Guía del usuario de

AWS HealthOmics

Version latest

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

AWS HealthOmics: Guía del usuario de

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

Las marcas registradas y la imagen comercial de Amazon no se pueden utilizar en 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 HealthOmics? 1
Aviso importante 1
HealthOmics features 1
Conceptos 2
Flujos de trabajo 3
Almacenamiento 3
Análisis 4
Servicios relacionados 4
¿Cómo acceder HealthOmics 5
Regiones y puntos finales para AWS HealthOmics 5
Más información 5
AWS HealthOmics cambio en la disponibilidad del almacén de variantes y del almacén de anotaciones 7
Descripción general de las opciones de migración 7
Opciones de migración para la lógica ETL 8
Opciones de migración para el almacenamiento 8
Análisis 8
AWS Socios 9
Ejemplos 9
Athena DDL 9
Crear tablas con Python (sin Athena) 10
Con HealthOmicsfiguración 13
Inscríbase en una Cuenta de AWS 13
Creación de un usuario con acceso administrativo 14
Cree permisos de IAM para HealthOmics 15
Conéctese con repositorios de código externos 15
Uso de Amazon Q CLI con HealthOmics 16
Introducción 17
Uso de un flujo de trabajo de Ready2Run en la consola HealthOmics 17
Ejemplos de solicitudes para Amazon Q CLI 18
Flujos de trabajo privados 19
Creación de flujos de trabajo 20
Integración del repositorio de Git 21
Archivos de definición de flujo de trabajo 25

Archivos de plantillas de parámetros 81
Imágenes de contenedor 94
Archivos README del flujo de trabajo 108
Opcional: licencias de Sentieon 111
Linteros de flujo de trabajo 112
Operaciones de flujo de trabajo 113
Control de versiones del flujo de trabajo 130
Versión predeterminada 132
Creación de una versión 132
Actualizar una versión 139
Eliminación de una versión 141
HealthOmics corre 143
Ejecute tipos de almacenamiento 144
Ejecute los modos de retención 148
Ejecute las entradas 149
Ejecución del ciclo de vida 154
Ejecute las salidas 158
Motivos de error de ejecución 161
Ciclo de vida de las tareas 165
Ejecute la optimización 168
Ejecute operaciones 177
Ejecuta grupos 190
Prioridad de ejecución 190
Crea un grupo de carreras mediante la consola 191
Cree un grupo de ejecución mediante la CLI 192
Elimina un grupo de carreras mediante la consola 193
Eliminar un grupo de ejecución mediante la CLI 193
Almacenamiento en caché de llamadas 193
Cómo funciona el almacenamiento en caché de llamadas 194
Crear una caché de ejecución 201
Actualización de una caché de ejecución 203
Eliminar una caché de ejecución 203
Contenido de una caché de ejecución 204
Funciones de almacenamiento en caché específicas del motor 205
Uso de la caché de ejecución 206
Compartir flujos de trabajo 211

Suscribirse a un flujo de trabajo compartido 212
Supervisar el estado de un flujo de trabajo compartido 212
Compartir un flujo de trabajo privado mediante la consola 213
Compartir un flujo de trabajo privado mediante la CLI 213
Aceptar un flujo de trabajo compartido mediante la consola 214
Ejecutar un flujo de trabajo compartido mediante la consola 214
Ejecutar un flujo de trabajo compartido mediante la API 215
Flujos de trabajo Ready2Run 216
Flujos de trabajo disponibles 217
Suscribirse a los flujos de trabajo de Sentieon 223
Inicio de los flujos de trabajo de Ready2Run (consola) 224
Inicio de los flujos de trabajo de Ready2Run (API) 225
HealthOmics almacenamiento 227
HealthOmics ETags 228
Amazon S3 ETags 228
¿Cómo se HealthOmics calcula ETags 229
Crear una tienda de referencia 230
Crear un almacén de referencias mediante la consola 230
Creación de un almacén de referencias mediante la CLI 231
Crear un almacén de secuencias 236
Crear un almacén de secuencias mediante la consola 237
Creación de un almacén de secuencias mediante la CLI 238
Actualización de un almacén de secuencias 240
Actualización de las etiquetas de los conjuntos de lectura de un almacén de secuencias 241
Importación de archivos genómicos 241
Eliminar tiendas 242
Importación de conjuntos de lecturas a un almacén de secuencias 243
Cargar archivos a Amazon S3 243
Creación de un archivo de manifiesto 244
Iniciar el trabajo de importación 247
Supervise el trabajo de importación 247
Busque los archivos de secuencias importados 249
Obtén detalles sobre un conjunto de lecturas 252
Descargue los archivos de datos del conjunto de lectura 254
Carga directa a un almacén de secuencias 254
Carga directamente a un almacén de secuencias mediante el AWS CLI 255

Configure una ubicación alternativa ... 260
Exportación de conjuntos de lectura ... 261
Acceso a conjuntos de lectura con Amazon S3 URIs ... 264
Estructura de URI de Amazon S3 en el HealthOmics almacenamiento ... 265
Uso de IGV alojado o local para acceder a los conjuntos de lectura ... 266
Usando Samtools o en HTSlib HealthOmics ... 267
Uso de Mountpoint HealthOmics ... 267
Utilizándolo CloudFront con HealthOmics ... 268
Activar conjuntos de lectura ... 268
HealthOmics análisis ... 272
Creación de tiendas de variantes ... 273
Crear un almacén de variantes mediante la consola ... 273
Crear un almacén de variantes mediante la API ... 274
Creación de trabajos de importación de tiendas de variantes ... 276
Creación de almacenes de anotaciones ... 280
Crear un almacén de anotaciones mediante la consola ... 280
Crear un almacén de anotaciones mediante la API ... 281
Creación de trabajos de importación del almacén de anotaciones ... 283
Crear un trabajo de importación de anotaciones mediante la API ... 283
Parámetros adicionales para los formatos TSV y VCF ... 285
Creación de almacenes de anotaciones con formato TSV ... 287
Inicio de trabajos de importación con formato VCF ... 290
Creación de versiones del almacén de anotaciones ... 290
Eliminar almacenes de análisis ... 294
Consulta de datos de análisis ... 295
Configuración de Lake Formation ... 296
Configuración de Athena para consultas ... 299
Ejecutando consultas ... 300
Compartir tiendas de análisis ... 302
Crear un recurso compartido en la tienda ... 302
Intercambio de recursos ... 304
Crear un recurso compartido ... 305
Recupera información sobre un recurso compartido ... 305
Consulta las acciones de las que eres propietario ... 306
Consulta las acciones aceptadas de otras cuentas ... 306
Eliminar un recurso compartido ... 306

Etiquetado de recursos en HealthOmics 308
Aviso importante 308
Recursos de etiquetado HealthOmics 308
Prácticas recomendadas 310
Requisitos de etiquetado 310
Secuencia: almacene, lea y establezca etiquetas 311
Agregar una etiqueta 311
Enumeración de etiquetas 312
Eliminación de etiquetas 313
Permisos 314
Políticas de usuario 314
Defina permisos de IAM personalizados para las ejecuciones 316
Roles de servicio 317
Ejemplo de políticas de servicio de IAM 318
Ejemplo de CloudFormation plantilla 321
Permisos de Amazon ECR 323
Cree una política de recursos para el repositorio de Amazon ECR 323
Ejecutar flujos de trabajo con contenedores multicuenta 324
Políticas de Amazon ECR para flujos de trabajo compartidos 326
Las políticas de Amazon ECR extraen la memoria caché 329
Permisos de recursos 333
Permisos de Lake Formation 333
Permisos de URI de Amazon S3 334
Uso compartido basado en políticas 335
Ejemplo de restricción 339
Seguridad 343
Protección de datos 343
Cifrado en reposo 345
Cifrado en tránsito 356
Identity and Access Management 356
Público 356
Autenticación con identidades 357
Administración del acceso con políticas 358
¿Cómo AWS HealthOmics funciona con IAM 360
Ejemplos de políticas basadas en identidades 367
AWS políticas gestionadas 370

Resolución de problemas ... 373
Validación de conformidad ... 375
Resiliencia ... 377
Puntos de conexión de VPC (AWS PrivateLink) ... 378
Consideraciones sobre los puntos HealthOmics finales de VPC ... 378
Creación de un punto final de VPC de interfaz para HealthOmics ... 379
Crear una política de puntos de conexión de VPC para HealthOmics ... 380
Consideraciones especiales para acceder a los conjuntos de lectura mediante Amazon S3 URIs ... 381
Supervisión de AWS HealthOmics ... 382
Registro de acceso a S3 ... 383
CloudWatch métricas ... 384
Visualización de AWS HealthOmics métricas ... 384
Crear una alarma de ... 385
CloudWatch Registros ... 386
Tipos de registro para HealthOmics flujos de trabajo ... 386
Inicia sesión CloudWatch ... 387
Inicia sesión en Amazon S3 ... 388
CloudWatch Registros interactivos en la CLI ... 389
Acceder a CloudWatch los registros desde la consola ... 389
CloudTrail registros ... 390
HealthOmics información en CloudTrail ... 391
Descripción de las entradas de los archivos de HealthOmics registro ... 392
EventBridge ... 393
Configurar EventBridge para HealthOmics ... 394
EventBridge eventos en HealthOmics ... 395
Estructura de mensaje de evento ... 397
Ejemplos de mensajes de eventos ... 398
Resolución de problemas ... 401
Solución de problemas de flujos de trabajo ... 401
¿Cómo puedo solucionar una ejecución fallida? ... 401
¿Cómo soluciono los problemas de una tarea fallida? ... 401
¿Dónde puedo encontrar los registros del motor de las ejecuciones completadas correctamente? ... 402
¿Cómo puedo reducir el tamaño de los parámetros de entrada de un flujo de trabajo? ... 402
¿Por qué no se completa mi ejecución? ... 402

Solución de problemas de almacenamiento en caché de llamadas 402
¿Por qué mi ejecución no se guarda en la caché? 402
¿Por qué una tarea no utiliza la entrada de caché? 402
¿Por qué está desactivado el almacenamiento en caché de llamadas de una tarea? 403
Solución de problemas de almacenes de datos 404
¿Por qué GetObject falla S3 en mi equipo de lectura? 404
¿Por qué no puedo ver mi tienda de anotaciones o mi tienda de variantes en Athena? 405
¿Por qué no puedo acceder a mi almacén de datos en Athena? 405
Solución de problemas con Amazon Q CLI 405
Cuotas 406
Cuotas de servicio 406
Cuotas de tamaño fijo 411
Cuotas de tamaño de los archivos de Analytics 412
cuotas de tamaño de archivos de almacenamiento 412
Cuotas de tamaño fijo del flujo de trabajo 414
Cuotas de tamaño fijo del flujo de trabajo de Ready2Run 417
cuotas de API 421
Cuotas generales de API 421
Cuotas de API de almacenamiento 421
Cuotas de la API de flujo 423
Cuotas de la API de análisis 424
Historial de revisión 426
..... cdxxxii

¿Qué es AWS HealthOmics?

AWS HealthOmics es un servicio que cumple con los requisitos de la HIPAA y que acelera las pruebas de diagnóstico clínico, el descubrimiento de fármacos y la investigación agrícola mediante la gestión integral de la compleja infraestructura que sustenta sus flujos de trabajo bioinformáticos. HealthOmics es compatible con los lenguajes de flujo de trabajo estándares del sector (WDL, Nextflow, CWL) y amplía sin problemas la infraestructura bioinformática para admitir los datos de decenas de miles de pruebas al día, todo ello con un coste por muestra predecible. HealthOmics gestiona las complejidades técnicas, como la gestión de los recursos informáticos y el mantenimiento de los motores de flujo de trabajo, para que pueda centrarse por completo en los avances científicos.

Temas

- Aviso importante
- HealthOmics features
- HealthOmics conceptos
- Servicios relacionados
- ¿Cómo acceder HealthOmics
- Regiones y puntos finales para AWS HealthOmics
- Más información

Aviso importante

HealthOmics está diseñado únicamente para transferir, almacenar, formatear o mostrar datos, y para proporcionar soporte de infraestructura y configuración para administrar los flujos de trabajo. HealthOmics no sustituye el asesoramiento, el diagnóstico o el tratamiento de un médico profesional, y no pretende curar, tratar, mitigar, prevenir o diagnosticar ninguna enfermedad o afección de salud. Usted es responsable de instituir una revisión humana como parte de cualquier uso de cualquier producto de terceros destinado a fundamentar la toma de decisiones clínicas AWS HealthOmics, incluso en asociación con él.

HealthOmics features

Casos de uso principales para HealthOmics:

- Diagnóstico clínico: cree y escale los flujos de trabajo de las pruebas de diagnóstico con costes predecibles y una infraestructura totalmente gestionada que crezca con el volumen de pruebas.
- Descubrimiento de fármacos: acelere la investigación terapéutica organizando modelos biológicos básicos a escala, lo que permitirá una rápida iteración entre millones de posibles candidatos.
- Investigación agrícola: mejore las características de los cultivos, como la tolerancia a la sequía y la resistencia a las plagas, mediante flujos de trabajo impulsados por la IA que mejoran la seguridad alimentaria y la productividad agrícola.

Principales beneficios de: HealthOmics

- Escalabilidad: escale los flujos de trabajo a más de 100 000 V simultáneos CPUs para realizar decenas de miles de pruebas al día sin necesidad de gestionar la infraestructura y con un coste por muestra predecible.
- Céntrese en la ciencia, no en la infraestructura: utilice lenguajes de flujo de trabajo conocidos y, APIs al mismo tiempo, gestione AWS automáticamente la infraestructura y la gestión de datos entre bastidores.
- Mantenga el cumplimiento: los registros de auditoría integrales, el seguimiento de la procedencia de los datos y una infraestructura compatible con la HIPAA diseñada para los flujos de trabajo clínicos respaldan el desarrollo de soluciones que cumplen con los out-of-the-box requisitos reglamentarios.

HealthOmics consta de tres componentes principales:

- HealthOmics flujos de trabajo: ejecute cálculos bioinformáticos en una infraestructura escalada y aprovisionada automáticamente.
- HealthOmics almacenamiento: almacene y comparta petabytes de datos genómicos de manera eficiente a un bajo costo por gigabase.
- HealthOmics análisis: prepare los datos genómicos para los análisis multiómicos y multimodales.

Utilice estos componentes de forma independiente o combínelos para crear una solución. end-to-end

HealthOmics conceptos

Este tema cubre las definiciones de los conceptos y términos clave específicos de HealthOmics, para ayudarlo a comprender la terminología HealthOmics utilizada en esta guía.

Temas

- Flujos de trabajo
- Almacenamiento
- Análisis

Flujos de trabajo

Con HealthOmics Workflows, puede procesar y analizar sus datos genómicos.

- Flujo de trabajo: la definición general de un proceso integral, incluidos los parámetros y las referencias a las herramientas. Las definiciones de flujo de trabajo se pueden expresar como WDL, Nextflow o CWL. Cada flujo de trabajo creado tiene un identificador único.
- Ejecutar: una sola invocación de un flujo de trabajo. Una ejecución individual utiliza los datos de entrada definidos y produce una salida. Cada ejecución creada tiene un identificador único.
- Tarea: los procesos individuales de una ejecución. HealthOmics Los flujos de trabajo utilizan estas especificaciones informáticas definidas para ejecutar la tarea. Cada tarea tiene un identificador único.
- Grupo de ejecuciones: grupo de ejecuciones para el que puedes establecer el número máximo de vCPU, la duración máxima o el máximo de ejecuciones simultáneas para limitar los recursos informáticos utilizados por ejecución. Puedes especificar y configurar las prioridades de tus ejecuciones dentro de un grupo de ejecuciones. Por ejemplo, puedes especificar que se ejecute una ejecución de alta prioridad antes que una de menor prioridad, creando una cola de prioridad. El uso de un grupo de ejecución es opcional y cada grupo de ejecución tiene un identificador único.

Almacenamiento

El almacenamiento de datos se divide en almacenes de secuencias, para las secuencias genómicas e información relacionada, y un almacén de referencias, para todos los genomas de referencia. Los siguientes términos describen las implementaciones específicas de. HealthOmics

- Almacén de secuencias: almacén de datos para el almacenamiento de archivos genómicos. Puede tener uno o más almacenes de secuencias en su interior HealthOmics. Los permisos de acceso y el AWS KMS cifrado se pueden configurar en un almacén de secuencias para controlar quién tiene acceso a los datos.

- Conjunto de lecturas: un conjunto de lecturas es una abstracción de las lecturas genómicas, que se almacenan en los formatos FASTQ, BAM o CRAM. Los conjuntos de lectura pueden importarse a almacenes de secuencias y anotarse con metadatos. Puede aplicar permisos a los conjuntos de lectura mediante el control de acceso basado en atributos (ABAC).
- Referencia: una referencia genómica se utiliza con las lecturas para identificar en qué parte del genoma se mapea una lectura específica o un grupo de lecturas. Están en formato FASTA y se almacenan en el almacén de referencias.
- Almacén de referencia: almacén de datos para el almacenamiento de genomas de referencia. Puede tener un único almacén de referencias en cada cuenta y región.

Análisis

Puede transformar y analizar sus datos genómicos con HealthOmics Analytics. Cree un almacén de variantes o un almacén de anotaciones para incluir información adicional para sus consultas.

- Almacén de variantes: almacén de datos que almacena datos de variantes a escala de población. Los almacenes de variantes admiten tanto el formato de llamada de variantes genómicas (gvCF) como las entradas VCF.
- Almacén de anotaciones: banco de datos que representa una base de datos de anotaciones, como una de un archivo TSV/CSV, VCF o General Feature Format (). GFF3 Los almacenes de anotaciones se asignan al mismo sistema de coordenadas que los almacenes de variantes durante una importación.

Servicios relacionados

Los siguientes servicios funcionan con. HealthOmics

- Amazon Elastic Container Registry: cada flujo de trabajo privado utiliza una imagen de Amazon ECR (en un repositorio privado de Amazon ECR) para contener todos los ejecutables, bibliotecas y scripts necesarios para ejecutar el flujo de trabajo.
- Amazon Simple Storage Service: Amazon S3 proporciona almacenamiento de archivos para los datos de la tienda y del flujo de trabajo.
- AWS Lake Formation — Lake Formation gestiona el acceso a los datos de sus almacenes de datos de Analytics.
- Amazon Athena: usa Athena para realizar consultas en tus tiendas de variantes.

- Amazon SageMaker AI: utilice la SageMaker IA para ejecutar HealthOmics tareas con los cuadernos Jupyter.
- GitHub connections— Utilice conexiones para conectar sus repositorios de código externos a sus flujos de trabajo. HealthOmics

¿Cómo acceder HealthOmics

Puede acceder a AWS HealthOmics las funciones mediante la consola de administración, la CLI SDKs o la API.

- AWS Consola de administración: proporciona una interfaz web a la que puede acceder HealthOmics.
- AWS Command Line Interface (AWS CLI): proporciona comandos para un amplio conjunto de AWS servicios AWS HealthOmics, incluidos y es compatible con Windows, macOS y Linux. Para obtener más información sobre la instalación de AWS CLI, consulte AWS Command Line Interface.
- AWS SDKs — AWS proporciona SDKs (kits de desarrollo de software) que consisten en bibliotecas y código de muestra para varios lenguajes de programación y plataformas (incluidos Java, Python, Ruby, .NET, iOS y Android). SDKs Proporcionan una forma cómoda de utilizarlos HealthOmics mediante programación. Para obtener más información, consulta el Centro de desarrolladores del AWS SDK.
- AWS API: puede usar las operaciones de la API para acceder a ellas y administrarlas HealthOmics mediante programación. Para obtener más información, consulte la Referencia de la API de HealthOmics .

Regiones y puntos finales para AWS HealthOmics

Para obtener una lista completa de regiones y puntos finales, consulte la Referencia AWS general.

Además de las AWS regiones que están activas de forma predeterminada, también hay regiones de suscripción que deben activarse. Para obtener más información sobre cómo activar o desactivar una región, consulta Especificar qué AWS regiones puede usar tu cuenta en la guía de administración de AWS cuentas.

Más información

Obtenga más información HealthOmics en estos talleres y tutoriales:

- HealthOmics taller: taller de principio a HealthOmics fin
- AWS recursos genómicos — Repositorios públicos de Amazon ECR relacionados con la genómica
- Tutoriales de Python: tutoriales de Jupyter Notebook sobre HealthOmics almacenamiento GitHub, análisis y flujos de trabajo

Familiarícese con HealthOmics herramientas adicionales que AWS ofrecen:

- WDL linter: HealthOmics linter para WDL
- linter Nextflow — linter para Nextflow HealthOmics
- HealthOmics Herramienta auxiliar de Amazon ECR: herramienta de ayuda de Amazon ECR para HealthOmics
- HealthOmics herramientas sobre GitHub : herramientas con las que trabajar HealthOmics (gestor de transferencias, analizador de URI, reejecución de Omics, analizador de ejecución).

AWS HealthOmics cambio en la disponibilidad del almacén de variantes y del almacén de anotaciones

Tras pensarlo detenidamente, decidimos cerrar las tiendas de AWS HealthOmics variantes y las tiendas de anotaciones a los nuevos clientes a partir del 7 de noviembre de 2025. Los clientes existentes pueden seguir utilizando el servicio con normalidad.

En la siguiente sección, se describen las opciones de migración para ayudarte a migrar tus tiendas de variantes y tiendas de análisis a nuevas soluciones. Si tienes alguna pregunta o duda, crea un caso de soporte en support.console.aws.amazon.com.

Temas

- Descripción general de las opciones de migración
- Opciones de migración para la lógica ETL
- Opciones de migración para el almacenamiento
- Análisis
- AWS Socios
- Ejemplos

Descripción general de las opciones de migración

Las siguientes opciones de migración ofrecen una alternativa al uso de almacenes de variantes y almacenes de anotaciones:

1. Utilice la implementación HealthOmics de referencia de la lógica ETL proporcionada.

   Utilice los depósitos de tablas de S3 como almacenamiento y siga utilizando los servicios de AWS análisis existentes.
2. Cree una solución mediante una combinación de los AWS servicios existentes.

   Para ETL, puedes escribir trabajos ETL de Glue personalizados o usar código HAIL o GLOW de código abierto en EMR para transformar los datos de variantes.

   Utilice los cubos de tablas de S3 para almacenar y siga utilizando los servicios de análisis existentes AWS

3. Seleccione un AWS socio que ofrezca una variante y una alternativa al almacén de anotaciones.

Opciones de migración para la lógica ETL

Tenga en cuenta las siguientes opciones de migración para la lógica de ETL:

1. HealthOmics proporciona la lógica ETL del almacén de variantes actual como HealthOmics flujo de trabajo de referencia. Puede utilizar el motor de este flujo de trabajo para impulsar exactamente el mismo proceso de ETL de datos de variantes que el almacén de variantes, pero con un control total sobre la lógica de ETL.

   Este flujo de trabajo de referencia está disponible previa solicitud. Para solicitar acceso, crea un caso de soporte en support.console.aws.amazon.com.
2. Para transformar los datos de variantes, puedes escribir trabajos ETL de Glue personalizados o usar código HAIL o GLOW de código abierto en EMR.

Opciones de migración para el almacenamiento

Como reemplazo del almacén de datos hospedado en un servicio, puede utilizar los cubos de tablas de Amazon S3 para definir un esquema de tablas personalizado. Para obtener más información sobre los cubos de tablas, consulte los cubos de tablas en la Guía del usuario de Amazon S3.

Puede usar cubos de tablas para tablas Iceberg totalmente administradas en Amazon S3.

Puede plantear un caso de soporte para solicitar al HealthOmics equipo que migre los datos de su almacén de variantes o anotaciones al depósito de tablas de Amazon S3 que haya configurado.

Una vez que los datos se hayan rellenado en el depósito de tablas de Amazon S3, puede eliminar los almacenes de variantes y los almacenes de anotaciones. Para obtener más información, consulta Eliminar almacenes HealthOmics de análisis.

Análisis

Para el análisis de datos, siga utilizando servicios de AWS análisis, como Amazon Athena, Amazon EMR, Amazon Redshift o Amazon Quick.

AWS Socios

Puede trabajar con un AWS socio que proporcione ETL personalizable, esquemas de tablas, herramientas integradas de consulta y análisis e interfaces de usuario para interactuar con los datos.

Ejemplos

Los siguientes ejemplos muestran cómo crear tablas adecuadas para almacenar datos de VCF y GVCF.

Athena DDL

Puede usar el siguiente ejemplo de DDL en Athena para crear una tabla adecuada para almacenar datos de VCF y GVCF en una sola tabla. Este ejemplo no es el equivalente exacto de la estructura de almacenamiento de variantes, pero funciona bien para un caso de uso genérico.

Crea tus propios valores para DATABASE_NAME y TABLE_NAME al crear la tabla.

```
 CREATE TABLE <DATABASE_NAME>. <TABLE_NAME> (
  sample_name string,
  variant_name string COMMENT 'The ID field in VCF files, '.' indicates no name',
  chrom string,
  pos bigint,
  ref string,
  alt array <string>,
  qual double,
  filter string,
  genotype string,
  info map <string, string>,
  attributes map <string, string>,
  is_reference_block boolean COMMENT 'Used in GVCF for non-variant sites')
PARTITIONED BY (bucket(128, sample_name), chrom)
LOCATION '{URL}/'
TBLPROPERTIES (
  'table_type'='iceberg',
  'write_compression'='zstd'
);
```

Crear tablas con Python (sin Athena)

El siguiente ejemplo de código de Python muestra cómo crear las tablas sin usar Athena.

```
 import boto3
from pyiceberg.catalog import Catalog, load_catalog
from pyiceberg.schema import Schema
from pyiceberg.table import Table
from pyiceberg.table.sorting import SortOrder, SortField, SortDirection, NullOrder
from pyiceberg.partitioning import PartitionSpec, PartitionField
from pyiceberg.transforms import IdentityTransform, BucketTransform
from pyiceberg.types import (
    NestedField,
    StringType,
    LongType,
    DoubleType,
    MapType,
    BooleanType,
    ListType
)


def load_s3_tables_catalog(bucket_arn: str) -> Catalog:
    session = boto3.session.Session()
    region = session.region_name or 'us-east-1'

    catalog_config = {
        "type": "rest",
        "warehouse": bucket_arn,
        "uri": f"https://s3tables.{region}.amazonaws.com/iceberg",
        "rest.sigv4-enabled": "true",
        "rest.signing-name": "s3tables",
        "rest.signing-region": region
    }

    return load_catalog("s3tables", **catalog_config)


def create_namespace(catalog: Catalog, namespace: str) -> None:
    try:
        catalog.create_namespace(namespace)
        print(f"Created namespace: {namespace}")
    except Exception as e:
```

```
        if "already exists" in str(e):
            print(f"Namespace {namespace} already exists.")
        else:
            raise e


def create_table(catalog: Catalog, namespace: str, table_name: str, schema: Schema,
                 partition_spec: PartitionSpec = None, sort_order: SortOrder = None) ->
 Table:
    if catalog.table_exists(f"{namespace}.{table_name}"):
        print(f"Table {namespace}.{table_name} already exists.")
        return catalog.load_table(f"{namespace}.{table_name}")

    create_table_args = {
        "identifier": f"{namespace}.{table_name}",
        "schema": schema,
        "properties": {"format-version": "2"}
    }

    if partition_spec is not None:
        create_table_args["partition_spec"] = partition_spec
    if sort_order is not None:
        create_table_args["sort_order"] = sort_order

    table = catalog.create_table(**create_table_args)
    print(f"Created table: {namespace}.{table_name}")
    return table


def main(bucket_arn: str, namespace: str, table_name: str):
    # Schema definition
    genomic_variants_schema = Schema(
        NestedField(1, "sample_name", StringType(), required=True),
        NestedField(2, "variant_name", StringType(), required=True),
        NestedField(3, "chrom", StringType(), required=True),
        NestedField(4, "pos", LongType(), required=True),
        NestedField(5, "ref", StringType(), required=True),
        NestedField(6, "alt", ListType(element_id=1000, element_type=StringType(),
 element_required=True), required=True),
        NestedField(7, "qual", DoubleType()),
        NestedField(8, "filter", StringType()),
        NestedField(9, "genotype", StringType()),
        NestedField(10, "info", MapType(key_type=StringType(), key_id=1001,
 value_type=StringType(), value_id=1002)),
```

```
        NestedField(11, "attributes", MapType(key_type=StringType(), key_id=2001,
 value_type=StringType(), value_id=2002)),
        NestedField(12, "is_reference_block", BooleanType()),
        identifier_field_ids=[1, 2, 3, 4]
    )

    # Partition and sort specifications
    partition_spec = PartitionSpec(
        PartitionField(source_id=1, field_id=1001, transform=BucketTransform(128),
 name="sample_bucket"),
        PartitionField(source_id=3, field_id=1002, transform=IdentityTransform(),
 name="chrom")
    )

    sort_order = SortOrder(
        SortField(source_id=3, transform=IdentityTransform(),
 direction=SortDirection.ASC, null_order=NullOrder.NULLS_LAST),
        SortField(source_id=4, transform=IdentityTransform(),
 direction=SortDirection.ASC, null_order=NullOrder.NULLS_LAST)
    )

    # Connect to catalog and create table
    catalog = load_s3_tables_catalog(bucket_arn)
    create_namespace(catalog, namespace)
    table = create_table(catalog, namespace, table_name, genomic_variants_schema,
 partition_spec, sort_order)

    return table


if __name__ == "__main__":
    bucket_arn = 'arn:aws:s3tables:<REGION>:<ACCOUNT_ID>:bucket/<TABLE_BUCKET_NAME'
    namespace = "variant_db"
    table_name = "genomic_variants"

    main(bucket_arn, namespace, table_name)
```

Con HealthOmicsfiguración

Para configurarlo AWS HealthOmics, inscríbase en un Cuenta de AWS, cree un usuario administrativo y gestione de forma segura el acceso de otros usuarios.

Temas

- Inscríbase en una Cuenta de AWS
- Creación de un usuario con acceso administrativo
- Cree permisos de IAM para HealthOmics
- Conéctese con repositorios de código externos
- Uso de Amazon Q CLI con HealthOmics

Inscríbase en una Cuenta de AWS

Si no tiene uno 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 AWSse 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 AWScomo 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 .

Cree permisos de IAM para HealthOmics

Para usarlos HealthOmics, configure los siguientes permisos de IAM:

- Políticas de IAM basadas en la identidad a las que puedan acceder los usuarios de su cuenta. HealthOmics
- Una función de servicio de IAM para acceder HealthOmics a los recursos en su nombre.
- Permisos en otros servicios (como Lake Formation y Amazon ECR) para que sus usuarios y el HealthOmics servicio accedan a los recursos.

Para obtener más información sobre la configuración de los permisos de IAM HealthOmics, consulte. Permisos de IAM para HealthOmics

Conéctese con repositorios de código externos

Con AWS HealthOmics, puede administrar sus flujos de trabajo mediante repositorios basados en Git mediante. AWS CodeConnections HealthOmics usa esta conexión para acceder a tus repositorios de código fuente.

Antes de trabajar con repositorios de código externos, sigue la guía de configuración de conexiones para empezar a trabajar con ellos. AWS CodeConnections Compruebe que ha creado las políticas y los permisos de IAM adecuados para su AWS cuenta. Para obtener una lista de los proveedores de Git compatibles y más información, consulta ¿Para qué proveedores externos puedo crear conexiones? .

Crea una conexión

Para crear una conexión con tu proveedor de repositorios preferido, sigue el tutorial Crear una conexión.

Uso de Amazon Q CLI con HealthOmics

Amazon Q CLI proporciona interacciones en lenguaje natural AWS HealthOmics, lo que le permite realizar flujos de trabajo genómicos complejos y tareas de análisis mediante comandos conversacionales. Para usar la CLI de Amazon Q, asegúrese de configurar los permisos de IAM HealthOmics y otros servicios (como CloudWatch Amazon ECR o Amazon S3) para que Amazon Q acceda a sus recursos.

El tutorial de IA generativa de HealthOmics Agentic proporciona una step-by-step guía para configurar los archivos de contexto y permitir que Amazon Q CLI cree, ejecute y optimice sus flujos de trabajo. AWS HealthOmics

Empezar con HealthOmics

Para empezar HealthOmics, asegúrese de haber configurado correctamente sus permisos y funciones de IAM para HealthOmics.

Uso de un flujo de trabajo de Ready2Run en la consola HealthOmics

El siguiente ejercicio muestra cómo utilizar un flujo de trabajo de Ready2Run. El flujo de trabajo de Ready2Run está preconfigurado con los parámetros y las referencias de herramientas que necesita para ejecutar el flujo de trabajo. El publicador de flujos de trabajo proporciona datos de muestra, por lo que no es necesario que cree sus propios datos.

1. Abra la consola de HealthOmics .
2. Seleccione el panel de navegación (≡) en la parte superior izquierda y seleccione los flujos de trabajo Ready2Run.
3. En la página de flujos de trabajo de Ready2Run, elija el flujo de trabajo. ESMFold for up to 800 residues La consola abre la página de detalles de ese flujo de trabajo.
4. La pestaña de detalles proporciona información sobre el flujo de trabajo. Para probar el flujo de trabajo, en la parte superior derecha de la página, selecciona Iniciar ejecución.
5. En la página Especificar los detalles de la ejecución, introduzca un nombre de ejecución.
6. Introduzca o seleccione una ubicación de Amazon S3 para el resultado de la ejecución.
7. En el modo de retención de metadatos de ejecución, elija si desea conservar o eliminar los metadatos de ejecución.
8. En el panel de funciones de servicio, selecciona Crear y usar una nueva función de servicio.
9. Elija Siguiente.
10. En la página Añadir valores de parámetros, elija Ejecutar flujo de trabajo con datos de prueba de Ready2Run.
11. Elija Siguiente.
12. Revisa las entradas y, a continuación, selecciona Iniciar ejecución.

Ejemplos de solicitudes para Amazon Q CLI

Amazon Q CLI puede ejecutar flujos de trabajo genómicos y tareas de análisis AWS HealthOmics mediante comandos de lenguaje natural. Las siguientes instrucciones de ejemplo le permiten crear flujos de trabajo, gestionar las ejecuciones y analizar los datos genómicos. Para obtener más información y ejemplos de instrucciones al respecto HealthOmics, consulte el tutorial sobre IA generativa de HealthOmics Agentic en. GitHub

- «Cree un archivo de flujo de trabajo WDL 1.1 en el que se ejecutará`main.wdl`. HealthOmics El flujo de trabajo tomará un genoma de referencia como entrada y pares de archivos fastq. Indexará el genoma de referencia mediante BWA y, a continuación, asignará cada par de archivos fastq a la referencia. Finalmente, combine cada BAM mapeado en un solo archivo BAM y genere este archivo y su índice bai».
- «Package el flujo de trabajo y créelo en HealthOmics»
- «Actualice el archivo inputs.json para usar archivos reales de mi bucket de Amazon S3`omics-my-bucket-with-genome-data`» (proporcione una ubicación específica del bucket de Amazon S3 o deje que Amazon Q explore)
- «Encuentre los contenedores adecuados en mis repositorios de Amazon ECR y actualice inputs.json para usarlos»
- «Busque o cree un rol de IAM adecuado para usarlo al ejecutar el flujo de trabajo»
- «Crear una caché de ejecución para mi flujo de trabajo»
- «Ejecute el flujo de trabajo en HealthOmics»
- «Compruebe el estado de la ejecución»

> **Warning**
>
> Cuando trabaje con Amazon Q CLI, revise todo el contenido generado y las acciones propuestas antes de continuar. Proporcione comentarios para mejorar la calidad de la respuesta y adaptarla a los requisitos de su flujo de trabajo. Para obtener más información, consulte Consideraciones de seguridad y prácticas recomendadas para Amazon Q.

Flujos de trabajo privados en HealthOmics

Utilice los flujos de trabajo privados cuando desee crear su propia definición de flujo de trabajo. La definición del flujo de trabajo especifica información sobre el flujo de trabajo y define las tareas del flujo de trabajo. Una ejecución es una única invocación de un flujo de trabajo y una tarea es un proceso único dentro de la ejecución.

HealthOmics admite las definiciones de flujo de trabajo que se crean en el lenguaje de descripción del flujo de trabajo (WDL), el lenguaje común de flujo de trabajo (CWL) o Nextflow.

HealthOmics los flujos de trabajo ofrecen las siguientes funciones opcionales:

- Run groups— Puede añadir flujos de trabajo privados a un grupo de ejecución para controlar el uso de la computación. Un grupo de ejecuciones es un conjunto de ejecuciones de flujos de trabajo que comparten un conjunto de límites de recursos, como el máximo de ejecuciones simultáneas y la duración máxima de las ejecuciones. Estos límites se establecen para controlar los recursos informáticos que consume el grupo de ejecuciones.
- Call caching— Puedes usar una caché de llamadas para guardar y reutilizar los resultados de las tareas, lo que se traduce en una menor duración de las ejecuciones y en un ahorro de costes de cálculo.
- Sharing workflows— Puedes compartir tus flujos de trabajo privados con otros usuarios Cuentas de AWS de la misma región.
- Workflow versions— Puedes crear versiones de un flujo de trabajo privado. El control de versiones del flujo de trabajo permite a los usuarios elegir cuándo empezar a utilizar la funcionalidad actualizada. Las versiones de los flujos de trabajo son inmutables y proporcionan el mismo nivel de procedencia de los datos que los flujos de trabajo.

Para obtener información sobre la configuración de los permisos de IAM para los flujos de trabajo, consulte. Permisos de IAM para HealthOmics

Para ver ejemplos completos de cómo usar flujos de trabajo HealthOmics privados, consulte los tutoriales de HealthOmics Github o el tutorial integral del taller de AWS para HealthOmics.

Temas

- Crear flujos de trabajo privados en HealthOmics
- Control de versiones del flujo de trabajo en HealthOmics

- Uso de HealthOmics corridas
- Uso de grupos de HealthOmics ejecución
- Almacenamiento en caché de llamadas para ejecuciones HealthOmics
- Compartir HealthOmics flujos de trabajo

Crear flujos de trabajo privados en HealthOmics

Los flujos de trabajo privados dependen de una variedad de recursos que se crean y configuran antes de crear el flujo de trabajo:

- Workflow definition file:Un archivo de definición de flujo de trabajo escrito en WDLNextflow, oCWL. La definición del flujo de trabajo especifica las entradas y salidas de las ejecuciones que utilizan el flujo de trabajo. También incluye especificaciones para las ejecuciones y tareas de ejecución del flujo de trabajo, incluidos los requisitos de procesamiento y memoria. El archivo de definición del flujo de trabajo debe estar en `.zip` formato. Para obtener más información, consulte Archivos de definición de flujos de trabajo.
  - Puede usar Amazon Q CLI para crear y validar sus archivos de definición de flujo de trabajo en WDL, Nextflow y CWL. Para obtener más información, consulte Ejemplos de instrucciones para Amazon Q CLI y el tutorial sobre HealthOmics IA generativa de Agentic en. GitHub
- (Optional) Parameter template file:Un archivo de plantilla de parámetros escrito en. JSON Cree el archivo para definir los parámetros de ejecución o HealthOmics genere la plantilla de parámetros automáticamente. Para obtener más información, consulte Archivos de plantillas de parámetros para HealthOmics flujos de trabajo.
- Amazon ECR container images:Cree un repositorio privado de Amazon ECR para el flujo de trabajo. Cree imágenes de contenedores en el repositorio privado o sincronice el contenido de un registro upstream compatible con su repositorio privado de Amazon ECR.
- (Optional) Sentieon licenses:Solicite una Sentieon licencia para usar el Sentieon software en flujos de trabajo privados.

Si lo desea, puede ejecutar un resumen de la definición del flujo de trabajo antes o después de crearlo. En linter este tema se describen las linters disponibles en. HealthOmics

Temas

- HealthOmics integración del flujo de trabajo con repositorios basados en Git
- Archivos de definición de flujos de trabajo en HealthOmics

- Archivos de plantillas de parámetros para HealthOmics flujos de trabajo
- Imágenes de contenedores para flujos de trabajo privados
- HealthOmics Archivos README de flujo de trabajo
- Solicitud de licencias de Sentieon para flujos de trabajo privados
- Lentes de flujo de trabajo en HealthOmics
- HealthOmics operaciones de flujo de trabajo

HealthOmics integración del flujo de trabajo con repositorios basados en Git

Al crear un flujo de trabajo (o una versión del flujo de trabajo), se proporciona una definición del flujo de trabajo para especificar información sobre el flujo de trabajo, las ejecuciones y las tareas. HealthOmics puede recuperar la definición del flujo de trabajo como un archivo.zip (almacenado localmente o en un bucket de Amazon S3) o desde un repositorio compatible basado en Git.

La HealthOmics integración con los repositorios basados en Git permite las siguientes capacidades:

- Creación directa de flujos de trabajo a partir de instancias públicas, privadas y autogestionadas.
- Integración de los archivos README del flujo de trabajo y las plantillas de parámetros de los repositorios.
- Support para GitHub GitLab, y repositorios de Bitbucket.

Al utilizar un repositorio basado en Git, se evitan los pasos manuales de descargar archivos de definición de flujo de trabajo y archivos de plantillas de parámetros de entrada, crear un archivo.zip y, a continuación, almacenar el archivo en S3. Esto simplifica la creación de flujos de trabajo para escenarios como los siguientes:

1. Desea empezar rápidamente a utilizar un flujo de trabajo común de código abierto, como nf-core. HealthOmicsrecupera automáticamente todos los archivos de plantillas de parámetros de entrada y definición del flujo de trabajo del repositorio de nf-core GitHub y los utiliza para crear su nuevo flujo de trabajo.
2. Está utilizando un flujo de trabajo público de GitHub, y estarán disponibles algunas actualizaciones nuevas. Puede crear fácilmente una nueva versión del HealthOmics flujo de trabajo utilizando la definición de flujo de trabajo actualizada GitHub como fuente. Los usuarios del flujo de trabajo pueden elegir entre el flujo de trabajo original o la nueva versión del flujo de trabajo que haya creado.

3. Tu equipo está creando una canalización propia que no es pública. Guardas tu código en un repositorio de git privado y utilizas esta definición de flujo de trabajo para tus HealthOmics flujos de trabajo. El equipo actualiza la definición del flujo de trabajo con frecuencia como parte de un ciclo de vida de desarrollo del flujo de trabajo iterativo. Puedes crear fácilmente nuevas versiones del flujo de trabajo según sea necesario desde tu repositorio privado.

Temas

- Repositorios basados en Git compatibles
- Configure las conexiones a repositorios de código externos
- Acceder a repositorios autogestionados
- Cuotas relacionadas con los repositorios de código externos
- Permisos de IAM necesarios

Repositorios basados en Git compatibles

HealthOmics admite repositorios públicos y privados para los siguientes proveedores basados en Git:

- GitHub
- GitLab
- Bitbucket

HealthOmics admite repositorios autogestionados para los siguientes proveedores basados en Git:

- GitHubEnterpriseServer
- GitLabSelfManaged

HealthOmics admite el uso de conexiones entre cuentas para GitHub, GitLab y Bitbucket. Configure los permisos compartidos a través de AWS Resource Access Manager. Para ver un ejemplo, consulte Conexiones compartidas en la guía del CodePipeline usuario.

Configure las conexiones a repositorios de código externos

Conecte sus flujos de trabajo a repositorios basados en Git mediante AWS. CodeConnection HealthOmics utiliza esta conexión para acceder a sus repositorios de código fuente.

> **Note**
>
> El CodeConnections servicio de AWS no está disponible en la región il-central-1. Para esta región, configure el servicio us-east-1 para crear flujos de trabajo o versiones de flujos de trabajo desde un repositorio.

Cree una conexión de

Antes de poder crear conexiones, siga las instrucciones de Configuración de las conexiones de la Guía del usuario de las herramientas de la consola para desarrolladores.

Para crear una conexión, siga las instrucciones de la Guía del usuario de las herramientas de la consola para desarrolladores.

Configure la autorización para la conexión

Debe autorizar la conexión mediante el OAuth flujo del proveedor. Asegúrese de que el estado de la conexión es `AVAILABLE` antes de usarla.

Para ver ejemplos, consulta la entrada del blog Cómo crear AWS HealthOmics flujos de trabajo a partir del contenido de Git.

Acceder a repositorios autogestionados

Para configurar las conexiones a un repositorio GitLab autogestionado, utiliza un token de acceso personal de administrador al crear un host. Al crear la conexión subsiguiente, se accede a Oauth con la cuenta del cliente.

El siguiente ejemplo configura una conexión a un repositorio autogestionado: GitLab

1. Configure el acceso al token de acceso personal de un usuario administrador.

   Para configurar una PAT en un repositorio GitLab autogestionado, consulta los tokens de acceso personal en GitLab Docs.
2. Creación de un alojamiento

   a. Ve a CodePipeline>Configuración>Conexiones.
   b. Seleccione la pestaña Anfitriones y, a continuación, elija Crear anfitrión.
   c. Configure los siguientes campos:

- Introduzca el nombre del anfitrión
- Para el tipo de proveedor, elija GitLab Autogestionado
- Introduzca la URL del host
- Introduzca la información de la VPC si el host está definido en una VPC

d. Elija Crear host, que crea el host en estado PENDIENTE.

e. Para completar la configuración, elija Configurar anfitrión.

f. Introduzca el token de acceso personal (PAT) de un usuario administrador y, a continuación, seleccione Continuar.

3. Creación de la conexión

a. Seleccione Crear conexiones en la pestaña Conexiones.

b. Para el tipo de proveedor, seleccione GitLab autogestionado.

c. En Configuración de conexión > Introducir nombre de conexión, introduzca la URL del host que creó anteriormente.

d. Si solo se puede acceder a la instancia GitLab autogestionada a través de una VPC, configure los detalles de la VPC.

e. Elija Actualizar conexión pendiente. La ventana modal lo redirige a la página de GitLab inicio de sesión.

f. Introduzca el nombre de usuario y la contraseña de la cuenta del cliente y complete el proceso de autorización.

g. Para configurarla por primera vez, selecciona Autorizar AWS Connector para Gitlab Self Managed.

Cuotas relacionadas con los repositorios de código externos

Para HealthOmics la integración con repositorios de código externos, hay un tamaño máximo para un repositorio, cada archivo del repositorio y cada archivo README. Para obtener más información, consulte HealthOmics cuotas de tamaño fijas en el flujo de trabajo.

Permisos de IAM necesarios

Añada las siguientes acciones a su política de IAM basada en la identidad:

```
"codeconnections:CreateConnection",
"codeconnections:GetConnection",
```

```
    "codeconnections:GetHost",
    "codeconnections:ListConnections",
    "codeconnections:UseConnection"
```

Archivos de definición de flujos de trabajo en HealthOmics

Utilice una definición de flujo de trabajo para especificar información sobre el flujo de trabajo, las ejecuciones y las tareas de las ejecuciones. Las definiciones de flujo de trabajo se crean en uno o más archivos mediante un lenguaje de definición de flujos de trabajo. HealthOmics admite definiciones de flujo de trabajo escritas en WDL, Nextflow o CWL.

HealthOmics admite las siguientes opciones para las definiciones de flujos de trabajo de WDL:

- WDL: proporciona un motor WDL que cumple con las especificaciones.
- WDL indulgente: diseñado para gestionar los flujos de trabajo migrados desde Cromwell. Es compatible con las directivas de Cromwell de los clientes y con algunas lógicas no conformes. Para obtener más información, consulte Conversión de tipos implícita en WDL lenient.

Para obtener información sobre cada uno de los lenguajes del flujo de trabajo, consulte las secciones detalladas específicas del idioma que aparecen a continuación.

Los siguientes tipos de información se especifican en la definición del flujo de trabajo:

- Language version— El idioma y la versión de la definición del flujo de trabajo.
- Compute and memory— Los requisitos de cómputo y memoria para las tareas del flujo de trabajo.
- Inputs— Ubicación de las entradas a las tareas del flujo de trabajo. Para obtener más información, consulte HealthOmics ejecutar entradas.
- Outputs— Ubicación para guardar las salidas que generan las tareas.
- Task resources— Requisitos de cómputo y memoria para cada tarea.
- Accelerators— otros recursos que requieran las tareas, como los aceleradores.

Temas

- HealthOmics requisitos de definición del flujo de trabajo
- Soporte de versiones para lenguajes de definición HealthOmics de flujos de trabajo
- Requisitos de cómputo y memoria para HealthOmics las tareas
- Resultados de tareas en una definición de HealthOmics flujo de trabajo

- Recursos de tareas en una definición de HealthOmics flujo de trabajo
- Aceleradores de tareas en una definición de flujo de trabajo HealthOmics
- Especificaciones de la definición del flujo de trabajo de WDL
- Especificaciones de la definición del flujo de trabajo de Nextflow
- Especificaciones de la definición del flujo de trabajo de CWL
- Ejemplos de definiciones de flujo de trabajo

HealthOmics requisitos de definición del flujo de trabajo

Los archivos HealthOmics de definición del flujo de trabajo deben cumplir los siguientes requisitos:

- Las tareas deben definir input/output los parámetros, los repositorios de contenedores de Amazon ECR y las especificaciones de tiempo de ejecución, como la asignación de memoria o CPU.
- Compruebe que sus funciones de IAM tienen los permisos necesarios.
  - Su flujo de trabajo tiene acceso a los datos de entrada de AWS recursos, como Amazon S3.
  - Su flujo de trabajo tiene acceso a los servicios de repositorio externos cuando los necesita.
- Declare los archivos de salida en la definición del flujo de trabajo. Para copiar los archivos de ejecución intermedia a la ubicación de salida, declare que son salidas del flujo de trabajo.
- Las ubicaciones de entrada y salida deben estar en la misma región que el flujo de trabajo.
- HealthOmics Las entradas del flujo de trabajo de almacenamiento deben estar en `ACTIVE` estado. HealthOmics no importará las entradas con un `ARCHIVED` estado, lo que provocará un error en el flujo de trabajo. Para obtener información sobre las entradas de objetos de Amazon S3, consulteHealthOmics ejecutar entradas.
- La main ubicación del flujo de trabajo es opcional si el archivo ZIP contiene una única definición de flujo de trabajo o un archivo denominado «principal».
  - Ruta de ejemplo: `workflow-definition/main-file.wdl`
- Antes de crear un flujo de trabajo desde Amazon S3 o desde su unidad local, cree un archivo zip con los archivos de definición del flujo de trabajo y cualquier dependencia, como los subflujos de trabajo.
- Le recomendamos que declare los contenedores de Amazon ECR en el flujo de trabajo como parámetros de entrada para la validación de los permisos de Amazon ECR.

Consideraciones adicionales sobre Nextflow:

- /bin

  Las definiciones del flujo de trabajo de Nextflow pueden incluir una carpeta /bin con scripts ejecutables. Esta ruta tiene acceso a las tareas de solo lectura y a los ejecutables. Las tareas que se basan en estos scripts deben usar un contenedor creado con los intérpretes de scripts adecuados. La mejor práctica es llamar directamente al intérprete. Por ejemplo:

```
process my_bin_task {
   ...
   script:
      """
      python3 my_python_script.py
      """
}
```

- includeConfig

  Las definiciones de flujo de trabajo basadas en NextFlow pueden incluir archivos nextflow.config que ayudan a abstraer las definiciones de parámetros o procesar los perfiles de recursos. Para facilitar el desarrollo y la ejecución de canalizaciones de Nextflow en varios entornos, utilice una configuración HealthOmics específica que añada a la configuración global mediante la directiva IncludeConfig. Para mantener la portabilidad, configure el flujo de trabajo para que incluya el archivo solo cuando se esté ejecutando mediante el HealthOmics siguiente código:

```
// at the end of the nextflow.config file
if ("$AWS_WORKFLOW_RUN") {
    includeConfig 'conf/omics.config'
}
```

- Reports

  HealthOmics no admite los informes de arrastre, rastreo y ejecución generados por el motor. Puede generar alternativas a los informes de seguimiento y ejecución mediante una combinación de GetRun llamadas a la GetRunTask API.

Consideraciones adicionales sobre la CWL:

- Container image uri interpolation

HealthOmics permite que la propiedad DockerPull del sea una expresión DockerRequirement de javascript en línea. Por ejemplo:

```
requirements:
  DockerRequirement:
    dockerPull: "$(inputs.container_image)"
```

Esto le permite especificar la imagen del contenedor URIs como parámetros de entrada al flujo de trabajo.

- Javascript expressions

  Las expresiones de JavaScript deben ser `strict mode` compatibles.

- Operation process

  HealthOmics no es compatible con los procesos de operación de CWL.

Soporte de versiones para lenguajes de definición HealthOmics de flujos de trabajo

HealthOmics admite archivos de definición de flujos de trabajo escritos en Nextflow, WDL o CWL. En las siguientes secciones se proporciona información sobre la compatibilidad de las HealthOmics versiones con estos idiomas.

Temas

- Compatibilidad con las versiones WDL
- Compatibilidad con la versión CWL
- Compatibilidad con la versión de Nextflow

Compatibilidad con las versiones WDL

HealthOmics admite las versiones 1.0 y 1.1 y la versión de desarrollo de la especificación WDL.

Cada documento de la WDL debe incluir una declaración de versión para especificar la versión (principal y secundaria) de la especificación a la que se adhiere. Para obtener más información sobre las versiones, consulte Control de versiones de la WDL

Las versiones 1.0 y 1.1 de la especificación WDL no admiten este tipo. `Directory` Para usar el `Directory` tipo para entradas o salidas, defina la versión development en la primera línea del archivo:

```
version development  # first line of .wdl file
     ... remainder of the file ...
```

Compatibilidad con la versión CWL

HealthOmics es compatible con las versiones 1.0, 1.1 y 1.2 del lenguaje CWL.

Puede especificar la versión de idioma en el archivo de definición del flujo de trabajo de CWL. Para obtener más información sobre CWL, consulte la guía del usuario de CWL

Compatibilidad con la versión de Nextflow

HealthOmics admite tres versiones estables de Nextflow. Nextflow suele lanzar una versión estable cada seis meses. HealthOmics no es compatible con las versiones «edge» mensuales.

HealthOmics admite las funciones publicadas en cada versión, pero no las funciones de vista previa.

Versiones compatibles

HealthOmics es compatible con las siguientes versiones de Nextflow:

- Nextflow v22.04.01 DSL 1 y DSL 2
- Nextflow v23.10.0 DSL 2 (predeterminado)
- Nextflow v24.10.8 DSL 2

Para migrar su flujo de trabajo a la última versión compatible (v24.10.8), siga la guía de actualización de Nextflow.

Hay algunos cambios importantes al migrar de la versión 23 a la 24 de Nextflow, tal y como se describe en las siguientes secciones de la guía de migración de Nextflow:

- Cambios importantes en la versión 24.04
- Cambios importantes en la versión 24.10

Detecta y procesa las versiones de Nextflow

HealthOmics detecta la versión de DSL y la versión de Nextflow que especifique. Determina automáticamente la mejor versión de Nextflow para ejecutar en función de estas entradas.

Versión DSL

HealthOmics detecta la versión DSL solicitada en el archivo de definición del flujo de trabajo. Por ejemplo, puede especificar:`nextflow.enable.dsl=2`.

HealthOmics es compatible con DSL 2 de forma predeterminada. Proporciona compatibilidad con versiones anteriores de DSL 1, si se especifica en el archivo de definición del flujo de trabajo.

- Si especifica DSL 2, HealthOmics ejecuta Nextflow v23.10.0, a menos que especifique Nextflow v22.04.0 o v24.10.8.
- Si especifica DSL 1, ejecuta Nextflow v22.04 (la única versión compatible que HealthOmics ejecuta DSL 1). DSL1
- Si no especificas una versión de DSL o si no HealthOmics puedes analizar la información de DSL por algún motivo (por ejemplo, por errores de sintaxis en el archivo de definición del flujo de trabajo), el HealthOmics valor predeterminado es DSL 2 y ejecuta Nextflow v23.10.0.
- Para actualizar su flujo de trabajo de DSL 1 a DSL 2 y aprovechar las últimas versiones y funciones del software de Nextflow, consulte Migración desde DSL 1.

Versiones de Nextflow

HealthOmics detecta la versión de Nextflow solicitada en el archivo de configuración de Nextflow (nextflow.config), si usted proporciona este archivo. Le recomendamos que añada la `nextflowVersion` cláusula al final del archivo para evitar anulaciones inesperadas de las configuraciones incluidas. Para obtener más información, consulte Configuración de Nextflow.

Puede especificar una versión o un rango de versiones de Nextflow mediante la siguiente sintaxis:

```
// exact match
manifest.nextflowVersion = '1.2.3'

// 1.2 or later (excluding 2 and later)
manifest.nextflowVersion = '1.2+'

// 1.2 or later
manifest.nextflowVersion = '>=1.2'
```

```
// any version in the range 1.2 to 1.5
manifest.nextflowVersion = '>=1.2, <=1.5'

// use the "!" prefix to stop execution if the current version
// doesn't match the required version.
manifest.nextflowVersion = '!>=1.2'
```

HealthOmics procesa la información de la versión de Nextflow de la siguiente manera:

- Si especifica una versión exacta que HealthOmics sea compatible, HealthOmics utilizará = esa versión.
- Si se utiliza ! para especificar una versión exacta o un rango de versiones que no son compatibles, se HealthOmics genera una excepción y no se ejecuta correctamente. Considere usar esta opción si quiere ser estricto con las solicitudes de versión y fallar rápidamente si la solicitud incluye versiones no compatibles.
- Si especifica un rango de versiones, HealthOmics usa la última versión compatible de ese rango, a menos que el rango incluya la versión 24.10.8. En este caso, HealthOmics da preferencia a una versión anterior. Por ejemplo, si el rango abarca tanto la versión 23.10.0 como la 24.10.8, elige la versión 23.10.0. HealthOmics
- Si no hay ninguna versión solicitada, o si las versiones solicitadas no son válidas o no se pueden analizar por algún motivo:
  - Si especificó DSL 1, HealthOmics ejecute Nextflow v22.04.
  - De lo contrario, ejecuta Nextflow v23.10.0 HealthOmics .

Puede recuperar la siguiente información sobre la versión de Nextflow que HealthOmics se utilizó en cada ejecución:

- Los registros de ejecución contienen información sobre la versión real de Nextflow que HealthOmics se utilizó para la ejecución.
- HealthOmics agrega advertencias en los registros de ejecución si no hay una coincidencia directa con la versión solicitada o si necesita usar una versión diferente a la que especificó.
- La respuesta a la operación de la GetRun API incluye un campo (`engineVersion`) con la versión real de Nextflow que HealthOmics se utilizó para la ejecución. Por ejemplo:

```
"engineVersion":"22.04.0"
```

Requisitos de cómputo y memoria para HealthOmics las tareas

HealthOmics ejecuta tus tareas de flujo de trabajo privadas en una instancia ómica. HealthOmics proporciona varios tipos de instancias para adaptarse a distintos tipos de tareas. Cada tipo de instancia tiene una configuración fija de memoria y vCPU (y una configuración de GPU fija para los tipos de instancias de computación acelerada). El costo de usar una instancia ómica varía según el tipo de instancia. Para obtener más información, consulta la página HealthOmics de precios.

Para las tareas de un flujo de trabajo, especifique la memoria necesaria y v CPUs en el archivo de definición del flujo de trabajo. Cuando se ejecuta una tarea de flujo de trabajo, HealthOmics asigna la instancia ómica más pequeña que aloje la memoria solicitada y v. CPUs Por ejemplo, si una tarea necesita 64 GiB de memoria y 8 vCPUs, HealthOmics selecciona. `omics.r.2xlarge`

Te recomendamos que revises los tipos de instancias y establezcas la v CPUs y el tamaño de memoria solicitados para que coincidan con la instancia que mejor se adapte a tus necesidades. El contenedor de tareas utiliza el número de v CPUs y el tamaño de memoria que especifiques en el archivo de definición del flujo de trabajo, incluso si el tipo de instancia tiene v CPUs y memoria adicionales.

La siguiente lista contiene información adicional sobre la asignación de vCPU y memoria:

- Las asignaciones de recursos de contenedores son límites estrictos. Si una tarea se queda sin memoria o intenta utilizar una v adicionalCPUs , la tarea genera un registro de errores y se cierra.
- Si no especificas ningún requisito de procesamiento o memoria, HealthOmics selecciona una configuración con 1 vCPU omics.c.large y 1 GiB de memoria, y la establece de forma predeterminada.
- La configuración mínima que puede solicitar es de 1 vCPU y 1 GiB de memoria.
- Si especificas vCPUs, memoria o GPUs que supere los tipos de instancia admitidos, HealthOmics se mostrará un mensaje de error y el flujo de trabajo no pasará las validaciones
- Si especificas unidades fraccionarias, HealthOmics redondea al entero más cercano.
- HealthOmics reserva una pequeña cantidad de memoria (un 5%) para los agentes de administración y registro, por lo que es posible que la aplicación de la tarea no disponga siempre de toda la memoria asignada.
- HealthOmics hace coincidir los tipos de instancias para adaptarse a los requisitos de procesamiento y memoria que especifiques y puede usar una combinación de generaciones de hardware. Por este motivo, puede haber pequeñas variaciones en los tiempos de ejecución de una misma tarea.

En estos temas se proporcionan detalles sobre los tipos de instancias HealthOmics compatibles.

Temas

- Tipos de instancias estándar
- Instancias optimizadas para computación
- Instancias optimizadas para memoria
- Instancias de computación acelerada

> **Note**
>
> En el caso de las instancias estándar, optimizadas para cómputo y memoria, aumenta el tamaño del ancho de banda de la instancia si la instancia requiere un mayor rendimiento. Las instancias Amazon EC2 con menos de 16 vCPU (tamaño 4xl o menor) pueden experimentar un aumento de rendimiento. Para obtener más información sobre el rendimiento de las instancias de Amazon EC2, consulte el ancho de banda disponible de las instancias de Amazon EC2.

Tipos de instancias estándar

Para los tipos de instancias estándar, las configuraciones buscan un equilibrio entre la potencia de procesamiento y la memoria.

HealthOmics admite las instancias de 32 y 48 tamaños en estas regiones: EE.UU. Oeste (Oregón) y EE.UU. Este (Norte de Virginia).

Instancia	Número de v CPUs	Memoria
omics.m.large	2	8 GiB
omics.m.xlarge	4	16 GiB
omics.m.2xlarge	8	32 GiB
omics.m.4xlarge	16	64 GiB
omics.m. 8 x grande	32	128 GiB

Instancia	Número de v CPUs	Memoria
omics.m. 12 x grande	48	192 GiB
omics.m. 16 x grande	64	256 GiB
omics.m. 24 x grande	96	384 GiB
omics.m. 32 x grande	128	512 GiB
omics.m. 48 x grande	192	768 GiB

Instancias optimizadas para computación

En el caso de los tipos de instancias optimizadas para la informática, las configuraciones tienen más potencia informática y menos memoria.

HealthOmics admite las instancias de 32 y 48 tamaños en estas regiones: EE.UU. Oeste (Oregón) y EE.UU. Este (Norte de Virginia).

Instancia	Número de v CPUs	Memoria
omics.c.large	2	4 GiB
omics.c.xlarge	4	8 GiB
omics.c.2xlarge	8	16 GiB
omics.c.4 x grande	16	32 GiB
omics.c.8 x grande	32	64 GiB
omics.c.12 x grande	48	96 GiB
omics.c.16 x grande	64	128 GiB
omics.c.24 x grande	96	192 GiB
omics.c.32xgrande	128	256 GiB

Instancia	Número de v CPUs	Memoria
omics.c.48 x grande	192	384 GiB

Instancias optimizadas para memoria

En el caso de los tipos de instancias optimizadas para la memoria, las configuraciones tienen menos potencia de procesamiento y más memoria.

HealthOmics admite las instancias de 32 y 48 tamaños en estas regiones: EE.UU. Oeste (Oregón) y EE.UU. Este (Norte de Virginia).

Instancia	Número de v CPUs	Memoria
omics.r.large	2	16 GiB
omics.r.xlarge	4	32 GiB
omics.r.2xlarge	8	64 GiB
omics.r.4xlarge	16	128 GiB
omics.r. 8 x grande	32	256 GiB
omics.r. 12 x grande	48	384 GiB
omics.r. 16 x grande	64	512 GiB
omics.r. 24 x grande	96	768 GiB
omics.r. 32 x grande	128	1024 GiB
omics.r. 48 veces más grande	192	1536 GiB

Instancias de computación acelerada

Si lo desea, puede especificar los recursos de la GPU para cada tarea de un flujo de trabajo, de modo que se HealthOmics asigne una instancia de computación acelerada a la tarea. Para obtener

información sobre cómo especificar la información de la GPU en el archivo de definición del flujo de trabajo, consulte. Aceleradores de tareas en una definición de flujo de trabajo HealthOmics

Si especificas un acelerador de tareas que admite varios tipos de instancias, HealthOmics selecciona el tipo de instancia en función de la disponibilidad. Si hay más de un tipo de instancia disponible, HealthOmics da preferencia a la instancia de menor coste. La excepción es el acelerador de tareas nvidia-t4-a10g-l4, que da preferencia a la instancia de última generación disponible en su región.

Las instancias G4 no son compatibles en la región de Israel (Tel Aviv). Las instancias G5 no son compatibles en la región de Asia Pacífico (Singapur).

Temas

- Tipos de instancias G6 y G6e
- Instancias G4 y G5

Tipos de instancias G6 y G6e

HealthOmics admite las siguientes configuraciones de instancias de computación acelerada G6. Todas las instancias de omics.g6 utilizan Nvidia L4. GPUs

HealthOmics admite las instancias G6 y G6e en las siguientes regiones: EE.UU. Oeste (Oregón) y EE.UU. Este (Norte de Virginia).

Instancia	Número de v CPUs	Memoria	Número de GPUs	Memoria de GPU	
omics.g6.xlarge	4	16 GiB	1	24 GiB	
omics.g6.2xgrande	8	32 GiB	1	24 GiB	
omics.g6.4 x grande	16	64 GiB	1	24 GiB	
omics.g6.8 x grande	32	128 GiB	1	24 GiB	

Instancia	Número de v CPUs	Memoria	Número de GPUs	Memoria de GPU	
omics.g6.12x grande	48	192 GiB	4	96 GiB	
omics.g6.16x grande	64	256 GiB	1	24 GiB	
omics.g6.24x grande	96	384 GiB	4	96 GiB	

Todas las instancias de omics.g6e utilizan Nvidia L40s. GPUs

Instancia	Número de v CPUs	Memoria	Número de GPUs	Memoria de GPU	
omics.g6e .xlarge	4	32 GiB	1	48 GiB	
omics.g6e .2xlarge	8	64 GiB	1	48 GiB	
omics.g6e .4xgrande	16	128 GiB	1	48 GiB	
omics.g6e.8x grande	32	256 GiB	1	48 GiB	
omics.g6e .12x grande	48	384 GiB	4	192 GiB	
omics.g6e .16x grande	64	512 GiB	1	48 GiB	
omics.g6e .24x grande	96	768 GiB	4	192 GiB	

Instancias G4 y G5

HealthOmics admite las siguientes configuraciones de instancias de computación acelerada G4 y G5.

Todas las instancias de omics.g5 utilizan Nvidia Tesla A10G. GPUs

Instancia	Número de v CPUs	Memoria	Número de GPUs	Memoria de GPU	
omics.g5.xlarge	4	16 GiB	1	24 GiB	
omics.g5.2xlarge	8	32 GiB	1	24 GiB	
omics.g5.4 x grande	16	64 GiB	1	24 GiB	
omics.g5.8 x grande	32	128 GiB	1	24 GiB	
omics.g5.12x grande	48	192 GiB	4	96 GiB	
omics.g5.16x grande	64	256 GiB	1	24 GiB	
omics.g5.24x grande	96	384 GiB	4	96 GiB	

Todas las instancias de omics.g4dn utilizan Nvidia Tesla T4. GPUs

Instancia	Número de v CPUs	Memoria	Número de GPUs	Memoria de GPU	
omics.g4dn.xlarge	4	16 GiB	1	16 GiB	

Instancia	Número de v CPUs	Memoria	Número de GPUs	Memoria de GPU	
omics.g4dn.2xlarge	8	32 GiB	1	16 GiB	
omics.g4dn.4xgrande	16	64 GiB	1	16 GiB	
omics.g4dn.8xlarge	32	128 GiB	1	16 GiB	
omics.g4dn.12 x grande	48	192 GiB	4	64 GiB	
omics.g4dn.16 x grande	64	256 GiB	1	24 GiB	

Resultados de tareas en una definición de HealthOmics flujo de trabajo

Los resultados de las tareas se especifican en la definición del flujo de trabajo. De forma predeterminada, HealthOmics descarta todos los archivos de tareas intermedias cuando se completa el flujo de trabajo. Para exportar un archivo intermedio, debe definirlo como una salida.

Si utiliza el almacenamiento en caché de llamadas, HealthOmics guarda los resultados de las tareas en la memoria caché, incluidos los archivos intermedios que defina como resultados.

Los siguientes temas incluyen ejemplos de definiciones de tareas para cada uno de los lenguajes de definición de flujos de trabajo.

Temas

- Resultados de tareas para WDL
- Resultados de tareas para Nextflow
- Resultados de tareas para CWL

Resultados de tareas para WDL

Para las definiciones de flujo de trabajo escritas en WDL, defina los resultados en la sección de flujo outputs de trabajo de nivel superior.

HealthOmics

Temas

- Resultado de la tarea para STDOUT
- Resultado de la tarea para STDERR
- Salida de la tarea a un archivo
- Salida de la tarea a una matriz de archivos

Resultado de la tarea para STDOUT

En este ejemplo, se crea una tarea con un nombre `SayHello` que refleja el contenido de STDOUT en el archivo de resultados de la tarea. La stdout función WDL captura el contenido de STDOUT (en este ejemplo, la cadena de entrada Hello World!) en un archivo. stdout_file

Como HealthOmics crea registros para todo el contenido de STDOUT, el resultado también aparece en los CloudWatch registros, junto con otra información de registro de STDERR para la tarea.

```
version 1.0
 workflow HelloWorld {
    input {
        String message = "Hello, World!"
        String ubuntu_container = "123456789012.dkr.ecr.us-east-1.amazonaws.com/
dockerhub/library/ubuntu:20.04"
    }

    call SayHello {
        input:
            message = message,
            container = ubuntu_container
    }

    output {
        File stdout_file = SayHello.stdout_file
    }
}
```

```
task SayHello {
    input {
        String message
        String container
    }

    command <<<
        echo "~{message}"
        echo "Current date: $(date)"
        echo "This message was printed to STDOUT"
    >>>

    runtime {
        docker: container
        cpu: 1
        memory: "2 GB"
    }

    output {
        File stdout_file = stdout()
    }
}
```

Resultado de la tarea para STDERR

En este ejemplo, se crea una tarea con un nombre `SayHello` que refleja el contenido de STDERR en el archivo de resultados de la tarea. La stderr función WDL captura el contenido STDERR (en este ejemplo, la cadena de entrada Hello World!) en un archivo. stderr_file

Como HealthOmics crea registros para todo el contenido de STDERR, el resultado aparecerá en CloudWatch Registros junto con otra información de registro de STDERR correspondiente a la tarea.

```
version 1.0
 workflow HelloWorld {
    input {
        String message = "Hello, World!"
        String ubuntu_container = "123456789012.dkr.ecr.us-east-1.amazonaws.com/
dockerhub/library/ubuntu:20.04"
    }

    call SayHello {
        input:
            message = message,
```

```
            container = ubuntu_container
    }

    output {
        File stderr_file = SayHello.stderr_file
    }
}

task SayHello {
    input {
        String message
        String container
    }

    command <<<
        echo "~{message}" >&2
        echo "Current date: $(date)" >&2
        echo "This message was printed to STDERR" >&2
    >>>

    runtime {
        docker: container
        cpu: 1
        memory: "2 GB"
    }

    output {
        File stderr_file = stderr()
    }
}
```

Salida de la tarea a un archivo

En este ejemplo, la SayHello tarea crea dos archivos (message.txt y info.txt) y los declara explícitamente como las salidas con nombre (message_file e info_file).

```
version 1.0
workflow HelloWorld {
    input {
        String message = "Hello, World!"
        String ubuntu_container = "123456789012.dkr.ecr.us-east-1.amazonaws.com/
dockerhub/library/ubuntu:20.04"
    }
```

```
    call SayHello {
        input:
            message = message,
            container = ubuntu_container
    }

    output {
        File message_file = SayHello.message_file
        File info_file = SayHello.info_file
    }
}

task SayHello {
    input {
        String message
        String container
    }

    command <<<
        # Create message file
        echo "~{message}" > message.txt

        # Create info file with date and additional information
        echo "Current date: $(date)" > info.txt
        echo "This message was saved to a file" >> info.txt
    >>>

    runtime {
        docker: container
        cpu: 1
        memory: "2 GB"
    }

    output {
        File message_file = "message.txt"
        File info_file = "info.txt"
    }
}
```

Salida de la tarea a una matriz de archivos

En este ejemplo, la `GenerateGreetings` tarea genera una matriz de archivos como resultado de la tarea. La tarea genera dinámicamente un archivo de saludo para cada miembro de la matriz de entrada`names`. Como los nombres de los archivos no se conocen hasta el momento de ejecución, la definición de salida utiliza la función WDL glob () para generar todos los archivos que coincidan con el patrón. `*_greeting.txt`

```
version 1.0
 workflow HelloArray {
    input {
        Array[String] names = ["World", "Friend", "Developer"]
        String ubuntu_container = "123456789012.dkr.ecr.us-east-1.amazonaws.com/
dockerhub/library/ubuntu:20.04"
    }

    call GenerateGreetings {
        input:
            names = names,
            container = ubuntu_container
    }

    output {
        Array[File] greeting_files = GenerateGreetings.greeting_files
    }
}

task GenerateGreetings {
    input {
        Array[String] names
        String container
    }

    command  <<<
        # Create a greeting file for each name
        for name in ~{sep=" " names}; do
            echo "Hello, $name!" > ${name}_greeting.txt
        done
    >>>

    runtime {
        docker: container
        cpu: 1
```

```
        memory: "2 GB"
    }

    output {
        Array[File] greeting_files = glob("*_greeting.txt")
    }
}
```

Resultados de tareas para Nextflow

Para las definiciones de flujo de trabajo escritas en Nextflow, defina una directiva PublishDir para exportar el contenido de las tareas a su bucket de Amazon S3 de salida. Establezca el valor PublishDir en. `/mnt/workflow/pubdir`

HealthOmics Para poder exportar archivos a Amazon S3, los archivos deben estar en este directorio.

Si una tarea genera un grupo de archivos de salida para usarlos como entradas en una tarea posterior, le recomendamos que agrupe estos archivos en un directorio y emita el directorio como resultado de la tarea. La enumeración de cada archivo individual puede provocar un cuello de botella de E/S en el sistema de archivos subyacente. Por ejemplo:

```
process my_task {
      ...
      // recommended
      output "output-folder/", emit: output

      // not recommended
      // output "output-folder/**", emit: output
      ...
  }
```

Resultados de tareas para CWL

Para las definiciones de flujo de trabajo escritas en CWL, puede especificar los resultados de las tareas mediante `CommandLineTool` tareas. En las siguientes secciones se muestran ejemplos de `CommandLineTool` tareas que definen distintos tipos de resultados.

Temas

- Resultado de la tarea para STDOUT
- Resultado de la tarea para STDERR
- Salida de la tarea a un archivo

- Salida de la tarea a una matriz de archivos

Resultado de la tarea para STDOUT

En este ejemplo se crea una `CommandLineTool` tarea que reproduce el contenido de STDOUT en un archivo de salida de texto denominado. output.txt Por ejemplo, si proporciona la siguiente entrada, el resultado de la tarea será ¡Hola mundo! en el output.txt archivo.

```
{
    "message": "Hello World!"
}
```

La `outputs` directiva especifica que el nombre de la salida es example_out y su tipo es`stdout`. Para que una tarea posterior consuma el resultado de esta tarea, se referirá al resultado como`example_out`.

Como HealthOmics crea registros para todo el contenido de STDERR y STDOUT, el resultado también aparece en los CloudWatch registros, junto con otra información de registro de STDERR para la tarea.

```
cwlVersion: v1.2
class: CommandLineTool
baseCommand: echo
stdout: output.txt
inputs:
  message:
    type: string
    inputBinding:
      position: 1
outputs:
  example_out:
    type: stdout

requirements:
    DockerRequirement:
        dockerPull: 123456789012.dkr.ecr.us-east-1.amazonaws.com/dockerhub/library/
ubuntu:20.04
    ResourceRequirement:
        ramMin: 2048
        coresMin: 1
```

Resultado de la tarea para STDERR

En este ejemplo se crea una `CommandLineTool` tarea que reproduce el contenido de STDERR en un archivo de salida de texto denominado. stderr.txt La tarea modifica el `baseCommand` modo que se `echo` escribe en STDERR (en lugar de en STDOUT).

La `outputs` directiva especifica que el nombre de la salida es stderr_out y su tipo es. `stderr`

Como HealthOmics crea registros para todo el contenido de STDERR y STDOUT, el resultado aparecerá en los CloudWatch registros, junto con otra información de registro de STDERR para la tarea.

```
cwlVersion: v1.2
class: CommandLineTool
baseCommand: [bash, -c]
stderr: stderr.txt
inputs:
  message:
    type: string
    inputBinding:
      position: 1
      shellQuote: true
      valueFrom: "echo $(self) >&2"
outputs:
  stderr_out:
    type: stderr

requirements:
    DockerRequirement:
        dockerPull: 123456789012.dkr.ecr.us-east-1.amazonaws.com/dockerhub/library/
ubuntu:20.04
    ResourceRequirement:
        ramMin: 2048
        coresMin: 1
```

Salida de la tarea a un archivo

En este ejemplo se crea una `CommandLineTool` tarea que crea un archivo tar comprimido a partir de los archivos de entrada. El nombre del archivo se proporciona como parámetro de entrada (archive_name).

La outputs directiva especifica que el tipo `archive_file` de salida es `File` y utiliza una referencia al parámetro de entrada para enlazarlo `archive_name` al archivo de salida.

```
cwlVersion: v1.2
class: CommandLineTool
baseCommand: [tar, cfz]
inputs:
  archive_name:
    type: string
    inputBinding:
      position: 1
  input_files:
    type: File[]
    inputBinding:
      position: 2

outputs:
  archive_file:
    type: File
    outputBinding:
      glob: "$(inputs.archive_name)"

requirements:
    DockerRequirement:
        dockerPull: 123456789012.dkr.ecr.us-east-1.amazonaws.com/dockerhub/library/
ubuntu:20.04
    ResourceRequirement:
        ramMin: 2048
        coresMin: 1
```

Salida de la tarea a una matriz de archivos

En este ejemplo, la `CommandLineTool` tarea crea una matriz de archivos mediante el `touch` comando. El comando usa las cadenas del parámetro `files-to-create` de entrada para nombrar los archivos. El comando genera una matriz de archivos. La matriz incluye todos los archivos del directorio de trabajo que coincidan con el `glob` patrón. En este ejemplo se utiliza un patrón comodín («*») que coincide con todos los archivos.

```
cwlVersion: v1.2
class: CommandLineTool
baseCommand: touch
inputs:
  files-to-create:
    type:
      type: array
```

```
      items: string
    inputBinding:
      position: 1
outputs:
  output-files:
    type:
      type: array
      items: File
    outputBinding:
      glob: "*"

requirements:
    DockerRequirement:
        dockerPull: 123456789012.dkr.ecr.us-east-1.amazonaws.com/dockerhub/library/
ubuntu:20.04
    ResourceRequirement:
        ramMin: 2048
        coresMin: 1
```

Recursos de tareas en una definición de HealthOmics flujo de trabajo

En la definición del flujo de trabajo, defina lo siguiente para cada tarea:

- La imagen del contenedor de la tarea. Para obtener más información, consulte Imágenes de contenedores para flujos de trabajo privados.
- El número CPUs y la memoria necesarios para la tarea. Para obtener más información, consulte Requisitos de cómputo y memoria para HealthOmics las tareas.

HealthOmics ignora las especificaciones de almacenamiento por tarea. HealthOmics proporciona almacenamiento de ejecución al que pueden acceder todas las tareas de la ejecución. Para obtener más información, consulte Ejecute tipos de almacenamiento en HealthOmics flujos de trabajo.

WDL

```
task my_task {
   runtime {
      container: "<aws-account-id>.dkr.ecr.<aws-region>.amazonaws.com/<image-name>"
      cpu: 2
      memory: "4 GB"
   }
   ...
```

```
}
```

En el caso de un flujo de trabajo de WDL, intenta realizar hasta dos HealthOmics reintentos para una tarea que no funcione debido a errores de servicio (la solicitud de API devuelve un código de estado HTTP 5XX). Para obtener más información sobre los reintentos de tareas, consulte. La tarea se reintenta

Puede excluirse del comportamiento de reintento especificando la siguiente configuración para la tarea en el archivo de definición de la WDL:

```
runtime {
   preemptible: 0
}
```

NextFlow

```
process my_task {
   container "<aws-account-id>.dkr.ecr.<aws-region>.amazonaws.com/<image-name>"
   cpus 2
   memory "4 GiB"
   ...
}
```

CWL

```
cwlVersion: v1.2
class: CommandLineTool
requirements:
    DockerRequirement:
        dockerPull: "<aws-account-id>.dkr.ecr.<aws-region>.amazonaws.com/<image-
name>"
    ResourceRequirement:
        coresMax: 2
        ramMax: 4000 # specified in mebibytes
```

Aceleradores de tareas en una definición de flujo de trabajo HealthOmics

En la definición del flujo de trabajo, si lo desea, puede especificar las especificaciones del acelerador de la GPU para una tarea. HealthOmics admite los siguientes valores de especificación de aceleración, junto con los tipos de instancias compatibles:

Especificaciones del acelerador	Tipos de instancias de Healthomics
nvidia-tesla-t4	G4
nvidia-tesla-t4-a10g	G4 y G5
nvidia-tesla-a10 g	G5
nvidia-t4-a10g-l4	G4, G5 y G6
nvidia-l4-a10g	G5 y G6
nvidia-l4	G6
nvidia-l40s	G6e

Si especificas un tipo de acelerador que admite varios tipos de instancias, HealthOmics selecciona el tipo de instancia en función de la capacidad disponible. Si ambos tipos de instancias están disponibles, HealthOmics da preferencia a la instancia de menor coste. La excepción es el acelerador de tareas nvidia-t4-a10g-l4, que da preferencia a la instancia de última generación disponible.

Para obtener más información sobre los tipos de instanciasInstancias de computación acelerada, consulte.

En el siguiente ejemplo, la definición del flujo de trabajo `nvidia-l4` se especifica como acelerador:

WDL

```
task my_task {
 runtime {
    ...
    acceleratorCount: 1
    acceleratorType: "nvidia-l4"
 }
 ...
}
```

NextFlow

```
process my_task {
 ...
 accelerator 1, type: "nvidia-l4"
 ...
}
```

CWL

```
cwlVersion: v1.2
class: CommandLineTool
requirements:
  ...
  cwltool:CUDARequirement:
      cudaDeviceCountMin: 1
      cudaComputeCapability: "nvidia-l4"
      cudaVersionMin: "1.0"
```

Especificaciones de la definición del flujo de trabajo de WDL

En los siguientes temas se proporcionan detalles sobre los tipos y directivas disponibles para las definiciones de flujos de trabajo de la WDL en. HealthOmics

Temas

- Conversión de tipos implícita en WDL lenient
- Definición del espacio de nombres en input.json
- Tipos primitivos en WDL

- Tipos complejos en WDL
- Directivas de la WDL
- Metadatos de tareas en WDL
- Ejemplo de definición de flujo de trabajo de WDL

Conversión de tipos implícita en WDL lenient

HealthOmics admite la conversión de tipos implícita en el archivo input.json y en la definición del flujo de trabajo. Para utilizar la conversión de tipos implícita, especifique el motor de flujo de trabajo como compatible con la WDL al crear el flujo de trabajo. WDL lenient está diseñado para gestionar los flujos de trabajo migrados desde Cromwell. Es compatible con las directivas de Cromwell de los clientes y con algunas lógicas no conformes.

WDL lenient admite la conversión de tipos para los siguientes elementos de la lista de excepciones limitadas de WDL:

- De flote a Int, donde la coerción no produce pérdida de precisión (por ejemplo, 1,0 se asigna a 1).
- Coloque la cadena en Int/Float, donde la coerción no produce pérdida de precisión.
- Asigne [W, X] a Array [Pair [Y, Z]], en el caso de que W sea coercible para Y y X sea coercible para Z.
- Array [Pair [W, X]] para mapear [Y, Z], en el caso de que W sea coercible para Y y X sea coercible para Z (por ejemplo, 1.0 se mapea con 1).

Para utilizar la conversión de tipos implícita, especifique el motor de flujo de trabajo como WDL_LENIENT al crear el flujo de trabajo o la versión del flujo de trabajo.

En la consola, el parámetro del motor de flujo de trabajo se denomina Idioma. En la API, el parámetro del motor de flujo de trabajo se denomina motor. Para obtener más información, consulte Crear un flujo de trabajo privado o Crear una versión de flujo de trabajo.

Definición del espacio de nombres en input.json

HealthOmics admite variables totalmente calificadas en input.json. Por ejemplo, si declara dos variables de entrada denominadas número1 y número2 en el flujo de trabajo: SumWorkflow

```
workflow SumWorkflow {
  input {
    Int number1
```

```
    Int number2
  }
}
```

Puedes usarlas como variables totalmente calificadas en input.json:

```
{
    "SumWorkflow.number1": 15,
    "SumWorkflow.number2": 27
}
```

Tipos primitivos en WDL

La siguiente tabla muestra cómo las entradas de la WDL se asignan a los tipos primitivos coincidentes. HealthOmics proporciona una compatibilidad limitada con la coerción de tipos, por lo que se recomienda establecer tipos explícitos.

Tipos primitivos

Tipo WDL	Tipo JSON	Ejemplo de WDL	Ejemplo de clave y valor de JSON	Notas
`Boolean`	`boolean`	`Boolean b`	`"b": true`	El valor debe estar en minúsculas y sin comillas.
`Int`	`integer`	`Int i`	`"i": 7`	No debe estar entre comillas.
`Float`	`number`	`Float f`	`"f": 42.2`	No debe estar entre comillas.
`String`	`string`	`String s`	`"s": "characters"`	Las cadenas JSON que son un URI deben asignarse a un archivo WDL para importarlas.

Tipo WDL	Tipo JSON	Ejemplo de WDL	Ejemplo de clave y valor de JSON	Notas
`File`	`string`	`File f`	`"f": "s3://amzn-s3-demo-bucket1/path/to/file"`	Amazon S3 y el HealthOmics almacenamiento URIs se importan siempre que la función de IAM proporcionada para el flujo de trabajo tenga acceso de lectura a estos objetos. No se admite ningún otro esquema de URI (como `file://https://`, `yftp://`). El URI debe especificar un objeto. No puede ser un directorio, lo que significa que no puede terminar con un `/`.

Tipo WDL	Tipo JSON	Ejemplo de WDL	Ejemplo de clave y valor de JSON	Notas
`Directory`	`string`	`Directory d`	`"d": "s3://bucket/path/"`	El `Directory` tipo no está incluido en WDL 1.0 ni 1.1, por lo que tendrá que añadirlo `version development` al encabezado del archivo WDL. El URI debe ser un URI de Amazon S3 y tener un prefijo que termine en «/». Todo el contenido del directorio se copiará de forma recursiva en el flujo de trabajo como una sola descarga. Solo `Directory` debe contener archivos relacionados con el flujo de trabajo.

Tipos complejos en WDL

La siguiente tabla muestra cómo las entradas de la WDL se asignan a los tipos JSON complejos coincidentes. Los tipos complejos de WDL son estructuras de datos compuestas por tipos primitivos. Las estructuras de datos, como las listas, se convertirán en matrices.

Tipos complejos

Tipo WDL	Tipo JSON	Ejemplo de WDL	Ejemplo de clave y valor de JSON	Notas
`Array`	`array`	`Array[Int] nums`	`“nums": [1, 2, 3]`	Los miembros de la matriz deben seguir el formato del tipo de matriz WDL.
`Pair`	`object`	`Pair[String, Int] str_to_i`	`“str_to_i": {"left": "0", "right": 1}`	Cada valor del par debe usar el formato JSON del tipo de WDL correspondiente.
`Map`	`object`	`Map[Int, String] int_to_string`	`"int_to_string": { 2: "hello", 1: "goodbye" }`	Cada entrada del mapa debe usar el formato JSON del tipo de WDL correspondiente.
`Struct`	`object`	`struct SampleBamAndIndex { String sample_name File bam File bam_index`	`"b_and_i": { "sample_name": "NA12878", "bam": "s3://amzn-s3-demo-bucket1/`	Los nombres de los miembros de la estructura deben coincidir exactamente con los nombres de las claves de los objetos JSON. Cada valor debe usar el formato

Tipo WDL	Tipo JSON	Ejemplo de WDL	Ejemplo de clave y valor de JSON	Notas
		`} SampleBam AndIndex b_and_i`	`NA12878.b am", "bam_inde x": "s3:// amzn- s3-demo- bucket1/ NA12878.b am.bai" }`	JSON del tipo WDL correspon diente.
`Object`	N/A	N/A	N/A	El `Object` tipo WDL está desactualizado y debe sustituirse por él `Struct` en todos los casos.

Directivas de la WDL

HealthOmics admite las siguientes directivas en todas las versiones de la WDL compatibles HealthOmics .

Configure los recursos de la GPU

HealthOmics admite los atributos de tiempo de ejecución acceleratorType y acceleratorCount con todas las instancias de GPU compatibles. HealthOmics también admite los alias denominados gpuType ygpuCount, que tienen la misma funcionalidad que sus homólogos aceleradores. Si la definición de WDL contiene ambas directivas, HealthOmics utiliza los valores del acelerador.

El siguiente ejemplo muestra cómo utilizar estas directivas:

```
runtime {
    gpuCount: 2
    gpuType: "nvidia-tesla-t4"
}
```

Configure el reintento de tareas para los errores de servicio

HealthOmics admite hasta dos reintentos para una tarea que haya fallado debido a errores de servicio (5XX códigos de estado HTTP). Puede configurar el número máximo de reintentos (1 o 2) y optar por no volver a intentarlo en caso de errores de servicio. De forma predeterminada, HealthOmics intenta un máximo de dos reintentos.

En el siguiente ejemplo, se establece `preemptible` la exclusión de los reintentos en caso de errores de servicio:

```
{
  preemptible: 0
}
```

Para obtener más información sobre los reintentos de tareas HealthOmics, consulte. La tarea se reintenta

Configure el reintento de tareas para que no haya memoria suficiente

HealthOmics admite los reintentos de una tarea que ha fallado porque se ha quedado sin memoria (código de salida del contenedor 137, código de estado HTTP 4XX). HealthOmics duplica la cantidad de memoria para cada reintento.

De forma predeterminada, HealthOmics no lo vuelve a intentar en este tipo de errores. Utilice la `maxRetries` directiva para especificar el número máximo de reintentos.

El ejemplo siguiente se establece `maxRetries` en 3, de modo que se HealthOmics intenta completar la tarea un máximo de cuatro intentos (el intento inicial más tres reintentos):

```
runtime {
    maxRetries: 3
}
```

> **Note**
>
> Para reintentar la tarea por falta de memoria, se requiere la versión 4.2.3 o posterior de GNU findutils. El contenedor de imágenes predeterminado incluye este paquete. HealthOmics Si especifica una imagen personalizada en su definición de WDL, asegúrese de que la imagen incluya GNU findutils 4.2.3+.

Configure los códigos de retorno

El atributo returnCodes proporciona un mecanismo para especificar un código de retorno, o un conjunto de códigos de retorno, que indica que una tarea se ha ejecutado correctamente. El motor WDL respeta los códigos de retorno que se especifican en la sección de tiempo de ejecución de la definición de la WDL y establece el estado de las tareas en consecuencia.

```
runtime {
    returnCodes: 1
}
```

HealthOmics también admite un alias denominado continueOnReturnCode, que tiene las mismas funciones que ReturnCodes. Si especifica ambos atributos, HealthOmics utiliza el valor ReturnCodes.

Metadatos de tareas en WDL

HealthOmics admite las siguientes opciones de metadatos para las tareas de la WDL.

Deshabilite el almacenamiento en caché a nivel de tarea con el atributo volatile

El atributo volatile le permite deshabilitar el almacenamiento en caché de llamadas para tareas específicas de su flujo de trabajo de WDL. Cuando una tarea está marcada como volátil, siempre se ejecutará y nunca utilizará los resultados almacenados en caché, incluso cuando el almacenamiento en caché esté habilitado para su ejecución.

Añada el atributo volátil a la sección de metadatos de la definición de la tarea:

```
task my_volatile_task {
    meta {
        volatile: true
    }

    input {
        String input_file
    }

    command {
        echo "Processing ${input_file}" > output.txt
    }

    output {
        File result = "output.txt"
```

```
    }
}
```

Ejemplo de definición de flujo de trabajo de WDL

Los siguientes ejemplos muestran las definiciones de flujos de trabajo privados para convertir de `CRAM` a `BAM` en WDL. El `BAM` flujo de trabajo `CRAM` to define dos tareas y utiliza las herramientas del `genomes-in-the-cloud` contenedor, como se muestra en el ejemplo y que está disponible públicamente.

El siguiente ejemplo muestra cómo incluir el contenedor Amazon ECR como parámetro. Esto permite HealthOmics verificar los permisos de acceso a su contenedor antes de que comience la ejecución.

```
{
   ...
   "gotc_docker":"<account_id>.dkr.ecr.<region>.amazonaws.com/genomes-in-the-
cloud:2.4.7-1603303710"
}
```

En el siguiente ejemplo, se muestra cómo especificar los archivos que se van a utilizar en la ejecución cuando los archivos se encuentran en un bucket de Amazon S3.

```
{
    "input_cram": "s3://amzn-s3-demo-bucket1/inputs/NA12878.cram",
    "ref_dict": "s3://amzn-s3-demo-bucket1/inputs/Homo_sapiens_assembly38.dict",
    "ref_fasta": "s3://amzn-s3-demo-bucket1/inputs/Homo_sapiens_assembly38.fasta",
    "ref_fasta_index": "s3://amzn-s3-demo-bucket1/inputs/
Homo_sapiens_assembly38.fasta.fai",
    "sample_name": "NA12878"
}
```

Si desea especificar archivos de un almacén de secuencias, indíquelo como se muestra en el siguiente ejemplo, utilizando el URI del almacén de secuencias.

```
{
    "input_cram": "omics://429915189008.storage.us-west-2.amazonaws.com/111122223333/
readSet/4500843795/source1",
    "ref_dict": "s3://amzn-s3-demo-bucket1/inputs/Homo_sapiens_assembly38.dict",
    "ref_fasta": "s3://amzn-s3-demo-bucket1/inputs/Homo_sapiens_assembly38.fasta",
    "ref_fasta_index": "s3://amzn-s3-demo-bucket1/inputs/
Homo_sapiens_assembly38.fasta.fai",
```

```
    "sample_name": "NA12878"
}
```

A continuación, puede definir su flujo de trabajo en WDL, como se muestra en el siguiente ejemplo.

```
 version 1.0
workflow CramToBamFlow {
    input {
        File ref_fasta
        File ref_fasta_index
        File ref_dict
        File input_cram
        String sample_name
        String gotc_docker = "<account>.dkr.ecr.us-west-2.amazonaws.com/genomes-in-the-
cloud:latest"
    }
    #Converts CRAM to SAM to BAM and makes BAI.
    call CramToBamTask{
         input:
            ref_fasta = ref_fasta,
            ref_fasta_index = ref_fasta_index,
            ref_dict = ref_dict,
            input_cram = input_cram,
            sample_name = sample_name,
            docker_image = gotc_docker,
     }
     #Validates Bam.
     call ValidateSamFile{
        input:
           input_bam = CramToBamTask.outputBam,
           docker_image = gotc_docker,
     }
     #Outputs Bam, Bai, and validation report to the FireCloud data model.
     output {
         File outputBam = CramToBamTask.outputBam
         File outputBai = CramToBamTask.outputBai
         File validation_report = ValidateSamFile.report
      }
}
#Task definitions.
task CramToBamTask {
    input {
       # Command parameters
```

```
        File ref_fasta
        File ref_fasta_index
        File ref_dict
        File input_cram
        String sample_name
        # Runtime parameters
        String docker_image
    }
   #Calls samtools view to do the conversion.
   command {
        set -eo pipefail

        samtools view -h -T ~{ref_fasta} ~{input_cram} |
        samtools view -b -o ~{sample_name}.bam -
        samtools index -b ~{sample_name}.bam
        mv ~{sample_name}.bam.bai ~{sample_name}.bai
    }

    #Runtime attributes:
    runtime {
        docker: docker_image
    }

    #Outputs a BAM and BAI with the same sample name
     output {
         File outputBam = "~{sample_name}.bam"
         File outputBai = "~{sample_name}.bai"
    }
}

#Validates BAM output to ensure it wasn't corrupted during the file conversion.
task ValidateSamFile {
   input {
      File input_bam
      Int machine_mem_size = 4
      String docker_image
   }
   String output_name = basename(input_bam, ".bam") + ".validation_report"
   Int command_mem_size = machine_mem_size - 1
   command {
        java -Xmx~{command_mem_size}G -jar /usr/gitc/picard.jar \
        ValidateSamFile \
        INPUT=~{input_bam} \
        OUTPUT=~{output_name} \
```

```
        MODE=SUMMARY \
        IS_BISULFITE_SEQUENCED=false
    }
    runtime {
    docker: docker_image
    }
   #A text file is generated that lists errors or warnings that apply.
    output {
        File report = "~{output_name}"
    }
}
```

Especificaciones de la definición del flujo de trabajo de Nextflow

HealthOmics es compatible con Nextflow y. DSL1 DSL2 Para obtener más información, consulte Compatibilidad con la versión de Nextflow.

Nextflow DSL2 se basa en el lenguaje de programación Groovy, por lo que los parámetros son dinámicos y la coerción de tipos es posible utilizando las mismas reglas que Groovy. Los parámetros y valores proporcionados por el JSON de entrada están disponibles en el mapa de parámetros () `params` del flujo de trabajo.

Temas

- Utilice los complementos nf-schema y nf-validation
- Especifique el almacenamiento URIs
- Directivas de Nextflow
- Exporta el contenido de la tarea

Utilice los complementos nf-schema y nf-validation

> **Note**
>
> Resumen del soporte para complementos HealthOmics :
>
> - v22.04: no hay soporte para complementos
> - v23.10: admite y `nf-schema` `nf-validation`
> - v24.10 — admite `nf-schema`

HealthOmics proporciona el siguiente soporte para los complementos de Nextflow:

- Para la versión 23.10 de Nextflow, HealthOmics preinstala el complemento nf-validation @1 .1.1.
- Para Nextflow v23.10 y versiones posteriores, preinstala el complemento nf-schema @2 .3.0. HealthOmics
- No puede recuperar complementos adicionales durante la ejecución de un flujo de trabajo. HealthOmics ignora cualquier otra versión del complemento que especifique en el `nextflow.config` archivo.
- Para Nextflow v24 y versiones posteriores, `nf-schema` es la nueva versión del complemento obsoleto. `nf-validation` Para obtener más información, consulte nf-schema en el repositorio de Nextflow. GitHub

Especifique el almacenamiento URIs

Cuando se utiliza un Amazon S3 o un HealthOmics URI para crear un archivo o un objeto de ruta de Nextflow, el objeto coincidente está disponible para el flujo de trabajo, siempre y cuando se conceda el acceso de lectura. Se permite el uso de prefijos o directorios en Amazon S3 URIs. Para ver ejemplos, consulte Formatos de parámetros de entrada de Amazon S3.

HealthOmics admite parcialmente el uso de patrones globales en Amazon S3 URIs o HealthOmics Storage URIs. Utilice patrones globales en la definición del flujo de trabajo para la creación de nuestros canales`path`. `file` Para conocer el comportamiento esperado y los casos exactos, consulteManejo de Nextflow del patrón Glob en las entradas de Amazon S3.

Directivas de Nextflow

Las directivas de Nextflow se configuran en el archivo de configuración o en la definición del flujo de trabajo de Nextflow. La siguiente lista muestra el orden de prioridad que se HealthOmics utiliza para aplicar los ajustes de configuración, de menor a mayor prioridad:

1. Configuración global en el archivo de configuración.
2. Sección de tareas de la definición del flujo de trabajo.
3. Selectores específicos de la tarea en el archivo de configuración.

Temas

- Estrategia de reintento de tareas mediante errorStrategy
- El intento de reintento de la tarea se realiza mediante maxRetries

- Opte por no participar en la tarea y vuelva a intentarlo con omicsRetryOn5xx
- Duración de la tarea según la directiva time

Estrategia de reintento de tareas mediante **`errorStrategy`**

Utilice la `errorStrategy` directiva para definir la estrategia en caso de errores en las tareas. De forma predeterminada, cuando una tarea regresa con una indicación de error (un estado de salida distinto de cero), la tarea se detiene y HealthOmics finaliza toda la ejecución. Si lo estableces`retry`, `errorStrategy` HealthOmics intenta volver a intentar la tarea fallida una vez. Para aumentar el número de reintentos, consulte. El intento de reintento de la tarea se realiza mediante `maxRetries`

```
process {
    label 'my_label'
    errorStrategy 'retry'

    script:
    """
    your-command-here
    """
}
```

Para obtener información sobre cómo HealthOmics gestiona los reintentos de tareas durante una ejecución, consulte. La tarea se reintenta

El intento de reintento de la tarea se realiza mediante **`maxRetries`**

De forma predeterminada, HealthOmics no intenta reintentar una tarea fallida ni intenta volver a intentarlo si se configura. `errorStrategy` Para aumentar el número máximo de reintentos, `errorStrategy` defina `retry` y configure el número máximo de reintentos mediante la directiva. `maxRetries`

El siguiente ejemplo establece el número máximo de reintentos en 3 en la configuración global.

```
process {
    errorStrategy = 'retry'
    maxRetries = 3
}
```

El siguiente ejemplo muestra cómo configurarlo `maxRetries` en la sección de tareas de la definición del flujo de trabajo.

```
process myTask {
    label 'my_label'
    errorStrategy 'retry'
    maxRetries 3

    script:
    """
    your-command-here
    """
}
```

El siguiente ejemplo muestra cómo especificar la configuración específica de la tarea en el archivo de configuración de Nextflow, en función de los selectores de nombres o etiquetas.

```
process {
    withLabel: 'my_label' {
        errorStrategy = 'retry'
        maxRetries = 3
    }

    withName: 'myTask' {
        errorStrategy = 'retry'
        maxRetries = 3
    }
}
```

Opte por no participar en la tarea y vuelva a intentarlo con **`omicsRetryOn5xx`**

En las versiones 23 y 24 de Nextflow, HealthOmics admite reintentos de tareas si la tarea falló debido a errores de servicio (códigos de estado HTTP 5XX). De forma predeterminada, HealthOmics intenta reintentar hasta dos veces una tarea fallida.

Puede configurarlo `omicsRetryOn5xx` para excluirse del reintento de la tarea en caso de errores de servicio. Para obtener más información sobre cómo volver a intentar una tarea HealthOmics, consulte. La tarea se reintenta

El siguiente ejemplo se configura `omicsRetryOn5xx` en la configuración global para optar por no volver a intentar la tarea.

```
process {
    omicsRetryOn5xx = false
```

```
}
```

El siguiente ejemplo muestra cómo realizar la configuración `omicsRetryOn5xx` en la sección de tareas de la definición del flujo de trabajo.

```
process myTask {
    label 'my_label'
    omicsRetryOn5xx = false

    script:
    """
    your-command-here
    """
}
```

El siguiente ejemplo muestra cómo establecer `omicsRetryOn5xx` una configuración específica para una tarea en el archivo de configuración de Nextflow, en función de los selectores de nombres o etiquetas.

```
process {
    withLabel: 'my_label' {
        omicsRetryOn5xx = false
    }

    withName: 'myTask' {
        omicsRetryOn5xx = false
    }
}
```

Duración de la tarea según la directiva **`time`**

HealthOmics proporciona una cuota ajustable (consulteHealthOmics cuotas de servicio) para especificar la duración máxima de una ejecución. Para los flujos de trabajo de las versiones 23 y 24 de Nextflow, también puede especificar la duración máxima de las tareas mediante la directiva Nextflow. `time`

Durante el desarrollo de un nuevo flujo de trabajo, establecer la duración máxima de las tareas te ayuda a atrapar las tareas fuera de control y las tareas de larga duración.

Para obtener más información sobre la directiva horaria de Nextflow, consulta la directiva horaria en la referencia de Nextflow.

HealthOmics proporciona el siguiente soporte para la directiva horaria de Nextflow:

1. HealthOmics admite una granularidad de 1 minuto para la directiva de tiempo. Puede especificar un valor entre 60 segundos y el valor de duración máxima de la ejecución.
2. Si introduce un valor inferior a 60, lo HealthOmics redondea a 60 segundos. Para valores superiores a 60, HealthOmics redondea al minuto más cercano.
3. Si el flujo de trabajo admite reintentos para una tarea, HealthOmics reintenta la tarea si se agota el tiempo de espera.
4. Si se agota el tiempo de espera de una tarea (o se agota el tiempo de espera del último reintento), HealthOmics cancela la tarea. Esta operación puede tener una duración de uno a dos minutos.
5. Cuando se agota el tiempo de espera de la tarea, HealthOmics establece el estado de ejecución y tarea como fallido y cancela las demás tareas en ejecución (para las tareas en estado Iniciativo, Pendiente o En ejecución). HealthOmics exporta los resultados de las tareas que completó antes de que se agotara el tiempo de espera a la ubicación de salida de S3 que haya designado.
6. El tiempo que una tarea pasa en estado pendiente no se tiene en cuenta para la duración de la tarea.
7. Si la ejecución forma parte de un grupo de ejecución y se agota el tiempo de espera antes que el temporizador de la tarea, la ejecución y la tarea pasan al estado fallido.

Especifique la duración del tiempo de espera mediante una o más de las siguientes unidades:`ms`,`s`, `mh`, od.

El siguiente ejemplo muestra cómo especificar la configuración global en el archivo de configuración de Nextflow. Establece un tiempo de espera global de 1 hora y 30 minutos.

```
process {
    time = '1h30m'
}
```

El siguiente ejemplo muestra cómo especificar una directiva horaria en la sección de tareas de la definición del flujo de trabajo. En este ejemplo se establece un tiempo de espera de 3 días, 5 horas y 4 minutos. Este valor tiene prioridad sobre el valor global del archivo de configuración, pero no tiene prioridad sobre una directiva de tiempo específica de la tarea en el archivo de `my_label` configuración.

```
process myTask {
    label 'my_label'
```

```
    time '3d5h4m'

    script:
    """
    your-command-here
    """
}
```

En el siguiente ejemplo, se muestra cómo especificar las directivas horarias específicas de la tarea en el archivo de configuración de Nextflow, en función de los selectores de nombres o etiquetas. En este ejemplo, se establece un valor de tiempo de espera global de la tarea de 30 minutos. Establece un valor de 2 horas para la tarea `myTask` y establece un valor de 3 horas para las tareas con etiqueta`my_label`. Para las tareas que coinciden con el selector, estos valores tienen prioridad sobre el valor global y el valor de la definición del flujo de trabajo.

```
process {
    time = '30m'

    withLabel: 'my_label' {
        time = '3h'
    }

    withName: 'myTask' {
        time = '2h'
    }
}
```

Exporta el contenido de la tarea

Para los flujos de trabajo escritos en Nextflow, defina una directiva PublishDir para exportar el contenido de las tareas a su bucket de salida de Amazon S3. Como se muestra en el siguiente ejemplo, defina el valor PublishDir en. `/mnt/workflow/pubdir` Para exportar archivos a Amazon S3, los archivos deben estar en este directorio.

```
 nextflow.enable.dsl=2

  workflow {
    CramToBamTask(params.ref_fasta, params.ref_fasta_index, params.ref_dict,
 params.input_cram, params.sample_name)
    ValidateSamFile(CramToBamTask.out.outputBam)
  }
```

```
process CramToBamTask {
  container "<account>.dkr.ecr.us-west-2.amazonaws.com/genomes-in-the-cloud"

  publishDir "/mnt/workflow/pubdir"

  input:
      path ref_fasta
      path ref_fasta_index
      path ref_dict
      path input_cram
      val sample_name

  output:
      path "${sample_name}.bam", emit: outputBam
      path "${sample_name}.bai", emit: outputBai

  script:
  """
      set -eo pipefail

      samtools view -h -T $ref_fasta $input_cram |
      samtools view -b -o ${sample_name}.bam -
      samtools index -b ${sample_name}.bam
      mv ${sample_name}.bam.bai ${sample_name}.bai
  """
}

process ValidateSamFile {
  container "<account>.dkr.ecr.us-west-2.amazonaws.com/genomes-in-the-cloud"

  publishDir "/mnt/workflow/pubdir"

  input:
      file input_bam

  output:
      path "validation_report"

  script:
  """
      java -Xmx3G -jar /usr/gitc/picard.jar \
      ValidateSamFile \
      INPUT=${input_bam} \
```

```
        OUTPUT=validation_report \
        MODE=SUMMARY \
        IS_BISULFITE_SEQUENCED=false
    """
  }
```

Especificaciones de la definición del flujo de trabajo de CWL

Los flujos de trabajo escritos en Common Workflow Language, o CWL, ofrecen una funcionalidad similar a los flujos de trabajo escritos en WDL y Nextflow. Puede utilizar Amazon S3 o el HealthOmics almacenamiento URIs como parámetros de entrada.

Si define la entrada en un archivo secundario de un subflujo de trabajo, añada la misma definición en el flujo de trabajo principal.

HealthOmics los flujos de trabajo no admiten los procesos de operación. Para obtener más información sobre los procesos operativos en los flujos de trabajo de CWL, consulte la documentación de CWL.

La mejor práctica es definir un flujo de trabajo de CWL independiente para cada contenedor que utilice. Le recomendamos que no codifique la entrada de DockerPull con una URI de Amazon ECR fija.

Temas

- Convierta los flujos de trabajo de CWL para usarlos HealthOmics
- Opte por no participar en la tarea, vuelva a intentarlo utilizando omicsRetryOn5xx
- Repite un paso del flujo de trabajo
- Vuelva a intentar las tareas con más memoria
- Ejemplos

Convierta los flujos de trabajo de CWL para usarlos HealthOmics

Para convertir una definición de flujo de trabajo de CWL existente y utilizarla HealthOmics, realice los siguientes cambios:

- Sustituya todos los contenedores URIs de Docker por Amazon URIs ECR.
- Asegúrese de que todos los archivos del flujo de trabajo estén declarados en el flujo de trabajo principal como entrada y de que todas las variables estén definidas de forma explícita.

- Asegúrese de que todo el JavaScript código cumpla con el modo estricto.

Opte por no participar en la tarea, vuelva a intentarlo utilizando **`omicsRetryOn5xx`**

HealthOmics admite reintentos de tareas si la tarea ha fallado debido a errores de servicio (5XX códigos de estado HTTP). De forma predeterminada, HealthOmics intenta reintentar hasta dos veces una tarea fallida. Para obtener más información sobre cómo volver a intentar una tarea HealthOmics, consulte. La tarea se reintenta

Para excluirse del reintento de tareas por errores de servicio, configure la `omicsRetryOn5xx` directiva en la definición del flujo de trabajo. Puede definir esta directiva en los requisitos o en las sugerencias. Recomendamos añadir la directiva como sugerencia de portabilidad.

```
requirements:
  ResourceRequirement:
    omicsRetryOn5xx: false

hints:
  ResourceRequirement:
    omicsRetryOn5xx: false
```

Los requisitos anulan las sugerencias. Si la implementación de una tarea proporciona un requisito de recursos en sugerencias que también se proporcionan en los requisitos de un flujo de trabajo adjunto, los requisitos adjuntos tienen prioridad.

Si el mismo requisito de tarea aparece en distintos niveles del flujo de trabajo, HealthOmics utiliza la entrada más específica de `requirements` (o`hints`, si no hay ninguna entrada). `requirements` La siguiente lista muestra el orden de prioridad que se HealthOmics utiliza para aplicar los valores de configuración, de menor a mayor prioridad:

- Nivel de flujo de trabajo
- Nivel de escalón
- Sección de tareas de la definición del flujo de trabajo

El siguiente ejemplo muestra cómo configurar la `omicsRetryOn5xx` directiva en los diferentes niveles del flujo de trabajo. En este ejemplo, el requisito de nivel de flujo de trabajo anula las sugerencias de nivel de flujo de trabajo. Las configuraciones de requisitos en los niveles de tarea y paso anulan las configuraciones de las sugerencias.

```
class: Workflow
# Workflow-level requirement and hint
requirements:
  ResourceRequirement:
    omicsRetryOn5xx: false

hints:
  ResourceRequirement:
    omicsRetryOn5xx: false  # The value in requirements overrides this value

steps:
  task_step:
    # Step-level requirement
    requirements:
      ResourceRequirement:
        omicsRetryOn5xx: false
    # Step-level hint
    hints:
      ResourceRequirement:
        omicsRetryOn5xx: false
    run:
      class: CommandLineTool
      # Task-level requirement
      requirements:
        ResourceRequirement:
          omicsRetryOn5xx: false
      # Task-level hint
      hints:
        ResourceRequirement:
          omicsRetryOn5xx: false
```

Repite un paso del flujo de trabajo

HealthOmics admite la repetición en bucle de un paso del flujo de trabajo. Puede usar bucles para ejecutar los pasos del flujo de trabajo repetidamente hasta que se cumpla una condición específica. Esto resulta útil para los procesos iterativos en los que es necesario repetir una tarea varias veces o hasta conseguir un resultado determinado.

Nota: La funcionalidad de bucle requiere la versión 1.2 o posterior de CWL. Los flujos de trabajo que utilizan versiones de CWL anteriores a la 1.2 no admiten las operaciones de bucle.

Para utilizar bucles en su flujo de trabajo de CWL, defina un requisito de bucle. El siguiente ejemplo muestra la configuración de los requisitos de bucles:

```
requirements:
  - class: "http://commonwl.org/cwltool#Loop"
    loopWhen: $(inputs.counter < inputs.max)
    loop:
      counter:
        loopSource: result
        valueFrom: $(self)
    outputMethod: last
```

El `loopWhen` campo controla cuándo termina el bucle. En este ejemplo, el bucle continúa mientras el contador sea inferior al valor máximo. El `loop` campo define cómo se actualizan los parámetros de entrada entre iteraciones. `loopSource`Especifica qué salida de la iteración anterior se transfiere a la siguiente iteración. El `outputMethod` campo configurado para `last` devolver solo el resultado de la iteración final.

Vuelva a intentar las tareas con más memoria

HealthOmics admite el reintento automático de los errores de las out-of-memory tareas. Cuando se cierra una tarea con el código 137 (out-of-memory), HealthOmics crea una nueva tarea con una mayor asignación de memoria en función del multiplicador especificado.

> **Note**
>
> HealthOmics reintenta out-of-memory los errores hasta 3 veces o hasta que la asignación de memoria alcance los 1536 GiB, el límite que se alcance primero.

El siguiente ejemplo muestra cómo configurar el reintento: out-of-memory

```
hints:
  ResourceRequirement:
    ramMin: 4096
  http://arvados.org/cwl#OutOfMemoryRetry:
    memoryRetryMultiplier: 2.5
```

Cuando se produce un error en una tarea debido a out-of-memory, HealthOmics calcula la asignación de memoria de los reintentos mediante la fórmula:. `previous_run_memory ×`

`memoryRetryMultiplier` En el ejemplo anterior, si se produce un error en la tarea con 4096 MB de memoria, el reintento utilizará 4096 × 2,5 = 10.240 MB de memoria.

El `memoryRetryMultiplier` parámetro controla la cantidad de memoria adicional que se debe asignar a los reintentos:

- Valor predeterminado: si no especificas un valor, el valor predeterminado es 2 (duplica la memoria)
- Rango válido: debe ser un número positivo superior 1 a. Los valores no válidos provocan un error de validación de 4 XX
- Valor mínimo efectivo: los valores entre 1 y `1.5` se incrementan automáticamente `1.5` para garantizar un aumento significativo de la memoria y evitar reintentos excesivos

Ejemplos

El siguiente es un ejemplo de un flujo de trabajo escrito en CWL.

```
cwlVersion: v1.2
class: Workflow

inputs:
in_file:
type: File
secondaryFiles: [.fai]

out_filename: string
docker_image: string


outputs:
copied_file:
type: File
outputSource: copy_step/copied_file

steps:
copy_step:
in:
  in_file: in_file
  out_filename: out_filename
  docker_image: docker_image
out: [copied_file]
```

```
run: copy.cwl
```

El siguiente archivo define la `copy.cwl` tarea.

```
cwlVersion: v1.2
class: CommandLineTool
baseCommand: cp

inputs:
in_file:
type: File
secondaryFiles: [.fai]
inputBinding:
  position: 1

out_filename:
type: string
inputBinding:
  position: 2
docker_image:
type: string

outputs:
copied_file:
type: File
outputBinding:
    glob: $(inputs.out_filename)

requirements:
InlineJavascriptRequirement: {}
DockerRequirement:
dockerPull: "$(inputs.docker_image)"
```

El siguiente es un ejemplo de un flujo de trabajo escrito en CWL con un requisito de GPU.

```
cwlVersion: v1.2
class: CommandLineTool
baseCommand: ["/bin/bash", "docm_haplotypeCaller.sh"]
$namespaces:
cwltool: http://commonwl.org/cwltool#
requirements:
```

```
cwltool:CUDARequirement:
cudaDeviceCountMin: 1
cudaComputeCapability: "nvidia-tesla-t4"
cudaVersionMin: "1.0"
InlineJavascriptRequirement: {}
InitialWorkDirRequirement:
listing:
- entryname: 'docm_haplotypeCaller.sh'
  entry: |
          nvidia-smi --query-gpu=gpu_name,gpu_bus_id,vbios_version --format=csv

inputs: []
outputs: []
```

Ejemplos de definiciones de flujo de trabajo

El siguiente ejemplo muestra la misma definición de flujo de trabajo en WDL, Nextflow y CWL.

WDL

```
version 1.1

task my_task {
   runtime { ... }
   inputs {
       File input_file
       String name
       Int threshold
   }

   command <<<
   my_tool --name ~{name} --threshold ~{threshold} ~{input_file}
   >>>

   output {
       File results = "results.txt"
   }
}

workflow my_workflow {
   inputs {
       File input_file
       String name
```

```
        Int threshold = 50
    }

    call my_task {
        input:
           input_file = input_file,
           name = name,
           threshold = threshold
    }
    outputs {
        File results = my_task.results
    }
}
```

Nextflow

```
nextflow.enable.dsl = 2

params.input_file = null
params.name = null
params.threshold = 50

process my_task {
   // <directives>

   input:
     path input_file
     val name
     val threshold

   output:
     path 'results.txt', emit: results

   script:
     """
     my_tool --name ${name} --threshold ${threshold} ${input_file}
     """



}

workflow MY_WORKFLOW {
   my_task(
```

```
        params.input_file,
        params.name,
        params.threshold
    )
}

workflow {
    MY_WORKFLOW()
}
```

CWL

```
cwlVersion: v1.2
class: Workflow

requirements:
    InlineJavascriptRequirement: {}

inputs:
   input_file: File
   name: string
   threshold: int

outputs:
    result:
        type: ...
        outputSource: ...

steps:
    my_task:
        run:
            class: CommandLineTool
            baseCommand: my_tool
            requirements:
                ...
            inputs:
                name:
                    type: string
                    inputBinding:
                        prefix: "--name"
                threshold:
```

```
                type: int
                inputBinding:
                    prefix: "--threshold"
            input_file:
                type: File
                inputBinding: {}
    outputs:
        results:
            type: File
            outputBinding:
                glob: results.txt
```

Archivos de plantillas de parámetros para HealthOmics flujos de trabajo

Las plantillas de parámetros definen los parámetros de entrada de un flujo de trabajo. Puede definir los parámetros de entrada para que su flujo de trabajo sea más flexible y versátil. Por ejemplo, puede definir un parámetro para la ubicación en Amazon S3 de los archivos del genoma de referencia. Las plantillas de parámetros se pueden proporcionar a través de un servicio de repositorio basado en Git o de su unidad local. A continuación, los usuarios pueden ejecutar el flujo de trabajo utilizando varios conjuntos de datos.

Puede crear la plantilla de parámetros para su flujo de trabajo o HealthOmics puede generar la plantilla de parámetros automáticamente.

La plantilla de parámetros es un archivo JSON. En el archivo, cada parámetro de entrada es un objeto con nombre que debe coincidir con el nombre de la entrada del flujo de trabajo. Cuando se inicia una ejecución, si no se proporcionan valores para todos los parámetros necesarios, se produce un error en la ejecución.

El objeto de parámetro de entrada incluye los siguientes atributos:

- description— Este atributo obligatorio es una cadena que la consola muestra en la página de inicio de la ejecución. Esta descripción también se conserva como metadatos de ejecución.
- optional— Este atributo opcional indica si el parámetro de entrada es opcional. Si no especifica el optional campo, el parámetro de entrada es obligatorio.

En el siguiente ejemplo de plantilla de parámetros se muestra cómo especificar los parámetros de entrada.

```
{
  "myRequiredParameter1": {
     "description": "this parameter is required",
  },
  "myRequiredParameter2": {
     "description": "this parameter is also required",
     "optional": false
  },
  "myOptionalParameter": {
     "description": "this parameter is optional",
     "optional": true
  }
}
```

Generación de plantillas de parámetros

HealthOmics genera la plantilla de parámetros analizando la definición del flujo de trabajo para detectar los parámetros de entrada. Si proporciona un archivo de plantilla de parámetros para un flujo de trabajo, los parámetros del archivo anulan los parámetros detectados en la definición del flujo de trabajo.

Existen pequeñas diferencias entre la lógica de análisis de los motores CWL, WDL y Nextflow, como se describe en las siguientes secciones.

Temas

- Detección de parámetros para CWL
- Detección de parámetros para WDL
- Detección de parámetros para Nextflow

Detección de parámetros para CWL

En el motor de flujo de trabajo CWL, la lógica de análisis hace las siguientes suposiciones:

- Todos los tipos admitidos que admiten valores NULL se marcan como parámetros de entrada opcionales.
- Todos los tipos admitidos que no sean nulos se marcan como parámetros de entrada obligatorios.
- Todos los parámetros con valores predeterminados se marcan como parámetros de entrada opcionales.

- Las descripciones se extraen de la `label` sección de la definición del `main` flujo de trabajo. Si no `label` se especifica, la descripción estará en blanco (una cadena vacía).

En las tablas siguientes se muestran ejemplos de interpolación de CWL. Para cada ejemplo, el nombre del parámetro es. x Si el parámetro es obligatorio, debe proporcionar un valor para el parámetro. Si el parámetro es opcional, no es necesario que proporcione un valor.

En esta tabla se muestran ejemplos de interpolación de CWL para tipos primitivos.

Input	Ejemplo de entrada/salida	Obligatorio
`x:` `  type: int`	1 o 2 o...	Sí

`x:`
`  type: int`
`  default: 2`
	El valor predeterminado es 2. La entrada válida es 1 o 2 o...	No

`x:` `  type: int?`	La entrada válida es Ninguna o 1 o 2 o...	No

`x:`
`  type: int?`
`  default: 2`
	El valor predeterminado es 2. La entrada válida es Ninguna o 1 o 2 o...	No

La siguiente tabla muestra ejemplos de interpolación de CWL para tipos complejos. Un tipo complejo es una colección de tipos primitivos.

Input	Ejemplo de entrada/salida	Obligatorio

`x:`
`  type: array`
`  items: int`
	[] o [1,2,3]	Sí

`x:`	Ninguno o [] o [1,2,3]	No

Input	Ejemplo de entrada/salida	Obligatorio
`type: array?` `items: int`		

`x:`
`type: array`
`items: int?`
	[] o [Ninguno, 3, Ninguno]	Sí

`x:`
`type: array?`
`items: int?`
	[Ninguno] o Ninguno o [1,2,3] o [Ninguno, 3] pero no []	No

Detección de parámetros para WDL

En el motor de flujo de trabajo de la WDL, la lógica de análisis hace las siguientes suposiciones:

- Todos los tipos admitidos que admiten valores NULL se marcan como parámetros de entrada opcionales.
- Para los tipos admitidos que no admiten valores nulos:
  - Cualquier variable de entrada con asignación de literales o expresiones se marca como parámetro opcional. Por ejemplo:

    ```
     Int x = 2
    Float f0 = 1.0 + f1
    ```

  - Si no se ha asignado ningún valor o expresión a los parámetros de entrada, se marcarán como parámetros obligatorios.
- Las descripciones se `parameter_meta` extraen de la definición del `main` flujo de trabajo. Si no `parameter_meta` se especifica, la descripción aparecerá en blanco (una cadena vacía). Para obtener más información, consulte la especificación WDL para los metadatos de los parámetros.

En las tablas siguientes se muestran ejemplos de interpolación de la WDL. Para cada ejemplo, el nombre del parámetro es. x Si el parámetro es obligatorio, debe proporcionar un valor para el parámetro. Si el parámetro es opcional, no es necesario que proporcione un valor.

En esta tabla se muestran ejemplos de interpolación de WDL para tipos primitivos.

Input	Ejemplo de entrada/salida	Obligatorio
Int x	1 o 2 o...	Sí
Int x = 2	2	No
Int x = 1+2	3	No
Int x = y+z	y+z	No
¿Int? x	Ninguno o 1 o 2 o...	Sí
¿Int? x = 2	Ninguno o 2	No
¿Int? x = 1+2	Ninguno o 3	No
¿Int? x = y+z	Ninguno o y+z	No

La siguiente tabla muestra ejemplos de interpolación de WDL para tipos complejos. Un tipo complejo es un conjunto de tipos primitivos.

Input	Ejemplo de entrada/salida	Obligatorio		
Array [Int] x	[1,2,3] o []	Sí		
Matriz [Int] + x	[1], pero no []	Sí		
¿Matriz [Int]? x	Ninguno o [] o [1,2,3]	No		
Matriz [Int?] x	[] o [Ninguno, 3, Ninguno]	Sí		
Matriz [Int?] =? x	[Ninguno] o Ninguno o [1,2,3] o [Ninguno, 3] pero no []	No		

Input	Ejemplo de entrada/salida	Obligatorio		
Ejemplo de estructura {String a, Int y} más adelante en las entradas: Sample mySample	`String a = mySample.a Int y = mySample.y`	Sí		
Ejemplo de estructura {String a, Int y} más adelante en las entradas: ¿Ejemplo? ¿Mi muestra	`if (defined( mySample)) { String a = mySample.a Int y = mySample.y }`	No		

Detección de parámetros para Nextflow

En el caso de Nextflow, HealthOmics genera la plantilla de parámetros analizando el archivo. `nextflow_schema.json` Si la definición del flujo de trabajo no incluye un archivo de esquema, HealthOmics analiza el archivo de definición del flujo de trabajo principal.

Temas

- Analizar el archivo de esquema
- Analizando el archivo principal
- Parámetros anidados
- Ejemplos de interpolación de Nextflow

Analizar el archivo de esquema

Para que el análisis funcione correctamente, asegúrese de que el archivo de esquema cumpla los siguientes requisitos:

- El archivo de esquema recibe un nombre `nextflow_schema.json` y se encuentra en el mismo directorio que el archivo de flujo de trabajo principal.
- El archivo de esquema es un JSON válido, tal como se define en cualquiera de los siguientes esquemas:
  - esquema json. org/draft/2020-12/schema.
  - esquema json. org/draft-07/schema.

HealthOmics analiza el `nextflow_schema.json` archivo para generar la plantilla de parámetros:

- Extrae todo properties lo que está definido en el esquema.
- Incluye la propiedad description si está disponible para la propiedad.
- Identifica si cada parámetro es opcional u obligatorio, en función del required campo de la propiedad.

El siguiente ejemplo muestra un archivo de definición y el archivo de parámetros generado.

```
{
    "$schema": "https://json-schema.org/draft/2020-12/schema",
    "type": "object",
    "$defs": {
        "input_options": {
            "title": "Input options",
            "type": "object",
            "required": ["input_file"],
            "properties": {
                "input_file": {
                    "type": "string",
                    "format": "file-path",
                    "pattern": "^s3://[a-z0-9.-]{3,63}(?:/\\S*)?$",
                    "description": "description for input_file"
                },
                "input_num": {
                    "type": "integer",
                    "default": 42,
```

```
                    "description": "description for input_num"
                }
            }
        },
        "output_options": {
            "title": "Output options",
            "type": "object",
            "required": ["output_dir"],
            "properties": {
                "output_dir": {
                    "type": "string",
                    "format": "file-path",
                    "description": "description for output_dir",
                }
            }
        }
    },
    "properties": {
        "ungrouped_input_bool": {
            "type": "boolean",
            "default": true
        }
    },
    "required": ["ungrouped_input_bool"],
    "allOf": [
        { "$ref": "#/$defs/input_options" },
        { "$ref": "#/$defs/output_options" }
    ]
}
```

La plantilla de parámetros generada:

```
{
    "input_file": {
        "description": "description for input_file",
        "optional": False
    },
    "input_num": {
        "description": "description for input_num",
        "optional": True
    },
    "output_dir": {
        "description": "description for output_dir",
```

```
        "optional": False
    },
    "ungrouped_input_bool": {
        "description": None,
        "optional": False
    }
}
```

Analizando el archivo principal

Si la definición del flujo de trabajo no incluye un `nextflow_schema.json` archivo, HealthOmics analiza el archivo de definición del flujo de trabajo principal.

HealthOmics analiza las `params` expresiones que se encuentran en el archivo principal de definición del flujo de trabajo y en el `nextflow.config` archivo. Todas `params` las que tengan valores predeterminados se marcan como opcionales.

Para que el análisis funcione correctamente, tenga en cuenta los siguientes requisitos:

- HealthOmics analiza solo el archivo de definición del flujo de trabajo principal. Para garantizar que se capturen todos los parámetros, le recomendamos que los conecte todos los params submódulos y flujos de trabajo importados.
- El archivo de configuración es opcional. Si define uno, asígnele un nombre `nextflow.config` y colóquelo en el mismo directorio que el archivo de definición del flujo de trabajo principal.

El siguiente ejemplo muestra un archivo de definición y la plantilla de parámetros generada.

```
params.input_file = "default.txt"
params.threads = 4
params.memory = "8GB"

workflow {
    if (params.version) {
        println "Using version: ${params.version}"
    }
}
```

La plantilla de parámetros generada:

```
{
```

```
    "input_file": {
        "description": None,
        "optional": True
    },
    "threads": {
        "description": None,
        "optional": True
    },
    "memory": {
        "description": None,
        "optional": True
    },
    "version": {
        "description": None,
        "optional": False
    }
}
```

Para los valores predeterminados que se definen en nextflow.config, HealthOmics recopila `params` las asignaciones y los parámetros declarados en ellos`params {}`, como se muestra en el siguiente ejemplo. En las sentencias de asignación, `params` debe aparecer en el lado izquierdo de la sentencia.

```
params.alpha = "alpha"
params.beta = "beta"

params {
    gamma = "gamma"
    delta = "delta"
}

env {
   // ignored, as this assignment isn't in the params block
   VERSION = "TEST"
}

// ignored, as params is not on the left side
interpolated_image = "${params.cli_image}"
```

La plantilla de parámetros generada:

```
{
```

```
    // other params in your main workflow defintion
    "alpha": {
        "description": None,
        "optional": True
    },
    "beta": {
        "description": None,
        "optional": True
    },
    "gamma": {
        "description": None,
        "optional": True
    },
    "delta": {
        "description": None,
        "optional": True
    }
}
```

Parámetros anidados

Ambos `nextflow.config` permiten `nextflow_schema.json` parámetros anidados. Sin embargo, la plantilla de HealthOmics parámetros solo requiere los parámetros de nivel superior. Si su flujo de trabajo utiliza un parámetro anidado, debe proporcionar un objeto JSON como entrada para ese parámetro.

Parámetros anidados en archivos de esquema

HealthOmics omite los anidados params al analizar un archivo. `nextflow_schema.json` Por ejemplo, si define el siguiente archivo: `nextflow_schema.json`

```
{
    "properties": {
        "input": {
            "properties": {
                "input_file": { ... },
                "input_num": { ... }
            }
        },
        "input_bool": { ... }
    }
}
```

HealthOmics ignora `input_file` y `input_num` cuando genera la plantilla de parámetros:

```
{
    "input": {
        "description": None,
        "optional": True
    },
    "input_bool": {
        "description": None,
        "optional": True
    }
}
```

Al ejecutar este flujo de trabajo, HealthOmics espera un `input.json` archivo similar al siguiente:

```
{
   "input": {
       "input_file": "s3://bucket/obj",
       "input_num": 2
   },
   "input_bool": false
}
```

Parámetros anidados en los archivos de configuración

HealthOmics no recopila los anidados params en un `nextflow.config` archivo y los omite durante el análisis. Por ejemplo, si define el siguiente archivo: `nextflow.config`

```
params.alpha = "alpha"
  params.nested.beta = "beta"

  params {
      gamma = "gamma"
      group {
          delta = "delta"
      }
  }
```

HealthOmics ignora `params.nested.beta` y `params.group.delta` cuando genera la plantilla de parámetros:

```
{
```

```
    "alpha": {
        "description": None,
        "optional": True
    },
    "gamma": {
        "description": None,
        "optional": True
    }
}
```

Ejemplos de interpolación de Nextflow

La siguiente tabla muestra ejemplos de interpolación de Nextflow para los parámetros del archivo principal.

Parámetros	Obligatorio
params.input_file	Sí
params.input_file = "s3://bucket/data.json»	No
params.nested.input_file	N/A
params.nested.input_file = "s3://bucket/data.json»	N/A

La siguiente tabla muestra ejemplos de interpolación de Nextflow para los parámetros del archivo. `nextflow.config`

Parámetros	Obligatorio
`params.input_file = "s3://bucket/data.json"`	No
`params { input_file = "s3://bucket/data.json" }`	No

Parámetros	Obligatorio
`params { nested { input_file = "s3://bucket/data.json" } }`	N/A
`input_file = params.input_file`	N/A

Imágenes de contenedores para flujos de trabajo privados

HealthOmics admite imágenes de contenedores alojadas en repositorios privados de Amazon ECR. Puede crear imágenes de contenedores y subirlas al repositorio privado. También puede usar su registro privado de Amazon ECR como caché de extracción para sincronizar el contenido de los registros anteriores.

Su repositorio de Amazon ECR debe residir en la misma AWS región que la cuenta que llama al servicio. La imagen del contenedor Cuenta de AWS puede ser propiedad de otra persona, siempre que el repositorio de imágenes de origen proporcione los permisos adecuados. Para obtener más información, consulte Políticas de acceso multicuenta a Amazon ECR.

Le recomendamos que defina la imagen del contenedor de Amazon ECR URIs como parámetros de su flujo de trabajo para poder verificar el acceso antes de que comience la ejecución. También facilita la ejecución de un flujo de trabajo en una nueva región al cambiar el parámetro de región.

> **Note**
>
> HealthOmics no admite contenedores ARM ni admite el acceso a repositorios públicos.

Para obtener información sobre la configuración de los permisos de IAM para acceder HealthOmics a Amazon ECR, consulte. HealthOmics Permisos de recursos

Temas

- Sincronización con registros de contenedores de terceros

- Consideraciones generales sobre las imágenes de contenedores de Amazon ECR
- Variables de entorno para los HealthOmics flujos de trabajo
- Uso de Java en imágenes de contenedores de Amazon ECR
- Añadir entradas de tareas a una imagen de contenedor de Amazon ECR

Sincronización con registros de contenedores de terceros

Puede utilizar las reglas de caché de extracción de Amazon ECR para sincronizar los repositorios de un registro ascendente compatible con sus repositorios privados de Amazon ECR. Para obtener más información, consulte Sincronizar un registro ascendente en la Guía del usuario de Amazon ECR.

La memoria caché extraíble crea automáticamente el repositorio de imágenes en su registro privado al crear la memoria caché y se sincroniza automáticamente con la imagen almacenada en caché cuando se producen cambios en la imagen original.

HealthOmics admite la caché extraíble para los siguientes registros ascendentes:

- Amazon ECR Public
- Registro de imágenes de contenedores de Kubernetes
- Quay
- Docker Hub
- Para Microsoft Azure Container Registry
- GitHub Registro de contenedores
- GitLab Registro de contenedores

HealthOmics no admite la extracción de caché para un repositorio privado upstream de Amazon ECR.

Entre las ventajas de utilizar la memoria caché de extracción de Amazon ECR se incluyen las siguientes:

1. Evita tener que migrar manualmente las imágenes del contenedor a Amazon ECR o sincronizar las actualizaciones del repositorio de terceros.
2. Los flujos de trabajo acceden a las imágenes de contenedores sincronizadas de su repositorio privado, lo que resulta más fiable que descargar contenido en tiempo de ejecución desde un registro público.

3. Como las cachés de extracción de Amazon ECR utilizan una estructura de URI predecible, el HealthOmics servicio puede mapear automáticamente el URI privado de Amazon ECR con el URI del registro anterior. No es necesario que actualice ni sustituya los valores de URI en la definición del flujo de trabajo.

Temas

- Configuración de la memoria caché de extracción
- Asignaciones de registro
- Mapeos de imágenes

Configuración de la memoria caché de extracción

Amazon ECR proporciona un registro para usted Cuenta de AWS en cada región. Asegúrese de crear la configuración de Amazon ECR en la misma región en la que planea ejecutar el flujo de trabajo.

En las siguientes secciones se describen las tareas de configuración de la memoria caché extraíble.

Tareas de configuración

- Cree una regla de extracción de caché
- Permisos de registro para el registro inicial
- Plantillas de creación de repositorios
- Creación de un flujo de trabajo

Cree una regla de extracción de caché

Cree una regla de caché de extracción de Amazon ECR para cada registro ascendente que contenga imágenes que desee almacenar en caché. Una regla especifica un mapeo entre un registro ascendente y el repositorio privado de Amazon ECR.

En el caso de un registro previo que requiera autenticación, debe proporcionar sus credenciales mediante AWS Secrets Manager.

> **Note**
>
> No modifique una regla de extracción de caché mientras una ejecución activa utilice el repositorio privado. La ejecución podría fallar o, lo que es más grave, provocar que tu canalización utilice imágenes inesperadas.

Para obtener más información, consulte Creación de una regla de extracción de caché en la Guía del usuario de Amazon Elastic Container Registry.

Cree una regla de extracción de caché mediante la consola

Para configurar la memoria caché de extracción, siga estos pasos con la consola Amazon ECR:

1. Abra la consola de Amazon ECR: https://console.aws.amazon.com /ecr
2. En el menú de la izquierda, en Registro privado, expanda Características y configuración y, a continuación, seleccione Extraer caché.
3. En la página de extracción de caché, selecciona Añadir regla.
4. En el panel de registro inicial, selecciona el registro inicial para sincronizarlo con tu registro privado y, a continuación, selecciona Siguiente.
5. Si el registro principal requiere autenticación, la consola abre una nueva página en la que puede especificar el secreto de SageMaker IA que contiene sus credenciales. Elija Siguiente.
6. En Especificar espacios de nombres, en el panel de espacios de nombres de Cache, elige si quieres crear los repositorios privados con un prefijo de repositorio específico o sin prefijo. Si opta por usar un prefijo, especifique el nombre del prefijo en el prefijo del repositorio de caché.
7. En el panel del espacio de nombres de Upstream, elija si desea extraer datos de los repositorios ascendentes con un prefijo de repositorio específico o sin prefijo. Si opta por utilizar un prefijo, especifique el nombre del prefijo en el prefijo del repositorio de Upstream.

   El panel de ejemplos del espacio de nombres muestra un ejemplo de solicitud de cambios, una URL anterior y la URL del repositorio de caché que se ha creado.
8. Elija Siguiente.
9. Revisa la configuración y selecciona Crear para crear la regla.

Para obtener más información, consulte Crear una regla de extracción de memoria caché (consola AWS de administración).

Cree una regla de extracción de caché mediante la CLI

Utilice el create-pull-through-cache-rule comando Amazon ECR para crear una regla de extracción de caché. En el caso de los registros anteriores que requieren autenticación, almacene las credenciales en un secreto de Secrets Manager.

En las siguientes secciones se proporcionan ejemplos de cada registro ascendente compatible.

Para Amazon ECR Public

En el siguiente ejemplo se crea una regla de caché de extracción para el registro público de Amazon ECR. Especifica un prefijo de repositorio `ecr-public`, lo que da como resultado que cada repositorio se cree mediante la regla de caché de extracción para tener el esquema de nomenclatura de `ecr-public/`*`upstream-repository-name`*.

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

Para el registro de contenedores de Kubernetes

En el siguiente ejemplo se crea una regla de extracción de caché para el registro público de Kubernetes. Especifica un prefijo de repositorio `kubernetes`, lo que da como resultado que cada repositorio se cree mediante la regla de caché de extracción para tener el esquema de nomenclatura de `kubernetes/`*`upstream-repository-name`*.

```
aws ecr create-pull-through-cache-rule \
     --ecr-repository-prefix kubernetes \
     --upstream-registry-url registry.k8s.io \
     --region us-east-1
```

Para Quay

El siguiente ejemplo crea una regla de extracción de caché para el registro público de Quay. Especifica un prefijo de repositorio `quay`, lo que da como resultado que cada repositorio se cree mediante la regla de caché de extracción para tener el esquema de nomenclatura de `quay/`*`upstream-repository-name`*.

```
aws ecr create-pull-through-cache-rule \
```

```
    --ecr-repository-prefix quay \
    --upstream-registry-url quay.io \
    --region us-east-1
```

Para Docker Hub

El siguiente ejemplo crea una regla de caché de extracción para el registro de Docker Hub. Especifica un prefijo de repositorio `docker-hub`, lo que da como resultado que cada repositorio se cree mediante la regla de caché de extracción para tener el esquema de nomenclatura de `docker-hub/`*`upstream-repository-name`*. Debe especificar el nombre de recurso de Amazon (ARN) completo del secreto que contiene las credenciales de Docker Hub.

```
aws ecr create-pull-through-cache-rule \
    --ecr-repository-prefix docker-hub \
    --upstream-registry-url registry-1.docker.io \
    --credential-arn arn:aws:secretsmanager:us-east-1:111122223333:secret:ecr-
pullthroughcache/example1234 \
    --region us-east-1
```

Para GitHub Container Registry

En el siguiente ejemplo, se crea una regla de extracción de caché para el GitHub Container Registry. Especifica un prefijo de repositorio `github`, lo que da como resultado que cada repositorio se cree mediante la regla de caché de extracción para tener el esquema de nomenclatura de `github/`*`upstream-repository-name`*. Debe especificar el nombre de recurso de Amazon (ARN) completo del secreto que contiene sus credenciales de GitHub Container Registry.

```
aws ecr create-pull-through-cache-rule \
    --ecr-repository-prefix github \
    --upstream-registry-url ghcr.io \
    --credential-arn arn:aws:secretsmanager:us-east-1:111122223333:secret:ecr-
pullthroughcache/example1234 \
    --region us-east-1
```

Para Microsoft Azure Container Registry

El siguiente ejemplo crea una regla de caché de extracción para el Microsoft Azure Container Registry. Especifica un prefijo de repositorio `azure`, lo que da como resultado que cada repositorio se cree mediante la regla de caché de extracción para tener el esquema de nomenclatura de

`azure/`*`upstream-repository-name`*. Debe especificar el nombre de recurso de Amazon (ARN) completo del secreto que contiene las credenciales de Microsoft Azure Container Registry.

```
aws ecr create-pull-through-cache-rule \
     --ecr-repository-prefix azure \
     --upstream-registry-url myregistry.azurecr.io \
     --credential-arn arn:aws:secretsmanager:us-east-1:111122223333:secret:ecr-
pullthroughcache/example1234 \
     --region us-east-1
```

Para GitLab Container Registry

En el siguiente ejemplo, se crea una regla de extracción de caché para el GitLab Container Registry. Especifica un prefijo de repositorio `gitlab`, lo que da como resultado que cada repositorio se cree mediante la regla de caché de extracción para tener el esquema de nomenclatura de `gitlab/`*`upstream-repository-name`*. Debe especificar el nombre de recurso de Amazon (ARN) completo del secreto que contiene sus credenciales de GitLab Container Registry.

```
aws ecr create-pull-through-cache-rule \
     --ecr-repository-prefix gitlab \
     --upstream-registry-url registry.gitlab.com \
     --credential-arn arn:aws:secretsmanager:us-east-1:111122223333:secret:ecr-
pullthroughcache/example1234 \
     --region us-east-1
```

Para obtener más información, consulte Crear una regla de extracción de caché (CLI) en la Guía del usuario de Amazon ECR.

Puede usar el comando get-run-task CLI para recuperar información sobre la imagen del contenedor utilizada para una tarea específica:

```
aws omics get-run-task --id 1234567 --task-id <task_id>
```

El resultado incluye la siguiente información sobre la imagen del contenedor:

```
"imageDetails": {
   "image": "string",
   "imageDigest": "string",
   "sourceImage": "string",
         ...
```

```
}
```

Permisos de registro para el registro inicial

Utilice los permisos de registro HealthOmics para poder utilizar la caché de extracción y extraer las imágenes del contenedor al registro privado de Amazon ECR. Añada una política de registro de Amazon ECR al registro que proporcione los contenedores utilizados en las ejecuciones.

La siguiente política otorga permiso al HealthOmics servicio para crear repositorios con los prefijos de caché de extracción especificados e iniciar extracciones ascendentes en estos repositorios.

1. En la consola Amazon ECR, abra el menú de la izquierda, en Registro privado, expanda Permisos de registro y, a continuación, seleccione Generar declaración.
2. En la parte superior derecha, selecciona JSON. Introduce una política similar a la siguiente:

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPTCinRegPermissions",
            "Effect": "Allow",
            "Principal": {
                "Service": "omics.amazonaws.com"
            },
            "Action": [
                "ecr:CreateRepository",
                "ecr:BatchImportUpstreamImage"
            ],
            "Resource": [
                "arn:aws:ecr:us-east-1:123456789012:repository/ecr-public/*",
                "arn:aws:ecr:us-east-1:123456789012:repository/docker-hub/*"
            ]
        }
    ]
}
```

Plantillas de creación de repositorios

Para utilizar el almacenamiento en caché mediante extracción HealthOmics, el repositorio de Amazon ECR debe tener una plantilla de creación de repositorios. La plantilla define los ajustes de configuración para cuando usted o Amazon ECR creen un repositorio privado para un registro principal.

Cada plantilla contiene un prefijo de espacio de nombres de repositorio, que Amazon ECR utiliza para hacer coincidir los nuevos repositorios con una plantilla específica. Las plantillas especifican la configuración de todos los ajustes del repositorio, incluidas las políticas de acceso basadas en los recursos, la inmutabilidad de las etiquetas, el cifrado y las políticas de ciclo de vida.

Para obtener más información, consulte las plantillas de creación de repositorios en la Guía del usuario de Amazon Elastic Container Registry.

Cómo crear una plantilla de creación de repositorios:

1. En la consola de Amazon ECR, abra el menú de la izquierda, en Registro privado, expanda Características y configuración y, a continuación, seleccione Plantillas de creación de repositorios.
2. Seleccione Crear plantilla.
3. En Detalles de la plantilla, selecciona Extraer caché.
4. Elige si quieres aplicar esta plantilla a un prefijo específico o a todos los repositorios que no coincidan con otra plantilla.

   Si eliges un prefijo específico, introduce el valor del prefijo del espacio de nombres en Prefix. Especificó este prefijo al crear la regla de PTC.
5. Elija Siguiente.
6. En la página Añadir configuración de creación de repositorios, introduzca los permisos del repositorio. Utilice uno de los ejemplos de declaraciones de política o introduzca uno similar al ejemplo siguiente:

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
```

```
            "Sid": "PTCRepoCreationTemplate",
            "Effect": "Allow",
            "Principal": {
                "Service": "omics.amazonaws.com"
            },
            "Action": [
                "ecr:BatchGetImage",
                "ecr:GetDownloadUrlForLayer"
            ],
            "Resource": "*"
        }
    ]
}
```

7. Si lo desea, puede añadir la configuración del repositorio, como la política del ciclo de vida y las etiquetas. Amazon ECR aplica estas reglas a todas las imágenes de contenedores creadas para la caché de extracción que utilizan el prefijo especificado.
8. Elija Siguiente.
9. Revise la configuración y seleccione Siguiente.

Creación de un flujo de trabajo

Al crear un nuevo flujo de trabajo o una nueva versión del flujo de trabajo, revise las asignaciones del registro y actualícelas si es necesario. Para obtener más información, consulte Crear un flujo de trabajo privado.

Asignaciones de registro

Defina las asignaciones de registro para mapear entre los prefijos de su registro privado de Amazon ECR y los nombres de registro anteriores.

Para obtener más información sobre las asignaciones de registro de Amazon ECR, consulte Creación de una regla de extracción de caché en Amazon ECR.

El siguiente ejemplo muestra las asignaciones de registro a Docker Hub, Quay y Amazon ECR Public.

```
{
    "registryMappings": [
        {
            "upstreamRegistryUrl": "registry-1.docker.io",
```

```
            "ecrRepositoryPrefix": "docker-hub"
        },
        {
            "upstreamRegistryUrl": "quay.io",
            "ecrRepositoryPrefix": "quay"
        },
        {
            "upstreamRegistryUrl": "public.ecr.aws",
            "ecrRepositoryPrefix": "ecr-public"
        }
    ]
}
```

Mapeos de imágenes

Defina las asignaciones de imágenes para mapearlas entre los nombres de las imágenes tal como se definen en sus flujos de trabajo privados de Amazon ECR y los nombres de las imágenes en el registro principal.

Puede utilizar las asignaciones de imágenes con registros que admitan la extracción de memoria caché. También puedes usar mapeos de imágenes con registros anteriores en los que no se HealthOmics admite la extracción de memoria caché. Tienes que sincronizar manualmente el registro original con tu repositorio privado.

Para obtener más información sobre las asignaciones de imágenes de Amazon ECR, consulte Creación de una regla de extracción de caché en Amazon ECR.

El siguiente ejemplo muestra las asignaciones de imágenes privadas de Amazon ECR a una imagen genómica pública y a la imagen más reciente de Ubuntu.

```
{
    "imageMappings": [
        {
            "sourceImage": "public.ecr.aws/aws-genomics/broadinstitute/gatk:4.6.0.2",
            "destinationImage": "123456789012.dkr.ecr.us-east-1.amazonaws.com/
broadinstitute/gatk:4.6.0.2"
        },
        {
            "sourceImage": "ubuntu:latest",
            "destinationImage": "123456789012.dkr.ecr.us-east-1.amazonaws.com/custom/
ubuntu:latest",
        }
```

```
    ]
}
```

Consideraciones generales sobre las imágenes de contenedores de Amazon ECR

- Arquitectura

  HealthOmics admite contenedores x86_64. Si su máquina local está basada en ARM, como Apple Mac, utilice un comando como el siguiente para crear una imagen de contenedor x86_64:

  ```
  docker build --platform amd64 -t my_tool:latest .
  ```

- Punto de entrada y carcasa

  HealthOmics Los motores de flujo de trabajo inyectan scripts bash como una sustitución de comandos en las imágenes del contenedor utilizadas en las tareas del flujo de trabajo. Por lo tanto, las imágenes de los contenedores deben crearse sin un PUNTO DE ENTRADA específico, de modo que el shell bash sea el predeterminado.

- Rutas montadas

  Un sistema de archivos compartido se monta en las tareas del contenedor en /tmp. Se anularán todos los datos o herramientas integrados en la imagen del contenedor en esta ubicación.

  La definición del flujo de trabajo está disponible para las tareas mediante un montaje de solo lectura en /mnt/workflow.

- Tamaño de imagen

  Consulte los tamaños máximos HealthOmics cuotas de tamaño fijas en el flujo de trabajo de imagen del contenedor.

Variables de entorno para los HealthOmics flujos de trabajo

HealthOmics proporciona variables de entorno que contienen información sobre el flujo de trabajo que se ejecuta en el contenedor. Puede utilizar los valores de estas variables en la lógica de las tareas del flujo de trabajo.

Todas las variables HealthOmics del flujo de trabajo comienzan con el `AWS_WORKFLOW_` prefijo. Este prefijo es un prefijo de variable de entorno protegido. No utilice este prefijo para sus propias variables en los contenedores de flujo de trabajo.

HealthOmics proporciona las siguientes variables de entorno de flujo de trabajo:

AWS_REGION

Esta variable es la región en la que se ejecuta el contenedor.

AWS_WORKFLOW_EJECUTAR

Esta variable es el nombre de la ejecución actual.

AWS_WORKFLOW_RUN_ID

Esta variable es el identificador de ejecución de la ejecución actual.

AWS_WORKFLOW_RUN_UUID

Esta variable es el UUID de ejecución de la ejecución actual.

AWS_WORKFLOW_TAREA

Esta variable es el nombre de la tarea actual.

AWS_WORKFLOW_TASK_ID

Esta variable es el identificador de la tarea actual.

AWS_WORKFLOW_TASK_UUID

Esta variable es el UUID de la tarea actual.

El siguiente ejemplo muestra los valores típicos de cada variable de entorno:

```
AWS Region: us-east-1
Workflow Run: arn:aws:omics:us-east-1:123456789012:run/6470304
Workflow Run ID: 6470304
Workflow Run UUID: f4d9ed47-192e-760e-f3a8-13afedbd4937
Workflow Task: arn:aws:omics:us-east-1:123456789012:task/4192063
Workflow Task ID: 4192063
Workflow Task UUID: f0c9ed49-652c-4a38-7646-60ad835e0a2e
```

Uso de Java en imágenes de contenedores de Amazon ECR

Si una tarea de flujo de trabajo utiliza una aplicación Java como GATK, tenga en cuenta los siguientes requisitos de memoria para el contenedor:

- Las aplicaciones Java utilizan memoria de pila y memoria de pila. De forma predeterminada, la memoria de pila máxima es un porcentaje de la memoria total disponible en el contenedor. Este valor predeterminado depende de la distribución específica de la JVM y de la versión de la JVM, así que consulte la documentación correspondiente a su JVM o establezca explícitamente el máximo de memoria de pila mediante las opciones de la línea de comandos de Java (como `-Xmx`).
- No establezcas que la memoria de pila máxima sea el 100% de la asignación de memoria del contenedor, ya que la pila de JVM también requiere memoria. También se necesita memoria para el recolector de elementos no utilizados de la JVM y para cualquier otro proceso del sistema operativo que se ejecute en el contenedor.
- Algunas aplicaciones Java, como GATK, pueden utilizar invocaciones de métodos nativos u otras optimizaciones, como los archivos de mapeo de memoria. Estas técnicas requieren asignaciones de memoria que se realicen «fuera del montón», que no están controladas por el parámetro de pila máxima de la JVM.

  Si sabe (o sospecha) que su aplicación Java asigna memoria fuera del montón, asegúrese de que la asignación de la memoria a las tareas incluya los requisitos de memoria fuera del montón.

  Si estas asignaciones fuera del montón provocan que el contenedor se quede sin memoria, normalmente no aparecerá ningún OutOfMemory error de Java, ya que la JVM no controla esta memoria.

Añadir entradas de tareas a una imagen de contenedor de Amazon ECR

Añada todos los ejecutables, bibliotecas y scripts necesarios para ejecutar una tarea de flujo de trabajo a la imagen de Amazon ECR que se utiliza para ejecutar la tarea.

Se recomienda evitar el uso de scripts, archivos binarios y bibliotecas externos a la imagen del contenedor de tareas. Esto es especialmente importante cuando se utilizan `nf-core` flujos de trabajo que utilizan un `bin` directorio como parte del paquete de flujo de trabajo. Si bien este directorio estará disponible para la tarea de flujo de trabajo, se monta como un directorio de solo lectura. Los recursos necesarios de este directorio deben copiarse en la imagen de la tarea y estar disponibles en tiempo de ejecución o al crear la imagen del contenedor utilizada para la tarea.

Consulte HealthOmics cuotas de tamaño fijas en el flujo de trabajo el tamaño máximo de la imagen del contenedor que HealthOmics admite.

HealthOmics Archivos README de flujo de trabajo

Puede cargar un archivo README.md que contenga instrucciones, diagramas e información esencial para su flujo de trabajo. Cada versión del flujo de trabajo admite un archivo README, que puede actualizar en cualquier momento.

Los requisitos de README incluyen:

- El archivo README debe estar en formato markdown (.md)
- Tamaño máximo de archivo: 500 KiB

Temas

- Utilice un archivo README existente
- Condiciones de renderizado

Utilice un archivo README existente

READMEs los archivos exportados desde los repositorios de Git contienen enlaces relativos que normalmente no funcionan fuera del repositorio. HealthOmics La integración de Git los convierte automáticamente en enlaces absolutos para una representación adecuada en la consola, lo que elimina la necesidad de actualizar manualmente las URL.

En el caso de READMEs importar desde Amazon S3 o unidades locales, las imágenes y los enlaces deben ser públicas URLs o tener sus rutas relativas actualizadas para que se representen correctamente.

> **Note**
>
> Las imágenes deben estar alojadas públicamente para poder mostrarse en la HealthOmics consola. Las imágenes almacenadas en GitHub Enterprise Server GitLab Self-Managed los repositorios no se pueden renderizar.

Condiciones de renderizado

La HealthOmics consola interpola imágenes y enlaces de acceso público mediante rutas absolutas. Para renderizar URLs desde repositorios privados, el usuario debe tener acceso al repositorio.

Nuestros GitHub Enterprise Server GitLab Self-Managed repositorios, que utilizan dominios personalizados, HealthOmics no pueden resolver los enlaces relativos ni renderizar las imágenes almacenadas en estos repositorios privados.

En la siguiente tabla se muestran los elementos de marcado compatibles con la vista README de la AWS consola.

Elemento	AWS consola
Alertas	Sí, pero sin iconos
Insignias	Sí
Formato de texto básico	Sí
Bloques de código	Sí, pero no tiene la funcionalidad de resaltar la sintaxis ni de copiar
Secciones plegables	Sí
Encabezados	Sí
Formatos de imagen	Sí
Imagen (en la que se puede hacer clic)	Sí
Saltos de línea	Sí
Diagrama de sirena	Solo puede abrir un gráfico, mover la posición del gráfico y copiar código
Cuotas	Sí
Subíndice y superíndice	Sí
Tablas	Sí, pero no admite la alineación del texto
Alineación de texto	Sí

Uso de imagen y enlace URLs

En función del proveedor de fuentes, estructura tu base URLs para las páginas e imágenes en los siguientes formatos.

- `{username}`: El nombre de usuario en el que está alojado el repositorio.
- `{repo}`: el nombre del repositorio.
- `{ref}`: La referencia de origen (rama, etiqueta e ID de confirmación).
- `{path}`: la ruta del archivo a la página o imagen del repositorio.

Proveedor de fuentes	URL de la página	URL de la imagen

GitHub	`https://github.com/{username}/{repo}/blob/{ref}/{path}`	`https://github.com/{username}/{repo}/blob/{ref}/{path}?raw=true`
		
		`https://raw.githubusercontent.com/{username}/{repo}/{ref}/{path}`

GitLab	`https://gitlab.com/{username}/{repo}/-/blob/{ref}/{path}`	`https://gitlab.com/{username}/{repo}/-/raw/{ref}/{path}`
Bitbucket	`https://bitbucket.org/{username}/{repo}/src/{ref}/{path}`	`https://bitbucket.org/{username}/{repo}/raw/{ref}/{path}`

GitHubGitLab, y Bitbucket admiten tanto páginas como imágenes URLs que enlazan a un repositorio público. En la siguiente tabla, se muestra la compatibilidad de cada proveedor de fuentes con la representación de imágenes y enlaces URLs en repositorios privados.

Soporte para repositorios privados		
Proveedor de fuentes	URL de la página	URL de la imagen
GitHub	Solo con acceso al repositorio	No
GitLab	Solo con acceso al repositorio	No
Bitbucket	Solo con acceso al repositorio	No

Solicitud de licencias de Sentieon para flujos de trabajo privados

Si su flujo de trabajo privado utiliza el software Sentieon, necesitará una licencia de Senieon. Siga estos pasos para solicitar y configurar una licencia para el software Sentieon:

- Solicite una licencia de Sentieon
  - Envíe un correo electrónico al grupo de soporte de Sentieon (support@sentieon.com) para solicitar una licencia de software.
    - Introduce tu AWS seudónimo de Canonical en el correo electrónico.
    - Sigue estas instrucciones para encontrar tu seudónimo de AWS Canonical.
- Actualiza tu rol HealthOmics de servicio para concederle acceso al proxy del servidor de licencias de Sentieon y al depósito Omics de Sentieon en tu región. El siguiente ejemplo permite el acceso a. `us-east-1` Si es necesario, sustituya este texto por su región.

  JSON

```
{
          "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObjectAcl",
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::omics-ap-us-east-1/*",
                "arn:aws:s3:::sentieon-omics-license-us-east-1/*"
            ]
```

```
        }
    ]
}
```

- Genere un caso de AWS soporte para acceder al proxy del servidor de licencias de Sentieon.
  - Para crear un caso de soporte, vaya a support.console.aws.amazon.com.
  - Indique su región y su Cuenta de AWS región en el caso de soporte. Su cuenta se añade a la lista de permitidos para el proxy del servidor de licencias.
- Cree su flujo de trabajo privado utilizando el contenedor Sentieon y el script de licencia Sentieon.
  - Para obtener instrucciones adicionales sobre el uso de las herramientas de Sentieon en flujos de trabajo privados, consulte Sentieon-Amazon-Omics en. GitHub
- La versión 202112.07 y superior del software Sentieon son compatibles con el servidor proxy de licencias. HealthOmics Para utilizar las versiones del software Sentieon anteriores a la 202112.07, póngase en contacto con el servicio de asistencia de Sentieon.

Lentes de flujo de trabajo en HealthOmics

Tras crear un flujo de trabajo, le recomendamos que ejecute un filtro en el flujo de trabajo antes de iniciar la primera ejecución. El linter detecta errores que pueden provocar un error en la ejecución.

En el caso de WDL, ejecuta HealthOmics automáticamente un linter al crear el flujo de trabajo. El resultado del linter está disponible en el `statusMessage` campo de la respuesta. get-workflow Utilice el siguiente comando CLI para recuperar la salida de estado (utilice el ID de flujo de trabajo del flujo de trabajo de WDL que creó):

```
aws omics get-workflow
   –id 123456
   –query 'statusMessage'
```

HealthOmics proporciona linters que puede ejecutar en la definición del flujo de trabajo antes de crearlo. Ejecuta estos linters en las canalizaciones existentes a las que vayas a migrar. HealthOmics

- WDL— Una imagen pública de Amazon ECR para ejecutar un linter de WDL.

- Nextflow— Una imagen pública de Amazon ECR para ejecutar las reglas de Linter para Nextflow. Puede acceder al código fuente de este linter desde. GitHub
- CWL— no disponible

HealthOmics operaciones de flujo de trabajo

Para crear un flujo de trabajo privado, necesitas:

- Workflow definition file:Un archivo de definición de flujo de trabajo escrito en WDLNextflow, oCWL. La definición del flujo de trabajo especifica las entradas y salidas de las ejecuciones que utilizan el flujo de trabajo. También incluye especificaciones para las ejecuciones y tareas de ejecución del flujo de trabajo, incluidos los requisitos de procesamiento y memoria. El archivo de definición del flujo de trabajo debe estar en `.zip` formato. Para obtener más información, consulte Archivos de definición de flujo de trabajo en HealthOmics.
  - Puede usar Amazon Q CLI para crear y validar sus archivos de definición de flujo de trabajo en WDL, Nextflow y CWL. Para obtener más información, consulte Ejemplos de instrucciones para Amazon Q CLI y el tutorial sobre HealthOmics IA generativa de Agentic en. GitHub
- (Optional) Parameter template file:Un archivo de plantilla de parámetros escrito en. JSON Cree el archivo para definir los parámetros de ejecución o HealthOmics genere la plantilla de parámetros automáticamente. Para obtener más información, consulte Archivos de plantillas de parámetros para HealthOmics flujos de trabajo.
- Amazon ECR container images:Cree repositorios privados de Amazon ECR para cada contenedor utilizado en el flujo de trabajo. Cree imágenes de contenedores para el flujo de trabajo y guárdelas en un repositorio privado, o sincronice el contenido de un registro anterior compatible con su repositorio privado de ECR.
- (Optional) Sentieon licenses:Solicita una Sentieon licencia para usar el Sentieon software en flujos de trabajo privados.

Para archivos de definición de flujo de trabajo de más de 4 MiB (comprimidos), elija una de estas opciones durante la creación del flujo de trabajo:

- Cárguelo en una carpeta de Amazon Simple Storage Service y especifique la ubicación.
- Cargue en un repositorio externo (tamaño máximo de 1 GiB) y especifique los detalles del repositorio.

Tras crear un flujo de trabajo, puede actualizar la siguiente información del flujo de trabajo con la `UpdateWorkflow` operación:

- Name
- Description (Descripción)

- Tipo de almacenamiento predeterminado
- Capacidad de almacenamiento predeterminada (con ID de flujo de trabajo)
- Archivo README.md

Para cambiar otra información del flujo de trabajo, cree un nuevo flujo de trabajo o versión del flujo de trabajo.

Utilice el control de versiones del flujo de trabajo para organizar y estructurar sus flujos de trabajo. Las versiones también le ayudan a gestionar la introducción de actualizaciones iterativas del flujo de trabajo. Para obtener más información acerca de las versiones, consulte Crear una versión de flujo de trabajo.

Temas

- Crear un flujo de trabajo privado
- Actualizar un flujo de trabajo privado
- Eliminar un flujo de trabajo privado
- Compruebe el estado del flujo de trabajo
- Hacer referencia a los archivos del genoma a partir de una definición de flujo de trabajo

Crear un flujo de trabajo privado

Cree un flujo de trabajo mediante la HealthOmics consola, los comandos AWS CLI o uno de los AWS SDKs.

> **Note**
>
> No incluya ninguna información de identificación personal (PII) en los nombres de los flujos de trabajo. Estos nombres están visibles en CloudWatch los registros.

Al crear un flujo de trabajo, HealthOmics asigna un identificador único universal (UUID) al flujo de trabajo. El UUID del flujo de trabajo es un identificador único global (GUID) que es único en todos los flujos de trabajo y versiones del flujo de trabajo. Para la procedencia de los datos, le recomendamos que utilice el UUID del flujo de trabajo para identificar los flujos de trabajo de forma exclusiva.

Si sus tareas de flujo de trabajo utilizan herramientas externas (ejecutables, bibliotecas o scripts), cree estas herramientas en una imagen contenedora. Dispone de las siguientes opciones para alojar la imagen del contenedor:

- Aloje la imagen del contenedor en el registro privado del ECR. Requisitos previos para esta opción:
  - Cree un repositorio privado de ECR o elija uno existente.
  - Configure la política de recursos de ECR como se describe en. Permisos de Amazon ECR
  - Sube la imagen de tu contenedor al repositorio privado.
- Sincronice la imagen del contenedor con el contenido de un registro de terceros compatible. Requisitos previos para esta opción:
  - En el registro privado del ECR, configure una regla de extracción de memoria caché para cada registro ascendente. Para obtener más información, consulte Mapeos de imágenes.
  - Configure la política de recursos de ECR como se describe en. Permisos de Amazon ECR
  - Cree plantillas de creación de repositorios. La plantilla define la configuración para cuando Amazon ECR cree el repositorio privado para un registro ascendente.
  - Cree asignaciones de prefijos para reasignar las referencias de las imágenes del contenedor en la definición del flujo de trabajo a los espacios de nombres de la caché de ECR.

Al crear un flujo de trabajo, se proporciona una definición de flujo de trabajo que contiene información sobre el flujo de trabajo, las ejecuciones y las tareas. HealthOmics puede recuperar la definición del flujo de trabajo como un archivo.zip almacenado localmente o en un bucket de Amazon S3, o desde un repositorio compatible basado en Git.

Temas

- Crear un flujo de trabajo mediante la consola
- Creación de un flujo de trabajo mediante la CLI
- Crear un flujo de trabajo mediante un SDK

Crear un flujo de trabajo mediante la consola

Pasos para crear un flujo de trabajo

1. Abra la consola de HealthOmics .
2. Si es necesario, abra el panel de navegación izquierdo (≡). Elija Flujos de trabajo privados.
3. En la página Flujos de trabajo privados, selecciona Crear flujo de trabajo.

4. En la página Definir flujo de trabajo, proporciona la siguiente información:

   1. Nombre del flujo de trabajo: nombre distintivo de este flujo de trabajo. Recomendamos configurar los nombres de los flujos de trabajo para organizar las ejecuciones en la AWS HealthOmics consola y en CloudWatch los registros.
   2. Descripción (opcional): descripción de este flujo de trabajo.

5. En el panel de definición del flujo de trabajo, proporcione la siguiente información:

   1. Idioma del flujo de trabajo (opcional): seleccione el idioma de especificación del flujo de trabajo. De lo contrario, HealthOmics determina el idioma a partir de la definición del flujo de trabajo.
   2. Para la fuente de definición de flujo de trabajo, elija importar la carpeta de definición desde un repositorio basado en Git, una ubicación de Amazon S3 o desde una unidad local.

      a. Para importar desde un servicio de repositorio:

      > **Note**
      >
      > HealthOmics admite repositorios públicos y privados paraGitHub,GitLab, BitbucketGitHub self-managed,GitLab self-managed.

      i. Elija una conexión para conectar sus AWS recursos al repositorio externo. Para crear una conexión, consulte[Conéctese con repositorios de código externos](#).

      > **Note**
      >
      > Los clientes de la TLV región deben crear una conexión en la región IAD (us-east-1) para crear un flujo de trabajo.

      ii. En ID de repositorio completo, introduce tu ID de repositorio como nombre de usuario/nombre de repositorio. Comprueba que tienes acceso a los archivos de este repositorio.

      iii. En Referencia de origen (opcional), introduce una referencia de origen del repositorio (rama, etiqueta o ID de confirmación). HealthOmics usa la rama predeterminada si no se especifica ninguna referencia a la fuente.

      iv. En Excluir patrones de archivos, introduzca los patrones de archivo para excluir carpetas, archivos o extensiones específicos. Esto ayuda a administrar el tamaño de los

datos al importar archivos del repositorio. Hay un máximo de 50 patrones y los patrones deben seguir la sintaxis del patrón global. Por ejemplo:

A. `tests/`

B. `*.jpeg`

C. `large_data.zip`

b. Para seleccionar la carpeta de definiciones de S3:

i. Introduzca la ubicación de Amazon S3 que contiene la carpeta comprimida de definiciones de flujos de trabajo. El bucket de Amazon S3 debe estar en la misma región que el flujo de trabajo.

ii. Si su cuenta no es propietaria del depósito de Amazon S3, introduzca el ID de AWS cuenta del propietario del depósito en el ID de cuenta del propietario del depósito de S3. Esta información es necesaria para HealthOmics poder verificar la propiedad del bucket.

c. Para seleccionar la carpeta de definiciones de una fuente local:

i. Introduzca la ubicación de la unidad local de la carpeta comprimida de definiciones de flujo de trabajo.

3. Ruta principal del archivo de definición del flujo de trabajo (opcional): introduzca la ruta del archivo desde la carpeta o el repositorio de definiciones de flujo de trabajo comprimido hasta el `main` archivo. Este parámetro no es necesario si solo hay un archivo en la carpeta de definición del flujo de trabajo o si el archivo principal se denomina «principal».

6. En el panel del archivo README (opcional), seleccione la fuente del archivo README y proporcione la siguiente información:

- Para Importar desde un servicio de repositorio, en la ruta del archivo README, introduzca la ruta al archivo README del repositorio.
- En Seleccione un archivo de S3, en el archivo README de S3, introduzca el URI de Amazon S3 para el archivo README.
- En Seleccionar un archivo de una fuente local: en README (opcional), elija Elegir archivo para seleccionar el archivo marcado (.md) que desea cargar.

7. En el panel de configuración del almacenamiento de ejecución predeterminado, proporciona el tipo de almacenamiento de ejecución predeterminado y la capacidad para las ejecuciones que utilizan este flujo de trabajo:

1. Tipo de almacenamiento de ejecución: elija si desea utilizar el almacenamiento estático o dinámico como predeterminado para el almacenamiento de ejecución temporal. El valor predeterminado es el almacenamiento estático.

2. Capacidad de almacenamiento de ejecución (opcional): para el tipo de almacenamiento de ejecución estático, puede introducir la cantidad predeterminada de almacenamiento de ejecución necesaria para este flujo de trabajo. El valor predeterminado de este parámetro es 1200 GiB. Puede anular estos valores predeterminados al iniciar una ejecución.

8. Etiquetas (opcional): puedes asociar hasta 50 etiquetas a este flujo de trabajo.
9. Elija Siguiente.
10. En la página Añadir parámetros de flujo de trabajo (opcional), seleccione la fuente de parámetros:

    1. Para analizar desde un archivo de definición de flujo de trabajo, HealthOmics analizará automáticamente los parámetros del flujo de trabajo del archivo de definición del flujo de trabajo.
    2. Para Proporcionar una plantilla de parámetros desde el repositorio de Git, usa la ruta al archivo de plantillas de parámetros de tu repositorio.
    3. Para seleccionar un archivo JSON de una fuente local, sube un JSON archivo de una fuente local que especifique los parámetros.
    4. En Introducir manualmente los parámetros del flujo de trabajo, introduzca manualmente los nombres y las descripciones de los parámetros.
11. En el panel de vista previa de los parámetros, puede revisar o cambiar los parámetros de esta versión del flujo de trabajo. Si restaura el JSON archivo, perderá los cambios locales que haya realizado.
12. Elija Siguiente.
13. En la página de reasignación de URI del contenedor, en el panel Reglas de mapeo, puede definir las reglas de mapeo de URI para su flujo de trabajo.

    En Fuente del archivo de mapeo, selecciona una de las siguientes opciones:

    - Ninguna: no se requieren reglas de mapeo.
    - Seleccione el archivo JSON de S3: especifique la ubicación en S3 para el archivo de mapeo.
    - Seleccione el archivo JSON de una fuente local: especifique la ubicación del archivo de mapeo en su dispositivo local.
    - Introduzca los mapeos manualmente: introduzca los mapeos de registro y los mapeos de imágenes en el panel de mapeos.
14. La consola muestra el panel de mapeos. Si elige un archivo fuente de mapeo, la consola muestra los valores del archivo.

a. En las asignaciones del registro, puede editar las asignaciones o añadir asignaciones (máximo 20 asignaciones del registro).

   Cada mapeo de registro contiene los siguientes campos:

   - URL del registro ascendente: el URI del registro ascendente.
   - Prefijo del repositorio ECR: el prefijo del repositorio que se utilizará en el repositorio privado de Amazon ECR.
   - (Opcional) Prefijo del repositorio ascendente: el prefijo del repositorio en el registro ascendente.
   - (Opcional) ID de cuenta ECR: ID de cuenta de la cuenta propietaria de la imagen del contenedor ascendente.

b. En los mapeos de imágenes, puede editar los mapeos de imágenes o añadir mapeos (máximo 100 mapeos de imágenes).

   Cada mapeo de imágenes contiene los siguientes campos:

   - Imagen de origen: especifica el URI de la imagen de origen en el registro original.
   - Imagen de destino: especifica el URI de la imagen correspondiente en el registro privado de Amazon ECR.

15. Elija Siguiente.
16. Revise la configuración del flujo de trabajo y, a continuación, seleccione Crear flujo de trabajo.

Creación de un flujo de trabajo mediante la CLI

Si los archivos de flujo de trabajo y el archivo de plantilla de parámetros están en su máquina local, puede crear un flujo de trabajo mediante el siguiente comando CLI.

```
aws omics create-workflow  \
  --name "my_workflow"   \
  --definition-zip fileb://my-definition.zip \
  --parameter-template file://my-parameter-template.json
```

La `create-workflow` operación devuelve la siguiente respuesta:

```
{
  "arn": "arn:aws:omics:us-west-2:....",
```

```
  "id": "1234567",
  "status": "CREATING",
  "tags": {
      "resourceArn": "arn:aws:omics:us-west-2:...."
  },
  "uuid": "64c9a39e-8302-cc45-0262-2ea7116d854f"
}
```

Parámetros opcionales que se deben usar al crear un flujo de trabajo

Puede especificar cualquiera de los parámetros opcionales al crear un flujo de trabajo. Para obtener información sobre la sintaxis, consulte CreateWorkflowla referencia de la HealthOmics API de AWS.

Temas

- Especifique la ubicación de Amazon S3 de la definición del flujo de trabajo
- Usa la definición de flujo de trabajo de un repositorio basado en Git
- Especifique un archivo readme
- mainEspecifique el archivo de definición
- Especifique el tipo de almacenamiento de ejecución
- Especifique la configuración de la GPU
- Configure los parámetros de mapeo de la memoria caché extraíble

Especifique la ubicación de Amazon S3 de la definición del flujo de trabajo

Si el archivo de definición de flujo de trabajo se encuentra en una carpeta de Amazon S3, especifique la ubicación mediante el `definition-uri` parámetro, como se muestra en el siguiente ejemplo. Si su cuenta no es propietaria del bucket de Amazon S3, proporcione el Cuenta de AWS ID del propietario.

```
aws omics create-workflow  \
  --name Test  \
  --definition-uri s3://omics-bucket/workflow-definition/  \
  --owner-id  123456789012
    ...
```

Usa la definición de flujo de trabajo de un repositorio basado en Git

Para usar la definición de flujo de trabajo de un repositorio compatible basado en Git, usa el `definition-repository` parámetro en tu solicitud. No proporciones ningún otro `definition` parámetro, ya que una solicitud falla si incluye más de una fuente de entrada.

El `definition-respository` parámetro contiene los siguientes campos:

- connectionArn— ARN de la conexión de código que conecta los recursos de AWS al repositorio externo.
- fullRepositoryId— Introduzca el ID del repositorio como`owner-name/repo-name`. Compruebe que tiene acceso a los archivos de este repositorio.
- sourceReference(Opcional): introduzca un tipo de referencia de repositorio (BRANCH, TAG o COMMIT) y un valor.

  HealthOmics usa la última confirmación de la rama predeterminada si no especificas una referencia de origen.
- excludeFilePatterns(Opcional): introduce los patrones de los archivos para excluir carpetas, archivos o extensiones específicos. Esto ayuda a administrar el tamaño de los datos al importar archivos del repositorio. Proporcione un máximo de 50 patrones. Los patrones deben seguir la sintaxis del patrón global. Por ejemplo:
  - `tests/`
  - `*.jpeg`
  - `large_data.zip`

Al especificar la definición del flujo de trabajo desde un repositorio basado en Git, utilícela `parameter-template-path` para especificar el archivo de plantilla de parámetros. Si no proporciona este parámetro, HealthOmics crea el flujo de trabajo sin una plantilla de parámetros.

El siguiente ejemplo muestra los parámetros relacionados con el contenido de un repositorio privado basado en Git:

```
aws omics create-workflow \
    --name custom-variant \
    --description "Custom variant calling pipeline" \
    --engine "WDL" \
    --definition-repository '{
```

```
            "connectionArn": "arn:aws:codeconnections:us-
east-1:123456789012:connection/abcd1234-5678-90ab-cdef-1234567890ab",
            "fullRepositoryId": "myorg/my-genomics-workflows",
            "sourceReference": {
                "type": "BRANCH",
                "value": "main"
            },
            "excludeFilePatterns": ["tests/**", "*.log"]
      }' \
    --main "workflows/variant-calling/main.wdl" \
    --parameter-template-path "parameters/variant-calling-params.json" \
    --readme-path "docs/variant-calling-README.md" \
    --storage-type "DYNAMIC" \
```

Para ver más ejemplos, consulte la entrada del blog Cómo crear HealthOmics flujos de trabajo de AWS a partir del contenido de Git.

Especifique un archivo readme

Puede especificar la ubicación del archivo README mediante uno de los siguientes parámetros:

- readme-markdown— Entrada de cadena o un archivo en su máquina local.
- readme-uri— El URI de un archivo almacenado en S3.
- readme-path — La ruta al archivo README del repositorio.

Utilice readme-path solo junto con definition-respository. Si no especificas ningún parámetro README, HealthOmics importa el archivo README.md de nivel raíz en el repositorio (si existe).

Los siguientes ejemplos muestran cómo especificar la ubicación del archivo README mediante readme-path y readme-uri.

```
# Using README from repository
aws omics create-workflow \
    --name "documented-workflow" \
    --definition-repository '...' \
    --readme-path "docs/workflow-guide.md"

# Using README from S3
aws omics create-workflow \
    --name "s3-readme-workflow" \
    --definition-repository '...' \
```

```
    --readme-uri "s3://my-bucket/workflow-docs/readme.md"
```

Para obtener más información, consulte HealthOmics Archivos README de flujo de trabajo.

mainEspecifique el archivo de definición

Si incluye varios archivos de definición del flujo de trabajo, utilice el `main` parámetro para especificar el archivo de definición principal del flujo de trabajo.

```
aws omics create-workflow  \
  --name Test  \
  --main multi_workflow/workflow2.wdl  \
    ...
```

Especifique el tipo de almacenamiento de ejecución

Puede especificar el tipo de almacenamiento de ejecución predeterminado (DINÁMICO o ESTÁTICO) y la capacidad de almacenamiento de ejecución (necesaria para el almacenamiento estático). Para obtener más información sobre los tipos de almacenamiento de ejecución, consulteEjecute tipos de almacenamiento en HealthOmics flujos de trabajo.

```
aws omics create-workflow  \
  --name my_workflow   \
  --definition-zip fileb://my-definition.zip \
  --parameter-template file://my-parameter-template.json   \
  --storage-type 'STATIC'  \
  --storage-capacity 1200  \
```

Especifique la configuración de la GPU

Usa el parámetro accelerators para crear un flujo de trabajo que se ejecute en una instancia de procesamiento acelerado. En el siguiente ejemplo, se muestra cómo utilizar el parámetro. `accelerators` La configuración de la GPU se especifica en la definición del flujo de trabajo. Consulte Instancias de computación acelerada.

```
aws omics create-workflow --name workflow name \
   --definition-uri s3://amzn-s3-demo-bucket1/GPUWorkflow.zip \
   --accelerators GPU
```

Configure los parámetros de mapeo de la memoria caché extraíble

Si utiliza la función de mapeo de caché extraíble de Amazon ECR, puede anular los mapeos predeterminados. Para obtener más información sobre los parámetros de configuración del contenedor, consulte. Imágenes de contenedores para flujos de trabajo privados

En el siguiente ejemplo, el archivo `mappings.json` contiene este contenido:

```
{
    "registryMappings": [
        {
            "upstreamRegistryUrl": "registry-1.docker.io",
            "ecrRepositoryPrefix": "docker-hub"
        },
        {
            "upstreamRegistryUrl": "quay.io",
            "ecrRepositoryPrefix": "quay",
            "accountId": "123412341234"
        },
        {

            "upstreamRegistryUrl": "public.ecr.aws",
            "ecrRepositoryPrefix": "ecr-public"
        }
    ],
    "imageMappings": [{
            "sourceImage": "docker.io/library/ubuntu:latest",
            "destinationImage": "healthomics-docker-2/custom/ubuntu:latest",
            "accountId": "123412341234"
        },
        {
            "sourceImage": "nvcr.io/nvidia/k8s/dcgm-exporter",
            "destinationImage": "healthomics-nvidia/k8s/dcgm-exporter"
        }
    ]
}
```

Especifique los parámetros de mapeo en el comando create-workflow:

```
aws omics create-workflow  \
     ...
--container-registry-map-file file://mappings.json
```

```
    ...
```

También puede especificar la ubicación en S3 del archivo de parámetros de mapeo:

```
aws omics create-workflow  \
     ...
--container-registry-map-uri s3://amzn-s3-demo-bucket1/test.zip
    ...
```

Crear un flujo de trabajo mediante un SDK

Puede crear un flujo de trabajo mediante uno de los SDKs. El siguiente ejemplo muestra cómo crear un flujo de trabajo mediante el SDK de Python.

```
import boto3

omics = boto3.client('omics')

with open('definition.zip', 'rb') as f:
   definition = f.read()

response = omics.create_workflow(
   name='my_workflow',
   definitionZip=definition,
   parameterTemplate={ ... }
)
```

Actualizar un flujo de trabajo privado

Puede actualizar un flujo de trabajo mediante la HealthOmics consola, los comandos AWS CLI o uno de los AWS SDKs.

> **Note**
>
> No incluya ninguna información de identificación personal (PII) en los nombres de los flujos de trabajo. Estos nombres están visibles en CloudWatch los registros.

Temas

- Actualización de un flujo de trabajo mediante la consola

- Actualización de un flujo de trabajo mediante la CLI
- Actualizar un flujo de trabajo mediante un SDK

Actualización de un flujo de trabajo mediante la consola

Pasos para actualizar un flujo de trabajo

1. Abra la consola de HealthOmics .
2. Si es necesario, abra el panel de navegación izquierdo (≡). Elija Flujos de trabajo privados.
3. En la página Flujos de trabajo privados, elija el flujo de trabajo que desee actualizar.
4. En la página de flujo de trabajo:
   - Si el flujo de trabajo tiene versiones, asegúrese de seleccionar la versión predeterminada.
   - Elija Editar seleccionado en la lista de acciones.
5. En la página Editar flujo de trabajo, puede cambiar cualquiera de los siguientes valores:
   - Nombre del flujo de trabajo.
   - Descripción del flujo de trabajo.
   - El tipo de almacenamiento de ejecución predeterminado para el flujo de trabajo.
   - La capacidad de almacenamiento de ejecución predeterminada (si el tipo de almacenamiento de ejecución es almacenamiento estático). Para obtener más información sobre la configuración de almacenamiento de ejecución predeterminada, consulteCrear un flujo de trabajo mediante la consola.
6. Seleccione Guardar cambios para aplicar los cambios.

Actualización de un flujo de trabajo mediante la CLI

Como se muestra en el siguiente ejemplo, puede actualizar el nombre y la descripción del flujo de trabajo. También puede cambiar el tipo de almacenamiento de ejecución predeterminado (ESTÁTICO o DINÁMICO) y la capacidad de almacenamiento de ejecución (para el tipo de almacenamiento estático). Para obtener más información sobre los tipos de almacenamiento de ejecución, consulteEjecute tipos de almacenamiento en HealthOmics flujos de trabajo.

```
aws omics update-workflow    \
  --id 1234567    \
  --name my_workflow      \
```

```
  --description "updated workflow"    \
  --storage-type 'STATIC'    \
  --storage-capacity 1200
```

No recibes una respuesta a la `update-workflow` solicitud.

Actualizar un flujo de trabajo mediante un SDK

Puede actualizar un flujo de trabajo mediante uno de los SDKs.

El siguiente ejemplo muestra cómo actualizar un flujo de trabajo mediante el SDK de Python.

```
import boto3

omics = boto3.client('omics')

response = omics.update_workflow(
   name='my_workflow',
   description='updated workflow'
)
```

Eliminar un flujo de trabajo privado

Cuando ya no necesite un flujo de trabajo, puede eliminarlo mediante la HealthOmics consola, los comandos AWS CLI o uno de los AWS SDKs. Puede eliminar un flujo de trabajo que cumpla los siguientes criterios:

- Su estado es ACTIVO o FALLIDO.
- No tiene acciones activas.
- Has eliminado todas las versiones del flujo de trabajo.

La eliminación de un flujo de trabajo no afecta a las ejecuciones en curso que estén utilizando el flujo de trabajo.

Temas

- Eliminar un flujo de trabajo mediante la consola
- Eliminar un flujo de trabajo mediante la CLI
- Eliminar un flujo de trabajo mediante un SDK

Eliminar un flujo de trabajo mediante la consola

Para eliminar un flujo de trabajo

1. Abra la consola de HealthOmics .
2. Si es necesario, abra el panel de navegación izquierdo (≡). Elija Flujos de trabajo privados.
3. En la página Flujos de trabajo privados, elija el flujo de trabajo que desee eliminar.
4. En la página de flujo de trabajo, elija Eliminar lo seleccionado en la lista de acciones.
5. En el modal Eliminar flujo de trabajo, introduce «confirmar» para confirmar la eliminación.
6. Elija Eliminar.

Eliminar un flujo de trabajo mediante la CLI

El siguiente ejemplo muestra cómo se puede utilizar el AWS CLI comando para eliminar un flujo de trabajo. Para ejecutar el ejemplo, sustituya el *workflow id* identificador por el ID del flujo de trabajo que desee eliminar.

```
aws omics delete-workflow
  --id workflow id
```

HealthOmics no envía una respuesta a la `delete-workflow` solicitud.

Eliminar un flujo de trabajo mediante un SDK

Puede eliminar un flujo de trabajo mediante uno de los SDKs.

El siguiente ejemplo muestra cómo eliminar un flujo de trabajo mediante el SDK de Python.

```
import boto3

omics = boto3.client('omics')

response = omics.delete_workflow(
   id='1234567'
)
```

Compruebe el estado del flujo de trabajo

Tras crear el flujo de trabajo, puede verificar el estado y ver otros detalles del flujo de trabajo mediante get-workflow, como se muestra.

```
aws omics get-workflow --id 1234567
```

La respuesta incluye los detalles del flujo de trabajo, incluido el estado, como se muestra.

```
{
    "arn": "arn:aws:omics:us-west-2:....",
    "creationTime": "2022-07-06T00:27:05.542459"
    "id": "1234567",
    "engine": "WDL",
    "status": "ACTIVE",
    "type": "PRIVATE",
    "main": "workflow-crambam.wdl",
    "name": "workflow_name",
    "storageType": "STATIC",
    "storageCapacity": "1200",
    "uuid": "64c9a39e-8302-cc45-0262-2ea7116d854f"
  }
```

Puede iniciar una ejecución con este flujo de trabajo una vez que el estado pase a`ACTIVE`.

Hacer referencia a los archivos del genoma a partir de una definición de flujo de trabajo

Se puede hacer HealthOmics referencia a un objeto de almacén de referencia con un URI como el siguiente. Usa el tuyo propio *`account ID`* *`reference store ID`* y *`reference ID`* donde se indique.

```
omics://account ID.storage.us-west-2.amazonaws.com/reference store id/reference/id
```

Algunos flujos de trabajo requerirán tanto `SOURCE` los `INDEX` archivos como los del genoma de referencia. El URI anterior es la forma abreviada predeterminada y, de forma predeterminada, será el archivo FUENTE. Para especificar cualquiera de los archivos, puede utilizar el formulario URI largo, de la siguiente manera.

```
omics://account ID.storage.us-west-2.amazonaws.com/reference store id/reference/id/
source
omics://account ID.storage.us-west-2.amazonaws.com/reference store id/reference/id/
index
```

El uso de un conjunto de lecturas secuenciales tendría un patrón similar, como se muestra.

```
aws omics create-workflow \
     --name workflow name \
     --main sample workflow.wdl \
     --definition-uri omics://account ID.storage.us-
west-2.amazonaws.com/sequence_store_id/readSet/id \
     --parameter-template file://parameters_sample_description.json
```

Algunos conjuntos de lecturas, como los basados en FASTQ, pueden contener lecturas emparejadas. En los siguientes ejemplos, se denominan SOURCE1 y SOURCE2. Los formatos como BAM y CRAM solo tendrán un SOURCE1 archivo. Algunos conjuntos de lectura contendrán archivos de índice, como archivos `bai` `ocrai`. El URI anterior es la forma abreviada predeterminada y se utilizará de forma predeterminada en el SOURCE1 archivo. Para especificar el archivo o índice exacto, puede utilizar el formulario URI largo, de la siguiente manera.

```
omics://123456789012.storage.us-west-2.amazonaws.com/<sequence_store_id>/readSet/<id>/
source1
omics://123456789012.storage.us-west-2.amazonaws.com/<sequence_store_id>/readSet/<id>/
source2
omics://123456789012.storage.us-west-2.amazonaws.com/<sequence_store_id>/readSet/<id>/
index
```

El siguiente es un ejemplo de un archivo JSON de entrada que usa dos Omics Storage URIs.

```
{
   "input_fasta": "omics://123456789012.storage.us-west-2.amazonaws.com/
<reference_store_id>/reference/<id>",
   "input_cram": "omics://123456789012.storage.us-west-2.amazonaws.com/
<sequence_store_id>/readSet/<id>"
}
```

Haga referencia al archivo JSON de entrada en el AWS CLI agregándolo `--inputs file://<input_file.json>` a su solicitud de inicio y ejecución.

Control de versiones del flujo de trabajo en HealthOmics

Si necesita realizar cambios en un flujo de trabajo, puede crear uno nuevo o una nueva versión del flujo de trabajo. Las versiones son inmutables, excepto en lo que respecta a los cambios de configuración permitidos que no afectan a la lógica de ejecución.

Las versiones de flujo de trabajo ofrecen las siguientes ventajas:

- Las versiones forman un grupo lógico de flujos de trabajo relacionados. Puede añadir un nombre definido por el usuario a cada versión del flujo de trabajo para gestionarlas más fácilmente (especialmente en el caso de un flujo de trabajo con un gran número de versiones).
- Puede ejecutar varias versiones de un flujo de trabajo al mismo tiempo.
- Todas las versiones de un flujo de trabajo comparten el mismo ID de flujo de trabajo y ARN base, lo que puede simplificar la administración de canalizaciones después de modificar un flujo de trabajo.
- Las versiones de los flujos de trabajo proporcionan el mismo nivel de procedencia de los datos que los flujos de trabajo. Las versiones son inmutables y HealthOmics crean un ARN único para cada versión del flujo de trabajo. El ARN de la versión incluye el ID del flujo de trabajo y el nombre de la versión, como se muestra en el siguiente ejemplo:

  ```
  arn:aws:omics:us-west-2:123456789012:workflow/1234567/version/
  myUniqueVersionName
  ```
- Si tienes un flujo de trabajo compartido, puedes actualizarlo sin interrumpir a los suscriptores (que pueden seguir usando la versión anterior). Los suscriptores pueden acceder a todas las versiones del flujo de trabajo. Si creas una nueva versión, no necesitas volver a compartir el flujo de trabajo.
- Al iniciar la ejecución de un flujo de trabajo, puede especificar la versión del flujo de trabajo.
  - Los usuarios pueden optar por permanecer en una versión estable durante las ejecuciones de producción y probar la última versión para una ejecución de prueba.
  - Los usuarios pueden volver a la versión anterior de un flujo de trabajo si tienen problemas con la nueva versión.
  - Los suscriptores de un flujo de trabajo compartido pueden elegir qué versión usar.

Temas

- Versión de flujo de trabajo predeterminada
- Crear una versión de flujo de trabajo
- Actualizar una versión del flujo de trabajo
- Eliminar una versión del flujo de trabajo

Versión de flujo de trabajo predeterminada

Después de crear una o más versiones de un flujo de trabajo, HealthOmics trata el flujo de trabajo original como la versión predeterminada. Al iniciar una ejecución, si lo desea, puede especificar una versión del flujo de trabajo para la ejecución. Si no especifica una versión al iniciar una ejecución, HealthOmics utiliza la versión predeterminada.

En la consola, HealthOmics indica el flujo de trabajo original con una etiqueta de versión predeterminada. La consola usa esta etiqueta solo después de crear una o más versiones del flujo de trabajo. El flujo de trabajo original siempre es la versión predeterminada. No puede asignar ninguna otra versión como predeterminada.

No puedes eliminar la versión predeterminada de un flujo de trabajo si hay otras versiones asociadas al flujo de trabajo. Para obtener más información, consulte Eliminar un flujo de trabajo privado.

Crear una versión de flujo de trabajo

Al crear una nueva versión de un flujo de trabajo, debe especificar los valores de configuración de la nueva versión. No hereda ningún valor de configuración del flujo de trabajo.

Al crear la versión, proporcione un nombre de versión que sea exclusivo para este flujo de trabajo. No puede cambiar el nombre después de HealthOmics crear la versión.

El nombre de la versión debe empezar por una letra o un número y puede incluir letras mayúsculas y minúsculas, números, guiones, puntos y guiones bajos. La longitud máxima es de 64 caracteres. Por ejemplo, puede usar un esquema de nomenclatura simple, como versión1, versión2 o versión3. También puede hacer coincidir las versiones de su flujo de trabajo con sus propias convenciones de control de versiones internas, como 2.7.0, 2.7.1 o 2.7.2.

Si lo desea, utilice el campo de descripción de la versión para añadir notas sobre esta versión. Por ejemplo: Fix for syntax error in workflow definition.

> **Note**
>
> No incluyas ningún dato de identificación personal (PII) en el nombre de la versión. Los nombres de las versiones aparecen en el ARN de la versión del flujo de trabajo.

HealthOmics asigna un ARN único a la versión del flujo de trabajo. El ARN es único en función de la combinación del ID del flujo de trabajo y el nombre de la versión.

> **Warning**
>
> Tras eliminar una versión del flujo de trabajo, HealthOmics permite reutilizar el nombre de la versión para otra versión del flujo de trabajo. La práctica recomendada es no volver a utilizar los nombres de las versiones. Si reutilizas un nombre, el flujo de trabajo y cada versión tienen un UUID único que puedes usar para determinar su procedencia.

Temas

- Crea una versión del flujo de trabajo mediante la consola
- Cree una versión de flujo de trabajo mediante la CLI
- Cree una versión del flujo de trabajo mediante un SDK
- Compruebe el estado de una versión de flujo de trabajo

Crea una versión del flujo de trabajo mediante la consola

Pasos para crear una versión de flujo de trabajo

1. Abra la consola de HealthOmics .
2. Si es necesario, abra el panel de navegación izquierdo (≡). Elija Flujos de trabajo privados.
3. En la página Flujos de trabajo privados, elija el flujo de trabajo para la nueva versión.
4. En la página de detalles del flujo de trabajo, selecciona Crear nueva versión.
5. En la página Crear versión, proporcione la siguiente información:
    1. Nombre de la versión: introduzca un nombre para la versión del flujo de trabajo que sea único en todo el flujo de trabajo.
    2. Descripción de la versión (opcional): puede utilizar el campo de descripción para añadir notas sobre esta versión.
6. En el panel de definición del flujo de trabajo, proporciona la siguiente información:
    1. Idioma del flujo de trabajo (opcional): seleccione el idioma de especificación para la versión del flujo de trabajo. De lo contrario, HealthOmics determina el idioma a partir de la definición del flujo de trabajo.
    2. Para la fuente de definición de flujo de trabajo, elija importar la carpeta de definición desde un repositorio basado en Git, una ubicación de Amazon S3 o desde una unidad local.

a. Para importar desde un servicio de repositorio:

> **Note**
>
> HealthOmics admite repositorios públicos y privados paraGitHub,GitLab, BitbucketGitHub self-managed,GitLab self-managed.

i. Elija una conexión para conectar sus AWS recursos al repositorio externo. Para crear una conexión, consulte[Conéctese con repositorios de código externos](#).

> **Note**
>
> Los clientes de la TLV región deben crear una conexión en la región IAD (us-east-1) para crear un flujo de trabajo.

ii. En ID de repositorio completo, introduce tu ID de repositorio como nombre de usuario/nombre de repositorio. Comprueba que tienes acceso a los archivos de este repositorio.

iii. En Referencia de origen (opcional), introduce una referencia de origen del repositorio (rama, etiqueta o ID de confirmación). HealthOmics usa la rama predeterminada si no se especifica ninguna referencia a la fuente.

iv. En Excluir patrones de archivos, introduzca los patrones de archivo para excluir carpetas, archivos o extensiones específicos. Esto ayuda a administrar el tamaño de los datos al importar archivos del repositorio. Hay un máximo de 50 patrones y los patrones deben seguir la sintaxis del [patrón global](#). Por ejemplo:

A. `tests/`

B. `*.jpeg`

C. `large_data.zip`

b. Para seleccionar la carpeta de definiciones de S3:

i. Introduzca la ubicación de Amazon S3 que contiene la carpeta comprimida de definiciones de flujos de trabajo. El bucket de Amazon S3 debe estar en la misma región que el flujo de trabajo.

ii. Si su cuenta no es propietaria del depósito de Amazon S3, introduzca el ID de AWS cuenta del propietario del depósito en el ID de cuenta del propietario del depósito de S3. Esta información es necesaria para HealthOmics poder verificar la propiedad del bucket.

c. Para seleccionar la carpeta de definiciones de una fuente local:

i. Introduzca la ubicación de la unidad local de la carpeta comprimida de definiciones de flujo de trabajo.

3. Ruta principal del archivo de definición del flujo de trabajo (opcional): introduzca la ruta del archivo desde la carpeta o el repositorio de definiciones de flujo de trabajo comprimido hasta el `main` archivo. Este parámetro no es necesario si solo hay un archivo en la carpeta de definición del flujo de trabajo o si el archivo principal se denomina «principal».

7. En el panel del archivo README (opcional), seleccione la fuente del archivo README y proporcione la siguiente información:

- Para Importar desde un servicio de repositorio, en la ruta del archivo README, introduzca la ruta al archivo README del repositorio.
- En Seleccione un archivo de S3, en el archivo README de S3, introduzca el URI de Amazon S3 para el archivo README.
- En Seleccionar un archivo de una fuente local: en README (opcional), elija Elegir archivo para seleccionar el archivo marcado (.md) que desea cargar.

8. En el panel de configuración del almacenamiento de ejecución predeterminado, proporciona el tipo de almacenamiento de ejecución predeterminado y la capacidad para las ejecuciones que utilizan este flujo de trabajo:

1. Tipo de almacenamiento de ejecución: elija si desea utilizar el almacenamiento estático o dinámico como predeterminado para el almacenamiento de ejecución temporal. El valor predeterminado es el almacenamiento estático.
2. Capacidad de almacenamiento de ejecución (opcional): para el tipo de almacenamiento de ejecución estático, puede introducir la cantidad predeterminada de almacenamiento de ejecución necesaria para este flujo de trabajo. El valor predeterminado de este parámetro es 1200 GiB. Puede anular estos valores predeterminados al iniciar una ejecución.

9. Etiquetas (opcional): puedes asociar hasta 50 etiquetas a esta versión del flujo de trabajo.

10. Elija Siguiente.

11. En la página Añadir parámetros de flujo de trabajo (opcional), seleccione la fuente de parámetros:

1. Para analizar desde un archivo de definición de flujo de trabajo, HealthOmics analizará automáticamente los parámetros del flujo de trabajo del archivo de definición del flujo de trabajo.

2. Para Proporcionar una plantilla de parámetros desde el repositorio de Git, usa la ruta al archivo de plantillas de parámetros de tu repositorio.
3. Para seleccionar un archivo JSON de una fuente local, sube un JSON archivo de una fuente local que especifique los parámetros.
4. En Introducir manualmente los parámetros del flujo de trabajo, introduzca manualmente los nombres y las descripciones de los parámetros.

12. En el panel de vista previa de los parámetros, puede revisar o cambiar los parámetros de esta versión del flujo de trabajo. Si restaura el JSON archivo, perderá los cambios locales que haya realizado.
13. En la página de reasignación de URI del contenedor, en el panel Reglas de mapeo, puede definir las reglas de mapeo de URI para su flujo de trabajo.

    En Fuente del archivo de mapeo, selecciona una de las siguientes opciones:

    - Ninguna: no se requieren reglas de mapeo.
    - Seleccione el archivo JSON de S3: especifique la ubicación en S3 para el archivo de mapeo.
    - Seleccione el archivo JSON de una fuente local: especifique la ubicación del archivo de mapeo en su dispositivo local.
    - Introduzca los mapeos manualmente: introduzca los mapeos de registro y los mapeos de imágenes en el panel de mapeos.

14. La consola muestra el panel de mapeos. Si elige un archivo fuente de mapeo, la consola muestra los valores del archivo.

    a. En las asignaciones del registro, puede editar las asignaciones o añadir asignaciones (máximo 20 asignaciones del registro).

       Cada mapeo de registro contiene los siguientes campos:

       - URL del registro ascendente: el URI del registro ascendente.
       - Prefijo del repositorio ECR: el prefijo del repositorio que se utilizará en el repositorio privado de Amazon ECR.
       - (Opcional) Prefijo del repositorio ascendente: el prefijo del repositorio en el registro ascendente.
       - (Opcional) ID de cuenta ECR: ID de cuenta de la cuenta propietaria de la imagen del contenedor ascendente.

b. En los mapeos de imágenes, puede editar los mapeos de imágenes o añadir mapeos (máximo 100 mapeos de imágenes).

   Cada mapeo de imágenes contiene los siguientes campos:

   - Imagen de origen: especifica el URI de la imagen de origen en el registro original.
   - Imagen de destino: especifica el URI de la imagen correspondiente en el registro privado de Amazon ECR.

15. Elija Siguiente.
16. Revise la configuración de la versión y, a continuación, seleccione Crear versión.

Cuando se crea la versión, la consola vuelve a la página de detalles del flujo de trabajo y muestra la nueva versión en la tabla Flujos de trabajo y versiones.

Cree una versión de flujo de trabajo mediante la CLI

Puede crear una versión del flujo de trabajo mediante la operación de `CreateWorkflowVersion` API. Para los parámetros opcionales, HealthOmics utiliza los siguientes valores predeterminados:

Parámetro	Predeterminado/a
Motor	Determinado a partir de la definición del flujo de trabajo
Tipo de almacenamiento	STATIC
Capacidad de almacenamiento (para almacenamiento estático)	1200 GiB
Principal	Se determina en función del contenido de la carpeta de definición del flujo de trabajo. Para obtener más información, consulte HealthOmics requisitos de definición del flujo de trabajo.
Aceleradores	none
Tags	none

El siguiente ejemplo de CLI crea una versión de flujo de trabajo con almacenamiento estático como almacenamiento de ejecución predeterminado:

```
aws omics create-workflow-version \
--workflow-id 1234567  \
--version-name "my_version" \
--engine WDL \
--definition-zip fileb://workflow-crambam.zip \
--description "my version description" \
--main file://workflow-params.json \
--parameter-template file://workflow-params.json \
--storage-type='STATIC'   \
--storage-capacity 1200   \
--tags example123=string  \
--accelerators GPU
```

Si el archivo de definición de flujo de trabajo se encuentra en una carpeta de Amazon S3, introduzca la ubicación mediante el `definition-uri` parámetro en lugar de`definition-zip`. Para obtener más información, consulte CreateWorkflowVersionla referencia de la HealthOmics API de AWS.

Recibirá la siguiente respuesta a la `create-workflow-version` solicitud.

```
{
  "workflowId": "1234567",
  "versionName": "my_version",
  "arn": "arn:aws:omics:us-west-2:123456789012:workflow/1234567/version/3",
  "status": "ACTIVE",
  "tags": {
    "environment": "production",
    "owner": "team-alpha"
  },
  "uuid": "0ac9a563-355c-fc7a-1b47-a115167af8a2"
}
```

Cree una versión del flujo de trabajo mediante un SDK

Puede crear un flujo de trabajo mediante uno de los SDKs.

El siguiente ejemplo muestra cómo crear una versión de flujo de trabajo mediante el SDK de Python.

```
import boto3
```

```
omics = boto3.client('omics')

with open('definition.zip', 'rb') as f:
   definition = f.read()

response = omics.create_workflow_version(
   workflowId='1234567',
   versionName='my_version',
   requestId='my_request_1'
   definitionZip=definition,
   parameterTemplate={ ... }
)
```

Compruebe el estado de una versión de flujo de trabajo

Después de crear la versión del flujo de trabajo, puede verificar el estado y ver otros detalles del flujo de trabajo mediante get-workflow-versionla siguiente información.

```
aws omics get-workflow-version
--workflow-id 9876543
--version-name "my_version"
```

La respuesta proporciona los detalles del flujo de trabajo, incluido el estado, tal y como se muestra.

```
{
  "workflowId": "1234567",
  "versionName": "3.0.0",
  "arn": "arn:aws:omics:us-west-2:123456789012:workflow/1234567/version/3.0.0",
  "status": "ACTIVE",
  "description": ...
  "uuid": "0ac9a563-355c-fc7a-1b47-a115167af8a2"
}
```

Para poder iniciar una ejecución con esta versión del flujo de trabajo, el estado debe pasar a`ACTIVE`.

Actualizar una versión del flujo de trabajo

Puede actualizar la descripción y la configuración de almacenamiento de ejecución predeterminada para una versión de flujo de trabajo privado. Para cambiar cualquier otra información de la versión del flujo de trabajo, cree una nueva versión.

Temas

- Actualice una versión del flujo de trabajo mediante la consola
- Actualizar una versión de flujo de trabajo mediante la CLI
- Actualiza una versión del flujo de trabajo mediante un SDK

Actualice una versión del flujo de trabajo mediante la consola

Para actualizar una versión del flujo de trabajo

1. Abra la consola de HealthOmics .
2. Si es necesario, abra el panel de navegación izquierdo (≡). Elija Flujos de trabajo privados.
3. En la página Flujos de trabajo privados, elija el flujo de trabajo.
4. En la página Flujo de trabajo, elija la versión del flujo de trabajo que desee actualizar y seleccione Editar en la lista de acciones.
   - Si elige la versión predeterminada, la consola abre la página Editar flujo de trabajo. Para obtener más información, consulte Actualizar un flujo de trabajo privado.
   - Si elige una versión definida por el usuario, la consola abre la página Editar versión.
5. En la página Editar versión, proporcione la siguiente información
   - Descripción de la versión (opcional): descripción de esta versión.
6. En el panel de configuración del almacenamiento de ejecución predeterminado, proporcione los siguientes valores predeterminados para las ejecuciones que utilizan esta versión de flujo de trabajo. Puede anular los valores predeterminados al iniciar una ejecución:
   - En el tipo de almacenamiento Ejecutar, selecciona Estático o Dinámico.
   - Para el almacenamiento de ejecución estático, seleccione la cantidad predeterminada de capacidad de almacenamiento de ejecución para las ejecuciones que utilizan esta versión de flujo de trabajo. El valor predeterminado de este parámetro es 1200 GiB.
7. Seleccione Save changes (Guardar cambios).

La consola vuelve a la página de detalles del flujo de trabajo y muestra un encabezado de página con la versión actualizada del flujo de trabajo.

Actualizar una versión de flujo de trabajo mediante la CLI

Puede actualizar los parámetros de una versión de flujo de trabajo mediante el siguiente comando CLI. La combinación del identificador del flujo de trabajo y el nombre de la versión identifica de forma exclusiva la versión.

```
aws omics update-workflow-version
--workflow-id 1234567
--version-name "my_version"
--storage-type 'STATIC'
--storage-capacity 2400
--description "version description"
```

No recibirá ninguna respuesta a la `update-workflow-version` solicitud.

Actualiza una versión del flujo de trabajo mediante un SDK

Puede actualizar una versión del flujo de trabajo mediante una de las SDKs. El siguiente ejemplo del SDK de Python muestra cómo actualizar el tipo de almacenamiento y la descripción de una versión de flujo de trabajo.

```
import boto3

omics = boto3.client('omics')

response = omics.update_workflow_version(
   workflowID=1234567,
   versionName='3.0.0',
   storageType='DYNAMIC',
   description='new version description'
)
```

Eliminar una versión del flujo de trabajo

Puede eliminar una versión de flujo de trabajo definida por el usuario mediante la consola, la CLI o una de las SDKs. La eliminación de una versión del flujo de trabajo no afecta a las ejecuciones en curso que estén utilizando la versión del flujo de trabajo.

No puede eliminar elVersión de flujo de trabajo predeterminada. Se eliminan todas las versiones definidas por el usuario y, a continuación, se elimina el flujo de trabajo.

Temas

- Elimine una versión del flujo de trabajo mediante la consola
- Eliminar una versión del flujo de trabajo mediante la CLI
- Elimine una versión del flujo de trabajo mediante un SDK

Elimine una versión del flujo de trabajo mediante la consola

Para eliminar una versión del flujo de trabajo

1. Abra la consola de HealthOmics .
2. Si es necesario, abra el panel de navegación izquierdo (≡). Elija Flujos de trabajo privados.
3. En la página Flujos de trabajo privados, elija el flujo de trabajo.
4. En la página Flujo de trabajo, elija la versión del flujo de trabajo que desee eliminar y elija Eliminar seleccionada en la lista de acciones.
5. En el modal Eliminar versión del flujo de trabajo, introduzca «confirmar» para confirmar la eliminación.
6. Elija Eliminar.

La consola muestra un encabezado de página con la versión del flujo de trabajo eliminada.

Eliminar una versión del flujo de trabajo mediante la CLI

Puede eliminar una versión de flujo de trabajo definida por el usuario mediante el siguiente comando CLI. La combinación del identificador del flujo de trabajo y el nombre de la versión identifica de forma exclusiva la versión.

```
aws omics delete-workflow-version
--workflow-id 9876543
--version-name "my_version"
```

No recibirá ninguna respuesta a la `delete-workflow-version` solicitud.

Elimine una versión del flujo de trabajo mediante un SDK

Puede eliminar un flujo de trabajo mediante uno de los SDKs.

El siguiente ejemplo muestra cómo eliminar un flujo de trabajo mediante el SDK de Python.

```
import boto3

omics = boto3.client('omics')

response = omics.delete_workflow_version(
   workflowID=1234567,
   versionName='3.0.0'
)
```

Uso de HealthOmics corridas

Tras crear un flujo de trabajo, puede iniciar las ejecuciones utilizando el flujo de trabajo.

Al iniciar una ejecución, HealthOmics asigna almacenamiento de ejecución temporal para que el motor de flujo de trabajo lo utilice durante la ejecución. Para garantizar el aislamiento y la seguridad de los datos, HealthOmics aprovisiona el almacenamiento al principio de cada ejecución y lo desaprovisiona al final de la ejecución.

HealthOmics proporciona varias cuotas relacionadas con las ejecuciones y las tareas del flujo de trabajo. Los valores predeterminados son intencionadamente conservadores para ayudarle a evitar sobrecostes inesperados. Puede solicitar un aumento de dichas cuotas. Para obtener más información, consulte HealthOmics cuotas de servicio.

Al iniciar una ejecución, HealthOmics asigna un identificador de ejecución y un uuid de ejecución a la ejecución. Las carreras de una cuenta tienen una duración única. IDs Sin embargo, HealthOmics reutiliza la ejecución eliminada IDs, por lo que una ejecución y una ejecución eliminada pueden tener el mismo ID de ejecución. Además, es poco frecuente, pero es posible que un flujo de trabajo compartido tenga el mismo ID de ejecución que una ejecución de tu cuenta.

run uuidEs un identificador único global (GUID) que puedes usar para identificar corridas entre cuentas o para distinguir entre dos corridas de tu cuenta que tienen el mismo ID de corrida.

> **Note**
>
> A efectos de la procedencia de los datos, le recomendamos que lo utilice para identificar de forma exclusiva run uuid las corridas. También run uuid es el mejor identificador para vincular a su sistema interno de gestión de la información de laboratorio (LIMs) o al sistema de seguimiento de muestras.

Puede utilizar Amazon Q CLI para optimizar las ejecuciones y analizar el rendimiento de las mismas. Para obtener más información, consulte Ejemplos de instrucciones para la CLI de Amazon Q y el tutorial sobre HealthOmics IA generativa de Agentic en. GitHub

Temas

- Ejecute tipos de almacenamiento en HealthOmics flujos de trabajo
- Modo de retención de ejecución para HealthOmics carreras
- HealthOmics ejecutar entradas
- Ejecute el ciclo de vida en un HealthOmics flujo de trabajo
- HealthOmics ejecutar salidas
- Motivos de error de ejecución
- Ciclo de vida de una tarea en HealthOmics ejecución
- Ejecute la optimización para un HealthOmics flujo de trabajo privado
- Ejecute operaciones en HealthOmics

Ejecute tipos de almacenamiento en HealthOmics flujos de trabajo

Al iniciar una ejecución, HealthOmics asigna almacenamiento de ejecución temporal para que el motor de flujo de trabajo lo utilice durante la ejecución. HealthOmicsproporciona el almacenamiento de ejecución temporal como un sistema de archivos.

Para un flujo de trabajo o una ejecución de flujo de trabajo determinados, puede elegir un almacenamiento de ejecución dinámico o estático. De forma predeterminada, HealthOmics proporciona almacenamiento de ejecución DINÁMICO.

> **Note**
>
> El uso del almacenamiento de Run incurre en cargos en tu cuenta. Para obtener información sobre los precios del almacenamiento de ejecución estático y dinámico, consulta HealthOmicslos precios.

En las siguientes secciones se proporciona información que se debe tener en cuenta a la hora de decidir qué tipo de almacenamiento de ejecución utilizar.

Almacenamiento de ejecución dinámico

Recomendamos utilizar el almacenamiento de ejecución dinámico para la mayoría de las ejecuciones, incluidas las que requieren tiempos de inicio más rápidos, las ejecuciones en las que no se conocen de antemano las necesidades de almacenamiento y para los ciclos de pruebas de desarrollo iterativos.

No es necesario estimar el almacenamiento o el rendimiento necesarios para la ejecución. HealthOmics escala dinámicamente el tamaño del almacenamiento hacia arriba o hacia abajo, en función del uso del sistema de archivos durante la ejecución. HealthOmics también escala el rendimiento de forma dinámica en función de las necesidades del flujo de trabajo. Una ejecución nunca falla debido a un error de almacenamiento insuficiente en el sistema de archivos.

El almacenamiento de ejecución dinámico proporciona un provisioning/deprovisioning tiempo de ejecución más rápido que el almacenamiento de ejecución estático. Una configuración más rápida es una ventaja para la mayoría de los flujos de trabajo y también es una ventaja durante development/test los ciclos.

Una vez completada la ejecución (ruta de éxito o ruta de error), la operación de la API GetRun devuelve el almacenamiento máximo utilizado por la ejecución en el campo StorageCapacity. También puedes encontrar esta información en los registros del manifiesto de ejecución ubicados en el grupo de registros. omics En el caso de una ejecución de almacenamiento dinámico que se complete en 2 horas, es posible que el valor máximo de almacenamiento no esté disponible.

En el caso del almacenamiento de ejecución dinámica, la ejecución aprovisiona un sistema de archivos que utiliza el protocolo NFS. NFS considera que las operaciones CREATE, DELETE y RENAME de archivos no son idempotentes, lo que, en ocasiones, puede generar condiciones de carrera para estas operaciones que el código debe gestionar correctamente. Por ejemplo, el código no debería fallar si intenta eliminar un archivo que no existe. Antes de adoptar el almacenamiento de ejecución dinámica, recomendamos ajustar el código del flujo de trabajo para que sea resistente a las operaciones de archivos no idempotentes. Consulte Ejemplos de código para el manejo seguro de operaciones no idempotentes.

Ejemplos de código para el manejo seguro de operaciones no idempotentes

El siguiente ejemplo de Python muestra cómo eliminar un archivo sin errores si el archivo no existe.

```
import os
import errno
```

```
def remove_file(file_path):
    try:
        os.remove(file_path)
    except OSError as e:
        # If the error is "No such file or directory", ignore it (or log it)
        if e.errno != errno.ENOENT:
            # Otherwise, raise the error
            raise

# Example usage
remove_file("myfile")
```

Los siguientes ejemplos utilizan el shell Bash. Para eliminar un archivo de forma segura, incluso si no existe, usa:

```
rm -f my_file
```

Para mover (cambiar el nombre) de un archivo de forma segura, ejecute el comando move solo si el archivo `old_name` existe en el directorio actual.

```
[ -f old_name ] && mv old_name new_name
```

Para crear un directorio, utilice el siguiente comando:

```
mkdir -p mydir/subdir/
```

Almacenamiento de ejecución estático

Para el almacenamiento de ejecuciones estáticas, la ejecución aprovisiona un sistema de archivos que usa el protocolo Lustre. De forma predeterminada, este protocolo es resistente a las operaciones de archivos no idempotentes. No es necesario ajustar el código de flujo de trabajo para gestionar operaciones de archivos no idempotentes.

HealthOmics asigna una cantidad fija de almacenamiento de ejecución. Este valor se especifica al iniciar la ejecución. El almacenamiento de ejecución predeterminado es de 1200 GiB si no especificas un valor. Cuando especificas un valor para el tamaño de almacenamiento en la solicitud de StartRun API, el sistema redondea el valor al múltiplo más cercano de 1200 GiB. Si ese tamaño de almacenamiento no está disponible, se redondea al múltiplo más cercano de 2400 GiB.

Para el almacenamiento en ejecución estática, HealthOmics aprovisiona los siguientes valores de rendimiento:

- Rendimiento de referencia de 200 MB/s por TiB de capacidad de almacenamiento aprovisionada.
- Rendimiento en ráfaga de hasta 1300 MB/s por TiB de capacidad de almacenamiento aprovisionada.

Si el tamaño de almacenamiento especificado es demasiado bajo, se produce un error en la ejecución y se produce un error de almacenamiento insuficiente para el sistema de archivos. El almacenamiento estático en ejecución es ideal para flujos de trabajo predecibles con requisitos de almacenamiento conocidos.

El almacenamiento de ejecución estática es adecuado para cargas de trabajo grandes y en ráfagas con una alta concurrencia de tareas (por ejemplo, un gran volumen de RNASeq muestras procesadas en paralelo). Proporciona un mayor rendimiento del sistema de archivos por GiB y un menor costo por GiB que el almacenamiento de ejecución dinámica.

Calcular el almacenamiento de ejecución estático necesario

Un flujo de trabajo requiere capacidad adicional cuando utiliza el almacenamiento de ejecución estático (en comparación con el almacenamiento de ejecución dinámica), ya que la instalación del sistema de archivos base utiliza el 7% de la capacidad del sistema de archivos estático.

Si ejecuta un flujo de trabajo de almacenamiento de ejecución dinámico para medir el almacenamiento máximo utilizado por la ejecución, utilice el siguiente cálculo para determinar la cantidad mínima de almacenamiento estático necesaria:

```
static storage required =
       maximum storage in GiB used by the dynamic run storage
         + (total static file system size in GiB * 0.07)
```

Por ejemplo:

```
   Maximum storage measured from a dynamic run storage workflow run:  500GiB
     File system size: 1200GiB
     7% of the file system size:  84GiB
     500 + 84 = 584GiB of static run storage required for this run.
```

Por lo tanto, 1200 GiB (la capacidad mínima de almacenamiento de ejecución estática) son suficientes para esta ejecución.

Modo de retención de ejecución para HealthOmics carreras

Una vez completada una ejecución, HealthOmics archiva los metadatos de la ejecución en CloudWatch. De forma predeterminada, CloudWatch conserva los datos de la ejecución de forma indefinida, a menos que cambies la política CloudWatch de retención. Los resultados de ejecución también se almacenan en Amazon S3 hasta que los elimine.

Uno de los ajustables HealthOmics cuotas de servicio es el maximum number of runs (active and inactive) de una región. HealthOmics conserva los metadatos de ejecución de hasta este número de ejecuciones para que los utilicen las operaciones de la consola y la API (ListRuns y GetRun). Al iniciar una ejecución, puede configurar el parámetro del modo de retención de la ejecución para indicar el comportamiento de retención de la ejecución. El parámetro admite los valores REMOVE y RETAIN.

En el caso de una ejecución nueva con el modo de retención establecido en REMOVE, si HealthOmics intenta añadir la ejecución después de haber guardado el número máximo de ejecuciones, se eliminan automáticamente los metadatos de la ejecución más antigua que ha establecido el modo REMOVE. Esta eliminación no afecta a los datos almacenados en Amazon S3 CloudWatch o Amazon S3.

RETAIN es el valor predeterminado para el modo de retención de ejecuciones. En el caso de las ejecuciones en este modo, el sistema no elimina los metadatos de las ejecuciones. Si HealthOmics alcanza el número máximo de corridas, todas configuradas en RETAIN, no podrá crear corridas adicionales hasta que elimine algunas corridas.

Si planea ejecutar un lote de más corridas que el número máximo de ejecuciones al mismo tiempo, asegúrese de configurar el modo de retención de ejecuciones en ELIMINAR. De lo contrario, el lote HealthOmics fallará al intentar iniciar la siguiente ejecución después del máximo.

Consideraciones adicionales sobre el uso del modo de retención REMOVE:

- Cuando empiece a utilizar REMOVE como modo de retención por primera vez, considere la posibilidad de eliminar una o más ejecuciones que utilicen el modo RETAIN para liberar espacios. A medida que inicie más ejecuciones de REMOVE, la eliminación automática se hará cargo, por lo que habrá suficientes ranuras disponibles para nuevas ejecuciones.
- Si desea volver a ejecutar una ejecución archivada (o un conjunto de ejecuciones), utilice la HealthOmics herramienta CLI para volver a ejecutar. Para obtener más información y ejemplos de

cómo utilizar esta herramienta, consulte la reejecución de Omics en el repositorio de herramientas. HealthOmics GitHub

- Se recomienda configurar un nombre único para cada ejecución. Después HealthOmics de eliminar una ejecución, no podrás usar la consola o la API para buscar el nombre o el ID de la ejecución. Sin embargo, puedes utilizarlos CloudWatch para buscar el nombre de la ejecución, así que usa nombres únicos para obtener los mejores resultados de búsqueda.
- Puede usar el CloudWatch start-query comando para obtener información sobre una ejecución archivada. Si el nombre de la ejecución no es único, la consulta puede devolver varios manifiestos. Los parámetros de hora de inicio y hora de finalización definen el intervalo de tiempo de la búsqueda.

```
aws logs start-query  \
    --log-group-name "/aws/omics/WorkflowLog" \
    --query-string 'filter @logStream like "manifest" and @message like "myRunName"'
 \
    --end-time <END-EPOCH-TIME> --start-time <START-EPOCH-TIME>
```

El start-query comando devuelve un identificador de consulta. Al pasar el identificador de consulta al get-query-results comando, se obtienen los resultados de la consulta.

```
aws logs get-query-results --query-id QueryId
```

HealthOmics ejecutar entradas

Si la definición del flujo de trabajo especifica archivos de entrada para el flujo de trabajo o las tareas del flujo de trabajo, HealthOmics organiza los archivos en un volumen provisional dedicado a la ejecución del flujo de trabajo. Estos archivos de entrada son de solo lectura, lo que impide que las tareas modifiquen las posibles entradas para convertirlas en otras tareas del flujo de trabajo. En el caso de las importaciones de directorios, los directorios también son de solo lectura.

Muchas aplicaciones de genómica asumen que los archivos de índice están ubicados en el mismo lugar que los archivos de secuencia (por ejemplo, el `bai` archivo complementario de un archivo). `bam` Para incluir los archivos de índice, especifíquelos como entradas de tareas en la definición del flujo de trabajo.

Temas

- Administrar el tamaño de los parámetros de ejecución

- Formatos de parámetros de entrada de Amazon S3
- Estados del archivo de entrada de Amazon S3

Administrar el tamaño de los parámetros de ejecución

Al iniciar una ejecución, se especifican las entradas de la ejecución en el objeto o archivo JSON de los parámetros de la ejecución. Puede especificar hasta 50 KB de parámetros de ejecución para el flujo de trabajo. Puede utilizar las siguientes técnicas para mantenerse dentro de esta restricción de tamaño:

- Utilice importaciones de directorios

  Para especificar una gran cantidad de archivos de entrada, especifique un parámetro como la ubicación de Amazon S3 que contiene todos los archivos, en lugar de especificar un parámetro para cada ubicación de archivo. Para obtener más información, consulte el tema siguiente (Formatos de parámetros de entrada de Amazon S3).
- Utilice una hoja de muestra

  Una hoja de muestra es un archivo CSV o TSV con una columna para la dirección fastq.gz (o dos para la lectura por pares) y columnas adicionales para los metadatos, como los nombres de los ejemplos. La hoja de muestra se especifica como un parámetro de entrada de ejecución en lugar de un parámetro para cada archivo de entrada.

  El flujo de trabajo define la forma en que la hoja de muestra se asigna a las estructuras de datos del flujo de trabajo. Si bien puede escribir código para hojas de muestra en WDL y CWL, son más comunes en ellas. NextFlow Para ver un ejemplo, consulta la hoja de muestra en el sitio de GitHub nf-core.

Formatos de parámetros de entrada de Amazon S3

Para un parámetro de entrada que acepta una ubicación de Amazon S3, el parámetro puede especificar la ubicación de un archivo o de todo un directorio de archivos. El uso de un directorio tiene las siguientes ventajas:

- Conveniencia: se especifica el nombre del directorio como parámetro. No incluyes el nombre de cada archivo.

- Compacidad: el tamaño máximo del archivo del parámetro de entrada es de 50 KB. Si proporciona una lista larga de nombres de archivos de entrada, puede superar este máximo.

Amazon S3 es un sistema de almacenamiento de objetos plano, por lo que no admite directorios. Para agrupar los archivos en un «directorio», se asigna a cada archivo el mismo prefijo de clave de objeto. Para obtener más información sobre los prefijos de clave de objeto de Amazon S3, consulte Organizar objetos mediante prefijos.

HealthOmics interpreta el valor del parámetro de entrada de la siguiente manera:

- Si la ubicación de Amazon S3 no termina con una barra diagonal ni utiliza el patrón global, se HealthOmics espera que el valor del parámetro sea la clave de un objeto de Amazon S3.

  Por ejemplo, usted especifica introducir `s3://myfiles/runs/inputs/a/file1.fastq` file1.fastq
- Si la ubicación de Amazon S3 termina con una barra diagonal, HealthOmics interpreta el valor del parámetro como un prefijo de Amazon S3. Carga todos los objetos de Amazon S3 con ese prefijo.

  Por ejemplo, puede especificar que se `s3://myfiles/runs/inputs/a/` carguen todos los objetos cuyas claves comiencen con este prefijo.
- En el caso de Nextflow, HealthOmics admite parcialmente el patrón global de Amazon S3 URIs en los parámetros de entrada.

  Por ejemplo, puede especificar `“s3://myfiles/runs/inputs/a/*.gz”` que se ingresen todos los archivos.gz cuyas claves comiencen con este prefijo.

Manejo de Nextflow del patrón Glob en las entradas de Amazon S3

Patrón global	HealthOmics Comportamiento del partido	Notas
s3://bucket/directory/*.txt	Coincide con todos los `.txt` objetos de cualquier profundidad con el prefijo s3://bucket/directory/. For example, matches s3://bucket/directory/	

Patrón global	HealthOmics Comportamiento del partido	Notas
	abc.txt or s3://bucket/directory/subDir/123.txt, etc.	
s3://bucket/directory/**/*.txt	Coincide con todos los `.txt` objetos de cualquier profundidad con el prefijo s3://bucket/directory/. For example, matches s3://bucket/directory/abc.txt or s3://bucket/directory/subDir/123.txt, etc.	En S3, `**` es equivalente a`*`.
s3://bucket/directory/{a,b}.txt	s3://bucket/directory/a.txt, s3://bucket/directory/b.txt	
s3://bucket/directory/?. txt	Coincide con los objetos de la raíz del prefijo cuyo nombre de archivo es un solo carácter seguido `.txt` de. Por ejemplo, coincide con s3://.txt bucket/directory/a.txt but not s3://bucket/directory/someDir/a.txt or s3://bucket/directory/someDir/subDir/a	
s3://bucket/directory/[0-9].txt	s3://bucket/directory/0.txt, s3://bucket/directory/1.txt, ... ,s3://bucket/directory/9.txt	
s3://bucket/directory/[0-9].txt	s3://bucket/directory/1.txt, s3://bucket/directory/2.txt, s3://bucket/directory/3.txt	
s3://bucket/directory/[0-9].txt	s3://bucket/directory/b.txt, s3://bucket/directory/c.txt, ... ,s3://bucket/directory/Y.txt	

Manejo específico de la barra doble en las entradas de Amazon S3

HealthOmics conserva el comportamiento del motor nativo de cada motor de flujo de trabajo al gestionar barras dobles en Amazon S3 URIs, de modo que no necesite realizar ningún cambio en sus flujos de trabajo cuando los migre a. HealthOmics En las siguientes secciones se describe cómo gestiona cada motor los distintos escenarios.

WDL

Si el parámetro de entrada incluye una barra doble en el centro o al final del URI, el motor WDL conserva la barra doble.

Parámetro de entrada	Ubicación esperada	
s3://myfiles/runs/inputs//file1.fastq	s3://1.fastq myfiles/runs/inputs//file	
s3:////myfiles/runs/inputs	s3://myfiles/runs/inputs//	

Siguiente flujo

Si el parámetro de entrada incluye una barra doble en el centro de la URI, el motor Nextflow conserva la barra doble. En el caso de una barra doble al final de la URI, el motor de Nextflow la resuelve en una sola barra.

Parámetro de entrada	Ubicación esperada	
s3://myfiles/runs/inputs//file1.fastq	s3://1.fastq myfiles/runs/inputs//file	
s3://myfiles//runs/inputs//*.gz	s3://myfiles//runs/inputs//*.gz	
s3://myfiles//runs/inputs//	s3://myfiles//runs/inputs/	

BACALAO

Si el parámetro de entrada incluye una barra doble en el centro o al final del URI, el motor CWL conserva la barra doble.

Parámetro de entrada	Ubicación esperada	
s3://myfiles// runs/inputs//file 1.fastq	s3://myfiles// 1.fastq runs/inpu ts//file	
s3://myfiles//runs/inputs//	s3://myfiles//runs/inputs//	

Estados del archivo de entrada de Amazon S3

HealthOmics puede recuperar los objetos de Amazon S3 que S3 entrega en tiempo real. En el caso de los objetos que se encuentran en los siguientes estados de almacenamiento archivado, restore los objetos a los que deben estar disponibles: HealthOmics

- Clases de almacenamiento flexibles de recuperación o archivo profundo en Amazon S3 Glacier.
- Niveles de acceso archivado o acceso a archivos profundos en niveles inteligentes.

Para obtener información sobre la restauración de objetos, consulte Restauración de un objeto archivado en la Guía del usuario de Amazon S3.

Ejecute el ciclo de vida en un HealthOmics flujo de trabajo

Puedes hacer un seguimiento del progreso de una carrera supervisando el estado de la ejecución. HealthOmics actualiza el estado de la ejecución a medida que la ejecución avanza en su ciclo de vida.

Puede recuperar el estado de la ejecución mediante cualquiera de los siguientes métodos:

- La HealthOmics consola muestra el estado de cada ejecución en la Runs página.
- La operación GetRun de la API devuelve el estado de ejecución actual.
- Puede supervisar el estado de la ejecución mediante EventBridge eventos. Para obtener más información, consulte Uso EventBridge con AWS HealthOmics.

Temas

- Valores de estado de ejecución
- La tarea se reintenta
- Implicaciones del estado de ejecución en los precios

Valores de estado de ejecución

Al iniciar una ejecución, HealthOmics establece el estado de la ejecución enPending. A medida que la ejecución avanza en su ciclo de vida, HealthOmics actualiza el valor de estado para reflejar su progreso actual.

> **Note**
>
> No se incurre en cargos durante ningún estado de ejecución que no sea En ejecución. Para obtener más información, consulte la siguiente sección.

HealthOmics admite los siguientes valores de estado de ejecución:

Pending (Pendiente)

La ejecución está en la cola esperando para empezar. Por lo general, las carreras permanecen pendientes durante un breve periodo de tiempo antes de que comiencen.

- Las ejecuciones pueden permanecer pendientes durante más tiempo si envías muchos trabajos al mismo tiempo.
- Las ejecuciones permanecen en pendiente cuando tu cuenta alcanza el número máximo de ejecuciones simultáneas.
- Una ejecución permanece en Pendiente si forma parte de un grupo de ejecuciones que ha alcanzado alguno de sus valores máximos de recursos.
- Puede ajustar las prioridades de ejecución para que determinadas ejecuciones en cola comiencen antes que otras. Para obtener más información sobre la prioridad de ejecución, consultePrioridad de ejecución.

Inicio

HealthOmics crea la ejecución y aprovisiona los recursos necesarios para la ejecución (como el almacenamiento temporal de la ejecución y el nodo motor).

- HealthOmics aprovisiona el almacenamiento temporal de la ejecución al inicio de la ejecución y lo desaprovisiona cuando la ejecución se detiene.

Running

Una ejecución permanece en estado En ejecución durante el proceso de importación, el procesamiento de cada tarea y el proceso de exportación.

- HealthOmics importa los archivos de entrada al sistema de archivos de almacenamiento temporal en ejecución. Los archivos de entrada son de solo lectura, para evitar que las tareas modifiquen las entradas y las conviertan en otras tareas de un flujo de trabajo.
- Durante la exportación de archivos, HealthOmics exporta los archivos de salida del sistema de archivos de almacenamiento en ejecución a la ubicación S3.
- HealthOmics entrega los registros de ejecución y de tareas CloudWatch en tiempo real mientras el estado de ejecución es En ejecución. Para obtener más información, consulte Inicia sesión CloudWatch.

Detención

Una vez finalizado el proceso de exportación, la ejecución pasa al estado de parada.

- HealthOmics desaprovisiona todos los recursos (incluidos el sistema de archivos de almacenamiento en ejecución y el nodo motor).

Completado

La ejecución pasa a Completada cuando se HealthOmics completa el desaprovisionamiento de recursos.

- HealthOmics ha completado todas las tareas ejecutadas y ha exportado los datos de salida sin errores.
- Los resultados de ejecución están disponibles en la ubicación de salida URI de Amazon S3 especificada. Para WDL y CWL, HealthOmics genera un archivo de resumen de los resultados de la ejecución, que proporciona información sobre la. HealthOmics ejecutar salidas
- Los registros del manifiesto de ejecución final y los registros del motor (si corresponde) están disponibles en. CloudWatch
- En el caso de las ejecuciones que admiten reintentos de tareas, una ejecución con el estado Completada puede incluir una o más tareas que hayan fallado. Siempre que se haya reintentado correctamente una tarea para cada tarea fallida, la ejecución HealthOmics pasará a Completada. HealthOmics asigna un nuevo identificador de tarea a cada reintento, de modo que la ejecución incluya la tarea IDs correspondiente a los intentos fallidos y a los intentos completados.

Con error

HealthOmics ha detectado uno o más errores y no ha podido completar todas las tareas ejecutadas.

- Una ejecución fallida pasa al estado de parada y, al mismo tiempo, HealthOmics desaprovisiona los recursos.

Cancelado

Un usuario inició una solicitud para cancelar la ejecución.

- HealthOmics detiene cualquier tarea en ejecución y desaprovisiona todos los recursos.
- HealthOmics no exporta ningún dato de salida de una ejecución cuando un usuario cancela una ejecución. No tienes acceso a ningún archivo intermedio en el caso de una ejecución cancelada.
- Tu cuenta incurre en cargos por las tareas y los recursos que la ejecución consumió durante el estado En ejecución antes de la cancelación.
- No se cobrará ningún cargo si cancelas una ejecución en estado Pendiente o Inicial.

La tarea se reintenta

HealthOmics admite los reintentos de tareas para las tareas que fallan debido a errores de servicio (5XX códigos de estado HTTP).

Si todas las tareas de la ejecución finalmente se completan, incluso si es necesario volver a intentarlo, la ejecución HealthOmics pasa a Completada. HealthOmics asigna un nuevo identificador de tarea a cada reintento, de modo que la ejecución incluya la tarea IDs correspondiente a los intentos fallidos y a los intentos completados.

El comportamiento de reintento predeterminado depende del lenguaje de definición que utilice el flujo de trabajo. El valor predeterminado de Nextflow es que no se reintenten. En el caso de WDL y CWL, HealthOmics intenta reintentar hasta dos veces una tarea fallida, pero puede optar por no volver a intentarlo para tareas específicas o para todas las tareas de un flujo de trabajo. El reintento de tareas resulta útil para corregir los errores de servicio intermitentes. Sin embargo, podría considerar la posibilidad de excluirse de una tarea que sea idempotente.

Para obtener información específica sobre cada lenguaje de definición de flujo de trabajo, consulte los siguientes temas:

- WDL: configura el comportamiento de reintento de tareas en la definición del flujo de trabajo. Consulte Configurar el comportamiento de reintento de tareas de la WDL.
- Nextflow: configure el comportamiento de reintento de tareas en el archivo de configuración de Nextflow o en la definición del flujo de trabajo. Consulte Configurar el comportamiento de reintento de tareas de Nextflow.

- CWL: configure el comportamiento de reintento de tareas en la definición del flujo de trabajo. Consulte Configurar el comportamiento de reintento de tareas de CWL.

Implicaciones del estado de ejecución en los precios

Tu cuenta puede incurrir en cargos mientras el estado de ejecución sea En ejecución. No se te cobrará nada durante ningún otro estado de ejecución. Por ejemplo, los recursos no se cobran cuando la ejecución se inicia o se detiene.

Una ejecución con el estado En ejecución tiene las siguientes implicaciones de facturación:

- Su cuenta incurre en cargos por el uso del sistema de archivos de almacenamiento en ejecución mientras el estado de ejecución sea En ejecución. Para obtener información sobre los tipos de almacenamiento de ejecución, consulte. Ejecute tipos de almacenamiento en HealthOmics flujos de trabajo
- Su cuenta incurre en cargos por la ejecución de tareas, en función de los recursos informáticos y de memoria que especificó para cada tarea en la definición del flujo de trabajo y en función de la duración de la tarea. Para obtener más información, consulte Requisitos de cómputo y memoria para HealthOmics las tareas.
- Cada tarea tiene un límite de facturación mínimo de un minuto. Si ejecutas una tarea durante menos de un minuto, se te cobrará un cargo por el uso mínimo de un minuto. Si es posible, agrupa las tareas pequeñas para optimizar los costes. La agrupación de tareas también reduce el tiempo de ejecución al evitar la creación de varias tareas secuenciales.

Para obtener información adicional sobre los HealthOmics precios, consulta los precios. HealthOmics

HealthOmics ejecutar salidas

Cuando se completa una ejecución de WDL o CWL, los resultados incluyen un archivo de resumen de resultados (en formato JSON) en el que se enumeran todos los resultados producidos por la ejecución. Puede utilizar el archivo de resumen de salida para los siguientes fines:

- Determine mediante programación los archivos de salida que generó la ejecución.
- Valide que la ejecución produjo todos los resultados esperados.

Temas

- Ejecute el resumen de resultados para WDL
- Ejecute el resumen de resultados para CWL

Ejecute el resumen de resultados para WDL

Cuando se completa una ejecución de WDL, HealthOmics crea un archivo de resumen de salida denominado. output.json

Para cada salida del flujo de trabajo, hay un key/value par correspondiente en el archivo. La clave contiene el nombre del flujo de trabajo y el nombre de la salida en el siguiente formato:`WorkflowName.output_name`. Para la salida de un archivo, el valor es un URI de S3 que apunta a la ubicación de salida en S3 donde se almacena el archivo. En el caso de una salida Array [File], el valor es una matriz de S3 URIs.

En el siguiente ejemplo, se muestra el output.json archivo de un flujo de trabajo denominadoBWAMappingWorkflow.

```
{
  "BWAMappingWorkflow.bam_indexes": [
    "s3://omics-outputs/8886192/out/bam_indexes/0/
pbmc8k_S1_L007_R1_001.sorted.bam.bai",
    "s3://omics-outputs/8886192/out/bam_indexes/1/pbmc8k_S1_L008_R1_001.sorted.bam.bai"
  ],
  "BWAMappingWorkflow.mapping_stats": "s3://omics-outputs/8886192/out/mapping_stats/
genome_mapping_final_stats.txt",
  "BWAMappingWorkflow.merged_bam": "s3://omics-outputs/8886192/out/merged_bam/
genome_mapping.merged.bam",
  "BWAMappingWorkflow.merged_bam_index": "s3://omics-outputs/8886192/out/
merged_bam_index/genome_mapping.merged.bam.bai",
  "BWAMappingWorkflow.reference_index_tar": "s3://omics-outputs/8886192/out/
reference_index_tar/reference_index.tar",
  "BWAMappingWorkflow.sorted_bams": [
    "s3://omics-outputs/8886192/out/sorted_bams/0/pbmc8k_S1_L007_R1_001.sorted.bam",
    "s3://omics-outputs/8886192/out/sorted_bams/1/pbmc8k_S1_L008_R1_001.sorted.bam"
  ],
  "BWAMappingWorkflow.unmapped_bams": [
    "s3://omics-outputs/8886192/out/unmapped_bams/0/
pbmc8k_S1_L007_R1_001.unmapped.bam",
    "s3://omics-outputs/8886192/out/unmapped_bams/1/pbmc8k_S1_L008_R1_001.unmapped.bam"
  ]
}
```

Si el flujo de trabajo produce salidas con tipos que no son de archivo (como String, Int, Float o Bool), el valor del campo es una primitiva de JSON. Por ejemplo:

```
{
  "MyWorkflow.my_int_ouput": 1,
  "MyWorkflow.my_bool_output": false,
    ...
}
```

Ejecute el resumen de resultados para CWL

Cuando se completa una ejecución de CWL, HealthOmics crea un archivo de resumen de salida con el nombre de outputs.json la siguiente ubicación:

```
{my-S3outputpath}/{runId}/{run-uuid}/logs/outputs.json
```

El archivo de resumen de resultados incluye una lista de resultados. Cada salida es un key/value par, donde la clave es el nombre de la salida. El valor es un objeto que incluye las siguientes propiedades:

- ubicación: la ruta completa al archivo de salida
- basename: la parte del nombre de archivo de la ruta
- class: el tipo de salida, que normalmente es Archivo
- tamaño: el tamaño del archivo en bytes

En el siguiente ejemplo, el archivo output.json tiene una lista de dos archivos de salida.

```
{
  "example_output": {
    "location": "{my-S3outputpath}/{runId}/{run-uuid}/out/output.txt",
    "basename": "output.txt",
    "class": "File",
    "size": 13
  },
  "another_output": {
    "location": "{my-S3outputpath}/{runId}/{run-uuid}/out/metrics.json",
    "basename": "metrics.json",
    "class": "File",
```

```
    "size": 256
  }
}
```

Motivos de error de ejecución

Si se produce un error en una ejecución, utilice la operación de la GetRunAPI para recuperar el motivo del error.

Revisa el motivo del error para ayudarte a solucionar el error de la ejecución. En la tabla siguiente se enumeran los motivos de cada error junto con una descripción del error.

Motivo del error	Descripción del error
ASSUME_ROLE_FAILED	HealthOmics no tiene permiso para asumir el rol. Especifique el HealthOmics principal de la relación de confianza del rol.
NO SE PUEDE INICIAR _CONTAINER_ERROR	No se puede iniciar la tarea de flujo de trabajo:*name*, id: container using image:. *ID image name* Asegúrese de que la imagen es válida e inténtelo de nuevo.
NO SE PUEDE INICIAR CONTAINER_SIZE_ERROR	No se puede iniciar la tarea de flujo de trabajo:, id: container using image:. *name ID image name* Asegúrese de que el tamaño de la imagen sea inferior a 45 GiB (95 GiB para una instancia de GPU) e inténtelo de nuevo.
ECR_PERMISSION_ERROR	HealthOmics no tiene permiso para acceder al URI de la imagen. Confirme que el repositorio privado de Amazon ECR existe y que ha concedido acceso a la entidad principal del HealthOmics servicio.
EXPORT_FAILED	Se produjo un error en la exportación. Compruebe que el depósito de salida existe y que la función de ejecución tiene permiso de escritura en el depósito.
FILE_SYSTEM_OUT_OF _SPACE	El sistema de archivos no tiene suficiente espacio. Aumente el tamaño del sistema de archivos y vuelva a ejecutarlo.

Motivo del error	Descripción del error
ERROR DE VERIFICACIÓN DE IMAGEN	No se pudo verificar la imagen. *image name* Para corregir el problema, intenta extraer la imagen y, a continuación, volver a subirla al repositorio de ECR.
ERROR DE IMPORTACIÓN	Error en la importación. Compruebe que el archivo de entrada existe y que la función de ejecución puede acceder a la entrada.
INACTIVE_OMICS_STORAGE_RESOURCE	El URI de almacenamiento no está en estado ACTIVO. HealthOmics Active el conjunto de lectura e inténtelo de nuevo. Para obtener más información sobre la activación de los conjuntos de lecturas, consulteActivar conjuntos de lectura en HealthOmics.
INPUT_URI_NOT_FOUND	El URI proporcionado no existe:. *uri* Compruebe que la ruta URI existe y confirme que el rol puede acceder al objeto.
INSTANCE_RESERVATION_FAILED	No hay suficiente capacidad de instancia para completar la ejecución del flujo de trabajo. Espera e intenta ejecutar de nuevo el flujo de trabajo.
INVALID_ECR_IMAGE_URI	La estructura del URI de imagen de Amazon ECR no es válida. Proporcione un URI válido e inténtelo de nuevo.
TASK_RESOURCE_VALUE NO VÁLIDO	La GPU, la CPU o la memoria solicitadas son demasiado altas para la capacidad informática disponible o son inferiores al valor mínimo de 1 para la tarea. *ID*
URI_INPUT NO VÁLIDO	La estructura del URI no es válida. *uri* Comprueba la estructura del URI e inténtalo de nuevo.
MODIFIED_INPUT_RESOURCE	El URI proporcionado se modificó después de que se *uri* iniciara la ejecución. Vuelva a intentar la ejecución.
OUT_OF_MEMORY_ERROR	La tarea del flujo de trabajo se quedó sin memoria. *ID* Aumente el valor de memoria en la definición del flujo de trabajo e intente la ejecución de nuevo.

Motivo del error	Descripción del error
RUN_TASK_FAILED	La ejecución falló porque la tarea falló. Para depurar el error de la tarea, utilice la operación de GetRunTaskAPI y la transmisión de Amazon CloudWatch Logs.
RUN_TIMED_OUT	Se agota el tiempo de espera después de unos minutos. *number*
SERVICE_ERROR	Se ha producido un error transitorio en el servicio. Intente ejecutar de nuevo el flujo de trabajo.
TASK_TIMED_OUT	Se agotó el tiempo de espera de la tarea después *id* de unos segundos. *number*
TAMAÑO DE ENTRADA NO ADMITIDO	El tamaño total de la entrada es demasiado alto. Reduzca el tamaño de entrada e inténtelo de nuevo.
WORKFLOW_RUN_FAILED	Falló la ejecución del flujo de trabajo. Revise el flujo de registro del motor CloudWatch Logs *ID* para depurar el error.
WORKFLOW_VER_VALIDATION_FAILED	HealthOmics no es compatible con la versión de Nextflow solicitada: --. *version* La última versión compatible es*version*. Modifique su versión de Nextflow por una versión compatible e inténtelo de nuevo.
UNSUPPORTED_GPU_INSTANCE_TYPE	El tipo de instancia solicitado no se admite en. *Region* Vuelva a intentar la ejecución con un tipo de instancia de GPU compatible en esta región. Los tipos de instancias disponibles son*GPU instance types*.

Guía para ejecuciones que no responden

Al desarrollar nuevos flujos de trabajo, las ejecuciones o tareas específicas pueden «atascarse» o «bloquearse» si hay problemas con el código y las tareas no salen correctamente de los procesos. Esto puede resultar difícil de solucionar y detectar, ya que es normal que las tareas se ejecuten durante períodos prolongados. Para evitar e identificar las ejecuciones que no responden, siga las prácticas recomendadas que se sugieren en las siguientes secciones.

Prácticas recomendadas para evitar ejecuciones que no respondan

- Asegúrese de cerrar todos los archivos abiertos en el código de la tarea. En ocasiones, abrir demasiados archivos puede provocar problemas de creación de subprocesos en los motores de flujo de trabajo.
- Los procesos en segundo plano creados por una tarea de flujo de trabajo deberían cerrarse al finalizar la tarea. Sin embargo, si un proceso en segundo plano no se cierra correctamente, debe cerrar dicho proceso de forma explícita en el código de la tarea.
- Asegúrese de que sus procesos no se repitan sin salir. Esto puede provocar una ejecución que no responda y, para resolverlo, es necesario cambiar el código de definición del flujo de trabajo.
- Asigne la memoria y la CPU adecuadas a sus tareas. Analice los CloudWatch registros o utilícelos Ejecute Analyzer cuando el flujo de trabajo se haya completado correctamente para comprobar que dispone de una asignación informática óptima. Utilice el `headroom` parámetro Run Analyzer para incluir más margen de maniobra y garantizar que los procesos cuenten con recursos suficientes para completarlos. Incluya al menos un 5% de margen en la memoria y la CPU asignadas para tener en cuenta los procesos del sistema operativo en segundo plano.
  - Además, aumenta el tamaño del ancho de banda de la instancia si la instancia requiere un rendimiento superior. EC2 Las instancias de Amazon con menos de 16 v CPUs (tamaño 4xl o menor) pueden experimentar un aumento de rendimiento. Para obtener más información sobre el rendimiento de las EC2 instancias de Amazon, consulta el ancho de banda de instancias EC2 disponible en Amazon.
- Asegúrese de utilizar el tamaño de sistema de archivos correcto para sus ejecuciones. En el caso de las ejecuciones que no responden y utilizan almacenamiento de ejecución estática, considere la posibilidad de aumentar la asignación de almacenamiento de ejecución estática para permitir un mayor rendimiento de E/S y una mayor capacidad de almacenamiento en el sistema de archivos. Analice el manifiesto de ejecución para ver el almacenamiento máximo del sistema de archivos y utilice el analizador de ejecución para determinar si es necesario aumentar la asignación del sistema de archivos.

Prácticas recomendadas para detectar las ejecuciones que no responden

- Al desarrollar nuevos flujos de trabajo, usa un grupo de ejecución con el límite máximo de tiempo de ejecución establecido para catch runaway code. Por ejemplo, si una ejecución tarda 1 hora en completarse, colócala en un grupo de carreras que agote el tiempo de espera después de 2 o 3 horas (o un período de tiempo diferente según tu caso de uso) para atrapar las tareas agotadas. Además, aplique un búfer para tener en cuenta la variación en los tiempos de procesamiento.

- Configure una serie de grupos de ejecuciones con diferentes límites máximos de tiempo de ejecución. Por ejemplo, puedes asignar ejecuciones cortas a un grupo de ejecuciones que las termine después de unas horas y a un grupo de carreras largas que termine las carreras después de unos días, en función de la duración prevista del flujo de trabajo.
- HealthOmics tiene un límite de servicio de duración máxima de ejecución predeterminado de 604 800 segundos, o 7 días, que se puede ajustar mediante una solicitud en la herramienta de cuotas. Solicita un aumento del límite de servicio de esta cuota únicamente si tienes ejecuciones que se aproximan a una semana de duración. Si tiene una combinación de carreras cortas y largas y no utiliza grupos de carreras, considere la posibilidad de colocar las carreras largas en una cuenta independiente con un límite de servicio de duración máxima de ejecución superior.
- Inspeccione los CloudWatch registros para ver si hay tareas que sospeche que podrían no responder. Si una tarea normalmente genera declaraciones de registro regulares y no lo ha hecho durante un período prolongado, es probable que la tarea esté atascada o bloqueada.

¿Qué hacer si se produce una ejecución que no responde

- Cancela la carrera para evitar incurrir en costes adicionales.
- Inspeccione los registros de tareas para comprobar si algún proceso no se pudo cerrar correctamente.
- Inspeccione los registros del motor para identificar cualquier comportamiento anormal del motor.
- Compare los registros de tareas y del motor de la ejecución que no respondió con los de las ejecuciones idénticas que se completaron correctamente. Esto puede ayudar a identificar cualquier diferencia que pueda haber provocado la falta de respuesta.
- Si no puede determinar la causa raíz, presente un caso de apoyo e incluya lo siguiente:
  - El ARN de la ejecución atascada y el ARN de una ejecución idéntica que se completó correctamente.
  - Registros del motor (disponibles una vez que la ejecución se ha cancelado o ha fallado)
  - Registros de tareas para la tarea que no responde. No necesitamos registros de tareas para todas las tareas del flujo de trabajo para solucionar los problemas.

Ciclo de vida de una tarea en HealthOmics ejecución

Una tarea es un proceso único dentro de una ejecución. HealthOmics asigna cada tarea de su flujo de trabajo al tipo de instancia de computación ómica que mejor se adapte a los recursos necesarios

para la tarea. Los recursos necesarios se especifican en la definición del flujo de trabajo. Para obtener más información, consulteRequisitos de cómputo y memoria para HealthOmics las tareas.

HealthOmics proporciona almacenamiento de ejecución temporal para que lo utilice la tarea. HealthOmics copia los archivos de entrada de la tarea en el almacenamiento temporal de ejecución como archivos de solo lectura. HealthOmics proporciona enlaces simbólicos para que la tarea pueda acceder a los archivos de entrada desde el directorio de trabajo. La tarea solo tiene acceso a los archivos declarados en el archivo de definición del flujo de trabajo.

Valores de estado de la tarea

Puede realizar un seguimiento del progreso de una tarea supervisando el estado de la tarea. Al iniciar una ejecución, HealthOmics establece el estado de la tarea en el Pending de cada tarea de la ejecución. Cuando la tarea se inicia y avanza a lo largo de su ciclo de vida, HealthOmics actualiza el valor de estado para reflejar su progreso actual.

Puede recuperar el estado de la tarea mediante cualquiera de los métodos siguientes:

- La HealthOmics consola muestra el estado de cada tarea en ejecución en la Run details página.
- La operación GetRunTask de la API devuelve el estado de la tarea.
- Puede supervisar el estado de las tareas mediante EventBridge eventos. Para obtener más información, consulte Uso EventBridge con AWS HealthOmics.

Puede recuperar el estado actual de una tarea mediante la operación de la GetRunTask API. La HealthOmics consola muestra el estado de cada tarea en ejecución en la Run details página.

HealthOmics admite los siguientes valores de estado de la tarea:

Pendiente

La tarea está en la cola esperando para empezar. Las tareas permanecen pendientes durante un breve periodo de tiempo antes de que comiencen.

- Las tareas permanecen pendientes después de que tu cuenta haya alcanzado el número máximo de tareas simultáneas.
- Las tareas permanecen pendientes si la ejecución forma parte de un grupo de ejecuciones que ha alcanzado alguno de sus valores máximos de recursos.

- Puede ajustar las prioridades de ejecución para que determinadas ejecuciones en cola y sus tareas comiencen antes que otras ejecuciones en cola. Para obtener más información sobre la prioridad de ejecución, consulte Prioridad de ejecución

Inicio

HealthOmics está creando la tarea y aprovisionando los recursos necesarios para la misma, como el nodo de tareas del flujo de trabajo.

Running

El estado de la tarea es En ejecución mientras HealthOmics se está procesando la tarea.

Detención

Tras completar el procesamiento de la tarea y exportar los datos de salida, la tarea pasa a Detenida.

- HealthOmics desaprovisiona el nodo de tareas del flujo de trabajo.

Completado

HealthOmics ha terminado de procesar la tarea y ha transferido los datos de salida al sistema de archivos de almacenamiento en ejecución.

Con error

HealthOmics ha detectado un error al procesar la tarea y no la ha completado.

- La tarea pasa al estado Parada (HealthOmics desaprovisiona los recursos) y, después, al estado Fallido.
- Si el error es un error de servicio (código de estado HTTP 5XX) y el flujo de trabajo admite HealthOmics reintentos para esta tarea, intenta volver a procesarla. HealthOmics asigna un nuevo identificador de tarea al reintento.

Cancelado

HealthOmics detiene la tarea tras una solicitud iniciada por el usuario para cancelar la ejecución.

- La tarea pasa al estado de parada (HealthOmics desaprovisiona los recursos) y, después, al estado cancelada.

Solución de problemas de tareas de flujo

A continuación, se indican las prácticas recomendadas y las consideraciones para solucionar problemas de sus tareas.

- Los registros de tareas `STDERR` se basan en la tarea `STDOUT` y son generados por ella. Si la aplicación utilizada en la tarea no produce ninguno de estos datos, no habrá ningún registro de tareas. Para facilitar la depuración, utilice las aplicaciones en `verbose` modo.
- Para ver los comandos que se están ejecutando en una tarea junto con sus valores interpolados, utilice el `set -x` comando Bash. Esto puede ayudar a determinar si la tarea está utilizando las entradas correctas e identificar los errores que podrían haber impedido que la tarea se ejecutara según lo previsto.
- Utilice el `echo` comando para enviar los valores de las variables a `STDOUT` o`STDERR`. Esto le ayuda a confirmar que se están configurando como se esperaba.
- Usa comandos como `ls -l <name_of_input_file>` estos para confirmar que las entradas están presentes y tienen el tamaño esperado. Si no lo están, esto podría revelar un problema con una tarea anterior que produce resultados vacíos debido a un error.
- Utilice el comando `df -Ph . | awk 'NR==2 {print $4}'` de un script de tareas para determinar el espacio disponible actualmente para la tarea y ayudar a identificar situaciones en las que podría necesitar ejecutar el flujo de trabajo con una asignación de almacenamiento adicional.

Al incluir cualquiera de los comandos anteriores en un script de tareas, se supone que el contenedor de tareas también incluye estos comandos y que se encuentran en el entorno `path` del contenedor.

Ejecute la optimización para un HealthOmics flujo de trabajo privado

Puedes optimizar las ejecuciones teniendo en cuenta el coste total, el tiempo total de ejecución o una combinación de ambos. HealthOmics proporciona datos y herramientas que le ayudarán a tomar decisiones de optimización de las ejecuciones. La optimización de la ejecución no se aplica a los flujos de trabajo de Ready2Run, ya que no tienes ningún control sobre la forma en que el servicio gestiona el aprovisionamiento de recursos para estos flujos de trabajo.

El primer paso consiste en comprender el uso actual de los recursos de la tarea y el coste de las tareas en ejecución y, a continuación, aplicar métodos para optimizar el coste y el rendimiento de la ejecución.

Temas

- Ejecute Analyzer
- Determine los costos de ejecución
- Determine el uso del tiempo de ejecución
- Métodos para optimizar las ejecuciones

- Impacto de la variación del tamaño de los archivos entre ejecuciones
- Métodos para optimizar la simultaneidad de recursos

Ejecute Analyzer

HealthOmics proporciona una herramienta de código abierto denominada Run Analyzer. Esta herramienta extrae información sobre el uso de los recursos a nivel de tarea para una ejecución y sugiere oportunidades de optimización del coste y el rendimiento de la ejecución.

> **Note**
>
> Run Analyzer estima los costos de las tareas y los posibles ahorros de costos en función de los precios AWS de lista en el momento en que se ejecuta la herramienta. Evalúe las recomendaciones de optimización e implemente las que mejor se adapten a sus casos de uso. Pruebe las optimizaciones que adopte para asegurarse de que funcionan para su ejecución.

Run Analyzer realiza las siguientes tareas:

- Evalúa los cuellos de botella de memoria y computación.
- Identifica las tareas que están sobreaprovisionadas de memoria o CPU y recomienda nuevos tamaños de instancia que pueden reducir los costes.
- Calcula las estimaciones de costes de las tareas individuales y calcula los posibles ahorros de costes si aplicas las recomendaciones.
- Le proporciona una vista cronológica de las tareas para que pueda verificar las dependencias de las tareas y la secuencia de procesamiento. El cronograma también le ayuda a identificar las tareas de larga duración.
- Proporciona recomendaciones sobre el tamaño del sistema de archivos para el almacenamiento en ejecución.
- Muestra los tiempos de aprovisionamiento de las tareas para que pueda identificar las áreas en las que las grandes cargas de contenedores pueden estar ralentizando el tiempo de aprovisionamiento.
- La herramienta incluye un parámetro de entrada (margen de maniobra) que puede utilizar para controlar la agresividad de las recomendaciones de optimización.

Las siguientes secciones incluyen sugerencias específicas para usar Run Analyzer para optimizar las ejecuciones.

Determine los costos de ejecución

Puede utilizar los siguientes métodos y directrices para determinar los costes de ejecución:

- Para ver los costes de funcionamiento totales de un período de facturación, sigue estos pasos:

  1. Abra la consola Billing and Cost Management y seleccione Bills.
  2. En Cargos por servicio, expanda Omics.
  3. Amplíe la región y, a continuación, consulte el costo de todas sus ejecuciones desglosado por tipo de instancia ómica, tipo de almacenamiento de ejecución y flujo de trabajo Ready2Run.
- Para generar un informe de costes que incluya información sobre cada ejecución, sigue estos pasos:

  1. Abra la consola de Billing and Cost Management y seleccione Exportaciones de datos.
  2. Elija Crear para crear una nueva exportación de datos.
  3. Introduzca un nombre de exportación para la exportación de datos. Mantenga los demás campos en sus valores predeterminados para crear un informe CUR (costo y uso).
  4. En la granularidad del tiempo, seleccione por hora o por día.
  5. En Configuración de almacenamiento de exportación de datos, lleve a cabo estos pasos de configuración:

     a. Configure un bucket de Amazon S3 para la exportación de datos.
     b. Para el control de versiones de archivos, seleccione si desea sobrescribir el archivo de exportación existente o crear un archivo nuevo cada vez.

     El sistema genera el primer informe en las próximas 24 horas y genera los informes posteriores una vez al día.
  6. Para obtener más información sobre cómo crear la exportación de datos, consulte Creación de exportaciones de datos en la Guía del usuario de exportación de AWS datos.
- Puede etiquetar sus recorridos para supervisar y optimizar los costes por categoría, por ejemplo, por equipo o por proyecto. Si utilizas etiquetas, sigue estos pasos para ver los costes de las carreras por categoría de etiquetas:

  1. Abra la consola de Billing and Cost Management y seleccione Cost Explorer.

2. En Parámetros del informe > Agrupar por, elija Etiqueta como dimensión y seleccione el nombre de etiqueta deseado.

- Para ver el uso de los recursos en las tareas, consulta el manifiesto de ejecución e inicia sesión CloudWatch. Para obtener más información, consulte Supervisión HealthOmics con CloudWatch registros.
- Usa la Ejecute Analyzer herramienta para extraer la información sobre el uso de los recursos de la tarea para una ejecución.

Determine el uso del tiempo de ejecución

Puedes usar los siguientes métodos para ayudarte a investigar el uso del tiempo de ejecución:

- En la página Ejecuciones de la consola, puede ver el tiempo total de ejecución de una ejecución.
- En la página de detalles de la ejecución, puedes ver los siguientes elementos:
  - Consulta el tiempo total de ejecución de una ejecución.
  - Muestra el tiempo de ejecución de cada tarea de la ejecución.
  - Elija uno de los enlaces para ver los registros en Amazon S3 o para ver los registros de ejecución o los registros del manifiesto de ejecución CloudWatch.
- En la lista Ejecutar tareas, seleccione el enlace Ver registros de una tarea para ver los registros de la tarea CloudWatch.
- La respuesta a la operación de la `listRuns` API incluye la hora de inicio y la hora de finalización de la ejecución, para que puedas calcular el tiempo total de ejecución.
- La Ejecute Analyzer herramienta muestra la duración de las tareas en una vista de cronograma. Esta herramienta proporciona una representación visual de la secuencia de procesamiento de la tarea, que puede hacer coincidir con el orden esperado.

Métodos para optimizar las ejecuciones

HealthOmics aprovisiona, administra y optimiza automáticamente los recursos que realizan la preparación de datos (como la importación y exportación de datos). HealthOmics también inicia y ejecuta el motor de flujo de trabajo de su flujo de trabajo. Sin embargo, puede influir en las horas de inicio de la ejecución, las horas de inicio de las tareas y el tiempo de ejecución general de las tareas configurando varias configuraciones de ejecución. El enfoque general de la definición y el diseño del flujo de trabajo también afecta al tiempo de ejecución de las tareas. En la siguiente lista se describen los factores que pueden afectar al rendimiento de la ejecución y de las tareas:

Ejecute el tipo de almacenamiento

El tipo de almacenamiento de ejecución afecta al rendimiento de la ejecución y al tiempo de aprovisionamiento de la ejecución. El almacenamiento de ejecución dinámica se aprovisiona más rápido y nunca se queda sin memoria, ya que se escala de forma dinámica en función de las necesidades de almacenamiento de ejecución. El almacenamiento de ejecución dinámica también es una buena opción para los flujos de trabajo en desarrollo, en los que a menudo es posible iniciar y detener un flujo de trabajo para solucionar problemas.

El almacenamiento de ejecución estática requiere tiempos de aprovisionamiento del sistema de archivos más prolongados, pero puede completar algunas ejecuciones más rápido, normalmente si las ejecuciones tienen una alta concurrencia de tareas o requieren más de 9,6 TiB de capacidad del sistema de archivos. El almacenamiento de ejecuciones estáticas es ideal para flujos de trabajo de larga duración con requisitos elevados. I/O

Para ayudarlo a evaluar el costo en comparación con el rendimiento de cada tipo de almacenamiento de ejecución para una ejecución determinada, puede probar las pruebas A/B para ver qué tipo de almacenamiento de ejecución ofrece un mejor rendimiento. Además, considere la posibilidad de utilizar el almacenamiento de ejecución dinámico para sus ciclos de desarrollo y, a continuación, utilice el almacenamiento de ejecución estático para las ejecuciones de producción a escala.

Para obtener más información sobre los tipos de almacenamiento de ejecución Ejecute tipos de almacenamiento en HealthOmics flujos de trabajo

Aprovisione en exceso el almacenamiento estático de ejecución

Si el cálculo de las tareas del flujo de trabajo se ve limitado por I/O, consider over-provisioning the static run storage. Storage cost increases with its size, but maximum throughput of the file system also increases. If an expensive compute task is experiencing I/O cuellos de botella, aumentar el tamaño del sistema de archivos para reducir el tiempo de ejecución de la tarea puede reducir el coste total.

Reduzca el tamaño de las imágenes del contenedor

Cuando se inicia cada tarea, HealthOmics carga el contenedor que especificó para la tarea. Los contenedores más grandes tardan más en cargarse. Optimice sus contenedores para que sean lo más pequeños posible para mejorar la eficiencia a la hora de lanzar nuevas tareas. Si agrega conjuntos de datos de gran tamaño a sus contenedores, considere la posibilidad de almacenar los conjuntos de datos en S3 y hacer que su flujo de trabajo importe los datos de S3. Para

conocer los tamaños máximos de contenedores admitidos HealthOmics , consulte. HealthOmics cuotas de tamaño fijas en el flujo de trabajo

Tamaño de tarea

Puede combinar pequeñas tareas secuenciales en una sola tarea para ahorrar tiempo de aprovisionamiento de tareas. Además, HealthOmics tiene una duración mínima de un minuto por tarea, por lo que la combinación de tareas puede reducir los costes. Dentro de la tarea combinada, es posible que pueda utilizar canales Unix para evitar el I/O coste de serializar y deserializar los archivos.

Compresión de archivos

Evite comprimir demasiado los archivos intermedios del flujo de trabajo. La mayoría de los formatos de genómica utilizan la compresión «gzip» o «block gzip». Descomprimir el archivo de entrada de la tarea y volver a comprimir el archivo de salida de la tarea puede consumir un gran porcentaje del uso total de la CPU de la tarea. Algunas aplicaciones de genómica permiten establecer el nivel de compresión al serializar los resultados. Al reducir el nivel de compresión, puede reducir el tiempo de CPU, aunque los archivos más grandes aumentan el tiempo dedicado a escribir en el disco. En función de la tarea y de la aplicación, podrá encontrar el nivel de compresión óptimo para los archivos intermedios que reduzcan el tiempo de ejecución. Le recomendamos que comience por centrarse en las tareas con los archivos de salida más grandes. Un nivel de compresión de 2 funciona bien en varios escenarios. Puede empezar con este nivel para su caso de uso y comparar los resultados probando otros niveles de compresión.

Recuento de hilos

Si especificas subprocesos en la definición de tu tarea, establece el número de subprocesos en el mismo valor que el número de subprocesos solicitadosCPUs.

Especifique el cómputo y la memoria

Si no especificas recursos de memoria o cómputo en tu tarea, HealthOmics asigna el tipo de instancia más pequeño (`omics.c.large`) como predeterminado. Declara de forma explícita tus requisitos de memoria y procesamiento si quieres HealthOmics asignar un tipo de instancia más grande.

HealthOmics asigna la cantidad de recursos vCPUs, de memoria y de GPU que solicites. Por ejemplo, si solicitas 15 v CPUs y 33 GiB, HealthOmics asigna una instancia omics.m.4xl (16 v, 64 GB) para tu tareaCPUs, pero tu tarea solo puede usar 15 v y 33 GiB. CPUs Por lo tanto, te recomendamos que solicites v y recursos de memoria que coincidan con una instancia ómica. CPUs

Batch varias muestras en una sola ejecución

Como el aprovisionamiento del sistema de archivos lleva tiempo al inicio de la ejecución, puede ahorrar tiempo de aprovisionamiento agrupando varias muestras en la misma ejecución. Tenga en cuenta los siguientes factores antes de decidirse por este enfoque:

- Una sola muestra defectuosa puede provocar un error en un flujo de trabajo, por lo que agrupar muestras por lotes podría aumentar el número de flujos de trabajo fallidos. Si no está seguro de que su flujo de trabajo vaya a funcionar correctamente la mayoría de las veces, una ejecución por muestra podría ser una mejor opción.
- HealthOmics asigna un sistema de archivos de almacenamiento de una ejecución para todo el flujo de trabajo. Para un lote de muestras, asegúrese de especificar una cantidad de almacenamiento lo suficientemente grande como para procesar todas las muestras.
- Hay una cantidad máxima de almacenamiento de ejecuciones por flujo de trabajo, por lo que puede limitar la cantidad de muestras que puede añadir al lote.
- El tamaño mínimo de almacenamiento de las tiradas es de 1,2 TiB, por lo que el procesamiento por lotes puede reducir los costes si el flujo de trabajo utiliza mucho menos almacenamiento que el mínimo para cada muestra.
- El almacenamiento de ejecución puede gestionar varias conexiones simultáneas, por lo que tener varias tareas utilizando el mismo almacenamiento de ejecución no debería provocar I/O cuellos de botella.
- Cada ejecución tiene su propio conjunto de etiquetas. Si etiqueta los flujos de trabajo con información para presupuestar o realizar un seguimiento, puede ser mejor utilizar ejecuciones independientes.
- Las funciones de IAM se aplican a toda la ejecución. Cada usuario tiene acceso a todos los datos de un lote de muestras. Al tener flujos de trabajo separados, podrá utilizar permisos más detallados.
- HealthOmics establece cuotas a nivel de cuenta para el número máximo de flujos de trabajo simultáneos y el número máximo de tareas simultáneas en un flujo de trabajo. Para obtener información sobre cómo solicitar un aumento de estas cuotas, consulte. HealthOmics cuotas de servicio

Utilice parámetros para las imágenes del contenedor

Parametriza las imágenes del contenedor en lugar de incrustarlas URIs en el flujo de trabajo. Cuando son parámetros de ejecución, HealthOmics valida que la ejecución tenga acceso a sus contenedores antes de que comience la ejecución. De lo contrario, la tarea fallará durante la

ejecución, cuando haya incurrido en cargos por cualquier tarea completada. Además, dado que se trata de entradas parametrizadas, HealthOmics genera una suma de comprobación en el manifiesto de la ejecución, lo que mejora la procedencia de la ejecución.

Usa un linter

Use un linter para encontrar errores comunes en el flujo de trabajo antes de ejecutar un nuevo flujo de trabajo. Para obtener más información, consulte Lentes de flujo de trabajo en HealthOmics.

Se usa EventBridge para marcar problemas

Utilice alertas EventBridge personalizadas para atrapar las anomalías específicas de su lógica empresarial.

Utilice almacenes de secuencias

Considere la posibilidad de utilizar un almacén de secuencias para sus datos de origen a fin de ahorrar costes de almacenamiento. Para obtener más información, consulte la entrada de HealthOmics blog sobre cómo almacenar datos ómicos de forma rentable a cualquier escala.

Impacto de la variación del tamaño de los archivos entre ejecuciones

Los usuarios suelen diseñar y probar las ejecuciones con un conjunto pequeño de datos de prueba y, a continuación, se encuentran con una amplia variedad de datos con una variación significativa en el tamaño de los archivos en las tiradas de producción. Asegúrese de tener en cuenta esta variación al optimizar la ejecución.

En la siguiente lista se describen las recomendaciones de optimización cuando hay una variación significativa en el tamaño de los archivos:

Varíe los tamaños de los archivos en los datos de las pruebas

Intente utilizar datos de prueba durante el desarrollo que tengan una cantidad de varianza representativa.

Utilice Run Analyzer

Utilice la herramienta Run Analyzer en una variedad de muestras para tener en cuenta la variación en el tamaño de los datos.

Puede usar el analizador de corridas para comprender la varianza entre las ejecuciones en sus muestras de datos de producción. Utilice `--batch` el modo de Run Analyzer para generar

estadísticas para un lote de ejecuciones y analizar los recursos informáticos máximos necesarios para gestionar los valores atípicos de los conjuntos de datos.

Por ejemplo, puede proporcionar al analizador de ejecución una celda de datos de flujo completo en modo por lotes para comprender el uso máximo de vCPU y memoria en la celda de flujo completo.

Reduzca la variación de tamaño de los conjuntos de datos de entrada

Si observa una gran variación en los tamaños de las muestras, puede bifurcar las muestras en sentido ascendente HealthOmics y seleccionar diferentes tamaños de sistema de archivos para cada lote a fin de ahorrar costes de almacenamiento.

En WDL, utilice la `size` función para bifurcar la asignación de recursos para tareas individuales en el caso de muestras grandes y pequeñas. Aplica esta estrategia a tus tareas más costosas para lograr el mayor impacto.

En Nextflow, usa recursos condicionales para organizar la asignación de recursos por niveles según el tamaño o el nombre del archivo. Para obtener más información, consulte Recursos de procesos condicionales en el sitio de GitHub Nextflow.

No optimice demasiado pronto

Finalice el código y la lógica de su flujo de trabajo antes de invertir en importantes esfuerzos de ajuste del rendimiento. Cambiar el código puede tener un impacto significativo en los recursos necesarios. Si optimizas una ejecución demasiado pronto en el proceso de desarrollo, es posible que la optimices en exceso o que tengas que volver a optimizarla si la definición del flujo de trabajo cambia más adelante.

Vuelva a ejecutar la herramienta Run Analyzer periódicamente

Si realiza cambios en la definición de su flujo de trabajo a lo largo del tiempo o si la varianza de la muestra cambia, ejecute periódicamente la herramienta Run Analyzer para ayudarle a realizar optimizaciones adicionales.

Métodos para optimizar la simultaneidad de recursos

HealthOmics proporciona las siguientes funciones para ayudarle a controlar y gestionar los costes cuando el procesamiento se ejecuta a gran escala:

- Utilice grupos de ejecución para controlar los costes y el uso de los recursos. Puede establecer valores máximos en el grupo de ejecuciones para el número de ejecuciones simultáneas CPUs

GPUs, v y el tiempo total de ejecución por tarea. Si distintos equipos o grupos utilizan la misma cuenta, puedes crear un grupo de carreras independiente para cada equipo. Puedes controlar el uso de los recursos y los costes por equipo y configurar los valores máximos del grupo de carreras. Para obtener más información, consulte Uso de grupos de HealthOmics ejecución.

- Durante el desarrollo, puedes configurar un grupo de ejecución independiente con valores máximos más bajos para catch runaway tasks.
- Service Quotas también ayuda a proteger tu cuenta de solicitudes de recursos excesivas. Para obtener información sobre Service Quotas, incluida la forma de solicitar aumentos en el valor de las cuotas, consulte HealthOmics cuotas de servicio

Ejecute operaciones en HealthOmics

Puedes iniciar, volver a ejecutar, clonar, cancelar o eliminar una ejecución:

- Start— HealthOmics crea una nueva ejecución con los ajustes de configuración que especifique y, a continuación, inicia la ejecución.
- Rerun— HealthOmics crea una nueva ejecución que es un duplicado de la ejecución especificada. Puede volver a ejecutar una ejecución eliminada con la HealthOmics rerun herramienta.
- Clone— Puede clonar una ejecución existente mediante la consola. La consola abre la página de clonación de la ejecución y rellena previamente los campos de configuración con los valores de la ejecución existente. Puede modificar los valores según sea necesario e iniciar la ejecución clonada.
- Cancel— Puede cancelar una ejecución que aún no se haya completado. Cuando cancelas una ejecución, HealthOmics no guarda ninguno de los resultados de la ejecución.
- Delete— Puede eliminar las corridas completadas manualmente o configurar el modo de retención de corridas HealthOmics para eliminar automáticamente las corridas más antiguas. Para obtener más información sobre el modo de retención, consultethe section called “Ejecute los modos de retención”.

Temas

- Iniciar una carrera en HealthOmics
- Vuelva a ejecutar una ejecución HealthOmics
- Clona una carrera en HealthOmics
- Cancela una carrera HealthOmics

- Eliminar una carrera en HealthOmics

Iniciar una carrera en HealthOmics

Al iniciar una ejecución, se especifican los recursos que se HealthOmics asignarán para su uso durante la ejecución.

Especifique el tipo de almacenamiento de la ejecución y la cantidad de almacenamiento (para el almacenamiento estático). Para garantizar el aislamiento y la seguridad de los datos, HealthOmics aprovisiona el almacenamiento al inicio de cada ejecución y lo desaprovisiona al final de la ejecución. Para obtener información adicional, consulta Ejecute tipos de almacenamiento en HealthOmics flujos de trabajo.

Especifique una ubicación de Amazon S3 para los archivos de salida. Si ejecuta un gran volumen de flujos de trabajo de forma simultánea, utilice una salida de Amazon S3 independiente URIs para cada flujo de trabajo a fin de evitar la limitación de los cubos. Para obtener más información, consulte Organizar objetos mediante prefijos en la Guía del usuario de Amazon S3 y Escalar las conexiones de almacenamiento horizontalmente en el documento técnico Optimización del rendimiento de Amazon S3.

También puede especificar la prioridad de ejecución. El impacto de la prioridad en la ejecución depende de si la ejecución está asociada a un grupo de ejecuciones. Para obtener información adicional, consulta Prioridad de ejecución.

Si un flujo de trabajo tiene una o más versiones, puede especificar una versión al iniciar la ejecución. Si no especifica una versión, HealthOmics inicia la versión predeterminada del flujo de trabajo.

Al usar la HealthOmics API, puedes proporcionar un ID de solicitud único para cada ejecución. El ID de solicitud es un token de idempotencia que se HealthOmics utiliza para identificar las solicitudes duplicadas e inicia la ejecución solo una vez.

> **Note**
>
> Al iniciar una ejecución, debe especificar una función de servicio de IAM. Si lo desea, la consola puede crear el rol de servicio automáticamente. Para obtener más información, consulte Funciones de servicio para AWS HealthOmics.

Temas

- HealthOmics ejecutar parámetros
- Iniciar una ejecución mediante la consola
- Iniciar una ejecución mediante la API
- Obtén información sobre una ejecución

HealthOmics ejecutar parámetros

Al iniciar una ejecución, se especifican las entradas de la ejecución en el archivo JSON de parámetros de ejecución o se pueden introducir los valores de los parámetros en línea. Para obtener información sobre cómo administrar el tamaño del archivo JSON de parámetros de ejecución, consulteAdministrar el tamaño de los parámetros de ejecución.

HealthOmics admite los siguientes tipos de JSON para los valores de los parámetros.

Tipo JSON	Ejemplo de clave y valor	Notas
booleano	«b»: verdadero	El valor no está entre comillas y está todo en minúsculas.
entero	«i»: 7	El valor no está entre comillas.
número	«f»: 42.3	El valor no está entre comillas.
cadena	«s» :"caracteres»	El valor está entre comillas. Utilice el tipo de cadena para los valores de texto y URIs. El objetivo del URI debe ser el tipo de entrada esperado.
array	«a»: [1,2,3]	El valor no está entre comillas. Cada uno de los miembros de la matriz debe tener el tipo definido por el parámetro de entrada.
objeto	«o»: {"izquierda» :"a», «derecha» :1}	En WDL, el objeto se asigna a un par, mapa o estructura de WDL

Iniciar una ejecución mediante la consola

Para iniciar una ejecución

1. Abra la consola de HealthOmics .
2. Si es necesario, abre el panel de navegación izquierdo (≡). Elija Ejecuciones.
3. En la página Ejecuciones, seleccione Iniciar ejecución.
4. En el panel de detalles de la ejecución, proporcione la siguiente información

   - Origen del flujo de trabajo: elija un flujo de trabajo propio o un flujo de trabajo compartido.
   - ID del flujo de trabajo: el ID del flujo de trabajo asociado a esta ejecución.
   - Versión del flujo de trabajo (opcional): seleccione una versión del flujo de trabajo para usarla en esta ejecución. Si no selecciona una versión, la ejecución utilizará la versión predeterminada del flujo de trabajo.
   - Nombre de la ejecución: nombre distintivo de esta ejecución.
   - Prioridad de ejecución (opcional): la prioridad de esta ejecución. Los números más altos especifican una prioridad más alta y las tareas de mayor prioridad se ejecutan primero.
   - Tipo de almacenamiento de ejecución: especifique aquí el tipo de almacenamiento para anular el tipo de almacenamiento de ejecución predeterminado especificado para el flujo de trabajo. El almacenamiento estático asigna una cantidad fija de almacenamiento para la ejecución. El almacenamiento dinámico se amplía y reduce según sea necesario para cada tarea de la ejecución.
   - Capacidad de almacenamiento en ejecución: en el caso del almacenamiento estático en ejecución, especifique la cantidad de almacenamiento necesaria para la ejecución. Esta entrada anula la cantidad de almacenamiento de ejecución predeterminada especificada para el flujo de trabajo.
   - Seleccione el destino de salida de S3: la ubicación de S3 en la que se guardarán las salidas de la ejecución.
   - ID de cuenta del propietario del bucket de salida (opcional): si su cuenta no es propietaria del bucket de salida, introduzca el Cuenta de AWS ID del propietario del bucket. Esta información es necesaria para HealthOmics poder verificar la propiedad del depósito.
   - Modo de retención de metadatos de ejecución: elija si desea conservar los metadatos de todas las ejecuciones o hacer que el sistema elimine los metadatos de las ejecuciones más antiguas cuando su cuenta alcance el número máximo de ejecuciones. Para obtener más información, consulte Modo de retención de ejecución para HealthOmics carreras.

5. En Función de servicio, puedes usar una función de servicio existente o crear una nueva.
6. (Opcional) En el caso de las etiquetas, puedes asignar hasta 50 etiquetas a la ejecución.
7. Elija Siguiente.
8. En la página Agregar valores de parámetros, proporcione los parámetros de ejecución. Puede cargar un archivo JSON que especifique los parámetros o introducir los valores manualmente.
9. Elija Siguiente.
10. En el panel Grupo de ejecuciones, si lo desea, puede especificar un grupo de ejecuciones para esta ejecución. Para obtener más información, consulte Uso de grupos de HealthOmics ejecución.
11. En el panel Caché de ejecución, si lo desea, puede especificar una caché de ejecución para esta ejecución. Para obtener más información, consulte Configuración de una ejecución con caché de ejecución mediante la consola.
12. Elija Review and start run (Revisar e iniciar ejecución).
13. Tras revisar la configuración de ejecución, seleccione Iniciar ejecución.

Iniciar una ejecución mediante la API

Utilice la operación API start-run para crear e iniciar una ejecución.

En el siguiente ejemplo, se especifican el identificador del flujo de trabajo y la función de servicio. En este ejemplo, se establece el modo de retención en`REMOVE`. Para obtener más información sobre el modo de retención, consulteModo de retención de ejecución para HealthOmics carreras.

```
aws omics start-run
     --workflow-id workflow id \
     --role-arn arn:aws:iam::1234567892012:role/service-role/
OmicsWorkflow-20221004T164236 \
     --name workflow name \
     --retention-mode REMOVE
```

Como respuesta, obtendrá el siguiente resultado. `uuid`Es exclusivo de la ejecución y, además, se `outputUri` puede usar para rastrear dónde se escriben los datos de salida.

```
{
    "arn": "arn:aws:omics:us-west-2:....:run/1234567",
    "id": "123456789",
    "uuid":"96c57683-74bf-9d6d-ae7e-f09b097db14a",
```

```
    "outputUri":"s3://bucket/folder/8405154/96c57683-74bf-9d6d-ae7e-f09b097db14a"
    "status": "PENDING"
}
```

Incluye un archivo de parámetros

Si la plantilla de parámetros de un flujo de trabajo declara algún parámetro obligatorio, puede proporcionar un archivo JSON local con las entradas al iniciar la ejecución de un flujo de trabajo. El archivo JSON contiene el nombre exacto de cada parámetro de entrada y un valor para el parámetro.

Haz referencia al archivo JSON de entrada que aparece en el AWS CLI agregándolo `--parameters file://<input_file.json>` a tu `start-run` solicitud. Para obtener más información sobre los parámetros de ejecución, consulteHealthOmics ejecutar entradas.

Proporcione un identificador de solicitud

Puedes proporcionar una única `requestId` para cada ejecución. El ID de solicitud es un token de idempotencia que se HealthOmics utiliza para atrapar solicitudes duplicadas. No iniciará una ejecución si el ID de solicitud es un duplicado de una ejecución anterior.

Si utiliza infraestructura (como funciones Lambda o funciones escalonadas) para organizar los inicios de las ejecuciones, lo mejor es proporcionar un identificador de solicitud único para cada solicitud. StartRun Esto garantiza que si su infraestructura inicia de forma inadvertida una ejecución que ya ha iniciado, HealthOmics no iniciará la ejecución duplicada. Por ejemplo, si la infraestructura intenta recuperarse de un error inicial, puede volver a ejecutar un script que intente iniciar ejecuciones que son solicitudes duplicadas.

Elija una versión del flujo de trabajo

Puede especificar una versión del flujo de trabajo para la ejecución. Si no especifica una versión, HealthOmics inicia la ejecución con la versión de flujo de trabajo predeterminada.

```
aws omics start-run
     --workflow-id workflow id \
      ...
     --workflow-version-name '1.2.1'
```

Anule el tipo de almacenamiento de ejecución

Puede anular el tipo de almacenamiento de ejecución predeterminado que se estableció en el flujo de trabajo.

```
aws omics start-run
       --workflow-id workflow id \
        ...
       --storage-type STATIC
       --storage-capacity 2400
```

Ejecute un flujo de trabajo de GPU

También puedes especificar un ID de flujo de trabajo de GPU, como se muestra en el siguiente ejemplo:

```
aws omics start-run
       --workflow-id workflow id \
       --role-arn arn:aws:iam::1234567892012:role/service-role/
OmicsWorkflow-20221004T164236 \
       --name GPUTestRunModel \
       --output-uri s3://amzn-s3-demo-bucket1
```

Obtén información sobre una ejecución

Puedes usar el ID de la respuesta con la API get-run para comprobar el estado de una ejecución, como se muestra.

```
aws omics get-run --id run id
```

La respuesta de esta operación de API indica el estado de la ejecución del flujo de trabajo. Los estados posibles son `PENDING` `STARTINGRUNNING`, y`COMPLETED`. Cuando se ejecuta`COMPLETED`, puede encontrar un archivo de salida llamado `outfile.txt` en el bucket de Amazon S3 de salida, en una carpeta que lleva el nombre del ID de la ejecución.

La operación de API get-run también devuelve otros detalles, como si el flujo de trabajo es `Ready2Run` o`PRIVATE`, el motor del flujo de trabajo y los detalles del acelerador. En el siguiente ejemplo, se muestra la respuesta de get-run para una ejecución de un flujo de trabajo privado, descrita en la WDL, con un acelerador de GPU y sin etiquetas asignadas a la ejecución.

```
{
    "arn": "arn:aws:omics:us-west-2:123456789012:run/7830534",
    "id": "7830534",
    "uuid":"96c57683-74bf-9d6d-ae7e-f09b097db14a",
    "outputUri":"s3://bucket/folder/8405154/96c57683-74bf-9d6d-ae7e-f09b097db14a"
```

```
    "status": "COMPLETED",
    "workflowId": "4074992",
    "workflowType": "PRIVATE",
    "workflowVersionName": "3.0.0",
    "roleArn": "arn:aws:iam::123456789012:role/service-role/
OmicsWorkflow-20221004T164236",
    "name": "RunGroupMaxGpuTest",
    "runGroupId": "9938959",
    "digest":
 "sha256:a23a6fc54040d36784206234c02147302ab8658bed89860a86976048f6cad5ac",
    "accelerators": "GPU",
    "outputUri": "s3://amzn-s3-demo-bucket1",
    "startedBy": "arn:aws:sts::123456789012:assumed-role/Admin/<role_name>",
    "creationTime": "2023-04-07T16:44:22.262471+00:00",
    "startTime": "2023-04-07T16:56:12.504000+00:00",
    "stopTime": "2023-04-07T17:22:29.908813+00:00",
    "tags": {}
}
```

Puedes ver el estado de todas las ejecuciones con la operación de API list-runs, como se muestra.

```
aws omics list-runs
```

Para ver todas las tareas completadas en una ejecución específica, usa la list-run-tasksAPI.

```
aws omics list-run-tasks --id task ID
```

Para obtener los detalles de una tarea específica, usa la get-run-task API.

```
aws omics get-run-task --id <run_id> --task-id task ID
```

Una vez completada la ejecución, los metadatos se envían a CloudWatch Under the Stream**`manifest/run/<run ID>/<run UUID>`**.

A continuación, se muestra un ejemplo del manifiesto.

```
{
    "arn": "arn:aws:omics:us-east-1:123456789012:run/1695324",
    "creationTime": "2022-08-24T19:53:55.284Z",
    "resourceDigests": {
      "s3://omics-data/broad-references/hg38/v0/Homo_sapiens_assembly38.dict":
 "etag:3884c62eb0e53fa92459ed9bff133ae6",
```

```
      "s3://omics-data/broad-references/hg38/v0/Homo_sapiens_assembly38.fasta":
 "etag:e307d81c605fb91b7720a08f00276842-388",
      "s3://omics-data/broad-references/hg38/v0/Homo_sapiens_assembly38.fasta.fai":
 "etag:f76371b113734a56cde236bc0372de0a",
      "s3://omics-data/intervals/hg38-mjs-whole-chr.500M.intervals":
 "etag:27fdd1341246896721ec49a46a575334",
      "s3://omics-data/workflow-input-lists/dragen-gvcf-list.txt":
 "etag:e22f5aeed0b350a66696d8ffae453227"
    },
    "digest":
 "sha256:a5baaff84dd54085eb03f78766b0a367e93439486bc3f67de42bb38b93304964",
    "engine": "WDL",
    "main": "gatk4-basic-joint-genotyping-v2.wdl",
    "name": "1044-gvcfs",
    "outputUri": "s3://omics-data/workflow-output",
    "parameters": {
      "callset_name": "cohort",
      "input_gvcf_uris": "s3://omics-data/workflow-input-lists/dragen-gvcf-list.txt",
      "interval_list": "s3://omics-data/intervals/hg38-mjs-whole-chr.500M.intervals",
      "ref_dict": "s3://omics-data/broad-references/hg38/v0/
Homo_sapiens_assembly38.dict",
      "ref_fasta": "s3://omics-data/broad-references/hg38/v0/
Homo_sapiens_assembly38.fasta",
      "ref_fasta_index": "s3://omics-data/broad-references/hg38/v0/
Homo_sapiens_assembly38.fasta.fai"
    },
    "roleArn": "arn:aws:iam::123456789012:role/OmicsServiceRole",
    "startedBy": "arn:aws:sts::123456789012:assumed-role/admin/ahenroid-Isengard",
    "startTime": "2022-08-24T20:08:22.582Z",
    "status": "COMPLETED",
    "stopTime": "2022-08-24T20:08:22.582Z",
    "storageCapacity": 9600,
    "uuid": "a3b0ca7e-9597-4ecc-94a4-6ed45481aeab",
    "workflow": "arn:aws:omics:us-east-1:123456789012:workflow/1558364",
    "workflowType": "PRIVATE"
  },
  {
    "arn": "arn:aws:omics:us-east-1:123456789012:task/1245938",
    "cpus": 16,
    "creationTime": "2022-08-24T20:06:32.971290",
    "image": "123456789012.dkr.ecr.us-west-2.amazonaws.com/gatk",
    "imageDigest":
 "sha256:8051adab0ff725e7e9c2af5997680346f3c3799b2df3785dd51d4abdd3da747b",
    "memory": 32,
```

```
    "name": "geno-123",
    "run": "arn:aws:omics:us-east-1:123456789012:run/1695324",
    "startTime": "2022-08-24T20:08:22.278Z",
    "status": "SUCCESS",
    "stopTime": "2022-08-24T20:08:22.278Z",
    "uuid": "44c1a30a-4eee-426d-88ea-1af403858f76"
  },
  ...
```

Los metadatos de ejecución no se eliminan si no están presentes en los CloudWatch registros.

Vuelva a ejecutar una ejecución HealthOmics

En el caso de las ejecuciones que aún no hayas eliminado, usa la consola o la API para volver a ejecutar la ejecución. Para las ejecuciones que hayas eliminado, usa la herramienta. HealthOmics rerun

Temas

- Vuelva a ejecutar una ejecución con la consola
- Vuelva a ejecutar una ejecución con la API
- Uso de la herramienta Rerun

Vuelva a ejecutar una ejecución con la consola

Desde la consola, sigue estos pasos para volver a ejecutar una ejecución:

1. Abra la consola de HealthOmics .
2. Si es necesario, abre el panel de navegación izquierdo (≡). Elija Ejecuciones.
3. En la página Ejecuciones, seleccione la ejecución que desee volver a ejecutar.
4. En el menú de acciones situado encima de la tabla, selecciona Volver a ejecutar.

Vuelva a ejecutar una ejecución con la API

Usa la operación StartRun de API para volver a ejecutar una ejecución existente. Proporcione las siguientes entradas obligatorias:

- Un ARN de rol de servicio ()`roleArn`.
- El ID de la ejecución que se va a duplicar (`runId`).

- Una ubicación de Amazon S3 donde la ejecución guarda las salidas de la ejecución (`outputUri`).

```
aws omics start-run
     --run-id run id \
     --role-arn arn:aws:iam::1234567892012:role/service-role/
OmicsWorkflow-20221004T164236 \
     --output-uri s3://workflow-output-b6f2fce1
```

Uso de la herramienta Rerun

En el caso de una ejecución eliminada, puede descargar y utilizar la HealthOmics rerun herramienta para volver a ejecutarla. La herramienta recupera la información de ejecución del manifiesto de CloudWatch registros. Descargue la rerun herramienta desde el GitHub repositorio de HealthOmics herramientas.

En el siguiente ejemplo, se muestra cómo utilizar la rerun herramienta.

```
aws-healthomics-rerun 9876543
```

Si la ejecución existe en CloudWatch, recibirá una respuesta similar a la del siguiente ejemplo de salida. Si el flujo de trabajo ya no existe, recibirá un mensaje de error.

```
Original request:
{
  "workflowId": "9679729",
  "roleArn": "arn:aws:iam::123456789012:role/DemoRole",
  "name": "sample_rerun",
  "parameters": {
    "image": "123456789012.dkr.ecr.us-west-2.amazonaws.com/default:latest",
    "file1": "omics://123456789012.storage.us-west-2.amazonaws.com/8647780323/
readSet/6389608538"
  },
  "outputUri": "s3://workflow-output-bcf2fcb1"
}
StartRun request:
{
  "workflowId": "9679729",
  "roleArn": "arn:aws:iam::123456789012:role/DemoRole",
  "name": "new test",
  "parameters": {
```

```
    "image": "123456789012.dkr.ecr.us-west-2.amazonaws.com/default:latest",
    "file1": "omics://123456789012.storage.us-west-2.amazonaws.com/8647780323/
readSet/6389608538"
  },
  "outputUri": "s3://workflow-output-bcf2fcb1"
}
StartRun response:
{
  "arn": "arn:aws:omics:us-west-2:123456789012:run/9171779",
  "id": "9171779",
  "status": "PENDING",
  "tags": {}
}
```

Clona una carrera en HealthOmics

Puede clonar una ejecución existente mediante la HealthOmics consola. La clonación crea una nueva ejecución con los valores de configuración de la ejecución clonada. Puede modificar estos valores predeterminados y añadir otras entradas opcionales.

1. Abra la consola de HealthOmics .
2. Si es necesario, abra el panel de navegación izquierdo (≡). Elija Ejecuciones.
3. En la página Ejecuciones, seleccione la ejecución que desee clonar.
4. En el menú de acciones situado encima de la tabla, seleccione Clonar ejecución. La consola abre el formulario de ejecución de clonación. El formulario es idéntico a Start run, excepto que la consola rellena el formulario con todos los valores relevantes de la ejecución clonada.

   La consola crea un nuevo ID de ejecución para el clon de ejecución y agrega este ID de ejecución como sufijo al nombre de la ejecución.

   A medida que avance por las páginas del formulario, podrá ajustar los valores de configuración según sea necesario.
5. Tras revisar la configuración de ejecución, elija Iniciar ejecución.

Cancela una carrera HealthOmics

Puedes cancelar una carrera si su estado es `PENDINGSTARTING`,`RUNNING`, o`STOPPING`.

> **Note**
>
> Al cancelar una ejecución, HealthOmics no guarda ninguno de los resultados de la ejecución.

Desde la consola, sigue estos pasos para cancelar una ejecución:

1. Abra la consola de HealthOmics .
2. Si es necesario, abre el panel de navegación izquierdo (≡). Elija Ejecuciones.
3. En la página Ejecuciones, seleccione la ejecución que desee cancelar.
4. La consola abre la página de detalles de la ejecución. En el banner de estado de la parte superior de la página, selecciona Detener la ejecución.
5. Introduzca confirmar para detener la ejecución.

Para cancelar una ejecución mediante la API, utilice la operación CancelRun API.

En el siguiente ejemplo, se muestra cómo cancelar una ejecución mediante AWS CLI . Para ejecutar el ejemplo, sustituya el *run id* por el ID de la ejecución que desea cancelar. Si se ejecuta correctamente, no hay respuesta.

```
aws omics cancel-run --id run id
```

Eliminar una carrera en HealthOmics

Cuando ya no necesites una ejecución, puedes eliminarla mediante la AWS CLI API o la consola. Puedes eliminar una ejecución cuando su estado es `COMPLETED` o`CANCELED`.

Desde la consola, sigue estos pasos para eliminar una ejecución:

1. Abra la consola de HealthOmics .
2. Si es necesario, abre el panel de navegación izquierdo (≡). Elija Ejecuciones.
3. En la página Ejecuciones, seleccione una o más corridas para eliminarlas.
4. En el menú de acciones situado encima de la tabla, seleccione Eliminar.
5. En el formulario modal, escriba confirm para confirmar la eliminación.

El siguiente AWS CLI comando elimina una ejecución. Para ejecutar el ejemplo, sustituya el *run id* por el ID de la ejecución que desee eliminar. No hay respuesta si la ejecución se elimina correctamente.

```
aws omics delete-run --id run id
```

Uso de grupos de HealthOmics ejecución

Si lo desea, puede crear un grupo de ejecuciones para limitar los recursos informáticos de las ejecuciones que añada al grupo. Los grupos de carreras pueden ayudarte a:

- Coloca tus carreras en cola para no sobrepasar los límites de servicio.
- Establece una duración máxima de ejecución para detectar las tareas que se están agotando.
- Administra la prioridad de cada ejecución para que las más importantes se completen primero.

Si estableces el máximo de vCPU, GPU o ejecuciones simultáneas, las tareas en ejecución se pondrán en cola cuando se alcance el máximo. Si estableces una duración máxima de ejecución, la ejecución fallará si supera la duración máxima.

Usa la configuración de prioridad de ejecución para establecer la prioridad dentro de un grupo de ejecución.

Los límites de servicio tienen prioridad sobre los límites de los grupos de ejecución. Por ejemplo, si estableces el máximo de un grupo de carreras en un valor superior al máximo de tu servicio en una región, HealthOmics se aplicará el máximo de servicio.

Temas

- Prioridad de ejecución
- Crea un grupo de carreras mediante la consola
- Cree un grupo de ejecución mediante la CLI
- Elimina un grupo de carreras mediante la consola
- Eliminar un grupo de ejecución mediante la CLI

Prioridad de ejecución

Puede usar la prioridad de ejecución para establecer la prioridad de las ejecuciones en un grupo de ejecuciones.

Si varias ejecuciones tienen la misma prioridad, la ejecución que se inició primero tiene la prioridad más alta.

También puedes establecer una prioridad para una ejecución que no esté en un grupo de carreras. La prioridad se compara con las prioridades de todas las demás ejecuciones que no están en un grupo de carreras

La prioridad de las carreras se establece cuando se inicia la ejecución. Para obtener más información, consulte Iniciar una carrera en HealthOmics.

Crea un grupo de carreras mediante la consola

Para crear un grupo de ejecución

1. Abra la consola de HealthOmics .
2. Si es necesario, abre el panel de navegación izquierdo (≡). Seleccione Ejecutar grupos.
3. En la página Ejecutar grupos, elija Crear grupo de ejecución.
4. En la página de detalles para crear un grupo de carreras, proporcione la siguiente información
   - Nombre del grupo de carreras: nombre exclusivo para este grupo de ejecución.
   - Número máximo de vCPU para ejecuciones simultáneas: la cantidad máxima de v CPUs que se puede ejecutar simultáneamente en todas las ejecuciones activas del grupo de ejecuciones.
   - Máximo GPUs: el número máximo GPUs que se puede ejecutar simultáneamente en todas las ejecuciones activas del grupo de ejecuciones.
   - Tiempo máximo de ejecución (minutos) por ejecución: el tiempo máximo de cada ejecución (en minutos). Si una ejecución supera el tiempo máximo de ejecución, la ejecución falla automáticamente.
   - Número máximo de ejecuciones simultáneas: el número máximo de ejecuciones que se pueden ejecutar al mismo tiempo.
5. (opcional) Puedes añadir hasta 50 etiquetas al grupo de carreras.
6. Elige Crear grupo de carreras.

Cree un grupo de ejecución mediante la CLI

Para crear un grupo de carreras, usa la operación de create-run-groupAPI para crear un grupo de carreras con el nombre`TestRunGroup`. En el siguiente ejemplo, se establece un máximo de 20 CPUs GPUs, 10 y 5 ejecuciones y una duración máxima de 600 minutos.

```
aws omics create-run-group --name TestRunGroup \
--max-cpus 20 \
--max-gpus 10 \
--max-duration 600 \
--max-runs 5
```

La respuesta de esta operación de API incluye el ID de la nueva operación creada`RunGroup`.

```
{
    "arn": "arn:aws:omics:us-west-2:12345678901:runGroup/2839621",
    "id": "2839621",
    "tags": {}
}
```

Para obtener información adicional sobre el grupo de ejecución, usa este ID con la operación de get-run-groupAPI, como se muestra en el siguiente ejemplo.

```
aws omics get-run-group --id run group id
```

La respuesta incluye la configuración de límites para el grupo de ejecuciones y las etiquetas asignadas.

```
{
    "arn": "arn:aws:omics:us-west-2:776893852117:runGroup/2839621",
    "id": "2839621",
    "name": "TestRunGroup",
    "maxCpus": 20,
    "maxRuns": 5,
    "maxDuration": 600,
    "creationTime": "2024-06-12T15:35:39.191730+00:00",
    "tags": {},
    "maxGpus": 10
}
```

También puedes usar la operación de la list-run-groupAPI para ver todos los grupos de carreras creados.

```
aws omics list-run-groups
```

Elimina un grupo de carreras mediante la consola

Puede eliminar un grupo de carreras si no hay ninguna ejecución asociada a ese grupo de carreras con el estado `PENDINGSTARTING`,`RUNNING`, o`STOPPING`.

Para eliminar un grupo de carreras, sigue estos pasos.

1. Abra la consola de HealthOmics .
2. Si es necesario, abre el panel de navegación izquierdo (≡). Seleccione Ejecutar grupos.
3. En la página Ejecutar grupos, elija el grupo de ejecución que desee eliminar y elija Eliminar entre xx.

Eliminar un grupo de ejecución mediante la CLI

Puede eliminar un grupo de carreras si no hay ninguna ejecución asociada a ese grupo de ejecuciones con el estado `PENDINGSTARTING`,`RUNNING`, o`STOPPING`.

En el siguiente ejemplo, se muestra cómo se puede utilizar AWS CLI para eliminar un grupo de carreras. No recibirás ninguna respuesta. Para ejecutar el ejemplo, sustituya el *`run group id`* por el ID del grupo de ejecución que desee eliminar.

```
aws omics delete-run-group --id run group id
```

Almacenamiento en caché de llamadas para ejecuciones HealthOmics

AWS HealthOmics admite el almacenamiento en caché de llamadas, también conocido como currículum, para flujos de trabajo privados. El almacenamiento en caché de llamadas guarda los resultados de las tareas de flujo de trabajo completadas una vez finalizada la ejecución. Las ejecuciones posteriores pueden utilizar los resultados de las tareas de la memoria caché, en lugar de volver a calcular los resultados de las tareas. El almacenamiento de llamadas en caché reduce el uso

de recursos informáticos, lo que se traduce en una menor duración de las ejecuciones y un ahorro de costes informáticos.

Puede acceder a los archivos de salida de las tareas en caché una vez finalizada la ejecución. Para realizar tareas avanzadas de depuración y solución de problemas, puede almacenar en caché los archivos de tareas intermedias especificando estos archivos como resultados de las tareas en la definición del flujo de trabajo.

Puede utilizar el almacenamiento en caché de llamadas para guardar los resultados de las tareas finalizadas tras las ejecuciones fallidas. La siguiente ejecución comienza con la última tarea completada correctamente, en lugar de volver a calcular las tareas completadas.

Si HealthOmics no encuentra una entrada de caché que coincida con una tarea, la ejecución no fallará. HealthOmics vuelve a calcular la tarea y las tareas dependientes.

Para obtener información sobre la solución de problemas de almacenamiento en caché de llamadas, consulte. Solución de problemas de almacenamiento en caché de llamadas

Temas

- Cómo funciona el almacenamiento en caché de llamadas
- Crear una caché de ejecución
- Actualización de una caché de ejecución
- Eliminar una caché de ejecución
- Contenido de una caché de ejecución
- Funciones de almacenamiento en caché específicas del motor
- Uso de la caché de ejecución

Cómo funciona el almacenamiento en caché de llamadas

Para usar el almacenamiento en caché de llamadas, debe crear una caché de ejecución y configurarla para que tenga una ubicación de Amazon S3 asociada para los datos en caché. Al iniciar una ejecución, debe especificar la caché de ejecución. Una caché de ejecución no está dedicada a un flujo de trabajo. Las ejecuciones de varios flujos de trabajo pueden usar la misma caché.

Durante la fase de exportación de una ejecución, el sistema exporta los resultados de las tareas completadas a la ubicación de Amazon S3. Para exportar archivos de tareas intermedias, declare estos archivos como resultados de tareas en la definición del flujo de trabajo. El almacenamiento

en caché de llamadas también guarda internamente los metadatos y crea hashes únicos para cada entrada de la caché.

Para cada tarea de una ejecución, el motor de flujo de trabajo detecta si hay una entrada de caché coincidente para esta tarea. Si no hay ninguna entrada de caché coincidente, HealthOmics calcula la tarea. Si hay una entrada de caché coincidente, el motor recupera los resultados almacenados en caché.

Para hacer coincidir las entradas de la caché, HealthOmics utiliza el mecanismo de hash que se incluye en los motores de flujo de trabajo nativos. HealthOmicsamplía estas implementaciones de hash existentes para tener en cuenta HealthOmics variables, como las ETags de S3 y los resúmenes de contenedores ECR.

HealthOmics admite el almacenamiento en caché de llamadas para las siguientes versiones de lenguajes de flujo de trabajo:

- Las versiones 1.0 y 1.1 de WDL y la versión de desarrollo
- Nextflow, versiones 23.10 y 24.10
- Todas las versiones de CWL

> **Note**
>
> HealthOmics no admite el almacenamiento en caché de llamadas para los flujos de trabajo de Ready2Run.

Temas

- Modelo de responsabilidad compartida
- Requisitos de almacenamiento en caché para las tareas
- Ejecute el rendimiento de la caché
- Almacene en caché los eventos de retención e invalidación de datos

Modelo de responsabilidad compartida

Los usuarios comparten la responsabilidad de determinar si las tareas y AWS las ejecuciones son buenas candidatas para el almacenamiento de llamadas en caché. El almacenamiento en caché de

llamadas logra los mejores resultados cuando todas las tareas son idempotentes (las ejecuciones repetidas de una tarea con las mismas entradas producen los mismos resultados).

Sin embargo, si una tarea incluye elementos no deterministas (como la generación de números aleatorios o la hora del sistema), las ejecuciones repetidas de la tarea con las mismas entradas pueden dar como resultado resultados diferentes. Esto puede afectar a la eficacia del almacenamiento en caché de las llamadas de las siguientes maneras:

- Si HealthOmics utiliza una entrada de caché (creada por una ejecución anterior) que no es idéntica a la salida que generaría la ejecución de la tarea para la ejecución actual, la ejecución puede producir resultados diferentes a los de la misma ejecución sin almacenamiento en caché.
- HealthOmics es posible que no encuentre una entrada de caché coincidente para una tarea que debería coincidir, debido a que los resultados de la tarea no son deterministas. Si no encuentra la entrada de caché válida, la ejecución vuelve a calcular innecesariamente la tarea, lo que reduce las ventajas de ahorro de costes que supone el uso del almacenamiento en caché de llamadas.

Los siguientes son comportamientos conocidos de las tareas que pueden provocar resultados no deterministas que afecten a los resultados del almacenamiento en caché de las llamadas:

- Uso de generadores de números aleatorios.
- Depende del tiempo del sistema.
- Uso de la simultaneidad (las condiciones de carrera pueden provocar una variación en la salida).
- Obtener archivos locales o remotos más allá de lo especificado en los parámetros de entrada de la tarea.

Para ver otros escenarios que pueden provocar un comportamiento no determinista, consulte Entradas de procesos no deterministas en el sitio de documentación de Nextflow.

Si sospecha que una tarea produce resultados que no son deterministas, considere la posibilidad de utilizar las funciones del motor de flujo de trabajo para evitar almacenar en caché tareas específicas que no sean deterministas. Para obtener instrucciones sobre cómo inhabilitar el almacenamiento en caché para tareas individuales en cada uno de los lenguajes de flujo de trabajo compatibles, consulte. Funciones de almacenamiento en caché específicas del motor

Le recomendamos que revise detenidamente sus requisitos específicos de flujo de trabajo y tareas antes de habilitar el almacenamiento en caché de llamadas en cualquier entorno en el que un almacenamiento en caché de llamadas ineficaz o resultados diferentes a los esperados puedan

suponer un riesgo. Por ejemplo, las posibles limitaciones del almacenamiento en caché de llamadas deben considerarse detenidamente al determinar si el almacenamiento en caché de llamadas es adecuado para los casos de uso clínico.

Requisitos de almacenamiento en caché para las tareas

HealthOmics almacena en caché los resultados de las tareas que cumplen los siguientes requisitos:

- La tarea debe definir un contenedor. HealthOmics no almacenará en caché los resultados de una tarea sin contenedor.
- La tarea debe producir uno o más resultados. Los resultados de la tarea se especifican en la definición del flujo de trabajo.
- La definición del flujo de trabajo no debe utilizar valores dinámicos. Por ejemplo, si pasa un parámetro a una tarea con un valor que aumenta con cada ejecución, HealthOmics no se almacenan en caché los resultados de la tarea.

> **Note**
>
> Si varias tareas de una ejecución utilizan la misma imagen de contenedor, HealthOmics proporciona la misma versión de imagen para todas estas tareas. Tras HealthOmics extraer la imagen, ignora cualquier actualización de la imagen del contenedor durante la ejecución. Este enfoque proporciona una experiencia predecible y coherente y evita los posibles problemas que puedan surgir a causa de las actualizaciones de la imagen del contenedor que se despliegan a mitad de la ejecución.

Ejecute el rendimiento de la caché

Al activar el almacenamiento en caché de llamadas durante una ejecución, es posible que notes los siguientes impactos en el rendimiento de la ejecución:

- Durante la primera ejecución, HealthOmics guarda los datos en caché de las tareas de la ejecución. Es posible que los tiempos de exportación sean más largos durante esta ejecución, ya que el almacenamiento en caché de las llamadas aumenta la cantidad de datos de exportación.
- En las siguientes ejecuciones, al reanudar una ejecución desde la memoria caché, es posible que se reduzca el número de pasos de procesamiento y el tiempo de ejecución.

- Si también opta por declarar los archivos intermedios como salidas, los tiempos de exportación podrían ser incluso más largos, ya que estos datos pueden ser más detallados.

Almacene en caché los eventos de retención e invalidación de datos

El objetivo principal de una caché de ejecución es optimizar el cálculo de las tareas en ejecución. Si hay una entrada de caché válida que coincida con una tarea, HealthOmics utiliza la entrada de caché en lugar de volver a calcular la tarea. De lo contrario, HealthOmics vuelve al comportamiento predeterminado del servicio, que consiste en volver a calcular la tarea y las tareas dependientes. Con este enfoque, los errores en la memoria caché no provocan un error en la ejecución.

Se recomienda administrar el tamaño de la caché de ejecución. Con el tiempo, es posible que las entradas de la caché dejen de ser válidas debido a las actualizaciones del motor de flujo de trabajo o del HealthOmics servicio, o a los cambios realizados en la ejecución o en las tareas de ejecución. En las siguientes secciones se proporcionan detalles adicionales.

Temas

- Actualizaciones de la versión del manifiesto y actualización de los datos
- Ejecute el comportamiento de la caché
- Controle el tamaño de la caché de ejecución

Actualizaciones de la versión del manifiesto y actualización de los datos

Periódicamente, el HealthOmics servicio puede introducir nuevas funciones o actualizaciones del motor de flujo de trabajo que invaliden algunas o todas las entradas de la memoria caché de ejecución. En esta situación, es posible que las ejecuciones pierdan la memoria caché una sola vez.

HealthOmics crea un archivo de manifiesto JSON para cada entrada de caché. En el caso de las ejecuciones iniciadas después del 12 de febrero de 2025, el archivo de manifiesto incluye un parámetro de versión. Si una actualización del servicio invalida alguna entrada de la caché, HealthOmics incrementa el número de versión para que puedas identificar las entradas de la caché antiguas para eliminarlas.

En el siguiente ejemplo, se muestra un archivo de manifiesto con la versión establecida en 2:

```
{
     "arn": "arn:aws:omics:us-west-2:12345678901:runCache/0123456/
cacheEntry/1234567-195f-3921-a1fa-ffffcef0a6a4",
```

```
    "s3uri": "s3://example/1234567-d0d1-e230-
d599-10f1539f4a32/1348677/4795326/7e8c69b1-145f-3991-a1fa-ffffcef0a6a4",
    "taskArn": "arn:aws:omics:us-west-2:12345678901:task/4567891",
    "workDir": "/mnt/workflow/1234567-d0d1-e230-d599-10f1539f4a32/workdir/call-
TxtFileCopyTask/5w6tn5feyga7noasjuecdeoqpkltrfo3/wxz2fuddlo6hc4uh5s2lreaayczduxdm",
    "files": [
        {
            "name": "output_txt_file",
            "path": "out/output_txt_file/outfile.txt",
            "etag": "ajdhyg9736b9654673b9fbb486753bc8"
        }
    ],
    "nextflowContext": {},
    "otherOutputs": {},
    "version": 2,
  }
```

Para las ejecuciones con entradas de caché que ya no son válidas, reconstruya la caché para crear nuevas entradas válidas. Realice los siguientes pasos para cada ejecución:

1. Inicie la ejecución una vez con la retención de caché establecida en CACHE ALWAYS. Esta ejecución crea las nuevas entradas de caché.
2. Para las siguientes ejecuciones, establezca la retención de la caché en su configuración anterior (CACHE ALWAYS o CACHE ON FAILURE).

Para limpiar las entradas de caché que ya no son válidas, puede eliminarlas del bucket de caché de Amazon S3. HealthOmics nunca reutiliza estas entradas de caché. Si decides conservar las entradas que no son válidas, esto no afectará a tus tiradas.

> **Note**
>
> El almacenamiento en caché de llamadas guarda los datos de salida de las tareas en la ubicación de Amazon S3 especificada para la caché, lo que supone un coste para usted. Cuenta de AWS

Ejecute el comportamiento de la caché

Puede configurar el comportamiento de la caché de ejecución para guardar los resultados de las tareas de las ejecuciones que fallan (caché en caso de error) o de todas las ejecuciones (caché

siempre). Al crear una caché de ejecuciones, se establece el comportamiento predeterminado de la caché para todas las ejecuciones que utilizan esta caché. Puede anular el comportamiento predeterminado al iniciar una ejecución.

Cache on failurees útil si está depurando un flujo de trabajo que falla después de que varias tareas se hayan completado correctamente. La ejecución siguiente se reanudará desde la última tarea completada correctamente si todas las variables únicas consideradas en el hash son idénticas a las de la ejecución anterior.

Cache alwayses útil si está actualizando una tarea en un flujo de trabajo que se completa correctamente. Te recomendamos que sigas estos pasos:

1. Cree una nueva ejecución. Establezca el comportamiento de la caché en Caché siempre e inicie la ejecución.
2. Una vez completada la ejecución, actualice la tarea en el flujo de trabajo e inicie una nueva ejecución con el comportamiento configurado Guardar siempre en caché. Esta ejecución procesa la tarea actualizada y cualquier tarea posterior que dependa de la tarea actualizada. Todas las demás tareas utilizan los resultados almacenados en caché.
3. Repita el paso 2 según sea necesario, hasta que se complete el desarrollo de la tarea actualizada.
4. Utilice la tarea actualizada según sea necesario en futuras ejecuciones. Recuerde cambiar las siguientes ejecuciones a la memoria caché en caso de error si piensa utilizar entradas nuevas o diferentes para estas ejecuciones.

> **Note**
>
> Recomendamos utilizar siempre el modo Caché mientras se utiliza el mismo conjunto de datos de prueba, pero no para un lote de ejecuciones. Si configura este modo para un lote grande de ejecuciones, el sistema puede exportar grandes cantidades de datos a Amazon S3, lo que se traduce en un aumento de los tiempos de exportación y de los costes de almacenamiento.

Controle el tamaño de la caché de ejecución

HealthOmics no elimina ni archiva automáticamente ningún dato de la caché de ejecución ni aplica las reglas de limpieza de Amazon S3 para gestionar los datos de la caché. Le recomendamos que

realice limpiezas de caché periódicas para ahorrar en los costes de almacenamiento de Amazon S3 y mantener el tamaño de la caché de ejecución manejable. Puede eliminar los archivos directamente o establecer retention/replication políticas de datos en el depósito de caché de ejecución.

Por ejemplo, puede configurar una política de ciclo de vida de Amazon S3 para que los objetos caduquen después de 90 días, o puede limpiar manualmente los datos de la caché al final de cada proyecto de desarrollo.

La siguiente información puede ayudarle a administrar el tamaño de los datos de la caché:

- Puede ver la cantidad de datos que hay en la memoria caché consultando Amazon S3. HealthOmics no supervisa ni informa sobre el tamaño de la caché.
- Si eliminas una entrada de caché válida, la ejecución posterior no fallará. HealthOmics vuelve a calcular la tarea y las tareas dependientes.
- Si modifica los nombres de la caché o las estructuras de directorios de manera que no HealthOmics pueda encontrar una entrada coincidente para una tarea, HealthOmics vuelve a calcular la tarea.

Si necesitas comprobar si una entrada de caché sigue siendo válida, comprueba el número de versión del manifiesto de la caché. Para obtener más información, consulte Actualizaciones de la versión del manifiesto y actualización de los datos.

Crear una caché de ejecución

Cuando crea una caché de ejecución, especifica una ubicación de Amazon S3 para los datos de la caché. Se debe poder acceder a estos datos de forma inmediata. El almacenamiento en caché de llamadas no recupera los objetos archivados en Glacier (como las clases de almacenamiento GFR y GDA).

Si el depósito de Amazon S3 para los datos de la caché es propiedad de otra Cuenta de AWS persona, proporcione ese ID de cuenta al crear la caché de ejecución.

Crear una caché de ejecución mediante la consola

Desde la consola, sigue estos pasos para crear una caché de ejecución.

1. Abra la consola de HealthOmics .
2. Si es necesario, abre el panel de navegación izquierdo (≡). Seleccione Ejecutar cachés.

3. En la página Ejecutar cachés, selecciona Crear caché de ejecución.
4. En el panel de detalles de la caché de ejecución de la página Crear caché de ejecución, configure estos campos:

   a. Introduzca un nombre para la caché de ejecución.

   b. (Opcional) Introduzca una descripción.

   c. Introduzca una ubicación S3 para la salida almacenada en caché. Elija un depósito en la misma región que su flujo de trabajo.

   d. (Opcional) Introduzca el nombre Cuenta de AWS del propietario del bucket para verificar la propiedad del bucket. Si no ingresas ningún valor, el valor predeterminado es tu ID de cuenta.

   e. En Comportamiento de la caché, configura el comportamiento predeterminado (ya sea para almacenar en caché los resultados de las ejecuciones fallidas o de todas las ejecuciones). Al iniciar una ejecución, si lo desea, puede anular el comportamiento predeterminado.
5. (Opcional) Asocie una o más etiquetas a la caché de ejecución.
6. Seleccione Crear caché de ejecución. La consola muestra la nueva caché de ejecución en la tabla Cachés de ejecución.

Creación de una caché de ejecución mediante la CLI

Utilice el comando create-run-cacheCLI para crear una caché de ejecución. El comportamiento predeterminado de la caché es`CACHE_ON_FAILURE`.

```
aws omics create-run-cache \
      --name "workflow 123 run cache" \
      --description "my run cache" \
      --cache-s3-location "s3://amzn-s3-demo-bucket" \
      --cache-behavior "CACHE_ALWAYS"                \
      --cache-bucket-owner-id  "111122223333"
```

Si la creación se realiza correctamente, recibirá una respuesta con los siguientes campos.

```
{
  "arn": "string",
  "id": "string",
  "status": "ACTIVE"
```

```
  "tags": {}
  }
```

Actualización de una caché de ejecución

Puede cambiar el nombre, la descripción, las etiquetas o el comportamiento de la caché, pero no su ubicación en S3.

Actualización de una caché de ejecución mediante la consola

Desde la consola, sigue estos pasos para actualizar una caché de ejecución.

1. Abra la consola de HealthOmics .
2. Si es necesario, abre el panel de navegación izquierdo (≡). Seleccione Ejecutar cachés.
3. En la tabla Ejecutar cachés, elija la caché de ejecución que desee actualizar y, a continuación, seleccione Editar.
4. En el panel de detalles de la caché de ejecución, puede actualizar los campos de nombre, descripción y comportamiento de la caché de ejecución.
5. (Opcional) Asocie una o más etiquetas nuevas a la caché de ejecución o elimine las etiquetas existentes.
6. Selecciona Guardar caché de ejecución.

Actualización de una caché de ejecución mediante la CLI

Utilice el comando update-run-cacheCLI para actualizar una caché de ejecución.

```
aws omics update-run-cache \
      --name "workflow 123 run cache" \
      --id "workflow id" \
      --description "my run cache" \
      --cache-behavior "CACHE_ALWAYS"
```

Si la actualización se realiza correctamente, recibirá una respuesta sin campos de datos.

Eliminar una caché de ejecución

Puede eliminar una caché de ejecución si no la está utilizando ninguna ejecución activa. Si alguna ejecución utiliza la caché de ejecuciones, espere a que se completen o puede cancelarlas.

Al eliminar una caché de ejecución, se eliminan el recurso y sus metadatos, pero no se eliminan los datos de Amazon S3. Después de eliminar la caché, no podrá volver a adjuntarla ni usarla para ejecuciones posteriores.

Los datos en caché permanecen en Amazon S3 para su inspección. Puede eliminar los datos de caché antiguos mediante Delete las operaciones estándar de S3. También puede crear una política de ciclo de vida de Amazon S3 para caducar los datos en caché que ya no utilice.

Eliminar una caché de ejecución mediante la consola

Desde la consola, sigue estos pasos para eliminar una caché de ejecución.

1. Abra la consola de HealthOmics .
2. Si es necesario, abre el panel de navegación izquierdo (≡). Seleccione Ejecutar cachés.
3. En la tabla Ejecutar cachés, elija la caché de ejecución que desee eliminar.
4. En el menú de la tabla Ejecutar cachés, seleccione Eliminar.
5. En el cuadro de diálogo modal, guarde el enlace de datos de la caché de Amazon S3 para consultarlo en el futuro y, a continuación, confirme que desea eliminar la caché de ejecución.

   Puede usar el enlace de Amazon S3 para inspeccionar los datos en caché, pero no puede volver a vincular los datos a otra caché de ejecución. Elimine los datos de la caché cuando haya terminado la inspección.

Eliminar una caché de ejecución mediante la CLI

Utilice el comando delete-run-cacheCLI para eliminar una caché de ejecución.

```
aws omics delete-run-cache \
      --id "my cache id"
```

Si la eliminación se realiza correctamente, recibirá una respuesta sin campos de datos.

Contenido de una caché de ejecución

HealthOmics organiza la caché de carreras con la siguiente estructura en el bucket de S3:

```
s3://{cache.S3location}/{cache.uuid}/runID/taskID/{cacheentry.uuid}/
```

El cache.uuid es el identificador único global de la caché. El cacheentry.uuid es el uuid único a nivel mundial para una tarea en caché. HealthOmics asigna los uuids a las cachés y a las tareas.

Para todos los motores de flujo de trabajo, la caché contiene los siguientes archivos:

- El {cacheentryuuid}.json archivo: HealthOmics crea este archivo de manifiesto, que contiene información sobre la memoria caché, incluida una lista de todos los elementos de la memoria caché y la versión de la memoria caché.
- Archivos de salida de la tarea: cada resultado de la tarea consta de uno o más archivos, según lo defina la tarea.

Para un flujo de trabajo que usa Nextflow, el motor de Nextflow crea estos archivos adicionales en la memoria caché:

- El command.out archivo: este archivo contiene el contenido estándar de la ejecución de la tarea.
- El .exitcode archivo: este archivo contiene el código de salida de la tarea (un entero).

> **Note**
>
> Si desea acceder a los archivos de tareas intermedias de su caché de ejecución para solucionar problemas avanzados, declare estos archivos como resultados de tareas en la definición del flujo de trabajo.

Funciones de almacenamiento en caché específicas del motor

HealthOmics intenta proporcionar una implementación coherente del almacenamiento de llamadas en caché en todos los motores de flujo de trabajo. Existen algunas diferencias según la forma en que cada motor de flujo de trabajo gestiona casos específicos:

- Nextflow
  - No se garantiza el almacenamiento en caché en diferentes versiones de Nextflow. Por ejemplo, si ejecuta una tarea en la versión 23.10.0 y, posteriormente, ejecuta la misma tarea en la versión 24.10.8, HealthOmics podría considerarse que la segunda ejecución es una falta de memoria caché.

  - Puedes desactivar el almacenamiento en caché para tareas individuales mediante la directiva de caché. false Para obtener información sobre esta directiva, consulte los procesos de la especificación Nextflow.
  - HealthOmics usa el modo indulgente de Nextflow, pero no admite el modo de almacenamiento en caché profundo.
  - El almacenamiento en caché evalúa cada objeto S3 individual si se utiliza un patrón global en la ruta S3 hacia las entradas de una tarea. Si añades un objeto nuevo, HealthOmics vuelve a calcular solo las tareas que utilizan el nuevo objeto.
  - HealthOmics no almacena en caché los reintentos de tareas. Este comportamiento es coherente con el comportamiento predeterminado de Nextflow.
- WDL
  - HealthOmics admite el nuevo tipo de «directorio» para las entradas cuando se utiliza la versión de desarrollo del flujo de trabajo de la WDL. Para el almacenamiento en caché de llamadas, si algún objeto del directorio cambia, HealthOmics vuelve a calcular todas las tareas que entran en el directorio.
  - HealthOmics admite el almacenamiento en caché a nivel de tarea, pero no el almacenamiento en caché a nivel de flujo de trabajo.
  - Puede deshabilitar el almacenamiento en caché para tareas individuales mediante el atributo volatile. Para obtener más información, consulte Deshabilite el almacenamiento en caché a nivel de tarea con el atributo volatile.
- CWL
  - Los resultados constantes de las tareas no se ven de forma explícita en los manifiestos. HealthOmics almacena en caché las salidas constantes como archivos intermedios.
  - Puede controlar el almacenamiento en caché de tareas individuales mediante esta función. WorkReuse

Uso de la caché de ejecución

De forma predeterminada, las ejecuciones no utilizan una caché de ejecución. Para usar una caché para la ejecución, debe especificar la caché de ejecución y el comportamiento de la caché de ejecución al iniciar la ejecución.

Una vez completada una ejecución, puedes usar la consola, los CloudWatch registros o las operaciones de la API para realizar un seguimiento de las visitas a la caché o solucionar los problemas de la caché. Para más detalles, consulte Seguimiento de la información de

almacenamiento en caché de las llamadas y Solución de problemas de almacenamiento en caché de llamadas.

Si una o más tareas de una ejecución generan resultados no deterministas, te recomendamos encarecidamente que no utilices el almacenamiento en caché de llamadas durante la ejecución o que excluyas estas tareas específicas del almacenamiento en caché. Para obtener más información, consulte Modelo de responsabilidad compartida.

> **Note**
>
> Al iniciar una ejecución, proporciona una función de servicio de IAM. Para utilizar el almacenamiento en caché de llamadas, el rol de servicio necesita permiso para acceder a la ubicación Amazon S3 de la caché de ejecución. Para obtener más información, consulte Funciones de servicio para AWS HealthOmics.

Puede utilizar Amazon Q CLI para analizar y gestionar los datos de la caché de ejecución. Para obtener más información, consulte Ejemplos de instrucciones para la CLI de Amazon Q y el tutorial sobre HealthOmics IA generativa de Agentic en. GitHub

Temas

- Configuración de una ejecución con caché de ejecución mediante la consola
- Configuración de una ejecución con caché de ejecución mediante la CLI
- Casos de error en las cachés de ejecución
- Seguimiento de la información de almacenamiento en caché de las llamadas

Configuración de una ejecución con caché de ejecución mediante la consola

Desde la consola, se configura la caché de ejecución de una ejecución al iniciarla.

1. Abra la consola de HealthOmics .
2. Si es necesario, abre el panel de navegación izquierdo (≡). Elija Ejecuciones.
3. En la página Ejecuciones, seleccione la ejecución que desee iniciar.
4. Seleccione Iniciar ejecución y complete los pasos 1 y 2 de Iniciar ejecución tal y como se describe enIniciar una ejecución mediante la consola.

5. En el paso 3 de Iniciar ejecución, elija Seleccionar una caché de ejecución existente.
6. Seleccione la caché en la lista desplegable del ID de caché de ejecución.
7. Para anular el comportamiento predeterminado de la caché de ejecución, elija el comportamiento de la caché de la ejecución. Para obtener más información, consulte Ejecute el comportamiento de la caché.
8. Continúe con el paso 4 de Iniciar la ejecución.

Configuración de una ejecución con caché de ejecución mediante la CLI

Para iniciar una ejecución que utilice una caché de ejecución, añada el parámetro cache-id al comando CLI start-run. Si lo desea, utilice el `cache-behavior` parámetro para anular el comportamiento predeterminado que configuró para la caché de ejecución. El siguiente ejemplo muestra solo los campos de caché del comando:

```
aws omics start-run \
        ...
      --cache-id "xxxxxx"    \
      --cache-behavior  CACHE_ALWAYS
```

Si la operación se realiza correctamente, recibirá una respuesta sin campos de datos.

Casos de error en las cachés de ejecución

En los siguientes escenarios, es HealthOmics posible que no se almacenen en caché los resultados de las tareas, ni siquiera en una ejecución con el comportamiento de caché establecido en Almacenar siempre en caché.

- Si la ejecución detecta un error antes de que la primera tarea se complete correctamente, no habrá ningún resultado de caché que exportar.
- Si se produce un error en el proceso de exportación, HealthOmics no guarda los resultados de la tarea en la ubicación de caché de Amazon S3.
- Si la ejecución falla debido a un filesystem out of space error, el almacenamiento en caché de llamadas no guarda ningún resultado de la tarea.
- Si cancelas una ejecución, el almacenamiento en caché de llamadas no guarda los resultados de ninguna tarea.

- Si se agota el tiempo de espera de la ejecución, el almacenamiento en caché de llamadas no guarda ningún resultado de la tarea, incluso si configuraste la ejecución para usar la caché en caso de error.

Seguimiento de la información de almacenamiento en caché de las llamadas

Puede realizar un seguimiento de los eventos de almacenamiento en caché de llamadas (como las visitas a la caché de ejecución) mediante la consola, la CLI o CloudWatch los registros.

Temas

- Realice un seguimiento de las visitas a la caché mediante la consola
- Realice un seguimiento del almacenamiento en caché de llamadas mediante la CLI
- Realice un seguimiento del almacenamiento en caché de llamadas mediante registros CloudWatch

Realice un seguimiento de las visitas a la caché mediante la consola

En la página de detalles de una ejecución, la tabla Ejecutar tareas muestra la información sobre los aciertos de caché de cada tarea. La tabla también incluye un enlace a la entrada de caché asociada. Utilice el siguiente procedimiento para ver la información de aciertos de la caché de una ejecución.

1. Abra la consola de HealthOmics .
2. Si es necesario, abra el panel de navegación izquierdo (≡). Elija Ejecuciones.
3. En la página Ejecuciones, elija la ejecución que desee inspeccionar.
4. En la página de detalles de la ejecución, seleccione la pestaña Ejecutar tareas para ver la tabla de tareas.
5. Si una tarea tiene un acceso de caché, la columna de aciertos de caché contiene un enlace a la ubicación de entrada de la caché de ejecución en Amazon S3.
6. Elija el enlace para inspeccionar la entrada de la caché de ejecución.

Realice un seguimiento del almacenamiento en caché de llamadas mediante la CLI

Utilice el comando CLI get-run para confirmar si la ejecución utilizó una caché de llamadas.

```
aws omics get-run --id 1234567
```

En la respuesta, si el `cacheId` campo está establecido, la ejecución utiliza esa caché.

Utilice el comando list-run-tasksCLI para recuperar la ubicación de los datos de la caché de cada tarea en caché de la ejecución.

```
aws omics list-run-tasks --id 1234567
```

En la respuesta, si el campo cacheHit de una tarea es verdadero, el campo Caches3uri proporciona la ubicación de los datos de la caché de esa tarea.

También puede usar el comando get-run-taskCLI para recuperar la ubicación de los datos de la caché para una tarea específica:

```
aws omics get-run-task --id 1234567 --task-id <task_id>
```

Realice un seguimiento del almacenamiento en caché de llamadas mediante registros CloudWatch

HealthOmics crea registros de actividad en caché en el grupo de `/aws/omics/WorkflowLog` CloudWatch registros. <cache_id><cache_uuid>Hay un flujo de registro para cada caché de ejecución: RunCache//.

Para las ejecuciones que utilizan el almacenamiento en caché de llamadas, HealthOmics genera entradas de CloudWatch registro para los siguientes eventos:

- crear una entrada de caché (CACHE_ENTRY_CREATED)
- hacer coincidir una entrada de caché (CACHE_HIT)
- no coincide con una entrada de caché (CACHE_MISS)

Para obtener más información sobre estos registros, consulte. Inicia sesión CloudWatch

Utilice la siguiente consulta de CloudWatch Insights en el grupo de `/aws/omics/WorkflowLog` registros para obtener el número de visitas a la caché por ejecución de esta caché:

```
filter @logStream like 'runCache/<CACHE_ID>/'
 fields @timestamp, @message
 filter logMessage like 'CACHE_HIT'
 parse "run: *," as run
 stats count(*) as cacheHits by run
```

Utilice la siguiente consulta para devolver el número de entradas de caché creadas por cada ejecución:

```
filter @logStream like 'runCache/<CACHE_ID>/'
 fields @timestamp, @message
 filter logMessage like 'CACHE_ENTRY_CREATED'
 parse "run: *," as run
 stats count(*) as cacheEntries by run
```

Compartir HealthOmics flujos de trabajo

Como propietario de un flujo de trabajo privado, puedes compartir el flujo de trabajo con alguien Cuenta de AWS de la misma región. Para compartir un flujo de trabajo con más de uno Cuenta de AWS, debe crear varios recursos compartidos del mismo flujo de trabajo.

Como propietario, puede revocar el acceso a un flujo de trabajo compartido eliminando el recurso compartido.

> **Note**
>
> HealthOmics permite automáticamente que un flujo de trabajo compartido acceda al repositorio de Amazon ECR mientras el flujo de trabajo se ejecuta en la cuenta del suscriptor. No es necesario conceder acceso adicional al repositorio para los flujos de trabajo compartidos.

Cuando compartes un flujo de trabajo, el suscriptor puede usar cualquiera de las versiones del flujo de trabajo. Si necesita un control de acceso a nivel de versión para un flujo de trabajo compartido, le recomendamos que cree flujos de trabajo independientes en lugar de utilizar versiones del flujo de trabajo.

Temas

- Suscribirse a un flujo de trabajo compartido
- Supervisar el estado de un flujo de trabajo compartido
- Compartir un flujo de trabajo privado mediante la consola
- Compartir un flujo de trabajo privado mediante la CLI
- Aceptar un flujo de trabajo compartido mediante la consola
- Ejecutar un flujo de trabajo compartido mediante la consola
- Ejecutar un flujo de trabajo compartido mediante la API

Suscribirse a un flujo de trabajo compartido

Para suscribirse a un flujo de trabajo compartido, siga estos pasos generales para aceptar y utilizar el flujo de trabajo:

1. Usa la consola o la API para aceptar el uso compartido. Configura tu región actual en la misma región que la solicitud de compartición.
   - Para encontrar la solicitud para compartir en la consola, ve a la página Todos los recursos compartidos y, a continuación, selecciona la pestaña Compartido conmigo.
2. Usa la consola o la API para crear una ejecución para el flujo de trabajo compartido.
   - Para encontrar la página de detalles del flujo de trabajo en la consola, ve a Compartido conmigo (consulta el paso 1) y, a continuación, selecciona el enlace Recurso para el flujo de trabajo compartido.
3. Usted proporciona sus propios datos de entrada para el flujo de trabajo.
4. El flujo de trabajo compartido se ejecuta en su Cuenta de AWS.

Como suscriptor de un flujo de trabajo compartido, el sistema le impide realizar las siguientes acciones del flujo de trabajo:

- Exportación de un flujo de trabajo compartido
- Volver a ejecutar el flujo de trabajo compartido
  - Se crea una nueva ejecución para el flujo de trabajo compartido.
- Volver a compartir el flujo de trabajo.
- Asignación de una etiqueta al flujo de trabajo.
- Eliminar el flujo de trabajo.
  - Cuando ya no necesite el flujo de trabajo, eliminará el recurso compartido del flujo de trabajo.

Consulte Uso compartido de recursos entre cuentas en AWS HealthOmics para obtener información adicional sobre el uso compartido de recursos.

Supervisar el estado de un flujo de trabajo compartido

HealthOmics envía un evento a EventBridge por cada cambio de estado de un flujo de trabajo compartido. Si desea recibir notificaciones sobre cambios de estado específicos, configure una

EventBridge regla para supervisar los eventos de cambio de estado de los flujos de trabajo compartidos. Por ejemplo:

- Desea recibir una notificación cada vez que reciba una solicitud para compartir un flujo de trabajo y cada vez que un usuario revoque un flujo de trabajo compartido.
- Tras iniciar una solicitud para compartir un flujo de trabajo, querrá recibir una notificación cuando el usuario acepte o rechace la solicitud.

Para obtener más información sobre el uso de eventos, consulteUso EventBridge con AWS HealthOmics.

Compartir un flujo de trabajo privado mediante la consola

Desde la consola, puedes compartir un flujo de trabajo privado con alguien de Cuenta de AWS la misma región que el flujo de trabajo.

Para compartir un flujo de trabajo privado

1. Abra la consola de HealthOmics .
2. Si es necesario, abre el panel de navegación izquierdo (≡). Elija Flujos de trabajo privados.
3. En la tabla Flujos de trabajo de la página Flujos de trabajo privados, seleccione el flujo de trabajo que desee compartir y elija Compartir.
4. En el panel de detalles de la página Compartir flujo de trabajo, introduzca un nombre descriptivo para el recurso compartido e introduzca el nombre Cuenta de AWS del suscriptor.
5. Elija Compartir recurso. La consola muestra los recursos compartidos en la página Todos los recursos compartidos.

El estado inicial del recurso compartido está pendiente. Una vez que el suscriptor acepta la acción, el estado cambia a activo.

Compartir un flujo de trabajo privado mediante la CLI

Utilice la operación de API create-share para crear un flujo de trabajo compartido. El suscriptor principal es Cuenta de AWS el usuario que tendrá acceso al flujo de trabajo.

```
aws omics create-share \
  --resource-arn "arn:aws:omics:us-west-2:555555555555:workflow/123456" \
```

```
  --principal-subscriber "123456789012" \
  --name "my_Share-123"
```

Si la creación se realiza correctamente, recibirá una respuesta con el ID y el estado del recurso compartido.

```
{
"shareId": "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a",
"name": "my_Share-123",
"status": "PENDING"
}
```

El recurso compartido permanece en estado pendiente hasta que el suscriptor lo acepte mediante la operación de `accept-share` API.

Consulte Uso compartido de recursos entre cuentas en AWS HealthOmics para ver otros ejemplos de uso de la API.

Aceptar un flujo de trabajo compartido mediante la consola

Puede usar la consola para aceptar un flujo de trabajo compartido ofrecido. Asegúrese de configurar la consola en la misma región que el flujo de trabajo.

1. Abra la consola de HealthOmics .
2. Si es necesario, abra el panel de navegación izquierdo (≡). Selecciona Todos los recursos compartidos y, a continuación, selecciona la pestaña Compartidos conmigo.
3. En la tabla Recursos compartidos conmigo, selecciona el recurso compartido del flujo de trabajo y, a continuación, selecciona Aceptar.

Tras aceptar el flujo de trabajo, elija el enlace Recurso del flujo de trabajo compartido para ver sus detalles.

Ejecutar un flujo de trabajo compartido mediante la consola

Tras aceptar un flujo de trabajo compartido, puede iniciar una ejecución en el flujo de trabajo.

1. Abra la consola de HealthOmics .
2. Si es necesario, abre el panel de navegación izquierdo (≡). Selecciona Todos los recursos compartidos y, a continuación, selecciona la pestaña Compartidos conmigo.

3. En la tabla Recursos compartidos conmigo, selecciona el enlace Recurso para el flujo de trabajo compartido.
4. En la página de detalles del flujo de trabajo, seleccione Crear ejecución.

   La consola abre la página Crear ejecución, con el tipo de flujo de trabajo (compartido) y el ID del flujo de trabajo rellenados previamente.
5. Configure los campos restantes en el formulario Crear ejecución. Para obtener información adicional, consulta Iniciar una ejecución mediante la consola.

Ejecutar un flujo de trabajo compartido mediante la API

Utilice get-workflow para recuperar el ARN del flujo de trabajo compartido.

```
aws omics get-workflow --id 1234567 \
--workflow-owner-id 55555555555
```

Cuando ejecute el flujo de trabajo, proporcione el Cuenta de AWS ID del propietario del flujo de trabajo y el ARN del flujo de trabajo compartido.

```
aws omics start-run --id 1234567 --workflow-owner-id 55555555555 \
--role-arn arn:aws:iam::1234567892012:role/service-role/OmicsWorkflow-20221004T164236 \
--name ArchiveTest --retention-mode REMOVE
```

Flujos de trabajo Ready2Run en HealthOmics

Los flujos de trabajo de Ready2Run son flujos de trabajo preconfigurados publicados por editores externos. Algunos editores, como Sentieon Inc., ofrecen flujos de trabajo por suscripción. Otros flujos de trabajo de Ready2Run no requieren suscripción y algunos flujos de trabajo son de código abierto, como los flujos de trabajo NF-Core.

Los flujos de trabajo de Ready2Run se adaptan bien a los siguientes escenarios:

- Desea centrarse en el análisis de los resultados de la canalización y en la generación de resultados, sin necesidad de configurar la infraestructura subyacente.
- Desea replicar sus resultados mediante flujos de trabajo establecidos.
- Como desarrollador de software, desea integrar su aplicación directamente con el HealthOmics SDK.

HealthOmics admite el control de versiones para los flujos de trabajo de Ready2Run. Para un flujo de trabajo de Ready2Run que ofrezca versiones, puede especificar el nombre de la versión al iniciar una ejecución.

Todos los flujos de trabajo de Ready2Run proporcionan registros, incluidos CloudWatch los registros, que puede utilizar para solucionar problemas.

> **Note**
>
> Los flujos de trabajo Ready2Run de Sentieon se basan en suscripciones. Cuando ejecutas un flujo de trabajo Ready2Run de Sentieon por primera vez en una cuenta, Sentieon crea automáticamente una licencia de evaluación de dos semanas para ti. Cuenta de AWS La licencia es válida para todos los flujos de trabajo Ready2Run de Sentieon. Una vez finalizado el período de evaluación, puede solicitar una licencia permanente o solicitar una extensión de la licencia de evaluación. Para obtener más información, consulte Subscribing to Sentieon Ready2Run workflows.

Temas

- Flujos de trabajo Ready2Run disponibles en HealthOmics
- Suscribirse a los flujos de trabajo de Sentieon Ready2Run

- Iniciar los flujos de trabajo de HealthOmics Ready2Run mediante la consola
- Iniciar los flujos de trabajo de HealthOmics Ready2Run mediante la API

Flujos de trabajo Ready2Run disponibles en HealthOmics

En la siguiente tabla se enumeran los flujos de trabajo de Ready2Run que están disponibles en. HealthOmics

Puede iniciar sesión en la HealthOmicsconsola para ver información detallada sobre estos flujos de trabajo, incluidos los parámetros de entrada y los diagramas de flujo de trabajo. Para obtener información sobre los precios de los flujos de trabajo de Ready2Run, consulte HealthOmics Precios.

> **Note**
>
> Cada flujo de trabajo de Ready2Run tiene un tamaño máximo de archivo de entrada. Estos tamaños máximos de archivo no se pueden ajustar.

Nombre del flujo de trabajo	Publicador	¿Es necesaria una suscripción?	Tamaño máximo del archivo de entrada (GiB)	Tiempo de ejecución estimado (HH:MM)
AlphaFold para 601-1200 residuos	Google DeepMind	No	1	11:15
AlphaFold para hasta 600 residuos	Google DeepMind	No	1	7:30
Base2Fastq para 2x150	Element Biosciences	No	1 000	1:45
Base2Fastq para 2x300	Element Biosciences	No	1 000	1:30

Nombre del flujo de trabajo	Publicador	¿Es necesaria una suscripción?	Tamaño máximo del archivo de entrada (GiB)	Tiempo de ejecución estimado (HH:MM)
Base2Fastq para 2x75	Element Biosciences	No	500	0:45
ESMFold para hasta 800 residuos	Meta Research	No	1	0:15
GATK-BP fq2bam	Instituto Broad	No	64	10:10
Línea germinal bam2vcf de GATK-BP para un genoma de 30 veces	Instituto Broad	No	39	2:45
Línea germinal gatK-BP fq2vcf para un genoma de 30x	Instituto Broad	No	64	12:30
GATK-BP Somatic ES bam2vcf	Instituto Broad	No	86	1:30
NVIDIA Parabricks BAM2 FQ2 BAM WGS hasta 30 veces	Corporación NVIDIA	No	80	1:39

Nombre del flujo de trabajo	Publicador	¿Es necesaria una suscripción?	Tamaño máximo del archivo de entrada (GiB)	Tiempo de ejecución estimado (HH:MM)
NVIDIA Parabricks BAM2 FQ2 BAM WGS hasta 50 veces	Corporación NVIDIA	No	120	2:45
NVIDIA Parabricks BAM2 FQ2 BAM WGS hasta 5 veces	Corporación NVIDIA	No	20	0:18
NVIDIA Parabricks FQ2 BAM WGS hasta 30 veces	Corporación NVIDIA	No	71	1:00
NVIDIA Parabricks FQ2 BAM WGS hasta 50 veces	Corporación NVIDIA	No	137	1:45
NVIDIA Parabricks FQ2 BAM WGS hasta 5 veces	Corporación NVIDIA	No	13	0:15
NVIDIA Parabrick s Germline DeepVariant WGS hasta 30 veces	Corporación NVIDIA	No	71	2:00

Nombre del flujo de trabajo	Publicador	¿Es necesaria una suscripción?	Tamaño máximo del archivo de entrada (GiB)	Tiempo de ejecución estimado (HH:MM)
NVIDIA Parabricks Germline DeepVariant WGS hasta 50 veces	Corporación NVIDIA	No	137	3:30
NVIDIA Parabricks Germline DeepVariant WGS hasta 5 veces	Corporación NVIDIA	No	12	0:30
NVIDIA Parabricks Germline HaplotypeCaller WGS hasta 30 veces	Corporación NVIDIA	No	71	1:15
NVIDIA Parabricks Germline HaplotypeCaller WGS hasta 50 veces	Corporación NVIDIA	No	137	2:00

Nombre del flujo de trabajo	Publicador	¿Es necesaria una suscripción?	Tamaño máximo del archivo de entrada (GiB)	Tiempo de ejecución estimado (HH:MM)
NVIDIA Parabrick s Germline HaplotypeCaller WGS hasta 5 veces	Corporación NVIDIA	No	13	0:15
NVIDIA Parabricks Somatic Mutect2 WGS hasta 50 veces	Corporación NVIDIA	No	196	0:45
sc RNAseq con Kallisto BUStools	Núcleo NF	No	119	1:30
sc RNAseq con salmón alevín frito	Núcleo NF	No	119	2:30
sc RNAseq con STARsolo	Núcleo NF	No	119	2:30
Sentieon Germline BAM WES hasta 300 veces	Sentieon, Inc.	Sí	9	1:00
Sentieon Germline BAM WGS hasta 32 veces	Sentieon, Inc.	Sí	18	1:30

Nombre del flujo de trabajo	Publicador	¿Es necesaria una suscripción?	Tamaño máximo del archivo de entrada (GiB)	Tiempo de ejecución estimado (HH:MM)
Sention Germline FASTA WES hasta 100 veces	Sentieon, Inc.	Sí	5	0:45
Sention Germline FASTA WES hasta 300 veces	Sentieon, Inc.	Sí	26	2:00
Sention Termline FASTA WGS hasta 32 veces	Sentieon, Inc.	Sí	51	3:30
Sentición para ONT LongRead	Sentieon, Inc.	Sí	25	1:30
Sentimiento LongRead por PacBio HiFi	Sentieon, Inc.	Sí	58	4:00
Sentieon Somatic WES	Sentieon, Inc.	Sí	50	2:30
Sention Somatic WGS	Sentieon, Inc.	Sí	113	4:30
Ultima Genomics hasta 40 DeepVariant veces	Ultima Genomics	No	91	1:55

Cuando utilizas un flujo de trabajo Ready2Run, el flujo de trabajo está preconfigurado y no se puede editar. A diferencia de los flujos de trabajo privados, los flujos de trabajo de Ready2Run no admiten lo siguiente:

- Aumentar el tamaño máximo del archivo de entrada
- Cambiar los recursos informáticos o ejecutar el almacenamiento
- Cambiar la definición del flujo de trabajo o los contenedores
- Añadir corridas a un grupo de corridas
- Compartir el flujo de trabajo

Si el editor ha compartido el flujo de trabajo de Ready2Run GitHub, usted puede crear su propio flujo de trabajo privado basado en el flujo de trabajo de Ready2Run. En la siguiente tabla se proporcionan enlaces a los GitHub flujos de trabajo de cada editor.

Publicador	Flujos de trabajo en GitHub
Google DeepMind, Meta Research	Flujos de trabajo de plegado
Element Biosciences	Para obtener información, póngase en contacto con Element Biosciences
Instituto Broad	Flujos de trabajo de GATK
Corporación NVIDIA	Flujos de trabajo de Parabricks
nf-core	Flujos de trabajo de NF-Core
Sention	Flujos de trabajo de Sentieon
Ultima Genomics	Flujos de trabajo de Ultima Genomics

Suscribirse a los flujos de trabajo de Sentieon Ready2Run

Los flujos de trabajo Ready2Run de Sentieon se basan en suscripciones. Cuando ejecutas un flujo de trabajo Ready2Run de Sentieon por primera vez en una cuenta, Sentieon crea automáticamente una licencia de evaluación de dos semanas para ti. Cuenta de AWS La licencia es válida para todos

los flujos de trabajo Ready2Run de Sentieon. Una vez finalizado el período de evaluación, puede solicitar una licencia permanente o solicitar una extensión de la licencia de evaluación.

Siga estos pasos para suscribirse a los flujos de trabajo de Sentieon Ready2Run:

- Encuentra tu AWS seudónimo de Canonical siguiendo estas instrucciones.
- Envía un correo electrónico al grupo de soporte de Sentieon (support@sentieon.com) para solicitar una licencia de software. Introduce tu AWS seudónimo de Canonical en el correo electrónico.

Iniciar los flujos de trabajo de HealthOmics Ready2Run mediante la consola

El uso de los flujos de trabajo de Ready2Run en la consola es similar al uso de un flujo de trabajo privado. Una diferencia clave es que el editor del flujo de trabajo proporciona datos de muestra, de modo que puede probar el flujo de trabajo sin tener que crear sus propios datos.

Para usar un flujo de trabajo Ready2Run en la consola

1. Abra la consola de HealthOmics .
2. Si es necesario, abra el panel de navegación izquierdo (≡). Elija los flujos de trabajo Ready2Run.
3. En la página de flujos de trabajo de Ready2Run, elija el flujo de trabajo que desee usar. La consola abre la página de detalles de ese flujo de trabajo.
4. La pestaña de detalles muestra información como el nombre, el precio de lista por ejecución, la descripción, el tipo de idioma del flujo de trabajo, la capacidad de almacenamiento de la ejecución, el estado, la fecha de creación y los parámetros con descripciones. La pestaña de detalles también indica si el flujo de trabajo requiere una suscripción.
5. Para usar el flujo de trabajo, selecciona Crear ejecución
6. En la página Especificar los detalles de la ejecución, introduzca un nombre de ejecución. Si lo desea, puede especificar la versión del flujo de trabajo. También puede añadir prioridad de ejecución a la ejecución.
7. Introduzca o seleccione una ubicación de Amazon S3 para el resultado de la ejecución.
8. En el modo de retención de metadatos de ejecución, elija si desea conservar o eliminar los metadatos de ejecución.
9. En el panel de funciones de servicio, elija si desea utilizar una función de servicio existente o crear una nueva.

10. (Opcional) Añade etiquetas para ayudar a identificar y gestionar tu carrera.
11. Elija Siguiente.
12. En la página Añadir parámetros, elige una de las opciones para añadir los valores de los parámetros de ejecución:

    - Seleccione un archivo de parámetros (en formato JSON) de una ubicación de Amazon S3.
    - Seleccione un archivo de parámetros (en formato JSON) de su unidad local.
    - Introduzca manualmente los valores de los parámetros.
    - Ejecute el flujo de trabajo con los datos de ejemplo de Ready2Run proporcionados por el editor del flujo de trabajo.
13. Si carga un archivo JSON, la consola analiza el archivo y realiza la validación en línea. A continuación, puede actualizar manualmente los valores de sus parámetros según sea necesario.
14. Elija Siguiente.
15. Revisa las entradas y, a continuación, selecciona Iniciar ejecución.

Iniciar los flujos de trabajo de HealthOmics Ready2Run mediante la API

La mayoría de las operaciones de la API se comportan de forma similar para los flujos de trabajo de Ready2Run y los flujos de trabajo privados.

Para obtener una lista de los flujos de trabajo de Ready2Run disponibles, utilice list-workflows con el parámetro establecido en RUN. `type` READY2

```
aws omics list-workflows --type READY2RUN
```

Tras identificar el flujo de trabajo que se va a ejecutar a partir de la respuesta list-workflows, puede usar get-workflow con el parámetro para obtener más detalles. `--id`

```
aws omics get-workflow --type READY2RUN --id workflow id
```

Para ejecutar un flujo de trabajo Ready2Run, puedes usar la operación API de inicio y ejecución con el parámetro de tipo de flujo de trabajo establecido en, como se muestra en el siguiente ejemplo `READY2RUN`

```
aws-omics start-run \
  --workflow-type READY2RUN \
  --workflow-id workflow id \
  --output-uri &example-s3-bucket; \
  --role-arn arn:aws:iam::1234567892012:role/service-role/OmicsWorkflow-20221004T164236
 \
  --parameters file:///path/to/parameters.json
```

Para especificar una versión del flujo de trabajo, utilice el parámetro workflow-version, como se muestra en este ejemplo.

```
aws-omics start-run \
  --workflow-type READY2RUN \
  ...
  --version-name '3.0.0'
```

Para supervisar tu ejecución, puedes usar la operación de API get-run, como se muestra.

```
aws-omics get-run \
  --id run id
```

HealthOmics almacenamiento

Utilice el HealthOmics almacenamiento para almacenar, recuperar, organizar y compartir datos genómicos de manera eficiente y a bajo costo. HealthOmics El almacenamiento comprende las relaciones entre los diferentes objetos de datos, de modo que puede definir qué conjuntos de lectura se originaron a partir de la misma fuente de datos. Esto le proporciona la procedencia de los datos.

Los datos que están almacenados en ese `ACTIVE` estado se pueden recuperar de forma inmediata. Los datos a los que no se ha accedido durante 30 días o más se almacenan en el `ARCHIVE` estado. Para acceder a los datos archivados, puedes reactivarlos mediante las operaciones de la API o la consola.

HealthOmics Los almacenes de secuencias están diseñados para preservar la integridad del contenido de los archivos. Sin embargo, la equivalencia bit a bit de los archivos de datos importados y los archivos exportados no se conserva debido a la compresión durante la estratificación activa y del archivo.

Durante la ingesta, HealthOmics genera una etiqueta de entidad o HealthOmics ETag, para permitir validar la integridad del contenido de los archivos de datos. Las partes de secuenciación se identifican y capturan ETag en el nivel de origen de un conjunto de lectura. El ETag cálculo no altera el archivo real ni los datos genómicos. Una vez creado un conjunto de lecturas, no ETag debería cambiar a lo largo del ciclo de vida de la fuente del conjunto de lecturas. Esto significa que si se vuelve a importar el mismo archivo, se calcula el mismo ETag valor.

Temas

- HealthOmics ETags y procedencia de los datos
- Crear una tienda HealthOmics de referencia
- Crear un almacén HealthOmics de secuencias
- Eliminar almacenes HealthOmics de referencias y secuencias
- Importación de conjuntos de lectura a un almacén HealthOmics de secuencias
- Carga directa a un almacén HealthOmics de secuencias
- Exportación de conjuntos de HealthOmics lectura a un bucket de Amazon S3
- Acceso a conjuntos de HealthOmics lectura con Amazon S3 URIs
- Activar conjuntos de lectura en HealthOmics

HealthOmics ETags y procedencia de los datos

Una HealthOmics ETag (etiqueta de entidad) es un hash del contenido ingerido en un almacén de secuencias. Esto simplifica la recuperación y el procesamiento de los datos y, al mismo tiempo, mantiene la integridad del contenido de los archivos de datos ingeridos. ETag Refleja los cambios en el contenido semántico del objeto, no en sus metadatos. El tipo de conjunto de lectura y el algoritmo especificados determinan cómo ETag se calcula. El ETag cálculo no altera el archivo ni los datos genómicos reales. Cuando el esquema de tipos de archivo del conjunto de lectura lo permite, el almacén de secuencias actualiza los campos que están vinculados a la procedencia de los datos.

Los archivos tienen una identidad bit a bit y una identidad semántica. La identidad bit a bit significa que los bits de un archivo son idénticos, y una identidad semántica significa que el contenido de un archivo es idéntico. La identidad semántica es resistente a los cambios en los metadatos y a los cambios de compresión, ya que captura la integridad del contenido del archivo.

Los conjuntos de lectura de los almacenes de HealthOmics secuencias se someten a compression/decompression ciclos y se realiza un seguimiento de la procedencia de los datos a lo largo del ciclo de vida de un objeto. Durante este procesamiento, la identidad bit a bit de un archivo ingerido puede cambiar y se espera que cambie cada vez que se activa un archivo; sin embargo, se mantiene la identidad semántica del archivo. La identidad semántica se captura como una etiqueta de HealthOmics entidad o ETag se calcula durante la ingesta del almacén de secuencias y está disponible como metadatos de conjuntos de lectura.

Cuando el esquema de tipos de archivos del conjunto de lectura lo permite, los campos de actualizaciones del almacén de secuencias se vinculan a la procedencia de los datos. En el caso de los archivos uBAM, BAM y CRAM, se añade una nueva etiqueta `@CO` or `Comment` al encabezado. El comentario contiene el ID del almacén de secuencias y la marca de tiempo de ingesta.

Amazon S3 ETags

Al acceder a un archivo mediante el URI de Amazon S3, las operaciones de la API de Amazon S3 también pueden devolver valores de Amazon S3 ETag y checksum. Los valores de Amazon S3 ETag y checksum difieren de los de HealthOmics ETags porque representan la identidad bit a bit del archivo. Para obtener más información sobre los objetos y los metadatos descriptivos, consulte la documentación de la API de objetos de Amazon S3. ETag Los valores de Amazon S3 pueden cambiar con cada ciclo de activación de un conjunto de lecturas y puede usarlos para validar la lectura de un archivo. Sin embargo, no almacene en caché ETag los valores de Amazon S3 para utilizarlos en la validación de la identidad del archivo durante el ciclo de vida del archivo, ya que no

son coherentes. Por el contrario, se HealthOmics ETag mantiene constante durante todo el ciclo de vida del conjunto de lectura.

¿Cómo se HealthOmics calcula ETags

ETag Se genera a partir de un hash del contenido del archivo ingerido. La familia de ETag algoritmos está configurada como de forma MD5up predeterminada, pero se puede configurar de forma diferente durante la creación del almacén de secuencias. Cuando ETag se calcula, el algoritmo y los hashes calculados se añaden al conjunto de lecturas. Los MD5 algoritmos admitidos para los tipos de archivos son los siguientes.

- FASTQ_ MD5up: calcula el MD5 hash de una fuente de conjunto de lectura FASTQ completa y sin comprimir.
- BAM_ MD5up — Calcula el MD5 hash de la sección de alineación de una fuente de conjunto de lectura BAM o uBAM sin comprimir tal como se representa en el SAM, en función de la referencia vinculada, si hay alguna disponible.
- CRAM_ MD5up: calcula el MD5 hash de la sección de alineación de la fuente del conjunto de lectura CRAM sin comprimir tal como se representa en el SAM, en función de la referencia vinculada.

> **Note**
>
> MD5 Se sabe que el hash es vulnerable a las colisiones. Por este motivo, dos archivos diferentes podrían tener lo mismo ETag si se hubieran fabricado para aprovechar la colisión conocida.

La SHA256 familia admite los siguientes algoritmos. Los algoritmos se calculan de la siguiente manera:

- FASTQ_ SHA256up: calcula el hash SHA-256 de una fuente de conjunto de lectura FASTQ completa y sin comprimir.
- BAM_ SHA256up: calcula el hash SHA-256 de la sección de alineación de una fuente de conjunto de lectura BAM o uBAM sin comprimir tal como se representa en el SAM, en función de la referencia vinculada, si hay alguna disponible.

- CRAM_ SHA256up: calcula el hash SHA-256 de la sección de alineación de una fuente de conjunto de lectura CRAM sin comprimir tal como se representa en el SAM, en función de la referencia vinculada.

La familia admite los siguientes algoritmos. SHA512 Los algoritmos se calculan de la siguiente manera:

- FASTQ_ SHA512up: calcula el hash SHA-512 de una fuente de conjunto de lectura FASTQ completa y sin comprimir.
- BAM_ SHA512up: calcula el hash SHA-512 de la sección de alineación de una fuente de conjunto de lectura BAM o uBAM sin comprimir tal como se representa en el SAM, basándose en la referencia vinculada, si hay alguna disponible.

- CRAM_ SHA512up : calcula el hash SHA-512 de la sección de alineación de una fuente de conjunto de lectura CRAM sin comprimir tal como se representa en el SAM, en función de la referencia vinculada.

Crear una tienda HealthOmics de referencia

Un almacén de referencia HealthOmics es un almacén de datos para el almacenamiento de genomas de referencia. Puede tener un único almacén de referencias en cada Cuenta de AWS región. Puede crear un almacén de referencias mediante la consola o la CLI.

Temas

- Crear un almacén de referencias mediante la consola
- Creación de un almacén de referencias mediante la CLI

Crear un almacén de referencias mediante la consola

Para crear un almacén de referencia

1. Abra la consola de HealthOmics .
2. Si es necesario, abra el panel de navegación izquierdo (≡). Seleccione Tienda de referencias.
3. Elija genomas de referencia en las opciones de almacenamiento de datos genómicos.

4. Puede elegir un genoma de referencia previamente importado o importar uno nuevo. Si no has importado un genoma de referencia, selecciona Importar genoma de referencia en la parte superior derecha.
5. En la página Crear un trabajo de importación de genomas de referencia, elija la opción de creación rápida o de creación manual para crear un almacén de referencias y, a continuación, proporcione la siguiente información.

   - Nombre del genoma de referencia: un nombre exclusivo para este almacén.
   - Descripción (opcional): descripción de este almacén de referencia.
   - Función de IAM: seleccione una función con acceso a su genoma de referencia.
   - Referencia de Amazon S3: seleccione el archivo de secuencia de referencia en un bucket de Amazon S3.
   - Etiquetas (opcional): proporciona hasta 50 etiquetas para esta tienda de referencia.

Creación de un almacén de referencias mediante la CLI

El siguiente ejemplo muestra cómo crear un almacén de referencias mediante AWS CLI. Puede tener una tienda de referencia por AWS región.

Los almacenes de referencia admiten el almacenamiento de archivos FASTA con las extensiones `.fasta.fa`,`.fas`,`.fsa`,`.faa`,`.fna`,`.ffn`,,`.frn`, `.mpfa.seq`,`.txt`. También se admite la `bgzip` versión de estas extensiones.

En el siguiente ejemplo, *`reference store name`* sustitúyalo por el nombre que haya elegido para su tienda de referencia.

```
aws omics create-reference-store --name "reference store name"
```

Recibirá una respuesta JSON con el ID y el nombre del almacén de referencia, el ARN y la marca de tiempo de cuando se creó el almacén de referencia.

```
{
    "id": "3242349265",
    "arn": "arn:aws:omics:us-west-2:555555555555:referenceStore/3242349265",
    "name": "MyReferenceStore",
    "creationTime": "2022-07-01T20:58:42.878Z"
}
```

Puedes usar el ID del almacén de referencia en comandos adicionales. AWS CLI Puede recuperar la lista de almacenes de referencia IDs vinculada a su cuenta mediante el list-reference-storescomando, como se muestra en el siguiente ejemplo.

```
aws omics list-reference-stores
```

Como respuesta, recibirá el nombre del almacén de referencia que acaba de crear.

```
{
    "referenceStores": [
        {
              "id": "3242349265",
              "arn": "arn:aws:omics:us-west-2:555555555555:referenceStore/3242349265",
              "name": "MyReferenceStore",
             "creationTime": "2022-07-01T20:58:42.878Z"
         }
     ]
}
```

Tras crear un almacén de referencias, puede crear trabajos de importación para cargar en él archivos de referencia genómica. Para ello, debe usar o crear un rol de IAM para acceder a los datos. A continuación, se muestra una política de ejemplo.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:GetBucketLocation"

            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket1",
                "arn:aws:s3:::amzn-s3-demo-bucket1/*"
            ]
        }
    ]
```

```
}
```

También debe tener una política de confianza similar a la del siguiente ejemplo.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": [
                   "omics.amazonaws.com"
                ]
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

Ahora puede importar un genoma de referencia. En este ejemplo, se utiliza el Genome Reference Consortium Human Build 38 (hg38), que es de acceso abierto y está disponible en AWS el Registro de Datos Abiertos. El depósito que aloja estos datos se encuentra en el este de EE. UU. (Ohio). Para usar depósitos en otras AWS regiones, puede copiar los datos a un depósito de Amazon S3 alojado en su región. Utilice el siguiente AWS CLI comando para copiar el genoma a su bucket de Amazon S3.

```
aws s3 cp s3://broad-references/hg38/v0/Homo_sapiens_assembly38.fasta s3://amzn-s3-
demo-bucket
```

A continuación, puede comenzar su trabajo de importación. Reemplace *reference store IDrole ARN*, y *source file path* con su propia entrada.

```
aws omics start-reference-import-job --reference-store-id reference store ID --role-
arn role ARN --sources source file path
```

Una vez importados los datos, recibirá la siguiente respuesta en JSON.

```
{
        "id": "7252016478",
        "referenceStoreId": "3242349265",
        "roleArn": "arn:aws:iam::111122223333:role/OmicsReferenceImport",
        "status": "CREATED",
        "creationTime": "2022-07-01T21:15:13.727Z"
}
```

Puede supervisar el estado de un trabajo mediante el siguiente comando. En el siguiente ejemplo, sustituya *reference store ID* y *job ID* por su ID de almacén de referencia y el ID del trabajo sobre los que desee obtener más información.

```
aws omics get-reference-import-job --reference-store-id reference store ID --id job ID
```

Como respuesta, recibirá una respuesta con los detalles de ese almacén de referencia y su estado.

```
{
    "id": "7252016478",
    "referenceStoreId": "3242349265",
    "roleArn": "arn:aws:iam::555555555555:role/OmicsReferenceImport",
    "status": "RUNNING",
    "creationTime": "2022-07-01T21:15:13.727Z",
    "sources": [
        {
            "sourceFile": "s3://amzn-s3-demo-bucket/Homo_sapiens_assembly38.fasta",
            "status": "IN_PROGRESS",
            "name": "MyReference"
        }
    ]
}
```

También puede encontrar la referencia que se importó enumerando sus referencias y filtrándolas según el nombre de la referencia. *reference store ID*Sustitúyala por el ID de tu tienda de referencia y añade un filtro opcional para reducir la lista.

```
aws omics list-references --reference-store-id reference store ID --filter
 name=MyReference
```

En respuesta, recibirá la siguiente información.

```
{
    "references": [
        {
            "id": "1234567890",
            "arn": "arn:aws:omics:us-west-2:555555555555:referenceStore/1234567890/
reference/1234567890",
            "referenceStoreId": "12345678",
            "md5": "7ff134953dcca8c8997453bbb80b6b5e",
            "status": "ACTIVE",
            "name": "MyReference",
            "creationTime": "2022-07-02T00:15:19.787Z",
            "updateTime": "2022-07-02T00:15:19.787Z"
        }
    ]
}
```

Para obtener más información sobre los metadatos de referencia, usa la operación de la get-reference-metadataAPI. En el siguiente ejemplo, *reference store ID* sustitúyalo por el ID de tu tienda de referencia y *reference ID* por el ID de referencia sobre el que deseas obtener más información.

```
aws omics get-reference-metadata --reference-store-id reference store ID --id reference
 ID
```

En respuesta, recibirás la siguiente información.

```
{
    "id": "1234567890",
    "arn": "arn:aws:omics:us-west-2:555555555555:referenceStore/referencestoreID/
reference/referenceID",
    "referenceStoreId": "1234567890",
    "md5": "7ff134953dcca8c8997453bbb80b6b5e",
    "status": "ACTIVE",
    "name": "MyReference",
    "creationTime": "2022-07-02T00:15:19.787Z",
    "updateTime": "2022-07-02T00:15:19.787Z",
    "files": {
        "source": {
            "totalParts": 31,
            "partSize": 104857600,
            "contentLength": 3249912778
        },
```

```
        "index": {
            "totalParts": 1,
            "partSize": 104857600,
            "contentLength": 160928
        }
    }
}
```

También puede descargar partes del archivo de referencia mediante get-reference. En el siguiente ejemplo, *reference store ID* sustitúyalo por el identificador de tu tienda de referencia y *reference ID* por el identificador de referencia desde el que deseas realizar la descarga.

```
aws omics get-reference --reference-store-id reference store ID --id reference ID --
part-number 1 outfile.fa
```

Crear un almacén HealthOmics de secuencias

HealthOmics Los almacenes de secuencias admiten el almacenamiento de archivos genómicos en los formatos no alineados de FASTQ (solo gzip) y. uBAM También es compatible con los formatos alineados de y. BAM CRAM

Los archivos importados se almacenan como conjuntos de lectura. Puede añadir etiquetas a los conjuntos de lectura y utilizar las políticas de IAM para controlar el acceso a los conjuntos de lectura. Los conjuntos de lectura alineados requieren un genoma de referencia para alinear las secuencias genómicas, pero es opcional para los conjuntos de lectura no alineados.

Para almacenar conjuntos de lecturas, primero debe crear un almacén de secuencias. Al crear un almacén de secuencias, puede especificar un bucket de Amazon S3 opcional como ubicación alternativa y la ubicación en la que se almacenan los registros de acceso a S3. La ubicación alternativa se utiliza para almacenar cualquier archivo que no pueda crear un conjunto de lecturas durante una carga directa. Las ubicaciones alternativas están disponibles para los almacenes de secuencias creados después del 15 de mayo de 2023. La ubicación de reserva se especifica al crear el almacén de secuencias.

Puede especificar hasta cinco claves de etiquetas de conjunto de lectura. Al crear o actualizar un conjunto de lecturas con una clave de etiqueta que coincide con una de estas claves, las etiquetas del conjunto de lecturas se propagan al objeto de Amazon S3 correspondiente. Las etiquetas del sistema creadas por HealthOmics se propagan de forma predeterminada.

Temas

- Crear un almacén de secuencias mediante la consola
- Creación de un almacén de secuencias mediante la CLI
- Actualización de un almacén de secuencias
- Actualización de las etiquetas de los conjuntos de lectura de un almacén de secuencias
- Importación de archivos genómicos

Crear un almacén de secuencias mediante la consola

Para crear un almacén de secuencias

1. Abra la consola de HealthOmics .
2. Si es necesario, abra el panel de navegación izquierdo (≡). Elija Sequence stores.
3. En la página Crear almacén de secuencias, proporcione la siguiente información

   - Nombre del almacén de secuencias: nombre exclusivo para este almacén.
   - Descripción (opcional): descripción de este almacén de secuencias.
4. Para la ubicación de respaldo en S3, especifique una ubicación de Amazon S3. HealthOmics utiliza la ubicación alternativa para almacenar los archivos que no puedan crear un conjunto de lecturas durante una carga directa. Debe conceder al HealthOmics servicio acceso de escritura a la ubicación alternativa de Amazon S3. Para ver una política de ejemplo, consulte Configure una ubicación alternativa.

   Las ubicaciones alternativas no están disponibles para los almacenes secuenciales creados antes del 16 de mayo de 2023.
5. (Opcional) En el caso de las claves de etiquetas de conjuntos de lectura para la propagación de S3, puede introducir hasta cinco claves de conjuntos de lectura para propagarlas desde un conjunto de lecturas a los objetos S3 subyacentes. Al propagar etiquetas de un conjunto de lecturas al objeto S3, puede conceder permisos de acceso a S3 en función de las etiquetas a los usuarios and/or finales para ver las etiquetas propagadas a través de la operación de la getObjectTagging API de Amazon S3.

   a. Introduzca un valor clave en el cuadro de texto. La consola crea un nuevo cuadro de texto para añadir la siguiente clave.

   b. (Opcional) Seleccione Eliminar para eliminar todas las claves.

6. En Cifrado de datos, seleccione si desea que el cifrado de datos sea propiedad y esté gestionado por una CMK gestionada por el cliente AWS o si desea que se utilice una CMK gestionada por el cliente.
7. (Opcional) En Acceso a datos de S3, seleccione si desea crear una nueva función y política para acceder al almacén de secuencias a través de Amazon S3.
8. (Opcional) Para el registro de acceso de S3, seleccione `Enabled` si desea que Amazon S3 recopile los registros de acceso.

   En Ubicación de registro de acceso en S3, especifique una ubicación de Amazon S3 para almacenar los registros. Este campo solo está visible si ha activado el registro de acceso a S3.
9. Etiquetas (opcional): proporciona hasta 50 etiquetas para este almacén de secuencias. Estas etiquetas son independientes de las etiquetas del conjunto de lectura que se configuran durante la import/tag actualización del conjunto de lectura

Después de crear la tienda, estará lista paraImportación de archivos genómicos.

Creación de un almacén de secuencias mediante la CLI

En el siguiente ejemplo, *`sequence store name`* sustitúyalo por el nombre que eligió para el almacén de secuencias.

```
aws omics create-sequence-store --name sequence store name --fallback-location "s3://
amzn-s3-demo-bucket"
```

Recibirás la siguiente respuesta en JSON, que incluye el número de ID del almacén de secuencias recién creado.

```
{
    "id": "3936421177",
    "arn": "arn:aws:omics:us-west-2:111122223333:sequenceStore/3936421177",
    "name": "sequence_store_example_name",
    "creationTime": "2022-07-13T20:09:26.038Z"
    "fallbackLocation" : "s3://amzn-s3-demo-bucket"
}
```

También puedes ver todos los almacenes de secuencias asociados a tu cuenta mediante el list-sequence-storescomando, como se muestra a continuación.

```
aws omics list-sequence-stores
```

Recibirás la siguiente respuesta.

```
{
    "sequenceStores": [
        {
            "arn": "arn:aws:omics:us-west-2:111122223333:sequenceStore/3936421177",
            "id": "3936421177",
            "name": "MySequenceStore",
            "creationTime": "2022-07-13T20:09:26.038Z",
            "updatedTime": "2024-09-13T04:11:31.242Z",
            "fallbackLocation" : "s3://amzn-s3-demo-bucket",
            "status": "Active"
        }
    ]
}
```

Puede utilizarla get-sequence-storepara obtener más información sobre un almacén de secuencias utilizando su ID, como se muestra en el siguiente ejemplo:

```
aws omics get-sequence-store --id sequence store ID
```

Recibirás la siguiente respuesta:

```
{
  "arn": "arn:aws:omics:us-west-2:123456789012:sequenceStore/sequencestoreID",
  "creationTime": "2024-01-12T04:45:29.857Z",
  "updatedTime": "2024-09-13T04:11:31.242Z",
  "description": null,
  "fallbackLocation": null,
  "id": "2015356892",
  "name": "MySequenceStore",
  "s3Access": {
      "s3AccessPointArn": "arn:aws:s3:us-
west-2:123456789012:accesspoint/592761533288-2015356892",
      "s3Uri": "s3://592761533288-2015356892-ajdpi90jdas90a79fh9a8ja98jdfa9jf98-
s3alias/592761533288/sequenceStore/2015356892/",
      "accessLogLocation": "s3://IAD-seq-store-log/2015356892/"
  },
  "sseConfig": {
```

```
      "keyArn": "arn:aws:kms:us-west-2:123456789012:key/eb2b30f5-635d-4b6d-b0f9-
d3889fe0e648",
      "type": "KMS"
  },
  "status": "Active",
  "statusMessage": null,
  "setTagsToSync": ["withdrawn","protocol"],
}
```


Tras la creación, también se pueden actualizar varios parámetros de la tienda. Esto se puede hacer a través de la consola o la `updateSequenceStore` operación de la API.

Actualización de un almacén de secuencias

Para actualizar un almacén de secuencias, sigue estos pasos:

1. Abra la consola de HealthOmics .
2. Si es necesario, abre el panel de navegación izquierdo (≡). Elija Sequence stores.
3. Elija el almacén de secuencias que desee actualizar.
4. En el panel de detalles, elija Editar.
5. En la página Editar detalles, puede actualizar los siguientes campos:

   - Nombre de almacén secuencial: nombre exclusivo para este almacén.
   - Descripción: descripción de este almacén de secuencias.
   - Ubicación alternativa en S3, especifique una ubicación de Amazon S3. HealthOmics utiliza la ubicación alternativa para almacenar los archivos que no puedan crear un conjunto de lectura durante una carga directa.
   - Lea las claves de conjunto de etiquetas para la propagación de S3; puede introducir hasta cinco claves de conjunto de lectura para propagarlas a Amazon S3.
   - (Opcional) Para el registro de acceso de S3, seleccione `Enabled` si desea que Amazon S3 recopile los registros de acceso.

     En Ubicación de registro de acceso en S3, especifique una ubicación de Amazon S3 para almacenar los registros. Este campo solo está visible si ha activado el registro de acceso a S3.
   - Etiquetas (opcional): proporciona hasta 50 etiquetas para este almacén de secuencias.

Actualización de las etiquetas de los conjuntos de lectura de un almacén de secuencias

Para actualizar las etiquetas del conjunto de lecturas u otros campos de un almacén de secuencias, sigue estos pasos:

1. Abra la consola de HealthOmics .
2. Si es necesario, abre el panel de navegación izquierdo (≡). Elija Sequence stores.
3. Elija el almacén de secuencias que desee actualizar.
4. Elija la pestaña Detalles.
5. Seleccione Editar.
6. Añada nuevas etiquetas de conjunto de lectura o elimine las etiquetas existentes, según sea necesario.
7. Actualice el nombre, la descripción, la ubicación alternativa o el acceso a los datos de S3, según sea necesario.
8. Seleccione Save changes (Guardar cambios).

Importación de archivos genómicos

Para importar archivos genómicos a un almacén de secuencias, siga estos pasos:

Para importar un archivo de genómica

1. Abra la consola de HealthOmics .
2. Si es necesario, abra el panel de navegación izquierdo (≡). Elija elegir tiendas de secuencias.
3. En la página de almacenes de secuencias, elija el almacén de secuencias al que desee importar los archivos.
4. En la página del almacén de secuencias individual, seleccione Importar archivos genómicos.
5. En la página Especificar los detalles de la importación, proporcione la siguiente información

   - Función de IAM: la función de IAM que puede acceder a los archivos genómicos de Amazon S3.
   - Genoma de referencia: el genoma de referencia para estos datos genómicos.
6. En la página Especificar el manifiesto de importación, especifique la siguiente información: archivo de manifiesto. El archivo de manifiesto es un archivo JSON o YAML que describe

la información esencial de sus datos genómicos. Para obtener información sobre el archivo de manifiesto, consulte. Importación de conjuntos de lectura a un almacén HealthOmics de secuencias

7. Haga clic en Crear trabajo de importación.

Eliminar almacenes HealthOmics de referencias y secuencias

Se pueden eliminar tanto los almacenes de referencia como los de secuencias. Los almacenes de secuencias solo se pueden eliminar si no contienen conjuntos de lectura, y los almacenes de referencia solo se pueden eliminar si no contienen referencias. Al eliminar un almacén de secuencias o referencias, también se eliminan todas las etiquetas asociadas a ese almacén.

En el siguiente ejemplo, se muestra cómo eliminar un almacén de referencias mediante. AWS CLI Si la acción se realiza correctamente, no recibirá ninguna respuesta. En el siguiente ejemplo, *reference store ID* sustitúyalo por el identificador de tu tienda de referencia.

```
aws omics delete-reference-store --id reference store ID
```

En el siguiente ejemplo, se muestra cómo eliminar un almacén de secuencias. No recibirá ninguna respuesta si la acción se realiza correctamente. En el siguiente ejemplo, sustitúyalo *sequence store ID* por tu ID de almacén de secuencias.

```
aws omics delete-sequence-store --id sequence store ID
```

También puedes eliminar una referencia de un almacén de referencias, como se muestra en el siguiente ejemplo. Las referencias solo se pueden eliminar si no se utilizan en un conjunto de lectura, un almacén de variantes o un almacén de anotaciones. En el siguiente ejemplo, *reference store ID* sustitúyelo por el ID de su almacén de referencias y *reference ID* sustitúyalo por el ID de la referencia que desee eliminar.

```
aws omics delete-reference  --id reference ID --reference-store-id reference store ID
```

Importación de conjuntos de lectura a un almacén HealthOmics de secuencias

Después de crear el almacén de secuencias, cree trabajos de importación para cargar los conjuntos de lecturas en el almacén de datos. Puede cargar sus archivos desde un bucket de Amazon S3 o puede cargarlos directamente mediante las operaciones sincrónicas de la API. Su bucket de Amazon S3 debe estar en la misma región que su almacén de secuencias.

Puede cargar cualquier combinación de conjuntos de lecturas alineados y no alineados en su almacén de secuencias; sin embargo, si alguno de los conjuntos de lectura de la importación está alineado, debe incluir un genoma de referencia.

Puede reutilizar la política de acceso de IAM que utilizó para crear el almacén de referencias.

En los temas siguientes se describen los pasos principales que debe seguir para importar un conjunto de lectura al almacén de secuencias y, a continuación, obtener información sobre los datos importados.

Temas

- Cargar archivos a Amazon S3
- Creación de un archivo de manifiesto
- Iniciar el trabajo de importación
- Supervise el trabajo de importación
- Busque los archivos de secuencias importados
- Obtén detalles sobre un conjunto de lecturas
- Descargue los archivos de datos del conjunto de lectura

Cargar archivos a Amazon S3

El siguiente ejemplo muestra cómo mover archivos a su bucket de Amazon S3.

```
aws s3 cp s3://1000genomes/phase1/data/HG00100/alignment/
HG00100.chrom20.ILLUMINA.bwa.GBR.low_coverage.20101123.bam s3://your-bucket
aws s3 cp s3://1000genomes/phase3/data/HG00146/sequence_read/SRR233106_1.filt.fastq.gz
 s3://your-bucket
aws s3 cp s3://1000genomes/phase3/data/HG00146/sequence_read/SRR233106_2.filt.fastq.gz
 s3://your-bucket
```

```
aws s3 cp s3://1000genomes/data/HG00096/alignment/
HG00096.alt_bwamem_GRCh38DH.20150718.GBR.low_coverage.cram s3://your-bucket
aws s3 cp s3://gatk-test-data/wgs_ubam/NA12878_20k/NA12878_A.bam s3://your-bucket
```

La muestra BAM y la CRAM utilizada en este ejemplo requieren diferentes referencias genómicas, Hg19 yHg38. Para obtener más información o acceder a estas referencias, consulte The Broad Genome References in the Registry of Open Data en AWS.

Creación de un archivo de manifiesto

También debe crear un archivo de manifiesto en JSON para modelar el trabajo de importación `import.json` (consulte el siguiente ejemplo). Si creas un almacén de secuencias en la consola, no tienes que especificar la `sequenceStoreId` `oroleARN`, por lo que el archivo de manifiesto comienza con la `sources` entrada.

API manifest

En el siguiente ejemplo, se importan tres conjuntos de lecturas mediante la API: uno FASTQBAM, uno y otroCRAM.

```
{
  "sequenceStoreId": "3936421177",
  "roleArn": "arn:aws:iam::555555555555:role/OmicsImport",
  "sources":
  [
      {
          "sourceFiles":
          {
              "source1": "s3://amzn-s3-demo-bucket/
HG00100.chrom20.ILLUMINA.bwa.GBR.low_coverage.20101123.bam"
          },
          "sourceFileType": "BAM",
          "subjectId": "mySubject",
          "sampleId": "mySample",
          "referenceArn": "arn:aws:omics:us-
west-2:555555555555:referenceStore/0123456789/reference/0000000001",
          "name": "HG00100",
          "description": "BAM for HG00100",
          "generatedFrom": "1000 Genomes"
      },
      {
```

```
          "sourceFiles":
          {
              "source1": "s3://amzn-s3-demo-bucket/SRR233106_1.filt.fastq.gz",
              "source2": "s3://amzn-s3-demo-bucket/SRR233106_2.filt.fastq.gz"
          },
          "sourceFileType": "FASTQ",
          "subjectId": "mySubject",
          "sampleId": "mySample",
          // NOTE: there is no reference arn required here
          "name": "HG00146",
          "description": "FASTQ for HG00146",
          "generatedFrom": "1000 Genomes"
      },
      {
          "sourceFiles":
          {
              "source1": "s3://amzn-s3-demo-bucket/
HG00096.alt_bwamem_GRCh38DH.20150718.GBR.low_coverage.cram"
          },
          "sourceFileType": "CRAM",
          "subjectId": "mySubject",
          "sampleId": "mySample",
          "referenceArn": "arn:aws:omics:us-
west-2:555555555555:referenceStore/0123456789/reference/0000000001",
          "name": "HG00096",
          "description": "CRAM for HG00096",
          "generatedFrom": "1000 Genomes"
      },
      {
          "sourceFiles":
          {
              "source1": "s3://amzn-s3-demo-bucket/NA12878_A.bam"
          },
          "sourceFileType": "UBAM",
          "subjectId": "mySubject",
          "sampleId": "mySample",
          // NOTE: there is no reference arn required here
          "name": "NA12878_A",
          "description": "uBAM for NA12878",
          "generatedFrom": "GATK Test Data"
      }
  ]
}
```

Console manifest

Este código de ejemplo se utiliza para importar un único conjunto de lecturas mediante la consola.

```
[
  {
      "sourceFiles":
      {
          "source1": "s3://amzn-s3-demo-bucket/
HG00100.chrom20.ILLUMINA.bwa.GBR.low_coverage.20101123.bam"
      },
      "sourceFileType": "BAM",
      "subjectId": "mySubject",
      "sampleId": "mySample",
      "name": "HG00100",
      "description": "BAM for HG00100",
      "generatedFrom": "1000 Genomes"
  },
  {
      "sourceFiles":
      {
          "source1": "s3://amzn-s3-demo-bucket/SRR233106_1.filt.fastq.gz",
          "source2": "s3://amzn-s3-demo-bucket/SRR233106_2.filt.fastq.gz"
      },
      "sourceFileType": "FASTQ",
      "subjectId": "mySubject",
      "sampleId": "mySample",
      "name": "HG00146",
      "description": "FASTQ for HG00146",
      "generatedFrom": "1000 Genomes"
  },
  {
      "sourceFiles":
      {
          "source1": "s3://your-bucket/
HG00096.alt_bwamem_GRCh38DH.20150718.GBR.low_coverage.cram"
      },
      "sourceFileType": "CRAM",
      "subjectId": "mySubject",
      "sampleId": "mySample",
      "name": "HG00096",
      "description": "CRAM for HG00096",
      "generatedFrom": "1000 Genomes"
  },
```

```
  {
      "sourceFiles":
      {
          "source1": "s3://amzn-s3-demo-bucket/NA12878_A.bam"
      },
      "sourceFileType": "UBAM",
      "subjectId": "mySubject",
      "sampleId": "mySample",
      "name": "NA12878_A",
      "description": "uBAM for NA12878",
      "generatedFrom": "GATK Test Data"
  }
]
```

Como alternativa, puedes cargar el archivo de manifiesto en formato YAML.

Iniciar el trabajo de importación

Para iniciar el trabajo de importación, utilice el siguiente AWS CLI comando.

```
aws omics start-read-set-import-job --cli-input-json file://import.json
```

Recibirá la siguiente respuesta, que indica que la creación del trabajo se ha realizado correctamente.

```
{
  "id": "3660451514",
  "sequenceStoreId": "3936421177",
  "roleArn": "arn:aws:iam::111122223333:role/OmicsImport",
  "status": "CREATED",
  "creationTime": "2022-07-13T22:14:59.309Z"
}
```

Supervise el trabajo de importación

Una vez iniciado el trabajo de importación, puede supervisar su progreso con el siguiente comando. En el siguiente ejemplo, *sequence store id* sustitúyalo por el identificador del almacén de secuencias y *job import ID* sustitúyelo por el identificador de importación.

```
aws omics get-read-set-import-job --sequence-store-id sequence store id --id job import
 ID
```

A continuación se muestran los estados de todos los trabajos de importación asociados al ID de almacén de secuencias especificado.


```
{
  "id": "1234567890",
  "sequenceStoreId": "1234567890",
  "roleArn": "arn:aws:iam::111122223333:role/OmicsImport",
  "status": "RUNNING",
  "statusMessage": "The job is currently in progress.",
  "creationTime": "2022-07-13T22:14:59.309Z",
  "sources": [
      {
          "sourceFiles":
          {
              "source1": "s3://amzn-s3-demo-bucket/
HG00100.chrom20.ILLUMINA.bwa.GBR.low_coverage.20101123.bam"
          },
          "sourceFileType": "BAM",
          "status": "IN_PROGRESS",
          "statusMessage": "The job is currently in progress."
          "subjectId": "mySubject",
          "sampleId": "mySample",
          "referenceArn": "arn:aws:omics:us-
west-2:111122223333:referenceStore/3242349265/reference/8625408453",
          "name": "HG00100",
          "description": "BAM for HG00100",
          "generatedFrom": "1000 Genomes",
          "readSetID": "1234567890"
      },
      {
          "sourceFiles":
          {
              "source1": "s3://amzn-s3-demo-bucket/SRR233106_1.filt.fastq.gz",
              "source2": "s3://amzn-s3-demo-bucket/SRR233106_2.filt.fastq.gz"
          },
          "sourceFileType": "FASTQ",
          "status": "IN_PROGRESS",
          "statusMessage": "The job is currently in progress."
          "subjectId": "mySubject",
          "sampleId": "mySample",
          "name": "HG00146",
          "description": "FASTQ for HG00146",
          "generatedFrom": "1000 Genomes",
```

```
            "readSetID": "1234567890"
        },
        {
            "sourceFiles":
            {
                "source1": "s3://amzn-s3-demo-bucket/
HG00096.alt_bwamem_GRCh38DH.20150718.GBR.low_coverage.cram"
            },
            "sourceFileType": "CRAM",
            "status": "IN_PROGRESS",
            "statusMessage": "The job is currently in progress."
            "subjectId": "mySubject",
            "sampleId": "mySample",
            "referenceArn": "arn:aws:omics:us-
west-2:111122223333:referenceStore/3242349265/reference/1234568870",
            "name": "HG00096",
            "description": "CRAM for HG00096",
            "generatedFrom": "1000 Genomes",
            "readSetID": "1234567890"
        },
        {
            "sourceFiles":
            {
                "source1": "s3://amzn-s3-demo-bucket/NA12878_A.bam"
            },
            "sourceFileType": "UBAM",
            "status": "IN_PROGRESS",
            "statusMessage": "The job is currently in progress."
            "subjectId": "mySubject",
            "sampleId": "mySample",
            "name": "NA12878_A",
            "description": "uBAM for NA12878",
            "generatedFrom": "GATK Test Data",
            "readSetID": "1234567890"
        }
    ]
}
```

Busque los archivos de secuencias importados

Una vez finalizado el trabajo, puede utilizar la operación de la list-read-setsAPI para buscar los archivos de secuencia importados. En el siguiente ejemplo, *`sequence store id`* sustitúyalos por el ID del almacén de secuencias.

```
aws omics list-read-sets --sequence-store-id sequence store id
```

Recibirás la siguiente respuesta.


```
{
  "readSets": [
      {
          "id": "0000000001",
          "arn": "arn:aws:omics:us-west-2:111122223333:sequenceStore/01234567890/
readSet/0000000001",
          "sequenceStoreId": "1234567890",
          "subjectId": "mySubject",
          "sampleId": "mySample",
          "status": "ACTIVE",
          "name": "HG00100",
          "description": "BAM for HG00100",
          "referenceArn": "arn:aws:omics:us-
west-2:111122223333:referenceStore/01234567890/reference/0000000001",
          "fileType": "BAM",
          "sequenceInformation": {
              "totalReadCount": 9194,
              "totalBaseCount": 928594,
              "generatedFrom": "1000 Genomes",
              "alignment": "ALIGNED"
          },
          "creationTime": "2022-07-13T23:25:20Z"
          "creationType": "IMPORT",
          "etag": {
              "algorithm": "BAM_MD5up",
              "source1": "d1d65429212d61d115bb19f510d4bd02"
          }
      },
      {
          "id": "0000000002",
          "arn": "arn:aws:omics:us-west-2:111122223333:sequenceStore/0123456789/
readSet/0000000002",
          "sequenceStoreId": "0123456789",
          "subjectId": "mySubject",
          "sampleId": "mySample",
          "status": "ACTIVE",
          "name": "HG00146",
          "description": "FASTQ for HG00146",
          "fileType": "FASTQ",
```

```
          "sequenceInformation": {
              "totalReadCount": 8000000,
              "totalBaseCount": 1184000000,
              "generatedFrom": "1000 Genomes",
              "alignment": "UNALIGNED"
          },
          "creationTime": "2022-07-13T23:26:43Z"
          "creationType": "IMPORT",
          "etag": {
              "algorithm": "FASTQ_MD5up",
              "source1": "ca78f685c26e7cc2bf3e28e3ec4d49cd"
          }
      },
      {
          "id": "0000000003",
          "arn": "arn:aws:omics:us-west-2:111122223333:sequenceStore/0123456789/
readSet/0000000003",
          "sequenceStoreId": "0123456789",
          "subjectId": "mySubject",
          "sampleId": "mySample",
          "status": "ACTIVE",
          "name": "HG00096",
          "description": "CRAM for HG00096",
          "referenceArn": "arn:aws:omics:us-
west-2:111122223333:referenceStore/0123456789/reference/0000000001",
          "fileType": "CRAM",
          "sequenceInformation": {
              "totalReadCount": 85466534,
              "totalBaseCount": 24000004881,
              "generatedFrom": "1000 Genomes",
              "alignment": "ALIGNED"
          },
          "creationTime": "2022-07-13T23:30:41Z"
          "creationType": "IMPORT",
          "etag": {
              "algorithm": "CRAM_MD5up",
              "source1": "66817940f3025a760e6da4652f3e927e"
          }
      },
      {
          "id": "0000000004",
          "arn": "arn:aws:omics:us-west-2:111122223333:sequenceStore/0123456789/
readSet/0000000004",
          "sequenceStoreId": "0123456789",
```

```
            "subjectId": "mySubject",
            "sampleId": "mySample",
            "status": "ACTIVE",
            "name": "NA12878_A",
            "description": "uBAM for NA12878",
            "fileType": "UBAM",
            "sequenceInformation": {
                "totalReadCount": 20000,
                "totalBaseCount": 5000000,
                "generatedFrom": "GATK Test Data",
                "alignment": "ALIGNED"
            },
            "creationTime": "2022-07-13T23:30:41Z"
            "creationType": "IMPORT",
            "etag": {
                "algorithm": "BAM_MD5up",
                "source1": "640eb686263e9f63bcda12c35b84f5c7"
            }
        }
    ]
}
```

Obtén detalles sobre un conjunto de lecturas

Para ver más detalles sobre un conjunto de lecturas, usa la operación de la GetReadSetMetadataAPI. En el siguiente ejemplo, *`sequence store id`* sustitúyalo por el ID del almacén de secuencias y *`read set id`* sustitúyelo por el ID del conjunto de lecturas.

```
aws omics get-read-set-metadata --sequence-store-id sequence store id --id read set id
```

Recibirás la siguiente respuesta.

```
{
"arn": "arn:aws:omics:us-west-2:123456789012:sequenceStore/2015356892/
readSet/9515444019",
"creationTime": "2024-01-12T04:50:33.548Z",
"creationType": "IMPORT",
"creationJobId": "33222111",
"description": null,
"etag": {
  "algorithm": "FASTQ_MD5up",
```

```
  "source1": "00d0885ba3eeb211c8c84520d3fa26ec",
  "source2": "00d0885ba3eeb211c8c84520d3fa26ec"
},
"fileType": "FASTQ",
"files": {
  "index": null,
  "source1": {
    "contentLength": 10818,
    "partSize": 104857600,
    "s3Access": {
      "s3Uri": "s3://accountID-sequence store ID-ajdpi90jdas90a79fh9a8ja98jdfa9jf98-
s3alias/592761533288/sequenceStore/2015356892/readSet/9515444019/
import_source1.fastq.gz"
},
    "totalParts": 1
  },
  "source2": {
    "contentLength": 10818,
    "partSize": 104857600,
    "s3Access": {
      "s3Uri": "s3://accountID-sequence store ID-ajdpi90jdas90a79fh9a8ja98jdfa9jf98-
s3alias/592761533288/sequenceStore/2015356892/readSet/9515444019/
import_source1.fastq.gz"
    },
    "totalParts": 1
  }
},
"id": "9515444019",
"name": "paired-fastq-import",
"sampleId": "sampleId-paired-fastq-import",
"sequenceInformation": {
  "alignment": "UNALIGNED",
  "generatedFrom": null,
  "totalBaseCount": 30000,
  "totalReadCount": 200
},
"sequenceStoreId": "2015356892",
"status": "ACTIVE",
"statusMessage": null,
"subjectId": "subjectId-paired-fastq-import"
}
```

Descargue los archivos de datos del conjunto de lectura

Puede acceder a los objetos de un conjunto de lecturas activo mediante la operación de la GetObject API Amazon S3. El URI del objeto se devuelve en la respuesta de la GetReadSetMetadataAPI. Para obtener más información, consulte Acceso a conjuntos de HealthOmics lectura con Amazon S3 URIs.

Como alternativa, utilice la operación HealthOmics GetReadSet API. Se puede utilizar GetReadSet para descargar en paralelo descargando partes individuales. Estas piezas son similares a las piezas de Amazon S3. El siguiente es un ejemplo de cómo descargar la parte 1 de un conjunto de lecturas. En el siguiente ejemplo, *sequence store id* sustitúyalo por el ID del almacén de secuencias y *read set id* sustitúyelo por el ID del conjunto de lectura.

```
aws omics get-read-set --sequence-store-id sequence store id --id read set id  --part-
number 1 outfile.bam
```

También puedes usar el Gestor de HealthOmics transferencias para descargar archivos para un conjunto de HealthOmics referencia o lectura. Puedes descargar el Gestor de HealthOmics transferencias aquí. Para obtener más información sobre el uso y la configuración del Transfer Manager, consulte este GitHubrepositorio.

Carga directa a un almacén HealthOmics de secuencias

Te recomendamos que utilices el Gestor de HealthOmics transferencias para añadir archivos a tu almacén de secuencias. Para obtener más información sobre el uso de Transfer Manager, consulte este GitHubrepositorio. También puede cargar sus conjuntos de lectura directamente a un almacén de secuencias mediante las operaciones de la API de carga directa.

Los conjuntos de lectura con carga directa aparecen primero en `PROCESSING_UPLOAD` el estado. Esto significa que las partes del archivo se están cargando actualmente y que puede acceder a los metadatos del conjunto de lectura. Una vez cargadas las partes y validadas las sumas de control, el conjunto de lecturas pasa a ser `ACTIVE` y se comporta igual que un conjunto de lecturas importado.

Si la carga directa falla, el estado del conjunto de lectura se muestra como. `UPLOAD_FAILED` Puede configurar un bucket de Amazon S3 como ubicación alternativa para los archivos que no se carguen. Las ubicaciones alternativas están disponibles para los almacenes secuenciales creados después del 15 de mayo de 2023.

Temas

- Carga directamente a un almacén de secuencias mediante el AWS CLI
- Configure una ubicación alternativa

Carga directamente a un almacén de secuencias mediante el AWS CLI

Para empezar, inicie una carga de varias partes. Para ello, utilice el AWS CLI, como se muestra en el siguiente ejemplo.

Para cargar directamente mediante AWS CLI comandos

1. Cree las partes separando los datos, como se muestra en el siguiente ejemplo.

```
split -b 100MiB SRR233106_1.filt.fastq.gz source1_part_
```

2. Una vez que los archivos fuente estén divididos en partes, cree una carga de conjuntos de lectura con varias partes, como se muestra en el siguiente ejemplo. Sustituya *sequence store ID* y los demás parámetros por el ID del almacén de secuencias y otros valores.

```
aws omics create-multipart-read-set-upload \
--sequence-store-id sequence store ID \
--name upload name \
--source-file-type FASTQ \
--subject-id subject ID \
--sample-id sample ID \
--description "FASTQ for HG00146" "description of upload" \
--generated-from "1000 Genomes""source of imported files"
```

Obtendrá los metadatos `uploadID` y otros en la respuesta. Úsalo `uploadID` para el siguiente paso del proceso de carga.

```
{
"sequenceStoreId": "1504776472",
"uploadId": "7640892890",
"sourceFileType": "FASTQ",
"subjectId": "mySubject",
"sampleId": "mySample",
"generatedFrom": "1000 Genomes",
"name": "HG00146",
"description": "FASTQ for HG00146",
"creationTime": "2023-11-20T23:40:47.437522+00:00"
```

```
}
```

3. Añada sus conjuntos de lectura a la carga. Si tu archivo es lo suficientemente pequeño, solo tienes que realizar este paso una vez. Para archivos más grandes, realice este paso para cada parte del archivo. Si carga una pieza nueva utilizando un número de pieza utilizado anteriormente, se sobrescribirá la pieza cargada anteriormente.

   En el siguiente ejemplo, sustituya *sequence store IDupload ID*, y los demás parámetros por sus valores.

```
aws omics upload-read-set-part \
--sequence-store-id sequence store ID \
--upload-id upload ID \
--part-source SOURCE1 \
--part-number part number \
--payload  source1/source1_part_aa.fastq.gz
```

   La respuesta es un identificador que puede utilizar para comprobar que el archivo cargado coincide con el archivo deseado.

```
{
"checksum": "984979b9928ae8d8622286c4a9cd8e99d964a22d59ed0f5722e1733eb280e635"
}
```

4. Continúe cargando las partes del archivo, si es necesario. Para comprobar que los conjuntos de lectura se han cargado, utilice la operación de la API list-read-set-upload-parts, tal y como se muestra a continuación. En el siguiente ejemplo, sustituya *sequence store ID upload ID*, y por *part source* su propia entrada.

```
aws omics list-read-set-upload-parts \
 --sequence-store-id sequence store ID \
 --upload-id upload ID \
 --part-source SOURCE1
```

   La respuesta devuelve el número de conjuntos de lectura, el tamaño y la marca de tiempo de su última actualización.

```
{
"parts": [
    {
```

```
        "partNumber": 1,
        "partSize": 104857600,
        "partSource": "SOURCE1",
        "checksum": "MVMQk+vB9C3Ge8ADHkbKq752n3BCUzyl41qEkqlOD5M=",
        "creationTime": "2023-11-20T23:58:03.500823+00:00",
        "lastUpdatedTime": "2023-11-20T23:58:03.500831+00:00"
    },
    {
        "partNumber": 2,
        "partSize": 104857600,
        "partSource": "SOURCE1",
        "checksum": "keZzVzJNChAqgOdZMvOmjBwrOPM0enPj1UAfs0nvRto=",
        "creationTime": "2023-11-21T00:02:03.813013+00:00",
        "lastUpdatedTime": "2023-11-21T00:02:03.813025+00:00"
    },
    {
        "partNumber": 3,
        "partSize": 100339539,
        "partSource": "SOURCE1",
        "checksum": "TBkNfMsaeDpXzEf3ldlbi0ipFDPaohKHyZ+LF1J4CHk=",
        "creationTime": "2023-11-21T00:09:11.705198+00:00",
        "lastUpdatedTime": "2023-11-21T00:09:11.705208+00:00"
    }
]
}
```

5. Para ver todas las cargas de conjuntos de lectura multiparte activos, usa list-multipart-read-set-uploads, tal y como se muestra a continuación. *`sequence store ID`*Sustitúyalo por el ID de tu propio almacén de secuencias.

```
aws omics list-multipart-read-set-uploads --sequence-store-id
            sequence store ID
```

Esta API solo devuelve las cargas de conjuntos de lectura de varias partes que estén en curso. Una vez que se hayan introducido los conjuntos de **`ACTIVE`** lecturas, o si la carga ha fallado, la carga no se devolverá en respuesta a la list-multipart-read-set API -uploads. Para ver los conjuntos de lecturas activos, usa la API. list-read-sets A continuación, se muestra un ejemplo de respuesta para list-multipart-read-set-uploads.

```
{
"uploads": [
```

```
    {
        "sequenceStoreId": "1234567890",
        "uploadId": "8749584421",
        "sourceFileType": "FASTQ",
        "subjectId": "mySubject",
        "sampleId": "mySample",
        "generatedFrom": "1000 Genomes",
        "name": "HG00146",
        "description": "FASTQ for HG00146",
        "creationTime": "2023-11-29T19:22:51.349298+00:00"
    },
    {
        "sequenceStoreId": "1234567890",
        "uploadId": "5290538638",
        "sourceFileType": "BAM",
        "subjectId": "mySubject",
        "sampleId": "mySample",
        "generatedFrom": "1000 Genomes",
        "referenceArn": "arn:aws:omics:us-
west-2:123456789012:referenceStore/8168613728/reference/2190697383",
        "name": "HG00146",
        "description": "BAM for HG00146",
        "creationTime": "2023-11-29T19:23:33.116516+00:00"
    },
    {
        "sequenceStoreId": "1234567890",
        "uploadId": "4174220862",
        "sourceFileType": "BAM",
        "subjectId": "mySubject",
        "sampleId": "mySample",
        "generatedFrom": "1000 Genomes",
        "referenceArn": "arn:aws:omics:us-
west-2:123456789012:referenceStore/8168613728/reference/2190697383",
        "name": "HG00147",
        "description": "BAM for HG00147",
        "creationTime": "2023-11-29T19:23:47.007866+00:00"
    }
]
}
```


6. Después de cargar todas las partes del archivo, usa complete-multipart-read-set-upload para finalizar el proceso de carga, como se muestra en el siguiente ejemplo. Sustituya *sequence store ID* y el parámetro de las piezas por sus propios valores. *upload ID*

```
aws omics complete-multipart-read-set-upload \
--sequence-store-id sequence store ID \
--upload-id upload ID \
--parts '[{"checksum":"gaCBQMe+rpCFZxLpoP6gydBoXaKKDA/
Vobh5zBDb4W4=","partNumber":1,"partSource":"SOURCE1"}]'
```

La respuesta para complete-multipart-read-set-upload es el conjunto de lecturas de IDs los conjuntos de lecturas importados.

```
{
"readSetId": "0000000001"
}
```

7. Para detener la carga, usa abort-multipart-read-set-upload con el ID de carga para finalizar el proceso de carga. Sustituya *sequence store ID* y *upload ID* por sus propios valores de parámetros.

```
aws omics abort-multipart-read-set-upload \
--sequence-store-id sequence store ID \
--upload-id upload ID
```

8. Una vez finalizada la carga, recupere los datos del conjunto de lectura mediante get-read-setel siguiente procedimiento. Si la carga aún se está procesando, get-read-setdevuelve metadatos limitados y los archivos de índice generados no están disponibles. Sustituya *sequence store ID* los demás parámetros por sus propios datos.

```
aws omics get-read-set
 --sequence-store-id sequence store ID \
 --id read set ID \
 --file SOURCE1 \
 --part-number 1 myfile.fastq.gz
```

9. Para comprobar los metadatos, incluido el estado de la carga, utiliza la operación de la get-read-set-metadataAPI.

```
aws omics get-read-set-metadata --sequence-store-id sequence store ID --id read set
 ID
```

La respuesta incluye detalles de metadatos como el tipo de archivo, el ARN de referencia, el número de archivos y la longitud de las secuencias. También incluye el estado. Los estados posibles son `PROCESSING_UPLOADACTIVE`, y`UPLOAD_FAILED`.


```
{
"id": "0000000001",
"arn": "arn:aws:omics:us-west-2:555555555555:sequenceStore/0123456789/
readSet/0000000001",
"sequenceStoreId": "0123456789",
"subjectId": "mySubject",
"sampleId": "mySample",
"status": "PROCESSING_UPLOAD",
"name": "HG00146",
"description": "FASTQ for HG00146",
"fileType": "FASTQ",
"creationTime": "2022-07-13T23:25:20Z",
"files": {
    "source1": {
        "totalParts": 5,
        "partSize": 123456789012,
        "contentLength": 6836725,

    },
    "source2": {
        "totalParts": 5,
        "partSize": 123456789056,
        "contentLength": 6836726
    }
},
'creationType": "UPLOAD"
}
```


Configure una ubicación alternativa

Al crear o actualizar un almacén de secuencias, puede configurar un bucket de Amazon S3 como ubicación de respaldo para los archivos que no se cargan. Las partes del archivo de esos conjuntos de lectura se transfieren a la ubicación alternativa. Las ubicaciones de respaldo están disponibles para los almacenes de secuencias creados después del 15 de mayo de 2023.

Cree una política de bucket de Amazon S3 para conceder acceso de HealthOmics escritura a la ubicación alternativa de Amazon S3, como se muestra en el siguiente ejemplo:

```
{
    "Effect": "Allow",
    "Principal": {
        "Service": "omics.amazonaws.com"
    },
    "Action": "s3:PutObject",
    "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
}
```

Si el depósito de Amazon S3 para los registros alternativos o de acceso utiliza una clave administrada por el cliente, añada los siguientes permisos a la política de claves:

```
 {
    "Sid": "Allow use of key",
    "Effect": "Allow",
    "Principal": {
        "Service": "omics.amazonaws.com"
    },
    "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey*"
    ],
    "Resource": "*"
}
```

Exportación de conjuntos de HealthOmics lectura a un bucket de Amazon S3

Puede exportar conjuntos de lectura como un trabajo de exportación por lotes a un bucket de Amazon S3. Para ello, cree primero una política de IAM que tenga acceso de escritura al bucket, similar al siguiente ejemplo de política de IAM.

JSON

```
{
  "Version":"2012-10-17",
```

```
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:GetBucketLocation"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket1",
        "arn:aws:s3:::amzn-s3-demo-bucket1/*"
      ]
    }
  ]
}
```

JSON

```
{
"Version":"2012-10-17",
"Statement": [
  {
      "Effect": "Allow",
      "Principal": {
          "Service": [
              "omics.amazonaws.com"
          ]
      },
      "Action": "sts:AssumeRole"
  }
]
}
```

Una vez establecida la política de IAM, comience el trabajo de exportación del conjunto de lectura. En el siguiente ejemplo, se muestra cómo hacerlo mediante la operación de API start-read-set-export-job. En el siguiente ejemplo, sustituya todos los parámetros,*sequence store ID*, *destinationrole ARN*, y*sources*, por su entrada.

```
aws omics start-read-set-export-job
--sequence-store-id sequence store id \
--destination valid s3 uri \
```

```
--role-arn role ARN \
--sources readSetId=read set id_1 readSetId=read set id_2
```

Recibirá la siguiente respuesta con información sobre el almacén de secuencias de origen y el bucket de Amazon S3 de destino.

```
{
"id": <job-id>,
"sequenceStoreId": <sequence-store-id>,
"destination": <destination-s3-uri>,
"status": "SUBMITTED",
"creationTime": "2022-10-22T01:33:38.079000+00:00"
}
```

Una vez iniciado el trabajo, puede determinar su estado mediante la operación de API get-read-set-export-job, como se muestra a continuación. Sustituya *sequence store ID* y *job ID* por su ID de almacén de secuencias y su ID de trabajo, respectivamente.

```
aws omics get-read-set-export-job --id job-id --sequence-store-id sequence store ID
```

Puede ver todos los trabajos de exportación inicializados para un almacén de secuencias mediante la operación de la API list-read-set-export-jobs, como se muestra a continuación. Sustitúyalo por el *sequence store ID* ID del almacén de secuencias.

```
aws omics list-read-set-export-jobs --sequence-store-id sequence store ID.
```

```
{
"exportJobs": [
  {
      "id": <job-id>,
      "sequenceStoreId": <sequence-store-id>,
      "destination": <destination-s3-uri>,
      "status": "COMPLETED",
      "creationTime": "2022-10-22T01:33:38.079000+00:00",
      "completionTime": "2022-10-22T01:34:28.941000+00:00"
  }
]
}
```

Además de exportar sus conjuntos de lectura, también puede compartirlos mediante el acceso a Amazon S3 URIs. Para obtener más información, consulte Acceso a conjuntos de HealthOmics lectura con Amazon S3 URIs.

Acceso a conjuntos de HealthOmics lectura con Amazon S3 URIs

Puede utilizar las rutas URI de Amazon S3 para acceder a los conjuntos de lecturas de su almacén de secuencias activo.

Con la ruta URI de Amazon S3, puede utilizar las operaciones de Amazon S3 para enumerar, compartir y descargar sus conjuntos de lectura. El acceso a través del S3 APIs acelera la colaboración y la integración de herramientas, dado que muchas herramientas del sector ya están diseñadas para leer desde S3. Además, puede compartir el acceso al S3 APIs con otras cuentas y proporcionar acceso de lectura a los datos entre regiones.

HealthOmics no admite el acceso mediante URI de Amazon S3 a conjuntos de lectura archivados. Al activar un conjunto de lecturas, se restaura en la misma ruta de URI cada vez.

Con los datos cargados en HealthOmics las tiendas, dado que el URI de Amazon S3 se basa en los puntos de acceso de Amazon S3, puede integrarse directamente con las herramientas estándar del sector que leen Amazon S3 URIs, como las siguientes:

- Aplicaciones de análisis visual como Integrative Genomics Viewer (IGV) o UCSC Genome Browser.
- Flujos de trabajo habituales con extensiones de Amazon S3, como CWL, WDL y Nextflow.
- Cualquier herramienta que pueda autenticar y leer desde el punto de acceso Amazon S3 URIs o leer Amazon S3 prefirmado. URIs
- Utilidades de Amazon S3 como Mountpoint o. CloudFront

Amazon S3 Mountpoint le permite utilizar un bucket de Amazon S3 como sistema de archivos local. Para obtener más información sobre Mountpoint e instalarlo para su uso, consulte Mountpoint for Amazon S3.

Amazon CloudFront es un servicio de red de entrega de contenido (CDN) creado para ofrecer un alto rendimiento, seguridad y comodidad para los desarrolladores. Para obtener más información sobre el uso de Amazon CloudFront, consulta la CloudFront documentación de Amazon. Para configurar CloudFront una tienda secuencial, ponte en contacto con el AWS HealthOmics equipo.

La cuenta raíz del propietario de los datos está habilitada para las acciones S3:GetObject, S3:GetObjectTagging y S3:List Bucket en el prefijo del almacén de secuencias. Para que un usuario de la cuenta acceda a los datos, debe crear una política de IAM y asociarla al usuario o rol. Para ver una política de ejemplo, consulte Permisos de acceso a los datos mediante Amazon S3 URIs.

Puede utilizar las siguientes operaciones de la API de Amazon S3 en los conjuntos de lectura activos para enumerar y recuperar sus datos. Puede acceder a los conjuntos de lectura archivados a través de Amazon S3 URIs después de haberlos activado.

- GetObject— Recupera un objeto de Amazon S3.
- HeadObject— La operación HEAD recupera los metadatos de un objeto sin devolver el objeto en sí. Esta operación es útil si solo desea los metadatos de un objeto.
- ListObjects y ListObject v2: devuelve algunos o todos (hasta 1000) los objetos de un depósito.
- CopyObject— Crea una copia de un objeto que ya está almacenado en Amazon S3. HealthOmicsadmite copiar en un punto de acceso Amazon S3, pero no escribir en un punto de acceso.

HealthOmics Los almacenes de secuencias mantienen la identidad semántica de los archivos a través ETags de ellos. A lo largo del ciclo de vida de un archivo, Amazon S3 ETag, que se basa en la identidad bit a bit, puede cambiar, pero HealthOmics ETag sigue siendo el mismo. Para obtener más información, consulte HealthOmics ETags y procedencia de los datos.

Temas

- Estructura de URI de Amazon S3 en el HealthOmics almacenamiento
- Uso de IGV alojado o local para acceder a los conjuntos de lectura
- Usando Samtools o en HTSlib HealthOmics
- Uso de Mountpoint HealthOmics
- Utilizándolo CloudFront con HealthOmics

Estructura de URI de Amazon S3 en el HealthOmics almacenamiento

Todos los archivos de Amazon S3 URIs tienen `omics:subjectId` etiquetas `omics:sampleId` de recursos. Puede utilizar estas etiquetas para compartir el acceso mediante las políticas de IAM siguiendo un patrón como`"s3:ExistingObjectTag/omics:subjectId": "pattern desired"`.

La estructura de archivos es la siguiente:

`.../account_id/sequenceStore/seq_store_id/readSet/read_set_id/files.`

En el caso de los archivos importados a almacenes de secuencias desde Amazon S3, el almacén de secuencias intenta mantener el nombre de la fuente original. Cuando los nombres entran en conflicto, el sistema agrega la información del conjunto de lecturas para garantizar que los nombres de los archivos sean únicos. Por ejemplo, en el caso de los conjuntos de lectura fastq, si ambos nombres de archivo son iguales, para que sean únicos, `sourceX` se inserta antes de .fastq.gz o .fq.gz. Para una carga directa, los nombres de los archivos siguen los siguientes patrones:

- Para FASTQ: *read_set_name* _ .fastq.gz *sourcex*
- uBAM/BAM/CRAMPara *read_set_name* —. *file extension*con extensiones de `.bam` o `.cram`. Un ejemplo es `NA193948.bam`.

En el caso de los conjuntos de lectura que son BAM o CRAM, los archivos de índice se generan automáticamente durante el proceso de ingesta. Para los archivos de índice generados, se aplica la extensión de índice adecuada al final del nombre del archivo. Tiene el patrón *<name of the Source the index is on>.<file index extension>.* Las extensiones de índice son `.bai` o `.crai`.

Uso de IGV alojado o local para acceder a los conjuntos de lectura

IGV es un navegador de genomas que se utiliza para analizar archivos BAM y CRAM. Requiere tanto el archivo como el índice porque solo muestra una parte del genoma a la vez. El IGV se puede descargar y usar localmente, y hay guías para crear un IGV alojado en AWS. La versión web pública no es compatible porque requiere CORS.

El IGV local se basa en la AWS configuración local para acceder a los archivos. Asegúrese de que el rol utilizado en esa configuración tenga una política adjunta que habilite los GetObject permisos kms: Decrypt y s3: para el URI s3 de los conjuntos de lectura a los que se accede. Después, en IGV, puedes usar «Archivo > cargar desde URL» y pegar el URI de la fuente y el índice. Como alternativa, los prefirmados se URLs pueden generar y usar de la misma manera, lo que omitirá la configuración de AWS. Tenga en cuenta que CORS no es compatible con el acceso mediante URI de Amazon S3, por lo que no se admiten solicitudes que dependan de CORS.

El ejemplo de IGV AWS alojado se basa en AWS Cognito para crear las configuraciones y los permisos correctos dentro del entorno. Asegúrese de crear una política que habilite los permisos KMS:Decrypt y s3: para GetObject el URI de Amazon S3 de los conjuntos de lectura a los que

se accede y añada esta política al rol asignado al grupo de usuarios de Cognito. Después, en IGV, puede usar «Archivo > cargar desde URL» e introducir el URI de la fuente y el índice. Como alternativa, los prefirmados se URLs pueden generar y usar de la misma manera, lo que evita la configuración de AWS.

Tenga en cuenta que el almacén de secuencias no aparecerá en la pestaña «Amazon», ya que solo muestra los depósitos de su propiedad en la región en la que está configurado el AWS perfil.

Usando Samtools o en HTSlib HealthOmics

HTSlib es la biblioteca principal que comparten varias herramientas, como Samtools, RSAMtools y otras. PySam Utilice HTSlib la versión 1.20 o posterior para obtener una compatibilidad perfecta con los puntos de acceso Amazon S3. Para las versiones anteriores de la HTSlib biblioteca, puede utilizar las siguientes soluciones alternativas:

- Establezca la variable de entorno para el host HTS Amazon S3 con:`export HTS_S3_HOST="s3.region.amazonaws.com"`.
- Genere una URL prefirmada para los archivos que desee usar. Si se utiliza un BAM o un CRAM, asegúrese de generar una URL prefirmada tanto para el archivo como para el índice. Después de eso, ambos archivos se pueden usar con las bibliotecas.
- Utilice Mountpoint para montar el almacén de secuencias o leer el prefijo establecido en el mismo entorno en el que utiliza las bibliotecas. HTSlib Desde aquí, se puede acceder a los archivos mediante las rutas de archivo locales.

Uso de Mountpoint HealthOmics

Mountpoint for Amazon S3 es un cliente de archivos sencillo y de alto rendimiento para montar un bucket de Amazon S3 como un sistema de archivos local. Con Mountpoint para Amazon S3, sus aplicaciones pueden acceder a los objetos almacenados en Amazon S3 mediante operaciones de archivos como abrir y leer. Mountpoint for Amazon S3 traduce automáticamente estas operaciones en llamadas a la API de objetos de Amazon S3, lo que proporciona a sus aplicaciones acceso al almacenamiento elástico y al rendimiento de Amazon S3 a través de una interfaz de archivos.

Mountpoint se puede instalar siguiendo las instrucciones de instalación de Mountpoint. Mountpoint usa el perfil de AWS local de la instalación y funciona a nivel de prefijo de Amazon S3. Asegúrese de que el perfil que se está utilizando tenga una política que permita los permisos s3:GetObject, s3:ListBucket y kms: Decrypt para el prefijo URI de Amazon S3 de los conjuntos de lectura o el almacén de secuencias al que se accede. Después, el depósito se puede montar mediante la siguiente ruta:

```
mount-s3 access point arn local path to mount --prefix prefix to sequence store or read
 set --region region
```

Utilizándolo CloudFront con HealthOmics

Amazon CloudFront es un servicio de red de entrega de contenido (CDN) creado para ofrecer un alto rendimiento, seguridad y comodidad para los desarrolladores. Los clientes que lo deseen CloudFront deben trabajar con el equipo de servicio para activar la CloudFront distribución. Trabaje con su equipo de cuentas para contratar al equipo HealthOmics de servicio.

Activar conjuntos de lectura en HealthOmics

Puede activar los conjuntos de lecturas que se archivan con la operación de la API start-read-set-activation-job o mediante la AWS CLI, como se muestra en el siguiente ejemplo. Sustituya *sequence store ID* y *read set id* por el ID del almacén de secuencias y el conjunto de lecturas. IDs

```
aws omics start-read-set-activation-job
     --sequence-store-id sequence store ID \
     --sources readSetId=read set ID readSetId=read set id_1 read set id_2
```

Recibirá una respuesta que contiene la información del trabajo de activación, como se muestra a continuación.

```
{
    "id": "12345678",
    "sequenceStoreId": "1234567890",
    "status": "SUBMITTED",
    "creationTime": "2022-10-22T00:50:54.670000+00:00"
}
```

Una vez que se inicie el trabajo de activación, puede supervisar su progreso con la operación de la API get-read-set-activation-job. A continuación se muestra un ejemplo de cómo utilizar el AWS CLI para comprobar el estado del trabajo de activación. Sustituya *job ID* y *sequence store ID* por el ID y el trabajo del almacén de secuencias IDs, respectivamente.

```
aws omics get-read-set-activation-job --id job ID --sequence-store-id sequence store ID
```

La respuesta resume el trabajo de activación, como se muestra a continuación.

```
{
    "id": 123567890,
    "sequenceStoreId": 123467890,
    "status": "SUBMITTED",
    "statusUpdateReason": "The job is submitted and will start soon.",
    "creationTime": "2022-10-22T00:50:54.670000+00:00",
    "sources": [
        {
            "readSetId": <reads set id_1>,
            "status": "NOT_STARTED",
            "statusUpdateReason": "The source is queued for the job."
        },
        {
            "readSetId": <read set id_2>,
            "status": "NOT_STARTED",
            "statusUpdateReason": "The source is queued for the job."
        }
    ]
}
```

Puede comprobar el estado de un trabajo de activación con la operación de la get-read-set-metadataAPI. Los estados posibles son `ACTIVE`ACTIVATING, y`ARCHIVED`. En el siguiente ejemplo, *`sequence store ID`* sustitúyalo por el ID del almacén de secuencias y *`read set ID`* sustitúyelo por el ID del conjunto de lectura.

```
aws omics get-read-set-metadata --sequence-store-id sequence store ID --id read set ID
```

La siguiente respuesta muestra que el conjunto de lecturas está activo.

```
{
    "id": "12345678",
    "arn": "arn:aws:omics:us-west-2:555555555555:sequenceStore/1234567890/
readSet/12345678",
    "sequenceStoreId": "0123456789",
    "subjectId": "mySubject",
    "sampleId": "mySample",
    "status": "ACTIVE",
    "name": "HG00100",
    "description": "HG00100 aligned to HG38 BAM",
    "fileType": "BAM",
```

```
    "creationTime": "2022-07-13T23:25:20Z",
    "sequenceInformation": {
        "totalReadCount": 1513467,
        "totalBaseCount": 163454436,
        "generatedFrom": "Pulled from SRA",
        "alignment": "ALIGNED"
    },
    "referenceArn": "arn:aws:omics:us-west-2:555555555555:referenceStore/0123456789/
reference/0000000001",
    "files": {
        "source1": {
            "totalParts": 2,
            "partSize":  10485760,
            "contentLength": 17112283,
            "s3Access": {
        "s3Uri": "s3://accountID-sequence store ID-ajdpi90jdas90a79fh9a8ja98jdfa9jf98-
s3alias/592761533288/sequenceStore/2015356892/readSet/9515444019/
import_source1.fastq.gz"
},
         },
        "index": {
            "totalParts": 1,
            "partSize": 53216,
            "contentLength": 10485760
            "s3Access": {
        "s3Uri": "s3://accountID-sequence store ID-ajdpi90jdas90a79fh9a8ja98jdfa9jf98-
s3alias/592761533288/sequenceStore/2015356892/readSet/9515444019/
import_source1.fastq.gz"
},
        }
    },
    "creationType": "IMPORT",
    "etag": {
        "algorithm": "BAM_MD5up",
        "source1": "d1d65429212d61d115bb19f510d4bd02"
    }
}
```


Puede ver todos los trabajos de activación del conjunto de lectura mediante list-read-set-activation-jobs, como se muestra en el siguiente ejemplo. En el siguiente ejemplo, sustitúyalo *sequence store ID* por tu ID de almacén de secuencias.

```
aws omics list-read-set-activation-jobs --sequence-store-id sequence store ID
```

Recibirás la siguiente respuesta.

```
{
    "activationJobs": [
        {
            "id": 1234657890,
            "sequenceStoreId": "1234567890",
            "status": "COMPLETED",
            "creationTime": "2022-10-22T01:33:38.079000+00:00",
            "completionTime": "2022-10-22T01:34:28.941000+00:00"
        }
    ]
}
```

HealthOmics analítica

> **Important**
>
> AWS HealthOmics las tiendas de variantes y las tiendas de anotaciones ya no están abiertas a nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte AWS HealthOmics cambio en la disponibilidad del almacén de variantes y del almacén de anotaciones.

HealthOmics La analítica permite almacenar y analizar variantes y anotaciones genómicas. Analytics proporciona dos tipos de recursos de almacenamiento: almacenes de variantes y almacenes de anotaciones. Estos recursos se utilizan para almacenar, transformar y consultar datos de variantes genómicas y datos de anotaciones. Tras importar los datos a un almacén de datos, puede utilizar Athena para realizar análisis avanzados de los datos.

Puede usar la HealthOmics consola o la API para crear y administrar almacenes, importar datos y compartir datos analíticos del almacén con sus colaboradores.

Los almacenes de variantes admiten datos en formatos VCF, y los almacenes de anotaciones admiten TSV/CSV y dan formato. GFF3 Las coordenadas genómicas se representan como intervalos de base cero, semicerrados y semiabiertos. Cuando sus datos están en el almacén de datos de HealthOmics análisis, el acceso a los archivos VCF se gestiona a través de él. AWS Lake Formation A continuación, puede consultar los archivos VCF mediante Amazon Athena. Las consultas deben utilizar la versión 3 del motor de consultas Athena. Para obtener más información sobre las versiones del motor de consultas de Athena, consulte la documentación de Amazon Athena.

Temas

- Creación de tiendas de HealthOmics variantes
- Creación de trabajos de importación de tiendas de HealthOmics variantes
- Creación de HealthOmics almacenes de anotaciones
- Creación de trabajos de importación para HealthOmics almacenes de anotaciones
- Creación de HealthOmics versiones de almacenes de anotaciones
- Eliminar almacenes HealthOmics de análisis

- Consulta de datos HealthOmics analíticos
- Compartir tiendas HealthOmics de análisis

Creación de tiendas de HealthOmics variantes

> **Important**
>
> AWS HealthOmics las tiendas de variantes y las tiendas de anotaciones ya no están abiertas a nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte AWS HealthOmics cambio en la disponibilidad del almacén de variantes y del almacén de anotaciones.

En los siguientes temas se describe cómo crear almacenes de HealthOmics variantes mediante la consola y la API.

Temas

- Crear un almacén de variantes mediante la consola
- Crear un almacén de variantes mediante la API

Crear un almacén de variantes mediante la consola

Puede crear un almacén de variantes mediante la HealthOmics consola.

1. Abra la consola de HealthOmics .
2. Si es necesario, abre el panel de navegación izquierdo (≡). Selecciona Variant stores.
3. En la página Crear tienda de variantes, proporciona la siguiente información
   - Nombre de la tienda de variantes: un nombre exclusivo para esta tienda.
   - Descripción (opcional): descripción de esta tienda de variantes.
   - Genoma de referencia: el genoma de referencia de este almacén de variantes.
   - Cifrado de datos: elija si desea que el cifrado de datos sea de su propiedad y lo gestione usted mismo. AWS
   - Etiquetas (opcional): proporciona hasta 50 etiquetas para este almacén de variantes.
4. Selecciona Crear tienda de variantes.

Crear un almacén de variantes mediante la API

Utilice la operación de la HealthOmics `CreateVariantStore` API para crear almacenes de variantes. También puede realizar esta operación con AWS CLI.

Para crear una tienda de variantes, debes proporcionar un nombre para la tienda y el ARN de una tienda de referencia. El almacén de variantes está listo para ingerir datos cuando su estado cambie a LISTO.

En el siguiente ejemplo, se utiliza AWS CLI para crear un almacén de variantes.

```
aws omics create-variant-store --name myvariantstore \
    --reference referenceArn="arn:aws:omics:us-
west-2:555555555555:referenceStore/123456789/reference/5987565360"
```

Para confirmar la creación de tu tienda de variantes, recibirás la siguiente respuesta.

```
{
    "creationTime": "2022-11-03T18:19:52.296368+00:00",
    "id": "45aeb91d5678",
    "name": "myvariantstore",
    "reference": {
        "referenceArn": "arn:aws:omics:us-west-2:555555555555:referenceStore/123456789/
reference/5987565360"
    },
    "status": "CREATING"
}
```

Para obtener más información sobre una tienda de variantes, usa la get-variant-storeAPI.

```
aws omics get-variant-store --name myvariantstore
```

Recibirás la siguiente respuesta.

```
{
    "id": "45aeb91d5678",
    "reference": {
        "referenceArn": "arn:aws:omics:us-west-2:555555555555:referenceStore/123456789/
reference/5987565360"
    },
    "status": "ACTIVE",
```

```
    "storeArn": "arn:aws:omics:us-west-2:555555555555:variantStore/myvariantstore",
    "name": "myvariantstore",
    "creationTime": "2022-11-03T18:19:52.296368+00:00",
    "updateTime": "2022-11-03T18:30:56.272792+00:00",
    "tags": {},
    "storeSizeBytes": 0
}
```

Para ver todas las tiendas de variantes asociadas a una cuenta, usa la list-variant-storesAPI.

```
aws omics list-variant-stores
```

Recibirás una respuesta en la que se enumeran todas las tiendas de variantes IDs, junto con sus estados y otros detalles, como se muestra en el siguiente ejemplo de respuesta.

```
{
    "variantStores": [
        {
            "id": "45aeb91d5678",
            "reference": {
                "referenceArn": "arn:aws:omics:us-
west-2:55555555555:referenceStore/5506874698"
            },
            "status": "ACTIVE",
            "storeArn": "arn:aws:omics:us-west-2:55555555555:variantStore/
new_variant_store",
            "name": "variantstore",
            "creationTime": "2022-11-03T18:19:52.296368+00:00",
            "updateTime": "2022-11-03T18:30:56.272792+00:00",
            "statusMessage": "",
            "storeSizeBytes": 141526
        }
    ]
}
```

También puedes filtrar las respuestas de la list-variant-storesAPI en función de los estados u otros criterios.

Los archivos VCF importados a almacenes de análisis creados a partir del 15 de mayo de 2023 tienen esquemas definidos para las anotaciones del Variant Effect Predictor (VEP). Esto facilita la consulta y el análisis de los datos VCF importados. El cambio no afecta a las tiendas creadas antes

del 15 de mayo de 2023, excepto si el `annotation fields` parámetro se incluye en la llamada a la API o a la CLI. En el caso de estas tiendas, el uso del `annotation fields` parámetro provocará un error en la solicitud.

Creación de trabajos de importación de tiendas de HealthOmics variantes

> **Important**
>
> AWS HealthOmics Las tiendas de variantes y las tiendas de anotaciones ya no están abiertas a nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte AWS HealthOmics cambio en la disponibilidad del almacén de variantes y del almacén de anotaciones.

En el siguiente ejemplo, se muestra cómo utilizar el AWS CLI para crear un trabajo de importación para un almacén de variantes.

```
aws omics start-variant-import-job \
       --destination-name myvariantstore \
       --runLeftNormalization false \
       --role-arn  arn:aws:iam::55555555555:role/roleName \
       --items source=s3://my-omics-bucket/sample.vcf.gz source=s3://my-omics-bucket/
sample2.vcf.gz
```

```
{
    "destinationName": "store_a",
    "roleArn": "....",
    "runLeftNormalization": false,
    "items": [
        {"source": "s3://my-omics-bucket/sample.vcf.gz"},
        {"source": "s3://my-omics-bucket/sample2.vcf.gz"}
    ]
}
```

En el caso de las tiendas creadas después del 15 de mayo de 2023, en el siguiente ejemplo se muestra cómo añadir el `--annotation-fields` parámetro. Los campos de anotación se definen con la importación.

```
aws omics start-variant-import-job \
   --destination-name annotationparsingvariantstore \
   --role-arn arn:aws:iam::123456789012:role/<role_name> \
   --items source=s3://pathToS3/sample.vcf
   --annotation-fields '{"VEP": "CSQ"}'
```

```
{
    "jobId": "981e2286-e954-4391-8a97-09aefc343861"
}
```

Se utiliza get-variant-import-jobpara comprobar el estado.

```
aws omics get-variant-import-job --job-id 08279950-a9e3-4cc3-9a3c-a574f9c9e229
```

Recibirás una respuesta en JSON que muestra el estado de tu trabajo de importación. Las anotaciones VEP del VCF se analizan para buscar la información almacenada en la columna INFO como un par. ID/Value El identificador predeterminado de la columna INFO de las anotaciones del predictor de efectos variantes de Ensembl es CSQ, pero puede utilizar el `--annotation-fields` parámetro para indicar un valor personalizado utilizado en la columna INFO. Actualmente, se admite el análisis sintáctico para las anotaciones VEP.

En el caso de un almacén creado antes del 15 de mayo de 2023 o de archivos VCF que no incluyen la anotación VEP, la respuesta no incluye ningún campo de anotación.

```
{
    "creationTime": "2023-04-11T17:52:37.241958+00:00",
    "destinationName": "annotationparsingvariantstore",
    "id": "7a1c67e3-b7f9-434d-817b-9c571fd63bea",
    "items": [

    {
       "jobStatus": "COMPLETED",
       "source": "s3://amzn-s3-demo-bucket/NA12878.2k.garvan.vcf"
    }
 ],
    "roleArn": "arn:aws:iam::555555555555:role/<role_name>",

    "runLeftNormalization": false,
    "status": "COMPLETED",
    "updateTime": "2023-04-11T17:58:22.676043+00:00",
```

```
}
```

Las anotaciones VEP que forman parte de los archivos VCF se almacenan como un esquema predefinido con la siguiente estructura. El campo extras se puede usar para almacenar cualquier campo VEP adicional que no esté incluido en el esquema predeterminado.

```
annotations struct<
   vep: array<struct<
      allele:string,
      consequence: array<string>,
      impact:string,
      symbol:string,
      gene:string,
      `feature_type`: string,
      feature: string,
      biotype: string,
      exon: struct<rank:string, total:string>,
      intron: struct<rank:string, total:string>,
      hgvsc: string,
      hgvsp: string,
      `cdna_position`: string,
      `cds_position`: string,
      `protein_position`: string,
      `amino_acids`: struct<reference:string, variant: string>,
      codons: struct<reference:string, variant: string>,
      `existing_variation`: array<string>,
      distance: string,
      strand: string,
      flags: array<string>,
      symbol_source: string,
      hgnc_id: string,
      `extras`: map<string, string>
    >>
>
```

El análisis se realiza con el máximo esfuerzo. Si la entrada del VEP no sigue las especificaciones estándar del VEP, no se analizará y la fila de la matriz estará vacía.

En el caso de un nuevo almacén de variantes, la respuesta para get-variant-import-jobincluiría los campos de anotación, como se muestra.

```
aws omics get-variant-import-job --job-id 08279950-a9e3-4cc3-9a3c-a574f9c9e229
```

Recibirás una respuesta en JSON que muestra el estado de tu trabajo de importación.

```
{
    "creationTime": "2023-04-11T17:52:37.241958+00:00",
    "destinationName": "annotationparsingvariantstore",
    "id": "7a1c67e3-b7f9-434d-817b-9c571fd63bea",
    "items": [

    {
    "jobStatus": "COMPLETED",
    "source": "s3://amzn-s3-demo-bucket/NA12878.2k.garvan.vcf"
    }
 ],
    "roleArn": "arn:aws:iam::123456789012:role/<role_name>",
    "runLeftNormalization": false,
    "status": "COMPLETED",
    "updateTime": "2023-04-11T17:58:22.676043+00:00",
    "annotationFields" : {"VEP": "CSQ"}
  }
}
```

Puede utilizarla list-variant-import-jobspara ver todos los trabajos de importación y sus estados.

```
aws omics list-variant-import-jobs --ids 7a1c67e3-b7f9-434d-817b-9c571fd63bea
```

La respuesta contiene la siguiente información.

```
{
    "variantImportJobs": [
    {
        "creationTime": "2023-04-11T17:52:37.241958+00:00",
        "destinationName": "annotationparsingvariantstore",
        "id": "7a1c67e3-b7f9-434d-817b-9c571fd63bea",
        "roleArn": "arn:aws:iam::55555555555:role/roleName",
        "runLeftNormalization": false,
        "status": "COMPLETED",
        "updateTime": "2023-04-11T17:58:22.676043+00:00",
        "annotationFields" : {"VEP": "CSQ"}
        }
    ]
  }
```

```
}
```

Si es necesario, puede cancelar un trabajo de importación con el siguiente comando.

```
aws omics cancel-variant-import-job
     --job-id edd7b8ce-xmpl-47e2-bc99-258cac95a508
```

Creación de HealthOmics almacenes de anotaciones

> **Important**
>
> AWS HealthOmics Los almacenes de variantes y los almacenes de anotaciones ya no están abiertos a nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte AWS HealthOmics cambio en la disponibilidad del almacén de variantes y del almacén de anotaciones.

Un almacén de anotaciones es un banco de datos que representa una base de datos de anotaciones, como una de un archivo TSV, VCF o GFF. Si se especifica el mismo genoma de referencia, los almacenes de anotaciones se asignan al mismo sistema de coordenadas que los almacenes de variantes durante la importación. En los temas siguientes se muestra cómo utilizar la HealthOmics consola y cómo AWS CLI crear y gestionar almacenes de anotaciones.

Temas

- Crear un almacén de anotaciones mediante la consola
- Crear un almacén de anotaciones mediante la API

Crear un almacén de anotaciones mediante la consola

Utilice el siguiente procedimiento para crear almacenes de anotaciones con la HealthOmics consola.

Para crear un almacén de anotaciones

1. Abra la consola de HealthOmics .
2. Si es necesario, abra el panel de navegación izquierdo (≡). Seleccione Almacenes de anotaciones.

3. En la página Almacenes de anotaciones, elija Crear almacén de anotaciones.
4. En la página Crear almacén de anotaciones, proporcione la siguiente información

   - Nombre del almacén de anotaciones: nombre exclusivo de este almacén.
   - Descripción (opcional): descripción de este genoma de referencia.
   - Formato de datos y detalles del esquema: seleccione el formato del archivo de datos y cargue la definición del esquema para este almacén.
   - Genoma de referencia: el genoma de referencia para esta anotación.
   - Cifrado de datos: elija si desea que el cifrado de datos sea de su propiedad y lo AWS gestione usted mismo.
   - Etiquetas (opcional): proporciona hasta 50 etiquetas para este almacén de anotaciones.
5. Seleccione Crear almacén de anotaciones.

Crear un almacén de anotaciones mediante la API

El siguiente ejemplo muestra cómo crear un almacén de anotaciones mediante. AWS CLI Para todas las operaciones AWS CLI y las de la API, debe especificar el formato de los datos.

```
aws omics create-annotation-store --name my_annotation_store \
           --store-format GFF \
           --reference referenceArn="arn:aws:omics:us-
west-2:555555555555:referenceStore/6505293348/reference/5987565360"
           --version-name new_version
```

Recibirá la siguiente respuesta para confirmar la creación de su almacén de anotaciones.

```
{
           "creationTime": "2022-08-24T20:34:19.229500Z",
           "id": "3b93cdef69d2",
           "name": "my_annotation_store",
           "reference": {
               "referenceArn": "arn:aws:omics:us-
west-2:555555555555:referenceStore/6505293348/reference/5987565360"
           },
           "status": "CREATING"
           "versionName": "my_version"
       }
```

Para obtener más información sobre un almacén de anotaciones, usa la get-annotation-storeAPI.

```
aws omics get-annotation-store --name my_annotation_store
```

Recibirás la siguiente respuesta.

```
{
          "id": "eeb019ac79c2",
          "reference": {
              "referenceArn": "arn:aws:omics:us-
west-2:555555555555:referenceStore/5638433913/reference/5871590330“
          },
          "status": "ACTIVE",
          "storeArn": "arn:aws:omics:us-west-2:555555555555:annotationStore/gffstore",
          "name": "my_annotation_store",
          "creationTime": "2022-11-05T00:05:19.136131+00:00",
          "updateTime": "2022-11-05T00:10:36.944839+00:00",
          "tags": {},
          "storeFormat": "GFF",
          "statusMessage": "",
          "storeSizeBytes": 0,
          "numVersions": 1
      }
```

Para ver todos los almacenes de anotaciones asociados a una cuenta, utilice la operación de list-annotation-storesAPI.

```
aws omics list-annotation-stores
```

Recibirá una respuesta en la que se enumeran todos los almacenes de anotaciones, junto con sus IDs estados y otros detalles, como se muestra en el siguiente ejemplo de respuesta.

```
{
           "annotationStores": [
               {
                  "id": "4d8f3eada259",
                   "reference":
                       "referenceArn": "arn:aws:omics:us-
west-2:555555555555:referenceStore/5638433913/reference/5871590330"
                   },
                   "status": "CREATING",
```

```
                "name": "gffstore",
                "creationTime": "2022-09-27T17:30:52.182990+00:00",
                "updateTime": "2022-09-27T17:30:53.025362+00:00"
            }
        ]
    }
```

También puede filtrar las respuestas en función del estado u otros criterios.

Creación de trabajos de importación para HealthOmics almacenes de anotaciones

> **Important**
>
> AWS HealthOmics Los almacenes de variantes y los almacenes de anotaciones ya no están abiertos a nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte AWS HealthOmics cambio en la disponibilidad del almacén de variantes y del almacén de anotaciones.

Temas

- Crear un trabajo de importación de anotaciones mediante la API
- Parámetros adicionales para los formatos TSV y VCF
- Creación de almacenes de anotaciones con formato TSV
- Inicio de trabajos de importación con formato VCF

Crear un trabajo de importación de anotaciones mediante la API

En el siguiente ejemplo, se muestra cómo utilizarla AWS CLI para iniciar un trabajo de importación de anotaciones.

```
aws omics start-annotation-import-job \
          --destination-name myannostore \
          --version-name myannostore \
          --role-arn arn:aws:iam::123456789012:role/roleName \
          --items source=s3://my-omics-bucket/sample.vcf.gz
          --annotation-fields '{"VEP": "CSQ"}'
```

Los almacenes de anotaciones creados antes del 15 de mayo de 2023 devuelven un mensaje de error si se incluyen los campos de anotación. No devuelven el resultado de ninguna operación de API relacionada con los trabajos de importación del almacén de anotaciones.

A continuación, puede utilizar la operación de get-annotation-import-jobAPI y el `job ID` parámetro para obtener más información sobre el trabajo de importación de anotaciones.

```
aws omics get-annotation-import-job --job-id 9e4198fb-fa85-446c-9301-9b823a1a8ba8
```

Recibirá la siguiente respuesta, incluidos los campos de anotación.

```
{
          "creationTime": "2023-04-11T19:09:25.049767+00:00",
          "destinationName": "parsingannotationstore",
          "versionName": "parsingannotationstore",
          "id": "9e4198fb-fa85-446c-9301-9b823a1a8ba8",
          "items": [
              {
                  "jobStatus": "COMPLETED",
                  "source": "s3://my-omics-bucket/sample.vep.vcf"
              }
          ],
          "roleArn": "arn:aws:iam::55555555555:role/roleName",
          "runLeftNormalization": false,
          "status": "COMPLETED",
          "updateTime": "2023-04-11T19:13:09.110130+00:00",
          "annotationFields" : {"VEP": "CSQ"}
       }
```

Para ver todos los trabajos de importación del almacén de anotaciones, utilice. list-annotation-import-jobs

```
aws omics list-annotation-import-jobs --ids 9e4198fb-fa85-446c-9301-9b823a1a8ba8
```

La respuesta incluye los detalles y los estados de los trabajos de importación del almacén de anotaciones.

```
{
```

```
        "annotationImportJobs": [
        {
            "creationTime": "2023-04-11T19:09:25.049767+00:00",
            "destinationName": "parsingannotationstore",
            "versionName": "parsingannotationstore",
            "id": "9e4198fb-fa85-446c-9301-9b823a1a8ba8",
            "roleArn": "arn:aws:iam::55555555555:role/roleName",
            "runLeftNormalization": false,
            "status": "COMPLETED",
            "updateTime": "2023-04-11T19:13:09.110130+00:00",
            "annotationFields" : {"VEP": "CSQ"}
        }
        ]
    }
```

Parámetros adicionales para los formatos TSV y VCF

Para los formatos TSV y VCF, hay parámetros adicionales que indican a la API cómo analizar la entrada.

> **Important**
>
> Los datos de anotación CSV que se exportan con los motores de consulta devuelven directamente la información de la importación del conjunto de datos. Si los datos importados contienen fórmulas o comandos, es posible que el archivo esté sujeto a una inyección de CSV. Por lo tanto, los archivos exportados con motores de consultas pueden generar advertencias de seguridad. Para evitar actividades malintencionadas, desactive los enlaces y las macros al leer los archivos de exportación.

El analizador TSV también realiza operaciones bioinformáticas básicas, como la normalización a la izquierda y la estandarización de las coordenadas genómicas, que se enumeran en la siguiente tabla.

Tipo de formato	Description (Descripción)
Genérico	Archivo de texto genérico. No hay información genómica.

Tipo de formato	Description (Descripción)
CHR_POS	Posición inicial: 1, agrega la posición final, que es la misma POS que.
CHR_POS_REF_ALT	Contiene información sobre los alelos contig, posición de 1 base e información sobre los alelos ref y alt.
CHR_START_END_REF_ALT_ONE_BASE	Contiene información sobre los alelos contig, start, end, ref y alt. Las coordenadas se basan en 1.
CHR_START_END_ZERO_BASE	Contiene las posiciones contigua, inicial y final. Las coordenadas se basan en 0.
CHR_START_END_ONE_BASE	Contiene las posiciones contigua, inicial y final. Las coordenadas se basan en 1.
CHR_START_END_REF_ALT_ZERO_BASE	Contiene información sobre los alelos contig, start, end, ref y alt. Las coordenadas se basan en 0.

Una solicitud de almacén de anotaciones de importación de TSV tiene el siguiente aspecto.

```
aws omics start-annotation-import-job \
--destination-name tsv_anno_example \
--role-arn arn:aws:iam::555555555555:role/demoRole \
--items source=s3://demodata/genomic_data.bed.gz \
--format-options '{ "tsvOptions": {
        "readOptions": {
            "header": false,
            "sep": "\t"
        }
    }
}'
```

Creación de almacenes de anotaciones con formato TSV

En el siguiente ejemplo, se crea un almacén de anotaciones con un archivo limitado por tabulaciones que contiene un encabezado, filas y comentarios. Las coordenadas son `CHR_START_END_ONE_BASED` y contiene el mapa HG19 genético de la sinopsis del mapa genético humano de la OMIM.

```
aws omics create-annotation-store --name mimgenemap \
  --store-format TSV \
  --reference=referenceArn=arn:aws:omics:us-
west-2:555555555555:referenceStore/6505293348/reference/2310864158 \
  --store-options=tsvStoreOptions='{
    annotationType=CHR_START_END_ONE_BASE,
    formatToHeader={CHR=chromosome, START=genomic_position_start,
 END=genomic_position_end},
    schema=[
      {chromosome=STRING},
      {genomic_position_start=LONG},
      {genomic_position_end=LONG},
      {cyto_location=STRING},
      {computed_cyto_location=STRING},
      {mim_number=STRING},
      {gene_symbols=STRING},
      {gene_name=STRING},
      {approved_gene_name=STRING},
      {entrez_gene_id=STRING},
      {ensembl_gene_id=STRING},
      {comments=STRING},
      {phenotypes=STRING},
      {mouse_gene_symbol=STRING}]}'
```

Puede importar archivos con o sin encabezado. Para indicar esto en una solicitud de CLI`header=false`, utilice lo que se muestra en el siguiente ejemplo de trabajo de importación.

```
aws omics start-annotation-import-job \
   --role-arn arn:aws:iam::555555555555:role/demoRole \
   --items=source=s3://amzn-s3-demo-bucket/annotation-examples/hg38_genemap2.txt \
   --destination-name output-bucket \
   --format-options=tsvOptions='{readOptions={sep="\t",header=false,comment="#"}}'
```

El siguiente ejemplo crea un almacén de anotaciones para un archivo bed. Una lima de cama es un archivo simple delimitado por tabulaciones. En este ejemplo, las columnas son el cromosoma, el inicio, el final y el nombre de la región. Las coordenadas se basan en cero y los datos no tienen encabezado.

```
aws omics create-annotation-store \
   --name cexbed --store-format TSV \
   --reference=referenceArn=arn:aws:omics:us-
west-2:555555555555:referenceStore/6505293348/reference/2310864158 \
   --store-options=tsvStoreOptions='{
   annotationType=CHR_START_END_ZERO_BASE,
   formatToHeader={CHR=chromosome, START=start, END=end},
   schema=[{chromosome=STRING}, {start=LONG}, {end=LONG}, {name=STRING}]}'
```

A continuación, puede importar el archivo bed al almacén de anotaciones mediante el siguiente comando CLI.

```
aws omics start-annotation-import-job \
   --role-arn arn:aws:iam::555555555555:role/demoRole \
   --items=source=s3://amzn-s3-demo-bucket/TruSeq_Exome_TargetedRegions_v1.2.bed \
   --destination-name cexbed \
   --format-options=tsvOptions='{readOptions={sep="\t",header=false,comment="#"}}'
```

El siguiente ejemplo crea un almacén de anotaciones para un archivo delimitado por tabulaciones que contiene las primeras columnas de un archivo VCF, seguidas de las columnas con información de anotación. Contiene las posiciones del genoma con información sobre el cromosoma, el inicio, los alelos de referencia y los alelos alternativos, y contiene un encabezado.

```
aws omics create-annotation-store --name gnomadchrx --store-format TSV \
--reference=referenceArn=arn:aws:omics:us-
west-2:555555555555:referenceStore/6505293348/reference/2310864158 \
--store-options=tsvStoreOptions='{
    annotationType=CHR_POS_REF_ALT,
    formatToHeader={CHR=chromosome, POS=start, REF=ref, ALT=alt},
    schema=[
        {chromosome=STRING},
        {start=LONG},
        {ref=STRING},
        {alt=STRING},
        {filters=STRING},
```

```
        {ac_hom=STRING},
        {ac_het=STRING},
        {af_hom=STRING},
        {af_het=STRING},
        {an=STRING},
        {max_observed_heteroplasmy=STRING}]}'
```

A continuación, importaría el archivo al almacén de anotaciones mediante el siguiente comando CLI.

```
aws omics start-annotation-import-job \
  --role-arn arn:aws:iam::555555555555:role/demoRole \
   --items=source=s3://amzn-s3-demo-bucket/
gnomad.genomes.v3.1.sites.chrM.reduced_annotations.tsv \
   --destination-name gnomadchrx \
   --format-options=tsvOptions='{readOptions={sep="\t",header=true,comment="#"}}'
```

El siguiente ejemplo muestra cómo un cliente puede crear un almacén de anotaciones para un archivo mim2gene. Un archivo mim2gene proporciona los vínculos entre los genes de OMIM y otro identificador de genes. Está delimitado por tabulaciones y contiene comentarios.

```
aws omics create-annotation-store \
  --name mim2gene \
  --store-format TSV \
  --reference=referenceArn=arn:aws:omics:us-
west-2:555555555555:referenceStore/6505293348/reference/2310864158 \
  --store-options=tsvStoreOptions='
    {annotationType=GENERIC,
    formatToHeader={},
    schema=[
        {mim_gene_id=STRING},
        {mim_type=STRING},
        {entrez_id=STRING},
        {hgnc=STRING},
        {ensembl=STRING}]}'
```

A continuación, puedes importar los datos a tu tienda de la siguiente manera.

```
aws omics start-annotation-import-job \
   --role-arn arn:aws:iam::555555555555:role/demoRole \
   --items=source=s3://xquek-dev-aws/annotation-examples/mim2gene.txt \
   --destination-name mim2gene \
```

```
   --format-options=tsvOptions='{readOptions={sep="\t",header=false,comment="#"}}'
```

Inicio de trabajos de importación con formato VCF

Para los archivos VCF, hay dos entradas adicionales que ignoran o incluyen esos parámetros`ignoreFilterField`, como se muestra. `ignoreQualField`

```
aws omics start-annotation-import-job --destination-name annotation_example\
  --role-arn arn:aws:iam::555555555555:role/demoRole \
  --items source=s3://demodata/example.garvan.vcf \
  --format-options '{ "vcfOptions": {
    "ignoreQualField": false,
    "ignoreFilterField": false
    }
   }'
```

También puede cancelar la importación de un almacén de anotaciones, como se muestra. Si la cancelación se realiza correctamente, no recibirá respuesta a esta AWS CLI llamada. Sin embargo, si no se encuentra el identificador del trabajo de importación o el trabajo de importación se ha completado, recibirá un mensaje de error.

```
aws omics cancel-annotation-import-job --job-id edd7b8ce-xmpl-47e2-bc99-258cac95a508
```

> **Note**
>
> Sus metadatos importan el historial de trabajos de get-annotation-import-job, get-variant-import-joblist-annotation-import-jobs, y list-variant-import-jobsse eliminan automáticamente al cabo de dos años. Los datos de variantes y anotaciones que se importan no se eliminan automáticamente y permanecen en los almacenes de datos.

Creación de HealthOmics versiones de almacenes de anotaciones

> **Important**
>
> AWS HealthOmics las tiendas de variantes y las tiendas de anotaciones ya no están abiertas a nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio

con normalidad. Para obtener más información, consulte AWS HealthOmics cambio en la disponibilidad del almacén de variantes y del almacén de anotaciones.

Puede crear nuevas versiones de los almacenes de anotaciones para recopilar diferentes versiones de sus bases de datos de anotaciones. Esto le ayuda a organizar los datos de las anotaciones, que se actualizan periódicamente.

Para crear una nueva versión de un almacén de anotaciones existente, usa la create-annotation-store-versionAPI como se muestra en el siguiente ejemplo.

```
aws omics create-annotation-store-version \
     --name my_annotation_store \
     --version-name my_version
```

Recibirás la siguiente respuesta con el ID de versión del almacén de anotaciones, lo que confirmará que se ha creado una nueva versión de tu anotación.

```
{
     "creationTime": "2023-07-21T17:15:49.251040+00:00",
     "id": "3b93cdef69d2",
     "name": "my_annotation_store",
     "reference": {
         "referenceArn": "arn:aws:omics:us-
west-2:555555555555:referenceStore/6505293348/reference/5987565360"
     },
     "status": "CREATING",
     "versionName": "my_version"
}
```

Para actualizar la descripción de una versión del almacén de anotaciones, puede utilizar esta opción update-annotation-store-versionpara añadir actualizaciones a una versión del almacén de anotaciones.

```
aws omics update-annotation-store-version \
    --name my_annotation_store \
    --version-name my_version \
    --description "New Description"
```

Recibirá la siguiente respuesta, que confirmará que la versión del almacén de anotaciones se ha actualizado.

```
{
     "storeId": "4934045d1c6d",
     "id": "2a3f4a44aa7b",
     "description":"New Description",
     "status": "ACTIVE",
     "name": "my_annotation_store",
     "versionName": "my_version",
     "creation Time": "2023-07-21T17:20:59.380043+00:00",
     "updateTime": "2023-07-21T17:26:17.892034+00:00"
}
```

Para ver los detalles de una versión del almacén de anotaciones, utilice. get-annotation-store-version

```
aws omics get-annotation-store-version --name my_annotation_store --version-name
 my_version
```

Recibirá una respuesta con el nombre, el estado y otros detalles de la versión.

```
{
     "storeId": "4934045d1c6d",
     "id": "2a3f4a44aa7b",
     "status": "ACTIVE",
     "versionArn": "arn:aws:omics:us-west-2:555555555555:annotationStore/
my_annotation_store/version/my_version",
     "name": "my_annotation_store",
     "versionName": "my_version",
     "creationTime": "2023-07-21T17:15:49.251040+00:00",
     "updateTime": "2023-07-21T17:15:56.434223+00:00",
     "statusMessage": "",
     "versionSizeBytes": 0
    }
```

Para ver todas las versiones de un almacén de anotaciones, puede utilizar lo siguiente list-annotation-store-versions, como se muestra en el siguiente ejemplo.

```
aws omics list-annotation-store-versions --name my_annotation_store
```

Recibirá una respuesta con la siguiente información

```
{
  "annotationStoreVersions": [
    {
     "storeId": "4934045d1c6d",
     "id": "2a3f4a44aa7b",
     "status": "CREATING",
     "versionArn": "arn:aws:omics:us-west-2:555555555555:annotationStore/
my_annotation_store/version/my_version_2",
     "name": "my_annotation_store",
     "versionName": "my_version_2",
     "creation Time": "2023-07-21T17:20:59.380043+00:00",
     "versionSizeBytes": 0
    },
    {
     "storeId": "4934045d1c6d",
     "id": "4934045d1c6d",
     "status": "ACTIVE",
     "versionArn": "arn:aws:omics:us-west-2:555555555555:annotationStore/
my_annotation_store/version/my_version_1",
     "name": "my_annotation_store",
     "versionName": "my_version_1",
     "creationTime": "2023-07-21T17:15:49.251040+00:00",
     "updateTime": "2023-07-21T17:15:56.434223+00:00",
     "statusMessage": "",
     "versionSizeBytes": 0
    }
}
```

Si ya no necesita una versión del almacén de anotaciones, puede utilizarla delete-annotation-store-versionspara eliminar una versión del almacén de anotaciones, como se muestra en el siguiente ejemplo.

```
aws omics delete-annotation-store-versions --name my_annotation_store --versions
 my_version
```

Si la versión de tienda se elimina sin errores, recibirá la siguiente respuesta.

```
{
  "errors": []
}
```

Si hay errores, recibirás una respuesta con los detalles de los errores, tal y como se muestra.

```
{
  "errors": [
    {
      "versionName": "my_version",
      "message": "Version with versionName: my_version was not found."
    }
  ]
}
```

Si intenta eliminar una versión del almacén de anotaciones que tiene un trabajo de importación activo, recibirá una respuesta con un error, como se muestra.

```
{
  "errors": [
    {
      "versionName": "my_version",
      "message": "version has an inflight import running"
    }
  ]
}
```

En este caso, puede forzar la eliminación de la versión del almacén de anotaciones, como se muestra en el siguiente ejemplo.

```
aws omics delete-annotation-store-versions --name my_annotation_store --versions
 my_version --force
```

Eliminar almacenes HealthOmics de análisis

> **Important**
>
> AWS HealthOmics Las tiendas de variantes y las tiendas de anotaciones ya no están abiertas a nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte AWS HealthOmics cambio en la disponibilidad del almacén de variantes y del almacén de anotaciones.

Al eliminar un almacén de variantes o anotaciones, el sistema también elimina todos los datos importados de ese almacén y cualquier etiqueta asociada.

El siguiente ejemplo muestra cómo eliminar un almacén de variantes mediante. AWS CLI Si la acción se realiza correctamente, el estado del almacén de variantes pasa a`DELETING`.

```
aws omics delete-variant-store --id <variant-store-id>
```

El siguiente ejemplo muestra cómo eliminar un almacén de anotaciones. Si la acción se realiza correctamente, el estado del almacén de anotaciones pasa a. `DELETING` Los almacenes de anotaciones no se pueden eliminar si existe más de una versión.

```
aws omics delete-annotation-store --id <annotation-store-id>
```

Consulta de datos HealthOmics analíticos

> **Important**
>
> AWS HealthOmics las tiendas de variantes y las tiendas de anotaciones ya no están abiertas a nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte AWS HealthOmics cambio en la disponibilidad del almacén de variantes y del almacén de anotaciones.

Puede realizar consultas en sus tiendas de variantes mediante AWS Lake Formation Amazon Athena o Amazon EMR. Antes de ejecutar cualquier consulta, complete los procedimientos de configuración (descritos en las siguientes secciones) para Lake Formation y Amazon Athena.

Para obtener información sobre Amazon EMR, consulte el tutorial: Introducción a Amazon EMR

En el caso de las tiendas de variantes creadas después del 26 de septiembre de 2024, HealthOmics divida la tienda por ID de muestra. Esta partición significa que HealthOmics utiliza el ID de muestra para optimizar el almacenamiento de la información de la variante. Las consultas que utilizan información de muestra como filtros devolverán los resultados más rápido, ya que la consulta escanea menos datos.

HealthOmics usa muestras IDs como nombres de archivos de partición. Antes de ingerir datos, compruebe si el ID de muestra contiene algún dato de PHI. Si es así, cambie la ID de la muestra

antes de ingerir los datos. Para obtener más información sobre qué contenido incluir y qué no incluir en la muestra IDs, consulte la guía en la página web sobre el cumplimiento de la AWS HIPAA.

Temas

- Configuración de Lake Formation para usar HealthOmics
- Configuración de Athena para consultas
- Ejecutar consultas en tiendas de HealthOmics variantes

Configuración de Lake Formation para usar HealthOmics

> **Important**
>
> AWS HealthOmics Las tiendas de variantes y las tiendas de anotaciones ya no están abiertas a nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte AWS HealthOmics cambio en la disponibilidad del almacén de variantes y del almacén de anotaciones.

Antes de usar Lake Formation para administrar HealthOmics los almacenes de datos, lleve a cabo los siguientes procedimientos de configuración de Lake Formation.

Temas

- Crear o verificar los administradores de Lake Formation
- Creación de enlaces de recursos mediante la consola de Lake Formation
- Configurar los permisos para AWS RAM los recursos compartidos

Crear o verificar los administradores de Lake Formation

Antes de poder crear un lago de datos en Lake Formation, debe definir uno o más administradores.

Los administradores son usuarios y roles con permisos para crear enlaces a recursos. Los administradores de los lagos de datos se configuran por cuenta y región.

Cree un usuario administrador en la consola de Lake Formation

1. Abra la consola AWS Lake Formation: consola Lake Formation
2. Si la consola muestra el panel Welcome to Lake Formation, seleccione Comenzar.

Lake Formation lo agrega a la tabla de administradores de Data Lake.

3. De lo contrario, en el menú de la izquierda, seleccione Funciones y tareas administrativas.
4. Añada los administradores adicionales que sean necesarios.

Creación de enlaces de recursos mediante la consola de Lake Formation

Para crear un recurso compartido que los usuarios puedan consultar, los controles de acceso predeterminados deben estar deshabilitados. Para obtener más información sobre cómo deshabilitar los controles de acceso predeterminados, consulte Cambiar la configuración de seguridad predeterminada de su lago de datos en la documentación de Lake Formation. Puede crear enlaces a recursos de forma individual o en grupo para poder acceder a los datos de Amazon Athena u otros AWS servicios (como Amazon EMR).

Crear enlaces a recursos en la consola de AWS Lake Formation y compartirlos con los usuarios de HealthOmics Analytics

1. Abra la consola AWS Lake Formation: consola Lake Formation
2. En la barra de navegación principal, selecciona Bases de datos.
3. En la tabla Bases de datos, seleccione la base de datos que desee.
4. En el menú Crear, seleccione el enlace de recursos.
5. Introduzca un nombre de enlace a un recurso. Si planea acceder a la base de datos desde Athena, introduzca un nombre utilizando solo letras minúsculas (hasta 256 caracteres).
6. Seleccione Crear.
7. El nuevo enlace al recurso ahora aparece en Bases de datos.

Conceda acceso al recurso compartido mediante la consola de Lake Formation

El administrador de la base de datos de Lake Formation puede conceder acceso al recurso compartido mediante el siguiente procedimiento.

1. Abre la consola de AWS Lake Formation: https://console.aws.amazon.com/lakeformation/
2. En la barra de navegación principal, selecciona Bases de datos.
3. En la página Bases de datos, seleccione el enlace al recurso que creó anteriormente.
4. En el menú Acciones, selecciona Otorgar al objetivo.

5. En la página Conceder permisos de datos, en la sección Principales, selecciona Usuarios o roles de IAM.
6. En el menú desplegable de usuarios o roles de IAM, busca el usuario al que quieres conceder acceso.
7. A continuación, en las etiquetas LF o en la tarjeta de recursos del catálogo, seleccione la opción Recursos del catálogo de datos con nombre asignado.
8. En el menú desplegable opcional de tablas, selecciona Todas las tablas o la tabla que creaste anteriormente.
9. En la tarjeta de permisos de tabla, en Permisos de tabla, elija Describir y Seleccionar.
10. A continuación, selecciona Otorgar.

Para ver los permisos de Lake Formation, elija los permisos de Data Lake en el panel de navegación principal. En la tabla se muestran las bases de datos y los enlaces a los recursos disponibles.

Configurar los permisos para AWS RAM los recursos compartidos

En la consola de AWS Lake Formation, seleccione los permisos de Data Lake en la barra de navegación principal para ver los permisos. En la página de permisos de datos, puede ver una tabla que muestra los tipos de recursos y las bases de datos y **ARN** qué está relacionado con un recurso compartido en RAM Resource Share. Si necesitas aceptar un recurso compartido AWS Resource Access Manager (AWS RAM), se lo AWS Lake Formation notificará en la consola.

HealthOmics puede aceptar implícitamente los AWS RAM recursos compartidos durante la creación de la tienda. Para aceptar el AWS RAM recurso compartido, el usuario o rol de IAM que invoca las operaciones `CreateVariantStore` o la `CreateAnnotationStore` API debe permitir las siguientes acciones:

- `ram:GetResourceShareInvitations`- Esta acción permite HealthOmics encontrar las invitaciones.
- `ram:AcceptResourceShareInvitation`- Esta acción HealthOmics permite aceptar la invitación mediante un token FAS.

Sin estos permisos, aparecerá un error de autorización durante la creación de la tienda.

Este es un ejemplo de política que incluye estas acciones. Agregue esta política al usuario o rol de IAM que acepta el AWS RAM recurso compartido.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "omics:*",
        "ram:AcceptResourceShareInvitation",
        "ram:GetResourceShareInvitations"
      ],
      "Resource": "*"
    }
  ]
}
```

Configuración de Athena para consultas

> **Important**
>
> AWS HealthOmics las tiendas de variantes y las tiendas de anotaciones ya no están abiertas a nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte AWS HealthOmics cambio en la disponibilidad del almacén de variantes y del almacén de anotaciones.

Puedes usar Athena para consultar variantes y anotaciones. Antes de ejecutar cualquier consulta, lleve a cabo las siguientes tareas de configuración:

Temas

- Configurar una ubicación de resultados de consulta mediante la consola Athena
- Configurar un grupo de trabajo con el motor Athena v3

Configurar una ubicación de resultados de consulta mediante la consola Athena

Para configurar una ubicación de resultados de consulta, siga estos pasos.

1. Abra la consola Athena: consola Athena
2. En la barra de navegación principal, selecciona el editor de consultas.
3. En el editor de consultas, selecciona la pestaña Configuración y, a continuación, selecciona Administrar.
4. Introduzca el prefijo S3 de una ubicación para guardar el resultado de la consulta.

Configurar un grupo de trabajo con el motor Athena v3

Para configurar un grupo de trabajo, siga estos pasos.

1. Abra la consola Athena: consola Athena
2. En la barra de navegación principal, elija Grupos de trabajo y, a continuación, Crear grupo de trabajo.
3. Introduzca un nombre para el grupo de trabajo.
4. Seleccione Athena SQL como tipo de motor.
5. En Actualizar el motor de consultas, seleccione Manual.
6. En Query version engine, selecciona Athena versión 3.
7. Elija Crear grupo de trabajo.

Ejecutar consultas en tiendas de HealthOmics variantes

> **Important**
>
> AWS HealthOmics las tiendas de variantes y las tiendas de anotaciones ya no están abiertas a nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte AWS HealthOmics cambio en la disponibilidad del almacén de variantes y del almacén de anotaciones.

Puedes realizar consultas en tu tienda de variantes con Amazon Athena. Tenga en cuenta que las coordenadas genómicas de los almacenes de variantes y anotaciones se representan como intervalos de base cero, semicerrados y semiabiertos.

Ejecute una consulta sencilla con la consola Athena

El siguiente ejemplo muestra cómo ejecutar una consulta sencilla.

1. Abra el editor de consultas de Athena: editor de consultas de Athena
2. En Grupo de trabajo, seleccione el grupo de trabajo que creó durante la configuración.
3. Compruebe que la fuente de datos sea. AwsDataCatalog
4. En Database, seleccione el enlace de recursos de base de datos que creó durante la configuración de Lake Formation.
5. Copie la siguiente consulta en el editor de consultas, en la pestaña Consulta 1:

```
SELECT * from omicsvariants limit 10
```

6. Seleccione Ejecutar para ejecutar la consulta. La consola rellena la tabla de resultados con las 10 primeras filas de la omicsvariants tabla.

Ejecute una consulta compleja con la consola de Athena

El siguiente ejemplo muestra cómo ejecutar una consulta compleja. Para ejecutar esta consulta, `ClinVar` impórtela al almacén de anotaciones.

Ejecute una consulta compleja

1. Abra el editor de consultas de Athena: editor de consultas de Athena
2. En Grupo de trabajo, seleccione el grupo de trabajo que creó durante la configuración.
3. Compruebe que la fuente de datos sea. AwsDataCatalog
4. En Database, seleccione el enlace de recursos de base de datos que creó durante la configuración de Lake Formation.
5. Seleccione la de + la parte superior derecha para crear una nueva pestaña de consulta denominada Consulta 2.
6. Copie la siguiente consulta en el editor de consultas, en la pestaña Consulta 2:

```
SELECT variants.sampleid,
  variants.contigname,
  variants.start,
  variants."end",
  variants.referenceallele,
  variants.alternatealleles,
  variants.attributes AS variant_attributes,
  clinvar.attributes AS clinvar_attributes
FROM omicsvariants as variants
```

```
INNER JOIN omicsannotations as clinvar ON
  variants.contigname=CONCAT('chr',clinvar.contigname)
  AND variants.start=clinvar.start
  AND variants."end"=clinvar."end"
  AND variants.referenceallele=clinvar.referenceallele
  AND variants.alternatealleles=clinvar.alternatealleles
WHERE clinvar.attributes['CLNSIG']='Likely_pathogenic'
```

7. Seleccione Ejecutar para empezar a ejecutar la consulta.

Compartir tiendas HealthOmics de análisis

> **Important**
>
> AWS HealthOmics Las tiendas de variantes y las tiendas de anotaciones ya no están abiertas a nuevos clientes. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte AWS HealthOmics cambio en la disponibilidad del almacén de variantes y del almacén de anotaciones.

Como propietario de un almacén de variantes o de un almacén de anotaciones, puede compartir el almacén con otras cuentas de AWS. El propietario puede revocar el acceso al recurso compartido eliminando el recurso compartido.

Como suscriptor de una tienda compartida, primero aceptas el recurso compartido. A continuación, puedes definir los flujos de trabajo que utilizan la tienda compartida. Los datos aparecen en forma de tabla tanto AWS Glue en Lake Formation como en Lake Formation.

Cuando ya no necesites acceder a la tienda, eliminas el recurso compartido.

Consulte Uso compartido de recursos entre cuentas en AWS HealthOmics para obtener información adicional sobre el uso compartido de recursos.

Crear un recurso compartido en la tienda

Para crear un recurso compartido de tienda, usa la operación de API create-share. El suscriptor principal es el Cuenta de AWS usuario que se suscribirá al recurso compartido. En el siguiente ejemplo, se crea un recurso compartido para una tienda de variantes. Para compartir una tienda con más de una cuenta, debes crear varios recursos compartidos de la misma tienda.

```
aws omics create-share  \
        --resource-arn "arn:aws:omics:us-west-2:555555555555:variantStore/
omics_dev_var_store" \
        --principal-subscriber "123456789012" \
        --name "my_Share-123"
```

Si la creación se realiza correctamente, recibirás una respuesta con el ID y el estado del recurso compartido.

```
{
       "shareId": "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a",
       "name": "my_Share-123",
       "status": "PENDING"
  }
```

El recurso compartido permanece en estado pendiente hasta que el suscriptor lo acepte mediante la operación de API accept-share.

Uso compartido de recursos entre cuentas en AWS HealthOmics

Utilice el uso compartido entre cuentas para compartir recursos con los colaboradores sin crear copias ni modificar las políticas de recursos de IAM. Los siguientes recursos permiten el uso compartido entre cuentas:

- HealthOmics tiendas de variantes
- HealthOmics almacenes de anotaciones
- Flujos de trabajo privados

Compartir un recurso incluye los siguientes pasos:

1. El propietario del recurso crea un recurso compartido y especifica el ARN del recurso y el Cuenta de AWS del suscriptor previsto. El recurso compartido permanece en estado pendiente hasta que el suscriptor lo acepte.
2. El suscriptor acepta el recurso compartido para acceder al recurso. El recurso compartido pasa al estado de activación.
3. El HealthOmics servicio proporciona a la cuenta del suscriptor acceso al recurso.
4. El propietario del recurso puede eliminar el recurso compartido o el suscriptor puede revocar su acceso al recurso compartido. El suscriptor no puede eliminar el recurso compartido ni el recurso asociado.

Temas

- Crear un recurso compartido
- Recupera información sobre un recurso compartido
- Consulta las acciones de las que eres propietario
- Consulta las acciones aceptadas de otras cuentas
- Eliminar un recurso compartido

Crear un recurso compartido

Puedes usar la operación de la API create-share para crear un recurso compartido. El suscriptor principal es Cuenta de AWS el usuario que se suscribirá al recurso compartido. En el siguiente ejemplo, se crea un recurso compartido para un almacén de variantes.

```
aws omics create-share \
    --resource-arn "arn:aws:omics:us-west-2:555555555555:variantStore/
omics_dev_var_store" \
    --principal-subscriber "123456789012"  \
    --name "my_Share-123"
```

Si la creación se realiza correctamente, recibirás una respuesta con el ID y el estado del recurso compartido.

```
{
"shareId": "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a",
     "name": "my_Share-123",
     "status": "PENDING"
}
```

El recurso compartido permanece en estado pendiente hasta que el suscriptor lo acepte mediante la operación de accept-share API.

```
 aws omics accept-share \
    --share-id "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a"
```

Una vez que el suscriptor acepta el recurso compartido, este pasa al estado activo.

```
{
"status": "ACTIVATING"
}
```

Recupera información sobre un recurso compartido

Utilice la operación de la API get-share para recuperar información sobre el recurso compartido.

```
aws omics get-share --share-id "495c21bedc889d07d0ab69d710a6841e-
dd75ab7a1a9c384fa848b5bd8e5a7e0a"
```

La respuesta de la API incluye información de metadatos sobre el recurso compartido.

```
{
  "share":
    {
      "shareId": "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a",
      "name": "my_Share-123",
      "resourceArn": "arn:aws:omics:us-west-2:555555555555:variantStore/
omics_dev_var_store",
      "principalSubscriber": "123456789012",
      "ownerId": "555555555555",
      "status": "PENDING"
    }
}
```

Consulta las acciones de las que eres propietario

Usa la API list-shares para recuperar información sobre cada una de las acciones que te pertenecen.

```
aws omics list-shares  --resource-owner SELF
```

La respuesta de la API incluye los metadatos de cada recurso compartido del que seas propietario.

Consulta las acciones aceptadas de otras cuentas

Usa la API list-shares para ver todas las acciones que has aceptado de otras cuentas.

```
aws omics list-shares  --resource-owner OTHER
```

La respuesta de la API incluye los metadatos de cada recurso compartido que hayas aceptado.

Eliminar un recurso compartido

Usa la API delete-share para eliminar un recurso compartido cuando ya no lo necesites.

```
aws omics delete-share \
    --share-id "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a"
```

Etiquetado de recursos en HealthOmics

Temas

- Aviso importante
- Recursos de etiquetado HealthOmics
- Secuencia: almacene, lea y establezca etiquetas
- Añadir una etiqueta a un HealthOmics recurso
- Listado de etiquetas de un recurso
- Eliminar etiquetas de un almacén de datos

Aviso importante

HealthOmics protege los datos de los clientes según las políticas del modelo de responsabilidad compartida de AWS. Esto significa que todos los datos de los clientes se cifran tanto en transición como en reposo. Sin embargo, no todos los nombres de recursos introducidos por el cliente, como los almacenes de datos o las operaciones basadas en tareas, están cifrados. Nunca deben contener información de identificación personal o información de salud protegida. Para obtener más información, consulte Seguridad en AWS HealthOmics.

Recursos de etiquetado HealthOmics

Puede asignar metadatos a sus recursos de AWS mediante etiquetas. Cada etiqueta es una marca que consta de una clave y un valor definidos por el usuario. Las etiquetas pueden ayudar a administrar, identificar, organizar, buscar y filtrar recursos de .

En este tema se describen las categorías y estrategias de etiquetado de uso común para ayudarle a implementar una estrategia de etiquetado coherente y eficaz. En las siguientes secciones se asume un conocimiento básico de los recursos de AWS, el etiquetado, la facturación detallada y AWS Identity and Access Management.

Cada etiqueta de tiene dos partes:

- Una clave de etiqueta (por ejemplo CostCenter, entorno o proyecto). Las claves de etiquetas distinguen entre mayúsculas y minúsculas.

- Un valor de etiqueta (por ejemplo, 111122223333 o Production). Al igual que las claves de etiquetas, los valores de las etiquetas distinguen mayúsculas de minúsculas.

Utilice etiquetas para clasificar los recursos según su finalidad, propietario, entorno u otro criterio. Para obtener más información, consulte Estrategias de etiquetado de AWS.

Puede añadir, cambiar o eliminar etiquetas de un recurso desde la consola de servicio del recurso, la API de servicio o la AWS CLI.

Para habilitar el etiquetado, asegúrate de que TagResources está autorizado. Puede autorizarlo TagResources adjuntando una política de IAM, como en el ejemplo siguiente.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "omics:Create*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "omics:Start*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "omics:Tag*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "omics:Untag*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "omics:List*",
            "Resource": "*"
```

```
        }
    ]
}
```

Prácticas recomendadas

Al crear una estrategia de etiquetado para los recursos de AWS, siga las prácticas recomendadas:

- No almacene información de identificación personal (PII), información de salud protegida (PHI) u otra información confidencial en etiquetas.
- Utilice un formato estandarizado que distinga mayúsculas de minúsculas para las etiquetas y aplíquelo de forma coherente a todos los tipos de recursos.
- Tenga en cuenta las directrices de etiquetas que admiten múltiples propósitos, como administrar el control de acceso a recursos, el seguimiento de costos, la automatización y la organización.
- Use herramientas automatizadas para ayudar a administrar las etiquetas de recursos. AWS Resource Groups y la API de etiquetado de Resource Groups permiten el control programático de las etiquetas, lo que permite administrar, buscar y filtrar etiquetas y recursos de forma automática.
- El etiquetado es más eficaz cuando se utilizan más etiquetas.
- Las etiquetas se pueden editar o modificar según las necesidades del usuario. Sin embargo, para actualizar las etiquetas de control de acceso, también debe actualizar las políticas que hacen referencia a esas etiquetas para controlar el acceso a sus recursos.

Requisitos de etiquetado

Las etiquetas tienen los siguientes requisitos:

- Las claves no pueden llevar el prefijo aws:.
- Las claves deben ser únicas dentro de un conjunto de etiquetas.
- Una clave debe tener entre 1 y 128 caracteres permitidos.
- Un valor debe tener entre 0 y 256 caracteres permitidos.
- No es necesario que los valores sean únicos por conjunto de etiquetas.
- Los caracteres permitidos para las claves y los valores son letras y dígitos Unicode, espacios en blanco y cualquiera de estos símbolos: _ . : / = + - @.
- Las claves y los valores distinguen entre mayúsculas y minúsculas.

Secuencia: almacene, lea y establezca etiquetas

En el caso de los almacenes de secuencias, las etiquetas creadas en el conjunto de lectura se ubican en el nivel de recursos del conjunto de lectura. Los conjuntos de lectura también contienen objetos debajo de ellos a los que se puede acceder, buscar y restringir mediante S3 APIs. De forma predeterminada, el ID de muestra (omics:SampleID) y el ID del sujeto (OMICS:SubjectID) se añaden al objeto.

Además, se pueden sincronizar hasta cinco etiquetas entre el conjunto de lectura y los objetos incluidos en él. La configuración de las etiquetas que se van a sincronizar es una configuración a nivel de tienda que se establece durante la creación o actualización de la tienda mediante el `propogatedSetLevelTags` parámetro.

Si ya hay datos en el almacén, la actualización de las claves puede llevar algún tiempo. Durante esta actualización, HealthOmics cambia el estado de la tienda aUpdating. Al finalizar, HealthOmics establece el estado de la tienda enActive. Mientras las etiquetas se propagan, es posible que no se apliquen los permisos que dependen de ellas. Los permisos se aplicarán una vez finalizada la propagación de la etiqueta.

Cuando las etiquetas se configuran o actualizan en el conjunto de lectura, el sistema decide si actualiza los objetos de ese conjunto de lectura, en función de la configuración de la tienda.

Añadir una etiqueta a un HealthOmics recurso

Añadir etiquetas a un recurso puede ayudarle a identificar y organizar sus recursos de AWS y a gestionar el acceso a ellos. En primer lugar, añada una o más etiquetas (pares clave-valor) a un recurso. Puedes usar hasta 50 etiquetas por recurso. También hay restricciones en cuanto a los caracteres que se pueden usar en los campos de clave y valor.

Tras añadir las etiquetas, puede crear políticas de IAM para gestionar el acceso al AWS recurso en función de estas etiquetas. Puede utilizar la HealthOmics consola o la AWS CLI para añadir etiquetas a un recurso. Añadir etiquetas a un repositorio puede afectar al acceso a dicho repositorio. Antes de añadir una etiqueta a un banco de datos, revise las políticas de IAM que puedan utilizar etiquetas para controlar el acceso a recursos como los almacenes de datos.

Las etiquetas de servicio se generan automáticamente tanto para un asunto como para un identificador de muestra para los almacenes de secuencias.

Siga estos pasos para usar el AWS CLI para agregar una etiqueta a un HealthOmics recurso. Por ejemplo, para añadir etiquetas a un almacén de secuencias mientras se está creando, utilizaría el siguiente comando en el AWS CLI. El nombre del almacén de secuencias es MySequenceStore, y las dos etiquetas añadidas con claves son clave1 y clave2 con valores como valor1 y valor2 respectivamente :

```
aws omics create-sequence-store --name "MySequenceStore" --tags key1=value1,key2=value2
```

La salida no muestra las etiquetas. Devuelve la siguiente respuesta.

```
{
    "id": "6860403586",
    "referenceStoreId": "4889894479",
    "roleArn": "arn:aws:iam::555555555555:role/ImportTest",
    "status": "CREATED",
    "creationTime": "2022-07-21T01:19:07.194Z"
}
```

Para añadir etiquetas a un recurso existente, ejecute el siguiente comando de ejemplo.

```
aws omics tag-resource --resource-arn arn:aws:omics:us-
west-2:555555555555:sequenceStore/2275234794 --tags key1=value1,key2=value2
```

Si se ejecuta correctamente, el comando no devuelve nada.

Listado de etiquetas de un recurso

Sigue estos pasos para utilizarlos AWS CLI para ver una lista de las AWS etiquetas de un HealthOmics recurso. Si no se han añadido etiquetas, la lista obtenida está vacía.

En la terminal o en la línea de comandos, ejecute el list-tags-for-resource comando como se muestra en el siguiente ejemplo.

```
aws omics list-tags-for-resource --resource-arn arn:aws:omics:us-
west-2:555555555555:sequenceStore/2275234794
```

En respuesta, recibirá una lista de etiquetas en formato JSON.

```
 {
    "tags": {
        "key1": "value1",
        "key2": "value2"
    }
}
```

Eliminar etiquetas de un almacén de datos

Puedes eliminar una o más etiquetas asociadas a un recurso. La eliminación de una etiqueta no la elimina de otros recursos de AWS que estén asociados a esa etiqueta.

En la terminal o en la línea de comandos, ejecute el comando untag-resource y especifique el nombre del recurso de Amazon (ARN) del recurso del que desea eliminar las etiquetas y la clave de etiqueta de la etiqueta que desea eliminar.

```
aws omics untag-resource --resource-arn arn:aws:omics:us-
west-2:555555555555:sequenceStore/2275234794 --tag-keys key1,key2
```

Si se ejecuta correctamente, este comando no devuelve ninguna respuesta. Para ver las etiquetas asociadas al recurso, ejecute el comando list-tags-for-resource.

Permisos de IAM para HealthOmics

Puedes usar AWS Identity and Access Management (IAM) para administrar el acceso a la HealthOmics API y a los recursos, como las tiendas y los flujos de trabajo. En el caso de los usuarios y las aplicaciones de su cuenta que la utilizan HealthOmics, usted administra los permisos mediante una política de permisos que puede aplicar a los usuarios, grupos o funciones de IAM.

Para administrar los permisos de los usuarios y las aplicaciones de sus cuentas, utilice las políticas que se HealthOmics proporcionan o redacte las suyas propias. La HealthOmics consola utiliza varios servicios para obtener información sobre la configuración y los activadores de la función. Puede utilizar las políticas proporcionadas tal cual o como punto de partida para políticas más restrictivas.

HealthOmics utiliza las funciones de servicio de IAM para acceder a otros servicios en su nombre. Por ejemplo, crearía o elegiría un rol de servicio al ejecutar un flujo de trabajo que lee datos de Amazon S3. Para algunas funciones, también debe configurar los permisos de los recursos de otros servicios. Revise estos requisitos antes de empezar a trabajar con HealthOmics

Para obtener más información acerca de IAM, consulte ¿Qué es IAM? en la Guía del usuario de IAM.

Temas

- Políticas de IAM basadas en identidad para HealthOmics
- Funciones de servicio para AWS HealthOmics
- Permisos de Amazon ECR
- HealthOmics Permisos de recursos
- Permisos de acceso a los datos mediante Amazon S3 URIs

Políticas de IAM basadas en identidad para HealthOmics

Para conceder acceso a los usuarios de tu cuenta HealthOmics, utilizas políticas basadas en la identidad AWS Identity and Access Management (IAM). Las políticas basadas en la identidad se pueden aplicar directamente a los usuarios de IAM o a los grupos y roles de IAM asociados a un usuario. También puede conceder a los usuarios de otra cuenta permiso para asumir un rol de su cuenta y tener acceso a sus recursos de HealthOmics.

Para conceder permiso a los usuarios para realizar acciones en una versión de flujo de trabajo, debe añadir el flujo de trabajo y la versión específica del flujo de trabajo a la lista de recursos.

La siguiente política de IAM permite a los usuarios acceder a todas las acciones de la HealthOmics API y transferirles las funciones de servicio. HealthOmics

Example Política de usuario

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "omics:*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "iam:PassRole"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": "omics.amazonaws.com"
        }
      }
    }
  ]
}
```

Cuando los utilizas HealthOmics, también interactúas con otros AWS servicios. Para acceder a estos servicios, utilice las políticas gestionadas que proporciona cada servicio. Para restringir el acceso a un subconjunto de recursos, puede usar las políticas administradas como punto de partida para crear sus propias políticas más restrictivas.

- AmazonS3 FullAccess: acceso a los buckets y objetos de Amazon S3 utilizados por los trabajos.

- Amazon EC2 ContainerRegistryFullAccess: acceso a los registros y repositorios de Amazon ECR para imágenes de contenedores de flujos de trabajo.
- AWSLakeFormationDataAdmin— Acceso a las bases de datos y tablas de Lake Formation creadas por las tiendas de análisis.
- ResourceGroupsandTagEditorFullAccess— Etiquete HealthOmics los recursos con operaciones de API de HealthOmics etiquetado.

Las políticas anteriores no permiten a los usuarios crear funciones de IAM. Para que un usuario con estos permisos pueda ejecutar un trabajo, un administrador debe crear el rol de servicio que otorga el HealthOmics permiso para acceder a las fuentes de datos. Para obtener más información, consulte Funciones de servicio para AWS HealthOmics.

Defina permisos de IAM personalizados para las ejecuciones

Puede incluir cualquier flujo de trabajo, ejecución o grupo de ejecución al que haga referencia la `StartRun` solicitud en una solicitud de autorización. Para ello, indique la combinación deseada de flujos de trabajo, ejecuciones o grupos de ejecuciones en la política de IAM. Por ejemplo, puede limitar el uso de un flujo de trabajo a una ejecución o grupo de ejecuciones específicos. También puedes especificar que un flujo de trabajo solo se use con un grupo de ejecuciones.

A continuación se muestra un ejemplo de política de IAM que permite un único flujo de trabajo con un único grupo de ejecuciones.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
      {
          "Effect": "Allow",
          "Action": [
              "omics:StartRun"
          ],
          "Resource": [
              "arn:aws:omics:us-west-2:123456789012:workflow/1234567",
              "arn:aws:omics:us-west-2:123456789012:runGroup/2345678"
```

```
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "omics:StartRun"
            ],
            "Resource": [
                "arn:aws:omics:us-west-2:123456789012:run/*",
                "arn:aws:omics:us-west-2:123456789012:runGroup/2345678"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "omics:GetRun",
                "omics:ListRunTasks",
                "omics:GetRunTask",
                "omics:CancelRun",
                "omics:DeleteRun"
            ],
            "Resource": [
                "arn:aws:omics:us-west-2:123456789012:run/*"
            ]
        }
    ]
}
```

Funciones de servicio para AWS HealthOmics

Un rol de servicio es un rol AWS Identity and Access Management (IAM) que otorga permisos para que un AWS servicio acceda a los recursos de su cuenta. Usted proporciona un rol de servicio AWS HealthOmics al iniciar un trabajo de importación o iniciar una ejecución.

La HealthOmics consola puede crear el rol necesario para usted. Si usa la HealthOmics API para administrar los recursos, cree el rol de servicio mediante la consola de IAM. Para obtener más información, consulte Crear un rol para delegar permisos a un Servicio de AWS.

Los roles de servicio deben tener la siguiente política de confianza.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "omics.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

La política de confianza permite que el HealthOmics servicio asuma la función.

Temas

- Ejemplo de políticas de servicio de IAM
- Ejemplo de CloudFormation plantilla

Ejemplo de políticas de servicio de IAM

En estos ejemplos, los nombres de los recursos y IDs las cuentas son marcadores de posición que debe reemplazar por valores reales.

En el siguiente ejemplo, se muestra la política de un rol de servicio que se puede usar para iniciar una ejecución. La política concede permisos para acceder a la ubicación de salida de Amazon S3, al grupo de registros del flujo de trabajo y al contenedor Amazon ECR para la ejecución.

> **Note**
>
> Si utiliza el almacenamiento en caché de llamadas para la ejecución, añada la ubicación de Amazon S3 de la caché de ejecución como recurso en los permisos de s3.

Example Política de roles de servicio para iniciar una ejecución

JSON

```
{
"Version":"2012-10-17",
"Statement": [
      {
          "Effect": "Allow",
          "Action": [
              "s3:GetObject",
              "s3:PutObject"
          ],
          "Resource": [
              "arn:aws:s3:::amzn-s3-demo-bucket1/*"
          ]
      },
      {
          "Effect": "Allow",
          "Action": [
              "s3:ListBucket"
          ],
          "Resource": [
              "arn:aws:s3:::amzn-s3-demo-bucket1"
          ]
      },
      {
          "Effect": "Allow",
          "Action": [
              "logs:DescribeLogStreams",
              "logs:CreateLogStream",
              "logs:PutLogEvents"
          ],
          "Resource": [
              "arn:aws:logs:us-east-1:123456789012:log-group:/aws/omics/
WorkflowLog:log-stream:*"
          ]
      },
      {
          "Effect": "Allow",
          "Action": [
              "logs:CreateLogGroup"
          ],
```

```
            "Resource": [
                "arn:aws:logs:us-east-1:123456789012:log-group:/aws/omics/
WorkflowLog:*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ecr:BatchGetImage",
                "ecr:GetDownloadUrlForLayer",
                "ecr:BatchCheckLayerAvailability"
            ],
            "Resource": [
                "arn:aws:ecr:us-east-1:123456789012:repository/*"
            ]
        }
    ]
}
```

En el siguiente ejemplo, se muestra la política de un rol de servicio que puedes usar para un trabajo de importación en una tienda. La política concede permisos para acceder a la ubicación de entrada de Amazon S3.

Example Función de servicio para el trabajo de la tienda de referencia

JSON

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

        {
```

```
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketLocation"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket"
            ]
        }

    ]
}
```

Ejemplo de CloudFormation plantilla

La siguiente CloudFormation plantilla de ejemplo crea un rol de servicio que da HealthOmics permiso para acceder a los buckets de Amazon S3 que tienen nombres con el prefijo y para cargar `omics-` registros de flujo de trabajo.

Example Permisos de Reference Store, Amazon S3 y CloudWatch Logs

```
Parameters:
  bucketName:
    Description: Bucket name
    Type: String

Resources:
  serviceRole:
    Type: AWS::IAM::Role
    Properties:
      Policies:
        - PolicyName: read-reference
          PolicyDocument:
            Version: 2012-10-17
            Statement:
            - Effect: Allow
              Action:
                - omics:*
              Resource: !Sub arn:${AWS::Partition}:omics:${AWS::Region}:
${AWS::AccountId}:referenceStore/*
        - PolicyName: read-s3
          PolicyDocument:
            Version: 2012-10-17
```

```
            Statement:
            - Effect: Allow
              Action:
                - s3:ListBucket
              Resource: !Sub arn:${AWS::Partition}:s3:::${bucketName}
            - Effect: Allow
              Action:
                - s3:GetObject
                - s3:PutObject
              Resource: !Sub arn:${AWS::Partition}:s3:::${bucketName}/*
        - PolicyName: upload-logs
          PolicyDocument:
            Version: 2012-10-17
            Statement:
            - Effect: Allow
              Action:
                - logs:DescribeLogStreams
                - logs:CreateLogStream
                - logs:PutLogEvents
              Resource: !Sub arn:${AWS::Partition}:logs:${AWS::Region}:
${AWS::AccountId}:loggroup:/aws/omics/WorkflowLog:log-stream:*
            - Effect: Allow
              Action:
                - logs:CreateLogGroup
              Resource: !Sub arn:${AWS::Partition}:logs:${AWS::Region}:
${AWS::AccountId}:loggroup:/aws/omics/WorkflowLog:*
      AssumeRolePolicyDocument: |
        {
          "Version": "2012-10-17",
          "Statement": [
            {
              "Action": [
                "sts:AssumeRole"
              ],
              "Effect": "Allow",
              "Principal": {
                "Service": [
                  "omics.amazonaws.com"
                ]
              }
            }
          ]
        }
```

Permisos de Amazon ECR

Antes de que el HealthOmics servicio pueda ejecutar un flujo de trabajo en un contenedor desde su repositorio privado de Amazon ECR, debe crear una política de recursos para el repositorio. La política otorga permiso al HealthOmics servicio para usar el contenedor. Agrega esta política de recursos a cada repositorio privado al que haga referencia el flujo de trabajo.

> **Note**
>
> El repositorio privado y el flujo de trabajo deben estar en la misma región.

Si el flujo de trabajo y el repositorio son propiedad de AWS cuentas diferentes, debes configurar los permisos entre cuentas.

No necesitas conceder acceso adicional al repositorio para los flujos de trabajo compartidos. Sin embargo, puedes crear políticas que permitan o denieguen el acceso de flujos de trabajo específicos a la imagen del contenedor.

Para utilizar la función de caché de extracción de Amazon ECR, debe crear una política de permisos de registro.

En las siguientes secciones se describe cómo configurar los permisos de recursos de Amazon ECR para estos escenarios. Para obtener más información sobre los permisos en Amazon ECR, consulte Permisos de registro privado en Amazon ECR.

Temas

- Cree una política de recursos para el repositorio de Amazon ECR
- Ejecutar flujos de trabajo con contenedores multicuenta
- Políticas de Amazon ECR para flujos de trabajo compartidos
- Las políticas de Amazon ECR extraen la memoria caché

Cree una política de recursos para el repositorio de Amazon ECR

Cree una política de recursos que permita al HealthOmics servicio ejecutar un flujo de trabajo utilizando un contenedor del repositorio. La política concede permiso al director del HealthOmics servicio para acceder a las acciones de Amazon ECR requeridas.

Siga estos pasos para crear la política:

1. Abra la página de repositorios privados en la consola de Amazon ECR y seleccione el repositorio al que va a conceder acceso.
2. En la barra de navegación lateral, seleccione Permisos.
3. Elija Edit (Edición de).
4. Selecciona Editar política JSON.
5. Añada la siguiente declaración de política y, a continuación, seleccione Guardar.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "omics workflow access",
            "Effect": "Allow",
            "Principal": {
                "Service": "omics.amazonaws.com"
            },
            "Action": [
                "ecr:GetDownloadUrlForLayer",
                "ecr:BatchGetImage",
                "ecr:BatchCheckLayerAvailability"
            ],
            "Resource": "*"
        }
    ]
}
```

Ejecutar flujos de trabajo con contenedores multicuenta

Si el flujo de trabajo y el contenedor son propiedad de diferentes AWS cuentas, debes configurar los siguientes permisos multicuenta:

1. Actualice la política de Amazon ECR del repositorio para conceder permiso de forma explícita a la cuenta propietaria del flujo de trabajo.

2. Actualice la función de servicio de la cuenta propietaria del flujo de trabajo para concederle acceso a la imagen del contenedor.

El siguiente ejemplo muestra una política de recursos de Amazon ECR que concede acceso a la cuenta propietaria del flujo de trabajo.

En este ejemplo:

- ID de cuenta de flujo de trabajo: 111122223333
- ID de cuenta del repositorio de contenedores: 444455556666
- Nombre del contenedor: samtools

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "omics.amazonaws.com"
            },
            "Action": [
                "ecr:BatchCheckLayerAvailability",
                "ecr:BatchGetImage",
                "ecr:GetDownloadUrlForLayer"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowAccessToTheServiceRoleOfTheAccountThatOwnsTheWorkflow",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:role/DemoCustomer"
            },
            "Action": [
                "ecr:BatchCheckLayerAvailability",
                "ecr:BatchGetImage",
                "ecr:GetDownloadUrlForLayer"
            ],
```

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

Para completar la configuración, agrega la siguiente declaración de política al rol de servicio de la cuenta propietaria del flujo de trabajo. La política concede permiso al rol de servicio para acceder a la imagen del contenedor «samtools». Asegúrese de reemplazar los números de cuenta, el nombre del contenedor y la región por sus propios valores.

```
{
    "Sid": "CrossAccountEcrRepoPolicy",
    "Effect": "Allow",
    "Action": ["ecr:BatchCheckLayerAvailability", "ecr:BatchGetImage",
 "ecr:GetDownloadUrlForLayer"],
    "Resource": "arn:aws:ecr:us-west-2:444455556666:repository/samtools"
}
```

Políticas de Amazon ECR para flujos de trabajo compartidos

> **Note**
>
> HealthOmics permite automáticamente que un flujo de trabajo compartido acceda al repositorio de Amazon ECR en la cuenta del propietario del flujo de trabajo, mientras el flujo de trabajo se ejecuta en la cuenta del suscriptor. No es necesario que concedas acceso adicional al repositorio para los flujos de trabajo compartidos. Para obtener más información, consulta Compartir HealthOmics flujos de trabajo.

De forma predeterminada, el suscriptor no tiene acceso al repositorio de Amazon ECR para usar los contenedores subyacentes. Si lo desea, puede personalizar el acceso al repositorio de Amazon ECR añadiendo claves de condición a la política de recursos del repositorio. En las siguientes secciones se proporcionan ejemplos de políticas.

Restrinja el acceso a flujos de trabajo específicos

Puede enumerar los flujos de trabajo individuales en una declaración de condición, de modo que solo estos flujos de trabajo puedan utilizar los contenedores del repositorio. La clave de

SourceArncondición especifica el ARN del flujo de trabajo compartido. El siguiente ejemplo concede permiso al flujo de trabajo especificado para usar este repositorio.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
     {
       "Sid": "OmicsAccessPrincipal",
       "Effect": "Allow",
       "Principal": {
         "Service": "omics.amazonaws.com"
       },
       "Action": [
         "ecr:GetDownloadUrlForLayer",
         "ecr:BatchGetImage",
         "ecr:BatchCheckLayerAvailability"
       ],
       "Resource": "*",
       "Condition": {
          "StringEquals": {
             "aws:SourceArn": "arn:aws:omics:us-
east-1:111122223333:workflow/1234567"
          }
       }
     }
  ]
}
```

Restrinja el acceso a cuentas específicas

Puede enumerar las cuentas de los suscriptores en una declaración de condiciones, de modo que solo estas cuentas tengan permiso para usar los contenedores del repositorio. La clave de SourceAccountcondición especifica la Cuenta de AWS del suscriptor. El siguiente ejemplo concede permiso a la cuenta especificada para usar este repositorio.

JSON

```
{
```

```
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "OmicsAccessPrincipal",
      "Effect": "Allow",
      "Principal": {
        "Service": "omics.amazonaws.com"
      },
      "Action": [
        "ecr:GetDownloadUrlForLayer",
        "ecr:BatchGetImage",
        "ecr:BatchCheckLayerAvailability"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "111122223333"
        }
      }
    }
  ]
}
```

También puede denegar los permisos de Amazon ECR a suscriptores específicos, como se muestra en el siguiente ejemplo de política.

JSON

```
{
      "Version":"2012-10-17",
      "Statement": [
          {
              "Sid": "OmicsAccessPrincipal",
              "Effect": "Allow",
              "Principal": {
                  "Service": "omics.amazonaws.com"
              },
              "Action": [
                  "ecr:GetDownloadUrlForLayer",
                  "ecr:BatchGetImage",
                  "ecr:BatchCheckLayerAvailability"
```

```
            ],
            "Resource": "*",
            "Condition": {
              "StringNotEquals": {
                "aws:SourceAccount": "111122223333"
              }
            }
        }
    ]
}
```

Las políticas de Amazon ECR extraen la memoria caché

Para utilizar la caché de extracción de Amazon ECR, debe crear una política de permisos de registro. También crea una plantilla de creación de repositorios, que define los permisos para los repositorios creados por la caché de extracción de Amazon ECR.

Las siguientes secciones incluyen ejemplos de estas políticas. Para obtener más información sobre la extracción de caché, consulte Sincronizar un registro ascendente con un registro privado de Amazon ECR en la Guía del usuario de Amazon Elastic Container Registry.

Política de permisos de registro

Para utilizar la caché de extracción de Amazon ECR, cree una política de permisos de registro. La política de permisos de registro permite controlar los permisos de replicación y extracción de la memoria caché.

Para la replicación entre cuentas, debes permitir de forma explícita Cuenta de AWS que cada una de ellas pueda replicar sus repositorios en tu registro.

De forma predeterminada, al crear una regla de extracción de caché, cualquier responsable de IAM que tenga permiso para extraer imágenes de un registro privado también puede usar la regla de extracción de memoria caché. Puedes usar los permisos de registro para restringir aún más estos permisos a repositorios específicos.

Agrega una política de permisos de registro a la cuenta propietaria de la imagen del contenedor.

En el siguiente ejemplo, la política permite al HealthOmics servicio crear repositorios para cada registro ascendente e iniciar solicitudes de extracción ascendentes desde los repositorios creados.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPTCinRegPermissions",
            "Effect": "Allow",
            "Principal": {
                "Service": "omics.amazonaws.com"
            },
            "Action": [
                "ecr:CreateRepository",
                "ecr:BatchImportUpstreamImage"
            ],
            "Resource": [
                "arn:aws:ecr:us-east-1:123456789012:repository/ecr-public/*",
                "arn:aws:ecr:us-east-1:123456789012:repository/docker-hub/*"
            ]
        }
    ]
}
```

Plantilla de creación de repositorios

Para utilizar pull through cache in HealthOmics, el repositorio de Amazon ECR debe tener una plantilla de creación de repositorios. La plantilla define los ajustes de configuración de los repositorios privados creados para un registro anterior.

Cada plantilla contiene un prefijo de espacio de nombres de repositorio, que Amazon ECR utiliza para hacer coincidir los nuevos repositorios con una plantilla específica. Las plantillas pueden especificar la configuración de todos los ajustes del repositorio, incluidas las políticas de acceso basadas en los recursos, la inmutabilidad de las etiquetas, el cifrado y las políticas de ciclo de vida. Para obtener más información, consulte las plantillas de creación de repositorios en la Guía del usuario de Amazon Elastic Container Registry.

En el siguiente ejemplo, la política permite al HealthOmics servicio iniciar solicitudes de extracción ascendentes desde los repositorios ascendentes.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "PTCRepoCreationTemplate",
            "Effect": "Allow",
            "Principal": {
                "Service": "omics.amazonaws.com"
            },
            "Action": [
                "ecr:BatchGetImage",
                "ecr:GetDownloadUrlForLayer"
            ],
            "Resource": "*"
        }
    ]
}
```

Políticas de acceso multicuenta a Amazon ECR

Para el acceso multicuenta, el propietario del repositorio privado actualiza la política de permisos de registro y la plantilla de creación del repositorio para permitir el acceso de la otra cuenta y de la función de ejecución de esa cuenta.

En la política de permisos de registro, añade una declaración de política para permitir que la función de ejecución de la otra cuenta acceda a las acciones de Amazon ECR:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowCrossAccountPTCinRegPermissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::123456789012:role/RUN_ROLE"},
            "Action": [
```

```
                "ecr:CreateRepository",
                "ecr:BatchGetImage",
                "ecr:BatchImportUpstreamImage"
            ],
            "Resource": "arn:aws:ecr:us-east-1:123456789012:repository/path/*"
        }
    ]
}
```

En la plantilla de creación del repositorio, añade una declaración de política para permitir que el rol de ejecución de la otra cuenta acceda a las nuevas imágenes del contenedor. Si lo desea, puede añadir declaraciones de condiciones para limitar el acceso a flujos de trabajo específicos:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowCrossAccountPTCinRepoCreationTemplate",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:role/RUN_ROLE"},
            "Action": [
                "ecr:BatchGetImage",
                "ecr:GetDownloadUrlForLayer"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceArn": "arn:aws:omics:us-
east-1:444455556666:workflow/WORKFLOW_ID",
                    "aws:SourceAccount": "111122223333"
                }
            }
        }
    ]
}
```

Agregue permisos para dos acciones adicionales (CreateRepository y BatchImportUpstreamImage) en la función de ejecución y especifique el recurso al que puede acceder la función de ejecución.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "CrossAccountPTCRunRolePolicy",
            "Effect": "Allow",
            "Action": [
                "ecr:CreateRepository",
                "ecr:BatchImportUpstreamImage",
                "ecr:BatchCheckLayerAvailability",
                "ecr:BatchGetImage",
                "ecr:GetDownloadUrlForLayer",
                "ecr:BatchGetImage"
            ],
            "Resource": "arn:aws:ecr:us-east-1:123456789012::repository/{path}/*"
        }
    ]
}
```

HealthOmics Permisos de recursos

AWS HealthOmics crea recursos en otros servicios y accede a ellos en tu nombre cuando ejecutas un trabajo o creas una tienda. En algunos casos, es necesario configurar los permisos en otros servicios para acceder a los recursos o permitir el acceso HealthOmics a ellos.

Para obtener información sobre los permisos de recursos relacionados con Amazon ECR, consultePermisos de Amazon ECR.

Permisos de Lake Formation

Antes de utilizar las funciones de análisis HealthOmics, configure los ajustes predeterminados de la base de datos en Lake Formation.

Para configurar los permisos de recursos en Lake Formation

1. Abra la página de configuración del catálogo de datos en la consola de Lake Formation.
2. Desactive los requisitos de control de acceso de IAM para bases de datos y tablas en Permisos predeterminados para bases de datos y tablas recién creadas.
3. Seleccione Save.

HealthOmics Analytics acepta datos automáticamente si su política de servicio tiene los permisos de RAM correctos, como en el siguiente ejemplo.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "omics:*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ram:AcceptResourceShareInvitation",
        "ram:GetResourceShareInvitations"
      ],
      "Resource": "*"
    }
  ]
}
```

Permisos de acceso a los datos mediante Amazon S3 URIs

Puede acceder a los datos del almacén de secuencias mediante operaciones de HealthOmics API o operaciones de API de Amazon S3.

Para el acceso a la HealthOmics API, HealthOmics los permisos se administran mediante una política de IAM. Sin embargo, el acceso a S3 requiere dos niveles de configuración: un permiso explícito en la política de acceso a S3 de la tienda y una política de IAM. Para obtener más información sobre el uso de las políticas de IAM con HealthOmics, consulte Funciones de servicio para. HealthOmics

Existen tres formas de compartir la capacidad de leer objetos mediante Amazon S3 APIs:

1. Uso compartido basado en políticas: este intercambio requiere habilitar el principio de IAM tanto en la política de acceso a S3 como redactar una política de IAM y adjuntarla al principio de IAM. Consulte el siguiente tema para obtener más información.
2. Prefirmado URLs : también puedes generar una URL prefirmada que se pueda compartir para un archivo del almacén de secuencias. Para obtener más información sobre la creación de prefirmados URLs con Amazon S3, consulte Uso de prefirmados URLs en la documentación de Amazon S3. La política de acceso a S3 al almacén de secuencias admite declaraciones para limitar las capacidades de las URL prefirmadas.
3. Funciones asumidas: cree una función en la cuenta del propietario de los datos que tenga una política de acceso que permita a los usuarios asumir esa función.

Temas

- Uso compartido basado en políticas
- Ejemplo de restricción

Uso compartido basado en políticas

Si accede a los datos del almacén secuencial mediante un URI directo de S3, HealthOmics proporciona medidas de seguridad mejoradas para la política de acceso al bucket de S3 asociada.

Las siguientes reglas se aplican a las nuevas políticas de acceso a S3. En el caso de las políticas existentes, las reglas se aplicarán la próxima vez que se actualice la política:

- Las políticas de acceso de S3 admiten los siguientes elementos de política
  - Versión, ID, declaración, razón, efecto, principio, acción, recurso, condición
- Las políticas de acceso de S3 admiten las siguientes claves de condición:
  - s3:ExistingObjectTag/<key>, s3: prefijo, s3: versión de firma, s3: TlsVersion

- Las políticas también admiten aws: con los siguientes operadores de condición: PrincipalArn y ArnEquals ArnLike

Si intenta añadir o actualizar una política para incluir un elemento o una condición no admitidos, el sistema rechazará la solicitud.

Temas

- Política de acceso a S3 predeterminada
- Personalización de la política de acceso
- Política de IAM
- Control de acceso basado en etiquetas

Política de acceso a S3 predeterminada

Al crear un almacén de secuencias, HealthOmics crea una política de acceso a S3 predeterminada que concede a la cuenta raíz del propietario del almacén de datos los siguientes permisos para todos los objetos accesibles del almacén de secuencias: S3: GetObjectGetObjectTagging, S3 y S3:ListBucket. La política que se crea de forma predeterminada es:

JSON

```
{
    "Version":"2012-10-17",
    "Statement":
    [
        {
            "Effect": "Allow",
            "Principal":
            {
                "AWS": "arn:aws:iam::111111111111:root"
            },
            "Action":
            [
                "s3:GetObject",
                "s3:GetObjectTagging"
            ],
```

```
            "Resource": "arn:aws:s3:us-
west-2:222222222222:accesspoint/111111111111-1234567890/object/111111111111/
sequenceStore/1234567890/*"
        },
        {
            "Effect": "Allow",
            "Principal":
            {
                "AWS": "arn:aws:iam::111111111111:root"
            },
            "Action": "s3:ListBucket",
            "Resource": "arn:aws:s3:us-
west-2:222222222222:accesspoint/111111111111-1234567890/111111111111/
sequenceStore/1234567890/*"
        }
    ]
}
```

Personalización de la política de acceso

Si la política de acceso a S3 está en blanco, no se permite el acceso a S3. Si hay una política vigente y necesitas eliminar el acceso a S3, úsala `deleteS3AccessPolicy` para eliminar todos los accesos.

Para añadir restricciones al uso compartido o conceder acceso a otras cuentas, puedes actualizar la política mediante la `PutS3AccessPolicy` API. Las actualizaciones de la política no pueden ir más allá del prefijo del almacén de secuencias ni de las acciones especificadas.

Política de IAM

Para permitir a un usuario o al principal de IAM el acceso mediante Amazon S3 APIs, además del permiso de la política de acceso de S3, se debe crear una política de IAM y adjuntarla al principal para conceder el acceso. Se puede aplicar una política que permita el acceso a la API de Amazon S3 a nivel de almacén de secuencias o a nivel de conjunto de lectura. En el nivel del conjunto de lecturas, los permisos se pueden restringir mediante el prefijo o mediante filtros de etiquetas de recursos para patrones de identificación de muestras o sujetos.

Si el almacén de secuencias utiliza una clave gestionada por el cliente (CMK), el responsable también debe tener derechos para utilizar la clave KMS para el descifrado. Para obtener más

información, consulte el acceso a KMS entre cuentas en la AWS Key Management Service Guía para desarrolladores.

El siguiente ejemplo proporciona a un usuario acceso a un almacén de secuencias. Puede ajustar el acceso con condiciones adicionales o filtros basados en los recursos.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111111111111:root"
      },
      "Action":
      [
         "s3:GetObject",
         "s3:GetObjectTagging"
      ],
      "Resource": "arn:aws:s3:us-
west-2:222222222222:accesspoint/111111111111-1234567890/object/111111111111/
sequenceStore/1234567890/*",
      "Condition": {
        "StringEquals": {
          "s3:ExistingObjectTag/omics:readSetStatus": "ACTIVE"
        }
      }
    },
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111111111111:root"
      },
      "Action": "s3:ListBucket",
      "Resource": "arn:aws:s3:us-
west-2:222222222222:accesspoint/111111111111-1234567890",
      "Condition": {
        "StringLike": {
          "s3:prefix": "111111111111/sequenceStore/1234567890/*"
        }
      }
```

```
    }
  ]
}
```

Control de acceso basado en etiquetas

Para utilizar el control de acceso basado en etiquetas, primero se debe actualizar el almacén de secuencias para propagar las claves de etiquetas que se utilizarán. Esta configuración se establece durante la creación o actualización del almacén de secuencias. Una vez que las etiquetas se han propagado, las condiciones de las etiquetas se pueden utilizar para añadir más restricciones. Las restricciones se pueden incluir en la política de acceso de S3 o en la política de IAM. El siguiente es un ejemplo de una política de acceso a S3 basada en pestañas que se podría establecer:

```
{
    "Sid": "tagRestrictedGets",
    "Effect": "Allow",
    "Principal":
    {
        "AWS": "arn:aws:iam::<target_restricted_account_id>:root"
    },
    "Action":
    [
        "s3:GetObject",
        "s3:GetObjectTagging"
    ],
    "Resource": "arn:aws:s3:us-west-2:222222222222:accesspoint/111111111111-1234567890/
object/111111111111/sequenceStore/1234567890/*",
    "Condition":
    {
        "StringEquals":
        {
            "s3:ExistingObjectTag/tagKey1": "tagValue1",
            "s3:ExistingObjectTag/tagKey2": "tagValue2"
        }
    }
}
```

Ejemplo de restricción

Escenario: crear un recurso compartido en el que el propietario de los datos pueda restringir la capacidad del usuario de descargar los datos «retirados».

En este escenario, el propietario de los datos (cuenta #111111111111) administraba un almacén de datos. Este propietario de los datos comparte los datos con una amplia gama de usuarios externos, incluido un investigador (cuenta #999999999999). Como parte de la gestión de los datos, el propietario de los datos recibe periódicamente solicitudes para retirar los datos de un participante. Para gestionar esta retirada, el propietario de los datos restringe primero el acceso a la descarga directa al recibir la solicitud y, finalmente, borra los datos según sus necesidades.

Para satisfacer esta necesidad, el propietario de los datos crea un almacén secuencial y cada conjunto de lecturas recibe una etiqueta que indica «estado», que pasará a ser «retirado» si se recibe la solicitud de retirada. En el caso de los datos con la etiqueta establecida en este valor, quieren asegurarse de que ningún usuario pueda ejecutar «getObject» en este archivo. Para realizar esta configuración, el propietario de los datos deberá asegurarse de seguir dos pasos.

Paso 1. En el caso del almacén de secuencias, asegúrese de que la etiqueta de estado esté actualizada para poder propagarse. Esto se hace añadiendo la clave de «estado» en el campo `propogatedSetLevelTags` al llamar `createSequenceStore` o `updateSequenceStore.`

Paso 2. Actualiza la política de acceso s3 de la tienda para restringir GetObject a los objetos con la etiqueta de estado configurada como retirada. Esto se hace actualizando la política de acceso a la tienda mediante la `PutS3AccesPolicy` API. La siguiente política permitiría a los clientes seguir viendo los archivos retirados al publicar objetos, pero les impediría acceder a ellos:

- Declaración 1 (restrictedGetWithdrawal): La cuenta 7139999 no puede recuperar los objetos retirados.
- Declaración 2 (ownerGetAll): La cuenta 1111, propietaria de los datos, puede recuperar todos los objetos, incluidos los objetos retirados.
- Declaración 3 (everyoneListAll): Todas las cuentas compartidas (1111 y 9999) pueden ejecutar la ListBucket operación con todo el prefijo.

JSON

```
{
    "Version":"2012-10-17",
    "Statement":
    [
        {
            "Sid": "restrictedGetWithdrawal",
            "Effect": "Allow",
```

```
            "Principal":
            {
                "AWS": "arn:aws:iam::999999999999:root"
            },
            "Action":
            [
                "s3:GetObject",
                "s3:GetObjectTagging"
            ],
            "Resource": "arn:aws:s3:us-
west-2:222222222222:accesspoint/111111111111-1234567890/object/111111111111/
sequenceStore/1234567890/*",
            "Condition":
            {
                "StringNotEquals":
                {
                    "s3:ExistingObjectTag/status": "withdrawn"
                }
            }
        },
        {
            "Sid": "ownerGetAll",
            "Effect": "Allow",
            "Principal":
            {
                "AWS": "arn:aws:iam::111111111111:root"
            },
            "Action":
            [
                "s3:GetObject",
                "s3:GetObjectTagging"
            ],
            "Resource": "arn:aws:s3:us-
west-2:222222222222:accesspoint/111111111111-1234567890/object/111111111111/
sequenceStore/1234567890/*",
            "Condition":
            {
                "StringEquals":
                {
                    "s3:ExistingObjectTag/omics:readSetStatus": "ACTIVE"
                }
            }
        },
        {
```

```
            "Sid": "everyoneListAll",
            "Effect": "Allow",
            "Principal":
            {
                "AWS": [
                    "arn:aws:iam::111111111111:root",
                    "arn:aws:iam::999999999999:root"
                ]
            },
            "Action": "s3:ListBucket",
            "Resource": "arn:aws:s3:us-
west-2:222222222222:accesspoint/111111111111-1234567890",
            "Condition":
            {
                "StringLike":
                {
                    "s3:prefix": "111111111111/sequenceStore/1234567890/*"
                }
            }
        }
    ]
}
```

Seguridad en AWS HealthOmics

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 AWS usted 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 conformidad que se aplican a AWS HealthOmics, consulte AWS Servicios dentro del alcance por programa de conformidad AWS Servicios incluidos .
- Seguridad en la nube: su responsabilidad viene determinada por el AWS servicio que utilice. También es responsable de otros factores, incluida la confidencialidad de los datos, los requisitos de la empresa y la legislación y la normativa aplicables.

Esta documentación le ayuda a entender cómo aplicar el modelo de responsabilidad compartida al utilizar AWS HealthOmics. En los temas siguientes, se muestra cómo configurar AWS HealthOmics para cumplir sus objetivos de seguridad y conformidad. También aprenderá a utilizar otros AWS servicios que le ayudan a supervisar y proteger sus HealthOmics recursos de AWS.

Temas

- Protección de datos en AWS HealthOmics
- Gestión de identidad y acceso en HealthOmics
- Validación de conformidad para AWS HealthOmics
- Resiliencia en HealthOmics
- AWS HealthOmics y puntos finales de VPC de interfaz ()AWS PrivateLink

Protección de datos en AWS HealthOmics

El AWS modelo de se aplica a protección de datos en AWS HealthOmics. Como se describe en este modelo, AWS es responsable de proteger la infraestructura global en la que se ejecutan

todos los Nube de AWS. Eres responsable de mantener el control sobre el contenido alojado en esta infraestructura. También eres responsable de las tareas de administración y configuración de seguridad para los Servicios de AWS que utiliza. Para obtener más información sobre la privacidad de los datos, consulte las Preguntas frecuentes sobre la privacidad de datos. Para obtener información sobre la protección de datos en Europa, consulte la publicación de blog sobre el Modelo de responsabilidad compartida de AWS y GDPR en el Blog de seguridad de AWS .

Con fines de protección de datos, le recomendamos que proteja Cuenta de AWS las credenciales y configure los usuarios individuales con AWS IAM Identity Center o AWS Identity and Access Management (IAM). De esta manera, solo se otorgan a cada usuario los permisos necesarios para cumplir sus obligaciones laborales. También recomendamos proteger sus datos de la siguiente manera:

- Utiliza la autenticación multifactor (MFA) en cada cuenta.
- Se utiliza SSL/TLS para comunicarse con AWS los recursos. Exigimos TLS 1.2 y recomendamos TLS 1.3.
- Configure la API y el registro de actividad de los usuarios con AWS CloudTrail. Para obtener información sobre el uso de CloudTrail senderos para capturar AWS actividades, consulte Cómo trabajar con CloudTrail senderos en la Guía del AWS CloudTrail usuario.
- Utilice soluciones de AWS cifrado, junto con todos los controles de seguridad predeterminados Servicios de AWS.
- Utiliza servicios de seguridad administrados avanzados, como Amazon Macie, que lo ayuden a detectar y proteger la información confidencial almacenada en Amazon S3.
- Si necesita módulos criptográficos validados por FIPS 140-3 para acceder a AWS través de una interfaz de línea de comandos o una API, utilice un punto final FIPS. Para obtener más información sobre los puntos de conexión de FIPS disponibles, consulte Estándar de procesamiento de la información federal (FIPS) 140-3.

Se recomienda encarecidamente no introducir nunca información confidencial o sensible, como por ejemplo, direcciones de correo electrónico de clientes, en etiquetas o campos de formato libre, tales como el campo Nombre. Esto incluye cuando trabaja con AWS HealthOmics u otros Servicios de AWS mediante la consola, la API o AWS SDKs. AWS CLI Cualquier dato que introduzca en etiquetas o campos de formato libre utilizados para los nombres se pueden emplear para los registros de facturación o diagnóstico. Si proporciona una URL a un servidor externo, recomendamos encarecidamente que no incluya la información de las credenciales en la URL para validar la solicitud para ese servidor.

Cifrado en reposo

Temas

- Claves propiedad de AWS
- Claves administradas por el cliente
- Creación de una clave administrada por el cliente
- Permisos de IAM necesarios para usar una clave administrada por el cliente
- Más información

Para proteger los datos confidenciales de los clientes en reposo, AWS HealthOmics proporciona cifrado de forma predeterminada mediante una clave de AWS Key Management Service (AWS KMS) propiedad del servicio. También se admiten las claves administradas por el cliente. Para obtener más información sobre la clave gestionada por el cliente, consulta Amazon Key Management Service.

Todos los almacenes de HealthOmics datos (almacenamiento y análisis) admiten el uso de claves administradas por el cliente. La configuración de cifrado no se puede cambiar una vez creado un almacén de datos. Si un almacén de datos utiliza una Clave propiedad de AWS, se indicará como AWS_OWNED_KMS_KEY y no verá la clave específica utilizada para el cifrado en reposo.

En el caso de los HealthOmics flujos de trabajo, el sistema de archivos temporales no admite las claves administradas por el cliente; sin embargo, todos los datos en reposo se cifran automáticamente mediante el algoritmo de cifrado por bloques XTS-AES-256 para cifrar el sistema de archivos. El usuario y el rol de IAM utilizados para iniciar la ejecución de un flujo de trabajo también deben tener acceso a las claves utilizadas para los depósitos de entrada y salida del flujo de trabajo. AWS KMS Los flujos de trabajo no utilizan concesiones y el AWS KMS cifrado se limita a los buckets de entrada y salida de Amazon S3. La función de IAM utilizada tanto para el flujo de trabajo APIs debe tener acceso a las AWS KMS claves utilizadas, así como a los buckets de entrada y salida de Amazon S3. Puede utilizar las funciones y permisos de IAM para controlar el acceso o las políticas. AWS KMS Para obtener más información, consulte Autenticación y control de acceso para AWS KMS.

Cuando se utiliza AWS Lake Formation con HealthOmics Analytics, los permisos de descifrado asociados a Lake Formation también se otorgan a los buckets de entrada y salida de Amazon S3. Puede encontrar más información sobre cómo se AWS Lake Formation gestionan los permisos en la AWS Lake Formation documentación.

HealthOmics Analytics concede a Lake Formation kms: Decrypt permisos para leer los datos cifrados de un bucket de Amazon S3. Siempre que tenga permisos para consultar los datos a través de Lake Formation, podrá leer los datos cifrados. El acceso a los datos se controla mediante el control de acceso a los datos en Lake Formation, no mediante una política clave de KMS. Para obtener más información, consulte las solicitudes de servicio de AWS AWS integradas en la documentación de Lake Formation.

Claves propiedad de AWS

De forma predeterminada, se HealthOmics utiliza Claves propiedad de AWS para cifrar automáticamente los datos en reposo, ya que estos datos pueden contener información confidencial, como información de identificación personal (PII) o información de salud protegida (PHI). Claves propiedad de AWS no se almacenan en su cuenta. Forman parte de un conjunto de claves de KMS que AWS posee y administra para su uso en varias cuentas de AWS.

Los servicios de AWS se pueden utilizar Claves propiedad de AWS para proteger sus datos. No puede ver, administrar Claves propiedad de AWS, acceder a su uso ni auditarlo. Sin embargo, no es necesario que realice ninguna acción ni que cambie programas para proteger las claves que cifran sus datos.

No se le cobra una cuota mensual ni una cuota de uso por su uso Claves propiedad de AWS, y estas no se tienen en cuenta para las cuotas de AWS KMS de su cuenta. Para obtener más información, consulte Claves administradas por AWS.

Claves administradas por el cliente

HealthOmics admite el uso de claves simétricas administradas por el cliente que usted crea, posee y administra para agregar una segunda capa de cifrado sobre el cifrado existente propiedad de AWS. Como usted tiene el control total de dicho cifrado, puede realizar tareas como las siguientes:

- Establecer y mantener políticas de claves, políticas de IAM y concesiones
- Rotar el material criptográfico
- Habilitar y deshabilitar políticas de claves
- Adición de etiquetas de
- Crear alias de clave
- Programar la eliminación de claves

También puede utilizarlas CloudTrail para realizar un seguimiento de las solicitudes que se HealthOmics envían AWS KMS en su nombre. Se aplican cargos adicionales de AWS KMS . Para obtener más información, consulta las claves gestionadas por el cliente.

Creación de una clave administrada por el cliente

Puede crear una clave simétrica administrada por el cliente mediante la consola de administración de AWS o la AWS KMS APIs.

Siga los pasos para crear claves simétricas administradas por el cliente de la Guía para desarrolladores de AWS Key Management Service.

Las políticas de clave controlan el acceso a la clave administrada por el cliente. Cada clave administrada por el cliente debe tener exactamente una política de clave, que contiene instrucciones que determinan quién puede usar la clave y cómo puede utilizarla. Al crear una clave administrada por el cliente, puede especificar una política de claves. Para obtener más información, consulte Administrar el acceso a las claves administradas por el cliente en la Guía para desarrolladores de AWS Key Management Service.

Para usar una clave administrada por el cliente con sus recursos de HealthOmics análisis, la persona que realiza la llamada debe incluir kms: CreateGrant operations en la política clave. Esto permite que el sistema utilice un token FAS para crear una concesión a una clave gestionada por el cliente que controle el acceso a una clave KMS específica. Esta clave permite al usuario acceder a las operaciones de kms:grant que necesite. HealthOmics Consulte Uso de subvenciones para obtener más información.

Para el HealthOmics análisis, se deben permitir las siguientes operaciones de API para la persona que realiza la llamada:

- kms: CreateGrant añade concesiones a una clave específica administrada por el cliente, lo que permite el acceso a las operaciones de concesión en HealthOmics Analytics.
- kms: DescribeKey proporciona los detalles de la clave gestionada por el cliente necesarios para validar la clave. Esto es necesario para todas las operaciones.
- kms: GenerateDataKey proporciona acceso a los recursos de cifrado en reposo para todas las operaciones de escritura. Además, esta acción proporciona detalles de la clave administrada por el cliente que el servicio puede usar para validar que la persona que llama tiene acceso para usar la clave.
- KMS:Decrypt proporciona acceso a las operaciones de lectura o búsqueda de recursos cifrados.

Para utilizar una clave gestionada por el cliente con sus recursos HealthOmics de almacenamiento, la política de claves debe permitir al director de HealthOmics servicio y al principal que realiza la llamada. Esto permite que el servicio valide que la persona que llama tiene acceso a la clave y utiliza la entidad principal de servicio para gestionar la tienda mediante la clave gestionada por el cliente. En cuanto al HealthOmics almacenamiento, la política de claves del director del servicio debe permitir las siguientes operaciones de API:

- kms: DescribeKey proporciona los detalles de la clave administrada por el cliente necesarios para validar la clave. Esto es necesario para todas las operaciones.
- kms: GenerateDataKey proporciona acceso a los recursos de cifrado en reposo para todas las operaciones de escritura. Además, esta acción proporciona detalles de la clave administrada por el cliente que el servicio puede usar para validar que la persona que llama tiene acceso para usar la clave.
- KMS:Decrypt proporciona acceso a las operaciones de lectura o búsqueda de recursos cifrados.

El siguiente ejemplo muestra una declaración de política que permite al director de un servicio crear e interactuar con un almacén de HealthOmics secuencias o referencias cifrado mediante la clave gestionada por el cliente:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
{
    "Effect": "Allow",
    "Principal": {
        "Service": "omics.amazonaws.com"
    },
    "Action": [
        "kms:Decrypt",
        "kms:DescribeKey",
        "kms:Encrypt",
        "kms:GenerateDataKey*"
    ],
    "Resource": "*"
        }
    ]
```

```
}
```

El siguiente ejemplo muestra una política que crea permisos para que un almacén de datos descifre los datos de un bucket de Amazon S3.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "omics:GetReference",
                "omics:GetReferenceMetadata"
            ],
            "Resource": [
                "arn:aws:omics:us-east-1:123456789012:referenceStore/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::[[s3path]]/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:123456789012:key/key_id"
            ],
            "Condition": {
              "StringEquals": {
                  "kms:ViaService": [
                    "s3.us-east-1.amazonaws.com"
```

```
                ]
            }
        }
    ]
}
```

Permisos de IAM necesarios para usar una clave administrada por el cliente

Al crear un recurso, como un almacén de datos AWS KMS cifrado mediante una clave administrada por el cliente, se requieren permisos tanto para la política de claves como para la política de IAM para el usuario o rol de IAM.

Puede usar la clave de ViaService condición kms: para limitar el uso de la clave KMS únicamente a las solicitudes que se originen en. HealthOmics

Para obtener más información sobre las políticas clave, consulte Habilitar las políticas de IAM en la Guía para desarrolladores de AWS Key Management Service.

Temas

- Permisos de la API de análisis
- Permisos de la API de almacenamiento
- Cómo se HealthOmics utilizan las subvenciones en AWS KMS
- Supervisar sus claves de cifrado para AWS HealthOmics

Permisos de la API de análisis

Para el HealthOmics análisis, el usuario o rol de IAM que crea las tiendas debe tener los permisos kms: CreateGrant kms:GenerateDataKey, kms: Decrypt y, además de kms: DescribeKey los permisos necesarios HealthOmics .

Permisos de la API de almacenamiento

Para el HealthOmics almacenamiento APIs, el usuario o rol de IAM que invoca las siguientes operaciones de API requiere los permisos enumerados:

CreateReferenceStore, CreateSequenceStore

Para crear un almacén, la persona que llama a IAM debe tener `kms:DescribeKey` los permisos necesarios y los permisos necesarios. HealthOmics El director del HealthOmics servicio llama `kms:GenerateDataKeyWithoutPlaintext` para realizar comprobaciones de validación de acceso para cargar y acceder a los datos.

StartReadSetImportJob, StartReferenceImportJob

Para iniciar los trabajos de importación de datos, la persona que llama a IAM debe tener `kms:Decrypt` `kms:GenerateDataKey` permisos para la clave de KMS en el almacén de importación y `kms:Decrypt` permisos en el depósito de Amazon S3 que contiene los objetos que se van a importar. Además, el rol transferido a la llamada debe tener `kms:Decrypt` permisos en el bucket de Amazon S3 que contiene los objetos que se van a importar. La persona que llama a IAM también debe tener permisos para transferir la función al trabajo.

CreateMultipartReadSetUpload, UploadReadSetPart, CompleteMultipartReadSetUpload

Para completar una carga en varias partes, la persona que llama a IAM debe tener `kms:Decrypt` y crear, `kms:GenerateDataKey` cargar y completar la carga en varias partes.

StartReadSetExportJob

Para iniciar un trabajo de exportación de datos, la persona que llama a IAM debe tener `kms:Decrypt` permiso para exportar la clave KMS del almacén `kms:GenerateDataKey` y `kms:Decrypt` permisos en el depósito de Amazon S3 que recibe los objetos. Además, el rol transferido a la llamada debe tener `kms:Decrypt` permisos en el bucket de Amazon S3 que recibe los objetos. La persona que llama a IAM también debe tener permisos para transferir la función al trabajo.

StartReadsetActivationJob

Para iniciar un trabajo de activación de un conjunto de lectura, la persona que llama a IAM debe tener los objetos `kms:Decrypt` y `kms:GenerateDataKey` permisos para utilizarlos.

GetReference, GetReadSet

Para leer los objetos del almacén, la persona que llama a IAM debe tener `kms:Decrypt` permiso para acceder a los objetos.

Lea el conjunto S3 GetObject

Para leer objetos del almacén mediante la `GetObject` API de Amazon S3, la persona que llama a IAM debe tener `kms:Decrypt` permiso para los objetos. Establezca este permiso tanto para la clave administrada por el cliente como para las Clave propiedad de AWS configuraciones.

Cómo se HealthOmics utilizan las subvenciones en AWS KMS

HealthOmics Analytics requiere una subvención para usar la clave de KMS administrada por el cliente. Las subvenciones no son obligatorias ni se utilizan para los HealthOmics flujos de trabajo. HealthOmics Storage utiliza la clave gestionada por el cliente directamente del director del servicio, por lo que no utilice una concesión. Cuando crea un almacén de análisis cifrado con una clave administrada por el cliente, HealthOmics Analytics crea una subvención en su nombre mediante el envío de una CreateGrantsolicitud a AWS KMS. Las subvenciones en AWS KMS se utilizan para dar HealthOmics acceso a una clave de KMS en una cuenta de cliente.

No se recomienda revocar o retirar las subvenciones que HealthOmics Analytics crea en su nombre. Si revoca o retira la concesión que le da HealthOmics permiso para usar las claves de AWS KMS de su cuenta, HealthOmics no puede acceder a estos datos, cifra los nuevos recursos enviados al almacén de datos o los descifra cuando se extraen.

Cuando revoca o retira una subvención HealthOmics, el cambio se produce inmediatamente. Para revocar los derechos de acceso, le recomendamos que elimine el almacén de datos en lugar de revocar la concesión. Al eliminar el almacén de datos, HealthOmics retira las concesiones en tu nombre.

Supervisar sus claves de cifrado para AWS HealthOmics

Puedes utilizarla CloudTrail para realizar un seguimiento de las solicitudes que se AWS HealthOmics envían AWS KMS en tu nombre cuando utilizas una clave gestionada por el cliente. Las entradas del CloudTrail registro muestran HealthOmics .AmazonAWS.com en el campo UserAgent para distinguir claramente las solicitudes realizadas por. HealthOmics

Los siguientes ejemplos son CloudTrail eventos para CreateGrant descifrar y supervisar las AWS KMS operaciones solicitadas para acceder DescribeKey HealthOmics a los datos cifrados por la clave gestionada por el cliente. GenerateDataKey

A continuación, también se muestra cómo permitir que los HealthOmics análisis accedan a una clave de KMS proporcionada por el cliente, lo que permite HealthOmics utilizar esa clave de KMS para cifrar todos los datos de los clientes en reposo. CreateGrant

No tienes la obligación de crear tus propias subvenciones. HealthOmics crea una subvención en su nombre mediante el envío de una CreateGrant solicitud a AWS KMS. Las concesiones in se AWS KMS utilizan para dar HealthOmics acceso a una AWS KMS clave de la cuenta de un cliente.


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "xx:test",
        "arn": "arn:AWS:sts::555555555555:assumed-role/user-admin/test",
        "accountId": "xx",
        "accessKeyId": "xxx",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "xxxx",
                "arn": "arn:AWS:iam::555555555555:role/user-admin",
                "accountId": "555555555555",
                "userName": "user-admin"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2022-11-11T01:36:17Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedBy": "apigateway.amazonAWS.com"
    },
    "eventTime": "2022-11-11T02:34:41Z",
    "eventSource": "kms.amazonAWS.com",
    "eventName": "CreateGrant",
    "AWSRegion": "us-west-2",
    "sourceIPAddress": "apigateway.amazonAWS.com",
    "userAgent": "apigateway.amazonAWS.com",
    "requestParameters": {
        "granteePrincipal": "AWS Internal",
        "keyId": "arn:AWS:kms:us-west-2:555555555555:key/a6e87d77-cc3e-4a98-a354-
e4c275d775ef",
        "operations": [
            "CreateGrant",
            "RetireGrant",
            "Decrypt",
            "GenerateDataKey"
```

```
        ]
    },
    "responseElements": {
        "grantId": "4869b81e0e1db234342842af9f5531d692a76edaff03e94f4645d493f4620ed7",
        "keyId": "arn:AWS:kms:us-west-2:245126421963:key/xx-cc3e-4a98-a354-
e4c275d775ef"
    },
    "requestID": "d31d23d6-b6ce-41b3-bbca-6e0757f7c59a",
    "eventID": "3a746636-20ef-426b-861f-e77efc56e23c",
    "readOnly": false,
    "resources": [
        {
            "accountId": "245126421963",
            "type": "AWS::KMS::Key",
            "ARN": "arn:AWS:kms:us-west-2:245126421963:key/xx-cc3e-4a98-a354-
e4c275d775ef"
        }
    ],
    "eventType": "AWSApiCall",
    "managementEvent": true,
    "recipientAccountId": "245126421963",
    "eventCategory": "Management"
}
```

El siguiente ejemplo muestra cómo se utiliza GenerateDataKey para garantizar que el usuario tenga los permisos necesarios para cifrar los datos antes de almacenarlos.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "EXAMPLEUSER",
        "arn": "arn:AWS:sts::111122223333:assumed-role/Sampleuser01",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLEKEYID",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "EXAMPLEROLE",
                "arn": "arn:AWS:iam::111122223333:role/Sampleuser01",
                "accountId": "111122223333",
                "userName": "Sampleuser01"
```

```
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2021-06-30T21:17:06Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedBy": "omics.amazonAWS.com"
    },
    "eventTime": "2021-06-30T21:17:37Z",
    "eventSource": "kms.amazonAWS.com",
    "eventName": "GenerateDataKey",
    "AWSRegion": "us-east-1",
    "sourceIPAddress": "omics.amazonAWS.com",
    "userAgent": "omics.amazonAWS.com",
    "requestParameters": {
        "keySpec": "AES_256",
        "keyId": "arn:AWS:kms:us-east-1:111122223333:key/EXAMPLE_KEY_ARN"
    },
    "responseElements": null,
    "requestID": "EXAMPLE_ID_01",
    "eventID": "EXAMPLE_ID_02",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:AWS:kms:us-east-1:111122223333:key/EXAMPLE_KEY_ARN"
        }
    ],
    "eventType": "AWSApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```


Más información

Los siguientes recursos proporcionan más información sobre el cifrado de datos en reposo.

Para obtener más información sobre los conceptos básicos de AWS Key Management Service, consulte la AWS KMS documentación.

Para obtener más información sobre las prácticas recomendadas de seguridad, AWS KMS consulte la documentación.

Cifrado en tránsito

AWS HealthOmics utiliza TLS 1.2+ para cifrar los datos en tránsito a través de los puntos finales públicos y a través de los servicios de backend.

Gestión de identidad y acceso en HealthOmics

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 recursos. AWS Los administradores de IAM controlan quién puede autenticarse (iniciar sesión) y quién puede autorizarse (tener permisos) para usar los recursos de AWS HealthOmics . La IAM es una Servicio de AWS herramienta que puede utilizar sin coste adicional.

Temas

- Público
- Autenticación con identidades
- Administración del acceso con políticas
- ¿Cómo AWS HealthOmics funciona con IAM
- Ejemplos de políticas basadas en la identidad para AWS HealthOmics
- AWS políticas gestionadas para AWS HealthOmics
- Solución de problemas de AWS HealthOmics identidad y acceso

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 Solución de problemas de AWS HealthOmics identidad y acceso).
- Administrador del servicio: determine el acceso de los usuarios y envíe las solicitudes de permiso (consulte ¿Cómo AWS HealthOmics funciona con IAM).
- Administrador de IAM: escribe las políticas para administrar el acceso (consulte Ejemplos de políticas basadas en la identidad para AWS HealthOmics).

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 recomiendaencarecidamente 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 HealthOmics funciona con IAM

Antes de usar IAM para administrar el acceso a AWS HealthOmics, conozca qué funciones de IAM están disponibles para usar con AWS. HealthOmics

Características de IAM que puede utilizar con AWS HealthOmics

Característica de IAM	HealthOmics soporte
Políticas basadas en identidades	Sí
Políticas basadas en recursos	No
Acciones de políticas	Sí
Recursos de políticas	Sí
Claves de condición de política	No
ACLs	No
ABAC (etiquetas en las políticas)	Sí
Credenciales temporales	Sí
Permisos de entidades principales	Sí
Roles de servicio	Sí
Roles vinculados al servicio	No

Para obtener una visión general de cómo HealthOmics 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.

Prevención de la sustitución confusa entre servicios

El problema de la sustitución confusa es un problema de seguridad en el que una entidad que no tiene permiso para realizar una acción puede obligar a una entidad con más privilegios a realizar la acción. En AWS, la suplantación de identidad entre servicios puede provocar el confuso problema de un diputado. La suplantación entre servicios puede producirse cuando un servicio (el servicio que lleva a cabo las llamadas) llama a otro servicio (el servicio al que se llama). El servicio que lleva a cabo las llamadas se puede manipular para utilizar sus permisos a fin de actuar en función de los recursos de otro cliente de una manera en la que no debe tener permiso para acceder. Para evitarlo, AWS proporciona herramientas que le ayudan a proteger sus datos para todos los servicios con entidades principales de servicio a las que se les ha dado acceso a los recursos de su cuenta.

Recomendamos usar las claves de contexto de condición `aws:SourceAccount`global `aws:SourceArn`y las claves de contexto en las políticas de recursos para limitar los permisos que AWS HealthOmics otorga a otro servicio al recurso.

Para evitar el confuso problema de los suplentes en las funciones que asumen HealthOmics, defina el valor de `aws:SourceArn` to `arn:aws:omics:`*`region`*`:`*`accountNumber`*`:*` en la política de confianza de la función. El comodín (*) aplica la condición a todos los HealthOmics recursos.

La siguiente política de relaciones de confianza otorga HealthOmics acceso a sus recursos y utiliza las claves contextuales `aws:SourceArn` y de condición `aws:SourceAccount` global para evitar el confuso problema de los diputados. Utilice esta política al crear un rol para HealthOmics.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
```

```
            "omics.amazonaws.com"
          ]
        },
        "Action": "sts:AssumeRole",
        "Condition": {
          "StringEquals": {
            "aws:SourceAccount": "123456789012"
          },
          "ArnLike": {
            "aws:SourceArn": "arn:aws:omics:us-east-1:123456789012:*"
          }
        }
      }
    ]
}
```

Políticas basadas en la identidad para HealthOmics

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 HealthOmics

Para ver ejemplos de políticas de AWS HealthOmics basadas en la identidad, consulte. Ejemplos de políticas basadas en la identidad para AWS HealthOmics

Políticas basadas en recursos dentro de HealthOmics

Admite políticas basadas en recursos: no

Las políticas basadas en recursos son documentos de política JSON que se asocian a un recurso. Los ejemplos de políticas basadas en recursos son 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. Para el recurso al que se asocia la política, la política define qué acciones puede realizar una entidad principal especificada en ese recurso y en qué condiciones. Debe especificar una entidad principal en una política basada en recursos. Los principales pueden incluir cuentas, usuarios, roles, usuarios federados o. Servicios de AWS

Para habilitar el acceso entre cuentas, puede especificar toda una cuenta o entidades de IAM de otra cuenta como la entidad principal de una política en función de recursos. Para obtener más información, consulte Acceso a recursos entre cuentas en IAM en la Guía del usuario de IAM.

Acciones políticas para HealthOmics

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 HealthOmics acciones, consulte Acciones definidas por AWS HealthOmics en la Referencia de autorización de servicios.

Las acciones políticas HealthOmics utilizan el siguiente prefijo antes de la acción:

```
omics
```

Para especificar varias acciones en una única instrucción, sepárelas con comas.

```
"Action": [
      "omics:action1",
      "omics:action2"
         ]
```

Para ver ejemplos de políticas de AWS HealthOmics basadas en la identidad, consulte. Ejemplos de políticas basadas en la identidad para AWS HealthOmics

Recursos de políticas para HealthOmics

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 HealthOmics recursos y sus respectivos tipos ARNs, consulte Recursos definidos por AWS HealthOmics en la Referencia de autorización de servicios. Para saber con qué acciones puede especificar el ARN de cada recurso, consulte Acciones definidas por AWS. HealthOmics

Para ver ejemplos de políticas de AWS HealthOmics basadas en la identidad, consulte. Ejemplos de políticas basadas en la identidad para AWS HealthOmics

Claves de condición de la política para HealthOmics

No se admiten las claves de condición de la política HealthOmics.

Listas de control de acceso (ACLs) en HealthOmics

Soporta ACLs: No

Las listas de control de acceso (ACLs) controlan qué directores (miembros de la cuenta, usuarios o roles) tienen permisos para acceder a un recurso. ACLs son similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de políticas JSON.

Control de acceso basado en atributos (ABAC) con HealthOmics

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.

Para obtener más información acerca del etiquetado de recursos de HealthOmics, consulte Etiquetado de recursos en HealthOmics.

El siguiente ejemplo muestra cómo se puede redactar una política de IAM que deniegue el acceso a un recurso sin una etiqueta específica.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "omics:*"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "Null": {
                  "aws:RequestTag/MyCustomTag": "true"
                }
```

```
            }
        }
    ]
}
```

Uso de credenciales temporales con HealthOmics

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 rol. 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 HealthOmics

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 descendentes. Para obtener información sobre las políticas a la hora de realizar solicitudes de FAS, consulte Sesiones de acceso directo.

Funciones de servicio para HealthOmics

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 HealthOmics la funcionalidad. Edite las funciones de servicio solo cuando se HealthOmics proporcionen instrucciones para hacerlo.

Funciones vinculadas al servicio para HealthOmics

Compatibilidad con roles vinculados al servicio: no

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 HealthOmics

De forma predeterminada, los usuarios y los roles no tienen permiso para crear o modificar HealthOmics los recursos de AWS. 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 por AWS HealthOmics, incluido el ARNs formato de cada uno de los tipos de recursos, consulte Acciones, recursos y claves de condición de AWS HealthOmics en la Referencia de autorización de servicios.

Temas

- Prácticas recomendadas sobre las políticas
- Uso de la consola HealthOmics
- 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, acceder o eliminar HealthOmics los recursos de AWS de su cuenta. 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 HealthOmics

Para acceder a la HealthOmics consola de AWS, debe tener un conjunto mínimo de permisos. Estos permisos deben permitirle enumerar y ver detalles sobre los HealthOmics recursos de AWS que tiene 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 necesita conceder permisos mínimos de consola a los usuarios que solo realicen llamadas a la AWS API AWS CLI o a la misma. En su lugar, permita el acceso únicamente a las acciones que coincidan con la operación de API que intentan realizar.

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 HealthOmics

Una política AWS administrada es una política independiente creada y administrada por AWS. AWS Las políticas administradas están diseñadas para proporcionar permisos para muchos casos de uso comunes, de modo que pueda empezar a asignar permisos a usuarios, grupos y funciones.

Ten en cuenta que es posible que las políticas AWS administradas no otorguen permisos con privilegios mínimos para tus casos de uso específicos, ya que están disponibles para que los usen todos los AWS clientes. Se recomienda definir políticas administradas por el cliente específicas para sus casos de uso a fin de reducir aún más los permisos.

No puedes cambiar los permisos definidos en AWS las políticas administradas. Si AWS actualiza los permisos definidos en una política AWS administrada, la actualización afecta a todas las identidades principales (usuarios, grupos y roles) a las que está asociada la política. AWS es más probable que actualice una política AWS administrada cuando Servicio de AWS se lance una nueva o cuando estén disponibles nuevas operaciones de API para los servicios existentes.

Para obtener más información, consulte Políticas administradas por AWS en la Guía del usuario de IAM.

AWS política gestionada: AmazonOmicsFullAccess

Puede adjuntar la `AmazonOmicsFullAccess` política a sus identidades de IAM para darles acceso total a HealthOmics ella.

Esta política otorga permisos de acceso total a todas las HealthOmics acciones. Al crear un almacén de anotaciones o variantes, Omics también le dará acceso a ese almacén mediante una invitación a compartir recursos en la consola Resource Access Manager (RAM). Para obtener más información sobre las invitaciones a Resource Share a través de Lake Formation, consulte la sección Intercambio de datos entre cuentas en Lake Formation. Para una política de administración de Omics, también necesita los siguientes permisos para acceder a su bucket de Amazon S3.

- PutObject
- GetObject
- ListBucket
- AbortMultipartUpload
- ListMultipartUploadParts

JSON

```
{
 "Version":"2012-10-17",
 "Statement": [
  {
   "Effect": "Allow",
   "Action": [
    "omics:*"
   ],
   "Resource": "*"
  },
  {
   "Effect": "Allow",
   "Action": [
    "ram:AcceptResourceShareInvitation",
    "ram:GetResourceShareInvitations"
   ],
   "Resource": "*",
   "Condition": {
    "StringEquals": {
     "aws:CalledViaLast": "omics.amazonaws.com"
    }
   }
  },
```

```
  {
   "Effect": "Allow",
   "Action": "iam:PassRole",
   "Resource": "*",
   "Condition": {
    "StringEquals": {
     "iam:PassedToService": "omics.amazonaws.com"
    }
   }
  }
 ]
}
```

AWS política gestionada: AmazonOmicsReadOnlyAccess

Puede adjuntar la `AWSOmicsReadOnlyAccess` política a sus identidades de IAM cuando desee limitar los permisos de esa identidad al acceso de solo lectura.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "omics:Get*",
                "omics:List*"
            ],
            "Resource": "*"
        }
    ]
}
```

HealthOmics actualizaciones de las políticas gestionadas AWS

Consulte los detalles sobre las actualizaciones de las políticas AWS administradas HealthOmics 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 HealthOmics documento.

Cambio	Descripción	Fecha
AmazonOmicsFullAccess - Se ha añadido una nueva política	HealthOmics se agregó una nueva política para otorgar al usuario acceso completo a todas las acciones y recursos. Para obtener más información, consulte AmazonOmicsFullAccess.	23 de febrero de 2023
HealthOmics comenzó a rastrear los cambios	HealthOmics comenzó a realizar un seguimiento de los cambios de sus políticas AWS gestionadas.	29 de noviembre de 2022
AmazonOmicsReadOnlyAccess - Se agregó una nueva política	HealthOmics se agregó una nueva política que limita el acceso a solo lectura. Para obtener más información, AmazonOmicsReadOnlyAccess.	29 de noviembre de 2022

Solución de problemas de AWS HealthOmics identidad y acceso

Utilice la siguiente información como ayuda para diagnosticar y solucionar problemas comunes que puedan surgir al trabajar con AWS HealthOmics e IAM.

Temas

- No estoy autorizado a realizar ninguna acción en HealthOmics
- No estoy autorizado a realizar tareas como: PassRole
- Quiero permitir que personas ajenas a mí accedan Cuenta de AWS a mis HealthOmics recursos

No estoy autorizado a realizar ninguna acción en HealthOmics

Si recibe un error que indica que no tiene autorización para realizar una acción, las políticas se deben actualizar para permitirle realizar la acción.

En el siguiente ejemplo, el error se produce cuando el usuario de IAM `mateojackson` intenta utilizar la consola para consultar los detalles acerca de un recurso ficticio *`my-example-widget`*, pero no tiene los permisos ficticios `omics:`*`GetWidget`*.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 omics:GetWidget on resource: my-example-widget
```

En este caso, la política del usuario `mateojackson` debe actualizarse para permitir el acceso al recurso *`my-example-widget`* mediante la acción `omics:`*`GetWidget`*.

Si necesita ayuda, póngase en contacto con su AWS administrador. El administrador es la persona que le proporcionó las credenciales de inicio de sesión.

No estoy autorizado a realizar tareas como: PassRole

Si recibe un error que indica que no está autorizado a realizar la `iam:PassRole` acción, sus políticas deben actualizarse para que pueda transferir una función a AWS HealthOmics.

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 al servicio. Para ello, debe tener permisos para transferir la función al servicio.

El siguiente ejemplo de error se produce cuando un usuario de IAM denominado `marymajor` intenta utilizar la consola para realizar una acción en AWS HealthOmics. Sin embargo, la acción requiere que el servicio cuente con permisos que otorguen un rol de servicio. Mary no tiene permisos para transferir la función 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 mí accedan Cuenta de AWS a mis HealthOmics 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 respaldan 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 HealthOmics admite estas funciones, consulte¿Cómo AWS HealthOmics funciona con IAM.
- Para obtener información sobre cómo proporcionar acceso a los recursos de su Cuentas de AWS propiedad, consulte Proporcionar acceso a un usuario de IAM en otro usuario de su propiedad Cuenta de AWS en la Guía del usuario de IAM.
- Para obtener información sobre cómo proporcionar acceso a tus recursos a terceros Cuentas de AWS, consulta Cómo proporcionar acceso a recursos que Cuentas de AWS son propiedad de terceros en la Guía del usuario de IAM.
- Para obtener información sobre cómo proporcionar acceso mediante una federación de identidades, consulte Proporcionar acceso a usuarios autenticados externamente (identidad federada) en la Guía del usuario de IAM.
- Para conocer sobre la diferencia entre las políticas basadas en roles y en recursos para el acceso entre cuentas, consulte Acceso a recursos entre cuentas en IAM en la Guía del usuario de IAM.

Validación de conformidad para AWS HealthOmics

Los auditores externos evalúan la seguridad y el cumplimiento AWS HealthOmics como parte de varios programas de AWS cumplimiento. Esto incluye HIPAA, FedRAMP y otros. La siguiente tabla muestra las certificaciones de conformidad del servicio. HealthOmics

Certificación	Enlace
HIPAA	Referencia de servicios que cumplen con los requisitos de la HIPAA

Certificación	Enlace
HiTrust-CSF	Marco de seguridad común de Health Information Trust Alliance
FedRAMP Moderate (East/West)	Programa federal de gestión de riesgos y autorizaciones
ESTRELLA ISO/CSA	Certificados ISO y CSA STAR
C5	Catálogo de controles de cumplimiento de la informática en la nube
DoD CC SRG IL2	Guía de requisitos de seguridad de la computación en la nube del Departamento de Defensa
ENS High	Esquema Nacional de Seguridad
FINMA	Autoridad de Supervisión de los Mercados Financieros de Suiza
ES UN MAPA	Programa de evaluación y gestión de la seguridad de los sistemas de información
OSPAR	Informe de auditoría del proveedor de servicios subcontratado
PCI	Estándar de seguridad de datos de la industria de tarjetas de pago
Pinakes	Asociación bancaria CCI: calificación de terceros
PiTuKri	Criterios para evaluar la seguridad de la información de los servicios en la nube
SOC 1,2,3	Controles del sistema y la organización

Para ver una lista de todos los AWS servicios incluidos en el ámbito de aplicación de programas de conformidad específicos, consulte Servicios de AWS incluidos . Para obtener información general, consulta Programas de conformidad de AWS.

Puede descargar informes de auditoría de terceros utilizando AWS Artifact. Para obtener más información, consulte Descarga de informes en AWS Artifact .

HealthOmics los almacenes de datos utilizan el ID de muestra para nombrar los archivos internos y etiquetar los recursos. Antes de ingerir datos, compruebe si el ID de muestra contiene algún dato de PHI. Si es así, cambie la ID de la muestra antes de ingerir los datos. Para obtener más información, consulte la guía en la página web sobre el cumplimiento de la AWS HIPAA.

Su responsabilidad de cumplimiento al AWS HealthOmics utilizarlos viene determinada por la confidencialidad de sus datos, los objetivos de cumplimiento de su empresa y las leyes y reglamentos aplicables. AWS proporciona los siguientes recursos para ayudar con el cumplimiento:

- Security and Compliance Quick Start Guides (Guías de inicio rápido de seguridad y conformidad) (Guías de inicio rápido de seguridad y conformidad): Estas guías de implementación analizan las consideraciones en materia de arquitectura y proporcionan los pasos para implementar los entornos de referencia centrados en la seguridad y la conformidad en AWS.
- Documento técnico sobre cómo diseñar una arquitectura basada en la seguridad y el cumplimiento de la HIPAA: en este documento técnico se describe cómo pueden utilizar las empresas para crear aplicaciones que cumplan con la HIPAA. AWS
- AWS Recursos de cumplimiento Recursos de de trabajo y guías puede aplicarse a su sector y ubicación.
- Evaluación de los recursos con las reglas de la guía para AWS Config desarrolladores: AWS Config evalúa en qué medida las configuraciones de sus recursos cumplen con las prácticas internas, las directrices del sector y las normas.
- AWS Security Hub CSPM— Este AWS servicio proporciona una visión integral del estado de su seguridad AWS que le ayuda a comprobar el cumplimiento de los estándares y las mejores prácticas del sector de la seguridad.

Resiliencia en HealthOmics

La infraestructura AWS global se basa en zonas Regiones de AWS de disponibilidad. Regiones de AWS proporcionan varias zonas de disponibilidad aisladas y separadas físicamente, que están conectadas mediante redes de baja latencia, alto rendimiento y alta redundancia. Con las zonas de

disponibilidad, puede diseñar y utilizar aplicaciones y bases de datos que realizan una conmutación por error automática entre las zonas sin interrupciones. Las zonas de disponibilidad tienen una mayor disponibilidad, tolerancia a errores y escalabilidad que las infraestructuras tradicionales de uno o varios centros de datos.

Para obtener más información sobre las zonas de disponibilidad Regiones de AWS y las zonas de disponibilidad, consulte Infraestructura global.AWS

Además de la infraestructura AWS global, AWS HealthOmics ofrece varias funciones que ayudan a respaldar sus necesidades de respaldo y resiliencia de datos.

AWS HealthOmics y puntos finales de VPC de interfaz ()AWS PrivateLink

Puede establecer una conexión privada entre su VPC y crear un punto final AWS HealthOmics de VPC de interfaz. Los puntos de enlace de la interfaz funcionan con una tecnología que puede usar para acceder de forma privada a las operaciones de la HealthOmics API sin una puerta de enlace a Internet, un dispositivo NAT, una conexión VPN o una conexión AWS Direct Connect. AWS PrivateLink Las instancias de su VPC no requieren direcciones IP públicas para comunicarse con las operaciones de la HealthOmics API. El tráfico entre tu VPC y HealthOmics no sale de la red de Amazon.

Cada punto de conexión de la interfaz está representado por una o más interfaces de red elásticas en las subredes.

Para obtener más información, consulte Interface VPC Endpoints (AWS PrivateLink) en la Guía del usuario de Amazon VPC.

Las políticas de puntos finales de VPC se admiten en todas las regiones, excepto HealthOmics en Israel (Tel Aviv). De forma predeterminada, HealthOmics se permite el acceso total a través del punto final.

Consideraciones sobre los puntos HealthOmics finales de VPC

Antes de configurar un punto de enlace de VPC de interfaz HealthOmics, asegúrese de revisar las propiedades y limitaciones del punto de enlace de interfaz en la Guía del usuario de Amazon VPC.

HealthOmics permite realizar llamadas a todas las acciones de la API de HealthOmics almacenamiento desde su VPC.

Las políticas de puntos de enlace de VPC no se admiten de forma HealthOmics predeterminada, pero puede crear un punto de enlace de VPC para HealthOmics tener acceso total a las operaciones de almacenamiento. HealthOmics Para más información, consulte Control del acceso a los servicios con puntos de conexión de VPC en la Guía del usuario de Amazon VPC.

Creación de un punto final de VPC de interfaz para HealthOmics

Puede crear un punto de enlace de VPC para el HealthOmics servicio mediante la consola de Amazon VPC o el (). AWS Command Line Interface AWS CLI Para obtener más información, consulte Creación de un punto de conexión de interfaz en la Guía del usuario de Amazon VPC.

Cree un punto final de VPC con HealthOmics los siguientes nombres de servicio:

- com.amazonaws. *region*.storage-omics
- com.amazonaws. *region*. control-storage-omics
- com.amazonaws. *region*.analytics-omics
- com.amazonaws. *region*.workflows-omics
- com.amazonaws. *region*.tags-omics

Las regiones EE.UU. Este (Norte de Virginia) y EE.UU. Oeste (Oregón) son compatibles AWS PrivateLink con los puntos finales del FIPS. Para estas regiones, también puede usar los siguientes nombres de servicio:

- com.amazonaws. *region*. storage-omics-fips
- com.amazonaws. *region*. control-storage-omics-fips
- com.amazonaws. *region*. analytics-omics-fips
- com.amazonaws. *region*. workflows-omics-fips
- com.amazonaws. *region*. tags-omics-fips

Si activas el DNS privado para el punto final, puedes realizar solicitudes a la API HealthOmics utilizando su nombre de DNS predeterminado para la región, por ejemplo`omics.us-east-1.amazonaws.com`.

Para más información, consulte Acceso a un servicio a través de un punto de conexión de interfaz en la Guía del usuario de Amazon VPC.

Crear una política de puntos de conexión de VPC para HealthOmics

Puede asociar una política de punto de conexión con su punto de conexión de VPC que controla el acceso a HealthOmics. La política especifica la siguiente información:

- La entidad principal que puede realizar acciones
- Las acciones que se pueden realizar
- Los recursos en los que se pueden llevar a cabo las acciones

Para más información, consulte Control del acceso a los servicios con puntos de conexión de VPC en la Guía del usuario de Amazon VPC.

Ejemplo: política de puntos finales de VPC para HealthOmics acciones.

El siguiente es un ejemplo de una política de puntos finales para HealthOmics. Cuando se adjunta a un punto final, esta política otorga acceso a HealthOmics las acciones a todos los principales de todos los recursos.

API

```
{
   "Statement":[
      {
         "Principal":"*",
         "Effect":"Allow",
         "Action":[
            "omics:List*"
         ],
         "Resource":"*"
      }
   ]
}
```

AWS CLI

```
aws ec2 modify-vpc-endpoint \
    --vpc-endpoint-id vpce-id \
    --region us-west-2 \
    --policy-document \
```

```
    "{\"Statement\":[{\"Principal\":\"*\",\"Effect\":\"Allow\",\"Action\":
[\"omics:List*\"],\"Resource\":\"*\"}]}"
```

Consideraciones especiales para acceder a los conjuntos de lectura mediante Amazon S3 URIs

Para acceder a los conjuntos de lectura a través de Amazon S3 URIs cuando utilice una conexión privada, configure los puntos de enlace de la PrivateLink interfaz en el almacén de secuencias. Una vez configurados, los puntos de enlace tienen los siguientes formatos:

```
com.amazonaws.region.storage-omics
com.amazonaws.region.control-storage-omics
```

Para usar los puntos de enlace de puerta de enlace, siga la guía Puntos de enlace de puerta de enlace para Amazon S3 para configurar los puntos de enlace de puerta de enlace. HealthOmics es propietario del bucket de Amazon S3, por lo que no tiene que crear ni ajustar la política del bucket. Los puntos de enlace de enlace se basan en la política asociada al usuario o rol que accede a los datos, pero también puede configurar los puntos de enlace con políticas más restrictivas. Estas políticas pueden incluir restricciones de acceso en función del ARN del punto de acceso de Amazon S3 y de las acciones de Amazon S3.

Supervisión de AWS HealthOmics

La supervisión es una parte importante del mantenimiento de la fiabilidad, la disponibilidad y el rendimiento de AWS HealthOmics y del resto de sus AWS soluciones. AWS proporciona las siguientes herramientas de supervisión para vigilar AWS HealthOmics, informar cuando algo va mal y tomar medidas automáticas cuando sea necesario:

- Amazon CloudWatch supervisa tus AWS recursos y las aplicaciones en las que ejecutas AWS en tiempo real. Puede recopilar métricas y realizar un seguimiento de las métricas, crear paneles personalizados y definir alarmas que le advierten o que toman medidas cuando una métrica determinada alcanza el umbral que se especifique. Por ejemplo, puedes CloudWatch hacer un seguimiento del uso de la CPU u otras métricas de tus EC2 instancias de Amazon y lanzar automáticamente nuevas instancias cuando sea necesario. Para obtener más información, consulta la Guía del CloudWatch usuario de Amazon.
- Amazon CloudWatch Logs le permite supervisar, almacenar y acceder a sus archivos de registro desde EC2 instancias de Amazon y otras fuentes. CloudTrail CloudWatch Los registros pueden monitorear la información de los archivos de registro y notificarle cuando se alcanzan ciertos umbrales. También se pueden archivar los datos del registro en un almacenamiento de larga duración. Para obtener más información, consulta la Guía del usuario CloudWatch de Amazon Logs.
- AWS CloudTrail captura las llamadas a la API y otros eventos relacionados que realiza la Cuenta de AWS o que se realizan en nombre de esta. Además, entrega los archivos de registro a un bucket de Amazon S3 especificado. También pueden identificar qué usuarios y cuentas llamaron a AWS, la dirección IP de origen de las llamadas y el momento en que estas se realizaron. Para obtener más información, consulte la Guía del usuario de AWS CloudTrail.
- Amazon EventBridge es un servicio de bus de eventos sin servidor que facilita la conexión de sus aplicaciones con datos de diversas fuentes. EventBridge ofrece un flujo de datos en tiempo real desde sus propias aplicaciones, aplicaciones Software-as-a-Service (SaaS) y AWS servicios, y dirige esos datos a destinos como Lambda. Esto le permite monitorear los eventos que ocurren en los servicios y crear arquitecturas basadas en eventos. Para obtener más información, consulta la Guía del EventBridge usuario de Amazon.

> **Note**
>
> Para obtener actualizaciones de los servicios, configure y supervise su Personal Health Dashboard. Para obtener más información sobre cómo administrar el panel de control, consulte Introducción a su AWS Health Dashboard.

Temas

- Registro de acceso a S3
- Monitorización HealthOmics con CloudWatch métricas
- Supervisión HealthOmics con CloudWatch registros
- Registro de llamadas a la AWS HealthOmics API mediante AWS CloudTrail
- Uso EventBridge con AWS HealthOmics

Registro de acceso a S3

Puede supervisar el acceso a la API de Amazon S3 para HealthOmics secuenciar los datos del almacén mediante los registros de acceso creados en el almacén. Puede utilizarlo CloudWatch para supervisar el acceso a S3 desde las operaciones de la HealthOmics API. CloudWatch proporciona visibilidad del acceso a Amazon S3 que se origina desde su propia cuenta. Si usted, como propietario de los datos, comparte el acceso a una cuenta de terceros, el registro de acceso no estará disponible en CloudWatch. En su lugar, utilice el registro de acceso de S3 de la tienda, que registra todos los accesos de S3 a los datos del bucket de Amazon S3 configurado.

Configure los registros de acceso de S3 mediante las operaciones de la `UpdateSequenceStore` API `CreateSequenceStore` o. Además, asegúrese de que el principal del HealthOmics servicio (`omics.amazonaws.com`) tenga `s3:PutObject` permisos para el prefijo S3 configurado.

> **Note**
>
> Los registros utilizan la configuración de cifrado predeterminada del depósito de destino. Si el depósito usa una clave administrada por el cliente, el director del servicio debe tener acceso para usar la clave para escribirla.

Para desactivar el registro de acceso, utilice `UpdateSequenceStore` y ponga en blanco la configuración del registro de acceso.

Monitorización HealthOmics con CloudWatch métricas

Puede monitorizar el HealthOmics uso CloudWatch, que recopila datos sin procesar y los procesa para convertirlos en métricas legibles prácticamente en tiempo real. Estas estadísticas se mantienen durante 15 meses, de forma que pueda obtener acceso a información histórica y disponer de una mejor perspectiva sobre el desempeño de su aplicación web o servicio. También puede establecer alarmas que vigilen determinados umbrales y enviar notificaciones o realizar acciones cuando se cumplan dichos umbrales. Para obtener más información, consulta la Guía del CloudWatch usuario de Amazon.

El AWS HealthOmics servicio informa de las siguientes métricas en el espacio de `AWS/Omics` nombres.

Las métricas del recuento de llamadas de la API se muestran a continuación. AWS HealthOmics APIs Solo se informa de la dimensión de funcionamiento de la API.

- Referencia y almacén de referencias APIs :CreateReferenceStore, DeleteReferenceStore, StartReferenceImportJob
- Almacén de secuencias y conjunto de lectura APIs — CreateSequenceStore DeleteSequenceStore, StartReadSetImportJob, StartReadSetActivationJob, StartReadSetExportJob
- Tienda de variantes APIs — CreateVariantStore, DeleteVariantStore, StartVariantImportJob, CancelVariantImportJob
- Almacén de anotaciones APIs — CreateAnnotationStore,, DeleteAnotationStore, StartAnnotationImportJob CancelAnnotationImportJob
- Flujo de trabajo, ejecución y grupo de ejecución APIs : CreateWorkflow DeleteWorkflow, StartRun, CancelRun, DeleteRun, CreateRunGroup DeleteRunGroup

Visualización de métricas de *AWS HealthOmics*

CloudWatch Las métricas de AWS HealthOmics se pueden ver en la CloudWatch consola.

Para ver las métricas (CloudWatch consola)

1. Inicie sesión en la consola de administración de AWS y abra la consola de CloudWatch .

2. Elija Métricas, elija Todas las métricas y, a continuación, elija AWS/Usage.
3. Servicio de filtrado para. AWS HealthOmics
4. Elija la dimensión, un nombre de métrica y, a continuación, Add to graph (Añadir al gráfico).
5. Elija un valor para el intervalo de fechas. El recuento de las métricas del intervalo de fechas seleccionado se muestra en el gráfico.

Crear una alarma usando CloudWatch

Una CloudWatch alarma vigila una única métrica durante un período de tiempo específico y realiza una o más acciones: enviar una notificación a un tema del Amazon Simple Notification Service (Amazon SNS) o a una política de Auto Scaling. La acción o las acciones se basan en el valor de la métrica en relación con un umbral determinado durante un número de períodos de tiempo que usted especifique. CloudWatch también puede enviarle un mensaje de Amazon SNS cuando la alarma cambie de estado.

CloudWatch las alarmas invocan acciones solo cuando el estado cambia y ha persistido durante el período que usted especifique.

Para ver las métricas (consola) CloudWatch

1. Inicie sesión en la consola de administración de AWS y abra la consola de CloudWatch .
2. Elija Alarms y, a continuación, seleccione Create Alarm.
3. Elija AWS/Usage y, a continuación, elija una AWS HealthOmics métrica mediante la dimensión Servicio.
4. En Time Range, elija un intervalo de tiempo para monitorizar y, a continuación, elija Next.
5. Introduzca un Nombre y una Descripción.
6. Para Whenever, elija >= y escriba un valor máximo.
7. Si quieres CloudWatch enviar un correo electrónico cuando se alcance el estado de alarma, en la sección Acciones, en Siempre que aparezca esta alarma, selecciona State is ALARM. En Enviar notificación a, selecciona una lista de correo o selecciona Nueva lista y crea una nueva lista de correo.
8. Obtenga una vista previa de la alarma en la sección Vista previa de la alarma. Si está satisfecho con la alarma, elija Create Alarm.

Supervisión HealthOmics con CloudWatch registros

HealthOmics genera una variedad de registros para ayudarte a entender y solucionar los problemas de tus carreras. Los registros están disponibles en dos sitios: CloudWatch y Amazon S3.

De forma predeterminada, las ejecuciones tienen el registro activado. Si lo desea, puede desactivar el registro de una ejecución configurando `LogLevel = OFF` la startrun solicitud.

> **Note**
>
> Para obtener actualizaciones de los servicios, configure y supervise su Personal Health Dashboard. Para obtener más información sobre cómo administrar el panel de control, consulte Introducción a su AWS Health Dashboard.

Temas

- Tipos de registro para HealthOmics flujos de trabajo
- Inicia sesión CloudWatch
- Inicia sesión en Amazon S3
- CloudWatch Registros interactivos en la CLI
- Acceder a CloudWatch los registros desde la consola

Tipos de registro para HealthOmics flujos de trabajo

HealthOmics proporciona los siguientes tipos de registros para los flujos de trabajo:

- Registros del motor: los motores de flujo de trabajo subyacentes (Nextflow, WDL y CWL) producen registros del motor para las ejecuciones. Estos registros pueden ayudarle a solucionar problemas de definición del flujo de trabajo.
- Registros de manifiestos de ejecución: estos registros proporcionan información de alto nivel sobre cada tarea en ejecución, como el estado de la tarea, la hora de inicio, la hora de finalización y el motivo del error (si la tarea ha fallado).

  Los registros de ejecución de manifiestos también incluyen estadísticas de utilización de los recursos que pueden resultar útiles para comprender las oportunidades de optimización de los recursos. Estas estadísticas incluyen:

  - Promedio de CPU
  - Máximo de CPU
  - CPUs reservadas
  - GPUs reservadas
  - memoryAverageGiB
  - memoryMaximumGiB
  - memoryReservedGiB
  - Segundos de ejecución
- Registros de ejecución: los registros de ejecución proporcionan el estado general de la ejecución y la hora en que las tareas individuales se inician, se ejecutan, se detienen y se completan. Los registros de ejecución también permiten ver los pasos de importación y exportación de archivos.
- Registros de tareas: los registros de tareas proporcionan información de registro detallada sobre las tareas individuales de la ejecución. Los resultados del registro de tareas dependen de la definición de la tarea y del lugar en el código que utilices las sentencias de registro. Si los registros de tareas no proporcionan el nivel de información que necesitas, considera añadir más sentencias de registro a la definición de la tarea para producir registros de tareas más detallados.
- Registros de caché de ejecución: los registros de caché de ejecución proporcionan el estado general de las cachés de ejecución y del almacenamiento en caché de los resultados de las tareas. Los registros de caché de ejecución proporcionan visibilidad de los aciertos y errores de caché de cada ejecución que utiliza el almacenamiento en caché.
- Outputs.json: para los flujos de trabajo de WDL y CWL, HealthOmics entrega un archivo generado por el motor, denominado, a `outputs.json` su bucket de Amazon S3 una vez finalizada la ejecución. Estos archivos incluyen una lista y un mapa de todos los resultados de la ejecución.

Inicia sesión CloudWatch

CloudWatch genera registros de flujo de trabajo para las ejecuciones fallidas y correctas. Todos los registros están disponibles para las ejecuciones fallidas y correctas, excepto los registros del motor que solo están disponibles para las ejecuciones fallidas.

Puede encontrar los registros del CloudWatch flujo de trabajo en el siguiente grupo de registros:`/aws/omics/WorkflowLog`. Además, el resultado de la operación de API get-run proporciona el flujo de CloudWatch registro ARNs para los registros del motor y los registros de ejecución.

De forma predeterminada, AWS conserva los CloudWatch registros indefinidamente. Puede ajustar la política de retención del grupo de registros para establecer un período de retención de entre 10 años y un día.

En la siguiente tabla se proporciona un resumen de los CloudWatch inicios de sesión HealthOmics. Todos los registros del flujo de trabajo están disponibles para las ejecuciones correctas y las fallidas, excepto los registros del motor que solo están disponibles para las ejecuciones fallidas.

Nombre de registro	Disponible en CloudWatch los registros	¿Cuándo está disponible el registro	Formato de flujo de registro
Registros del motor	Sí, para ejecuciones fallidas	Una vez finalizada la ejecución	run/ /engine *runID*
Ejecute los registros de manifiestos	Sí	Una vez completada la ejecución	manifest/ run//*runIDrunUUID*
Ejecuta registros	Sí	En tiempo real	ejecutar/ *runID*
Registros de tareas	Sí	En tiempo real	ejecutar/ /tarea/ *runID taskID*
Ejecute los registros de caché	Sí	En tiempo real	RunCache/ /*runCacheI d runCacheUUID*
Outputs.json (WDL y CWL)	No	n/a	n/a

Inicia sesión en Amazon S3

Solo los registros del motor y el `outputs.json` archivo se envían a Amazon S3.

Una vez finalizada la ejecución, los registros del motor se envían a su bucket de S3 y están disponibles de forma indefinida hasta que los elimine. Estos registros se encuentran en el directorio de registros del URI de salida de S3 que especificó para el flujo de trabajo.

La ruta al directorio de registros tiene el siguiente formato:`s3://{user_provided_path}/logs/`.

La siguiente tabla proporciona un resumen de los HealthOmics registros disponibles en su bucket de Amazon S3.

Nombre de registro	Disponible en Amazon S3	¿Cuándo está disponible el registro	Ruta de flujo de registro
Registros del motor	Sí	Una vez completada la ejecución	s3://*user_provided_path* /logs/engine.log
Outputs.json (WDL y CWL)	Sí	Una vez completada la ejecución	s3://*user_provided_path* /*runID*/*runUUID*/logs/outputs.json
Ejecute registros de manifiestos, registros de ejecución y registros de tareas	No	n/a	n/a

CloudWatch Registros interactivos en la CLI

Puede ver los CloudWatch registros de forma interactiva mediante el comando Live Tail en modo interactivo. Puedes hacer un seguimiento del progreso de las carreras en tiempo real y definir hasta 5 palabras clave para resaltarlas en los registros:

```
aws logs start-live-tail  \
  --mode interactive  \
  --log-group-identifiers arn:aws:logs:region:account-ID:log-group:/aws/omics/
WorkflowLog
```

Para obtener más información, consulta Start live tail en la Referencia de AWS CLI comandos.

Acceder a CloudWatch los registros desde la consola

Para acceder a los registros de una ejecución, puedes vincular directamente a estos registros desde la página de detalles de la ejecución de la HealthOmics consola.

1. Abra la consola de HealthOmics .
2. Si es necesario, abre el panel de navegación izquierdo (≡). Elija Ejecuciones.
3. Seleccione la ejecución en la tabla Ejecuciones.
4. En la página de detalles de la ejecución, puede elegir cualquiera de estas acciones:

    a. En Resumen de la ejecución, selecciona Ver registros de ejecución. La consola abre los registros de ejecución en la CloudWatch consola.

    b. En Resumen de ejecución, seleccione Ver registros en Amazon S3. La consola abre la carpeta de registros en la consola de Amazon S3.

    c. En Ejecutar tareas, seleccione Ver registros, Ver registros de ejecución o Ver registros de manifiesto de ejecución de una tarea. La consola abre los registros de la CloudWatch consola.

También puedes navegar hasta los registros desde la CloudWatch consola:

1. Abre la CloudWatch consola https://console.aws.amazon.com/cloudwatch/.
2. En el menú de la izquierda, selecciona Registrar grupos.
3. Seleccione el grupo de `/aws/omics/WorkflowLog`.

    Si la lista de grupos de registros es larga, puede introducir omics en el cuadro de texto de búsqueda para reducir la lista.
4. Cuando se abra la página de detalles del grupo de registros, elija el flujo de registros que desee ver. La consola muestra los eventos de este flujo de registro.

Registro de llamadas a la AWS HealthOmics API mediante AWS CloudTrail

AWS HealthOmics 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 HealthOmics. CloudTrail captura todas las llamadas a la API HealthOmics como eventos. Las llamadas capturadas incluyen llamadas desde la HealthOmics consola y llamadas en código a las operaciones de la HealthOmics API. Si crea una ruta, puede habilitar la entrega continua de CloudTrail eventos a un bucket de Amazon S3, incluidos los eventos para HealthOmics. 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 a HealthOmics qué dirección IP 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.

HealthOmics información en CloudTrail

CloudTrail está habilitada en tu cuenta Cuenta de AWS al crear la cuenta. Cuando se produce una actividad en HealthOmics, esa actividad se registra en un CloudTrail evento junto con otros eventos de AWS servicio en el historial de eventos. Puede ver, buscar y descargar eventos recientes en su Cuenta de AWS. Para obtener más información, consulte Visualización de eventos con el historial de CloudTrail eventos.

Para tener un registro continuo de tus eventos Cuenta de AWS, incluidos los eventos para HealthOmics ti, crea una ruta. Un rastro permite CloudTrail entregar archivos de registro a un bucket de Amazon S3. De forma predeterminada, cuando se crea un registro de seguimiento en la consola, el registro de seguimiento se aplica a todas las Regiones de AWS. La ruta registra los eventos de todas las regiones de la AWS partición y envía los archivos de registro al bucket de Amazon S3 que especifique. Además, puede configurar otros AWS servicios para analizar más a fondo los datos de eventos recopilados en los CloudTrail registros y actuar en función de ellos. Para más información, consulte los siguientes temas:

- Introducción a la creación de registros de seguimiento
- CloudTrail servicios e integraciones compatibles
- Configuración de las notificaciones de Amazon SNS para CloudTrail
- Recibir archivos de CloudTrail registro de varias regiones y recibir archivos de CloudTrail registro de varias cuentas

Todas HealthOmics las acciones se registran CloudTrail y se documentan en la referencia de la AWS HealthOmics API. Por ejemplo, las llamadas a `StartVariantImportJob` y `CreateWorkflow` las acciones generan entradas en los archivos de CloudTrail registro. `CreateReferenceeStore`

Cada entrada de registro o evento contiene información sobre quién generó la solicitud. La información de identidad del usuario le ayuda a determinar lo siguiente:

- Si la solicitud se realizó con las credenciales de 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 AWS servicio.

Para obtener más información, consulte el elemento userIdentity de CloudTrail .

Descripción de las entradas de los archivos de HealthOmics registro

Un rastro es una configuración que permite la entrega de eventos como archivos de registro a un bucket de Amazon S3 que especifique. CloudTrail Los archivos de registro contienen una o más entradas de registro. Un evento representa una solicitud única de cualquier fuente e incluye información sobre la acción solicitada, la fecha y la hora de la acción, los parámetros de la solicitud, etc. CloudTrail Los archivos de registro no son un registro ordenado de las llamadas a la API pública, por lo que no aparecen en ningún orden específico.

En el siguiente ejemplo, se muestra una entrada de CloudTrail registro que demuestra la CreateWorkflow acción.


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROAIU53LOGOMTOPXXNPG:username",
        "arn": "arn:aws:sts::account:assumed-role/admin/username",
        "accountId": "account-id",
        "accessKeyId": "accessKeyId",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAIU53LOGOMTOPXXNPG",
                "arn": "arn:aws:iam::account:role/admin",
                "accountId": "account",
                "userName": "admin"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2022-07-23T18:26:09Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2022-07-23T18:46:42Z",
    "eventSource": "omics.amazonaws.com",
    "eventName": "CreateWorkflow",
```

```
    "awsRegion": "us-west-2",
    "sourceIPAddress": "205.251.233.176",
    "userAgent": "aws-cli/1.22.45 Python/3.9.13 Darwin/20.6.0 botocore/1.23.45",
    "requestParameters": {
        "name": "parameter_name",
        "definitionZip": "czM6Ly93b3JrZmxvd2RlZi1oZWxsby9kZWZpbml0aW9uLnppcA==",
        "requestId": "d788a73c-b81b-45fb-a8a6-d8bb4449ec8a"
    },
    "responseElements": {
        "id": "1002571",
        "arn": "arn:aws:omics:us-west-2:555555555555:instance/i-b188560f ",
        "status": "CREATING",
        "tags": {
            "resourceArn": "arn:aws:omics:us-west-2:083685709690:workflow/1002571"
        }
    },
    "requestID": "842d731d-f264-4b08-a2c9-2f7d45e1eaa3",
    "eventID": "76872ca2-f208-4193-807d-7dd7ea34e6b2",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "083685709690",
    "eventCategory": "Management"
}
```


Uso EventBridge con AWS HealthOmics

HealthOmics envía eventos a Amazon EventBridge cuando los recursos cambian de estado. Los recursos incluyen trabajos de importación, trabajos de exportación, recursos compartidos, flujos de trabajo, tareas y ejecuciones. Para cada tipo de recurso, hay una lista de cambios de estado que generan un evento.

Un bus de eventos es un router que recibe eventos y los envía a los destinos. Su cuenta incluye un bus de eventos predeterminado que recibe automáticamente los eventos de AWS los servicios. Puede crear autobuses de eventos personalizados adicionales.

Puede crear EventBridge reglas para especificar las acciones que se deben realizar cuando el bus de eventos reciba eventos. Por ejemplo, puede crear una regla que le notifique los cambios de estado de un recurso.

Los escenarios más comunes para el uso de eventos incluyen:

- Para supervisar cuándo un usuario comparte un recurso contigo o revoca el uso compartido.
- Para controlar si una ejecución falla o se completa correctamente.

Para obtener más información sobre el uso EventBridge, consulta ¿Qué es Amazon EventBridge?

Temas

- Configurar EventBridge para HealthOmics
- EventBridge eventos en HealthOmics
- Estructura de mensaje de evento
- Ejemplos de mensajes de eventos

Configurar EventBridge para HealthOmics

Antes de poder supervisar los EventBridge eventos, cree un EventBridge autobús y cree reglas para los eventos de interés.

Configure un EventBridge bus

Puede usar el bus de eventos predeterminado Cuenta de AWS o configurar un bus de eventos personalizado. Para configurar un bus de eventos personalizado, siga estos pasos:

1. Abre la EventBridge consola: https://console.aws.amazon.com/events/.
2. En el panel de navegación de la izquierda, selecciona Event buses.
3. Seleccione Crear bus de eventos.
4. En el formulario Crear bus de eventos, introduzca un nombre para el autobús.
5. Elija Crear para crear el autobús.

Cree una EventBridge regla

El siguiente procedimiento muestra cómo crear una regla sencilla. Para obtener más información sobre las reglas, consulte Reglas en EventBridge.

1. Abre la EventBridge consola: https://console.aws.amazon.com/events/.
2. En el panel de navegación izquierdo, seleccione Rules (Reglas).

3. Seleccione Creación de regla. La consola abre el formulario Crear regla.
4. En Definir los detalles de la regla, proporcione un nombre para la regla.
   - En Nombre, introduzca un nombre para el autobús.
   - En Event Bus, seleccione el autobús para esta regla.
   - Elija Siguiente.
5. En Crear patrón de eventos, en Origen del evento, selecciona Eventos de AWS o eventos de EventBridge socios.
6. Desplácese hacia abajo hasta Patrón de eventos.
   a. En Fuente del evento, selecciona los servicios de AWS.
   b. Para el servicio de AWS, introduzca omics en el filtro de texto y selecciónelo AWS HealthOmicscomo servicio.
   c. En Tipo de evento, seleccione el evento de interés (o Todos los eventos).
   d. Elija Siguiente.
7. En Seleccionar objetivos, seleccione un objetivo para el evento. Por ejemplo, elija el servicio de AWS, elija el grupo de CloudWatch registros y configure un grupo de registros.

   Si hay muchos tipos de destino, EventBridge necesita permiso para enviar eventos al destino. La consola crea estos permisos por usted.
8. (Opcional) En Configurar etiquetas, asocie las etiquetas a la regla.
9. En Revisar y actualizar, revise la configuración y elija Crear regla.

EventBridge eventos en HealthOmics

La siguiente tabla muestra los eventos que se HealthOmics envían a EventBridge y la lista de posibles valores de estado para el evento.

Nombre de evento	Valores de estado posibles
Cambio de estado del trabajo de importación de anotaciones	Enviado, en curso, cancelado, completado, fallido o completado con errores
Cambio de estado de intercambio del almacén de anotaciones	Pendiente, activando, activo, borrado, eliminado, fallido

Nombre de evento	Valores de estado posibles
Cambio de estado del almacén de anotaciones	Error al crear, actualizar, eliminar, eliminar o crear
Leer Establecer cambio de estado del trabajo de activación	Enviado, en curso, completado, fallido o completado con errores
Cambiar el estado del trabajo de Leer Set Export	Enviado, en curso, completado, fallido o completado con errores
Leer Set Import Job Change	Enviado, en curso, completado, fallido o completado con errores
Lea Establecer cambio de estado	Al procesar la carga, la carga ha fallado, se ha activado, se ha archivado, se ha activado o se ha eliminado
Cambio de estado del trabajo de importación de referencia	Enviado, en curso, completado, fallido o completado con errores
Cambio de estado de referencia	Activo o eliminado
Cambio de estado del almacén de referencia	Creado, actualizado, activo o eliminado
Ejecute el cambio de estado	Pendiente, en ejecución, detenida, completada, eliminada, fallida o cancelada
Cambio de estado del almacén de secuencias	Creado, actualizado, activo o eliminado
Cambio de estado de la tarea	Pendiente, en ejecución, detenida, completada, eliminada, fallida o cancelada
Cambio de estado del trabajo de importación de variantes	Enviado, en curso, cancelado, completado, fallido o completado con errores
Cambio de estado de Variant Store Share	Pendiente, activando, activo, borrado, eliminado, fallido

Nombre de evento	Valores de estado posibles
Cambio de estado de la tienda de variantes	Error al crear, actualizar, eliminar, eliminar o crear
Cambio de estado del flujo de trabajo compartido	Pendiente, activando, activo, borrado, eliminado, fallido
Cambio de estado del flujo de trabajo	Creación correcta, error de creación, eliminación correcta o error de eliminación

Estructura de mensaje de evento

HealthOmics proporciona la mejor forma de enviar mensajes de eventos de cambio de estado a EventBridge. El evento es un objeto con estructura JSON que también contiene detalles de metadatos. Puede utilizar los metadatos como entrada para recrear el evento o para obtener más información. Los eventos incluyen los siguientes campos:

- `version`— Actualmente 0 (cero) para todos los eventos.
- `id`— Se genera un UUID de la versión 4 para cada evento.
- `detail-type`— El tipo de evento que se envía.
- `account`— El Cuenta de AWS ID de 12 dígitos del propietario del bucket.
- `source`— Identifica el servicio que generó el evento.
- `time`— La hora en que ocurrió el evento.
- `region`— Identifica el Región de AWS del depósito.
- `resources`— Una matriz JSON que contiene el nombre de recurso de Amazon (ARN) del bucket.
- `detail`— Un objeto JSON que contiene información sobre el evento.

Los eventos de ejecución incluyen los siguientes campos:

- `uuid`— El identificador único universal de la ejecución.
- `workflowId`— Identificador del flujo de trabajo asociado a esta ejecución.
- `workflowName`— Nombre del flujo de trabajo asociado a esta ejecución.
- `runId`— Identificador de ejecución.

- `runName`— Nombre de ejecución.
- `runOutputUri`— El URI en el que la ejecución escribirá sus datos de salida.

Ejemplos de mensajes de eventos

El siguiente ejemplo es un evento para un cambio en el estado de la ejecución y muestra los campos adicionales.

```
{
    "version":"0",
    "id":"c0e540f4-df38-b986-86c1-3e3730f971fe",
    "detail-type":"Run Status Change",
    "source":"aws.omics",
    "account":"123456789012",
    "time":"2022-10-20T22:07:35Z",
    "region":"us-west-2",
    "resources":[
        "arn:aws:omics:us-west-2:123456789012:run/2101313"
    ],
    "detail":{
        "omicsVersion":"1.0.0",
        "arn":"arn:aws:omics:us-west-2:123456789012:run/2101313",
        "status":"COMPLETED",
        "uuid":"153893cd-097a-40ec-aec7-838a97cd2b21",
        "runId": "1234567",
        "runName": "run name",
        "runOutputUri": "s3://amzn-s3-demo-bucket/run-output/2101313",
        "workflowId": "1234567",
        "workflowName": "workflow name"
    }
}
```

El siguiente ejemplo es un evento para un cambio en el estado de una tarea.

```
{
    "version": "0",
    "id": "718d6817-c868-26d3-8ef0-0dc9b2ac73f4",
    "detail-type": "Task Status Change",
    "source": "aws.omics",
    "account": "123456789012",
    "time": "2024-10-30T09:05:44Z",
```

```
    "region": "us-west-2",
    "resources": ["arn:aws:omics:us-west-2:123456789012:task/8888888"],
    "detail": {
        "omicsVersion": "1.0.0",
        "arn": "arn:aws:omics:us-west-2:123456789012:task/8888888",
        "status": "COMPLETED",
        "runArn": "arn:aws:omics:us-west-2:123456789012:run/2101313",
        "runUuid": "153893cd-097a-40ec-aec7-838a97cd2b21",
        "runId": "1234567",
        "runName": "run name",
        "workflowId": "1234567",
        "workflowName": "workflow name"
    }
}
```

A continuación se muestra un ejemplo de un evento para un cambio de estado de un conjunto de lecturas.

```
{
  "version": "0",
  "id": "64ca0eda-9751-dc55-c41a-1bd50b4fc9b7",
  "detail-type": "Read Set Status Change",
  "source": "aws.omics",
  "account": "123456789012",
  "time": "2023-04-04T17:53:06Z",
  "region": "us-west-2",
  "resources": ["arn:aws:omics:us-west-2:123456789012:sequenceStore/1234567890/
readSet/3456789012"],
  "detail": {
     "omicsVersion": "1.0.0",
     "arn": "arn:aws:omics:us-west-2:123456789012:sequenceStore/1234567890/
readSet/3456789012",
     "sequenceStoreId" : "1234567890",
     "id": "3456789012",
     "status": "PROCESSING_UPLOAD"
  }
}
```

Se crea un evento similar para un trabajo de importación de una tienda de variantes.

```
{
    "version": "0",
    "id": "6a7e8feb-b491-4cf7-a9f1-bf3703467718",
```

```
    "detail-type": "Variant Store Status Change",
    "source": "aws.omics",
    "account": "123456789012",
    "time": "2015-12-22T18:43:48Z",
    "region": "us-east-1",
    "resources": ["arn:aws:omics:us-east-1:123456789012:myvariantstore2"],
    "detail": {
       "omicsVersion": "1.0.0",
       "arn": "arn:aws:omics:us-east-1:123456789012:myvariantstore2",
       "status": "CREATED",
       "storeId": "6710c5f02610",
       "storeName": "myvariantstore2"
    }
}
```

El siguiente es un evento relacionado con un cambio en el estado de un trabajo de importación.

```
{
    "version": "0",
    "id": "6a7e8feb-b491-4cf7-a9f1-bf3703467718",
    "detail-type": "Variant Import Job Status Change",
    "source": "aws.omics",
    "account": "123456789012",
    "time": "2015-12-22T18:43:48Z",
    "region": "us-east-1",
    "resources": ["arn:aws:omics:us-east-1:123456789012:my_variant_store/
b64ea9a3-459f-4b68-92c3-3ddb83209fe9"],
    "detail": {
       "omicsVersion": "1.0.0",
       "arn": "arn:aws:omics:us-east-1:123456789012:my_variant_store/
b64ea9a3-459f-4b68-92c3-3ddb83209fe9",
       "status": "COMPLETED",
       "jobId": "b64ea9a3-459f-4b68-92c3-3ddb83209fe9",
       "storeId": "a74869f91e20",
       "storeName": "my_variant_store"
    }
}
```

Resolución de problemas

Los siguientes temas pueden ayudarle a solucionar los problemas que surjan al utilizar HealthOmics flujos de trabajo y almacenes de datos.

Temas

- Solución de problemas de flujos de trabajo
- Solución de problemas de almacenamiento en caché de llamadas
- Solución de problemas de almacenes de datos
- Solución de problemas con Amazon Q CLI

Solución de problemas de flujos de trabajo

Temas

- ¿Cómo puedo solucionar una ejecución fallida?
- ¿Cómo soluciono los problemas de una tarea fallida?
- ¿Dónde puedo encontrar los registros del motor de las ejecuciones completadas correctamente?
- ¿Cómo puedo reducir el tamaño de los parámetros de entrada de un flujo de trabajo?
- ¿Por qué no se completa mi ejecución?

¿Cómo puedo solucionar una ejecución fallida?

Utilice la operación GetRunde la API para recuperar el motivo del error. Para obtener más información, consulte Motivos de error de ejecución.

¿Cómo soluciono los problemas de una tarea fallida?

Revise el código de error del mensaje de error de la tarea para comprender el error. Revise los registros de la tarea CloudWatch para ver los mensajes de registro detallados de la tarea. Si no recibes mensajes de registro detallados, puedes revisar tu flujo de trabajo para generar más declaraciones de registro. Para obtener más información, consulte Supervisión HealthOmics con CloudWatch registros.

¿Dónde puedo encontrar los registros del motor de las ejecuciones completadas correctamente?

HealthOmics publica los registros únicamente CloudWatch para las ejecuciones fallidas. Si una ejecución se completa correctamente, HealthOmics envía los registros del motor a su bucket de Amazon S3. Para obtener más información, consulte Inicia sesión en Amazon S3.

¿Cómo puedo reducir el tamaño de los parámetros de entrada de un flujo de trabajo?

Puede especificar hasta 50 KB de parámetros de entrada para un flujo de trabajo. Puede utilizar las importaciones de directorios o las hojas de muestra para mantenerse dentro de esta restricción de tamaño. Para obtener más información, consulte Administrar el tamaño de los parámetros de ejecución.

¿Por qué no se completa mi ejecución?

Si hay problemas con el código y los procesos no se han ejecutado correctamente, es posible que la ejecución deje de responder o se quede «atascada». Para obtener más información sobre cómo prevenir y atrapar las carreras que no responden, consulteGuía para ejecuciones que no responden.

Solución de problemas de almacenamiento en caché de llamadas

Los siguientes temas pueden ayudarte a solucionar los problemas que surjan con el almacenamiento en caché de llamadas.

Temas

- ¿Por qué mi ejecución no se guarda en la caché?
- ¿Por qué una tarea no utiliza la entrada de caché?
- ¿Por qué está desactivado el almacenamiento en caché de llamadas de una tarea?

¿Por qué mi ejecución no se guarda en la caché?

1. Comprueba que la ejecución esté configurada para usar una caché. Para ello, marca el campo cacheID en la respuesta a la operación de la GetRun API. Mediante la CLI, ejecute este comando:`aws omics get-run –id <run_id>`.

2. Si la ejecución se realizó correctamente, compruebe que el comportamiento de la caché devuelto en la GetRun respuesta sea CACHE_ALWAYS. Si el comportamiento de la caché se establece en CACHE_ON_FAILURE, las ejecuciones solo se guardarán en la memoria caché cuando fallen.

¿Por qué una tarea no utiliza la entrada de caché?

<cache_id><cache_uuid>En el grupo de `/aws/omics/WorkflowLog` CloudWatch registros, abra el flujo de registros de la caché de ejecución: RunCache//.

1. Compruebe que una ejecución anterior haya creado una entrada de caché para la tarea que esperaba almacenar en caché. Las ejecuciones que se hayan guardado en la memoria caché se registrarán con un mensaje de registro de CACHE_ENTRY_CREATED.
2. Localice el registro CACHE_MISS de la tarea y ejecútelo completada. Si no hay ninguna entrada de registro, compruebe que la ejecución se configuró para usar la memoria caché.
3. Si se creó una entrada en la caché, compruebe que la CPUs memoria GPUs y el resumen del contenedor son idénticos para ambas tareas. El ARN de la tarea que creó la entrada de caché se encuentra en el mensaje de registro.
4. Si los requisitos de procesamiento de ambas tareas coinciden, compruebe que las entradas no hayan cambiado entre las tareas. Para ello, abra los registros del motor. Si la ejecución tiene el estado FALLIDA, los registros estarán en Cloudwatch Log Group/aws/omics/WorkflowLog. De lo contrario, los registros del motor se encuentran en el directorio de salida de la ejecución.

¿Por qué está desactivado el almacenamiento en caché de llamadas de una tarea?

Compruebe si la tarea está configurada para excluirse del almacenamiento en caché mediante las funciones del motor de flujo de trabajo:

- Para los flujos de trabajo de WDL: compruebe si la tarea tiene el valor volátil establecido `true` en la sección de metadatos
- Para los flujos de trabajo de Nextflow: compruebe si la tarea tiene la directiva de caché establecida en `false`
- Para los flujos de trabajo de CWL: compruebe si la tarea tiene la función EnableReuse configurada como `false` WorkReuse

Solución de problemas de almacenes de datos

Temas

- ¿Por qué GetObject falla S3 en mi equipo de lectura?
- ¿Por qué no puedo ver mi tienda de anotaciones o mi tienda de variantes en Athena?
- ¿Por qué no puedo acceder a mi almacén de datos en Athena?

¿Por qué GetObject falla S3 en mi equipo de lectura?

Por lo general, el error se debe a la falta de un permiso. El permiso de lectura del almacén de secuencias S3 es una configuración bidireccional que requiere tanto la política de acceso al almacén de secuencias S3 para permitir el acceso como que el principal de IAM tenga una política adjunta que permita el acceso. Para obtener más información sobre los requisitos de la política, consulte. Permisos de acceso a los datos mediante Amazon S3 URIs Compruebe que estén implementadas las siguientes configuraciones:

- La política de acceso de S3 al almacén de secuencias ha permitido explícitamente el acceso a la cuenta principal de IAM o a la raíz de la cuenta de la entidad principal.
- Compruebe que el principal de IAM tenga una política que otorgue permisos explícitos al recurso al que se accede. Tenga en cuenta que la política principal de IAM debe usar el ARN del punto de acceso y no la ruta basada en el alias del punto de acceso al definir los permisos y que el ARN está en esa condición y no se usa para especificar un recurso.
- Si tu tienda usa una clave gestionada por el cliente (CMK-KMS), asegúrate de que el director de IAM tenga kms: permisos para descifrar la clave. Consulta la guía de acceso multicuenta de KMS para configurar el uso en todas las cuentas.

Si tiene una política que utiliza controles de acceso basados en etiquetas, asegúrese de lo siguiente:

- Asegúrese de que el almacén de secuencias haya terminado de sincronizar las etiquetas. Para ello, el estado de la tienda debe ser active y noupdating.
- Asegúrese de que no haya errores tipográficos en la clave de la etiqueta o en el valor de la clave en el conjunto de lecturas y en la política.

¿Por qué no puedo ver mi tienda de anotaciones o mi tienda de variantes en Athena?

En Lake Formation, asegúrate de crear un enlace de recursos basado en la tienda que compartiste contigo. Una vez que hayas creado un enlace a un recurso al que tengas permiso de acceso, la tienda debería estar visible en Athena. Para obtener más información, consulte Configuración de Lake Formation para usar HealthOmics.

¿Por qué no puedo acceder a mi almacén de datos en Athena?

Si su almacén de anotaciones o variantes está visible, pero recibe un mensaje de error que indica que se ha denegado el acceso, compruebe qué versión del motor de consultas está utilizando. Solo se admiten las consultas que se ejecutan con la versión 3 del motor. Para obtener más información sobre las versiones del motor de consultas de Athena, consulte la documentación de Amazon Athena.

Solución de problemas con Amazon Q CLI

Amazon Q CLI puede ayudarle a agilizar el proceso de solución de problemas al:

- Analizar las ejecuciones del flujo de trabajo y depurar los errores de las tareas
- Recopilación de registros y mensajes de error relevantes
- Creación de casos de AWS Support con todos los registros de depuración necesarios adjuntos
- Elimina la información de identificación personal (PII) de la información enviada a Support AWS

Para obtener más información sobre el uso de Amazon Q CLI AWS HealthOmics para solucionar problemas y crear casos de soporte, consulte el tutorial de IA generativa de HealthOmics Agentic en. GitHub

> **Warning**
>
> Cuando trabaje con Amazon Q CLI, revise todo el contenido generado y las acciones propuestas antes de continuar. Proporcione comentarios para mejorar la calidad de la respuesta y adaptarla a los requisitos de su flujo de trabajo. Para obtener más información, consulte Consideraciones de seguridad y prácticas recomendadas para Amazon Q.

Cuotas para AWS HealthOmics

AWS rellena tu cuenta con los valores predeterminados de las HealthOmics cuotas. A menos que se indique lo contrario, el valor de cada cuota es el valor máximo por región.

> **Important**
>
> Puedes solicitar aumentos en la mayoría de las cuotas de servicio y de API. Consulte los siguientes temas para obtener información detallada.

Temas

- HealthOmics cuotas de servicio
- HealthOmics cuotas de tamaño fijo
- HealthOmics Cuotas de API

HealthOmics cuotas de servicio

La siguiente tabla muestra las cuotas HealthOmics de servicio, junto con sus valores predeterminados. Para ver las cuotas actuales de cada región, abra la consola Service Quotas.

> **Important**
>
> Puede solicitar un aumento a una cuota ajustable mediante la consola Service Quotas.

Para obtener más información sobre las cuotas de servicio, consulte Solicitar un aumento de cuota en la Guía del usuario de Service Quotas. Para una cuota que no esté disponible en la consola Service Quotas, utilice el formulario de aumento de cuota.

Name	Predeterminado	Ajusta	Description (Descripción)
Análisis: se almacena el número máximo de anotaciones	Cada región admitida: 10	Sí	El número máximo de almacenes de anotaciones en la región actual AWS

Name	Predeterminado	Ajusta	Description (Descripción)
Trabajos simultáneos de importación de almacenes de variantes o anotaciones	Cada región admitida: 5	Sí	El número máximo de trabajos de importación simultáneos en la región actual AWS
Análisis: número máximo de archivos por trabajo de importación de almacén de variantes	Cada región admitida: 1000	Sí	El número máximo de archivos por trabajo de importación de variantes en la región actual AWS
Análisis: número máximo de veces compartidas por almacén de anotacion es	Cada región admitida: 10	Sí	El número máximo de recursos compartidos por almacén de anotaciones en la región actual AWS
Análisis: número máximo de acciones por tienda de variantes	Cada región admitida: 10	Sí	El número máximo de acciones por almacén de variantes en la región actual AWS
Análisis: tamaño máximo de cada archivo en un trabajo de importación de variantes	Cada región admitida: 20 gigabytes	Sí	El tamaño máximo de un archivo en un trabajo de importación de variantes en la AWS región actual
Análisis: tamaño máximo de cada archivo en un trabajo de importación de anotaciones	Cada región admitida: 20 gigabytes	Sí	El tamaño máximo de un archivo en un trabajo de importación de anotacion es en la región actual AWS
Análisis: se almacena el número máximo de variantes	Cada región admitida: 10	Sí	El número máximo de tiendas de variantes en la región actual AWS

Name	Predeterminado	Ajusta	Description (Descripción)
Análisis: número máximo de versiones por almacén de anotaciones	Cada región admitida: 10	Sí	El número máximo de versiones por almacén de anotaciones en la región actual AWS
Configuraciones: configuraciones máximas	Cada región admitida: 10	Sí	El número máximo de configuraciones en la AWS región actual.
Almacenamiento: número máximo de trabajos de activación de conjuntos de lectura simultáneos	Cada región admitida: 25	Sí	El número máximo de trabajos de activación de conjuntos de lectura simultáneos en la región actual AWS
Almacenamiento: número máximo de trabajos de exportación simultáneos de secuencias y almacenes de referencia	Cada región admitida: 5	Sí	El número máximo de trabajos de exportación simultáneos desde un almacén de secuencias o referencias de la región actual AWS
Trabajos de importación de secuencias o referencias simultáneos	Cada región admitida: 5	Sí	El número máximo de trabajos de importaci ón simultáneos para un almacén de secuencias o referencias en la región actual AWS
Almacenamiento: número máximo de conjuntos de lectura por almacén de secuencias	Cada región admitida: 1 000 000	Sí	El número máximo de conjuntos de lectura en un almacén de secuencia s en la región actual AWS

Name	Predeterminado	Ajusta	Description (Descripción)
Almacenamiento: número máximo de referencias por almacén de referencias	Cada región admitida: 50	Sí	El número máximo de referencias en un almacén de referencias de la AWS región actual
Almacenamiento: número máximo de almacenes secuenciales	Cada región admitida: 20	Sí	El número máximo de almacenes de secuencias en la AWS región actual
Flujos de trabajo: máximo activo GPUs	Cada región admitida: 12	Sí	El número máximo de activos simultáneos GPUs en la AWS región actual. En us-east-1 y us-west-2, las solicitudes de aumento de cuota para valores de hasta 500 se aprueban automátic amente.
Flujos de trabajo: máximo de ejecucion es activas simultáneas mediante el almacenamiento de ejecuciones dinámicas	Cada región admitida: 50	Sí	El número máximo de ejecuciones activas que utilizan el almacenam iento de ejecucion es dinámicas en la AWS región actual. Las solicitudes de aumento de las cuotas por hasta un máximo de 200 valores se aprueban automáticamente.

Name	Predeterminado	Ajusta	Description (Descripción)
Flujos de trabajo: máximo de ejecucion es activas simultáneas mediante el almacenamiento de ejecuciones estáticas	Cada región admitida: 10	Sí	El número máximo de ejecuciones activas que utilizan el almacenam iento de ejecuciones estáticas en la AWS región actual. Las solicitudes de aumento de las cuotas por hasta un máximo de 50 valores se aprueban automátic amente.
Flujos de trabajo: número máximo de tareas simultáneas por ejecución	Cada región admitida: 25	Sí	El número máximo de tareas simultáneas en cada ejecución en la AWS región actual. En us-east-1 y us-west-2, las solicitudes de aumento de cuota para valores de hasta 100 se aprueban automáticamente.
Flujos de trabajo: duración máxima de ejecución	Cada región compatible: 604.800 segundos	Sí	La duración máxima de ejecución del flujo de trabajo en la región actual AWS .
Flujos de trabajo: número máximo de ejecuciones (activas o inactivas)	Cada región admitida: 100 000	Sí	El número máximo de ejecuciones (activas o inactivas) en la AWS región actual.

Name	Predeterminado	Ajusta	Description (Descripción)
Flujos de trabajo: número máximo de veces compartidas por flujo de	Cada región admitida: 100	Sí	El número máximo de comparticiones por flujo de trabajo en la AWS región actual
Flujos de trabajo: capacidad máxima de almacenamiento estático por ejecución	Cada región compatible: 9.600	Sí	La capacidad máxima de almacenamiento de ejecución estática en gibibytes (GiB) para cada ejecución en la región actual. AWS En us-east-1 y us-west-2, las solicitudes de aumento de cuota para valores de hasta 50 000 se aprueban automátic amente.
Flujos de trabajo: flujos de trabajo máximos	Cada región admitida: 1000	Sí	El número máximo de flujos de trabajo en la región actual. AWS
Flujos de trabajo: transacciones por segundo (TPS) de la StartRun operación	Cada región admitida: 5	Sí	El número máximo de transacciones por segundo (TPS) de la StartRun operación en la región actual AWS .

HealthOmics cuotas de tamaño fijo

Además deHealthOmics cuotas de servicio, HealthOmics incluye cuotas que tienen tamaños fijos. No puede solicitar un aumento de estos valores.

A menos que se indique lo contrario, cada cuota indica el valor máximo por región.

Temas

- HealthOmics analíticas: cuotas de tamaño fijo
- HealthOmics almacenamiento: cuotas de tamaño fijo
- HealthOmics cuotas de tamaño fijas en el flujo de trabajo
- HealthOmics Cuotas de tamaño fijo del flujo de trabajo Ready2Run

HealthOmics analíticas: cuotas de tamaño fijo

En la siguiente tabla se muestran los valores máximos admitidos para las cuotas de análisis. Estos valores no se pueden ajustar.

Name	Description (Descripción)	Máximo	Ajustable: sí/no
Análisis: número máximo de archivos por trabajo de importación del almacén de anotaciones	El número máximo de archivos por trabajo de importación de anotaciones.	1	No

HealthOmics almacenamiento: cuotas de tamaño fijo

En la siguiente tabla se muestran los valores máximos admitidos para los archivos de almacenamiento. Estos valores no se pueden ajustar.

Name	Description (Descripción)	Máximo	Ajustable: sí/no
Almacenamiento: tamaño máximo de la política de recursos de acceso a S3	El tamaño máximo de la política de recursos de acceso a S3	15 KB	No
Almacenamiento: número máximo de	El número máximo de claves de etiquetas	5	No

Name	Description (Descripción)	Máximo	Ajustable: sí/no
etiquetas a nivel de conjunto propagadas	de nivel establecido, por tienda, que se propagan al objeto S3		
Almacenamiento: cantidad máxima de conjuntos de lecturas por trabajo de activación	El número máximo de conjuntos de lecturas por trabajo de activación.	20	No
Almacenamiento: cantidad máxima de conjuntos de lecturas por trabajo de exportación	El número máximo de conjuntos de lecturas por trabajo de exportación.	100	No
Almacenamiento: cantidad máxima de conjuntos de lecturas por trabajo de importación	El número máximo de conjuntos de lecturas por trabajo de importación.	100	No
Almacenamiento: número máximo de almacenes de referencia	El número máximo de almacenes de referencia.	1	No
Almacenamiento: tamaño máximo de la pieza para una carga directa	El tamaño máximo de pieza que se puede cargar directame nte a un almacén de secuencias.	100 MB	No

Name	Description (Descripción)	Máximo	Ajustable: sí/no
Almacenamiento: máximo de partes en el archivo para la carga directa	El número máximo de partes de un archivo que se pueden cargar directamente a un almacén de secuencias.	10 000	No
Almacenamiento: tamaño máximo de referencia	El tamaño máximo de un archivo de referencia que se puede importar a un almacén de referencias.	15 GB	No
Almacenamiento: tamaño máximo de la fuente del conjunto de lectura	El tamaño máximo de un único archivo fuente de un conjunto de lecturas que se puede importar a un almacén de secuencias.	976 GB	No

HealthOmics cuotas de tamaño fijas en el flujo de trabajo

La siguiente tabla muestra los valores máximos admitidos para las cuotas de flujo de trabajo. Estos valores no se pueden ajustar.

Name	Description (Descripción)	Tamaño máximo	Ajustable: sí/no
Flujos de trabajo: número máximo de grupos de ejecución	El número máximo de grupos de ejecuciones.	1 000	No

Name	Description (Descripción)	Tamaño máximo	Ajustable: sí/no
Flujos de trabajo: caché de ejecución máxima	El número máximo de cachés de ejecución que puedes crear para una cuenta. Una o más ejecuciones pueden compartir la misma caché de ejecución. No hay una cuota para el número de ejecuciones que se HealthOmics pueden almacenar en caché por cuenta.	1 000	No
Flujos de trabajo: número máximo de versiones de flujo	El número máximo de versiones de flujo de trabajo por flujo de trabajo.	1 000	No
Flujos de trabajo: tamaño del contenedor de la instancia de CPU	El tamaño máximo de la imagen del contenedor para una instancia de CPU.	45 GiB	No
Flujos de trabajo: tamaño del contenedor de instancias de GPU	El tamaño máximo de la imagen del contenedor para una instancia de GPU.	95 GiB	No
Memoria compartida de la instancia de GPU /dev/shm	La cantidad máxima de memoria compartida por instancia de GPU.	8 GB por GPU	No

Name	Description (Descripción)	Tamaño máximo	Ajustable: sí/no
Flujos de trabajo: ejecute un archivo de parámetros	El tamaño máximo de un archivo de parámetros de ejecución.	50 000 bytes	No
Flujos de trabajo: archivo de plantilla de parámetros de flujo de	El número máximo de entradas y el tamaño máximo de archivo de un archivo de plantilla de parámetros de flujo de trabajo. Esta cuota se aplica a los flujos de trabajo que cree mediante la consola o la API.	1000 entradas, 400 KB	No
Flujos de trabajo: tamaño del archivo de definición del flujo de trabajo: API	El tamaño máximo del archivo de definición del flujo de trabajo al crear el flujo de trabajo mediante la operación de API o un AWS SDK.	100 MB	No
Flujos de trabajo: tamaño del archivo de definición del flujo de trabajo: consola (carga directa)	El tamaño máximo del archivo de definición del flujo de trabajo que puede proporcionar como carga directa al crear el flujo de trabajo mediante la consola.	4,4 MB	No

Name	Description (Descripción)	Tamaño máximo	Ajustable: sí/no
Flujos de trabajo - Tamaño del archivo de definición del flujo de trabajo - Consola (carga desde Amazon S3)	El tamaño máximo del archivo de definición del flujo de trabajo que puede proporcionar como carga desde Amazon S3 al crear el flujo de trabajo mediante la consola.	100 MB	No
Flujos de trabajo: tamaño del repositorio	El tamaño máximo de un repositorio de código externo.	1 GiB	No
Flujos de trabajo: tamaño de archivo individual del repositorio	El tamaño máximo de un archivo individual de un repositorio de código externo.	100 MiB	No
Flujos de trabajo: tamaño del archivo README	El tamaño máximo de un archivo README.	500 KiB	No

Para obtener sugerencias sobre cómo reducir el tamaño del archivo de parámetros de ejecución, consulteAdministrar el tamaño de los parámetros de ejecución.

HealthOmics Cuotas de tamaño fijo del flujo de trabajo Ready2Run

Cada flujo de trabajo de Ready2Run tiene un tamaño máximo de archivo de entrada. En la siguiente tabla, las unidades de tamaño de archivo se muestran en Gibibytes (GiB). Estos tamaños máximos de archivo no se pueden ajustar.

Nombre del flujo de trabajo Ready2Run	Tamaño máximo del archivo de entrada (GiB)	Ajustable (sí/no)
AlphaFold para 601-1200 residuos	1	No
AlphaFold para un máximo de 600 residuos	1	No
Bases2Fastq para 2x150	1 000	No
Bases2Fastq para 2x300	1 000	No
Bases2Fastq para 2x75	500	No
ESMFold para hasta 800 residuos	1	No
GATK-BP fq2bam	64	No
Línea germinal bam2vcf de GATK-BP para un genoma de 30 veces	39	No
Línea germinal gatK-BP fq2vcf para un genoma de 30 veces	64	No
GATK-BP ES somático bam2vcf	86	No
NVIDIA Parabricks BAM2 FQ2 BAM WGS hasta 30 veces	80	No
NVIDIA BAM2 FQ2 Parabricks BAM WGS hasta 50 veces	120	No
NVIDIA BAM2 FQ2 Parabricks BAM WGS hasta 5 veces	20	No

Nombre del flujo de trabajo Ready2Run	Tamaño máximo del archivo de entrada (GiB)	Ajustable (sí/no)
NVIDIA FQ2 Parabricks BAM WGS hasta 30 veces	71	No
NVIDIA FQ2 Parabricks BAM WGS hasta 50 veces	137	No
NVIDIA FQ2 Parabricks BAM WGS hasta 5 veces	13	No
NVIDIA Parabricks Germline WGS hasta 30 veces DeepVariant	71	No
NVIDIA Parabricks Germline WGS hasta 50 veces DeepVariant	137	No
NVIDIA Parabricks Germline WGS hasta 5 veces DeepVariant	12	No
NVIDIA Parabricks Germline WGS hasta 30 veces HaplotypeCaller	71	No
NVIDIA Parabricks Germline WGS hasta 50 veces HaplotypeCaller	137	No
NVIDIA Parabricks Germline WGS hasta 5 veces Haplotype Caller	13	No
NVIDIA Parabricks Somatic Mutect2 WGS hasta 50 veces	196	No

Nombre del flujo de trabajo Ready2Run	Tamaño máximo del archivo de entrada (GiB)	Ajustable (sí/no)
sc RNAseq con Kallisto BUStools	119	No
sc RNAseq con salmón Alevin-Fry	119	No
sc RNAseq con STARsolo	119	No
Sentieon Germline BAM WES hasta 300 veces	9	No
Sentieon Germline BAM WGS hasta 32 veces	18	No
Sentieon Germline FASTA WES hasta 100 veces	5	No
Sention Germline FASTQ WES hasta 300 veces	26	No
Sention Germline FASTA WGS hasta 32 veces	51	No
LongRead Sention para OMT	25	No
Sentieon LongRead para PacBio HiFi	58	No
Sentieon Somatic WES	50	No
Sentieon Somatic WGS	113	No
Ultima Genomics hasta 40 veces DeepVariant	91	No

HealthOmics Cuotas de API

HealthOmics tiene las siguientes cuotas relacionadas con las operaciones de la API. Cuando se indique, la cuota es ajustable. Para solicitar un aumento, utilice el formulario de aumento de cuota.

Para cada operación de API de la lista, la cuota es el número máximo de transacciones por segundo (TPS) para esa operación de API en cada región.

Temas

- Cuotas generales de API
- Cuotas de API de almacenamiento
- Cuotas de la API de flujo
- Cuotas de la API de análisis

Cuotas generales de API

En la siguiente tabla, se enumeran las operaciones generales de la API que se aplican a más de una categoría (almacenamiento, flujos de trabajo y análisis).

Operación de la API	TPS máximo predeterminado	Ajustable (sí/no)
AcceptShare, CreateShare, DeleteShare, GetShare, ListShares	1 TPS	Sí

Cuotas de API de almacenamiento

En la siguiente tabla se enumeran las operaciones de la API de almacenamiento.

Funcionamiento de la API de almacenamiento	TPS máximo predeterminado	Ajustable (sí/no)
CreateSequenceStore, UpdateSequenceStore, DeleteSequenceStore,	1 TPS	Sí

Funcionamiento de la API de almacenamiento	TPS máximo predeterminado	Ajustable (sí/no)
CreateReferenceStore, DeleteReferenceStore		
BatchDeleteReadSet, DeleteReference	1 TPS	Sí
CreateMultipartReadSetUpload, CompleteMultipartReadSetUpload, AbortMultipartReadSetUpload	1 TPS	No
Obtiene S3AccessPolicy, pone 3, elimina S3 AccessPolicy AccessPolicy	1 TPS	Sí
GetReference	10 TPS	Sí
UploadReadSetPart	10 TPS	Sí
GetReadSet	30 TPS	Sí
GetSequenceStore, ListSequenceStores	5 TPS	Sí
GetReadSetMetadata, ListReadSets	5 TPS	Sí
StartReadSetImportJob, GetReadSetImportJob, ListReadSetImportJobs	5 TPS	Sí
StartReadSetExportJob, GetReadSetExportJob, ListReadSetExportJobs	5 TPS	Sí
ListReferenceStores	5 TPS	Sí

Funcionamiento de la API de almacenamiento	TPS máximo predeterminado	Ajustable (sí/no)
StartReferencetImportJob, GetReferenceImportJob, ListReferenceImportJobs	5 TPS	Sí
ListReferences, GetReferenceMetadata	5 TPS	Sí
StartReadsetActivationJob	5 TPS	Sí
ListReadsetActivationJobs, GetReadSetActivationJob	5 TPS	Sí
ListMultipartReadSetUploads, ListReadSetUploadParts	5 TPS	Sí
TagResource, UntagResource, ListTagsForResource	5 TPS	Sí

Cuotas de la API de flujo

En la siguiente tabla se muestran las operaciones de la API del flujo de trabajo.

Operación de API de flujo de trabajo	TPS máximo predeterminado	Ajustable (sí/no)
StartRun	1 TPS	Sí
CreateWorkflow	5 TPS	Sí
CancelRun, DeleteRun, GetRun, GetRunTask, ListRunTasks, ListRuns	10 TPS	Sí
CreateRunGroup, DeleteRunGroup, GetRunGroup,	10 TPS	Sí

Operación de API de flujo de trabajo	TPS máximo predeterminado	Ajustable (sí/no)
ListRunGroups, UpdateRun Group		
CreateRunCache, UpdateRun Cache, DeleteRunCache, GetRunCache, ListRunCaches	10 TPS	Sí
DeleteWorkflow, GetWorkfl ow, ListWorkflows, UpdateWor kflow	10 TPS	Sí

Cuotas de la API de análisis

En la siguiente tabla se enumeran las operaciones de la API de análisis.

Funcionamiento de la API de análisis	TPS máximo predeterminado	Ajustable (sí/no)
CreateVariantStore, DeleteVar iantStore, GetVariantStore, ListVariantStores, UpdateVar iantStore	1 TPS	No
StartVariantImportJob, CancelVariantImportJob, GetVariantImportJob, ListVaria ntImportJobs	1 TPS	No
CreateAnnotationStore, DeleteAnnotationStore, GetAnnotationStore, ListAnnot ationStores, UpdateAnn otationStore	1 TPS	No

Funcionamiento de la API de análisis	TPS máximo predeterminado	Ajustable (sí/no)
StartAnnotationImportJob, ListAnnotationImportJobs, GetAnnotationImportJob, CancelAnnotationImportJob	1 TPS	No

Historial de documentos de la Guía HealthOmics del usuario

En la siguiente tabla se describen las versiones de la documentación de HealthOmics.

Cambio	Descripción	Fecha
AWS HealthOmics las tiendas de variantes y las tiendas de anotaciones ya no están abiertas a nuevos clientes.	AWS HealthOmics las tiendas de variantes y las tiendas de anotaciones ya no están abiertas a nuevos clientes. Para obtener más información, consulta el cambio de disponibilidad de los almacenes de AWS HealthOmics variantes y de anotaciones.	7 de noviembre de 2025
AWS HealthOmics Las tiendas de variantes y de anotacion es dejarán de estar abiertas a nuevos clientes a partir del 7 de noviembre de 2025.	AWS HealthOmics Las tiendas de variantes y de anotaciones dejarán de estar abiertas a nuevos clientes a partir del 7 de noviembre de 2025. Si quieres usar tiendas de variantes o tiendas de anotaciones, regístrat e antes de esa fecha. Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulta los cambios en la disponibilidad de los almacenes de AWS HealthOmics variantes y de anotaciones.	7 de octubre de 2025
Nuevas características	HealthOmics se agregó soporte para flujos de trabajo	28 de agosto de 2025

	para sincronizar un repositor io privado de Amazon ECR con un registro principal. Para obtener más informaci ón, consulte Imágenes de contenedores para flujos de trabajo privados en. HealthOmics	
Nuevas funciones de integraci ón de repositorios y README	Se agregó soporte para crear flujos de trabajo a partir de repositorios de código externos y archivos README.	24 de julio de 2025
Nuevas características	HealthOmics se agregó soporte para la interpolación automática de parámetros de Nextflow. Para obtener más información, consulte Archivos de plantillas de parámetros para flujos de trabajo. HealthOmics	27 de junio de 2025
Nuevas características	HealthOmics se ha añadido compatibilidad con los flujos de trabajo para interpolar los parámetros de ejecución de un archivo de definición de flujo de trabajo de la WDL. Para obtener más información, consulte Archivos de plantilla s de parámetros para flujos de trabajo. HealthOmics	30 de mayo de 2025

Nuevas características	HealthOmics se ha añadido compatibilidad con el control de versiones de los flujos de trabajo. Para obtener más información, consulte Control de versiones de flujos de trabajo en. HealthOmics	18 de abril de 2025
Nuevas características	HealthOmics se agregó un rendimiento elástico para el almacenamiento de ejecución dinámica. Para obtener más información, consulte Ejecutar tipos de almacenamiento en HealthOmics.	16 de abril de 2025
Nuevas características	HealthOmics se agregaron controles de acceso basados en atributos para las ubicaciones de Sequence Store S3 y la capacidad de sincronizar hasta cinco etiquetas de conjuntos de lectura con un objeto de Sequence Store S3. Para obtener más información, consulte Creación de un almacén de secuencias. HealthOmics	22 de noviembre de 2024

Nuevas características	HealthOmics se agregó soporte para el almacenamiento en caché de llamadas, también conocido como currículum, para flujos de trabajo privados. Para obtener más información, consulta Almacenamiento de llamadas en caché.	20 de noviembre de 2024
Nuevas características	HealthOmics se agregaron nuevos campos de API para ayudarte a mapear entre los trabajos de entrada del almacén de secuencias y los conjuntos de lectura.	29 de agosto de 2024
Nuevas características	HealthOmics se agregó soporte para administrar las versiones de Nextflow. Para obtener más información, consulte las versiones de Nextflow.	14 de agosto de 2024
Nuevas características	HealthOmics se agregó soporte para flujos de trabajo compartidos y almacenamiento dinámico de ejecuciones.	30 de abril de 2024
Nuevas características	HealthOmics se agregó soporte para el acceso de Amazon S3 a los almacenes de referencia y secuencias, y soporte para SHA256 ETags.	15 de abril de 2024

Nuevas características	HealthOmics se agregaron etiquetas de entidad (ETags) para los almacenes de secuencias.	6 de octubre de 2023
Nuevas características	HealthOmics se agregó el control de versiones de los almacenes de anotaciones y el uso compartido de almacenes analíticos.	15 de agosto de 2023
Nuevas características	HealthOmics se agregó el lenguaje común de flujo de trabajo (CWL) como idioma compatible con los flujos de trabajo. HealthOmics	30 de junio de 2023
Nuevas características	HealthOmics se agregaron nuevos flujos de trabajo Ready2Run, la compatibi lidad con GPU para los flujos de trabajo, el análisis de datos para los almacenes de anotaciones, la carga directa al HealthOmics almacenam iento y la integración con. EventBridge	15 de mayo de 2023
Nueva política administrada	HealthOmics se ha añadido una nueva política gestionada que proporciona acceso total. Para obtener más información, consulte Políticas administr adas por AWS.	23 de febrero de 2023

Nueva política administrada	HealthOmics agregó una nueva política administrada que limita el acceso a solo lectura. Para obtener más información, consulte Políticas administradas por AWS.	29 de noviembre de 2022
Versión inicial	Versión inicial de la guía HealthOmics del usuario	29 de noviembre de 2022

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.