubernetes node bootstrap.

Cuando se utiliza una AMI Amazon EKS predeterminada, las causas más comunes de este problema son las siguientes:

- El rol de la instancia no está configurado correctamente. Para obtener más información, consulte [Rol de IAM de nodo Amazon EKS](#) en la Guía del usuario de Amazon EKS.
- Las subredes no están configuradas correctamente. Para obtener más información, consulte los [Requisitos y consideraciones de la VPC y las subredes de Amazon EKS](#) en la Guía del usuario de Amazon EKS.
- El grupo de seguridad no está configurado correctamente. Para obtener más información, consulte [Requisitos y consideraciones del grupo de seguridad de Amazon EKS](#) en la Guía del usuario de Amazon EKS.

 Note

También puede aparecer una notificación de error en el Personal Health Dashboard (PHD).

AWS Batch en Amazon EKS, el trabajo está **RUNNABLE** estancado

Un `aws-auth` ConfigMap se crea y aplica de forma automática al clúster cuando crea un grupo de nodos administrados o cuando crea un grupo de nodos mediante `eksctl`. Un `aws-auth` ConfigMap se crea inicialmente para permitir que los nodos se unan a su clúster. Sin embargo, también se utiliza `aws-auth` ConfigMap para agregar acceso de control de acceso basado en roles (RBAC) a los usuarios y roles.

Compruebe que `aws-auth` ConfigMap se ha configurado correctamente:

1. Recupere los roles mapeados en `aws-auth` ConfigMap:

```
$ kubectl get configmap -n kube-system aws-auth -o yaml
```

2. Compruebe que `roleARN` está configurado de la siguiente manera.

```
roleARN: arn:aws:iam::aws_account_number:role/AWSServiceRoleForBatch
```

 Note

También puede revisar los registros del plano de control de Amazon EKS. Para obtener más información, consulte [Registros del plano de control del clúster de Amazon EKS](#) en la Guía del usuario de Amazon EKS.

Para resolver un problema en el que un trabajo está atascado en un estado **RUNNABLE**, le recomendamos que use `kubectl` para volver a aplicar el manifiesto. Para obtener más información, consulte [Paso 2: Prepare el clúster de Amazon EKS para AWS Batch](#). O bien, puede usar `kubectl` para editar manualmente el `aws-authConfigMap`. Para obtener más información, consulte [Habilitación del usuario de IAM y el acceso de rol a un clúster](#) en la Guía del usuario de Amazon EKS.

AWS Batch en Amazon EKS, el trabajo está **STARTING** estancado

Es posible que un trabajo permanezca en estado **STARTING** cuando el pod esté bloqueado en **PENDING** en `ContainerCreating` para recibir solicitudes prolongadas del kubelet (`pull`, `log`, `exec` y `attach`) hasta que se resuelva el problema de inicio del pod o se finalice el trabajo. [En los siguientes casos, AWS Batch se cancelará el trabajo en su nombre; de lo contrario, el trabajo deberá finalizarse manualmente mediante la API. `TerminateJob`](#)

Para comprobar el motivo por el que un trabajo puede estar bloqueado en **STARTING**, utilice [Tutorial: asignación de un trabajo en ejecución a un pod y un nodo](#) para buscar el `podName`, describa el pod:

```
% kubectl describe pod aws-batch.000c8190-87df-31e7-8819-176fe017a24a -n my-aws-batch-namespace
Name:          aws-batch.000c8190-87df-31e7-8819-176fe017a24a
Namespace:     my-aws-batch-namespace
...
Containers:
  default:
    ...
    State:      Waiting
    Reason:     ContainerCreating
    Ready:      False
    ...
Conditions:
  Type                               Status
  PodReadyToStartContainers          False
  Initialized                         True
  Ready                              False
  ContainersReady                    False
  PodScheduled                       True
  ...
Events:
  Type    Reason      Age    From    Message
  ----    -
  ----    -
  ----    -
  ----    -
  ----    -
```

```
Warning   FailedMount   2m32s   kubelet   Unable to attach or mount volumes: ...
```

Considere la posibilidad de configurar su clúster de EKS para [enviar los registros del plano de control a CloudWatch los registros](#) para obtener una visibilidad total.

Situación: Error al adjuntar o montar una recuperación de volumen persistente

Los trabajos que utilizan recuperaciones de volumen persistente en las que el volumen no se adjunta o no se monta son candidatos a ser cancelados. Esto puede deberse a una configuración incorrecta de la definición de trabajo. Consulte [Creación de una definición de trabajo de un solo nodo en los recursos de Amazon EKS](#) para obtener más detalles.

Compruebe que **aws-auth ConfigMap** se ha configurado correctamente

Compruebe que aws-auth ConfigMap se ha configurado correctamente:

1. Recupere los roles mapeados en el aws-auth ConfigMap.

```
$ kubectl get configmap -n kube-system aws-auth -o yaml
```

2. Compruebe que roleARN está configurado de la siguiente manera.

```
roleARN: arn:aws:iam::aws_account_number:role/AWSServiceRoleForBatch
```

Note

La ruta `aws-service-role/batch.amazonaws.com/` se ha eliminado del ARN del rol vinculado a un servicio. Esto se debe a un problema con el mapa de configuración de aws-auth. Para obtener más información, consulte [Roles con rutas que no funcionan cuando la ruta está incluida en su ARN en el aws-authconfigmap](#).

Note

También puede revisar los registros del plano de control de Amazon EKS. Para obtener más información, consulte [Registros del plano de control del clúster de Amazon EKS](#) en la Guía del usuario de Amazon EKS.

Para resolver un problema en el que un trabajo está atascado en un estado `RUNNABLE`, le recomendamos que use `kubectl` para volver a aplicar el manifiesto. Para obtener más información, consulte [Paso 2: Prepare el clúster de Amazon EKS para AWS Batch](#). O bien, puede usar `kubectl` para editar manualmente el `aws-authConfigMap`. Para obtener más información, consulte [Habilitación del usuario de IAM y el acceso de rol a un clúster](#) en la Guía del usuario de Amazon EKS.

Los permisos o enlaces de RBAC no están configurados correctamente

Si tiene algún problema con los permisos o los enlaces del RBAC, compruebe que el rol `aws-batch` de Kubernetes pueda acceder al espacio de nombres Kubernetes:

```
$ kubectl get namespace namespace --as=aws-batch
```

```
$ kubectl auth can-i get ns --as=aws-batch
```

También puede usar el comando **`kubectl describe`** para ver las autorizaciones de un rol de clúster o espacio de nombres Kubernetes.

```
$ kubectl describe clusterrole aws-batch-cluster-role
```

A continuación, se muestra un ejemplo del resultado.

```
Name:      aws-batch-cluster-role
Labels:    <none>
Annotations: <none>
PolicyRule:
  Resources            Non-Resource URLs  Resource Names
  Verbs
  -----
  -----
  configmaps          []                 []
[get list watch]
  nodes               []                 []
[get list watch]
  pods                []                 []
[get list watch]
  daemonsets.apps     []                 []
[get list watch]
```

deployments.apps	[]	[]
[get list watch]		
replicasets.apps	[]	[]
[get list watch]		
statefulsets.apps	[]	[]
[get list watch]		
clusterrolebindings.rbac.authorization.k8s.io	[]	[]
[get list]		
clusterroles.rbac.authorization.k8s.io	[]	[]
[get list]		
namespaces	[]	[]
[get]		
events	[]	[]
[list]		

```
$ kubectl describe role aws-batch-compute-environment-role -n my-aws-batch-namespace
```

A continuación, se muestra un ejemplo del resultado.

Name: aws-batch-compute-environment-role			
Labels: <none>			
Annotations: <none>			
PolicyRule:			
Resources	Non-Resource URLs	Resource Names	Verbs
-----	-----	-----	-----
pods	[]	[]	[create
get list watch delete patch]			
serviceaccounts	[]	[]	[get list]
rolebindings.rbac.authorization.k8s.io	[]	[]	[get list]
roles.rbac.authorization.k8s.io	[]	[]	[get list]

Para resolver este problema, vuelva a aplicar los permisos RBAC y comandos rolebinding. Para obtener más información, consulte [Paso 2: Prepare el clúster de Amazon EKS para AWS Batch](#).

Recurso: cuotas AWS Batch de servicio

En la siguiente tabla se muestran las cuotas de servicio AWS Batch que no se pueden cambiar. Cada cuota es específica de la región.

Recurso	Cuota
Número máximo de colas de trabajo. Para obtener más información, consulte Colas de trabajos .	50
Número máximo de entornos de computación en Amazon ECS y Amazon EKS. Para obtener más información, consulte Entornos informáticos para AWS Batch .	50
Número máximo de entornos de computación por clúster de Amazon EKS.	5
Número máximo de entornos de computación para cada cola de trabajos	3
Cantidad máxima de dependencias de un trabajo	20
Tamaño máximo de la definición del trabajo (para operaciones de la API RegisterJobDefinition)	24 KiB
El tamaño máximo de la carga del trabajo (para operaciones de API SubmitJob)	30 KiB
Tamaño máximo de matriz para los trabajos de matrices	10000
Cantidad máxima de trabajos en estado SUBMITTED	1000000
Número máximo de transacciones por segundo (TPS) para cada cuenta de operaciones SubmitJob	50
Número máximo de recursos consumibles	50 000
Número máximo de entornos de servicio. Para obtener más información, consulte Entornos de servicio .	50
Número máximo de entornos de computación para cada cola de trabajos	1

Recurso	Cuota
Número máximo de colas de trabajos asociadas a cada entorno de servicio habilitado para la administración de cuotas. Para obtener más información, consulte the section called “Gestión de cuotas” .	1
Número máximo de cuotas compartidas para cada cola de trabajos de administración de cuotas. Para obtener más información, consulte the section called “Gestión de cuotas” .	20
Tamaño máximo de una solicitud SubmitServiceJob	30 KiB
El tamaño máximo de la carga útil de las solicitudes de servicio de trabajo (para operaciones de API SubmitServiceJob)	10 KiB
Número máximo de transacciones por segundo (TPS) para cada cuenta de operaciones SubmitServiceJob	5
Número máximo de intentos con estrategia de reintento para un trabajo de servicio	10

En función de cómo se utilice AWS Batch, es posible que se apliquen cuotas adicionales. Para obtener información sobre las cuotas de Amazon EC2, consulte [Amazon EC2 Service Quotas](#) en Referencia general de AWS. Para obtener más información acerca de las cuotas de servicio de Amazon ECS, consulte [Amazon ECS Service Quotas](#) en Referencia general de AWS. Para obtener más información acerca de las cuotas de servicio de Amazon EKS, consulte las [Amazon EKS Service Quotas](#) en Referencia general de AWS.

Historial de documentos

En la siguiente tabla se describen los cambios importantes en la documentación desde la publicación inicial de AWS Batch. Actualizamos la documentación con frecuencia para dar respuesta a los comentarios que se nos envía.

Cambio	Descripción	Fecha
Se agregaron default_x86_64 y default_arm64	Se agregaron nuevos tipos de instancia permitidos: default_x86_64 y default_arm64 .	15 de agosto de 2025
Se agregaron entornos de servicio y trabajos de servicio	Se agregaron entornos de servicio y trabajos de servicio para su uso AWS Batch con SageMaker IA.	30 de julio de 2025
Se agregaron AWSServiceRoleForAWSBatchWithSageMaker y AWSServiceRoleForSageMaker	Se agregó un nuevo rol AWS vinculado al servicio AWSServiceRoleForAWSBatchWithSageMaker y una política administrada AWSBatchServiceRolePolicyForSageMaker que permite AWS Batch administrar la SageMaker IA en su nombre.	30 de julio de 2025
Añade compatibilidad con EKS AL 2023 AMIs	¿Cómo actualizar de EKS AL2 a EKS AL2023	24 de junio de 2025
Añade compatibilidad con un FireLens comando ECS Exec	Añade compatibilidad con un FireLens comando ECS Exec.	15 de abril de 2025
Añade compatibilidad con la programación basada en los recursos para AWS Batch	Añade compatibilidad con la programación basada en los recursos para AWS Batch	27 de febrero de 2025

Amazon Elastic Container Service, Amazon Elastic Kubernetes Service y. AWS Fargate

[Se AWS Batch actualizaron las versiones compatibles de Amazon EKS](#)

Se actualizaron las versiones de Amazon EKS AWS Batch compatibles para eliminar la versión 1.22.

11 de marzo de 2024

[Se AWS Batch actualizaron las versiones compatibles de Amazon EKS](#)

Se actualizaron las versiones de Amazon EKS AWS Batch compatibles para incluir la versión 1.29.

29 de febrero de 2024

[Reintentos automáticos de trabajo](#)

Se ha corregido el ejemplo de código.

29 de febrero de 2024

[Añade compatibilidad con trabajos con varios contenedores para AWS Batch](#)

Añade compatibilidad con trabajos de varios contenedores AWS Batch para Amazon Elastic Container Service, Amazon Elastic Kubernetes Service y. AWS Fargate

28 de febrero de 2024

[Se AWS Batch actualizaron las versiones compatibles de Amazon EKS](#)

Se actualizaron las versiones de Amazon EKS AWS Batch compatibles para incluir la versión 1.28

27 de enero de 2024

[Actualización de BatchServiceRolePolicy y AWSBatchServiceRole](#)

BatchServiceRolePolicy

5 de diciembre de 2023

Se ha actualizado para agregar asistencia al describir el historial de solicitudes y las actividades Amazon EC2 Auto Scaling de la flota de spot.

AWSBatchServiceRole

Se actualizó para añadir una declaración IDs, conceder AWS Batch permisos `ec2:DescribeSpotFleetRequestsHistory` y `autoscaling:DescribeScalingActivities`.

[AWS Batch en Amazon EKS](#)

AWS Batch añade compatibilidad para ejecutar trabajos en clústeres de Amazon EKS.

25 de octubre de 2022

[La prevención policial confusa entre servicios para AWS Batch](#)

AWS Batch ahora ofrece una solución alternativa al confuso problema de seguridad adjunto, que surge cuando una entidad (un servicio o una cuenta) es obligada por otra entidad a realizar una acción por parte de otra entidad.

6 de junio de 2022

Puntos de conexión de VPC de interfaz (AWS PrivateLink)	Se agregó soporte para configurar puntos finales de VPC de interfaz alimentados por. AWS PrivateLink Esto significa que puede crear una conexión privada entre su VPC y AWS Batch sin necesidad de acceso a través de una instancia de NAT, una conexión VPN o. Direct Connect	15 de abril de 2022
Actualizaciones mejoradas del entorno de computación	AWS Batch actualizaciones de soporte mejoradas para entornos informáticos.	14 de abril de 2022
AWS actualizaciones de políticas gestionadas: actualización de las políticas existentes	AWS Batch actualizó las políticas gestionadas existentes.	6 de diciembre de 2021
Programación de reparto justo	AWS Batch añade soporte para añadir políticas de programación a las colas de trabajos.	9 de noviembre de 2021
Amazon EFS	AWS Batch añade compatibilidad para añadir sistemas de archivos Amazon EFS a las definiciones de trabajo.	1 de abril de 2021
Rol vinculado al servicio añadido	AWS Batch añade el rol AWSServiceRoleForBatch vinculado al servicio.	10 de marzo de 2021
AWS Fargate soporte	AWS Batch añade soporte para ejecutar trabajos en los recursos de Fargate.	3 de diciembre de 2020

Etiquetado de recursos	AWS Batch añade compatibilidad para añadir etiquetas de metadatos a sus entornos informáticos, definiciones de trabajos, colas de trabajos y trabajos.	7 de octubre de 2020
Secretos	AWS Batch añade soporte para transmitir secretos a los trabajos.	1 de octubre de 2020
Registro	AWS Batch añade compatibilidad para especificar controladores de registro adicionales para los trabajos.	1 de octubre de 2020
Estrategias de asignación	AWS Batch añade compatibilidad con varias estrategias para elegir tipos de instancias.	16 de octubre de 2019
Compatibilidad con EFA	AWS Batch añade soporte para dispositivos Elastic Fabric Adapter (EFA).	2 de agosto de 2019
Programación de GPU	AWS Batch añade la programación de la GPU. Con esta función, puede especificar la cantidad que requiere GPUs cada trabajo y ampliar AWS Batch las instancias en consecuencia.	4 de abril de 2019

<u>Trabajos paralelos de varios nodos</u>	AWS Batch añade soporte para trabajos paralelos de varios nodos. Puede utilizar esta característica para ejecutar trabajos individuales que abarquen varias instancias de Amazon EC2.	19 de noviembre de 2018
<u>Permisos de nivel de recursos</u>	AWS Batch admite permisos a nivel de recursos en varias operaciones de API.	12 de noviembre de 2018
<u>Compatibilidad con las plantillas de lanzamiento de Amazon EC2</u>	AWS Batch añade compatibilidad con el uso de plantillas de lanzamiento con entornos informáticos.	12 de noviembre de 2018
<u>AWS Batch tiempos de espera de los trabajos</u>	AWS Batch añade compatibilidad con el tiempo de espera del trabajo. Con este soporte, puede configurar un tiempo de espera específico para sus trabajos de modo que, si un trabajo dura más de lo debido, AWS Batch finalice el trabajo.	5 de abril de 2018
<u>AWS Batch trabajos como objetivos EventBridge</u>	AWS Batch los puestos de trabajo están disponibles como EventBridge objetivos . Al crear reglas sencillas , puede hacer coincidir los eventos y enviar trabajos de AWS Batch en respuesta a ellos.	1 de marzo de 2018

[CloudTrail auditar para AWS Batch](#)

CloudTrail puede auditar las llamadas realizadas a las acciones AWS Batch de la API.

10 de enero de 2018

[Trabajos de matrices](#)

AWS Batch añade soporte para trabajos de matriz. Puede utilizar trabajos de matrices para el barrido de parámetros y para las cargas de trabajo de Monte Carlo.

28 de noviembre de 2017

[AWS Batch Etiquetado ampliado](#)

AWS Batch amplía el soporte para la función de etiquetado. Puede utilizar esta función para especificar etiquetas para las instancias de spot de Amazon EC2 lanzadas en entornos de computación gestionados.

26 de octubre de 2017

[AWS Batch flujo de eventos para EventBridge](#)

AWS Batch añade el flujo de eventos para EventBridge. Puede utilizar la transmisión de AWS Batch eventos para recibir notificaciones casi en tiempo real sobre el estado de los trabajos que se envían a sus colas de trabajos.

24 de octubre de 2017

[Reintentos automáticos de trabajo](#)

AWS Batch añade soporte para reintentos de trabajo. Con esta actualización, puede aplicar una estrategia de reintento a sus trabajos y definiciones de trabajo, que les permita reintentar ejecutars e automáticamente si fallan.

28 de marzo de 2017

[AWS Batch disponibilidad general](#)

AWS Batch se presenta, diseñado como un medio para ejecutar cargas de trabajo de computación por lotes en el Nube de AWS.

5 de enero de 2017

Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la version original de inglés, prevalecerá la version en inglés.