Guia do usuário

AWS HealthOmics

Versão latest

AWS HealthOmics: Guia do usuário

Table of Contents

O que é AWS HealthOmics?

AWS HealthOmics é um serviço qualificado pela HIPAA que acelera os testes de diagnóstico clínico, a descoberta de medicamentos e a pesquisa agrícola ao gerenciar totalmente a infraestrutura complexa por trás de seus fluxos de trabalho de bioinformática. HealthOmics suporta linguagens de fluxo de trabalho padrão do setor (WDL, Nextflow, CWL) e dimensiona perfeitamente a infraestrutura de bioinformática para suportar dados de dezenas de milhares de testes por dia, tudo com custo previsível por amostra. HealthOmics lida com as complexidades técnicas, como gerenciar recursos computacionais e manter mecanismos de fluxo de trabalho, para que você possa se concentrar inteiramente em descobertas científicas.

Tópicos

Aviso importante

HealthOmics destina-se somente à transferência, armazenamento, formatação ou exibição de dados e ao fornecimento de infraestrutura e suporte de configuração para gerenciar fluxos de trabalho. HealthOmics não substitui o aconselhamento, diagnóstico ou tratamento médico profissional e não se destina a curar, tratar, mitigar, prevenir ou diagnosticar qualquer doença ou condição de saúde. Você é responsável por instituir a avaliação humana como parte de qualquer uso de AWS HealthOmics, inclusive em associação com, qualquer produto de terceiros destinado a informar a tomada de decisões clínicas.

HealthOmics features

Casos de uso primários para HealthOmics:

- Diagnóstico clínico — Crie e escale fluxos de trabalho de testes de diagnóstico com custos previsíveis e infraestrutura totalmente gerenciada que cresce com o volume de testes.
- Descoberta de medicamentos — Acelere a pesquisa terapêutica orquestrando modelos de base biológica em grande escala, permitindo a rápida iteração entre milhões de candidatos em potencial.
- Pesquisa agrícola — Melhore as características das culturas, como tolerância à seca e resistência a pragas, por meio de fluxos de trabalho baseados em IA que melhoram a segurança alimentar e a produtividade agrícola.

Principais benefícios de HealthOmics:

- Escalabilidade — Dimensione fluxos de trabalho em mais de 100.000 v simultâneos CPUs para suportar dezenas de milhares de testes diários sem gerenciamento de infraestrutura e custo previsível por amostra.
- Concentre-se na ciência, não na infraestrutura — use linguagens de fluxo de trabalho familiares e, APIs ao mesmo tempo, gerencie AWS automaticamente a orquestração da infraestrutura e o gerenciamento de dados nos bastidores.
- Mantenha a conformidade — trilhas de auditoria abrangentes, rastreamento de proveniência de dados e infraestrutura qualificada pela HIPAA projetada para fluxos de trabalho clínicos — tudo isso out-of-the-box — apoiam o desenvolvimento de soluções que atendam aos requisitos regulatórios.

HealthOmics consiste em três componentes principais:

- HealthOmics fluxos de trabalho — Execute cálculos de bioinformática em uma infraestrutura provisionada e escalada automaticamente.
- HealthOmics armazenamento — armazene e compartilhe petabytes de dados genômicos de forma eficiente a um baixo custo por gigabase.
- HealthOmics análise — Prepare dados genômicos para análises multiômicas e multimodais.

Use esses componentes de forma independente ou combine-os para obter uma end-to-end solução.

HealthOmics conceitos

Este tópico aborda as definições dos principais conceitos e termos específicos de HealthOmics, para ajudá-lo a entender a terminologia de HealthOmics uso deste guia.

Tópicos

Fluxos de trabalho

Com HealthOmics os fluxos de trabalho, você pode processar e analisar seus dados genômicos.

- Fluxo de trabalho — A definição geral de um processo de ponta a ponta, incluindo parâmetros e referências a ferramentas. As definições de fluxo de trabalho podem ser expressas como WDL, Nextflow ou CWL. Cada fluxo de trabalho criado tem um identificador exclusivo.
- Executar — Uma única invocação de um fluxo de trabalho. Uma execução individual usa seus dados de entrada definidos e produz uma saída. Cada execução criada tem um identificador exclusivo.
- Tarefa — Os processos individuais em uma execução. HealthOmics Os fluxos de trabalho usam essas especificações de computação definidas para executar sua tarefa. Cada tarefa tem um identificador exclusivo.
- Grupo de execuções — Um grupo de execuções para as quais você pode definir o máximo de vCPU, a duração máxima ou o máximo de execuções simultâneas para ajudar a limitar os recursos computacionais usados por execução. Você pode especificar e configurar prioridades para suas execuções dentro de um grupo de corridas. Por exemplo, você pode especificar que uma execução de alta prioridade será executada antes de uma de menor prioridade, criando uma fila prioritária. É opcional usar um grupo de execução, e cada grupo de execução tem um identificador exclusivo.

Armazenamento

O armazenamento de dados é separado em armazenamentos de sequências, para suas sequências genômicas e informações relacionadas, e um armazenamento de referência, para todos os seus

genomas de referência. Os termos a seguir descrevem as implementações que são específicas do. HealthOmics

- Armazenamento de sequências — Um armazenamento de dados para o armazenamento de arquivos genômicos. Você pode ter um ou mais armazenamentos de sequências dentro HealthOmics. As permissões de acesso e a AWS KMS criptografia podem ser definidas em um armazenamento de sequências para controlar quem tem acesso aos dados.
- Conjunto de leitura — Um conjunto de leitura é uma abstração das leituras genômicas, que são armazenadas nos formatos FASTQ, BAM ou CRAM. Os conjuntos de leitura podem ser importados para armazenamentos de sequências e anotados com metadados. Você pode aplicar permissões para ler conjuntos usando o controle de acesso baseado em atributos (ABAC).
- Referência — Uma referência de genoma é usada com leituras para identificar onde em um genoma uma leitura específica, ou grupo de leituras, é mapeada. Eles estão no formato FASTA e são armazenados no repositório de referência.
- Armazenamento de referência — Um armazenamento de dados para o armazenamento de genomas de referência. Você pode ter uma única loja de referência em cada conta e região.

Analytics

Você pode transformar e analisar seus dados genômicos com HealthOmics o Analytics. Crie um repositório de variantes ou um repositório de anotações para incluir informações adicionais para suas consultas.

- Armazenamento de variantes — armazenamento de dados que armazena dados variantes em escala populacional. Os armazenamentos de variantes suportam entradas genômicas de Variant Call Format (gVCF) e VCF.
- Armazenamento de anotações — Um armazenamento de dados que representa um banco de dados de anotações, como um arquivo TSV/CSV, VCF ou General Feature Format (). GFF3 Os repositórios de anotações são mapeados para o mesmo sistema de coordenadas dos armazenamentos de variantes durante uma importação.

Serviços relacionados

Os serviços a seguir funcionam com HealthOmics.

- Amazon Elastic Container Registry — Cada fluxo de trabalho privado usa uma imagem do Amazon ECR (em um repositório privado do Amazon ECR) para conter todos os executáveis, bibliotecas e scripts necessários para executar o fluxo de trabalho.
- Amazon Simple Storage Service — O Amazon S3 fornece armazenamento de arquivos para dados de armazenamento e fluxo de trabalho.
- AWS Lake Formation — Lake Formation gerencia o acesso aos dados aos seus armazenamentos de dados do Analytics.
- Amazon Athena — Use o Athena para realizar consultas em suas lojas Variant.
- Amazon SageMaker AI — Use a SageMaker IA para executar HealthOmics tarefas usando notebooks Jupyter.
- GitHub connections— Use conexões para conectar seus repositórios de código externos aos seus fluxos de trabalho. HealthOmics

Como acessar HealthOmics

Você pode acessar os AWS HealthOmics recursos usando o console de gerenciamento, a CLI SDKs ou a API.

- AWS Console de gerenciamento — fornece uma interface da web que você pode usar para acessar HealthOmics.
- AWS Command Line Interface (AWS CLI) — Fornece comandos para um amplo conjunto de AWS serviços, inclusive AWS HealthOmics, e é compatível com Windows, macOS e Linux. Para obter mais informações sobre a instalação do AWS CLI, consulte AWS Command Line Interface.
- AWS SDKs — AWS fornece SDKs (kits de desenvolvimento de software) que consistem em bibliotecas e código de amostra para várias linguagens e plataformas de programação (incluindo Java, Python, Ruby, .NET, iOS e Android). Eles SDKs fornecem uma maneira conveniente de usar HealthOmics programaticamente. Para obter mais informações, consulte o AWS SDK Developer Center.
- AWS API — Você pode usar operações de API para acessar e gerenciar HealthOmics programaticamente. Para obter mais informações, consulte a Referência da API do HealthOmics .

Regiões e endpoints para AWS HealthOmics

Para obter uma lista completa de regiões e endpoints, consulte a Referência AWS geral.

Além das AWS regiões que estão ativas por padrão, também há regiões opcionais que precisam ser ativadas. Para saber mais sobre como ativar ou desativar uma região, consulte Especificar quais AWS regiões sua conta pode usar no guia de gerenciamento de AWS contas.

Saiba mais

Saiba mais sobre HealthOmics esses workshops e tutoriais:

- HealthOmics workshop — workshop de HealthOmics ponta a ponta
- AWS recursos genômicos — repositórios públicos do Amazon ECR relacionados à genômica
- Tutoriais em Python — tutoriais sobre o notebook Jupyter, abrangendo armazenamento, análise e fluxos de trabalho GitHub HealthOmics

Familiarize-se com HealthOmics ferramentas adicionais que AWS fornecem:

- Linter WDL — HealthOmics linter para WDL
- Nextflow linter — HealthOmics linter para Nextflow
- HealthOmics Ferramenta auxiliar Amazon ECR — Ferramenta auxiliar Amazon ECR para HealthOmics
- HealthOmics ferramentas ativadas GitHub — Ferramentas para trabalhar com HealthOmics (gerenciador de transferência, analisador de URI, reexecução do Omics, analisador de execução).

AWS HealthOmics alteração na disponibilidade da loja de variantes e da loja de anotações

Após uma análise cuidadosa, decidimos fechar lojas de AWS HealthOmics variantes e lojas de anotações para novos clientes a partir de 7 de novembro de 2025. Os clientes atuais podem continuar usando o serviço normalmente.

A seção a seguir descreve as opções de migração para ajudar você a migrar seus repositórios de variantes e de análises para novas soluções. Para qualquer dúvida ou preocupação, crie um caso de suporte em support.console.aws.amazon.com.

Tópicos

Visão geral das opções de migração

As opções de migração a seguir oferecem uma alternativa ao uso de armazenamentos de variantes e de anotações:

1. Use a implementação HealthOmics de referência fornecida da lógica ETL.

   Use buckets de tabela do S3 para armazenamento e continue usando os serviços de AWS análise existentes.
2. Crie uma solução usando uma combinação de AWS serviços existentes.

   Para ETL, você pode escrever trabalhos personalizados do Glue ETL ou usar o código HAIL ou GLOW de código aberto no EMR para transformar dados variantes.

   Use buckets de tabela do S3 para armazenamento e continue usando os serviços de análise existentes AWS

3. Selecione um AWS parceiro que ofereça uma alternativa de armazenamento de variantes e anotações.

Opções de migração para a lógica ETL

Considere as seguintes opções de migração para a lógica ETL:

1. HealthOmics fornece a lógica ETL do armazenamento de variantes atual como um HealthOmics fluxo de trabalho de referência. Você pode usar o mecanismo desse fluxo de trabalho para alimentar exatamente o mesmo processo ETL de dados variantes do armazenamento de variantes, mas com controle total sobre a lógica do ETL.

   Esse fluxo de trabalho de referência está disponível mediante solicitação. Para solicitar acesso, crie um caso de suporte em support.console.aws.amazon.com.
2. Para transformar dados variantes, você pode escrever trabalhos personalizados do Glue ETL ou usar o código HAIL ou GLOW de código aberto no EMR.

Opções de migração para armazenamento

Como substituto do armazenamento de dados hospedado em serviços, você pode usar buckets de tabela do Amazon S3 para definir um esquema de tabela personalizado.Para obter mais informações sobre compartimentos de tabela, consulte Buquetes de tabela no Guia do usuário do Amazon S3.

Você pode usar baldes de mesa para tabelas Iceberg totalmente gerenciadas no Amazon S3.

Você pode criar um caso de suporte para solicitar que a HealthOmics equipe migre os dados do seu armazenamento de variantes ou anotações para o bucket de tabelas do Amazon S3 que você configurou.

Depois que seus dados forem preenchidos no bucket de tabelas do Amazon S3, você poderá excluir seus armazenamentos de variantes e de anotações. Para obter mais informações, consulte Excluindo lojas de HealthOmics análise.

Analytics

Para análise de dados, continue usando serviços de AWS análise, como Amazon Athena, Amazon EMR, Amazon Redshift ou Amazon Quick Suite.

AWS Parceiros

Você pode trabalhar com um AWS parceiro que fornece ETL personalizável, esquemas de tabela, ferramentas integradas de consulta e análise e interfaces de usuário para interagir com dados.

Exemplos

Os exemplos a seguir mostram como criar tabelas adequadas para armazenar dados VCF e GVCF.

Athena DDL

Você pode usar o exemplo de DDL a seguir no Athena para criar uma tabela adequada para armazenar dados VCF e GVCF em uma única tabela. Esse exemplo não é o equivalente exato da estrutura de armazenamento de variantes, mas funciona bem para um caso de uso genérico.

Crie seus próprios valores para DATABASE_NAME e TABLE_NAME ao criar a tabela.

```
 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'
);
```

Crie tabelas usando Python (sem Athena)

O exemplo de código Python a seguir mostra como criar as tabelas sem usar o 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)
```

Conf HealthOmicsiguração

Para configurar AWS HealthOmics, inscreva-se em um Conta da AWS, crie um usuário administrativo e gerencie com segurança o acesso de usuários adicionais.

Tópicos

Inscreva-se para um Conta da AWS

Se você não tiver um Conta da AWS, conclua as etapas a seguir para criar um.

Para se inscrever em um Conta da AWS

1. Abra a https://portal.aws.amazon.com/billing/inscrição.
2. Siga as instruções online.

   Parte do procedimento de inscrição envolve receber uma chamada telefônica ou uma mensagem de texto e inserir um código de verificação pelo teclado do telefone.

   Quando você se inscreve em um Conta da AWS, um Usuário raiz da conta da AWSé criado. O usuário-raiz tem acesso a todos os Serviços da AWS e recursos na conta. Como prática recomendada de segurança, atribua o acesso administrativo a um usuário e use somente o usuário-raiz para executar tarefas que exigem acesso de usuário-raiz.

AWS envia um e-mail de confirmação após a conclusão do processo de inscrição. A qualquer momento, você pode visualizar a atividade atual da sua conta e gerenciar sua conta acessando https://aws.amazon.com/e escolhendo Minha conta.

Criar um usuário com acesso administrativo

Depois de se inscrever em um Conta da AWS, proteja seu Usuário raiz da conta da AWS AWS IAM Identity Center, habilite e crie um usuário administrativo para que você não use o usuário root nas tarefas diárias.

Proteja seu Usuário raiz da conta da AWS

1. Faça login Console de gerenciamento da AWScomo proprietário da conta escolhendo Usuário raiz e inserindo seu endereço de Conta da AWS e-mail. Na próxima página, insira a senha.

   Para obter ajuda ao fazer login usando o usuário-raiz, consulte Fazer login como usuário-raiz no Guia do usuário do Início de Sessão da AWS .
2. Habilite a autenticação multifator (MFA) para o usuário-raiz.

   Para obter instruções, consulte Habilitar um dispositivo de MFA virtual para seu usuário Conta da AWS raiz (console) no Guia do usuário do IAM.

Criar um usuário com acesso administrativo

1. Habilita o Centro de Identidade do IAM.

   Para obter instruções, consulte Habilitar o AWS IAM Identity Center no Guia do usuário do AWS IAM Identity Center .
2. No Centro de Identidade do IAM, conceda o acesso administrativo a um usuário.

   Para ver um tutorial sobre como usar o Diretório do Centro de Identidade do IAM como fonte de identidade, consulte Configurar o acesso do usuário com o padrão Diretório do Centro de Identidade do IAM no Guia AWS IAM Identity Center do usuário.

Iniciar sessão como o usuário com acesso administrativo

- Para fazer login com o seu usuário do Centro de Identidade do IAM, use o URL de login enviado ao seu endereço de e-mail quando o usuário do Centro de Identidade do IAM foi criado.

  Para obter ajuda para fazer login usando um usuário do IAM Identity Center, consulte Como fazer login no portal de AWS acesso no Guia Início de Sessão da AWS do usuário.

Atribuir acesso a usuários adicionais

1. No Centro de Identidade do IAM, crie um conjunto de permissões que siga as práticas recomendadas de aplicação de permissões com privilégio mínimo.

   Para obter instruções, consulte Criar um conjunto de permissões no Guia do usuário do AWS IAM Identity Center .
2. Atribua usuários a um grupo e, em seguida, atribua o acesso de autenticação única ao grupo.

   Para obter instruções, consulte Adicionar grupos no Guia do usuário do AWS IAM Identity Center .

Crie permissões do IAM para HealthOmics

Para usar HealthOmics, configure as seguintes permissões do IAM:

- Políticas baseadas em identidade do IAM para os usuários da sua conta acessarem. HealthOmics
- Uma função de serviço do IAM HealthOmics para acessar recursos em seu nome.
- Permissões em outros serviços (como Lake Formation e Amazon ECR) para que seus usuários e o HealthOmics serviço acessem recursos.

Para obter mais informações sobre como configurar as permissões do IAM para HealthOmics, consulte Permissões do IAM para HealthOmics.

Conecte-se com repositórios de código externos

Com AWS HealthOmics, você pode gerenciar seus fluxos de trabalho usando repositórios baseados em Git por meio de. Conexões de código da AWS HealthOmics usa essa conexão para acessar seus repositórios de código-fonte.

Antes de trabalhar com repositórios de código externos, siga o guia de configuração de conexões para começar a trabalhar com Conexões de código da AWS eles. Verifique se você criou as políticas e permissões adequadas do IAM para sua AWS conta. Para obter uma lista dos provedores Git compatíveis e obter mais informações, consulte Para quais provedores terceirizados posso criar conexões? .

Crie uma conexão

Para criar uma conexão com seu provedor de repositório preferido, siga o tutorial Criar uma conexão.

Usando o Amazon Q CLI com HealthOmics

O Amazon Q CLI fornece interações de linguagem natural com AWS HealthOmics, permitindo que você execute fluxos de trabalho genômicos complexos e tarefas de análise usando comandos de conversação. Para usar o Amazon Q CLI, certifique-se de configurar as permissões do IAM para HealthOmics outros serviços (como CloudWatch Amazon ECR ou Amazon S3) para que o Amazon Q acesse seus recursos.

O tutorial de IA generativa da HealthOmics Agentic fornece uma step-by-step orientação para configurar arquivos de contexto e permitir que o Amazon Q CLI crie, execute e otimize seus fluxos de trabalho. AWS HealthOmics

Começando com HealthOmics

Para começar HealthOmics, certifique-se de ter configurado corretamente suas permissões e funções do IAM para HealthOmics.

Usando um fluxo de trabalho Ready2Run no console HealthOmics

O exercício a seguir mostra como usar um fluxo de trabalho do Ready2Run. Um fluxo de trabalho do Ready2Run é pré-configurado com os parâmetros e referências de ferramentas necessários para executar o fluxo de trabalho. O editor do fluxo de trabalho fornece dados de amostra, para que você não precise criar seus próprios dados.

1. Abra o console de HealthOmics .
2. Selecione o painel de navegação (≡) no canto superior esquerdo e selecione fluxos de trabalho Ready2Run.
3. Na página de fluxos de trabalho do Ready2Run, escolha o fluxo de trabalho. ESMFold for up to 800 residues O console abre a página de detalhes desse fluxo de trabalho.
4. A guia de detalhes fornece informações sobre o fluxo de trabalho. Para testar o fluxo de trabalho, no canto superior direito da página, selecione Iniciar execução.
5. Na página Especificar detalhes da execução, insira um nome de execução.
6. Insira ou selecione um local do Amazon S3 para a saída da execução.
7. Para o modo de retenção de metadados Executar, escolha se deseja reter ou remover dados runmeta.
8. No painel Função de serviço, escolha Criar e usar uma nova função de serviço.
9. Escolha Próximo.
10. Na página Adicionar valores de parâmetros, escolha Executar fluxo de trabalho com dados de teste do Ready2Run.
11. Escolha Próximo.
12. Revise suas entradas e escolha Iniciar execução.

Exemplo de solicitações para Amazon Q CLI

O Amazon Q CLI pode executar fluxos de trabalho genômicos e tarefas de análise usando comandos de linguagem natural. AWS HealthOmics Os exemplos de prompts a seguir permitem criar fluxos de

trabalho, gerenciar execuções e analisar dados genômicos. Para obter mais informações e exemplos de solicitações HealthOmics, consulte o tutorial de IA generativa da HealthOmics Agentic em. GitHub

- “Crie um arquivo de fluxo de trabalho da WDL 1.1 à medida `main.wdl` que ele será executado HealthOmics. O fluxo de trabalho terá um genoma de referência como entrada e pares de arquivos fastq. Ele indexará o genoma de referência usando o BWA e, em seguida, mapeará cada par de arquivos fastq para a referência. Por fim, mescle cada BAM mapeado em um único arquivo BAM e imprima esse arquivo e seu índice bai.”
- “Package o fluxo de trabalho e crie-o em HealthOmics”
- “Atualize o arquivo inputs.json para usar arquivos reais do meu bucket do Amazon S3`omics-my-bucket-with-genome-data`" (forneça uma localização específica do bucket do Amazon S3 ou deixe o Amazon Q explorar)
- “Encontre contêineres adequados em meus repositórios Amazon ECR e atualize inputs.json para usá-los”
- “Encontre ou crie uma função do IAM adequada para usar ao executar o fluxo de trabalho”
- “Criar um cache de execução para meu fluxo de trabalho”
- “Execute o fluxo de trabalho em HealthOmics”
- “Verifique o status da execução”

> **Warning**
>
> Ao trabalhar com o Amazon Q CLI, revise todo o conteúdo gerado e as ações propostas antes de continuar. Forneça feedback para melhorar a qualidade da resposta e atender aos requisitos do seu fluxo de trabalho. Para obter mais informações, consulte Considerações de segurança e melhores práticas para o Amazon Q.

Fluxos de trabalho privados em HealthOmics

Use fluxos de trabalho privados quando quiser criar sua própria definição de fluxo de trabalho. A definição do fluxo de trabalho especifica informações sobre o fluxo de trabalho e define as tarefas do fluxo de trabalho. Uma execução é uma invocação única de um fluxo de trabalho, e uma tarefa é um único processo dentro da execução.

HealthOmics suporta definições de fluxo de trabalho que você cria em Workflow Description Language (WDL), Common Workflow Language (CWL) ou Nextflow.

HealthOmics os fluxos de trabalho fornecem os seguintes recursos opcionais:

- Run groups— Você pode adicionar fluxos de trabalho privados a um grupo de execução para controlar o uso da computação. Um grupo de execução é uma coleção de execuções de fluxo de trabalho que compartilham um conjunto de limites de recursos, como máximo de execuções simultâneas e duração máxima de execução. Você define esses limites para controlar os recursos computacionais que o grupo de execução consome.
- Call caching— Você pode usar um cache de chamadas para salvar e reutilizar saídas de tarefas, o que resulta em menores durações de execução e economia de custos de computação.
- Sharing workflows— Você pode compartilhar seus fluxos de trabalho privados com outras pessoas Contas da AWS na mesma região.
- Workflow versions— Você pode criar versões de um fluxo de trabalho privado. O controle de versão do fluxo de trabalho permite que os usuários escolham quando começar a usar a funcionalidade atualizada. As versões do fluxo de trabalho são imutáveis e fornecem o mesmo nível de proveniência de dados dos fluxos de trabalho.

Para obter informações sobre como configurar as permissões do IAM para fluxos de trabalho, consulte. Permissões do IAM para HealthOmics

Para obter exemplos completos de como usar fluxos de trabalho HealthOmics privados, consulte os tutoriais do HealthOmics Github ou o tutorial completo do workshop da AWS para. HealthOmics

Tópicos

- Criação de fluxos de trabalho privados em HealthOmics
- Controle de versão do fluxo de trabalho em HealthOmics
- Usando HealthOmics execuções

- Usando grupos de HealthOmics execução
- Cache de chamadas para execuções HealthOmics
- Compartilhamento de HealthOmics fluxos de trabalho

Criação de fluxos de trabalho privados em HealthOmics

Os fluxos de trabalho privados dependem de uma variedade de recursos que você cria e configura antes de criar o fluxo de trabalho:

- Workflow definition file:Um arquivo de definição de fluxo de trabalho escrito em WDLNextflow,, ouCWL. A definição do fluxo de trabalho especifica as entradas e saídas para execuções que usam o fluxo de trabalho. Também inclui especificações para as execuções e tarefas de execução do seu fluxo de trabalho, incluindo requisitos de computação e memória. O arquivo de definição do fluxo de trabalho deve estar no `.zip` formato. Para obter mais informações, consulte Arquivos de definição de fluxo de trabalho.
  - Você pode usar o Amazon Q CLI para criar e validar seus arquivos de definição de fluxo de trabalho em WDL, Nextflow e CWL. Para obter mais informações, consulte Exemplos de solicitações para a Amazon Q CLI e HealthOmics o tutorial de IA generativa da Agentic sobre. GitHub
- (Optional) Parameter template file:Um arquivo de modelo de parâmetro escrito emJSON. Crie o arquivo para definir os parâmetros de execução ou HealthOmics gere o modelo de parâmetro para você. Para obter mais informações, consulte Arquivos de modelo de parâmetros para HealthOmics fluxos de trabalho.
- Amazon ECR container images:Crie um repositório Amazon ECR privado para o fluxo de trabalho. Crie imagens de contêiner no repositório privado ou sincronize o conteúdo de um registro upstream compatível com seu repositório privado Amazon ECR.
- (Optional) Sentieon licenses:Solicite uma Sentieon licença para usar o Sentieon software em fluxos de trabalho privados.

Opcionalmente, você pode executar um linter na definição do fluxo de trabalho antes ou depois de criar o fluxo de trabalho. O linter tópico descreve as impressoras disponíveis em HealthOmics.

Tópicos

- HealthOmics integração do fluxo de trabalho com repositórios baseados em Git
- Arquivos de definição de fluxo de trabalho em HealthOmics

- Arquivos de modelo de parâmetros para HealthOmics fluxos de trabalho
- Imagens de contêiner para fluxos de trabalho privados
- HealthOmics Arquivos README do fluxo de trabalho
- Solicitando licenças Sentieon para fluxos de trabalho privados
- Impressoras de fluxo de trabalho em HealthOmics
- HealthOmics operações de fluxo de trabalho

HealthOmics integração do fluxo de trabalho com repositórios baseados em Git

Ao criar um fluxo de trabalho (ou uma versão do fluxo de trabalho), você fornece uma definição de fluxo de trabalho para especificar informações sobre o fluxo de trabalho, as execuções e as tarefas. HealthOmics pode recuperar a definição do fluxo de trabalho como um arquivo.zip (armazenado localmente ou em um bucket do Amazon S3) ou de um repositório compatível baseado em Git.

A HealthOmics integração com repositórios baseados em Git permite os seguintes recursos:

- Criação direta de fluxo de trabalho a partir de instâncias públicas, privadas e autogerenciadas.
- Integração de arquivos README do fluxo de trabalho e modelos de parâmetros de repositórios.
- Support para GitHub, GitLab, e repositórios Bitbucket.

Ao usar um repositório baseado em Git, você evita as etapas manuais de baixar arquivos de definição de fluxo de trabalho e arquivos de modelo de parâmetros de entrada, criar um arquivo.zip e, em seguida, preparar o arquivo para o S3. Isso simplifica a criação do fluxo de trabalho para cenários como os exemplos a seguir:

1. Você quer começar rapidamente usando um fluxo de trabalho comum de código aberto, como nf-core. HealthOmicsrecupera automaticamente todos os arquivos de modelo de definição de fluxo de trabalho e parâmetros de entrada do repositório nf-core GitHub e usa esses arquivos para criar seu novo fluxo de trabalho.
2. Você está usando um fluxo de trabalho público do GitHub, e algumas novas atualizações são disponibilizadas. Você pode criar facilmente uma nova versão do HealthOmics fluxo de trabalho usando a definição atualizada do fluxo de trabalho GitHub como fonte. Os usuários do seu fluxo de trabalho podem escolher entre o fluxo de trabalho original ou a nova versão do fluxo de trabalho que você criou.

3. Sua equipe está criando um funil proprietário que não é público. Você mantém seu código em um repositório git privado e usa essa definição de fluxo de trabalho para seus HealthOmics fluxos de trabalho. A equipe atualiza a definição do fluxo de trabalho com frequência como parte de um ciclo de vida de desenvolvimento de fluxo de trabalho iterativo. Você pode criar facilmente novas versões do fluxo de trabalho, conforme necessário, em seu repositório privado.

Tópicos

Repositórios baseados em Git compatíveis

HealthOmics oferece suporte a repositórios públicos e privados para os seguintes provedores baseados em Git:

- GitHub
- GitLab
- Bitbucket

HealthOmics oferece suporte a repositórios autogerenciados para os seguintes provedores baseados em Git:

- GitHubEnterpriseServer
- GitLabSelfManaged

HealthOmics suporta o uso de conexões entre contas para GitHub GitLab, e Bitbucket. Configure permissões compartilhadas por meio do AWS Resource Access Manager. Por exemplo, consulte Conexões compartilhadas no guia CodePipeline do usuário.

Configurar conexões com repositórios de código externos

Conecte seus fluxos de trabalho a repositórios baseados em Git usando a AWS. CodeConnection HealthOmics usa essa conexão para acessar seus repositórios de código-fonte.

> **Note**
>
> O CodeConnections serviço da AWS não está disponível na região il-central-1. Para essa região, configure o serviço us-east-1 para criar fluxos de trabalho ou versões de fluxo de trabalho a partir de um repositório.

Criar uma conexão

Antes de criar conexões, siga as instruções em Como configurar conexões no Guia do usuário das ferramentas do Developer Console.

Para criar uma conexão, siga as instruções em Criar uma conexão no Guia do usuário das ferramentas do Developer Console.

Configurar a autorização para a conexão

Você deve autorizar a conexão usando o OAuth fluxo do provedor. Verifique se o status da conexão é `AVAILABLE` antes de usá-la.

Para ver exemplos, consulte a postagem do blog Como criar AWS HealthOmics fluxos de trabalho a partir de conteúdo no Git.

Acessando repositórios autogerenciados

Para configurar conexões com um repositório GitLab autogerenciado, use um token de acesso pessoal de administrador ao criar um host. A criação subsequente da conexão acessa o Oauth com a conta do cliente.

O exemplo a seguir configura uma conexão com um repositório GitLab autogerenciado:

1. Configure o acesso ao token de acesso pessoal de um usuário administrador.

   Para configurar um PAT em um repositório GitLab autogerenciado, consulte Tokens de acesso pessoal no GitLab Docs.
2. Criar um host

a. Navegue até CodePipeline>Configurações>Conexões.
b. Escolha a guia Hosts e, em seguida, escolha Create Host.
c. Configure os campos a seguir.
   - Insira o nome do host
   - Para o tipo de provedor, escolha GitLab Autogerenciado
   - Insira o URL do host
   - Insira as informações da VPC se o host estiver definido em uma VPC
d. Escolha Criar host, que cria o host no estado PENDENTE.
e. Para concluir a configuração, escolha Configurar host.
f. Insira o token de acesso pessoal (PAT) de um usuário administrador e escolha Continuar.

3. Criar a conexão
   a. Escolha Criar conexões na guia Conexões.
   b. Para o tipo de provedor, selecione GitLab autogerenciado.
   c. Em Configurações de conexão > Inserir nome da conexão, insira a URL do host que você criou anteriormente.
   d. Se sua instância GitLab autogerenciada só puder ser acessada por meio de uma VPC, configure os detalhes da VPC.
   e. Escolha Atualizar conexão pendente. A janela modal redireciona você para a página de login. GitLab
   f. Insira o nome de usuário e a senha da conta do cliente e conclua o processo de autorização.
   g. Para a primeira configuração, escolha Autorizar AWS Connector for Gitlab Self Managed.

Cotas relacionadas a repositórios de código externos

Para HealthOmics integração com repositórios de código externos, há um tamanho máximo para um repositório, cada arquivo do repositório e cada arquivo README. Para obter detalhes, consulte HealthOmics cotas de tamanho fixo do fluxo de trabalho.

Permissões obrigatórias do IAM

Adicione as seguintes ações à sua política de IAM baseada em identidade:

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

Arquivos de definição de fluxo de trabalho em HealthOmics

Você usa uma definição de fluxo de trabalho para especificar informações sobre o fluxo de trabalho, as execuções e as tarefas nas execuções. Você cria definições de fluxo de trabalho em um ou mais arquivos usando uma linguagem de definição de fluxo de trabalho. HealthOmics suporta definições de fluxo de trabalho escritas em WDL, Nextflow ou CWL.

HealthOmics suporta as seguintes opções para definições de fluxo de trabalho da WDL:

- WDL — Fornece um mecanismo WDL em conformidade com as especificações.
- WDL leniente — Projetado para lidar com fluxos de trabalho migrados de Cromwell. Ele suporta as diretivas Cromwell do cliente e algumas lógicas não conformes. Para obter detalhes, consulte Conversão de tipo implícita em WDL leniente.

Para obter informações sobre cada uma das linguagens do fluxo de trabalho, consulte as seções detalhadas específicas do idioma abaixo.

Você especifica os seguintes tipos de informações na definição do fluxo de trabalho:

- Language version— O idioma e a versão da definição do fluxo de trabalho.
- Compute and memory— Os requisitos de computação e memória para tarefas no fluxo de trabalho.
- Inputs— Localização das entradas para as tarefas do fluxo de trabalho. Para obter mais informações, consulte HealthOmics entradas de execução.
- Outputs— Localização para salvar as saídas que as tarefas geram.
- Task resources— Requisitos de computação e memória para cada tarefa.
- Accelerators— outros recursos que as tarefas exigem, como aceleradores.

Tópicos

- HealthOmics requisitos de definição de fluxo de trabalho

- Suporte de versão para linguagens HealthOmics de definição de fluxo de trabalho
- Requisitos de computação e memória para tarefas HealthOmics
- Saídas de tarefas em uma definição de HealthOmics fluxo de trabalho
- Recursos de tarefas em uma definição de HealthOmics fluxo de trabalho
- Aceleradores de tarefas em uma definição de HealthOmics fluxo de trabalho
- Especificidades da definição do fluxo de trabalho da WDL
- Especificações da definição do fluxo de trabalho do Nextflow
- Especificidades da definição do fluxo de trabalho do CWL
- Exemplos de definições de fluxo de trabalho

HealthOmics requisitos de definição de fluxo de trabalho

Os arquivos HealthOmics de definição do fluxo de trabalho devem atender aos seguintes requisitos:

- As tarefas devem definir input/output parâmetros, repositórios de contêineres do Amazon ECR e especificações de tempo de execução, como alocação de memória ou CPU.
- Verifique se suas funções do IAM têm as permissões necessárias.
  - Seu fluxo de trabalho tem acesso aos dados de entrada de AWS recursos, como o Amazon S3.
  - Seu fluxo de trabalho tem acesso aos serviços de repositório externo quando necessário.
- Declare os arquivos de saída na definição do fluxo de trabalho. Para copiar arquivos de execução intermediários para o local de saída, declare-os como saídas do fluxo de trabalho.
- Os locais de entrada e saída devem estar na mesma região do fluxo de trabalho.
- HealthOmics as entradas do fluxo de trabalho de armazenamento devem estar em `ACTIVE` status. HealthOmics não importará entradas com um `ARCHIVED` status, fazendo com que o fluxo de trabalho falhe. Para obter informações sobre entradas de objetos do Amazon S3, consulte. HealthOmics entradas de execução
- A main localização do fluxo de trabalho é opcional se o arquivo ZIP contiver uma única definição de fluxo de trabalho ou um arquivo chamado “principal”.
  - Exemplo de caminho: `workflow-definition/main-file.wdl`
- Antes de criar um fluxo de trabalho a partir do Amazon S3 ou de sua unidade local, crie um arquivo zip dos arquivos de definição do fluxo de trabalho e de quaisquer dependências, como subfluxos de trabalho.

- Recomendamos que você declare os contêineres do Amazon ECR no fluxo de trabalho como parâmetros de entrada para validação das permissões do Amazon ECR.

Considerações adicionais sobre o Nextflow:

- /bin

  As definições do fluxo de trabalho do Nextflow podem incluir uma pasta /bin com scripts executáveis. Esse caminho tem acesso somente para leitura e executável às tarefas. As tarefas que dependem desses scripts devem usar um contêiner criado com os intérpretes de script apropriados. A melhor prática é ligar diretamente para o intérprete. Por exemplo:

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

- includeConfig

  As definições de fluxo de trabalho baseadas em Nextflow podem incluir arquivos nextflow.config que ajudam a abstrair definições de parâmetros ou perfis de recursos de processo. Para oferecer suporte ao desenvolvimento e à execução de pipelines Nextflow em vários ambientes, use uma configuração HealthOmics específica que você adiciona à configuração global usando a diretiva IncludeConfig. Para manter a portabilidade, configure o fluxo de trabalho para incluir o arquivo somente durante a execução HealthOmics usando o seguinte código:

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

- Reports

  HealthOmics não oferece suporte a relatórios de arrastamento, rastreamento e execução gerados pelo mecanismo. Você pode gerar alternativas para os relatórios de rastreamento e execução usando uma combinação de GetRun chamadas de GetRunTask API.

Considerações adicionais sobre a CWL:

- Container image uri interpolation

  HealthOmics permite que a propriedade DockerPull do DockerRequirement seja uma expressão javascript embutida. Por exemplo:

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

  Isso permite que você especifique a imagem do contêiner URIs como parâmetros de entrada para o fluxo de trabalho.
- Javascript expressions

  As expressões Javascript devem ser `strict mode` compatíveis.
- Operation process

  HealthOmics não oferece suporte aos processos de operação do CWL.

Suporte de versão para linguagens HealthOmics de definição de fluxo de trabalho

HealthOmics suporta arquivos de definição de fluxo de trabalho escritos em Nextflow, WDL ou CWL. As seções a seguir fornecem informações sobre o suporte de HealthOmics versão para esses idiomas.

Tópicos

Suporte à versão WDL

HealthOmics suporta as versões 1.0, 1.1 e a versão de desenvolvimento da especificação WDL.

Todo documento da Biblioteca Digital Mundial deve incluir uma declaração de versão para especificar a qual versão (principal e secundária) da especificação ele adere. Para obter mais informações sobre versões, consulte Controle de versão da WDL

As versões 1.0 e 1.1 da especificação WDL não suportam o `Directory` tipo. Para usar o `Directory` tipo para entradas ou saídas, defina a versão como development na primeira linha do arquivo:

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

Suporte à versão CWL

HealthOmics suporta as versões 1.0, 1.1 e 1.2 da linguagem CWL.

Você pode especificar a versão do idioma no arquivo de definição do fluxo de trabalho CWL. Para obter mais informações sobre o CWL, consulte o guia do usuário do CWL

Suporte à versão Nextflow

HealthOmics suporta três versões estáveis do Nextflow. O Nextflow normalmente lança uma versão estável a cada seis meses. HealthOmics não suporta os lançamentos mensais “edge”.

HealthOmics oferece suporte aos recursos lançados em cada versão, mas não aos recursos de pré-visualização.

Versões aceitas

HealthOmics suporta as seguintes versões do Nextflow:

- Nextflow v22.04.01 DSL 1 e DSL 2
- Nextflow v23.10.0 DSL 2 (padrão)
- Nextflow v24.10.8 DSL 2

Para migrar seu fluxo de trabalho para a versão mais recente compatível (v24.10.8), siga o guia de atualização do Nextflow.

Há algumas mudanças importantes ao migrar do Nextflow v23 para a v24, conforme descrito nas seções a seguir do guia de migração do Nextflow:

- Mudanças recentes em 24.04
- Alterações de última hora em 24.10

Detecte e processe versões do Nextflow

HealthOmics detecta a versão DSL e a versão do Nextflow que você especifica. Ele determina automaticamente a melhor versão do Nextflow a ser executada com base nessas entradas.

Versão DSL

HealthOmics detecta a versão DSL solicitada no arquivo de definição do fluxo de trabalho. Por exemplo, você pode especificar:`nextflow.enable.dsl=2`.

HealthOmics suporta DSL 2 por padrão. Ele fornece compatibilidade com versões anteriores do DSL 1, se especificado no arquivo de definição do fluxo de trabalho.

- Se você especificar DSL 2, HealthOmics executará o Nextflow v23.10.0, a menos que você especifique o Nextflow v22.04.0 ou v24.10.8.
- Se você especificar DSL 1, HealthOmics executa o Nextflow v22.04 DSL1 (a única versão compatível que executa DSL 1).
- Se você não especificar uma versão da DSL ou se não HealthOmics conseguir analisar as informações da DSL por qualquer motivo (como erros de sintaxe no arquivo de definição do fluxo de trabalho), o HealthOmics padrão é DSL 2 e executa o Nextflow v23.10.0.
- Para atualizar seu fluxo de trabalho do DSL 1 para o DSL 2 e aproveitar as versões mais recentes do Nextflow e os recursos do software, consulte Migração do DSL 1.

Versões do Nextflow

HealthOmics detecta a versão solicitada do Nextflow no arquivo de configuração do Nextflow (nextflow.config), se você fornecer esse arquivo. Recomendamos que você adicione a `nextflowVersion` cláusula no final do arquivo para evitar substituições inesperadas das configurações incluídas. Para obter mais informações, consulte Configuração do Nextflow.

Você pode especificar uma versão do Nextflow ou um intervalo de versões usando a seguinte sintaxe:

```
// 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 processa as informações da versão do Nextflow da seguinte forma:

- Se você usar = para especificar uma versão exata que HealthOmics ofereça suporte, HealthOmics use essa versão.
- Se você usar ! para especificar uma versão exata ou um intervalo de versões que não são suportadas, HealthOmics gera uma exceção e falha na execução. Considere usar essa opção se quiser ser rigoroso com as solicitações de versão e falhar rapidamente se a solicitação incluir versões sem suporte.
- Se você especificar um intervalo de versões, HealthOmics usa a versão mais recente compatível nesse intervalo, a menos que o intervalo inclua v24.10.8. Nesse caso, HealthOmics dá preferência a uma versão anterior. Por exemplo, se o intervalo abranger v23.10.0 e v24.10.8, escolha v23.10.0. HealthOmics
- Se não houver uma versão solicitada ou se as versões solicitadas não forem válidas ou não puderem ser analisadas por algum motivo:
  - Se você especificou DSL 1, HealthOmics executa o Nextflow v22.04.
  - Caso contrário, HealthOmics executa o Nextflow v23.10.0.

Você pode recuperar as seguintes informações sobre a versão do Nextflow HealthOmics usada para cada execução:

- Os registros de execução contêm informações sobre a versão real do Nextflow HealthOmics usada para a execução.
- HealthOmics adiciona avisos nos registros de execução se não houver uma correspondência direta com a versão solicitada ou se for necessário usar uma versão diferente da especificada.
- A resposta à operação da GetRun API inclui um campo (`engineVersion`) com a versão real do Nextflow HealthOmics usada para a execução. Por exemplo:

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

Requisitos de computação e memória para tarefas HealthOmics

HealthOmics executa suas tarefas privadas de fluxo de trabalho em uma instância omics. HealthOmics fornece uma variedade de tipos de instância para acomodar diferentes tipos de tarefas. Cada tipo de instância tem uma configuração fixa de memória e vCPU (e configuração de GPU fixa para tipos de instância de computação acelerada). O custo do uso de uma instância omics varia de acordo com o tipo de instância. Para obter detalhes, consulte a página HealthOmics de preços.

Para tarefas em um fluxo de trabalho, você especifica a memória necessária e v CPUs no arquivo de definição do fluxo de trabalho. Quando uma tarefa de fluxo de trabalho é executada, HealthOmics aloca a menor instância omics que acomoda a memória solicitada e v. CPUs Por exemplo, se uma tarefa precisar de 64 GiB de memória e 8 vCPUs, HealthOmics seleciona. `omics.r.2xlarge`

Recomendamos que você analise os tipos de instância e defina o v CPUs e o tamanho da memória solicitados para corresponder à instância que melhor atenda às suas necessidades. O contêiner de tarefas usa o número de v CPUs e o tamanho da memória que você especifica no arquivo de definição do fluxo de trabalho, mesmo que o tipo de instância tenha v CPUs e memória adicionais.

A lista a seguir contém informações adicionais sobre alocação de vCPU e memória:

- As alocações de recursos de contêineres são limites rígidos. Se uma tarefa ficar sem memória ou tentar usar v adicionalCPUs , a tarefa gerará um registro de erros e sairá.
- Se você não especificar nenhum requisito de computação ou memória, HealthOmics seleciona omics.c.large e usa como padrão uma configuração com 1 vCPU e 1 GiB de memória.
- A configuração mínima que você pode solicitar é de 1 vCPU e 1 GiB de memória.
- Se você especificar vCPUs, memory, or GPUs that exceda os tipos de instância compatíveis, HealthOmics gerará uma mensagem de erro e o fluxo de trabalho falhará nas validações
- Se você especificar unidades fracionárias, HealthOmics arredonda para o número inteiro mais próximo.
- HealthOmics reserva uma pequena quantidade de memória (5%) para agentes de gerenciamento e registro, portanto, a alocação total de memória pode nem sempre estar disponível para o aplicativo na tarefa.

- HealthOmics combina os tipos de instância para atender aos requisitos de computação e memória que você especifica e pode usar uma combinação de gerações de hardware. Por esse motivo, pode haver algumas pequenas variações nos tempos de execução da mesma tarefa.

Esses tópicos fornecem detalhes sobre os tipos de instância HealthOmics compatíveis.

Tópicos

> **Note**
>
> Para instâncias padrão, otimizadas para computação e memória, aumente o tamanho da largura de banda da instância se a instância exigir uma taxa de transferência maior. As instâncias do Amazon EC2 com menos de 16 vCPUs (tamanho 4xl e menores) podem experimentar uma explosão de taxa de transferência. Para obter mais informações sobre a taxa de transferência de instâncias do Amazon EC2, consulte Largura de banda da instância disponível do Amazon EC2.

Tipos de instância padrão

Para tipos de instância padrão, as configurações buscam um equilíbrio entre poder computacional e memória.

HealthOmics oferece suporte às instâncias de 32xlarge e 48xlarge nas seguintes regiões: Oeste dos EUA (Oregon) e Leste dos EUA (Norte da Virgínia).

Instância	Número de v CPUs	Memória
omics.m.large	2	8 GiB
omics.m.xlarge	4	16 GiB
omics.m.2xlarge	8	32 GiB

Instância	Número de v CPUs	Memória
omics.m.4xlarge	16	64 GiB
omics.m.8xlarge	32	128 GiB
omics.m.12xlarge	48	192 GiB
omics.m.16xlarge	64	256 GiB
omics.m.24xlarge	96	384 GiB
omics.m.32xlarge	128	512 GiB
omics.m.48xlarge	192	768 GiB

Instâncias otimizadas para computação

Para tipos de instância otimizados para computação, as configurações têm mais poder computacional e menos memória.

HealthOmics oferece suporte às instâncias de 32xlarge e 48xlarge nas seguintes regiões: Oeste dos EUA (Oregon) e Leste dos EUA (Norte da Virgínia).

Instância	Número de v CPUs	Memória
omics.c.large	2	4 GiB
omics.c.xlarge	4	8 GiB
ômics.c.2xlarge	8	16 GiB
omics.c.4xlarge	16	32 GiB
omics.c.8xlarge	32	64 GiB
omics.c.12xlarge	48	96 GiB
omics.c.16xlarge	64	128 GiB

Instância	Número de v CPUs	Memória
omics.c.24xlarge	96	192 GiB
omics.c.32xlarge	128	256 GiB
omics.c.48xlarge	192	384 GiB

Instâncias otimizadas para memória

Para tipos de instância otimizados para memória, as configurações têm menos poder computacional e mais memória.

HealthOmics oferece suporte às instâncias de 32xlarge e 48xlarge nas seguintes regiões: Oeste dos EUA (Oregon) e Leste dos EUA (Norte da Virgínia).

Instância	Número de v CPUs	Memória
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.8xlarge	32	256 GiB
omics.r.12xlarge	48	384 GiB
omics.r.16xlarge	64	512 GiB
omics.r.24xlarge	96	768 GiB
omics.r.32xlarge	128	1024 GiB
omics.r.48xlarge	192	1536 GiB

Instâncias de computação acelerada

Opcionalmente, você pode especificar recursos de GPU para cada tarefa em um fluxo de trabalho, de forma a HealthOmics alocar uma instância de computação acelerada para a tarefa. Para obter informações sobre como especificar as informações da GPU no arquivo de definição do fluxo de trabalho, consulteAceleradores de tarefas em uma definição de HealthOmics fluxo de trabalho.

Se você especificar uma GPU compatível com vários tipos de instância, HealthOmics selecionará o tipo de instância com base na disponibilidade. Se os dois tipos de instância estiverem disponíveis, HealthOmics dará preferência à instância de menor custo.

As instâncias G4 não são suportadas na região de Israel (Tel Aviv). As instâncias G5 não são suportadas na região Ásia-Pacífico (Cingapura).

Tópicos

Tipos de instância G6 e G6e

HealthOmics oferece suporte às seguintes configurações de instância de computação acelerada G6. Todas as instâncias omics.g6 usam Nvidia L4 ou Nvidia L4 A10G. GPUs

HealthOmics suporta as instâncias G6 e G6e nas seguintes regiões: Oeste dos EUA (Oregon) e Leste dos EUA (Norte da Virgínia).

Instância	Número de v CPUs	Memória	Número de GPUs	Memória da GPU	
omics.g6.xlarge	4	16 GiB	1	24 GiB	
omics.g6.2xlarge	8	32 GiB	1	24 GiB	
omics.g6.4xlarge	16	64 GiB	1	24 GiB	

Instância	Número de v CPUs	Memória	Número de GPUs	Memória da GPU	
omics.g6.8xlarge	32	128 GiB	1	24 GiB	
omics.g6.12xlarge	48	192 GiB	4	96 GiB	
omics.g6.16xlarge	64	256 GiB	1	24 GiB	
omics.g6.24xlarge	96	384 GiB	4	96 GiB	

Todas as instâncias omics.g6e usam Nvidia L40s. GPUs

Instância	Número de v CPUs	Memória	Número de GPUs	Memória da GPU	
omics.g6e.xlarge	4	32 GiB	1	48 GiB	
omics.g6e.2xlarge	8	64 GiB	1	48 GiB	
omics.g6e.4xlarge	16	128 GiB	1	48 GiB	
omics.g6e.8xlarge	32	256 GiB	1	48 GiB	
omics.g6e.12xlarge	48	384 GiB	4	192 GiB	
omics.g6e.16xlarge	64	512 GiB	1	48 GiB	

Instância	Número de v CPUs	Memória	Número de GPUs	Memória da GPU	
omics.g6e .24xlarge	96	768 GiB	4	192 GiB	

Instâncias G4 e G5

HealthOmics oferece suporte às seguintes configurações de instância de computação acelerada G4 e G5.

Todas as instâncias omics.g5 usam Nvidia L4 A10G, Nvidia Tesla A10G ou Nvidia Tesla T4 A10G. GPUs

Instância	Número de v CPUs	Memória	Número de GPUs	Memória da GPU	
omics.g5. xlarge	4	16 GiB	1	24 GiB	
omics.g5. 2xlarge	8	32 GiB	1	24 GiB	
omics.g5. 4xlarge	16	64 GiB	1	24 GiB	
omics.g5. 8xlarge	32	128 GiB	1	24 GiB	
omics.g5. 12xlarge	48	192 GiB	4	96 GiB	
omics.g5. 16xlarge	64	256 GiB	1	24 GiB	
omics.g5. 24xlarge	96	384 GiB	4	96 GiB	

Todas as instâncias omics.g4dn usam Nvidia Tesla T4 ou Nvidia Tesla T4 A10G. GPUs

Instância	Número de v CPUs	Memória	Número de GPUs	Memória da GPU	
omics.g4dn.xlarge	4	16 GiB	1	16 GiB	
omics.g4dn.2xlarge	8	32 GiB	1	16 GiB	
omics.g4dn.4xlarge	16	64 GiB	1	16 GiB	
omics.g4dn.8xlarge	32	128 GiB	1	16 GiB	
omics.g4dn.12xlarge	48	192 GiB	4	64 GiB	
omics.g4dn.16xlarge	64	256 GiB	1	24 GiB	

Saídas de tarefas em uma definição de HealthOmics fluxo de trabalho

Você especifica as saídas da tarefa na definição do fluxo de trabalho. Por padrão, HealthOmics descarta todos os arquivos de tarefas intermediárias quando o fluxo de trabalho é concluído. Para exportar um arquivo intermediário, você o define como uma saída.

Se você usar o cache de chamadas, HealthOmics salva as saídas da tarefa no cache, incluindo todos os arquivos intermediários que você definir como saídas.

Os tópicos a seguir incluem exemplos de definição de tarefas para cada uma das linguagens de definição de fluxo de trabalho.

Tópicos

- Saídas de tarefas para WDL
- Saídas de tarefas para Nextflow

- Saídas de tarefas para CWL

Saídas de tarefas para WDL

Para definições de fluxo de trabalho escritas em WDL, defina suas saídas na seção de fluxo outputs de trabalho de nível superior.

HealthOmics

Tópicos

- Saída de tarefa para STDOUT
- Saída de tarefa para STDERR
- Saída da tarefa para um arquivo
- Saída de tarefas para uma matriz de arquivos

Saída de tarefa para STDOUT

Este exemplo cria uma tarefa chamada `SayHello` que ecoa o conteúdo STDOUT no arquivo de saída da tarefa. A stdout função WDL captura o conteúdo STDOUT (neste exemplo, a string de entrada Hello World!) no arquivostdout_file.

Como HealthOmics cria registros para todo o conteúdo STDOUT, a saída também aparece em CloudWatch Logs, junto com outras informações de registro STDERR da tarefa.

```
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()
    }
}
```

Saída de tarefa para STDERR

Este exemplo cria uma tarefa chamada `SayHello` que ecoa o conteúdo STDERR no arquivo de saída da tarefa. A stderr função WDL captura o conteúdo STDERR (neste exemplo, a string de entrada Hello World!) no arquivostderr_file.

Como HealthOmics cria registros para todo o conteúdo STDERR, a saída aparecerá em CloudWatch Logs, junto com outras informações de registro STDERR da tarefa.

```
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()
    }
}
```

Saída da tarefa para um arquivo

Neste exemplo, a SayHello tarefa cria dois arquivos (message.txt e info.txt) e declara explicitamente esses arquivos como as saídas nomeadas (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"
    }
}
```

Saída de tarefas para uma matriz de arquivos

Neste exemplo, a `GenerateGreetings` tarefa gera uma matriz de arquivos como saída da tarefa. A tarefa gera dinamicamente um arquivo de saudação para cada membro da matriz de entrada. `names` Como os nomes dos arquivos não são conhecidos até o tempo de execução, a definição de saída usa a função WDL glob () para gerar todos os arquivos que correspondam ao padrão. `*_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")
    }
}
```

Saídas de tarefas para Nextflow

Para definições de fluxo de trabalho escritas no Nextflow, defina uma diretiva PublishDir para exportar o conteúdo da tarefa para seu bucket de saída do Amazon S3. Defina o valor PublishDir como `/mnt/workflow/pubdir`.

HealthOmics Para exportar arquivos para o Amazon S3, os arquivos devem estar nesse diretório.

Se uma tarefa produzir um grupo de arquivos de saída para uso como entradas para uma tarefa subseqüente, recomendamos que você agrupe esses arquivos em um diretório e emita o diretório como uma saída de tarefa. A enumeração de cada arquivo individual pode resultar em um gargalo de E/S no sistema de arquivos subjacente. Por exemplo:

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

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

Saídas de tarefas para CWL

Para definições de fluxo de trabalho escritas em CWL, você pode especificar as saídas da tarefa usando `CommandLineTool` tarefas. As seções a seguir mostram exemplos de `CommandLineTool` tarefas que definem diferentes tipos de saídas.

Tópicos

- Saída de tarefa para STDOUT
- Saída de tarefa para STDERR

- Saída da tarefa para um arquivo
- Saída de tarefas para uma matriz de arquivos

Saída de tarefa para STDOUT

Este exemplo cria uma `CommandLineTool` tarefa que ecoa o conteúdo STDOUT em um arquivo de saída de texto chamado. output.txt Por exemplo, se você fornecer a seguinte entrada, a saída da tarefa resultante será Hello World! no output.txt arquivo.

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

A `outputs` diretiva especifica que o nome da saída é example_out e seu tipo é`stdout`. Para que uma tarefa posterior consuma a saída dessa tarefa, ela se referiria à saída como`example_out`.

Como HealthOmics cria registros para todo o conteúdo STDERR e STDOUT, a saída também aparece em CloudWatch Logs, junto com outras informações de registro STDERR da tarefa.

```
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
```

Saída de tarefa para STDERR

Este exemplo cria uma `CommandLineTool` tarefa que ecoa o conteúdo STDERR em um arquivo de saída de texto chamado. stderr.txt A tarefa modifica o `baseCommand` para que seja `echo` gravado em STDERR (em vez de STDOUT).

A `outputs` diretiva especifica que o nome da saída é stderr_out e seu tipo é`stderr`.

Como HealthOmics cria registros para todo o conteúdo STDERR e STDOUT, a saída aparecerá em CloudWatch Logs, junto com outras informações de registro STDERR da tarefa.

```
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
```

Saída da tarefa para um arquivo

Este exemplo cria uma `CommandLineTool` tarefa que cria um arquivo tar compactado a partir dos arquivos de entrada. Você fornece o nome do arquivo como um parâmetro de entrada (archive_name).

A outputs diretiva especifica que o tipo de `archive_file` saída é `File` e usa uma referência ao parâmetro de entrada `archive_name` para vincular ao arquivo de saída.

```
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
```

Saída de tarefas para uma matriz de arquivos

Neste exemplo, a `CommandLineTool` tarefa cria uma matriz de arquivos usando o `touch` comando. O comando usa as cadeias de caracteres no parâmetro `files-to-create` de entrada para nomear os arquivos. O comando gera uma matriz de arquivos. A matriz inclui todos os arquivos no diretório de trabalho que correspondam ao `glob` padrão. Este exemplo usa um padrão curinga (“*”) que corresponde a todos os arquivos.

```
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 tarefas em uma definição de HealthOmics fluxo de trabalho

Na definição do fluxo de trabalho, defina o seguinte para cada tarefa:

- A imagem do contêiner para a tarefa. Para obter mais informações, consulte Imagens de contêiner para fluxos de trabalho privados.
- O número CPUs e a memória necessários para a tarefa. Para obter mais informações, consulte Requisitos de computação e memória para tarefas HealthOmics.

HealthOmics ignora todas as especificações de armazenamento por tarefa. HealthOmics fornece armazenamento de execução que todas as tarefas em execução podem acessar. Para obter mais informações, consulte Execute tipos de armazenamento em HealthOmics fluxos de trabalho.

WDL

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

```
}
```

Para um fluxo de trabalho WDL, tente até duas HealthOmics tentativas para uma tarefa que falha devido a erros de serviço (a solicitação de API retorna um código de status HTTP 5XX). Para obter mais informações sobre novas tentativas de tarefas, consulteTentativas de tarefas.

Você pode desativar o comportamento de nova tentativa especificando a seguinte configuração para a tarefa no arquivo de definição da 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 tarefas em uma definição de HealthOmics fluxo de trabalho

Na definição do fluxo de trabalho, você pode especificar opcionalmente a especificação do acelerador de GPU para uma tarefa. HealthOmics é compatível com os seguintes valores de especificação do acelerador, junto com os tipos de instância compatíveis:

Especificação do acelerador	Tipos de instância do Healthomics				
nvidia-tesla-t4	G4				
nvidia-tesla-t4-a 10g	G4 e G5				
nvidia-tesla-a10g	G5				
nvidia-l4-a10g	G5 e G6				
nvidia-l4	G6				
nvidia-l40s	G6e				

Se você especificar um tipo de acelerador que ofereça suporte a vários tipos de instância, HealthOmics selecionará o tipo de instância com base na capacidade disponível. Se os dois tipos de instância estiverem disponíveis, HealthOmics dará preferência à instância de menor custo.

Para obter detalhes sobre os tipos de instância, consulteInstâncias de computação acelerada.

No exemplo a seguir, a definição do fluxo de trabalho especifica `nvidia-l4` 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"
```

Especificidades da definição do fluxo de trabalho da WDL

Os tópicos a seguir fornecem detalhes sobre os tipos e diretivas disponíveis para as definições de fluxo de trabalho da WDL em. HealthOmics

Tópicos

Conversão de tipo implícita em WDL leniente

HealthOmics oferece suporte à conversão de tipo implícita no arquivo input.json e na definição do fluxo de trabalho. Para usar a conversão de tipo implícita, especifique o mecanismo de fluxo

de trabalho como tolerante à WDL ao criar o fluxo de trabalho. O WDL lenient foi projetado para lidar com fluxos de trabalho migrados de Cromwell. Ele suporta as diretivas Cromwell do cliente e algumas lógicas não conformes.

A WDL lenient suporta a conversão de tipos para os seguintes itens na lista de exceções limitadas da WDL:

- Flutue até Int, onde a coerção não resulta em perda de precisão (como 1,0 mapeia para 1).
- String to Int/Float, onde a coerção não resulta em perda de precisão.
- Mapeie [W, X] para Array [Pair [Y, Z]], no caso em que W é coercível para Y e X é coercível para Z.
- Array [Pair [W, X]] para Map [Y, Z], no caso em que W é coercível para Y e X é coercível para Z (como 1,0 mapas para 1).

Para usar a conversão de tipo implícita, especifique o mecanismo do fluxo de trabalho como WDL_LENIENT ao criar o fluxo de trabalho ou a versão do fluxo de trabalho.

No console, o parâmetro do mecanismo de fluxo de trabalho é chamado de Idioma. Na API, o parâmetro do mecanismo de fluxo de trabalho é chamado de mecanismo. Para acessar mais informações, consulte Crie um fluxo de trabalho privado ou Crie uma versão do fluxo de trabalho.

Definição de namespace em input.json

HealthOmics suporta variáveis totalmente qualificadas em input.json. Por exemplo, se você declarar duas variáveis de entrada chamadas número1 e número2 no fluxo de trabalho: SumWorkflow

```
workflow SumWorkflow {
  input {
    Int number1
    Int number2
  }
}
```

Você pode usá-las como variáveis totalmente qualificadas em input.json:

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

```
}
```

Tipos primitivos na Biblioteca Digital Mundial

A tabela a seguir mostra como as entradas na WDL são mapeadas para os tipos primitivos correspondentes. HealthOmics fornece suporte limitado para coerção de tipo, por isso recomendamos que você defina tipos explícitos.

Tipos primitivos

Tipo WDL	Tipo JSON	Exemplo de WDL	Exemplo de chave e valor JSON	Observações
`Boolean`	`boolean`	`Boolean b`	`"b": true`	O valor deve estar em minúsculas e sem aspas.
`Int`	`integer`	`Int i`	`"i": 7`	Não deve ser citado.
`Float`	`number`	`Float f`	`"f": 42.2`	Não deve ser citado.
`String`	`string`	`String s`	`"s": "characters"`	As cadeias de caracteres JSON que são um URI devem ser mapeadas para um arquivo WDL para serem importadas.
`File`	`string`	`File f`	`"f": "s3://amzn-s3-demo-bucket1/`	O Amazon S3 e o HealthOmics armazenamento URIs são importados,

Tipo WDL	Tipo JSON	Exemplo de WDL	Exemplo de chave e valor JSON	Observações
			`path/to/file"`	desde que a função do IAM fornecida para o fluxo de trabalho tenha acesso de leitura a esses objetos. Nenhum outro esquema de URI é suportado (como `file://https://`, `eftp://`). O URI deve especificar um objeto. Não pode ser um diretório, o que significa que não pode terminar com `/` a.

Tipo WDL	Tipo JSON	Exemplo de WDL	Exemplo de chave e valor JSON	Observações
`Directory`	`string`	`Directory d`	`"d": "s3://bucket/path/"`	O `Directory` tipo não está incluído na WDL 1.0 ou 1.1, então você precisará `version development` adicioná-lo ao cabeçalho do arquivo WDL. O URI deve ser um URI do Amazon S3 e com um prefixo que termine com '/'. Todo o conteúdo do diretório será copiado recursivamente para o fluxo de trabalho como um único download. O `Directory` deve conter somente arquivos relacionados ao fluxo de trabalho.

Tipos complexos em WDL

A tabela a seguir mostra como as entradas na WDL são mapeadas para os tipos JSON complexos correspondentes. Os tipos complexos na Biblioteca Digital Mundial são estruturas de dados compostas por tipos primitivos. Estruturas de dados, como listas, serão convertidas em matrizes.

Tipos complexos

Tipo WDL	Tipo JSON	Exemplo de WDL	Exemplo de chave e valor JSON	Observações
`Array`	`array`	`Array[Int] nums`	`“nums": [1, 2, 3]`	Os membros da matriz devem seguir o formato do tipo de matriz WDL.
`Pair`	`object`	`Pair[String, Int] str_to_i`	`“str_to_i": {"left": "0", "right": 1}`	Cada valor do par deve usar o formato JSON do tipo WDL correspondente.
`Map`	`object`	`Map[Int, String] int_to_string`	`"int_to_string": { 2: "hello", 1: "goodbye" }`	Cada entrada no mapa deve usar o formato JSON de seu tipo WDL correspondente.
`Struct`	`object`	`struct SampleBamAndIndex { String sample_name File bam File bam_index`	`"b_and_i": { "sample_name": "NA12878", "bam": "s3://amzn-s3-demo-bucket1/`	Os nomes dos membros da estrutura devem corresponder exatamente aos nomes das chaves do objeto JSON. Cada valor deve usar

Tipo WDL	Tipo JSON	Exemplo de WDL	Exemplo de chave e valor JSON	Observações
		`} SampleBam AndIndex b_and_i`	`NA12878.b am", "bam_inde x": "s3:// amzn- s3-demo- bucket1/ NA12878.b am.bai" }`	o formato JSON do tipo de WDL correspondente.
`Object`	N/D	N/D	N/D	O `Object` tipo WDL está desatualizado e deve ser substituído por `Struct` em todos os casos.

Diretivas na Biblioteca Digital Mundial

HealthOmics suporta as seguintes diretivas em todas as versões da WDL que HealthOmics oferecem suporte.

Configurar recursos de GPU

HealthOmics é compatível com atributos de tempo de execução acceleratorType e acceleratorCount com todas as instâncias de GPU compatíveis. HealthOmics também oferece suporte a aliases chamados gpuType egpuCount, que têm a mesma funcionalidade de seus equivalentes aceleradores. Se a definição da WDL contiver as duas diretivas, HealthOmics use os valores do acelerador.

O exemplo a seguir mostra como usar essas diretivas:

```
runtime {
    gpuCount: 2
```

```
    gpuType: "nvidia-tesla-t4"
}
```

Configurar a repetição de tarefas para erros de serviço

HealthOmics suporta até duas tentativas para uma tarefa que falhou devido a erros de serviço (códigos de status HTTP 5XX). Você pode configurar o número máximo de novas tentativas (1 ou 2) e pode optar por não participar de novas tentativas por erros de serviço. Por padrão, HealthOmics tenta no máximo duas tentativas.

O exemplo a seguir define `preemptible` a opção de não aceitar novas tentativas por erros de serviço:

```
{
  preemptible: 0
}
```

Para obter mais informações sobre novas tentativas de tarefas em HealthOmics, consulteTentativas de tarefas.

Configurar a repetição da tarefa para falta de memória

HealthOmics suporta novas tentativas para uma tarefa que falhou porque ficou sem memória (código de saída do contêiner 137, código de status HTTP 4XX). HealthOmics dobra a quantidade de memória para cada tentativa de repetição.

Por padrão, HealthOmics não tenta novamente para esse tipo de falha. Use a `maxRetries` diretiva para especificar o número máximo de novas tentativas.

O exemplo a seguir é definido `maxRetries` como 3, de modo que HealthOmics tente no máximo quatro tentativas para concluir a tarefa (a tentativa inicial mais três tentativas):

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

> **Note**
>
> A repetição da tarefa em caso de falta de memória requer o GNU findutils 4.2.3+. O contêiner de HealthOmics imagem padrão inclui esse pacote. Se você especificar uma imagem

personalizada em sua definição de WDL, certifique-se de que a imagem inclua GNU findutils 4.2.3+.

Configurar códigos de devolução

O atributo returnCodes fornece um mecanismo para especificar um código de retorno, ou um conjunto de códigos de retorno, que indica a execução bem-sucedida de uma tarefa. O mecanismo da WDL respeita os códigos de retorno que você especifica na seção de tempo de execução da definição da WDL e define o status das tarefas de acordo.

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

HealthOmics também oferece suporte a um alias chamado continueOnReturnCode, que tem os mesmos recursos de ReturnCodes. Se você especificar os dois atributos, HealthOmics usa o valor ReturnCodes.

Metadados de tarefas na WDL

HealthOmics suporta as seguintes opções de metadados para tarefas de WDL.

Desative o armazenamento em cache no nível da tarefa com o atributo volatile

O atributo volatile permite que você desabilite o cache de chamadas para tarefas específicas em seu fluxo de trabalho da WDL. Quando uma tarefa é marcada como volátil, ela sempre será executada e nunca usará resultados em cache, mesmo quando o cache estiver habilitado para execução.

Adicione o atributo volatile à seção meta da definição de sua tarefa:

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

    input {
        String input_file
    }
```

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

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

Exemplo de definição de fluxo de trabalho WDL

Os exemplos a seguir mostram definições de fluxo de trabalho privadas para conversão de `CRAM` para `BAM` em WDL. O `BAM` fluxo de trabalho `CRAM` to define duas tarefas e usa ferramentas do `genomes-in-the-cloud` contêiner, que são mostradas no exemplo e estão disponíveis publicamente.

O exemplo a seguir mostra como incluir o contêiner Amazon ECR como parâmetro. Isso permite HealthOmics verificar as permissões de acesso ao seu contêiner antes que ele inicie a execução.

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

O exemplo a seguir mostra como especificar quais arquivos usar em sua execução, quando os arquivos estão em um bucket do 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"
}
```

Se você quiser especificar arquivos de um armazenamento de sequência, indique isso conforme mostrado no exemplo a seguir, usando o URI para o armazenamento de sequência.

```
{
```

```
    "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"
}
```

Em seguida, você pode definir seu fluxo de trabalho na WDL, conforme mostrado no exemplo a seguir.

```
 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}"
     }
}
```

Especificações da definição do fluxo de trabalho do Nextflow

HealthOmics suporta Nextflow e. DSL1 DSL2 Para obter detalhes, consulte Suporte à versão Nextflow.

O Nextflow DSL2 é baseado na linguagem de programação Groovy, portanto, os parâmetros são dinâmicos e a coerção de tipos é possível usando as mesmas regras do Groovy. Os parâmetros e valores fornecidos pelo JSON de entrada estão disponíveis no mapa parameters (`params`) do fluxo de trabalho.

Tópicos

- Use plug-ins nf-schema e nf-validation
- Especifique o armazenamento URIs
- Diretivas Nextflow
- Exportar conteúdo da tarefa

Use plug-ins nf-schema e nf-validation

> **Note**
>
> Resumo do HealthOmics suporte para plug-ins:

- v22.04 — sem suporte para plug-ins
- v23.10 — suporta e `nf-schema` `nf-validation`
- v24.10 — suporta `nf-schema`

HealthOmics fornece o seguinte suporte para plug-ins Nextflow:

- Para o Nextflow v23.10, HealthOmics pré-instala o plug-in nf-validation @1 .1.1.
- Para o Nextflow v23.10 e versões posteriores, HealthOmics pré-instala o plug-in nf-schema @2 .3.0.
- Você não pode recuperar plug-ins adicionais durante a execução de um fluxo de trabalho. HealthOmics ignora todas as outras versões de plug-in que você especificar no `nextflow.config` arquivo.
- Para o Nextflow v24 e superior, `nf-schema` é a nova versão do plug-in obsoleto. `nf-validation` Para obter mais informações, consulte nf-schema no repositório Nextflow. GitHub

Especifique o armazenamento URIs

Quando um Amazon S3 ou HealthOmics URI é usado para construir um arquivo Nextflow ou objeto de caminho, ele disponibiliza o objeto correspondente para o fluxo de trabalho, desde que o acesso de leitura seja concedido. O uso de prefixos ou diretórios é permitido para o Amazon S3. URIs Para obter exemplos, consulte Formatos de parâmetros de entrada do Amazon S3.

HealthOmics suporta parcialmente o uso de padrões globais no Amazon URIs S3 HealthOmics ou no Storage. URIs Use padrões Glob na definição do fluxo de trabalho para a criação de `path` nossos `file` canais. Para o comportamento esperado e os casos exatos, consulteNextflow: Tratamento do padrão Glob nas entradas do Amazon S3.

Diretivas Nextflow

Você configura as diretivas do Nextflow no arquivo de configuração ou na definição do fluxo de trabalho do Nextflow. A lista a seguir mostra a ordem de precedência HealthOmics usada para aplicar as definições de configuração, da prioridade mais baixa para a mais alta:

1. Configuração global no arquivo de configuração.
2. Seção de tarefas da definição do fluxo de trabalho.
3. Seletores específicos de tarefas no arquivo de configuração.

Tópicos

Estratégia de repetição de tarefas usando **`errorStrategy`**

Use a `errorStrategy` diretiva para definir a estratégia para erros de tarefas. Por padrão, quando uma tarefa retorna com uma indicação de erro (um status de saída diferente de zero), a tarefa para e HealthOmics encerra toda a execução. Se você definir como`retry`, `errorStrategy` HealthOmics tentará uma nova tentativa da tarefa que falhou. Para aumentar o número de novas tentativas, consulteTentativas de repetição de tarefas usando `maxRetries`.

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

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

Para obter informações sobre como HealthOmics manipula novas tentativas de tarefas durante uma execução, consulteTentativas de tarefas.

Tentativas de repetição de tarefas usando **`maxRetries`**

Por padrão, HealthOmics não tenta nenhuma nova tentativa de uma tarefa com falha, nem tenta uma nova tentativa se você configurar. `errorStrategy` Para aumentar o número máximo de novas tentativas, `errorStrategy` defina `retry` e configure o número máximo de novas tentativas usando a `maxRetries` diretiva.

O exemplo a seguir define o número máximo de novas tentativas como 3 na configuração global.

```
process {
    errorStrategy = 'retry'
```

```
    maxRetries = 3
}
```

O exemplo a seguir mostra como definir `maxRetries` na seção de tarefas da definição do fluxo de trabalho.

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

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

O exemplo a seguir mostra como especificar a configuração específica da tarefa no arquivo de configuração do Nextflow, com base nos seletores de nome ou rótulo.

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

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

Opte por não participar da tarefa novamente usando **`omicsRetryOn5xx`**

Para Nextflow v23 e v24, HealthOmics suporta novas tentativas de tarefas se a tarefa falhar devido a erros de serviço (códigos de status HTTP 5XX). Por padrão, HealthOmics tenta até duas tentativas de uma tarefa com falha.

Você pode configurar `omicsRetryOn5xx` para desativar a repetição de tarefas em caso de erros de serviço. Para obter mais informações sobre a repetição de tarefas HealthOmics, consulte Tentativas de tarefas.

O exemplo a seguir é configurado `omicsRetryOn5xx` na configuração global para desativar a repetição da tarefa.

```
process {
    omicsRetryOn5xx = false
}
```

O exemplo a seguir mostra como configurar `omicsRetryOn5xx` na seção de tarefas da definição do fluxo de trabalho.

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

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

O exemplo a seguir mostra `omicsRetryOn5xx` como definir uma configuração específica da tarefa no arquivo de configuração do Nextflow, com base nos seletores de nome ou rótulo.

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

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

Duração da tarefa usando a **`time`** diretiva

HealthOmics fornece uma cota ajustável (consulteHealthOmics cotas de serviço) para especificar a duração máxima de uma execução. Para fluxos de trabalho Nextflow v23 e v24, você também pode especificar durações máximas de tarefas usando a diretiva Nextflow. `time`

Durante o desenvolvimento de um novo fluxo de trabalho, definir a duração máxima da tarefa ajuda você a capturar tarefas descontroladas e tarefas de longa execução.

Para obter mais informações sobre a diretiva de tempo do Nextflow, consulte a diretiva de tempo na referência do Nextflow.

HealthOmics fornece o seguinte suporte para a diretiva de tempo Nextflow:

1. HealthOmics suporta granularidade de 1 minuto para a diretiva de tempo. Você pode especificar um valor entre 60 segundos e o valor máximo da duração da execução.
2. Se você inserir um valor menor que 60, o HealthOmics arredonda para 60 segundos. Para valores acima de 60, HealthOmics arredonda para baixo para o minuto mais próximo.
3. Se o fluxo de trabalho suportar novas tentativas para uma tarefa, HealthOmics tente novamente a tarefa se o tempo limite atingir o tempo limite.
4. Se uma tarefa atingir o tempo limite (ou se a última tentativa atingir o tempo limite), a tarefa HealthOmics será cancelada. Essa operação pode ter uma duração de um a dois minutos.
5. No tempo limite da tarefa, HealthOmics define a execução e o status da tarefa como falha e cancela as outras tarefas na execução (para tarefas com status Inicial, Pendente ou Em execução). HealthOmics exporta as saídas das tarefas concluídas antes do tempo limite para o local de saída designado do S3.
6. O tempo que uma tarefa passa no status pendente não conta para a duração da tarefa.
7. Se a execução fizer parte de um grupo de execução e o grupo de execução expirar antes do cronômetro da tarefa, a execução e a tarefa passarão para o status de falha.

Especifique a duração do tempo limite usando uma ou mais das seguintes unidades:`ms`,,`s`, `mh`, ou`d`.

O exemplo a seguir mostra como especificar a configuração global no arquivo de configuração do Nextflow. Ele define um tempo limite global de 1 hora e 30 minutos.

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

O exemplo a seguir mostra como especificar uma diretiva de horário na seção de tarefas da definição do fluxo de trabalho. Este exemplo define um tempo limite de 3 dias, 5 horas e 4 minutos. Esse valor tem precedência sobre o valor global no arquivo de configuração, mas não tem precedência sobre uma diretiva de tempo específica da tarefa no arquivo de `my_label` configuração.

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

```
    time '3d5h4m'

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

O exemplo a seguir mostra como especificar diretivas de horário específicas da tarefa no arquivo de configuração do Nextflow, com base nos seletores de nome ou rótulo. Este exemplo define um valor global de tempo limite da tarefa de 30 minutos. Ele define um valor de 2 horas para a tarefa `myTask` e define um valor de 3 horas para tarefas com rótulo`my_label`. Para tarefas que correspondem ao seletor, esses valores têm precedência sobre o valor global e o valor na definição do fluxo de trabalho.

```
process {
    time = '30m'

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

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

Exportar conteúdo da tarefa

Para fluxos de trabalho escritos em Nextflow, defina uma diretiva PublishDir para exportar o conteúdo da tarefa para seu bucket de saída do Amazon S3. Conforme mostrado no exemplo a seguir, defina o valor publishDir como`/mnt/workflow/pubdir`. Para exportar arquivos para o Amazon S3, os arquivos devem estar nesse diretório.

```
 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
    """
}
```

Especificidades da definição do fluxo de trabalho do CWL

Os fluxos de trabalho escritos em Common Workflow Language, ou CWL, oferecem funcionalidade semelhante aos fluxos de trabalho escritos em WDL e Nextflow. Você pode usar o Amazon S3 ou o HealthOmics armazenamento URIs como parâmetros de entrada.

Se você definir a entrada em um SecondaryFile em um subfluxo de trabalho, adicione a mesma definição no fluxo de trabalho principal.

HealthOmics os fluxos de trabalho não oferecem suporte aos processos operacionais. Para saber mais sobre os processos operacionais nos fluxos de trabalho da CWL, consulte a documentação da CWL.

A melhor prática é definir um fluxo de trabalho CWL separado para cada contêiner que você usa. Recomendamos que você não codifique a entrada DockerPull com um URI fixo do Amazon ECR.

Tópicos

Converta fluxos de trabalho CWL para uso HealthOmics

Para converter uma definição de fluxo de trabalho CWL existente para uso HealthOmics, faça as seguintes alterações:

- Substitua todo o contêiner URIs Docker pelo Amazon URIs ECR.
- Certifique-se de que todos os arquivos do fluxo de trabalho sejam declarados no fluxo de trabalho principal como entrada e que todas as variáveis estejam definidas explicitamente.
- Certifique-se de que todo JavaScript código seja uma reclamação de modo estrito.

Opte por não participar da tarefa novamente usando **`omicsRetryOn5xx`**

HealthOmics suporta novas tentativas de tarefas se a tarefa falhar devido a erros de serviço (códigos de status HTTP 5XX). Por padrão, HealthOmics tenta até duas tentativas de uma tarefa com falha. Para obter mais informações sobre a repetição de tarefas HealthOmics, consulteTentativas de tarefas.

Para desativar a repetição da tarefa devido a erros de serviço, configure a `omicsRetryOn5xx` diretiva na definição do fluxo de trabalho. Você pode definir essa diretiva em requisitos ou dicas. Recomendamos adicionar a diretiva como uma dica de portabilidade.

```
requirements:
  ResourceRequirement:
    omicsRetryOn5xx: false

hints:
  ResourceRequirement:
    omicsRetryOn5xx: false
```

Os requisitos substituem as dicas. Se a implementação de uma tarefa fornece um requisito de recurso em dicas que também é fornecido pelos requisitos em um fluxo de trabalho envolvente, os requisitos anexos têm precedência.

Se o mesmo requisito de tarefa aparecer em níveis diferentes do fluxo de trabalho, HealthOmics usa a entrada mais específica de `requirements` (ou`hints`, se não houver entradas`requirements`). A lista a seguir mostra a ordem de precedência HealthOmics usada para aplicar as definições de configuração, da prioridade mais baixa para a mais alta:

- Nível do fluxo de trabalho
- Nível de etapa
- Seção de tarefas da definição do fluxo de trabalho

O exemplo a seguir mostra como configurar a `omicsRetryOn5xx` diretiva em diferentes níveis do fluxo de trabalho. Neste exemplo, o requisito no nível do fluxo de trabalho substitui as dicas no nível do fluxo de trabalho. As configurações de requisitos nos níveis de tarefa e etapa substituem as configurações de dicas.

```
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
```

Repetir uma etapa do fluxo de trabalho

HealthOmics suporta o loop de uma etapa do fluxo de trabalho. Você pode usar loops para executar etapas do fluxo de trabalho repetidamente até que uma condição especificada seja atendida. Isso é útil para processos iterativos em que você precisa repetir uma tarefa várias vezes ou até que um determinado resultado seja alcançado.

Nota: A funcionalidade de loop requer a versão 1.2 ou posterior do CWL. Os fluxos de trabalho usando versões do CWL anteriores à 1.2 não oferecem suporte a operações de loop.

Para usar loops em seu fluxo de trabalho CWL, defina um requisito de loop. O exemplo a seguir mostra a configuração dos requisitos de loop:

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

O `loopWhen` campo controla quando o loop termina. Neste exemplo, o loop continua enquanto o contador for menor que o valor máximo. O `loop` campo define como os parâmetros de entrada são atualizados entre as iterações. O `loopSource` especifica qual saída da iteração anterior alimenta a próxima iteração. O `outputMethod` campo definido como `last` retorna somente a saída da iteração final.

Repita as tarefas com maior memória

HealthOmics suporta repetição automática de falhas de out-of-memory tarefas. Quando uma tarefa sai com o código 137 (out-of-memory), HealthOmics cria uma nova tarefa com maior alocação de memória com base no multiplicador especificado.

> **Note**
>
> HealthOmics repete as out-of-memory falhas até 3 vezes ou até que a alocação de memória alcance 1536 GiB, qualquer que seja o limite atingido primeiro.

O exemplo a seguir mostra como configurar a nova out-of-memory tentativa:

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

Quando uma tarefa falha devido a out-of-memory, HealthOmics calcula a alocação de memória de repetição usando a fórmula:. `previous_run_memory × memoryRetryMultiplier` No exemplo acima, se a tarefa com 4096 MB de memória falhar, a nova tentativa usará 4096 × 2,5 = 10.240 MB de memória.

O `memoryRetryMultiplier` parâmetro controla a quantidade de memória adicional a ser alocada para tentativas de repetição:

- Valor padrão: se você não especificar um valor, o padrão é 2 (dobra a memória)
- Intervalo válido: deve ser um número positivo maior que1. Valores inválidos resultam em um erro de validação 4XX
- Valor efetivo mínimo: valores entre 1 e `1.5` são aumentados automaticamente `1.5` para garantir aumentos significativos de memória e evitar tentativas excessivas de repetição

Exemplos

Veja a seguir um exemplo de um fluxo de trabalho escrito em 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
```

O arquivo a seguir define a `copy.cwl` tarefa.

```
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)"
```

Veja a seguir um exemplo de um fluxo de trabalho escrito em CWL com um 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: []
```

Exemplos de definições de fluxo de trabalho

O exemplo a seguir mostra a mesma definição de fluxo de trabalho em WDL, Nextflow e 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
```

Arquivos de modelo de parâmetros para HealthOmics fluxos de trabalho

Os modelos de parâmetros definem os parâmetros de entrada para um fluxo de trabalho. Você pode definir parâmetros de entrada para tornar seu fluxo de trabalho mais flexível e versátil. Por exemplo, você pode definir um parâmetro para a localização dos arquivos do genoma de referência no Amazon S3. Os modelos de parâmetros podem ser fornecidos por meio de um serviço de repositório baseado em Git ou de sua unidade local. Os usuários podem então executar o fluxo de trabalho usando vários conjuntos de dados.

Você pode criar o modelo de parâmetro para seu fluxo de trabalho ou HealthOmics gerar o modelo de parâmetro para você.

O modelo de parâmetro é um arquivo JSON. No arquivo, cada parâmetro de entrada é um objeto nomeado que deve corresponder ao nome da entrada do fluxo de trabalho. Ao iniciar uma execução, se você não fornecer valores para todos os parâmetros necessários, a execução falhará.

O objeto do parâmetro de entrada inclui os seguintes atributos:

- description— Esse atributo obrigatório é uma string que o console exibe na página Iniciar execução. Essa descrição também é mantida como metadados de execução.
- optional— Esse atributo opcional indica se o parâmetro de entrada é opcional. Se você não especificar o optional campo, o parâmetro de entrada será obrigatório.

O exemplo de modelo de parâmetro a seguir mostra como especificar os parâmetros 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
  }
}
```

Geração de modelos de parâmetros

HealthOmics gera o modelo de parâmetros analisando a definição do fluxo de trabalho para detectar os parâmetros de entrada. Se você fornecer um arquivo de modelo de parâmetros para um fluxo de trabalho, os parâmetros em seu arquivo substituirão os parâmetros detectados na definição do fluxo de trabalho.

Há pequenas diferenças entre a lógica de análise dos mecanismos CWL, WDL e Nextflow, conforme descrito nas seções a seguir.

Tópicos

Detecção de parâmetros para CWL

No mecanismo de fluxo de trabalho da CWL, a lógica de análise faz as seguintes suposições:

- Todos os tipos anuláveis suportados são marcados como parâmetros de entrada opcionais.
- Todos os tipos não nulos suportados são marcados como parâmetros de entrada obrigatórios.
- Todos os parâmetros com valores padrão são marcados como parâmetros de entrada opcionais.
- As descrições são extraídas da `label` seção da definição do `main` fluxo de trabalho. Se não `label` for especificado, a descrição ficará em branco (uma string vazia).

As tabelas a seguir mostram exemplos de interpolação de CWL. Para cada exemplo, o nome do parâmetro é `x`. Se o parâmetro for necessário, você deverá fornecer um valor para o parâmetro. Se o parâmetro for opcional, você não precisará fornecer um valor.

Esta tabela mostra exemplos de interpolação CWL para tipos primitivos.

Entrada	Exemplo de entrada/saída	Obrigatório
`x:` `  type: int`	1 ou 2 ou...	Sim

`x:`
`  type: int`
`  default: 2`
	O valor padrão é 2. A entrada válida é 1 ou 2 ou...	Não

`x:` `  type: int?`	A entrada válida é Nenhuma ou 1 ou 2 ou...	Não

`x:`
`  type: int?`
`  default: 2`
	O valor padrão é 2. A entrada válida é Nenhuma ou 1 ou 2 ou...	Não

A tabela a seguir mostra exemplos de interpolação CWL para tipos complexos. Um tipo complexo é uma coleção de tipos primitivos.

Entrada	Exemplo de entrada/saída	Obrigatório

`x:`
`  type: array`
`  items: int`
	[] ou [1,2,3]	Sim

`x:`
`  type: array?`
`  items: int`
	Nenhum ou [] ou [1,2,3]	Não

`x:`
`  type: array`
`  items: int?`
	[] ou [Nenhum, 3, Nenhum]	Sim

Entrada	Exemplo de entrada/saída	Obrigatório

`x:`
`  type: array?`
`  items: int?`
	[Nenhum] ou Nenhum ou [1,2,3] ou [Nenhum, 3] mas não []	Não

Detecção de parâmetros para WDL

No mecanismo de fluxo de trabalho da WDL, a lógica de análise faz as seguintes suposições:

- Todos os tipos anuláveis suportados são marcados como parâmetros de entrada opcionais.
- Para tipos suportados não anuláveis:
  - Qualquer variável de entrada com atribuição de literais ou expressão é marcada como parâmetros opcionais. Por exemplo:

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

  - Se nenhum valor ou expressão tiver sido atribuído aos parâmetros de entrada, eles serão marcados como parâmetros obrigatórios.
- As descrições são extraídas da definição `parameter_meta` do `main` fluxo de trabalho. Se não `parameter_meta` for especificado, a descrição ficará em branco (uma string vazia). Para obter mais informações, consulte a especificação WDL para metadados de parâmetros.

As tabelas a seguir mostram exemplos de interpolação de WDL. Para cada exemplo, o nome do parâmetro é `x`. Se o parâmetro for necessário, você deverá fornecer um valor para o parâmetro. Se o parâmetro for opcional, você não precisará fornecer um valor.

Esta tabela mostra exemplos de interpolação de WDL para tipos primitivos.

Entrada	Exemplo de entrada/saída	Obrigatório
Int x	1 ou 2 ou...	Sim
Int x = 2	2	Não
Int x = 1+2	3	Não

Entrada	Exemplo de entrada/saída	Obrigatório
Int x = y+z	y+z	Não
Int? x	Nenhum ou 1 ou 2 ou...	Sim
Int? x = 2	Nenhum ou 2	Não
Int? x = 1+2	Nenhum ou 3	Não
Int? x = y+z	Nenhum ou y+z	Não

A tabela a seguir mostra exemplos de interpolação de WDL para tipos complexos. Um tipo complexo é uma coleção de tipos primitivos.

Entrada	Exemplo de entrada/saída	Obrigatório		
Matriz [Int] x	[1,2,3] ou []	Sim		
Matriz [Int] + x	[1], mas não []	Sim		
Matriz [Int]? x	Nenhum ou [] ou [1,2,3]	Não		
Matriz [Int?] x	[] ou [Nenhum, 3, Nenhum]	Sim		
Matriz [Int?] =? x	[Nenhum] ou Nenhum ou [1,2,3] ou [Nenhum, 3] mas não []	Não		
Amostra de estrutura {String a, Int y}	`String a = mySample.a Int y = mySample.y`	Sim		

Entrada	Exemplo de entrada/saída	Obrigatório		
posteriormente nas entradas: Sample MySample				
Amostra de estrutura {String a, Int y} posteriormente nas entradas: Amostra? Minha amostra	`if (defined( mySample)) { String a = mySample.a Int y = mySample.y }`	Não		

Detecção de parâmetros para Nextflow

Para o Nextflow, HealthOmics gera o modelo de parâmetro analisando o arquivo. `nextflow_schema.json` Se a definição do fluxo de trabalho não incluir um arquivo de esquema, HealthOmics analisará o arquivo principal de definição do fluxo de trabalho.

Tópicos

- Analisando o arquivo do esquema
- Analisando o arquivo principal
- Parâmetros aninhados
- Exemplos de interpolação Nextflow

Analisando o arquivo do esquema

Para que a análise funcione corretamente, verifique se o arquivo do esquema atende aos seguintes requisitos:

- O arquivo do esquema tem um nome `nextflow_schema.json` e está localizado no mesmo diretório do arquivo do fluxo de trabalho principal.

- O arquivo do esquema é um JSON válido, conforme definido em um dos seguintes esquemas:
  - esquema json. org/draft/2020-12/schema.
  - esquema json. org/draft-07/schema.

HealthOmics analisa o `nextflow_schema.json` arquivo para gerar o modelo de parâmetro:

- Extrai tudo properties o que está definido no esquema.
- Inclui a propriedade, description se disponível para a propriedade.
- Identifica se cada parâmetro é opcional ou obrigatório, com base no required campo da propriedade.

O exemplo a seguir mostra um arquivo de definição e o arquivo de parâmetros gerado.

```
{
    "$schema": "https://json-schema.org/draft/2020-12/schema",
    "type": "object",
    "$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" }
    ]
}
```

O modelo de parâmetro gerado:

```
{
    "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
    }
}
```

Analisando o arquivo principal

Se a definição do fluxo de trabalho não incluir um `nextflow_schema.json` arquivo, HealthOmics analisará o arquivo principal de definição do fluxo de trabalho.

HealthOmics analisa as `params` expressões encontradas no arquivo principal de definição do fluxo de trabalho e no `nextflow.config` arquivo. Todos `params` com valores padrão são marcados como opcionais.

Para que a análise funcione corretamente, observe os seguintes requisitos:

- HealthOmics analisa somente o arquivo de definição do fluxo de trabalho principal. Para garantir que todos os parâmetros sejam capturados, recomendamos que você conecte tudo a todos params os submódulos e fluxos de trabalho importados.
- O arquivo de configuração é opcional. Se você definir um, nomeie-o `nextflow.config` e coloque-o no mesmo diretório do arquivo principal de definição do fluxo de trabalho.

O exemplo a seguir mostra um arquivo de definição e o modelo de parâmetro gerado.

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

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

O modelo de parâmetro gerado:

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

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

Para valores padrão definidos em nextflow.config, HealthOmics coleta `params` atribuições e parâmetros declarados em, conforme mostrado no `params {}` exemplo a seguir. Nas declarações de atribuição, `params` devem aparecer no lado esquerdo da declaração.

```
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}"
```

O modelo de parâmetro gerado:

```
{
    // 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
    }
}
```

Parâmetros aninhados

Ambos `nextflow_schema.json` e `nextflow.config` permitem parâmetros aninhados. No entanto, o modelo de HealthOmics parâmetros requer somente os parâmetros de nível superior. Se seu fluxo de trabalho usa um parâmetro aninhado, você deve fornecer um objeto JSON como entrada para esse parâmetro.

Parâmetros aninhados em arquivos de esquema

HealthOmics pula aninhado params ao analisar um arquivo. `nextflow_schema.json` Por exemplo, se você definir o seguinte `nextflow_schema.json` arquivo:

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

HealthOmics ignora `input_file` e `input_num` quando gera o modelo de parâmetro:

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

```
        "optional": True
    }
}
```

Quando você executa esse fluxo de trabalho, HealthOmics espera um `input.json` arquivo semelhante ao seguinte:

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

Parâmetros aninhados em arquivos de configuração

HealthOmics não coleta aninhados params em um `nextflow.config` arquivo e os ignora durante a análise. Por exemplo, se você definir o seguinte `nextflow.config` arquivo:

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

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

HealthOmics ignora `params.nested.beta` e `params.group.delta` quando gera o modelo de parâmetro:

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

```
    }
}
```

Exemplos de interpolação Nextflow

A tabela a seguir mostra exemplos de interpolação do Nextflow para parâmetros no arquivo principal.

Parâmetros	Obrigatório
params.input_file	Sim
params.input_file = "s3://bucket/data.json”	Não
params.nested.input_file	N/D
params.nested.input_file = "s3://bucket/data.json”	N/D

A tabela a seguir mostra exemplos de interpolação do Nextflow para parâmetros no arquivo. `nextflow.config`

Parâmetros	Obrigatório
`params.input_file = "s3://bucket/data.json"`	Não
`params { input_file = "s3://bucket/data.json" }`	Não
`params { nested { input_file = "s3://bucket/data.json" } }`	N/D

Parâmetros	Obrigatório
`input_file = params.input_file`	N/D

Imagens de contêiner para fluxos de trabalho privados

HealthOmics oferece suporte a imagens de contêineres hospedadas em repositórios privados do Amazon ECR. Você pode criar imagens de contêiner e enviá-las para o repositório privado. Você também pode usar seu registro privado do Amazon ECR como um cache para sincronizar o conteúdo dos registros upstream.

Seu repositório Amazon ECR deve residir na mesma AWS região da conta que está chamando o serviço. Uma pessoa diferente Conta da AWS pode possuir a imagem do contêiner, desde que o repositório de imagens de origem forneça as permissões apropriadas. Para obter mais informações, consulte Políticas para acesso entre contas ao Amazon ECR.

Recomendamos que você defina a imagem do contêiner do Amazon ECR URIs como parâmetros em seu fluxo de trabalho para que o acesso possa ser verificado antes do início da execução. Também facilita a execução de um fluxo de trabalho em uma nova região alterando o parâmetro Região.

> **Note**
>
> HealthOmics não oferece suporte a contêineres ARM e não oferece suporte ao acesso a repositórios públicos.

Para obter informações sobre como configurar as permissões do IAM HealthOmics para acessar o Amazon ECR, consulte. HealthOmics Permissões de recursos

Tópicos

- Sincronização com registros de contêineres de terceiros
- Considerações gerais sobre imagens de contêineres do Amazon ECR
- Variáveis de ambiente para HealthOmics fluxos de trabalho
- Usando Java em imagens de contêiner do Amazon ECR
- Adicionar entradas de tarefas a uma imagem de contêiner do Amazon ECR

Sincronização com registros de contêineres de terceiros

Você pode usar as regras de cache pull through do Amazon ECR para sincronizar repositórios em um registro upstream compatível com seus repositórios privados do Amazon ECR. Para obter mais informações, consulte Sincronizar um registro upstream no Guia do usuário do Amazon ECR.

O cache de extração cria automaticamente o repositório de imagens em seu registro privado quando você cria o cache e sincroniza automaticamente com a imagem em cache quando há alterações na imagem upstream.

HealthOmics suporta cache pull through para os seguintes registros upstream:

- Amazon ECR Public
- Registro de imagens de contêineres do Kubernetes
- Quay
- Docker Hub
- Microsoft Azure Container Registry
- GitHub Registro de contêiner
- GitLab Registro de contêiner

HealthOmics não suporta cache pull through para um repositório privado upstream do Amazon ECR.

Os benefícios de usar o cache pull through do Amazon ECR incluem:

1. Você evita ter que migrar manualmente as imagens do contêiner para o Amazon ECR ou sincronizar atualizações do repositório de terceiros.
2. Os fluxos de trabalho acessam as imagens sincronizadas do contêiner em seu repositório privado, o que é mais confiável do que baixar conteúdo em tempo de execução de um registro público.
3. Como os caches pull through do Amazon ECR usam uma estrutura de URI previsível, o HealthOmics serviço pode mapear automaticamente o URI privado do Amazon ECR com o URI do registro upstream. Você não precisa atualizar e substituir os valores de URI na definição do fluxo de trabalho.

Tópicos

- Configurando o cache pull through
- Mapeamentos de registro

- Mapeamentos de imagens

Configurando o cache pull through

O Amazon ECR fornece um registro para você Conta da AWS em cada região. Certifique-se de criar a configuração do Amazon ECR na mesma região em que planeja executar o fluxo de trabalho.

As seções a seguir descrevem as tarefas de configuração do cache pull through.

Tarefas de configuração

- Crie uma regra de cache de pull through
- Permissões de registro para registro upstream
- Modelos de criação de repositório
- Criação do fluxo de trabalho

Crie uma regra de cache de pull through

Crie uma regra de cache pull through do Amazon ECR para cada registro upstream que tenha imagens que você deseja armazenar em cache. Uma regra especifica um mapeamento entre um registro upstream e o repositório privado do Amazon ECR.

Para um registro upstream que exija autenticação, você fornece suas credenciais usando o AWS Secrets Manager.

> **Note**
>
> Não altere uma regra de cache de pull through enquanto uma execução ativa estiver usando o repositório privado. A execução pode falhar ou, mais criticamente, fazer com que seu pipeline use imagens inesperadas.

Para obter mais informações, consulte Criação de uma regra de cache pull through no Guia do usuário do Amazon Elastic Container Registry.

Crie uma regra de cache de pull through usando o console

Para configurar o cache pull through, siga estas etapas usando o console do Amazon ECR:

1. Abra o console do Amazon ECR: https://console.aws.amazon.com /ecr

2. No menu à esquerda, em Registro privado, expanda Recursos e configurações. Em seguida, escolha Extrair pelo cache.
3. Na página Pull through cache, escolha Adicionar regra.
4. No painel Registro upstream, escolha o registro upstream para sincronizar com seu registro privado e, em seguida, escolha Avançar.
5. Se o registro upstream exigir autenticação, o console abrirá uma nova página na qual você especifica o segredo de SageMaker IA que contém suas credenciais. Escolha Próximo.
6. Em Especificar namespaces, no painel Namespace Cache, escolha se deseja criar os repositórios privados usando um prefixo de repositório específico ou sem prefixo. Se você optar por usar um prefixo, especifique o nome do prefixo em Prefixo do repositório de cache.
7. No painel do namespace Upstream, escolha se deseja extrair dos repositórios upstream usando um prefixo de repositório específico ou sem prefixo. Se você optar por usar um prefixo, especifique o nome do prefixo no prefixo do repositório Upstream.

   O painel de exemplo de Namespace mostra um exemplo de pull request, URL upstream e a URL do repositório de cache criado.
8. Escolha Próximo.
9. Revise a configuração e escolha Criar para criar a regra.

Para obter mais informações, consulte Criar uma regra de cache de pull through (AWS Management Console).

Crie uma regra de cache de pull through usando a CLI

Use o create-pull-through-cache-rule comando Amazon ECR para criar uma regra de cache pull through. Para registros upstream que exigem autenticação, armazene as credenciais em um segredo do Secrets Manager.

As seções a seguir fornecem exemplos para cada registro upstream compatível.

Para o Amazon ECR Public

O exemplo a seguir cria uma regra de cache de pull-through para o registro público do Amazon ECR. Ele especifica um prefixo de repositório de `ecr-public`, o que faz com que cada repositório criado usando a regra de cache de pull-through receba o esquema de nomes 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 registro de contêineres Kubernetes

O exemplo a seguir cria uma regra de cache de pull-through para o registro público do Kubernetes. Ele especifica um prefixo de repositório de `kubernetes`, o que faz com que cada repositório criado usando a regra de cache de pull-through receba o esquema de nomes 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

O exemplo a seguir cria uma regra de cache de pull-through para o registro público do Quay. Ele especifica um prefixo de repositório de `quay`, o que faz com que cada repositório criado usando a regra de cache de pull-through receba o esquema de nomes 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 o Docker Hub

O exemplo a seguir cria uma regra de cache de pull-through para o registro do Docker Hub. Ele especifica um prefixo de repositório de `docker-hub`, o que faz com que cada repositório criado usando a regra de cache de pull-through receba o esquema de nomes de `docker-hub/`*upstream-repository-name*. É necessário especificar o nome completo do Amazon Resource Name (ARN) do segredo que contém suas credenciais do 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

O exemplo a seguir cria uma regra de cache pull through para o GitHub Container Registry. Ele especifica um prefixo de repositório de `github`, o que faz com que cada repositório criado usando a regra de cache de pull-through receba o esquema de nomes de `github/`*`upstream-repository-name`*. Você deve especificar o Amazon Resource Name (ARN) completo do segredo que contém suas credenciais do 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 o Registro de Contêiner do Microsoft Azure

O exemplo a seguir cria uma regra de cache de pull-through para o Microsoft Azure Container Registry. Ele especifica um prefixo de repositório de `azure`, o que faz com que cada repositório criado usando a regra de cache de pull-through receba o esquema de nomes de `azure/`*`upstream-repository-name`*. É necessário especificar o nome completo do Amazon Resource Name (ARN) do segredo que contém suas credenciais do Docker Hub.

```
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

O exemplo a seguir cria uma regra de cache pull through para o GitLab Container Registry. Ele especifica um prefixo de repositório de `gitlab`, o que faz com que cada repositório criado usando a regra de cache de pull-through receba o esquema de nomes de `gitlab/`*`upstream-repository-name`*. Você deve especificar o Amazon Resource Name (ARN) completo do segredo que contém suas credenciais do 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 obter mais informações, consulte Criar uma regra de cache pull through (CLI) no Guia do usuário do Amazon ECR.

Você pode usar o comando get-run-task CLI para recuperar informações sobre a imagem do contêiner usada para uma tarefa específica:

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

A saída inclui as seguintes informações sobre a imagem do contêiner:

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

Permissões de registro para registro upstream

Use as permissões de registro HealthOmics para permitir o uso do cache de extração e a inserção das imagens do contêiner no registro privado do Amazon ECR. Adicione uma política de registro do Amazon ECR ao registro que fornece os contêineres usados nas execuções.

A política a seguir concede permissão para que o HealthOmics serviço crie repositórios com os prefixos de cache pull through especificados e inicie pulls upstream nesses repositórios.

1. No console do Amazon ECR, abra o menu à esquerda, em Registro privado, expanda Permissões do registro. Em seguida, escolha Gerar declaração.
2. No canto superior direito, escolha JSON. Insira uma política semelhante à seguinte:

   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/*"
            ]
        }
    ]
}
```

Modelos de criação de repositório

Para usar o cache pull through HealthOmics, o repositório Amazon ECR deve ter um modelo de criação de repositório. O modelo define as configurações para quando você ou o Amazon ECR criam um repositório privado para um registro upstream.

Cada modelo contém um prefixo de namespace de repositório, que o Amazon ECR usa para combinar novos repositórios com um modelo específico. Os modelos especificam a configuração de todas as configurações do repositório, incluindo políticas de acesso baseadas em recursos, imutabilidade de tags, criptografia e políticas de ciclo de vida.

Para obter mais informações, consulte Modelos de criação de repositórios no Guia do usuário do Amazon Elastic Container Registry.

Como criar um modelo de criação de repositório:

1. No console do Amazon ECR, abra o menu à esquerda, em Registro privado, expanda Recursos e configurações. Em seguida, escolha Modelos de criação de repositório.
2. Selecione Criar modelo.
3. Em Detalhes do modelo, escolha Extrair cache.

4. Escolha se deseja aplicar esse modelo a um prefixo específico ou a todos os repositórios que não correspondam a outro modelo.

   Se você escolher Um prefixo específico, insira o valor do prefixo do namespace em Prefixo. Você especificou esse prefixo ao criar a regra PTC.

5. Escolha Próximo.
6. Na página Adicionar configuração de criação de repositório, insira Permissões do repositório. Use um dos exemplos de declarações de política ou insira um semelhante ao exemplo a seguir:

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "PTCRepoCreationTemplate",
            "Effect": "Allow",
            "Principal": {
                "Service": "omics.amazonaws.com"
            },
            "Action": [
                "ecr:BatchGetImage",
                "ecr:GetDownloadUrlForLayer"
            ],
            "Resource": "*"
        }
    ]
}
```

7. Opcionalmente, você pode adicionar configurações do repositório, como políticas de ciclo de vida e tags. O Amazon ECR aplica essas regras a todas as imagens de contêiner criadas para o cache pull through que usam o prefixo especificado.
8. Escolha Próximo.
9. Revise a configuração e escolha Avançar.

Criação do fluxo de trabalho

Ao criar um novo fluxo de trabalho ou versão do fluxo de trabalho, revise os mapeamentos do registro e atualize-os, se necessário. Para obter detalhes, consulte Crie um fluxo de trabalho privado.

Mapeamentos de registro

Você define mapeamentos de registro para mapear entre prefixos em seu registro privado do Amazon ECR e os nomes de registro upstream.

Para obter mais informações sobre mapeamentos de registro do Amazon ECR, consulte Criação de uma regra de cache pull through no Amazon ECR.

O exemplo a seguir mostra mapeamentos de registro para Docker Hub, Quay e Amazon ECR Public.

```
{
    "registryMappings": [
        {
            "upstreamRegistryUrl": "registry-1.docker.io",
            "ecrRepositoryPrefix": "docker-hub"
        },
        {
            "upstreamRegistryUrl": "quay.io",
            "ecrRepositoryPrefix": "quay"
        },
        {
            "upstreamRegistryUrl": "public.ecr.aws",
            "ecrRepositoryPrefix": "ecr-public"
        }
    ]
}
```

Mapeamentos de imagens

Você define mapeamentos de imagem para mapear entre os nomes das imagens, conforme definido em seus fluxos de trabalho privados do Amazon ECR, e os nomes das imagens no registro upstream.

Você pode usar mapeamentos de imagem com registros que oferecem suporte ao cache pull through. Você também pode usar mapeamentos de imagem com registros upstream que HealthOmics não oferecem suporte para pull through cache. Você precisa sincronizar manualmente o registro upstream com seu repositório privado.

Para obter mais informações sobre mapeamentos de imagens do Amazon ECR, consulte Criação de uma regra de cache pull through no Amazon ECR.

O exemplo a seguir mostra mapeamentos de imagens privadas do Amazon ECR para uma imagem genômica pública e a imagem mais recente do 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",
        }
    ]
}
```

Considerações gerais sobre imagens de contêineres do Amazon ECR

- Arquitetura

  HealthOmics suporta contêineres x86_64. Se sua máquina local for baseada em ARM, como o Apple Mac, use um comando como o seguinte para criar uma imagem de contêiner x86_64:

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

- Ponto de entrada e concha

  HealthOmics os mecanismos de fluxo de trabalho injetam scripts bash como uma substituição de comando nas imagens de contêiner usadas pelas tarefas do fluxo de trabalho. Portanto, as imagens de contêiner devem ser criadas sem um ENTRYPOINT especificado, de forma que um shell bash seja o padrão.

- Caminhos montados

  Um sistema de arquivos compartilhado é montado nas tarefas do contêiner em /tmp. Todos os dados ou ferramentas incorporados à imagem do contêiner nesse local serão substituídos.

  A definição do fluxo de trabalho está disponível para tarefas por meio de uma montagem somente para leitura em /mnt/workflow.

- Tamanho da imagem

Consulte HealthOmics cotas de tamanho fixo do fluxo de trabalho os tamanhos máximos de imagem do contêiner.

Variáveis de ambiente para HealthOmics fluxos de trabalho

HealthOmics fornece variáveis de ambiente que têm informações sobre o fluxo de trabalho em execução no contêiner. Você pode usar os valores dessas variáveis na lógica das tarefas do seu fluxo de trabalho.

Todas as variáveis do HealthOmics fluxo de trabalho começam com o `AWS_WORKFLOW_` prefixo. Esse prefixo é um prefixo de variável de ambiente protegido. Não use esse prefixo para suas próprias variáveis em contêineres de fluxo de trabalho.

HealthOmics fornece as seguintes variáveis de ambiente de fluxo de trabalho:

AWS_REGION

Essa variável é a região em que o contêiner está sendo executado.

AWS_WORKFLOW_EXECUTAR

Essa variável é o nome da execução atual.

AWS_WORKFLOW_ID DE EXECUÇÃO

Essa variável é o identificador da execução atual.

AWS_WORKFLOW_EXECUTAR UUID

Essa variável é o UUID de execução da execução atual.

AWS_WORKFLOW_TAREFA

Essa variável é o nome da tarefa atual.

AWS_WORKFLOW_ID DA TAREFA

Essa variável é o identificador da tarefa atual.

AWS_WORKFLOW_TASK_UUID

Essa variável é o UUID da tarefa atual.

O exemplo a seguir mostra valores típicos para cada variável de ambiente:

```
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
```

Usando Java em imagens de contêiner do Amazon ECR

Se uma tarefa de fluxo de trabalho usa um aplicativo Java, como o GATK, considere os seguintes requisitos de memória para o contêiner:

- Os aplicativos Java usam memória de pilha e memória de pilha. Por padrão, a memória máxima da pilha é uma porcentagem da memória total disponível no contêiner. Esse padrão depende da distribuição específica da JVM e da versão da JVM, portanto, consulte a documentação relevante da sua JVM ou defina explicitamente o máximo de memória do heap usando as opções da linha de comando Java (como `-Xmx`).
- Não defina a memória máxima do heap como 100% da alocação de memória do contêiner, pois a pilha da JVM também requer memória. A memória também é necessária para o coletor de lixo da JVM e para qualquer outro processo do sistema operacional em execução no contêiner.
- Alguns aplicativos Java, como o GATK, podem usar invocações de métodos nativos ou outras otimizações, como arquivos de mapeamento de memória. Essas técnicas exigem alocações de memória executadas “fora da pilha”, que não são controladas pelo parâmetro máximo de pilha da JVM.

  Se você sabe (ou suspeita) que seu aplicativo Java aloca memória fora da pilha, certifique-se de que sua alocação de memória de tarefas inclua os requisitos de memória fora da pilha.

  Se essas alocações fora do heap fizerem com que o contêiner fique sem memória, você normalmente não verá um OutOfMemory erro de Java, porque a JVM não controla essa memória.

Adicionar entradas de tarefas a uma imagem de contêiner do Amazon ECR

Adicione todos os executáveis, bibliotecas e scripts necessários para executar uma tarefa de fluxo de trabalho na imagem do Amazon ECR usada para executar a tarefa.

É uma prática recomendada evitar o uso de scripts, binários e bibliotecas externos à imagem do contêiner de tarefas. Isso é especialmente importante ao usar `nf-core` fluxos de trabalho que

usam um `bin` diretório como parte do pacote de fluxo de trabalho. Embora esse diretório esteja disponível para a tarefa do fluxo de trabalho, ele é montado como um diretório somente para leitura. Os recursos necessários nesse diretório devem ser copiados na imagem da tarefa e disponibilizados em tempo de execução ou ao criar a imagem do contêiner usada para a tarefa.

Consulte HealthOmics cotas de tamanho fixo do fluxo de trabalho o tamanho máximo da imagem do contêiner HealthOmics compatível.

HealthOmics Arquivos README do fluxo de trabalho

Você pode carregar um arquivo README.md contendo instruções, diagramas e informações essenciais para seu fluxo de trabalho. Cada versão do fluxo de trabalho oferece suporte a um arquivo README, que você pode atualizar a qualquer momento.

Os requisitos do README incluem:

- O arquivo README deve estar no formato markdown (.md)
- Tamanho máximo do arquivo: 500 KiB

Tópicos

Use um README existente

READMEs exportados dos repositórios Git contêm links relativos que normalmente não funcionam fora do repositório. HealthOmics A integração com o Git os converte automaticamente em links absolutos para uma renderização adequada no console, eliminando a necessidade de atualizações manuais de URL.

Para READMEs importados do Amazon S3 ou de unidades locais, as imagens e os links devem ser públicos URLs ou ter seus caminhos relativos atualizados para uma renderização adequada.

> **Note**
>
> As imagens devem ser hospedadas publicamente para serem exibidas no HealthOmics console. Imagens armazenadas em repositórios GitHub Enterprise Server ou GitLab Self-Managed repositórios não podem ser renderizadas.

Condições de renderização

O HealthOmics console interpola imagens e links acessíveis ao público usando caminhos absolutos. Para renderizar URLs a partir de repositórios privados, o usuário deve ter acesso ao repositório. GitLab Self-ManagedOs repositórios for GitHub Enterprise Server or, que usam domínios personalizados, HealthOmics não conseguem resolver links relativos nem renderizar imagens armazenadas nesses repositórios privados.

A tabela a seguir mostra os elementos de marcação que são compatíveis com a visualização README do AWS console.

Elemento	AWS console
Alertas	Sim, mas sem ícones
Distintivos	Sim
Formatação básica de texto	Sim
Blocos de código	Sim, mas não tem a funcionalidade de realce de sintaxe e botão de cópia
Seções dobráveis	Sim
Cabeçalhos	Sim
Formatos de imagem	Sim
Imagem (clicável)	Sim
Quebras de linha	Sim
Diagrama de sereia	Só pode abrir o gráfico, mover a posição do gráfico e copiar o código
Cotações	Sim
Subscrito e sobrescrito	Sim
Tabelas	Sim, mas não suporta alinhamento de texto

Elemento	AWS console
Alinhamento de texto	Sim

Usando imagem e link URLs

Dependendo do seu provedor de origem, estruture sua base URLs para páginas e imagens nos seguintes formatos.

- `{username}`: o nome de usuário em que o repositório está hospedado.
- `{repo}`: O nome do repositório.
- `{ref}`: a referência da fonte (ramificação, tag e ID do commit).
- `{path}`: o caminho do arquivo para a página ou imagem no repositório.

Provedor de origem	URL da página	URL da imagem
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}`

GitHub,GitLab, e Bitbucket suportam páginas e imagens URLs vinculadas a um repositório público. A tabela a seguir mostra o suporte de cada provedor de origem para renderizar imagens e links URLs para repositórios privados.

Suporte a repositórios privados		
Provedor de origem	URL da página	URL da imagem
GitHub	Somente com acesso ao repositório	Não
GitLab	Somente com acesso ao repositório	Não
Bitbucket	Somente com acesso ao repositório	Não

Solicitando licenças Sentieon para fluxos de trabalho privados

Se seu fluxo de trabalho privado usa o software Sentieon, você precisa de uma licença Senieon. Siga estas etapas para solicitar e configurar uma licença para o software Sentieon:

- Solicite uma licença Sentieon
  - Envie um e-mail para o grupo de suporte da Sentieon (support@sentieon.com) para solicitar uma licença de software.
    - Forneça seu ID de usuário AWS canônico no e-mail.
    - Encontre sua ID de usuário da AWS Canonical seguindo estas instruções.
- Atualize sua função HealthOmics de serviço para conceder acesso ao proxy do servidor de licenciamento Sentieon e ao bucket do Sentieon Omics em sua região. O exemplo a seguir concede acesso em`us-east-1`. Se necessário, substitua esse texto pela sua região.

  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/*"
            ]
        }
    ]
}
```

- Gere um caso de AWS suporte para obter acesso ao proxy do servidor de licenças Sentieon.
  - Para criar um caso de suporte, navegue até support.console.aws.amazon.com.
  - Forneça sua região Conta da AWS e sua região no caso de suporte. Sua conta é adicionada à lista de permissões do proxy do servidor de licenciamento.
- Crie seu fluxo de trabalho privado usando o contêiner Sentieon e o script de licença Sentieon.
  - Para obter instruções adicionais sobre o uso das ferramentas Sentieon em fluxos de trabalho privados, consulte Sentieon-Amazon-Omics em. GitHub
- A versão 202112.07 e superior do software Sentieon oferece suporte ao proxy do HealthOmics servidor de licenciamento. Para usar as versões do software Sentieon anteriores a 202112.07, entre em contato com o suporte da Sentieon.

Impressoras de fluxo de trabalho em HealthOmics

Depois de criar um fluxo de trabalho, recomendamos que você execute um linter no fluxo de trabalho antes de iniciar a primeira execução. O linter detecta erros que podem fazer com que a execução falhe.

Para a WDL, executa HealthOmics automaticamente um linter quando você cria o fluxo de trabalho. A saída do linter está disponível no `statusMessage` campo da get-workflow resposta. Use o seguinte comando da CLI para recuperar a saída de status (use a ID do fluxo de trabalho do fluxo de trabalho da WDL que você criou):

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

HealthOmics fornece ponteiros que você pode executar na definição do fluxo de trabalho antes de criar o fluxo de trabalho. Execute esses linters nos pipelines existentes para os quais você está migrando. HealthOmics

- WDL— Uma imagem pública do Amazon ECR para executar um linter WDL.
- Nextflow— Uma imagem pública do Amazon ECR para executar as regras do Linter para o Nextflow. Você pode acessar o código-fonte desse linter em. GitHub
- CWL— não disponível

HealthOmics operações de fluxo de trabalho

Para criar um fluxo de trabalho privado, você precisa:

- Workflow definition file:Um arquivo de definição de fluxo de trabalho escrito em WDLNextflow, ouCWL. A definição do fluxo de trabalho especifica as entradas e saídas para execuções que usam o fluxo de trabalho. Também inclui especificações para as execuções e tarefas de execução do seu fluxo de trabalho, incluindo requisitos de computação e memória. O arquivo de definição do fluxo de trabalho deve estar no `.zip` formato. Para obter mais informações, consulte Arquivos de definição de fluxo de trabalho em HealthOmics.
  - Você pode usar o Amazon Q CLI para criar e validar seus arquivos de definição de fluxo de trabalho em WDL, Nextflow e CWL. Para obter mais informações, consulte Exemplos de solicitações para a Amazon Q CLI e HealthOmics o tutorial de IA generativa da Agentic sobre. GitHub
- (Optional) Parameter template file:Um arquivo de modelo de parâmetro escrito emJSON. Crie o arquivo para definir os parâmetros de execução ou HealthOmics gere o modelo de parâmetro para você. Para obter mais informações, consulte Arquivos de modelo de parâmetros para HealthOmics fluxos de trabalho.
- Amazon ECR container images:Crie repositórios privados do Amazon ECR para cada contêiner usado no fluxo de trabalho. Crie imagens de contêiner para o fluxo de trabalho e armazene-as em um repositório privado ou sincronize o conteúdo de um registro upstream compatível com seu repositório privado ECR.
- (Optional) Sentieon licenses:Solicite uma Sentieon licença para usar o Sentieon software em fluxos de trabalho privados.

Para arquivos de definição de fluxo de trabalho maiores que 4 MiB (zipados), escolha uma dessas opções durante a criação do fluxo de trabalho:

- Faça o upload para uma pasta do Amazon Simple Storage Service e especifique a localização.
- Faça o upload para um repositório externo (tamanho máximo de 1 GiB) e especifique os detalhes do repositório.

Depois de criar um fluxo de trabalho, você pode atualizar as seguintes informações do fluxo de trabalho com a `UpdateWorkflow` operação:

- Name (Nome)
- Description
- Tipo de armazenamento padrão
- Capacidade de armazenamento padrão (com ID do fluxo de trabalho)
- Arquivo README.md

Para alterar outras informações no fluxo de trabalho, crie um novo fluxo de trabalho ou uma nova versão do fluxo de trabalho.

Use o controle de versão do fluxo de trabalho para organizar e estruturar seus fluxos de trabalho. As versões também ajudam você a gerenciar a introdução de atualizações iterativas do fluxo de trabalho. Para obter mais informações sobre versões, consulte Crie uma versão do fluxo de trabalho.

Tópicos

- Crie um fluxo de trabalho privado
- Atualizar um fluxo de trabalho privado
- Excluir um fluxo de trabalho privado
- Verifique o status do fluxo de trabalho
- Referenciando arquivos de genoma a partir de uma definição de fluxo de trabalho

Crie um fluxo de trabalho privado

Crie um fluxo de trabalho usando o HealthOmics console, os comandos da AWS CLI ou um dos. AWS SDKs

> **Note**
>
> Não inclua nenhuma informação de identificação pessoal (PII) nos nomes dos fluxos de trabalho. Esses nomes são visíveis nos CloudWatch registros.

Ao criar um fluxo de trabalho, HealthOmics atribui um identificador exclusivo universal (UUID) ao fluxo de trabalho. O UUID do fluxo de trabalho é um identificador global exclusivo (guid) exclusivo em todos os fluxos de trabalho e versões do fluxo de trabalho. Para fins de proveniência de dados, recomendamos que você use o UUID do fluxo de trabalho para identificar fluxos de trabalho de forma exclusiva.

Se suas tarefas de fluxo de trabalho usarem ferramentas externas (executáveis, bibliotecas ou scripts), você cria essas ferramentas em uma imagem de contêiner. Você tem as seguintes opções para hospedar a imagem do contêiner:

- Hospede a imagem do contêiner no registro privado do ECR. Pré-requisitos para essa opção:
  - Crie um repositório privado ECR ou escolha um repositório existente.
  - Configure a política de recursos do ECR conforme descrito emPermissões do Amazon ECR.
  - Faça upload da imagem do contêiner para o repositório privado.
- Sincronize a imagem do contêiner com o conteúdo de um registro de terceiros compatível. Pré-requisitos para essa opção:
  - No registro privado do ECR, configure uma regra de cache pull through para cada registro upstream. Para obter mais informações, consulte Mapeamentos de imagens.
  - Configure a política de recursos do ECR conforme descrito emPermissões do Amazon ECR.
  - Crie modelos de criação de repositórios. O modelo define as configurações para quando o Amazon ECR cria o repositório privado para um registro upstream.
  - Crie mapeamentos de prefixo para remapear referências de imagem de contêiner na definição do fluxo de trabalho para namespaces de cache do ECR.

Ao criar um fluxo de trabalho, você fornece uma definição de fluxo de trabalho que contém informações sobre o fluxo de trabalho, as execuções e as tarefas. HealthOmics pode recuperar a definição do fluxo de trabalho como um arquivo.zip armazenado localmente ou em um bucket do Amazon S3, ou de um repositório compatível baseado em Git.

Tópicos

- Criando um fluxo de trabalho usando o console
- Criação de um fluxo de trabalho usando a CLI
- Criação de um fluxo de trabalho usando um SDK

Criando um fluxo de trabalho usando o console

Etapas para criar um fluxo de trabalho

1. Abra o console de HealthOmics .
2. Se necessário, abra o painel de navegação esquerdo (≡). Escolha fluxos de trabalho privados.
3. Na página Fluxos de trabalho privados, escolha Criar fluxo de trabalho.
4. Na página Definir fluxo de trabalho, forneça as seguintes informações:
    1. Nome do fluxo de trabalho: um nome distinto para esse fluxo de trabalho. Recomendamos definir nomes de fluxo de trabalho para organizar suas execuções no AWS HealthOmics console e nos CloudWatch registros.
    2. Descrição (opcional): uma descrição desse fluxo de trabalho.
5. No painel Definição do fluxo de trabalho, forneça as seguintes informações:
    1. Idioma do fluxo de trabalho (opcional): selecione o idioma de especificação do fluxo de trabalho. Caso contrário, HealthOmics determina o idioma a partir da definição do fluxo de trabalho.
    2. Para a fonte de definição do fluxo de trabalho, escolha importar a pasta de definição de um repositório baseado em Git, de um local do Amazon S3 ou de uma unidade local.
        a. Para importar de um serviço de repositório:

> **Note**
>
> HealthOmics suporta repositórios públicos e privados paraGitHub,,GitLab, BitbucketGitHub self-managed,GitLab self-managed.

            i. Escolha uma Conexão para conectar seus AWS recursos ao repositório externo. Para criar uma conexão, consulteConecte-se com repositórios de código externos.

> **Note**
>
> Os clientes da TLV região precisam criar uma conexão na região IAD (us-east-1) para criar um fluxo de trabalho.

ii. Em ID completa do repositório, insira sua ID do repositório como nome de usuário/nome do repositório. Verifique se você tem acesso aos arquivos neste repositório.

iii. Em Referência da fonte (opcional), insira uma referência da fonte do repositório (ramificação, tag ou ID do commit). HealthOmics usa a ramificação padrão se nenhuma referência de origem for especificada.

iv. Em Excluir padrões de arquivo, insira os padrões de arquivo para excluir pastas, arquivos ou extensões específicas. Isso ajuda a gerenciar o tamanho dos dados ao importar arquivos do repositório. Há no máximo 50 padrões, e os padrões devem seguir a sintaxe do padrão global. Por exemplo:

A. `tests/`

B. `*.jpeg`

C. `large_data.zip`

b. Para Selecionar pasta de definição do S3:

i. Insira a localização do Amazon S3 que contém a pasta de definição de fluxo de trabalho compactada. O bucket do Amazon S3 deve estar na mesma região do fluxo de trabalho.

ii. Se sua conta não for proprietária do bucket do Amazon S3, insira o ID da AWS conta do proprietário do bucket no ID da conta do proprietário do bucket do S3. Essas informações são necessárias para que HealthOmics possamos verificar a propriedade do bucket.

c. Para Selecionar pasta de definição de uma fonte local:

i. Insira a localização da unidade local da pasta de definição de fluxo de trabalho compactada.

3. Caminho do arquivo de definição do fluxo de trabalho principal (opcional): insira o caminho do arquivo da pasta de definição do fluxo de trabalho compactado ou do repositório para o `main` arquivo. Esse parâmetro não é necessário se houver somente um arquivo na pasta de definição do fluxo de trabalho ou se o arquivo principal tiver o nome “principal”.

6. No painel Arquivo README (opcional), selecione a Fonte do arquivo README e forneça as seguintes informações:

- Em Importar de um serviço de repositório, em Caminho do arquivo README, insira o caminho para o arquivo README dentro do repositório.
- Em Selecionar arquivo do S3, em Arquivo README no S3, insira o URI do Amazon S3 para o arquivo README.
- Em Selecionar arquivo de uma fonte local: em README - opcional, escolha Escolher arquivo para selecionar o arquivo markdown (.md) a ser carregado.

7. No painel Configuração de armazenamento de execução padrão, forneça o tipo de armazenamento de execução padrão e a capacidade para execuções que usam esse fluxo de trabalho:

    1. Tipo de execução de armazenamento: escolha se deseja usar armazenamento estático ou dinâmico como padrão para o armazenamento de execução temporária. O padrão é armazenamento estático.
    2. Capacidade de armazenamento de execução (opcional): para o tipo de armazenamento de execução estática, você pode inserir a quantidade padrão de armazenamento de execução necessária para esse fluxo de trabalho. O valor padrão para esse parâmetro é 1200 GiB. Você pode substituir esses valores padrão ao iniciar uma execução.

8. Tags (opcional): você pode associar até 50 tags a esse fluxo de trabalho.
9. Escolha Próximo.
10. Na página Adicionar parâmetros de fluxo de trabalho (opcional), selecione a fonte do parâmetro:

    1. Para Analisar do arquivo de definição do fluxo de trabalho, HealthOmics analisará automaticamente os parâmetros do fluxo de trabalho do arquivo de definição do fluxo de trabalho.
    2. Para Fornecer modelo de parâmetro do repositório Git, use o caminho para o arquivo de modelo de parâmetro do seu repositório.
    3. Em Selecionar arquivo JSON da fonte local, faça upload de um JSON arquivo de uma fonte local que especifique os parâmetros.
    4. Em Inserir manualmente os parâmetros do fluxo de trabalho, insira manualmente os nomes e as descrições dos parâmetros.

11. No painel de visualização de parâmetros, você pode revisar ou alterar os parâmetros dessa versão do fluxo de trabalho. Se você restaurar o JSON arquivo, perderá todas as alterações locais feitas.
12. Escolha Próximo.

13. Na página de remapeamento de URI de contêiner, no painel Regras de mapeamento, você pode definir regras de mapeamento de URI para seu fluxo de trabalho.

    Em Fonte do arquivo de mapeamento, selecione uma das seguintes opções:

    - Nenhuma — Não são necessárias regras de mapeamento.
    - Selecione o arquivo JSON do S3 — Especifique a localização do arquivo de mapeamento no S3.
    - Selecione o arquivo JSON de uma fonte local — Especifique a localização do arquivo de mapeamento em seu dispositivo local.
    - Inserir mapeamentos manualmente — insira os mapeamentos do registro e os mapeamentos de imagem no painel Mapeamentos.

14. O console exibe o painel Mapeamentos. Se você escolher um arquivo de origem de mapeamento, o console exibirá os valores do arquivo.

    a. Em Mapeamentos do registro, você pode editar os mapeamentos ou adicionar mapeamentos (máximo de 20 mapeamentos do registro).

       Cada mapeamento do registro contém os seguintes campos:

       - URL do registro upstream — O URI do registro upstream.
       - Prefixo do repositório ECR — O prefixo do repositório a ser usado no repositório privado do Amazon ECR.
       - (Opcional) Prefixo do repositório upstream — O prefixo do repositório no registro upstream.
       - (Opcional) ID da conta ECR — ID da conta que possui a imagem do contêiner upstream.

    b. Em Mapeamentos de imagem, você pode editar os mapeamentos de imagem ou adicionar mapeamentos (máximo de 100 mapeamentos de imagem).

       Cada mapeamento de imagem contém os seguintes campos:

       - Imagem de origem — especifica o URI da imagem de origem no registro upstream.
       - Imagem de destino — Especifica o URI da imagem correspondente no registro privado do Amazon ECR.

15. Escolha Próximo.

16. Revise a configuração do fluxo de trabalho e escolha Criar fluxo de trabalho.

Criação de um fluxo de trabalho usando a CLI

Se seus arquivos de fluxo de trabalho e o arquivo de modelo de parâmetros estiverem em sua máquina local, você poderá criar um fluxo de trabalho usando o seguinte comando da CLI.

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

A `create-workflow` operação retorna a seguinte resposta:

```
{
  "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"
}
```

Parâmetros opcionais a serem usados ao criar um fluxo de trabalho

Você pode especificar qualquer um dos parâmetros opcionais ao criar um fluxo de trabalho. Para obter detalhes de sintaxe, consulte CreateWorkflowa Referência da HealthOmics API da AWS.

Tópicos

- Especifique a definição do fluxo de trabalho (localização do Amazon S3)
- Use a definição de fluxo de trabalho de um repositório baseado em Git
- Especificar um arquivo Readme
- Especifique o arquivo main de definição
- Especifique o tipo de armazenamento de execução
- Especifique a configuração da GPU
- Configurar parâmetros de mapeamento de cache pull through

Especifique a definição do fluxo de trabalho (localização do Amazon S3)

Se seu arquivo de definição de fluxo de trabalho estiver localizado em uma pasta do Amazon S3, especifique a localização usando o `definition-uri` parâmetro, conforme mostrado no exemplo a seguir. Se sua conta não for proprietária do bucket Amazon S3, forneça o ID do Conta da AWS proprietário.

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

Use a definição de fluxo de trabalho de um repositório baseado em Git

Para usar a definição de fluxo de trabalho de um repositório compatível baseado em Git, use o `definition-repository` parâmetro em sua solicitação. Não forneça nenhum outro `definition` parâmetro, pois uma solicitação falhará se incluir mais de uma fonte de entrada.

O `definition-respository` parâmetro contém os seguintes campos:

- connectionArn— ARN da conexão de código que conecta seus recursos da AWS ao repositório externo.
- fullRepositoryId— Insira o ID do repositório como`owner-name/repo-name`. Verifique se você tem acesso aos arquivos neste repositório.
- sourceReference(Opcional) — Insira um tipo de referência do repositório (BRANCH, TAG ou COMMIT) e um valor.

  HealthOmics usa a confirmação mais recente na ramificação padrão se você não especificar uma referência de origem.
- excludeFilePatterns(Opcional) — Insira os padrões de arquivo para excluir pastas, arquivos ou extensões específicas. Isso ajuda a gerenciar o tamanho dos dados ao importar arquivos do repositório. Forneça no máximo 50 padrões. Os padrões devem seguir a sintaxe do padrão global. Por exemplo:
  - `tests/`
  - `*.jpeg`
  - `large_data.zip`

Ao especificar a definição do fluxo de trabalho em um repositório baseado em Git, use `parameter-template-path` para especificar o arquivo de modelo de parâmetros. Se você não fornecer esse parâmetro, HealthOmics cria o fluxo de trabalho sem um modelo de parâmetro.

O exemplo a seguir mostra os parâmetros relacionados ao conteúdo de um repositório privado baseado em 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 mais exemplos, consulte a postagem do blog Como criar HealthOmics fluxos de trabalho da AWS a partir de conteúdo no Git.

Especificar um arquivo Readme

Você pode especificar a localização do arquivo README usando um dos seguintes parâmetros:

- readme-markdown— Entrada de string ou um arquivo em sua máquina local.
- readme-uri— O URI de um arquivo armazenado no S3.
- readme-path — O caminho para o arquivo README no repositório.

Use readme-path somente em conjunto com definition-respository. Se você não especificar nenhum parâmetro README, HealthOmics importe o arquivo README.md de nível raiz no repositório (se ele existir).

Os exemplos a seguir mostram como especificar a localização do arquivo README usando readme-path e 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 obter mais informações, consulte HealthOmics Arquivos README do fluxo de trabalho.

Especifique o arquivo main de definição

Se você estiver incluindo vários arquivos de definição de fluxo de trabalho, use o `main` parâmetro para especificar o arquivo de definição principal para seu fluxo de trabalho.

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

Especifique o tipo de armazenamento de execução

Você pode especificar o tipo de armazenamento de execução padrão (DINÂMICO ou ESTÁTICO) e a capacidade de armazenamento de execução (necessária para armazenamento estático). Para obter mais informações sobre como executar tipos de armazenamento, consulteExecute tipos de armazenamento em HealthOmics fluxos de trabalho.

```
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 a configuração da GPU

Use o parâmetro accelerators para criar um fluxo de trabalho executado em uma instância de computação acelerada. O exemplo a seguir mostra como usar o `accelerators` parâmetro. Você especifica a configuração da GPU na definição do fluxo de trabalho. Consulte Instâncias de computação acelerada.

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

Configurar parâmetros de mapeamento de cache pull through

Se você estiver usando o recurso de mapeamento de cache pull through do Amazon ECR, poderá substituir os mapeamentos padrão. Para obter mais informações sobre os parâmetros de configuração do contêiner, consulteImagens de contêiner para fluxos de trabalho privados.

No exemplo a seguir, o arquivo `mappings.json` contém esse conteúdo:

```
{
    "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 os parâmetros de mapeamento no comando create-workflow:

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

Você também pode especificar a localização do arquivo de parâmetros de mapeamento no S3:

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

Criação de um fluxo de trabalho usando um SDK

Você pode criar um fluxo de trabalho usando um dos SDKs. O exemplo a seguir mostra como criar um fluxo de trabalho usando o SDK do 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={ ... }
)
```

Atualizar um fluxo de trabalho privado

Você pode atualizar um fluxo de trabalho usando o HealthOmics console, os comandos da AWS CLI ou um dos. AWS SDKs

> **Note**
>
> Não inclua nenhuma informação de identificação pessoal (PII) nos nomes dos fluxos de trabalho. Esses nomes são visíveis nos CloudWatch registros.

Tópicos

Atualizando um fluxo de trabalho usando o console

Etapas para atualizar um fluxo de trabalho

1. Abra o console de HealthOmics .
2. Se necessário, abra o painel de navegação esquerdo (≡). Escolha fluxos de trabalho privados.
3. Na página Fluxos de trabalho privados, escolha o fluxo de trabalho a ser atualizado.
4. Na página Fluxo de trabalho:
   - Se o fluxo de trabalho tiver versões, certifique-se de selecionar a versão padrão.
   - Escolha Editar selecionado na lista Ações.
5. Na página Editar fluxo de trabalho, você pode alterar qualquer um dos seguintes valores:
   - Nome do fluxo de trabalho.
   - Descrição do fluxo de trabalho.
   - O tipo padrão de armazenamento Executar para o fluxo de trabalho.
   - A capacidade de armazenamento de execução padrão (se o tipo de armazenamento de execução for armazenamento estático). Para obter mais informações sobre a configuração padrão do armazenamento de execução, consulteCriando um fluxo de trabalho usando o console.
6. Escolha Salvar alterações para aplicar as alterações.

Atualizando um fluxo de trabalho usando a CLI

Conforme mostrado no exemplo a seguir, você pode atualizar o nome e a descrição do fluxo de trabalho. Você também pode alterar o tipo de armazenamento de execução padrão (ESTÁTICO ou DINÂMICO) e a capacidade de armazenamento de execução (para o tipo de armazenamento estático). Para obter mais informações sobre como executar tipos de armazenamento, consulteExecute tipos de armazenamento em HealthOmics fluxos de trabalho.

```
aws omics update-workflow    \
  --id 1234567    \
  --name my_workflow      \
  --description "updated workflow"    \
  --storage-type 'STATIC'    \
  --storage-capacity 1200
```

Você não recebe uma resposta à `update-workflow` solicitação.

Atualização de um fluxo de trabalho usando um SDK

Você pode atualizar um fluxo de trabalho usando um dos SDKs.

O exemplo a seguir mostra como atualizar um fluxo de trabalho usando o SDK do Python.

```
import boto3

omics = boto3.client('omics')

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

Excluir um fluxo de trabalho privado

Quando você não precisar mais de um fluxo de trabalho, poderá excluí-lo usando o HealthOmics console, os comandos da AWS CLI ou um dos. AWS SDKs Você pode excluir um fluxo de trabalho que atenda aos seguintes critérios:

- Seu status é ATIVO ou FALHA.
- Não tem compartilhamentos ativos.

- Você excluiu todas as versões do fluxo de trabalho.

A exclusão de um fluxo de trabalho não afeta nenhuma execução em andamento que esteja usando o fluxo de trabalho.

Tópicos

Excluindo um fluxo de trabalho usando o console

Para excluir um fluxo de trabalho

1. Abra o console de HealthOmics .
2. Se necessário, abra o painel de navegação esquerdo (≡). Escolha fluxos de trabalho privados.
3. Na página Fluxos de trabalho privados, escolha o fluxo de trabalho a ser excluído.
4. Na página Fluxo de trabalho, escolha Excluir selecionado na lista Ações.
5. No modal Excluir fluxo de trabalho, insira “confirmar” para confirmar a exclusão.
6. Escolha Excluir.

Excluindo um fluxo de trabalho usando a CLI

O exemplo a seguir mostra como você pode usar o AWS CLI comando para excluir um fluxo de trabalho. Para executar o exemplo, *`workflow id`* substitua o pelo ID do fluxo de trabalho que você deseja excluir.

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

HealthOmics não envia uma resposta à `delete-workflow` solicitação.

Excluindo um fluxo de trabalho usando um SDK

Você pode excluir um fluxo de trabalho usando um dos SDKs.

O exemplo a seguir mostra como excluir um fluxo de trabalho usando o SDK do Python.

```
import boto3

omics = boto3.client('omics')

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

Verifique o status do fluxo de trabalho

Depois de criar seu fluxo de trabalho, você pode verificar o status e visualizar outros detalhes do fluxo de trabalho usando get-workflow, conforme mostrado.

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

A resposta inclui detalhes do fluxo de trabalho, incluindo o status, conforme mostrado.

```
{
    "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"
  }
```

Você pode iniciar uma execução usando esse fluxo de trabalho após a transição do status para`ACTIVE`.

Referenciando arquivos de genoma a partir de uma definição de fluxo de trabalho

Um objeto de armazenamento de HealthOmics referência pode ser referenciado com um URI como o seguinte. Use o seu próprio *`account ID`* e *`reference ID`* onde indicado. *`reference store ID`*

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

Alguns fluxos de trabalho exigirão INDEX arquivos SOURCE e para o genoma de referência. O URI anterior é o formato abreviado padrão e usará como padrão o arquivo SOURCE. Para especificar qualquer um dos arquivos, você pode usar o formulário URI longo, da seguinte forma.

```
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
```

Usar um conjunto de leitura de sequência teria um padrão semelhante, conforme mostrado.

```
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
```

Alguns conjuntos de leitura, como os baseados no FASTQ, podem conter leituras emparelhadas. Nos exemplos a seguir, eles são chamados de SOURCE1 SOURCE2 e. Formatos como BAM e CRAM terão apenas um SOURCE1 arquivo. Alguns conjuntos de leitura conterão arquivos INDEX, como `crai` arquivos `bai` ou. O URI anterior é o formato abreviado padrão e usará o SOURCE1 arquivo como padrão. Para especificar o arquivo ou índice exato, você pode usar o formato URI longo, da seguinte forma.

```
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
```

Veja a seguir um exemplo de um arquivo JSON de entrada que usa dois 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>"
}
```

Faça referência ao arquivo JSON de entrada no AWS CLI adicionando `--inputs file://<input_file.json>` à sua solicitação de início de execução.

Controle de versão do fluxo de trabalho em HealthOmics

Se precisar fazer alterações em um fluxo de trabalho, você pode criar um novo fluxo de trabalho ou uma nova versão do fluxo de trabalho. As versões são imutáveis, exceto pelas alterações de configuração permitidas que não afetam a lógica de execução.

As versões do fluxo de trabalho oferecem os seguintes benefícios:

- As versões formam um grupo lógico de fluxos de trabalho relacionados. Você pode adicionar um nome definido pelo usuário a cada versão do fluxo de trabalho para gerenciá-las com mais facilidade (especialmente para um fluxo de trabalho com um grande número de versões).
- Você pode executar várias versões de um fluxo de trabalho ao mesmo tempo.
- Todas as versões de um fluxo de trabalho compartilham a mesma ID de fluxo de trabalho e ARN base, o que pode simplificar o gerenciamento do pipeline após a modificação de um fluxo de trabalho.
- As versões de fluxo de trabalho fornecem o mesmo nível de proveniência de dados que os fluxos de trabalho. As versões são imutáveis e HealthOmics criam um ARN exclusivo para cada versão do fluxo de trabalho. O ARN da versão inclui o ID do fluxo de trabalho e o nome da versão, conforme mostrado no exemplo a seguir:

  `arn:aws:omics:us-west-2:123456789012:workflow/1234567/version/myUniqueVersionName`
- Se você possui um fluxo de trabalho compartilhado, pode atualizar o fluxo de trabalho sem interromper os assinantes (que podem continuar usando a versão anterior). Os assinantes podem acessar todas as versões do fluxo de trabalho. Se você criar uma nova versão, não precisará compartilhar novamente o fluxo de trabalho.
- Ao iniciar a execução de um fluxo de trabalho, você pode especificar a versão do fluxo de trabalho.
  - Os usuários podem optar por permanecer em uma versão estável para execuções de produção e experimentar a versão mais recente para um teste.

- Os usuários podem reverter para a versão anterior de um fluxo de trabalho se encontrarem problemas com a nova versão.
- Os assinantes de um fluxo de trabalho compartilhado podem escolher qual versão usar.

Tópicos

Versão padrão do fluxo de trabalho

Depois de criar uma ou mais versões de um fluxo de trabalho, HealthOmics trata o fluxo de trabalho original como a versão padrão. Ao iniciar uma execução, você pode, opcionalmente, especificar uma versão do fluxo de trabalho para a execução. Se você não especificar uma versão ao iniciar uma execução, HealthOmics usa a versão padrão.

No console, HealthOmics indica o fluxo de trabalho original com um rótulo de versão padrão. O console usa esse rótulo somente depois que você cria uma ou mais versões do fluxo de trabalho. O fluxo de trabalho original sempre permanece a versão padrão. Você não pode atribuir nenhuma outra versão como padrão.

Você não pode excluir a versão padrão de um fluxo de trabalho se houver outras versões associadas ao fluxo de trabalho. Para obter mais informações, consulte Excluir um fluxo de trabalho privado.

Crie uma versão do fluxo de trabalho

Ao criar uma nova versão de um fluxo de trabalho, você precisa especificar os valores de configuração para a nova versão. Ele não herda nenhum valor de configuração do fluxo de trabalho.

Ao criar a versão, forneça um nome de versão exclusivo para esse fluxo de trabalho. Você não pode alterar o nome depois de HealthOmics criar a versão.

O nome da versão deve começar com uma letra ou número e pode incluir letras maiúsculas e minúsculas, números, hífens, pontos e sublinhados. O tamanho máximo é de 64 caracteres. Por exemplo, você pode usar um esquema de nomenclatura simples, como versão1, versão2, versão3.

Você também pode combinar suas versões de fluxo de trabalho com suas próprias convenções internas de controle de versão, como 2.7.0, 2.7.1, 2.7.2.

Opcionalmente, use o campo de descrição da versão para adicionar notas sobre essa versão. Por exemplo: Fix for syntax error in workflow definition.

> **Note**
>
> Não inclua nenhuma informação de identificação pessoal (PII) no nome da versão. Os nomes das versões aparecem no ARN da versão do fluxo de trabalho.

HealthOmics atribui um ARN exclusivo à versão do fluxo de trabalho. O ARN é exclusivo com base na combinação de ID do fluxo de trabalho e nome da versão.

> **Warning**
>
> Depois de excluir uma versão do fluxo de trabalho, HealthOmics você pode reutilizar o nome da versão para uma versão diferente do fluxo de trabalho. A melhor prática é não reutilizar nomes de versões. Se você reutilizar um nome, o fluxo de trabalho e cada versão terão um UUID exclusivo que você pode usar como proveniência.

Tópicos

Crie uma versão do fluxo de trabalho usando o console

Etapas para criar uma versão do fluxo de trabalho

1. Abra o console de HealthOmics .
2. Se necessário, abra o painel de navegação esquerdo (≡). Escolha fluxos de trabalho privados.
3. Na página Fluxos de trabalho privados, escolha o fluxo de trabalho para a nova versão.

4. Na página de detalhes do fluxo de trabalho, escolha Criar nova versão.
5. Na página Criar versão, forneça as seguintes informações:

    1. Nome da versão: insira um nome para a versão do fluxo de trabalho que seja exclusivo em todo o fluxo de trabalho.
    2. Descrição da versão (opcional): você pode usar o campo de descrição para adicionar notas sobre essa versão.
6. No painel Definição do fluxo de trabalho, forneça as seguintes informações:

    1. Idioma do fluxo de trabalho (opcional): selecione o idioma de especificação para a versão do fluxo de trabalho. Caso contrário, HealthOmics determina o idioma a partir da definição do fluxo de trabalho.
    2. Para a fonte de definição do fluxo de trabalho, escolha importar a pasta de definição de um repositório baseado em Git, de um local do Amazon S3 ou de uma unidade local.

        a. Para importar de um serviço de repositório:

        > **Note**
        >
        > HealthOmics suporta repositórios públicos e privados paraGitHub,,GitLab, BitbucketGitHub self-managed,GitLab self-managed.

        i. Escolha uma Conexão para conectar seus AWS recursos ao repositório externo. Para criar uma conexão, consulteConecte-se com repositórios de código externos.

        > **Note**
        >
        > Os clientes da TLV região precisam criar uma conexão na região IAD (us-east-1) para criar um fluxo de trabalho.

        ii. Em ID completa do repositório, insira sua ID do repositório como nome de usuário/nome do repositório. Verifique se você tem acesso aos arquivos neste repositório.

        iii. Em Referência da fonte (opcional), insira uma referência da fonte do repositório (ramificação, tag ou ID do commit). HealthOmics usa a ramificação padrão se nenhuma referência de origem for especificada.

        iv. Em Excluir padrões de arquivo, insira os padrões de arquivo para excluir pastas, arquivos ou extensões específicas. Isso ajuda a gerenciar o tamanho dos dados ao

importar arquivos do repositório. Há no máximo 50 padrões, e os padrões devem seguir a sintaxe do padrão global. Por exemplo:

A. `tests/`

B. `*.jpeg`

C. `large_data.zip`

b. Para Selecionar pasta de definição do S3:

i. Insira a localização do Amazon S3 que contém a pasta de definição de fluxo de trabalho compactada. O bucket do Amazon S3 deve estar na mesma região do fluxo de trabalho.

ii. Se sua conta não for proprietária do bucket do Amazon S3, insira o ID da AWS conta do proprietário do bucket no ID da conta do proprietário do bucket do S3. Essas informações são necessárias para que HealthOmics possamos verificar a propriedade do bucket.

c. Para Selecionar pasta de definição de uma fonte local:

i. Insira a localização da unidade local da pasta de definição de fluxo de trabalho compactada.

3. Caminho do arquivo de definição do fluxo de trabalho principal (opcional): insira o caminho do arquivo da pasta de definição do fluxo de trabalho compactado ou do repositório para o `main` arquivo. Esse parâmetro não é necessário se houver somente um arquivo na pasta de definição do fluxo de trabalho ou se o arquivo principal tiver o nome “principal”.

7. No painel Arquivo README (opcional), selecione a Fonte do arquivo README e forneça as seguintes informações:

- Em Importar de um serviço de repositório, em Caminho do arquivo README, insira o caminho para o arquivo README dentro do repositório.
- Em Selecionar arquivo do S3, em Arquivo README no S3, insira o URI do Amazon S3 para o arquivo README.
- Em Selecionar arquivo de uma fonte local: em README - opcional, escolha Escolher arquivo para selecionar o arquivo markdown (.md) a ser carregado.

8. No painel Configuração de armazenamento de execução padrão, forneça o tipo de armazenamento de execução padrão e a capacidade para execuções que usam esse fluxo de trabalho:

1. Tipo de execução de armazenamento: escolha se deseja usar armazenamento estático ou dinâmico como padrão para o armazenamento de execução temporária. O padrão é armazenamento estático.

2. Capacidade de armazenamento de execução (opcional): para o tipo de armazenamento de execução estática, você pode inserir a quantidade padrão de armazenamento de execução necessária para esse fluxo de trabalho. O valor padrão para esse parâmetro é 1200 GiB. Você pode substituir esses valores padrão ao iniciar uma execução.

9. Tags (opcional): você pode associar até 50 tags a essa versão do fluxo de trabalho.
10. Escolha Próximo.
11. Na página Adicionar parâmetros de fluxo de trabalho (opcional), selecione a fonte do parâmetro:

    1. Para Analisar do arquivo de definição do fluxo de trabalho, HealthOmics analisará automaticamente os parâmetros do fluxo de trabalho do arquivo de definição do fluxo de trabalho.
    2. Para Fornecer modelo de parâmetro do repositório Git, use o caminho para o arquivo de modelo de parâmetro do seu repositório.
    3. Em Selecionar arquivo JSON da fonte local, faça upload de um JSON arquivo de uma fonte local que especifique os parâmetros.
    4. Em Inserir manualmente os parâmetros do fluxo de trabalho, insira manualmente os nomes e as descrições dos parâmetros.

12. No painel de visualização de parâmetros, você pode revisar ou alterar os parâmetros dessa versão do fluxo de trabalho. Se você restaurar o JSON arquivo, perderá todas as alterações locais feitas.
13. Na página de remapeamento de URI de contêiner, no painel Regras de mapeamento, você pode definir regras de mapeamento de URI para seu fluxo de trabalho.

    Em Fonte do arquivo de mapeamento, selecione uma das seguintes opções:

    - Nenhuma — Não são necessárias regras de mapeamento.
    - Selecione o arquivo JSON do S3 — Especifique a localização do arquivo de mapeamento no S3.
    - Selecione o arquivo JSON de uma fonte local — Especifique a localização do arquivo de mapeamento em seu dispositivo local.
    - Inserir mapeamentos manualmente — insira os mapeamentos do registro e os mapeamentos de imagem no painel Mapeamentos.

14. O console exibe o painel Mapeamentos. Se você escolher um arquivo de origem de mapeamento, o console exibirá os valores do arquivo.

a. Em Mapeamentos do registro, você pode editar os mapeamentos ou adicionar mapeamentos (máximo de 20 mapeamentos do registro).

   Cada mapeamento do registro contém os seguintes campos:

   - URL do registro upstream — O URI do registro upstream.
   - Prefixo do repositório ECR — O prefixo do repositório a ser usado no repositório privado do Amazon ECR.
   - (Opcional) Prefixo do repositório upstream — O prefixo do repositório no registro upstream.
   - (Opcional) ID da conta ECR — ID da conta que possui a imagem do contêiner upstream.

b. Em Mapeamentos de imagem, você pode editar os mapeamentos de imagem ou adicionar mapeamentos (máximo de 100 mapeamentos de imagem).

   Cada mapeamento de imagem contém os seguintes campos:

   - Imagem de origem — especifica o URI da imagem de origem no registro upstream.
   - Imagem de destino — Especifica o URI da imagem correspondente no registro privado do Amazon ECR.

15. Escolha Próximo.
16. Revise a configuração da versão e escolha Criar versão.

Quando a versão é criada, o console retorna à página de detalhes do fluxo de trabalho e exibe a nova versão na tabela Fluxos de trabalho e versões.

Crie uma versão do fluxo de trabalho usando a CLI

Você pode criar uma versão do fluxo de trabalho usando a operação `CreateWorkflowVersion` da API. Para parâmetros opcionais, HealthOmics usa os seguintes padrões:

Parameter	Padrão
Mecanismo	Determinado a partir da definição do fluxo de trabalho
Tipo de armazenamento	STATIC

Parameter	Padrão
Capacidade de armazenamento (para armazenamento estático)	1.200 GiB
Principal	Determinado com base no conteúdo da pasta de definição do fluxo de trabalho. Para obter detalhes, consulte HealthOmics requisitos de definição de fluxo de trabalho.
Aceleradores	nenhuma
Tags	nenhuma

O exemplo de CLI a seguir cria uma versão de fluxo de trabalho com armazenamento estático como armazenamento de execução padrão:

```
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
```

Se seu arquivo de definição de fluxo de trabalho estiver localizado em uma pasta do Amazon S3, insira o local usando o `definition-uri` parâmetro em vez de. `definition-zip` Para obter mais informações, consulte CreateWorkflowVersiona AWS HealthOmics API Reference.

Você recebe a seguinte resposta à `create-workflow-version` solicitação.

```
{
  "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"
}
```

Crie uma versão do fluxo de trabalho usando um SDK

Você pode criar um fluxo de trabalho usando um dos SDKs.

O exemplo a seguir mostra como criar uma versão do fluxo de trabalho usando o SDK do 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={ ... }
)
```

Verificar o status de uma versão do fluxo de trabalho

Depois de criar sua versão do fluxo de trabalho, você pode verificar o status e visualizar outros detalhes do fluxo de trabalho usando get-workflow-version, conforme mostrado.

```
aws omics get-workflow-version
--workflow-id 9876543
--version-name "my_version"
```

A resposta fornece detalhes do seu fluxo de trabalho, incluindo o status, conforme mostrado.

```
{
```

```
  "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"
}
```

Antes de iniciar uma execução com essa versão do fluxo de trabalho, o status deve ser transferido para`ACTIVE`.

Atualizar uma versão do fluxo de trabalho

Você pode atualizar a descrição e a configuração padrão do armazenamento de execução para uma versão privada do fluxo de trabalho. Para alterar qualquer outra informação na versão do fluxo de trabalho, crie uma nova versão.

Tópicos

- Atualizar uma versão do fluxo de trabalho usando o console
- Atualizar uma versão do fluxo de trabalho usando a CLI
- Atualizar uma versão do fluxo de trabalho usando um SDK

Atualizar uma versão do fluxo de trabalho usando o console

Para atualizar uma versão do fluxo de trabalho

1. Abra o console de HealthOmics .
2. Se necessário, abra o painel de navegação esquerdo (≡). Escolha fluxos de trabalho privados.
3. Na página Fluxos de trabalho privados, escolha o fluxo de trabalho.
4. Na página Fluxo de trabalho, escolha a versão do fluxo de trabalho a ser atualizada e escolha Editar selecionada na lista Ações.
   - Se você escolher a versão padrão, o console abrirá a página Editar fluxo de trabalho. Para obter mais informações, consulte Atualizar um fluxo de trabalho privado.
   - Se você escolher uma versão definida pelo usuário, o console abrirá a página Editar versão.
5. Na página Editar versão, forneça as seguintes informações
   - Descrição da versão (opcional) - Uma descrição dessa versão.

6. No painel Configuração de armazenamento de execução padrão, forneça os seguintes valores padrão para execuções que usam essa versão do fluxo de trabalho. Você pode substituir os valores padrão ao iniciar uma execução:
   - Para o tipo de armazenamento Executar, selecione Estático ou Dinâmico.
   - Para armazenamento estático de execução, selecione a quantidade padrão de capacidade de armazenamento de execução para execuções que usam essa versão do fluxo de trabalho. O valor padrão para esse parâmetro é 1200 GiB.
7. Escolha Salvar alterações.

O console retorna à página de detalhes do fluxo de trabalho e exibe um banner de página com a versão atualizada do fluxo de trabalho.

Atualizar uma versão do fluxo de trabalho usando a CLI

Você pode atualizar os parâmetros de uma versão do fluxo de trabalho usando o seguinte comando da CLI. A combinação do ID do fluxo de trabalho e do nome da versão identifica a versão de forma exclusiva.

```
aws omics update-workflow-version
--workflow-id 1234567
--version-name "my_version"
--storage-type 'STATIC'
--storage-capacity 2400
--description "version description"
```

Você não recebe resposta à `update-workflow-version` solicitação.

Atualizar uma versão do fluxo de trabalho usando um SDK

Você pode atualizar uma versão do fluxo de trabalho usando um dos SDKs. O exemplo de SDK para python a seguir mostra como atualizar o tipo de armazenamento e a descrição de uma versão do fluxo de trabalho.

```
import boto3

omics = boto3.client('omics')

response = omics.update_workflow_version(
```

```
    workflowID=1234567,
    versionName='3.0.0',
    storageType='DYNAMIC',
    description='new version description'
)
```

Excluir uma versão do fluxo de trabalho

Você pode excluir uma versão do fluxo de trabalho definida pelo usuário usando o console, a CLI ou um dos. SDKs A exclusão de uma versão do fluxo de trabalho não afeta nenhuma execução em andamento que esteja usando a versão do fluxo de trabalho.

Você não pode excluir Versão padrão do fluxo de trabalho o. Você exclui todas as versões definidas pelo usuário e, em seguida, exclui o fluxo de trabalho.

Tópicos

- Excluir uma versão do fluxo de trabalho usando o console
- Excluir uma versão do fluxo de trabalho usando a CLI
- Excluir uma versão do fluxo de trabalho usando um SDK

Excluir uma versão do fluxo de trabalho usando o console

Para excluir uma versão do fluxo de trabalho

1. Abra o console de HealthOmics .
2. Se necessário, abra o painel de navegação esquerdo (≡). Escolha fluxos de trabalho privados.
3. Na página Fluxos de trabalho privados, escolha o fluxo de trabalho.
4. Na página Fluxo de trabalho, escolha a versão do fluxo de trabalho a ser excluída e escolha Excluir selecionada na lista Ações.
5. No modal Excluir versão do fluxo de trabalho, digite “confirmar” para confirmar a exclusão.
6. Escolha Excluir.

O console exibe um banner de página com a versão excluída do fluxo de trabalho.

Excluir uma versão do fluxo de trabalho usando a CLI

Você pode excluir uma versão do fluxo de trabalho definida pelo usuário usando o seguinte comando da CLI. A combinação do ID do fluxo de trabalho e do nome da versão identifica a versão de forma exclusiva.

```
aws omics delete-workflow-version
--workflow-id 9876543
--version-name "my_version"
```

Você não recebe resposta à `delete-workflow-version` solicitação.

Excluir uma versão do fluxo de trabalho usando um SDK

Você pode excluir um fluxo de trabalho usando um dos SDKs.

O exemplo a seguir mostra como excluir um fluxo de trabalho usando o SDK do Python.

```
import boto3

omics = boto3.client('omics')

response = omics.delete_workflow_version(
   workflowID=1234567,
   versionName='3.0.0'
)
```

Usando HealthOmics execuções

Depois de criar um fluxo de trabalho, você pode iniciar as execuções usando o fluxo de trabalho.

Quando você inicia uma execução, HealthOmics aloca armazenamento de execução temporário para o mecanismo de fluxo de trabalho usar durante a execução. Para garantir o isolamento e a segurança dos dados, HealthOmics provisiona o armazenamento no início de cada execução e o desprovisiona no final da execução.

HealthOmics fornece várias cotas relacionadas às execuções e tarefas do fluxo de trabalho. Os valores padrão são intencionalmente conservadores, para ajudar a evitar custos excessivos inesperados. Você pode solicitar um aumento dessas cotas. Para obter mais informações, consulte HealthOmics cotas de serviço.

Quando você inicia uma execução, HealthOmics atribui uma ID de execução e um uuid de execução à execução. As execuções em uma conta têm uma execução exclusiva IDs. No entanto, HealthOmics reutiliza a execução excluída IDs, portanto, uma execução e uma execução excluída podem ter a mesma ID de execução. Além disso, é raro, mas possível, que um fluxo de trabalho compartilhado tenha o mesmo ID de execução de uma execução em sua conta.

run uuidÉ um identificador global exclusivo (guid) que você pode usar para identificar execuções entre contas ou para distinguir entre duas execuções em sua conta que têm a mesma ID de execução.

> **Note**
>
> Para fins de proveniência de dados, recomendamos que você use o run uuid para identificar execuções de forma exclusiva. Também run uuid é o melhor identificador para vincular ao sistema interno de gerenciamento de informações do laboratório (LIMs) ou ao sistema de rastreamento de amostras.

Você pode usar o Amazon Q CLI para otimizar suas execuções e analisar o desempenho das corridas. Para obter mais informações, consulte Exemplos de solicitações para a Amazon Q CLI e HealthOmics o tutorial de IA generativa da Agentic sobre. GitHub

Tópicos

- Execute tipos de armazenamento em HealthOmics fluxos de trabalho
- Execute o modo de retenção para HealthOmics execuções
- HealthOmics entradas de execução
- Execute o ciclo de vida em um fluxo de trabalho HealthOmics
- HealthOmics saídas de execução
- Motivos de falha na execução
- Ciclo de vida da tarefa em execução HealthOmics
- Execute a otimização para um HealthOmics fluxo de trabalho privado
- Execute operações em HealthOmics

Execute tipos de armazenamento em HealthOmics fluxos de trabalho

Quando você inicia uma execução, HealthOmics aloca armazenamento de execução temporário para o mecanismo de fluxo de trabalho usar durante a execução. HealthOmicsfornece o armazenamento temporário de execução como um sistema de arquivos.

Para um determinado fluxo de trabalho ou execução de fluxo de trabalho, você pode escolher o armazenamento de execução dinâmico ou estático. Por padrão, HealthOmics fornece armazenamento de execução DINÂMICA.

> **Note**
>
> O uso do armazenamento em execução gera cobranças em sua conta. Para obter informações sobre preços sobre armazenamento de execução estática e dinâmica, consulte HealthOmicspreços.

As seções a seguir fornecem informações a serem consideradas ao decidir qual tipo de armazenamento de execução usar.

Armazenamento de execução dinâmica

Recomendamos usar o armazenamento de execução dinâmica para a maioria das execuções, incluindo execuções que exigem tempos de início mais rápidos, execuções nas quais você não conhece as necessidades de armazenamento com antecedência e para ciclos de testes de desenvolvimento iterativos.

Você não precisa estimar o armazenamento ou a taxa de transferência necessários para a execução. HealthOmics aumenta ou diminui dinamicamente o tamanho do armazenamento, com base na utilização do sistema de arquivos durante a execução. HealthOmics também dimensiona dinamicamente a produtividade com base nas necessidades do fluxo de trabalho. Uma execução nunca falha devido a um erro de falta de armazenamento no sistema de arquivos.

O armazenamento de execução dinâmica fornece um provisioning/deprovisioning tempo mais rápido do que o armazenamento de execução estática. A configuração mais rápida é uma vantagem para a maioria dos fluxos de trabalho e também durante development/test os ciclos.

Após a conclusão da execução (caminho de sucesso ou caminho de falha), a operação da API GetRun retorna o armazenamento máximo usado pela execução no campo StorageCapacity. Você também pode encontrar essas informações nos registros do manifesto de execução localizados no

grupo de omics registros. Para uma execução de armazenamento dinâmico concluída em 2 horas, o valor máximo de armazenamento pode não estar disponível.

Para armazenamento dinâmico de execução, a execução provisiona um sistema de arquivos que usa o protocolo NFS. O NFS trata as operações CREATE, DELETE e RENAME de arquivos como não idempotentes, o que pode ocasionalmente levar a condições de corrida para essas operações que seu código precisa manipular normalmente. Por exemplo, seu código não deve falhar se tentar excluir um arquivo que não existe. Antes de adotar o armazenamento dinâmico de execução, recomendamos ajustar o código do fluxo de trabalho para torná-lo resiliente a operações de arquivo não idempotentes. Consulte Exemplos de código para manuseio seguro de operações não idempotentes.

Exemplos de código para manuseio seguro de operações não idempotentes

O exemplo de python a seguir mostra como excluir um arquivo sem falhar se o arquivo não existir.

```
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")
```

Os exemplos a seguir usam o shell Bash. Para remover um arquivo com segurança, mesmo que ele não exista, use:

```
rm -f my_file
```

Para mover (renomear) um arquivo com segurança, execute o comando move somente se o arquivo `old_name` existir no diretório atual.

```
[ -f old_name ] && mv old_name new_name
```

Para criar um diretório, use o seguinte comando:

```
mkdir -p mydir/subdir/
```

Armazenamento de execução estática

Para armazenamento estático de execução, a execução provisiona um sistema de arquivos que usa o protocolo Lustre. Por padrão, esse protocolo é resiliente a operações de arquivos não idempotentes. Você não precisa ajustar o código do fluxo de trabalho para lidar com operações de arquivo não idempotentes.

HealthOmics aloca uma quantidade fixa de armazenamento em execução. Você especifica esse valor ao iniciar a execução. O armazenamento de execução padrão é 1200 GiB, se você não especificar um valor. Quando você especifica um valor para o tamanho do armazenamento na solicitação da StartRun API, o sistema arredonda o valor para o múltiplo mais próximo de 1200 GiB. Se esse tamanho de armazenamento não estiver disponível, ele será arredondado para o múltiplo mais próximo de 2400 GiB.

Para armazenamento de execução estática, HealthOmics provisiona os seguintes valores de taxa de transferência:

- Taxa de transferência básica de 200 por MB/s TiB de capacidade de armazenamento provisionada.
- Taxa de transferência máxima de até 1300 por MB/s TiB de capacidade de armazenamento provisionada.

Se o tamanho de armazenamento especificado for muito baixo, a execução falhará com um erro de Falta de armazenamento para o sistema de arquivos. O armazenamento de execução estática é uma boa opção para fluxos de trabalho previsíveis com requisitos de armazenamento conhecidos.

O armazenamento de execução estática é adequado para cargas de trabalho grandes e intermitentes com alta simultaneidade de tarefas (por exemplo, um grande volume de RNASeq amostras processadas paralelamente). Ele fornece maior taxa de transferência do sistema de arquivos por GiB e menor custo por GiB do que o armazenamento de execução dinâmica.

Calculando o armazenamento de execução estática necessário

Um fluxo de trabalho exige capacidade adicional quando usa armazenamento de execução estática (em comparação com o armazenamento de execução dinâmica) porque a instalação básica do sistema de arquivos usa 7% da capacidade estática do sistema de arquivos.

Se você executar um fluxo de trabalho dinâmico de armazenamento de execução para medir o armazenamento máximo usado pela execução, use o cálculo a seguir para determinar a quantidade mínima de armazenamento estático necessária:

```
static storage required =
       maximum storage in GiB used by the dynamic run storage
         + (total static file system size in GiB * 0.07)
```

Por exemplo:

```
    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.
```

Portanto, 1200 GiB (a capacidade mínima para armazenamento de execução estática) é suficiente para essa execução.

Execute o modo de retenção para HealthOmics execuções

Após a conclusão de uma execução, HealthOmics arquiva os metadados da execução em. CloudWatch Por padrão, CloudWatch mantém os dados de execução indefinidamente, a menos que você altere a política CloudWatch de retenção. As saídas de execução também são armazenadas no Amazon S3 até que você as exclua.

Um dos ajustáveis HealthOmics cotas de serviço é o maximum number of runs (active and inactive) em uma região. HealthOmics retém os metadados de execução de até esse número de execuções para uso pelo console e pelas operações da API (ListRuns e). GetRun Ao iniciar uma execução, você pode definir o parâmetro do modo de retenção de execução para indicar o comportamento de retenção da execução. O parâmetro suporta os valores REMOVE e RETAIN.

Para uma nova execução com o modo de retenção definido como REMOVE, se HealthOmics tentar adicionar a execução depois de já ter salvo o número máximo de execuções, ele removerá

automaticamente os metadados da execução mais antiga que definiu o modo REMOVE. Essa remoção não afeta os dados armazenados no Amazon S3 CloudWatch ou no Amazon S3.

RETAIN é o valor padrão para o modo de retenção de execução. Para execuções nesse modo, o sistema não exclui os metadados de execução. Se HealthOmics atingir o número máximo de execuções, tudo definido como RETAIN, você não poderá criar execuções adicionais até excluir algumas execuções.

Se você planeja executar um lote com mais do que o número máximo de execuções ao mesmo tempo, certifique-se de definir o modo de retenção de execução como REMOVE. Caso contrário, o lote falhará ao HealthOmics tentar iniciar a próxima execução após o máximo.

Considerações adicionais sobre o uso do modo de retenção REMOVE:

- Ao começar a usar REMOVE como modo de retenção, considere excluir uma ou mais execuções que usam o modo RETAIN para liberar slots. À medida que você inicia outras execuções de REMOVE, a remoção automática assume o controle, então há vagas suficientes para novas execuções.
- Se você quiser executar novamente uma execução arquivada (ou um conjunto de execuções), use a ferramenta CLI de HealthOmics reexecução. Para obter mais informações e exemplos de como usar essa ferramenta, consulte Executar novamente o Omics no repositório de ferramentas. HealthOmics GitHub
- Recomendamos que você configure um nome exclusivo para cada execução. Depois de HealthOmics remover uma execução, você não poderá usar o console ou a API para encontrar o nome da execução ou o ID da execução. No entanto, você pode usar CloudWatch para pesquisar o nome da execução, então use nomes exclusivos para obter os melhores resultados de pesquisa.
- Você pode usar o CloudWatch start-query comando para obter informações sobre uma execução arquivada. Se o nome da execução não for exclusivo, a consulta poderá retornar vários manifestos. Os parâmetros de hora de início e hora de término definem o intervalo de tempo para a pesquisa.

```
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>
```

O start-query comando retorna um ID de consulta. Passar o ID da consulta para o get-query-results comando retorna os resultados da consulta.

```
aws logs get-query-results --query-id QueryId
```

HealthOmics entradas de execução

Se a definição do fluxo de trabalho especificar arquivos de entrada para o fluxo de trabalho ou para as tarefas do fluxo HealthOmics de trabalho, transforme os arquivos em um volume temporário dedicado à execução do fluxo de trabalho. Esses arquivos de entrada são somente para leitura, o que impede que as tarefas modifiquem possíveis entradas para outras tarefas no fluxo de trabalho. Para importações de diretórios, os diretórios também são somente para leitura.

Muitos aplicativos de genômica presumem que os arquivos de índice estão co-localizados com os arquivos de sequência (como um `bai` arquivo complementar para um `bam` arquivo). Para incluir arquivos de índice, especifique-os como entradas de tarefas na definição do fluxo de trabalho.

Tópicos

- Gerenciando o tamanho dos parâmetros de execução
- Formatos de parâmetros de entrada do Amazon S3
- Estados de arquivamento de entrada do Amazon S3

Gerenciando o tamanho dos parâmetros de execução

Ao iniciar uma execução, você especifica as entradas de execução no objeto ou arquivo JSON dos parâmetros de execução. Você pode especificar até 50 KB de parâmetros de execução para o fluxo de trabalho. Você pode usar as seguintes técnicas para permanecer dentro dessa restrição de tamanho:

- Use importações de diretórios

  Para especificar um grande número de arquivos de entrada, especifique um parâmetro como o local do Amazon S3 que contém todos os arquivos, em vez de especificar um parâmetro para cada local de arquivo. Para obter mais informações, consulte o próximo tópico (formatos de parâmetros de entrada do Amazon S3).
- Use uma folha de amostra

Uma planilha de amostra é um arquivo CSV ou TSV com uma coluna para o endereço fastq.gz (ou duas para leitura em pares) e colunas adicionais para metadados, como nomes de amostras. Você especifica a planilha de amostra como um parâmetro de entrada de execução em vez de um parâmetro para cada arquivo de entrada.

Seu fluxo de trabalho define como sua planilha de amostra é mapeada para estruturas de dados no fluxo de trabalho. Embora você possa escrever código para folhas de amostra em WDL e CWL, elas são mais comuns em. NextFlow Para ver um exemplo, consulte a planilha de amostra no site nf-core GitHub .

Formatos de parâmetros de entrada do Amazon S3

Para um parâmetro de entrada que aceita uma localização do Amazon S3, o parâmetro pode especificar a localização de um arquivo ou de um diretório inteiro de arquivos. Usar um diretório tem as seguintes vantagens:

- Conveniência — Você especifica o nome do diretório como parâmetro. Você não lista cada nome de arquivo.
- Compacidade — O tamanho máximo do arquivo do parâmetro de entrada é 50 KB. Se você fornecer uma longa lista de nomes de arquivos de entrada, poderá exceder esse máximo.

O Amazon S3 é um sistema plano de armazenamento de objetos, por isso não oferece suporte a diretórios. Você agrupa arquivos em um “diretório” dando a cada arquivo o mesmo prefixo de chave de objeto. Para obter mais informações sobre prefixos de chave de objeto do Amazon S3, consulte Organização de objetos usando prefixos.

HealthOmics interpreta o valor do parâmetro de entrada da seguinte forma:

- Se a localização do Amazon S3 não terminar com uma barra ou usar o padrão global, HealthOmics espere que o valor do parâmetro seja a chave para um objeto do Amazon S3.

  Por exemplo, você especifica `s3://myfiles/runs/inputs/a/file1.fastq` para inserir file1.fastq
- Se a localização do Amazon S3 terminar com uma barra, HealthOmics interpreta o valor do parâmetro como um prefixo do Amazon S3. Ele carrega todos os objetos do Amazon S3 com esse prefixo.

Por exemplo, você pode especificar `s3://myfiles/runs/inputs/a/` o carregamento de todos os objetos cujas chaves comecem com esse prefixo.

- Para o Nextflow, oferece suporte HealthOmics parcial ao padrão global do Amazon S3 nos parâmetros de entrada. URIs

  Por exemplo, você pode especificar "`s3://myfiles/runs/inputs/a/*.gz`" a entrada de todos os arquivos.gz cujas chaves comecem com esse prefixo.

Nextflow: Tratamento do padrão Glob nas entradas do Amazon S3

Padrão Glob	HealthOmics Comportamento da partida	Observações
s3://bucket/directory/*.txt	Corresponde a todos os `.txt` objetos em qualquer profundidade sob o prefixo s3://bucket/directory/. For example, matches s3://bucket/directory/abc.txt or s3://bucket/directory/subDir/123.txt etc.	
s3://bucket/directory/**/*.txt	Corresponde a todos os `.txt` objetos em qualquer profundidade sob o prefixo s3://bucket/directory/. For example, matches s3://bucket/directory/abc.txt or s3://bucket/directory/subDir/123.txt etc.	No S3, `**` é equivalente a. `*`
s3://bucket/directory/{a,b}.txt	s3://bucket/directory/a.txt, s3://bucket/directory/b.txt	
s3://bucket/directory/?. txt	Corresponde a objetos na raiz do prefixo cujo nome de arquivo é um único caractere seguido por. `.txt` Por exemplo, ele corresponde	

Padrão Glob	HealthOmics Comportamento da partida	Observações
	a s3://bucket/directory/a.txt but not s3://bucket/directory/someDir/a.txt or s3://bucket/directory/someDir/subDir/a.txt	
s3://bucket/directory/[0-9].txt	s3:///9.txt bucket/directory/0.txt, s3://bucket/directory/1.txt, ... ,s3://bucket/directory	
s3://bucket/directory/[0-9].txt	s3:///3.txt bucket/directory/1.txt, s3://bucket/directory/2.txt, s3://bucket/directory	
s3://bucket/directory/[0-9].txt	s3://bucket/directory/b.txt, s3://bucket/directory/c.txt, ... ,s3://bucket/directory/Y.txt	

Tratamento específico do idioma da barra dupla nas entradas do Amazon S3

HealthOmics retém o comportamento do mecanismo nativo de cada mecanismo de fluxo de trabalho ao lidar com barras duplas no Amazon S3 URIs, para que você não precise fazer nenhuma alteração em seus fluxos de trabalho ao migrá-los para. HealthOmics As seções a seguir descrevem como cada motor lida com vários cenários.

WDL

Se o parâmetro de entrada incluir uma barra dupla no meio ou no final do URI, o mecanismo WDL manterá a barra dupla.

Parâmetro de entrada	Localização esperada	
x3://1.fastq myfiles/runs/inputs//file	x3://1.fastq myfiles/runs/inputs//file	
s3:////myfiles/runs/inputs	s3:////myfiles/runs/inputs	

Próximo fluxo

Se o parâmetro de entrada incluir uma barra dupla no meio do URI, o mecanismo Nextflow manterá a barra dupla. Para uma barra dupla no final do URI, o mecanismo Nextflow a resolve em uma única barra.

Parâmetro de entrada	Localização esperada	
x3://1.fastq myfiles/runs/inputs//file	x3://1.fastq myfiles/runs/inputs//file	
s3://myfiles//runs/inputs//*.gz	s3://myfiles//runs/inputs//*.gz	
s3://myfiles//runs/inputs//	s3://myfiles//runs/inputs/	

CAPUZ

Se o parâmetro de entrada incluir uma barra dupla no meio ou no final do URI, o mecanismo CWL manterá a barra dupla.

Parâmetro de entrada	Localização esperada	
s3://myfiles// runs/inputs//file 1.fastq	s3://myfiles// runs/inputs//file 1.fastq	
s3://myfiles//runs/inputs//	s3://myfiles//runs/inputs//	

Estados de arquivamento de entrada do Amazon S3

HealthOmics pode recuperar objetos do Amazon S3 que o S3 entrega em tempo real. Para objetos que estão nos seguintes estados de armazenamento arquivado, restore os objetos para disponibilizá-los HealthOmics:

- Classes de armazenamento Flexible Retrieval ou Deep Archive no Amazon S3 Glacier.
- Camadas de acesso arquivado ou acesso profundo ao arquivamento em camadas inteligentes.

Para obter informações sobre restauração de objetos, consulte Restauração de um objeto arquivado no Guia do usuário do Amazon S3.

Execute o ciclo de vida em um fluxo de trabalho HealthOmics

Você pode acompanhar o progresso de uma execução monitorando o status da execução. HealthOmics atualiza o status da execução à medida que a execução prossegue em seu ciclo de vida.

Você pode recuperar o status de execução usando qualquer um dos seguintes métodos:

- O HealthOmics console exibe o status de cada execução na Runs página.
- A operação GetRun da API retorna o status de execução atual.
- Você pode monitorar o status da execução usando EventBridge eventos. Para obter mais informações, consulte Usando EventBridge com AWS HealthOmics.

Tópicos

Valores de status de execução

Quando você inicia uma execução, HealthOmics define o status da execução comoPending. Conforme a execução avança em seu ciclo de vida, HealthOmics atualiza o valor do status para refletir seu progresso atual.

> **Note**
>
> Você não incorre em cobranças durante nenhum status de corrida que não seja em execução. Consulte a próxima seção para obter detalhes.

HealthOmics suporta os seguintes valores de status de execução:

Pendente

A corrida está na fila, esperando para começar. Normalmente, as execuções permanecem pendentes por um breve período antes de começarem.

- As execuções podem permanecer pendentes por mais tempo se você enviar vários trabalhos ao mesmo tempo.
- As execuções permanecem pendentes após sua conta atingir o número máximo de execuções simultâneas.
- Uma execução permanece em Pendente se a execução fizer parte de um grupo de execução que atingiu qualquer um dos valores máximos de seus recursos.
- Você pode ajustar as prioridades de execução para que execuções específicas em fila comecem antes das outras. Para obter mais informações sobre a prioridade de execução, consulte Prioridade de execução.

Starting

HealthOmics cria a execução e provisiona os recursos necessários para a execução (como armazenamento temporário da execução e o nó do mecanismo).

- HealthOmics provisiona o armazenamento temporário de execução no início da execução e desprovisiona o armazenamento de execução quando a execução está sendo interrompida.

Em execução

Uma execução permanece no status Em execução durante o processo de importação, o processamento de cada tarefa e o processo de exportação.

- HealthOmics importa os arquivos de entrada para o sistema de arquivos de armazenamento de execução temporária. Os arquivos de entrada são somente para leitura, para evitar que as tarefas modifiquem as entradas para outras tarefas em um fluxo de trabalho.
- Durante a exportação do arquivo, HealthOmics exporta os arquivos de saída do sistema de arquivos de armazenamento executado para o local do S3.
- HealthOmics entrega os registros de execução e os registros de tarefas CloudWatch em tempo real enquanto o status de execução é Executando. Para obter mais informações, consulte Login CloudWatch.

Parando

Após a conclusão do processo de exportação, a execução passa para o status Interrompendo.

- HealthOmics desprovisiona todos os recursos (incluindo o sistema de arquivos de armazenamento em execução e o nó do mecanismo).

Concluído

A execução muda para Concluída após HealthOmics concluir o desprovisionamento do recurso.

- HealthOmics concluiu todas as tarefas de execução e exportou os dados de saída sem erros.
- As saídas de execução estão disponíveis no local de saída especificado do URI do Amazon S3. Para WDL e CWL, HealthOmics gera um arquivo de resumo da saída de execução, que fornece informações sobre o. HealthOmics saídas de execução
- Os registros do manifesto de execução final e os registros do mecanismo (se aplicável) estão disponíveis em CloudWatch.
- Para execuções que oferecem suporte a novas tentativas de tarefas, uma execução com status Concluído pode incluir uma ou mais tarefas que falharam. Desde que uma nova tentativa de tarefa seja bem-sucedida para cada tarefa que falhou, a execução será HealthOmics transferida para Concluída. HealthOmics atribui um novo ID de tarefa a cada nova tentativa, IDs para que a execução inclua a tarefa das tentativas malsucedidas e da tentativa concluída.

Falha

HealthOmics encontrou um ou mais erros e não conseguiu concluir todas as tarefas de execução.

- Uma execução com falha passa pelo status de parada enquanto HealthOmics desprovisiona os recursos.

Cancelado

Um usuário iniciou uma solicitação para cancelar a execução.

- HealthOmics interrompe qualquer tarefa em execução e desprovisiona todos os recursos.
- HealthOmics não exporta nenhum dado de saída de execução quando um usuário cancela uma execução. Você não tem acesso a nenhum arquivo intermediário para uma execução cancelada.
- Sua conta incorre em cobranças pelas tarefas e recursos que a execução consumiu durante o status Em execução antes do cancelamento.
- Não haverá cobranças se você cancelar uma corrida no status Pendente ou Inicial.

Tentativas de tarefas

HealthOmics suporta novas tentativas de tarefas para tarefas que falham devido a erros de serviço (códigos de status HTTP 5XX).

Se todas as tarefas na execução forem concluídas, mesmo que sejam necessárias novas tentativas, a execução será HealthOmics transferida para Concluída. HealthOmics atribui um novo ID de tarefa a cada nova tentativa, IDs para que a execução inclua a tarefa das tentativas malsucedidas e da tentativa concluída.

O comportamento padrão de repetição depende da linguagem de definição usada pelo fluxo de trabalho. O padrão para o Nextflow é sem novas tentativas. Para WDL e CWL, tente até duas HealthOmics tentativas de uma tarefa com falha, mas você pode optar por não repetir tarefas para tarefas específicas ou para todas as tarefas em um fluxo de trabalho. A repetição da tarefa é útil para solucionar erros de serviço intermitentes. No entanto, você pode considerar optar por não participar de uma tarefa que seja idempotente.

Para obter informações específicas sobre cada linguagem de definição de fluxo de trabalho, consulte os tópicos a seguir:

- WDL — Configure o comportamento de repetição de tarefas na definição do fluxo de trabalho. Consulte Configurar o comportamento de repetição de tarefas da WDL.
- Nextflow — Configure o comportamento de repetição da tarefa no arquivo de configuração do Nextflow ou na definição do fluxo de trabalho. Consulte Configurar o comportamento de repetição de tarefas do Nextflow.
- CWL — Configure o comportamento de repetição de tarefas na definição do fluxo de trabalho. Consulte Configurar o comportamento de repetição de tarefas do CWL.

Implicações de preços do status de execução

Sua conta pode incorrer em cobranças enquanto o status de execução estiver em execução. Você não incorre em cobranças durante nenhum outro status de execução. Por exemplo, não há cobrança por recursos quando a execução está iniciando ou parando.

Uma execução com status Em execução tem as seguintes implicações de cobrança:

- Sua conta incorre em cobranças pelo uso do sistema de arquivos de armazenamento em execução enquanto o status de execução é Executando. Para obter informações sobre os tipos de armazenamento de execução, consulteExecute tipos de armazenamento em HealthOmics fluxos de trabalho.
- Sua conta incorre em cobranças pela execução de tarefas, com base nos recursos de computação e memória que você especificou para cada tarefa na definição do fluxo de trabalho e com base na

duração da tarefa. Para obter mais informações, consulte Requisitos de computação e memória para tarefas HealthOmics.

- Cada tarefa tem um limite mínimo de cobrança de um minuto. Se você executar uma tarefa por menos de um minuto, incorrerá em uma cobrança pelo mínimo de um minuto de uso. Se possível, agrupe pequenas tarefas para otimizar os custos. O agrupamento de tarefas também reduz o tempo de execução, evitando a criação de várias tarefas sequenciais.

Para obter informações adicionais sobre HealthOmics preços, consulte os HealthOmics Preços.

HealthOmics saídas de execução

Quando uma execução de WDL ou CWL é concluída, as saídas incluem um arquivo de resumo de saída (no formato JSON) que lista todas as saídas produzidas pela execução. Você pode usar o arquivo de resumo de saída para estas finalidades:

- Determine programaticamente os arquivos de saída gerados pela execução.
- Valide se a execução produziu todas as saídas esperadas.

Tópicos

- Execute o resumo da saída para a WDL
- Execute o resumo da saída para CWL

Execute o resumo da saída para a WDL

Quando uma execução da WDL é concluída, HealthOmics cria um arquivo de resumo de saída chamado. output.json

Para cada saída do fluxo de trabalho, há um key/value par correspondente no arquivo. A chave contém o nome do fluxo de trabalho e o nome da saída no seguinte formato:`WorkflowName.output_name`. Para uma saída de arquivo, o valor é um URI do S3 apontando para o local de saída no S3 em que o arquivo está armazenado. Para uma saída Array [File], o valor é uma matriz de S3 URIs.

O exemplo a seguir mostra o output.json arquivo de um fluxo de trabalho chamadoBWAMappingWorkflow.

```
{
```

```
  "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"
  ]
}
```

Se o fluxo de trabalho produzir saídas com tipos que não sejam de arquivo (como String, Int, Float ou Bool), o valor do campo será um primitivo JSON. Por exemplo:

```
{
  "MyWorkflow.my_int_ouput": 1,
  "MyWorkflow.my_bool_output": false,
    ...
}
```

Execute o resumo da saída para CWL

Quando uma execução do CWL é concluída, HealthOmics cria um arquivo de resumo de saída nomeado outputs.json no seguinte local:

```
{my-S3outputpath}/{runId}/{run-uuid}/logs/outputs.json
```

O arquivo de resumo da saída inclui uma lista de saídas. Cada saída é um key/value par, em que a chave é o nome da saída. O valor é um objeto que inclui as seguintes propriedades:

- localização — O caminho totalmente qualificado para o arquivo de saída
- basename — A parte do nome do arquivo do caminho
- class — O tipo da saída, que normalmente é Arquivo
- size — O tamanho do arquivo em bytes

No exemplo a seguir, o arquivo output.json tem uma lista de dois arquivos de saída.

```
{
  "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 falha na execução

Se uma execução falhar, use a operação GetRunda API para recuperar o motivo da falha.

Analise o motivo da falha para ajudá-lo a solucionar o motivo da falha na execução. A tabela a seguir lista cada motivo da falha junto com uma descrição do erro.

Motivo da falha	Descrição do erro
FALHA EM ASSUMIR A FUNÇÃO	HealthOmics não tem permissão para assumir a função. Especifique o HealthOmics diretor na relação de confiança da função.
NÃO É POSSÍVEL INICIAR O CONTÊINER_ERROR	Não é possível iniciar a tarefa do fluxo de trabalho:*name*, id: *ID* contêiner usando a imagem:*image name*. Verifique se a imagem é válida e tente novamente.

Motivo da falha	Descrição do erro
NÃO É POSSÍVEL INICIAR O CONTÊINER_SIZE_ERROR	Não é possível iniciar a tarefa do fluxo de trabalho:*name*, id: *ID* contêiner usando a imagem:*image name*. Verifique se o tamanho da imagem é menor que 45 GiB (95 GiB para uma instância de GPU) e tente novamente.
ERRO_PERMISSÃO_ECR	HealthOmics não tem permissão para acessar o URI da imagem. Confirme se o repositório privado do Amazon ECR existe e concedeu acesso ao responsável pelo HealthOmics serviço.
FALHA NA EXPORTAÇÃO	A exportação falhou. Verifique se o bucket de saída existe e se a função de execução tem permissão de gravação no bucket.
SISTEMA DE ARQUIVOS FORA DO ESPAÇO	O sistema de arquivos não tem espaço suficiente. Aumente o tamanho do sistema de arquivos e execute novamente.
FALHA NA VERIFICAÇÃO DA IMAGEM	Não foi possível verificar a imagem*image name*. Para corrigir o problema, tente extrair a imagem e enviá-la novamente para o repositório ECR.
FALHA NA IMPORTAÇÃO	A importação falhou. Verifique se o arquivo de entrada existe e se a função de execução pode acessar a entrada.
INACTIVE_OMICS_STORAGE_RESOURCE	O URI HealthOmics de armazenamento não está no estado ATIVO. Ative o conjunto de leitura e tente novamente. Para saber mais sobre como ativar conjuntos de leitura, consulteAtivando conjuntos de leitura em HealthOmics.
URI DE ENTRADA NÃO ENCONTRADO	O URI fornecido não existe:*uri*. Verifique se o caminho do URI existe e confirme se a função pode acessar o objeto.
FALHA NA RESERVA DA INSTÂNCIA	Não há capacidade de instância suficiente para concluir a execução do fluxo de trabalho. Aguarde e tente executar o fluxo de trabalho novamente.
URI DE ECR_IMAGE_URI INVÁLIDO	A estrutura do URI da imagem do Amazon ECR não é válida. Forneça um URI válido e tente novamente.

Motivo da falha	Descrição do erro
VALOR_DO_TASK_RESOURCE_INVÁLIDO	A GPU, a CPU ou a memória solicitadas são muito altas para a capacidade computacional disponível ou são menores que o valor mínimo de 1 para a tarefa. *ID*
ENTRADA_URI_INVÁLIDA	A estrutura do URI não é válida*uri*. Verifique a estrutura do URI e tente novamente.
RECURSO_DE_ENTRADA_MODIFICADO	O URI fornecido *uri* foi modificado após o início da execução. Tente executar novamente.
ERROR_OF_OF_MEMORY_ERROR	A tarefa do fluxo de trabalho *ID* ficou sem memória. Aumente o valor da memória na definição do fluxo de trabalho e tente executar novamente.
FALHA NA EXECUÇÃO DA TAREFA	A execução falhou porque a tarefa falhou. Para depurar a falha da tarefa, use a operação da GetRunTaskAPI e o stream do Amazon CloudWatch Logs.
TEMPO LIMITE DE EXECUÇÃO	Tempo limite de execução após *number* minutos.
ERRO_DE_SERVIÇO	Houve um erro transitório no serviço. Tente executar o fluxo de trabalho novamente.
TEMPO LIMITE DE EXECUÇÃO DA TAREFA	A tarefa *id* atingiu o tempo limite após *number* alguns segundos.
TAMANHO_DE_ENTRADA NÃO SUPORTADO	O tamanho total da entrada é muito alto. Diminua o tamanho da entrada e tente novamente.
FALHA NA EXECUÇÃO DO FLUXO DE TRABALHO	Falha na execução do fluxo de trabalho. Analise o fluxo de CloudWatch registros do mecanismo de registros: *ID* para depurar a falha.

Motivo da falha	Descrição do erro
FALHA NA VALIDAÇÃO _DE_FLUXO DE TRABALHO	HealthOmics não suporta a versão solicitada do Nextflow: *version* --. A versão mais recente suportada é*version*. Modifique sua versão do Nextflow para uma versão compatível e tente novamente.
TIPO_DE_INSTÂNCIA DE GPU_NÃO SUPORTADO	O tipo de instância solicitado não é compatível com*Region*. Tente executar novamente com um tipo de instância de GPU compatível com essa região. Os tipos de instância disponíveis são*GPU instance types*.

Orientação para corridas sem resposta

Ao desenvolver novos fluxos de trabalho, as execuções ou tarefas específicas podem ficar “travadas” ou “travadas” se houver problemas com seu código e as tarefas não saírem dos processos adequadamente. Isso pode ser difícil de solucionar e capturar, pois é normal que as tarefas sejam executadas por longos períodos. Para evitar e identificar execuções que não respondem, siga as melhores práticas sugeridas nas seções a seguir.

Práticas recomendadas para evitar execuções sem resposta

- Certifique-se de fechar todos os arquivos abertos no código da tarefa. Abrir muitos arquivos pode ocasionalmente levar a problemas de segmentação nos mecanismos de fluxo de trabalho.
- Os processos em segundo plano criados por uma tarefa de fluxo de trabalho devem sair quando a tarefa for encerrada. No entanto, se um processo em segundo plano não sair corretamente, você deverá encerrá-lo explicitamente no código da tarefa.
- Garanta que seus processos não ocorram sem sair. Isso pode causar uma execução sem resposta e requer uma alteração no código de definição do fluxo de trabalho para ser resolvido.
- Forneça alocação adequada de memória e CPU para suas tarefas. Analise os CloudWatch registros ou use-os Execute o Analyzer em execuções concluídas com êxito do seu fluxo de trabalho para verificar se você tem a alocação de computação ideal. Use o `headroom` parâmetro Run Analyzer para incluir espaço adicional, garantindo que os processos tenham recursos suficientes para serem concluídos. Inclua pelo menos 5% de espaço livre na memória e na CPU alocadas, para considerar os processos do sistema operacional em segundo plano.

  - Além disso, aumente o tamanho da largura de banda da instância se a instância exigir uma taxa de transferência maior. EC2 As instâncias da Amazon com menos de 16 v CPUs (tamanho 4xl ou menor) podem experimentar uma explosão na taxa de transferência. Para obter mais informações sobre a taxa de transferência de EC2 instâncias da Amazon, consulte a largura de banda da instância EC2 disponível da Amazon.
- Verifique se você está usando o tamanho correto do sistema de arquivos para suas execuções. Para execuções que não respondem e estão usando armazenamento de execução estática, considere aumentar a alocação de armazenamento de execução estática para permitir maior taxa de transferência de E/S e capacidade de armazenamento no sistema de arquivos. Analise o manifesto de execução para ver o armazenamento máximo do sistema de arquivos e use o Run Analyzer para determinar se a alocação do sistema de arquivos precisa ser aumentada.

Práticas recomendadas para capturar execuções que não respondem

- Ao desenvolver novos fluxos de trabalho, use um grupo de execução com o limite máximo de tempo de execução definido para capturar o código em fuga. Por exemplo, se uma execução levar 1 hora para ser concluída, coloque-a em um grupo de execução que atinja o tempo limite após 2 ou 3 horas (ou em um período de tempo diferente com base no seu caso de uso) para capturar trabalhos desnecessários. Além disso, aplique um buffer para contabilizar a variação nos tempos de processamento.
- Configure uma série de grupos de execução com diferentes limites máximos de tempo de execução. Por exemplo, você pode atribuir tiragens curtas a um grupo de execução que encerra as execuções após algumas horas e a um grupo de corridas longas que encerra as execuções após alguns dias, com base na duração esperada do fluxo de trabalho.
- HealthOmics tem um limite máximo padrão de serviço de duração de execução de 604.800 segundos, ou 7 dias, que é ajustável por meio de uma solicitação na ferramenta de cotas. Solicite um aumento do limite de serviço dessa cota somente se você tiver execuções com duração aproximada de uma semana. Se você tem uma combinação de execuções curtas e longas e não está usando grupos de execução, considere colocar as execuções de longa duração em uma conta separada com um limite máximo de serviço de duração de execução maior.
- Inspecione os CloudWatch registros em busca de tarefas que você suspeita que possam não responder. Se uma tarefa normalmente gera instruções de log regulares e não o faz há um longo período, é provável que a tarefa esteja paralisada ou congelada.

O que fazer se você encontrar uma corrida sem resposta

- Cancele a corrida para evitar custos adicionais.
- Inspecione os registros de tarefas para verificar se algum processo falhou ao sair corretamente.
- Inspecione os registros do motor para identificar qualquer comportamento anormal do motor.
- Compare os registros de tarefas e mecanismos da execução sem resposta com os de execuções idênticas e concluídas com êxito. Isso pode ajudar a identificar quaisquer diferenças que possam ter causado o comportamento de não resposta.
- Se você não conseguir determinar a causa raiz, crie um caso de suporte e inclua o seguinte:
  - ARN da execução paralisada e ARN de uma execução idêntica que foi concluída com êxito.
  - Registros do motor (disponíveis quando a execução é cancelada ou falha)
  - Registros de tarefas para a tarefa que não responde. Não exigimos registros de tarefas de todas as tarefas do fluxo de trabalho para solucionar problemas.

Ciclo de vida da tarefa em execução HealthOmics

Uma tarefa é um processo único dentro de uma execução. HealthOmics mapeia cada tarefa em seu fluxo de trabalho para um tipo de instância de computação ômica que melhor se adapte aos recursos necessários da tarefa. Você especifica os recursos necessários na definição do fluxo de trabalho. Para obter mais informações, consulteRequisitos de computação e memória para tarefas HealthOmics.

HealthOmics fornece armazenamento temporário de execução para a tarefa a ser usada. HealthOmics copia os arquivos de entrada da tarefa para o armazenamento de execução temporária como arquivos somente para leitura. HealthOmics fornece links simbólicos para que a tarefa possa acessar os arquivos de entrada do diretório de trabalho. A tarefa tem acesso somente aos arquivos que você declara no arquivo de definição do fluxo de trabalho.

Valores de status da tarefa

Você pode acompanhar o progresso de uma tarefa monitorando o status da tarefa. Quando você inicia uma execução, HealthOmics define o status da tarefa Pending para cada tarefa na execução. Quando a tarefa é iniciada e progride em seu ciclo de vida, HealthOmics atualiza o valor do status para refletir seu progresso atual.

Você pode recuperar o status da tarefa usando qualquer um dos seguintes métodos:

- O HealthOmics console exibe o status de cada tarefa em uma execução na Run details página.
- A operação GetRunTask da API retorna o status da tarefa.

- Você pode monitorar o status da tarefa usando EventBridge eventos. Para obter mais informações, consulte Usando EventBridge com AWS HealthOmics.

Você pode recuperar o status atual de uma tarefa usando a operação da GetRunTask API. O HealthOmics console exibe o status de cada tarefa em uma execução na Run details página.

HealthOmics suporta os seguintes valores de status da tarefa:

Pendente

Sua tarefa está na fila, esperando para começar. As tarefas permanecem pendentes por um breve período antes de serem iniciadas.

- As tarefas permanecem pendentes após sua conta atingir o número máximo de tarefas simultâneas.
- As tarefas permanecerão pendentes se a execução fizer parte de um grupo de execução que atingiu qualquer um dos valores máximos de seus recursos.
- Você pode ajustar as prioridades de execução para que execuções específicas em fila e suas tarefas comecem antes de outras execuções em fila. Para obter mais informações sobre a prioridade de execução, consulte Prioridade de execução

Starting

HealthOmics está criando a tarefa e provisionando os recursos necessários para a tarefa, como o nó da tarefa do fluxo de trabalho.

Em execução

O status da tarefa é Executando enquanto HealthOmics está processando a tarefa.

Parando

Depois de concluir o processamento da tarefa e exportar os dados de saída, a tarefa passa para Interromper.

- HealthOmics desprovisiona o nó da tarefa do fluxo de trabalho.

Concluído

HealthOmics concluiu o processamento da tarefa e transferiu os dados de saída para o sistema de arquivos de armazenamento em execução.

Falha

HealthOmics encontrou um erro ao processar a tarefa e não a concluiu.

- A tarefa passa para o status Parando (HealthOmics desprovisiona os recursos) e depois para o status Falha.
- Se o erro for um erro de serviço (código de status HTTP 5XX) e o fluxo de trabalho suportar novas HealthOmics tentativas para essa tarefa, tente processar a tarefa novamente. HealthOmics atribui um novo ID de tarefa à nova tentativa.

Cancelado

HealthOmics interrompe a tarefa após uma solicitação iniciada pelo usuário para cancelar a execução.

- A tarefa passa para o status Interrompido (HealthOmics desprovisiona os recursos) e depois para o status Cancelado.

Solução de problemas de tarefas de fluxo

A seguir estão as melhores práticas e considerações para solucionar problemas em suas tarefas.

- Os registros de tarefas dependem `STDOUT` e são `STDERR` produzidos pela tarefa. Se o aplicativo usado na tarefa não produzir nenhuma delas, não haverá um registro de tarefas. Para ajudar na depuração, use os aplicativos no modo. `verbose`
- Para visualizar os comandos que estão sendo executados em uma tarefa junto com seus valores interpolados, use o comando `set -x` Bash. Isso pode ajudar a determinar se a tarefa está usando as entradas corretas e identificar onde os erros podem ter impedido que a tarefa fosse executada conforme o esperado.
- Use o `echo` comando para enviar os valores das variáveis para `STDOUT` ou`STDERR`. Isso ajuda você a confirmar que eles estão sendo definidos conforme o esperado.
- Use comandos como `ls -l <name_of_input_file>` para confirmar se as entradas estão presentes e têm o tamanho esperado. Se não estiverem, isso pode revelar um problema com uma tarefa anterior produzindo saídas vazias devido a um bug.
- Use o comando `df -Ph . | awk 'NR==2 {print $4}'` em um script de tarefas para determinar o espaço atualmente disponível para a tarefa e ajudar a identificar situações em que talvez seja necessário executar o fluxo de trabalho com alocação adicional de armazenamento.

A inclusão de qualquer um dos comandos anteriores em um script de tarefa pressupõe que o contêiner de tarefas também inclua esses comandos e que eles estejam no ambiente `path` do contêiner.

Execute a otimização para um HealthOmics fluxo de trabalho privado

Você pode otimizar as execuções de acordo com o custo total, o tempo total de execução ou uma combinação de ambos. HealthOmics fornece dados e ferramentas para ajudá-lo a tomar decisões de otimização. A otimização de execução não se aplica aos fluxos de trabalho do Ready2Run, porque você não tem controle sobre como o serviço gerencia o provisionamento de recursos para esses fluxos de trabalho.

A primeira etapa é entender o uso atual dos recursos da tarefa e o custo das tarefas em execução e, em seguida, aplicar métodos para otimizar o custo e o desempenho da execução.

Tópicos

Execute o Analyzer

HealthOmics fornece uma ferramenta de código aberto chamada Run Analyzer. Essa ferramenta extrai informações de uso de recursos em nível de tarefa para uma execução e sugere oportunidades de otimização de custo e desempenho de execução.

> **Note**
>
> O Run Analyzer estima os custos das tarefas e as possíveis economias de custo com base nos preços AWS sugeridos no momento em que você executa a ferramenta. Avalie as recomendações de otimização e implemente aquelas que façam sentido para seus casos de uso. Teste as otimizações que você adota para garantir que elas funcionem para sua corrida.

O Run Analyzer executa as seguintes tarefas:

- Avalia gargalos de memória e computação.

- Identifica tarefas que estão superprovisionadas para memória ou CPU e recomenda novos tamanhos de instância que podem reduzir custos.
- Calcula estimativas de custo para tarefas individuais e calcula a possível economia de custos se você aplicar as recomendações.
- Oferece uma visualização do cronograma das tarefas para que você possa verificar as dependências das tarefas e a sequência de processamento. A linha do tempo também ajuda você a identificar tarefas de longa duração.
- Fornece recomendações sobre o tamanho do sistema de arquivos para o armazenamento em execução.
- Mostra os tempos de provisionamento de tarefas para que você possa identificar áreas em que grandes cargas de contêineres podem estar diminuindo o tempo de provisionamento.
- A ferramenta inclui um parâmetro de entrada (espaço livre) que você pode usar para controlar a agressividade das recomendações de otimização.

As seções a seguir incluem sugestões específicas para usar o Run Analyzer para otimizar as execuções.

Determine os custos de operação

Você pode usar os seguintes métodos e diretrizes para determinar os custos de operação:

- Para ver os custos totais de execução de um período de cobrança, siga estas etapas:

  1. Abra o console Billing and Cost Management e escolha Bills.
  2. Em Cobranças por serviço, expanda Omics.
  3. Expanda a região e, em seguida, visualize o custo de todas as suas execuções discriminadas por tipo de instância omics, tipo de armazenamento de execução e fluxo de trabalho Ready2Run.

- Para gerar um relatório de custo que inclua informações para cada execução, siga estas etapas:

  1. Abra o console Billing and Cost Management e escolha Exportações de dados.
  2. Escolha Criar para criar uma nova exportação de dados.
  3. Insira um nome de exportação para a exportação de dados. Mantenha os outros campos em seus valores padrão para criar um relatório CUR (custo e uso).
  4. Em Granularidade de tempo, selecione por hora ou diariamente.

5. Em Configurações de armazenamento de exportação de dados, execute estas etapas de configuração:

   a. Configure um bucket do Amazon S3 para a exportação de dados.

   b. Em Controle de versão de arquivo, selecione se deseja sobrescrever o arquivo de exportação existente ou criar um novo arquivo a cada vez.

   O sistema gera o primeiro relatório nas próximas 24 horas e gera os relatórios subsequentes uma vez por dia.

6. Para obter mais informações sobre como criar a exportação de dados, consulte Criação de exportações de dados no Guia do usuário de exportação de AWS dados.

- Você pode marcar suas execuções para monitorar e otimizar os custos por categoria, como por equipe ou por projeto. Se você usa tags, siga estas etapas para ver os custos de execução por categoria de tag:

  1. Abra o console Billing and Cost Management e escolha Cost Explorer.
  2. Em Parâmetros do relatório > Agrupar por, escolha Tag como dimensão e selecione o nome do Tag desejado.

- Para ver o uso de recursos para tarefas, veja os logins do manifesto de execução CloudWatch. Para obter mais informações, consulte Monitoramento HealthOmics com CloudWatch registros.
- Use a Execute o Analyzer ferramenta para extrair informações de uso dos recursos da tarefa para uma execução.

Determine o uso do tempo de execução

Você pode usar os métodos a seguir para ajudá-lo a investigar o uso do tempo de execução:

- Na página Execuções do console, você pode ver o tempo total de execução de uma execução.
- Na página de detalhes da execução, você pode visualizar os seguintes itens:
  - Veja o tempo total de execução de uma corrida.
  - Visualize o tempo de execução de cada tarefa na execução.
  - Escolha um dos links para visualizar os registros no Amazon S3 ou para visualizar os registros de execução ou os registros de execução do manifesto. CloudWatch
- Na lista Executar tarefas, escolha o link Exibir registros de uma tarefa para ver os registros da tarefa CloudWatch.

- A resposta à operação da `listRuns` API inclui a hora de início e a hora de parada da execução, para que você possa calcular o tempo total de execução.
- A Execute o Analyzer ferramenta mostra as durações das tarefas em uma visualização da linha do tempo. Essa ferramenta fornece uma representação visual da sequência de processamento da tarefa, que você pode combinar com a ordem esperada.

Métodos para otimizar as execuções

HealthOmics provisiona, gerencia e otimiza automaticamente os recursos que realizam a preparação de dados (como importações e exportações de dados). HealthOmics também inicia e executa o mecanismo de fluxo de trabalho do seu fluxo de trabalho. No entanto, você pode influenciar os horários de início da execução, os horários de início da tarefa e o tempo geral de execução da tarefa definindo várias configurações de execução. Sua abordagem geral para a definição e o design do fluxo de trabalho também afeta o tempo de execução da tarefa. A lista a seguir descreve os fatores que podem afetar o desempenho da execução e da tarefa:

Execute o tipo de armazenamento

O tipo de armazenamento de execução tem um impacto no desempenho da execução e no tempo de provisionamento da execução. O armazenamento de execução dinâmica é provisionado com mais rapidez e nunca fica sem memória, pois ele é dimensionado dinamicamente de acordo com suas necessidades de armazenamento de execução. O armazenamento de execução dinâmica também é uma boa opção para fluxos de trabalho em desenvolvimento, nos quais você geralmente pode iniciar e interromper um fluxo de trabalho para solucionar problemas.

O armazenamento estático de execução exige tempos maiores de provisionamento do sistema de arquivos, mas pode concluir algumas execuções mais rapidamente, normalmente se as execuções tiverem alta simultaneidade de tarefas ou exigirem mais de 9,6 TiB de capacidade do sistema de arquivos. O armazenamento de execução estática é adequado para fluxos de trabalho de longa duração com altos I/O requisitos.

Para ajudá-lo a avaliar o custo versus o desempenho de cada tipo de armazenamento executado para uma determinada execução, você pode experimentar o teste A/B para ver qual tipo de armazenamento de execução oferece melhor desempenho. Além disso, considere usar o armazenamento de execução dinâmico para seus ciclos de desenvolvimento e, em seguida, use o armazenamento de execução estático para execuções de produção em grande escala.

Para obter mais informações sobre como executar tipos de armazenamento Execute tipos de armazenamento em HealthOmics fluxos de trabalho

Provisionar em excesso o armazenamento estático executado

Se o cálculo da tarefa do fluxo de trabalho estiver 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 gargalos, aumentar o tamanho do sistema de arquivos para reduzir o tempo de execução da tarefa pode reduzir o custo geral.

Reduza o tamanho das imagens do contêiner

Quando cada tarefa é iniciada, HealthOmics carrega o contêiner que você especificou para a tarefa. Contêineres maiores demoram mais para serem carregados. Otimize seus contêineres para serem tão pequenos quanto possível para melhorar a eficiência do lançamento de novas tarefas. Se você adicionar grandes conjuntos de dados aos seus contêineres, considere armazená-los no S3 e fazer com que seu fluxo de trabalho importe os dados do S3. Para obter os tamanhos máximos de HealthOmics contêineres que suportam, consulteHealthOmics cotas de tamanho fixo do fluxo de trabalho.

Tamanho da tarefa

Você pode combinar tarefas pequenas e sequenciais em uma única tarefa para economizar tempo de provisionamento de tarefas. Além disso, HealthOmics tem uma taxa mínima de duração de tarefa de um minuto, portanto, a combinação de tarefas pode reduzir os custos. Dentro da tarefa combinada, você poderá usar pipes Unix para evitar o I/O custo de serializar e desserializar arquivos.

Compressão de arquivos

Evite compactar excessivamente os arquivos intermediários do fluxo de trabalho. A maioria dos formatos genômicos usa compressão “gzip” ou “block gzip”. Descompactar o arquivo de entrada da tarefa e recomprimir o arquivo de saída da tarefa pode consumir uma grande porcentagem do uso geral da CPU da tarefa. Alguns aplicativos de genômica permitem que você defina o nível de compressão ao serializar as saídas. Ao reduzir o nível de compactação, você pode reduzir o tempo de CPU, embora arquivos maiores aumentem o tempo gasto gravando no disco. Dependendo da tarefa e do aplicativo, você pode encontrar o nível de compactação ideal para arquivos intermediários que resultam no menor tempo de execução. Recomendamos que você comece segmentando as tarefas com os maiores arquivos de saída. Um nível de compressão de 2 funciona bem em vários cenários. Você pode começar com esse nível para seu caso de uso e comparar os resultados experimentando outros níveis de compressão.

Contagem de tópicos

Se você especificar segmentos em sua definição de tarefa, defina o número de segmentos com o mesmo valor do número de v solicitadosCPUs.

Especifique computação e memória

Se você não especificar recursos de memória ou computação em sua tarefa, HealthOmics atribua o menor tipo de instância (`omics.c.large`) como padrão. Declare explicitamente seus requisitos de memória e computação se quiser HealthOmics atribuir um tipo de instância maior.

HealthOmics aloca o número de recursos vCPUs, memória e GPU que você solicita. Por exemplo, se você solicitar 15v CPUs e 33GiB, HealthOmics aloca uma instância omics.m.4xl (16v, 64GB) para sua tarefaCPUs, mas sua tarefa pode usar somente 15 v e 33GiB. CPUs Portanto, recomendamos que você solicite v CPUs e recursos de memória que correspondam a uma instância omics.

Batch várias amostras em uma única execução

Como o provisionamento do sistema de arquivos leva tempo no início da execução, você pode economizar tempo de provisionamento agrupando várias amostras na mesma execução. Considere os seguintes fatores antes de decidir sobre essa abordagem:

- Uma única amostra incorreta pode fazer com que um fluxo de trabalho falhe, portanto, o agrupamento de amostras em lote pode aumentar o número de fluxos de trabalho com falha. Se você não tem certeza de que seu fluxo de trabalho será bem-sucedido na maioria das vezes, uma execução por amostra pode ser uma abordagem melhor.
- HealthOmics aloca um sistema de arquivos de armazenamento executado para todo o fluxo de trabalho. Para um lote de amostras, certifique-se de especificar uma quantidade grande o suficiente de armazenamento em execução para processar todas as amostras.
- Há uma quantidade máxima de armazenamento de execução por fluxo de trabalho, o que pode restringir o número de amostras que você pode adicionar ao lote.
- O tamanho mínimo de armazenamento de execução é de 1,2 TiB, portanto, o agrupamento em lotes pode reduzir os custos se o fluxo de trabalho usar muito menos armazenamento do que o mínimo para cada amostra.
- O armazenamento de execução pode lidar com várias conexões simultâneas, portanto, ter várias tarefas usando o mesmo armazenamento de execução não deve causar I/O gargalos.
- Cada execução tem seu próprio conjunto de tags. Se você marcar fluxos de trabalho com informações para orçamento ou acompanhamento, talvez seja melhor usar execuções separadas.

- As funções do IAM se aplicam a toda a execução. Cada usuário tem acesso a todos os dados de um lote de amostras. Ter fluxos de trabalho separados permite que você use permissões mais refinadas.
- HealthOmics define cotas em nível de conta para o número máximo de fluxos de trabalho simultâneos e o número máximo de tarefas simultâneas em um fluxo de trabalho. Para obter informações sobre como solicitar um aumento para essas cotas, consulteHealthOmics cotas de serviço.

Use parâmetros para imagens de contêiner

Parametrize suas imagens de contêiner em vez de incorporá-las ao fluxo de trabalho URIs . Quando são parâmetros de execução, HealthOmics valida se a execução tem acesso aos seus contêineres antes do início da execução. Caso contrário, a tarefa falhará durante a execução, quando você tiver incorrido em cobranças por qualquer tarefa concluída. Além disso, como essas são entradas parametrizadas, HealthOmics gera uma soma de verificação no manifesto de execução, o que melhora a proveniência da execução.

Use um linter

Use um linter para encontrar erros comuns do fluxo de trabalho antes de executar um novo fluxo de trabalho. Para obter mais informações, consulte Impressoras de fluxo de trabalho em HealthOmics.

Use EventBridge para sinalizar problemas

Use alertas EventBridge personalizados para capturar anomalias específicas da sua lógica de negócios.

Use armazenamentos de sequências

Considere usar um armazenamento de sequências para seus dados de origem para economizar nos custos de armazenamento. Para obter mais informações, consulte Armazenar dados ômicos de maneira econômica em qualquer escala com a postagem do HealthOmics blog.

Impacto da variação do tamanho do arquivo entre as execuções

Os usuários geralmente projetam e testam execuções usando um pequeno conjunto de dados de teste e, em seguida, encontram uma grande variedade de dados com variação significativa no tamanho do arquivo nas execuções de produção. Certifique-se de considerar essa variação ao otimizar a execução.

A lista a seguir descreve recomendações para otimização quando há uma variação significativa nos tamanhos dos arquivos:

Varie os tamanhos dos arquivos em seus dados de teste

Tente usar dados de teste durante o desenvolvimento que tenham uma quantidade representativa de variação.

Use o Run Analyzer

Use a ferramenta Run Analyzer em uma variedade de amostras para considerar a variação nos tamanhos dos dados.

Você pode usar o analisador de execução para entender a variação entre as execuções em suas amostras de dados de produção. Use o `--batch` modo no Run Analyzer para gerar estatísticas para um lote de execuções e analisar o máximo de recursos computacionais necessários para lidar com valores discrepantes em seus conjuntos de dados.

Por exemplo, você pode fornecer ao Run Analyzer uma célula de fluxo completo de dados no modo batch para entender o pico de utilização da vCPU e da memória para a célula de fluxo completa.

Reduza a variação de tamanho dos conjuntos de dados de entrada

Se você observar uma grande variação nos tamanhos das amostras, poderá bifurcar as amostras antes HealthOmics e selecionar diferentes tamanhos de sistema de arquivos para cada lote, a fim de economizar nos custos de armazenamento em execução.

Na WDL, use a `size` função para bifurcar a alocação de recursos para tarefas individuais para amostras grandes e pequenas. Aplique essa estratégia às suas tarefas mais caras para ter o maior impacto.

No Nextflow, use recursos condicionais para hierarquizar a alocação de recursos com base no tamanho ou nome do arquivo. Para obter mais informações, consulte Recursos de processos condicionais no site Nextflow GitHub .

Não otimize muito cedo

Finalize o código e a lógica do seu fluxo de trabalho antes de investir em esforços significativos de ajuste de desempenho. Alterar seu código pode ter impactos significativos nos recursos necessários. Se você otimizar uma execução muito cedo no processo de desenvolvimento,

poderá otimizar demais ou talvez precise otimizar novamente se a definição do fluxo de trabalho mudar posteriormente.

Execute novamente a ferramenta Run Analyzer periodicamente

Se você fizer alterações na definição do fluxo de trabalho ao longo do tempo ou se a variação da amostra mudar, execute periodicamente a ferramenta Run Analyzer para ajudá-lo a fazer otimizações adicionais.

Métodos para otimizar a simultaneidade de recursos

HealthOmics fornece os seguintes recursos para ajudá-lo a controlar e gerenciar custos quando o processamento é executado em grande escala:

- Use grupos de execução para controlar seus custos e o uso de recursos. Você pode definir valores máximos no grupo de execução para o número de execuções simultâneas CPUs GPUs, v e tempo total de execução por tarefa. Se equipes ou grupos diferentes usarem a mesma conta, você poderá criar um grupo de corrida separado para cada equipe. Você pode controlar o uso de recursos e os custos por equipe e configurar os valores máximos do grupo de execução. Para obter mais informações, consulte Usando grupos de HealthOmics execução.
- Durante o desenvolvimento, você pode configurar um grupo de execução separado com valores máximos mais baixos para capturar tarefas descontroladas.
- As Cotas de Serviço também ajudam a proteger sua conta contra solicitações excessivas de recursos. Para obter informações sobre Cotas de Serviço, incluindo como solicitar aumentos no valor da cota, consulte HealthOmics cotas de serviço

Execute operações em HealthOmics

Você pode iniciar, executar novamente, clonar, cancelar ou excluir uma execução:

- Start— HealthOmics cria uma nova execução usando as configurações especificadas e, em seguida, inicia a execução.
- Rerun— HealthOmics cria uma nova execução que é uma duplicata da execução especificada. Você pode executar novamente uma execução excluída usando a HealthOmics rerun ferramenta.
- Clone— Você pode clonar uma execução existente usando o console. O console abre a página de execução do clone e preenche previamente os campos de configuração usando os valores da

execução existente. Você pode modificar os valores conforme necessário e iniciar a execução clonada.

- Cancel— Você pode cancelar uma corrida que ainda não foi concluída. Quando você cancela uma execução, HealthOmics não salva nenhuma das saídas da execução.
- Delete— Você pode excluir as execuções concluídas manualmente ou definir o modo de retenção de execuções HealthOmics para excluir automaticamente as execuções mais antigas. Para obter mais informações sobre o modo de retenção, consultethe section called “Execute os modos de retenção”.

Tópicos

Comece uma corrida em HealthOmics

Ao iniciar uma execução, você especifica os recursos que são HealthOmics alocados para uso durante a execução.

Especifique o tipo de armazenamento executado e a quantidade de armazenamento (para armazenamento estático). Para garantir o isolamento e a segurança dos dados, HealthOmics provisiona o armazenamento no início de cada execução e o desprovisiona no final da execução. Para obter informações adicionais, consulte Execute tipos de armazenamento em HealthOmics fluxos de trabalho.

Especifique uma localização do Amazon S3 para os arquivos de saída. Se você executa um grande volume de fluxos de trabalho simultaneamente, use uma URIs saída separada do Amazon S3 para cada fluxo de trabalho para evitar a limitação do bucket. Para obter mais informações, consulte Organização de objetos usando prefixos no Guia do usuário do Amazon S3 e Dimensione conexões de armazenamento horizontalmente no whitepaper Otimizando o desempenho do Amazon S3.

Você também pode especificar a prioridade de execução. O impacto da prioridade na execução depende de a execução estar associada a um grupo de execução. Para obter informações adicionais, consulte Prioridade de execução.

Se um fluxo de trabalho tiver uma ou mais versões, você poderá especificar uma versão ao iniciar a execução. Se você não especificar uma versão, HealthOmics iniciará a versão padrão do fluxo de trabalho.

Ao usar a HealthOmics API, você pode fornecer um ID de solicitação exclusivo para cada execução. O ID da solicitação é um token de idempotência HealthOmics usado para identificar solicitações duplicadas e inicia a execução apenas uma vez.

> **Note**
>
> Você especifica uma função de serviço do IAM ao iniciar uma execução. Opcionalmente, o console pode criar a função de serviço para você. Para obter mais informações, consulte Funções de serviço para AWS HealthOmics.

Tópicos

HealthOmics parâmetros de execução

Ao iniciar uma execução, você especifica as entradas de execução no arquivo JSON dos parâmetros de execução ou pode inserir os valores dos parâmetros em linha. Para obter informações sobre como gerenciar o tamanho do arquivo JSON de parâmetros de execução, consulteGerenciando o tamanho dos parâmetros de execução.

HealthOmics suporta os seguintes tipos de JSON para valores de parâmetros.

Tipo JSON	Exemplo de chave e valor	Observações
boolean	“b”: verdadeiro	O valor não está entre aspas e está todo em minúsculas.
integer	“i” :7	O valor não está entre aspas.
número	“f”: 42,3	O valor não está entre aspas.

Tipo JSON	Exemplo de chave e valor	Observações
string	“s”: “caracteres”	O valor está entre aspas. Use o tipo de string para valores de texto URIs e. O destino do URI deve ser o tipo de entrada esperado.
array	“a”: [1,2,3]	O valor não está entre aspas. Cada membro da matriz deve ter o tipo definido pelo parâmetro de entrada.
objeto	“o”: {"left” :"a”, “right” :1}	Na WDL, o objeto é mapeado para WDL Pair, Map ou Struct

Iniciando uma execução usando o console

Para começar uma corrida

1. Abra o console de HealthOmics .
2. Se necessário, abra o painel de navegação esquerdo (≡). Selecione Execuções.
3. Na página Execuções, escolha Iniciar execução.
4. No painel Detalhes da execução, forneça as seguintes informações

   - Fonte do fluxo de trabalho - escolha Fluxo de trabalho próprio ou Fluxo de trabalho compartilhado.
   - ID do fluxo de trabalho - O ID do fluxo de trabalho associado a essa execução.
   - Versão do fluxo de trabalho (opcional) - selecione uma versão do fluxo de trabalho para usar nessa execução. Se você não selecionar uma versão, a execução usará a versão padrão do fluxo de trabalho.
   - Nome da execução - Um nome distinto para essa execução.
   - Prioridade de execução (opcional) - A prioridade dessa execução. Números mais altos especificam uma prioridade mais alta, e as tarefas de maior prioridade são executadas primeiro.

- Executar tipo de armazenamento - especifique o tipo de armazenamento aqui para substituir o tipo de armazenamento de execução padrão especificado para o fluxo de trabalho. O armazenamento estático aloca uma quantidade fixa de armazenamento para a execução. O armazenamento dinâmico aumenta e diminui conforme necessário para cada tarefa em execução.
- Capacidade de armazenamento em execução - Para armazenamento em execução estática, especifique a quantidade de armazenamento necessária para a execução. Essa entrada substitui a quantidade padrão de armazenamento de execução especificada para o fluxo de trabalho.
- Selecione o destino de saída do S3 - O local do S3 onde as saídas de execução serão salvas.
- ID da conta do proprietário do bucket de saída (opcional) — se sua conta não for proprietária do bucket de saída, insira o Conta da AWS ID do proprietário do bucket. Essas informações são necessárias para que HealthOmics possamos verificar a propriedade do bucket.
- Modo de retenção de metadados de execução: escolha se deseja reter os metadados de todas as execuções ou fazer com que o sistema remova os metadados de execução mais antigos quando sua conta atingir o número máximo de execuções. Para obter mais informações, consulte Execute o modo de retenção para HealthOmics execuções.

5. Em Função de serviço, você pode usar uma função de serviço existente ou criar uma nova.
6. (Opcional) Para tags, você pode atribuir até 50 tags à execução.
7. Escolha Próximo.
8. Na página Adicionar valores de parâmetros, forneça os parâmetros de execução. Você pode fazer upload de um arquivo JSON que especifique os parâmetros ou inserir manualmente os valores.
9. Escolha Próximo.
10. No painel Grupo de execução, você pode, opcionalmente, especificar um grupo de execução para essa execução. Para obter mais informações, consulte Usando grupos de HealthOmics execução.
11. No painel Executar cache, você pode, opcionalmente, especificar um cache de execução para essa execução. Para obter mais informações, consulte Configurando uma execução com cache de execução usando o console.
12. Escolha Review and start run (Examinar e iniciar execução).
13. Depois de revisar a configuração de execução, escolha Iniciar execução.

Iniciando uma execução usando a API

Use a operação da API start-run para criar e iniciar uma execução.

O exemplo a seguir especifica a ID do fluxo de trabalho e a função de serviço. Este exemplo define o modo de retenção como`REMOVE`. Para obter mais informações sobre o modo de retenção, consulte[Execute o modo de retenção para HealthOmics execuções](#).

```
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
```

Em resposta, você obtém a seguinte saída. `uuid`É exclusivo para a execução e, junto com ele, `outputUri` pode ser usado para rastrear onde os dados de saída são gravados.

```
{
    "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"
}
```

Incluir um arquivo de parâmetros

Se o modelo de parâmetro de um fluxo de trabalho declarar quaisquer parâmetros necessários, você poderá fornecer um arquivo JSON local das entradas ao iniciar a execução de um fluxo de trabalho. O arquivo JSON contém o nome exato de cada parâmetro de entrada e um valor para o parâmetro.

Faça referência ao arquivo JSON de entrada no AWS CLI adicionando `--parameters file://<input_file.json>` à sua `start-run` solicitação. Para obter mais informações sobre os parâmetros de execução, consulte[HealthOmics entradas de execução](#).

Forneça um ID de solicitação

Você pode fornecer um item exclusivo `requestId` para cada corrida. O ID da solicitação é um token de idempotência usado HealthOmics para capturar solicitações duplicadas. Ela não iniciará uma execução se o ID da solicitação for uma duplicata de uma execução anterior.

Se você usa infraestrutura (como funções Lambda ou funções de etapas) para orquestrar o início da execução, a melhor prática é fornecer um ID de solicitação exclusivo para cada solicitação. StartRun Isso garante que, se sua infraestrutura inadvertidamente iniciar uma execução já iniciada, HealthOmics não inicie a execução duplicada. Por exemplo, se a infraestrutura estiver tentando se recuperar de um erro inicial, ela poderá executar novamente um script que tente iniciar execuções que sejam solicitações duplicadas.

Escolha uma versão do fluxo de trabalho

Você pode especificar uma versão do fluxo de trabalho para a execução. Se você não especificar uma versão, HealthOmics iniciará a execução com a versão padrão do fluxo de trabalho.

```
aws omics start-run
     --workflow-id workflow id \
      ...
     --workflow-version-name '1.2.1'
```

Substituir o tipo de armazenamento de execução

Você pode substituir o tipo de armazenamento de execução padrão que foi definido no fluxo de trabalho.

```
aws omics start-run
       --workflow-id workflow id \
        ...
       --storage-type STATIC
       --storage-capacity 2400
```

Execute um fluxo de trabalho de GPU

Você também pode especificar uma ID de fluxo de trabalho da GPU, conforme mostrado no exemplo a seguir:

```
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
```

Obtenha informações sobre uma corrida

Você pode usar o ID na resposta com a API get-run para verificar o status de uma execução, conforme mostrado.

```
aws omics get-run --id run id
```

A resposta dessa operação de API informa o status da execução do fluxo de trabalho. Os status possíveis são `PENDINGSTARTING`,`RUNNING`, e. `COMPLETED` Quando uma execução é executada`COMPLETED`, você pode encontrar um arquivo de saída chamado `outfile.txt` em seu bucket de saída do Amazon S3, em uma pasta com o nome do ID de execução.

A operação da API get-run também retorna outros detalhes, como se o fluxo de trabalho é `Ready2Run` ou`PRIVATE`, o mecanismo do fluxo de trabalho e os detalhes do acelerador. O exemplo a seguir mostra a resposta para get-run para uma execução de um fluxo de trabalho privado, descrita na WDL com um acelerador de GPU e sem tags atribuídas à execução.

```
{
    "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": {}
}
```

Você pode ver o status de todas as execuções com a operação da API list-runs, conforme mostrado.

```
aws omics list-runs
```

Para ver todas as tarefas concluídas em uma execução específica, use a list-run-tasksAPI.

```
aws omics list-run-tasks --id task ID
```

Para obter os detalhes de qualquer tarefa específica, use a get-run-task API.

```
aws omics get-run-task --id <run_id> --task-id task ID
```

Após a conclusão da execução, os metadados são enviados para o CloudWatch subfluxo. **manifest/run/<run ID>/<run UUID>**

Veja a seguir um exemplo do manifesto.

```
{
    "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"
  },
  ...
```


Os metadados de execução não são excluídos se não estiverem presentes nos CloudWatch registros.

Execute novamente uma corrida em HealthOmics

Para execuções que você ainda não excluiu, use o console ou a API para executar novamente a execução. Para execuções que você excluiu, use a ferramenta. HealthOmics rerun

Tópicos

- Execute novamente uma execução usando o console

- Execute novamente uma execução usando a API
- Usando a ferramenta Rerun

Execute novamente uma execução usando o console

No console, siga estas etapas para executar novamente uma execução:

1. Abra o console de HealthOmics .
2. Se necessário, abra o painel de navegação esquerdo (≡). Selecione Execuções.
3. Na página Execuções, selecione a execução a ser executada novamente.
4. No menu de ação acima da tabela, escolha Executar novamente.

Execute novamente uma execução usando a API

Use a operação StartRun da API para executar novamente uma execução existente. Forneça as seguintes entradas necessárias:

- Um ARN de função de serviço ()`roleArn`.
- O ID da execução para duplicar (`runId`).
- Um local do Amazon S3 onde a execução salva as saídas da execução (). `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
```

Usando a ferramenta Rerun

Para uma execução excluída, você pode baixar e usar a HealthOmics rerun ferramenta para executar novamente a execução. A ferramenta recupera as informações de execução do manifesto do CloudWatch Logs. Baixe a rerun ferramenta no GitHub repositório HealthOmics de ferramentas.

O exemplo a seguir mostra como usar a rerun ferramenta.

```
aws-healthomics-rerun 9876543
```

Se a execução existir em CloudWatch, você receberá uma resposta semelhante ao exemplo de saída a seguir. Se o fluxo de trabalho não existir mais, você receberá uma mensagem de erro.

```
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": {}
}
```

Clonar uma execução em HealthOmics

Você pode clonar uma execução existente usando o HealthOmics console. A clonagem cria uma nova execução usando os valores de configuração da execução clonada. Você pode modificar esses valores padrão e adicionar outras entradas opcionais.

1. Abra o console de HealthOmics .

2. Se necessário, abra o painel de navegação esquerdo (≡). Selecione Execuções.
3. Na página Execuções, selecione a execução a ser clonada.
4. No menu de ação acima da tabela, escolha Clone run. O console abre o formulário de execução do Clone. O formulário é idêntico ao Start run, exceto que o console preenche o formulário com todos os valores relevantes da execução clonada.

   O console cria uma nova ID de execução para o clone de execução e adiciona essa ID de execução como sufixo ao nome da execução.

   Ao percorrer as páginas do formulário, você pode ajustar os valores de configuração conforme necessário.
5. Depois de revisar a configuração de execução, escolha Iniciar execução.

Cancelar uma corrida HealthOmics

Você pode cancelar uma execução se seu status for `PENDINGSTARTING`,`RUNNING`, ou`STOPPING`.

> **Note**
>
> Quando você cancela uma execução, HealthOmics não salva nenhuma das saídas da execução.

No console, siga estas etapas para cancelar uma execução:

1. Abra o console de HealthOmics .
2. Se necessário, abra o painel de navegação esquerdo (≡). Selecione Execuções.
3. Na página Execuções, escolha a execução a ser cancelada.
4. O console abre a página de detalhes da execução. No banner de status na parte superior da página, escolha Parar de executar.
5. Digite confirm para interromper a execução.

Para cancelar uma execução usando a API, use a operação CancelRun da API.

O exemplo a seguir mostra como cancelar uma execução usando AWS CLI o. Para executar o exemplo, *`run id`* substitua o pelo ID da execução que você gostaria de cancelar. Se for bem-sucedido, não haverá resposta.

```
aws omics cancel-run --id run id
```

Excluir uma execução em HealthOmics

Quando não precisar mais de uma execução, você pode excluí-la usando AWS CLI a API ou o console. Você pode excluir uma execução quando seu status for `COMPLETED` ou`CANCELED`.

No console, siga estas etapas para excluir uma execução:

1. Abra o console de HealthOmics .
2. Se necessário, abra o painel de navegação esquerdo (≡). Selecione Execuções.
3. Na página Execuções, selecione uma ou mais execuções para excluir.
4. No menu de ação acima da tabela, escolha Excluir.
5. No formato modal, digite confirm para confirmar a exclusão.

O AWS CLI comando a seguir exclui uma execução. Para executar o exemplo, *run id* substitua o pelo ID da execução que você deseja excluir. Não haverá resposta se a execução for excluída com sucesso.

```
aws omics delete-run --id run id
```

Usando grupos de HealthOmics execução

Opcionalmente, você pode criar um grupo de execuções para limitar os recursos computacionais das execuções que você adiciona ao grupo. Grupos organizados podem ajudar você a:

- Coloque suas corridas na fila para que você não exceda os limites de serviço.
- Identifique tarefas desnecessárias definindo uma duração máxima de execução.
- Gerencie a prioridade de cada corrida para que as mais importantes sejam concluídas primeiro.

Se você definir o máximo de vCPU, GPU ou execuções simultâneas, as tarefas de execução entrarão na fila quando o máximo for atingido. Se você definir uma duração máxima de execução, a execução falhará se exceder a duração máxima.

Use a configuração de prioridade de execução para estabelecer a prioridade em um grupo de execução.

Os limites do serviço têm precedência sobre os limites do grupo de execução. Por exemplo, se você definir o máximo de um grupo de execução com um valor maior do que o máximo de serviço em uma região, HealthOmics aplica o máximo de serviço.

Tópicos

Prioridade de execução

Você pode usar a prioridade de execução para estabelecer a prioridade das execuções em um grupo de execução.

Se várias execuções tiverem a mesma prioridade, a execução que começou primeiro terá a prioridade mais alta.

Você também pode definir uma prioridade para uma corrida que não esteja em um grupo de corrida. A prioridade é comparada com as prioridades de todas as outras corridas que não estão em um grupo de corridas

Você define a prioridade da execução ao iniciar a execução. Para obter mais informações, consulte Comece uma corrida em HealthOmics.

Crie um grupo de execução usando o console

Para criar um grupo de execução

1. Abra o console de HealthOmics .
2. Se necessário, abra o painel de navegação esquerdo (≡). Escolha Executar grupos.
3. Na página Executar grupos, escolha Criar grupo de execução.
4. Na página Criar detalhes do grupo de execução, forneça as seguintes informações

    - Nome do grupo de execução - Um nome exclusivo para esse grupo de execução.

- Máximo de vCPU para execuções simultâneas - O número máximo de v CPUs que pode ser executado simultaneamente em todas as execuções ativas no grupo de execução.
- Máximo GPUs - O número máximo GPUs que pode ser executado simultaneamente em todas as execuções ativas no grupo de execução.
- Tempo máximo de execução (minutos) por corrida - O tempo máximo para cada corrida (em minutos). Se uma execução exceder o tempo máximo de execução, a execução falhará automaticamente.
- Máximo de execuções simultâneas - O número máximo de execuções que podem ser executadas ao mesmo tempo.

5. (opcional) Você pode adicionar até 50 tags ao grupo de execução.
6. Escolha Criar grupo de execução.

Crie um grupo de execução usando a CLI

Para criar um grupo de execução, use a operação de create-run-groupAPI para criar um grupo de execução chamado`TestRunGroup`. O exemplo a seguir define um máximo de 20 CPUs, 10 GPUs, 5 execuções e uma duração máxima de execução de 600 minutos.

```
aws omics create-run-group --name TestRunGroup \
--max-cpus 20 \
--max-gpus 10 \
--max-duration 600 \
--max-runs 5
```

A resposta dessa operação de API inclui o ID do recém-criado`RunGroup`.

```
{
    "arn": "arn:aws:omics:us-west-2:12345678901:runGroup/2839621",
    "id": "2839621",
    "tags": {}
}
```

Para obter informações adicionais sobre o grupo de execução, use esse ID com a operação da get-run-groupAPI, conforme mostrado no exemplo a seguir.

```
aws omics get-run-group --id run group id
```

A resposta inclui as configurações de limite para o grupo de execução e as tags atribuídas.

```
{
    "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
}
```

Você também pode usar a operação da list-run-groupAPI para visualizar todos os grupos de execução criados.

```
aws omics list-run-groups
```

Excluir um grupo de execução usando o console

Você pode excluir um grupo de execução se não houver execuções associadas a esse grupo de execução com o status de`PENDING`,`STARTING`,`RUNNING`, ou`STOPPING`.

Para excluir um grupo de execução, siga estas etapas.

1. Abra o console de HealthOmics .
2. Se necessário, abra o painel de navegação esquerdo (≡). Escolha Executar grupos.
3. Na página Executar grupos, escolha o grupo de execução a ser excluído e escolha Excluir no xx.

Excluir um grupo de execução usando a CLI

Você pode excluir um grupo de execução se não houver execuções associadas a esse grupo de execução com o status de`PENDING`,`STARTING`,`RUNNING`, ou`STOPPING`.

O exemplo a seguir mostra como você pode usar o AWS CLI para excluir um grupo de execução. Você não receberá uma resposta. Para executar o exemplo, *`run group id`* substitua o pelo ID do grupo de execução que você deseja excluir.

```
aws omics delete-run-group --id run group id
```

Cache de chamadas para execuções HealthOmics

AWS HealthOmics suporta cache de chamadas, também conhecido como currículo, para fluxos de trabalho privados. O cache de chamadas salva as saídas das tarefas concluídas do fluxo de trabalho após o término da execução. As execuções subsequentes podem usar as saídas da tarefa do cache, em vez de computar as saídas da tarefa novamente. O cache de chamadas reduz o uso de recursos computacionais, o que resulta em menores durações de execução e economia de custos computacionais.

Você pode acessar os arquivos de saída da tarefa em cache após a conclusão da execução. Para executar a depuração e a solução de problemas avançadas de tarefas, você pode armazenar em cache arquivos de tarefas intermediárias especificando esses arquivos como saídas de tarefas na definição do fluxo de trabalho.

Você pode usar o cache de chamadas para salvar os resultados da tarefa concluída de execuções com falha. A próxima execução começa com a última tarefa concluída com êxito, em vez de computar as tarefas concluídas novamente.

Se HealthOmics não encontrar uma entrada de cache correspondente para uma tarefa, a execução não falhará. HealthOmics recalcula a tarefa e suas tarefas dependentes.

Para obter informações sobre como solucionar problemas de cache de chamadas, consulteSolução de problemas de cache de chamadas.

Tópicos

Como funciona o cache de chamadas

Para usar o cache de chamadas, você cria um cache de execução e o configura para ter um local Amazon S3 associado aos dados em cache. Ao iniciar uma execução, você especifica o cache de execução. Um cache de execução não é dedicado a um fluxo de trabalho. As execuções de vários fluxos de trabalho podem usar o mesmo cache.

Durante a fase de exportação de uma execução, o sistema exporta as saídas concluídas da tarefa para o local do Amazon S3. Para exportar arquivos de tarefas intermediárias, declare esses arquivos como saídas de tarefas na definição do fluxo de trabalho. O cache de chamadas também salva internamente os metadados e cria hashes exclusivos para cada entrada do cache.

Para cada tarefa em execução, o mecanismo de fluxo de trabalho detecta se há uma entrada de cache correspondente para essa tarefa. Se não houver nenhuma entrada de cache correspondente, HealthOmics calcula a tarefa. Se houver uma entrada de cache correspondente, o mecanismo recuperará os resultados em cache.

Para combinar as entradas do cache, HealthOmics usa o mecanismo de hash incluído nos mecanismos de fluxo de trabalho nativos. HealthOmicsestende essas implementações de hash existentes para contabilizar HealthOmics variáveis, como S3 ETags e resumos de contêineres ECR.

HealthOmics oferece suporte ao cache de chamadas para essas versões de linguagem do fluxo de trabalho:

- Versões WDL 1.0, 1.1 e a versão de desenvolvimento
- Nextflow versões 23.10 e 24.10
- Todas as versões do CWL

> **Note**
>
> HealthOmics não oferece suporte ao cache de chamadas para fluxos de trabalho do Ready2Run.

Tópicos

- Eventos de retenção e invalidação de dados em cache

Modelo de responsabilidade compartilhada

Há uma responsabilidade compartilhada entre os usuários de determinar AWS se tarefas e execuções são boas candidatas para o armazenamento em cache de chamadas. O cache de chamadas alcança os melhores resultados quando todas as tarefas são idempotentes (execuções repetidas de uma tarefa usando as mesmas entradas produzem os mesmos resultados).

No entanto, se uma tarefa incluir elementos não determinísticos (como gerações de números aleatórios ou hora do sistema), execuções repetidas da tarefa usando as mesmas entradas podem resultar em saídas diferentes. Isso pode afetar a eficácia do armazenamento em cache de chamadas das seguintes formas:

- Se HealthOmics usar uma entrada de cache (criada por uma execução anterior) que não seja idêntica à saída que a execução da tarefa produziria para a execução atual, a execução poderá gerar resultados diferentes da mesma execução sem armazenamento em cache.
- HealthOmics pode não encontrar uma entrada de cache correspondente para uma tarefa que deva corresponder, devido às saídas de tarefas não determinísticas. Se não encontrar a entrada de cache válida, a execução recomputará desnecessariamente a tarefa, o que reduz os benefícios de economia de custos do uso do cache de chamadas.

A seguir estão os comportamentos de tarefas conhecidos que podem causar resultados não determinísticos que afetam os resultados do cache de chamadas:

- Usando geradores de números aleatórios.
- Dependência da hora do sistema.
- Usando a concorrência (condições de corrida podem causar variação na saída).
- Buscar arquivos locais ou remotos além do especificado nos parâmetros de entrada da tarefa.

Para outros cenários que podem causar comportamento não determinístico, consulte Entradas de processo não determinísticas no site de documentação do Nextflow.

Se você suspeitar que uma tarefa produz resultados não determinísticos, considere usar os recursos do mecanismo de fluxo de trabalho para evitar o armazenamento em cache de tarefas específicas que não sejam determinísticas. Para obter instruções sobre como desativar o armazenamento em

cache para tarefas individuais em cada linguagem de fluxo de trabalho compatível, consulteRecursos de cache específicos do mecanismo.

Recomendamos que você analise minuciosamente seus requisitos específicos de fluxo de trabalho e tarefas antes de ativar o cache de chamadas em qualquer ambiente em que o cache de chamadas ineficaz ou saídas diferentes das esperadas possam apresentar riscos. Por exemplo, as possíveis limitações do armazenamento em cache de chamadas devem ser cuidadosamente consideradas para determinar se o cache de chamadas é apropriado para casos de uso clínico.

Requisitos de armazenamento em cache para tarefas

HealthOmics armazena em cache as saídas de tarefas que atendem aos seguintes requisitos:

- A tarefa deve definir um contêiner. HealthOmics não armazenará em cache as saídas de uma tarefa sem contêiner.
- A tarefa deve produzir uma ou mais saídas. Você especifica as saídas da tarefa na definição do fluxo de trabalho.
- A definição do fluxo de trabalho não deve usar valores dinâmicos. Por exemplo, se você passar um parâmetro para uma tarefa com um valor que aumenta a cada execução, HealthOmics não armazena em cache as saídas da tarefa.

> **Note**
>
> Se várias tarefas em uma execução usarem a mesma imagem de contêiner, HealthOmics fornecerá a mesma versão de imagem para todas essas tarefas. Depois de HealthOmics extrair a imagem, ela ignora todas as atualizações na imagem do contêiner durante a execução. Essa abordagem fornece uma experiência previsível e consistente e evita possíveis problemas que possam surgir de atualizações na imagem do contêiner que são implantadas no meio da execução.

Execute o desempenho do cache

Ao ativar o cache de chamadas para uma execução, você pode observar os seguintes impactos no desempenho da execução:

- Durante a primeira execução, HealthOmics salva os dados do cache para as tarefas na execução. Você pode ter tempos de exportação mais longos para essa execução, porque o cache de chamadas aumenta a quantidade de dados de exportação.
- Em execuções subsequentes, ao retomar uma execução do cache, isso pode reduzir o número de etapas de processamento e reduzir o tempo de execução.
- Se você também optar por declarar arquivos intermediários como saídas, o tempo de exportação poderá ser ainda maior, pois esses dados podem ser mais detalhados.

Eventos de retenção e invalidação de dados em cache

O objetivo principal de um cache de execução é otimizar o cálculo das tarefas na execução. Se houver uma entrada de cache correspondente válida para uma tarefa, HealthOmics use a entrada de cache em vez de recalcular a tarefa. Caso contrário, HealthOmics reverte para o comportamento padrão do serviço, que é recalcular a tarefa e suas tarefas dependentes. Ao usar essa abordagem, falhas de cache não fazem com que a execução falhe.

Recomendamos que você gerencie o tamanho do cache de execução. Com o tempo, as entradas de cache podem não ser mais válidas devido às atualizações do mecanismo de fluxo de trabalho ou do HealthOmics serviço ou devido às alterações feitas na execução ou nas tarefas de execução. As seções a seguir fornecem detalhes adicionais.

Tópicos

Atualizações de versões do manifesto e atualização de dados

Periodicamente, o HealthOmics serviço pode introduzir novos recursos ou atualizações do mecanismo de fluxo de trabalho que invalidam algumas ou todas as entradas de cache de execução. Nessa situação, suas execuções podem sofrer uma única perda de cache.

HealthOmics cria um arquivo de manifesto JSON para cada entrada de cache. Para execuções iniciadas após 12 de fevereiro de 2025, o arquivo de manifesto inclui um parâmetro de versão. Se uma atualização de serviço invalidar qualquer entrada de cache, HealthOmics incrementa o número da versão para que você possa identificar as entradas de cache herdadas para remoção.

O exemplo a seguir mostra um arquivo de manifesto com a versão definida como 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 execuções com entradas de cache que não são mais válidas, reconstrua o cache para criar novas entradas válidas. Execute as seguintes etapas para cada execução:

1. Inicie a execução uma vez com a retenção de cache definida como CACHE ALWAYS. Essa execução cria as novas entradas de cache.
2. Para execuções subsequentes, defina a retenção do cache para a configuração anterior (CACHE ALWAYS ou CACHE ON FAILURE).

Para limpar as entradas de cache que não são mais válidas, você pode excluir essas entradas de cache do bucket cache do Amazon S3. HealthOmics nunca reutiliza essas entradas de cache. Se você optar por reter entradas que não são válidas, não haverá impacto em suas corridas.

> **Note**
>
> O cache de chamadas salva os dados de saída da tarefa no local do Amazon S3 especificado para o cache, o que gera cobranças para você. Conta da AWS

Comportamento do cache de execução

Você pode definir o comportamento do cache de execução para salvar as saídas da tarefa para execuções que falham (cache em caso de falha) ou para todas as execuções (cache sempre). Ao criar um cache de execução, você define o comportamento padrão do cache para todas as execuções que usam esse cache. Você pode substituir o comportamento padrão ao iniciar uma execução.

Cache on failureé útil se você estiver depurando um fluxo de trabalho que falha após a conclusão bem-sucedida de várias tarefas. A execução subsequente é retomada a partir da última tarefa concluída com êxito se todas as variáveis exclusivas consideradas pelo hash forem idênticas à execução anterior.

Cache alwaysé útil se você estiver atualizando uma tarefa em um fluxo de trabalho concluído com êxito. Recomendamos que você siga estas etapas:

1. Crie uma nova corrida. Defina o comportamento do Cache como Cache sempre e inicie a execução.
2. Depois que a execução for concluída, atualize a tarefa no fluxo de trabalho e inicie uma nova execução com o comportamento definido Cache always. Essa execução processa a tarefa atualizada e todas as tarefas subsequentes que dependam da tarefa atualizada. Todas as outras tarefas usam os resultados em cache.
3. Repita a etapa 2 conforme necessário, até que o desenvolvimento da tarefa atualizada esteja concluído.
4. Use a tarefa atualizada conforme necessário em execuções futuras. Lembre-se de alternar as execuções subsequentes para Cache em caso de falha se você planeja usar entradas novas ou diferentes para essas execuções.

> **Note**
>
> Recomendamos o modo Cache sempre usando o mesmo conjunto de dados de teste, mas não para um lote de execuções. Se você definir esse modo para um grande lote de execuções, o sistema poderá exportar grandes quantidades de dados para o Amazon S3, resultando em maiores tempos de exportação e custos de armazenamento.

Controle o tamanho do cache de execução

HealthOmics não exclui nem arquiva automaticamente nenhum dado de cache executado nem aplica as regras de limpeza do Amazon S3 para gerenciar os dados do cache. Recomendamos que você realize limpezas regulares do cache para economizar nos custos de armazenamento do Amazon S3 e manter o tamanho do cache de execução gerenciável. Você pode excluir arquivos diretamente ou definir retention/replication políticas de dados no bucket de execução do cache.

Por exemplo, você pode configurar uma política de ciclo de vida do Amazon S3 para expirar objetos após 90 dias ou pode limpar manualmente os dados do cache no final de cada projeto de desenvolvimento.

As informações a seguir podem ajudá-lo a gerenciar o tamanho dos dados do cache:

- Você pode ver quantos dados estão no cache verificando o Amazon S3. HealthOmics não monitora nem relata o tamanho do cache.
- Se você excluir uma entrada de cache válida, a execução subsequente não falhará. HealthOmics recalcula a tarefa e suas tarefas dependentes.
- Se você modificar nomes de cache ou estruturas de diretórios de forma que HealthOmics não consiga encontrar uma entrada correspondente para uma tarefa, HealthOmics recalculará a tarefa.

Se você precisar verificar se uma entrada de cache ainda é válida, verifique o número da versão do manifesto do cache. Para obter mais informações, consulte Atualizações de versões do manifesto e atualização de dados.

Criando um cache de execução

Ao criar um cache de execução, você especifica uma localização do Amazon S3 para os dados do cache. Esses dados devem estar imediatamente acessíveis. O cache de chamadas não recupera objetos arquivados no Glacier (como as classes de armazenamento GFR e GDA).

Se o bucket do Amazon S3 para os dados do cache pertencer a outra pessoa Conta da AWS, forneça esse ID da conta ao criar o cache de execução.

Criando um cache de execução usando o console

No console, siga estas etapas para criar um cache de execução.

1. Abra o console do HealthOmics .

2. Se necessário, abra o painel de navegação esquerdo (≡). Escolha Executar caches.
3. Na página Executar caches, escolha Criar cache de execução.
4. No painel de detalhes do cache de execução da página Criar cache de execução, configure estes campos:
   a. Insira um nome para o cache de execução.
   b. (Opcional) Insira uma descrição.
   c. Insira um local do S3 para a saída em cache. Escolha um bucket na mesma região do seu fluxo de trabalho.
   d. (Opcional) Insira o Conta da AWS do proprietário do bucket para verificar a propriedade do bucket. Se você não inserir um valor, o valor padrão será o ID da sua conta.
   e. Em Comportamento do cache, configure o comportamento padrão (seja para armazenar em cache as saídas para execuções com falha ou para todas as execuções). Ao iniciar uma execução, você pode, opcionalmente, substituir o comportamento padrão.
5. (Opcional) Associe uma ou mais tags ao cache de execução.
6. Escolha Criar cache de execução. O console exibe o novo cache de execução na tabela Executar caches.

Criando um cache de execução usando a CLI

Use o comando create-run-cacheCLI para criar um cache de execução. O comportamento padrão do cache é`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"
```

Se a criação for bem-sucedida, você receberá uma resposta com os seguintes campos.

```
{
  "arn": "string",
  "id": "string",
  "status": "ACTIVE"
```

```
  "tags": {}
  }
```

Atualizando um cache de execução

Você pode alterar o nome, a descrição, as tags ou o comportamento do cache, mas não a localização do cache no S3.

Atualizando um cache de execução usando o console

No console, siga estas etapas para atualizar um cache de execução.

1. Abra o console do HealthOmics .
2. Se necessário, abra o painel de navegação esquerdo (≡). Escolha Executar caches.
3. Na tabela Executar caches, escolha o cache de execução a ser atualizado e, em seguida, escolha Editar.
4. No painel de detalhes da execução do cache, você pode atualizar os campos de nome, descrição e comportamento do cache de execução.
5. (Opcional) Associe uma ou mais novas tags ao cache de execução ou remova as tags existentes.
6. Escolha Salvar cache de execução.

Atualizando um cache de execução usando a CLI

Use o comando update-run-cacheCLI para atualizar um cache de execução.

```
aws omics update-run-cache \
      --name "workflow 123 run cache" \
      --id "workflow id" \
      --description "my run cache" \
      --cache-behavior "CACHE_ALWAYS"
```

Se a atualização for bem-sucedida, você receberá uma resposta sem campos de dados.

Excluindo um cache de execução

Você pode excluir um cache de execução se nenhuma execução ativa o estiver usando. Se alguma execução estiver usando o cache de execução, aguarde a conclusão das execuções ou você poderá cancelá-las.

A exclusão de um cache de execução remove o recurso e seus metadados, mas não exclui os dados no Amazon S3. Depois de excluir o cache, você não poderá reconectá-lo nem usá-lo para execuções subsequentes.

Os dados em cache permanecem no Amazon S3 para sua inspeção. Você pode remover dados antigos do cache usando Delete operações padrão do S3. Como alternativa, crie uma política de ciclo de vida do Amazon S3 para expirar dados em cache que você não usa mais.

Excluindo um cache de execução usando o console

No console, siga estas etapas para excluir um cache de execução.

1. Abra o console do HealthOmics .
2. Se necessário, abra o painel de navegação esquerdo (≡). Escolha Executar caches.
3. Na tabela Executar caches, escolha o cache de execução a ser excluído.
4. No menu Executar tabela de caches, escolha Excluir.
5. Na caixa de diálogo modal, salve o link de dados do cache do Amazon S3 para referência futura e confirme que você deseja excluir o cache de execução.

   Você pode usar o link do Amazon S3 para inspecionar os dados em cache, mas não pode revincular os dados a outro cache de execução. Exclua os dados do cache quando terminar a inspeção.

Excluindo um cache de execução usando a CLI

Use o comando delete-run-cacheCLI para excluir um cache de execução.

```
aws omics delete-run-cache \
      --id "my cache id"
```

Se a exclusão for bem-sucedida, você receberá uma resposta sem campos de dados.

Conteúdo de um cache de execução

HealthOmics organiza seu cache de execução com a seguinte estrutura em seu bucket do S3:

```
s3://{cache.S3location}/{cache.uuid}/runID/taskID/{cacheentry.uuid}/
```

O cache.uuid é o ID global exclusivo do cache. O cacheentry.uuid é o uuid globalmente exclusivo para uma tarefa em cache. HealthOmics atribui os uuids a caches e tarefas.

Para todos os mecanismos de fluxo de trabalho, o cache contém os seguintes arquivos:

- O {cacheentryuuid}.json arquivo — HealthOmics cria esse arquivo de manifesto, que contém informações sobre o cache, incluindo uma lista de todos os itens no cache e a versão do cache.
- Arquivos de saída da tarefa — Cada saída da tarefa consiste em um ou mais arquivos, conforme definido pela tarefa.

Para um fluxo de trabalho que usa o Nextflow, o mecanismo do Nextflow cria esses arquivos adicionais no cache:

- O command.out arquivo — Esse arquivo contém o conteúdo padrão da execução da tarefa.
- O .exitcode arquivo — Esse arquivo contém o código de saída da tarefa (um número inteiro).

> **Note**
>
> Se você quiser acessar arquivos de tarefas intermediárias em seu cache de execução para solucionar problemas avançados, declare esses arquivos como saídas de tarefas na definição do fluxo de trabalho.

Recursos de cache específicos do mecanismo

HealthOmics tenta fornecer uma implementação consistente do cache de chamadas em todos os mecanismos de fluxo de trabalho. Há algumas diferenças com base na forma como cada mecanismo de fluxo de trabalho lida com casos específicos:

- Próximo fluxo
  - O armazenamento em cache em diferentes versões do Nextflow não é garantido. Por exemplo, se você executar uma tarefa na v23.10.0 e, posteriormente, executar a mesma tarefa na v24.10.8, HealthOmics considere a segunda execução como uma perda de cache.
  - Você pode desativar o armazenamento em cache para tarefas individuais usando a false diretiva cache. Para obter informações sobre essa diretiva, consulte os Processos na especificação Nextflow.

  - HealthOmics usa o modo tolerante Nextflow, mas não oferece suporte ao modo de cache profundo.
  - O armazenamento em cache avalia cada objeto individual do S3 se você usar um padrão global no caminho do S3 até as entradas de uma tarefa. Se você adicionar um novo objeto, HealthOmics recalcula somente as tarefas que usam o novo objeto.
  - HealthOmics não armazena em cache as novas tentativas de tarefas. Esse comportamento é consistente com o comportamento padrão do Nextflow.
- WDL
  - HealthOmics suporta o novo tipo de “diretório” para entradas quando você usa a versão de desenvolvimento do fluxo de trabalho da WDL. Para o armazenamento em cache de chamadas, se algum objeto no diretório mudar, HealthOmics recalcula todas as tarefas que entram no diretório.
  - HealthOmics oferece suporte ao cache no nível da tarefa, mas não ao cache no nível do fluxo de trabalho.
  - Você pode desativar o armazenamento em cache para tarefas individuais usando o atributo volatile. Para obter mais informações, consulte Desative o armazenamento em cache no nível da tarefa com o atributo volatile.
- CAPUZ
  - As saídas constantes das tarefas não são explicitamente visíveis nos manifestos. HealthOmics armazena em cache saídas constantes como arquivos intermediários.
  - Você pode controlar o armazenamento em cache para tarefas individuais usando o WorkReuserecurso.

Usando o cache de execução

Por padrão, as execuções não usam um cache de execução. Para usar um cache para a execução, você especifica o cache de execução e o comportamento do cache de execução ao iniciar a execução.

Após a conclusão da execução, você pode usar o console, os CloudWatch registros ou as operações da API para rastrear os acessos ao cache ou solucionar problemas de cache. Para obter mais detalhes, consulte Rastreamento de informações de cache de chamadas e Solução de problemas de cache de chamadas.

Se uma ou mais tarefas em uma execução gerarem saídas não determinísticas, é altamente recomendável que você não use o cache de chamadas para a execução ou desative essas tarefas específicas do armazenamento em cache. Para obter mais informações, consulte Modelo de responsabilidade compartilhada.

> **Note**
>
> Você fornece uma função de serviço do IAM ao iniciar uma execução. Para usar o cache de chamadas, a função de serviço precisa de permissão para acessar a localização do cache de execução do Amazon S3. Para obter mais informações, consulte Funções de serviço para AWS HealthOmics.

Você pode usar o Amazon Q CLI para analisar e gerenciar seus dados de cache de execução. Para obter mais informações, consulte Exemplos de solicitações para a Amazon Q CLI e HealthOmics o tutorial de IA generativa da Agentic sobre. GitHub

Tópicos

Configurando uma execução com cache de execução usando o console

No console, você configura o cache de execução para uma execução ao iniciar a execução.

1. Abra o console do HealthOmics .
2. Se necessário, abra o painel de navegação esquerdo (≡). Selecione Execuções.
3. Na página Execuções, escolha a execução a ser iniciada.
4. Escolha Iniciar execução e conclua as etapas 1 e 2 de Iniciar execução conforme descrito emIniciando uma execução usando o console.
5. Na etapa 3 de Iniciar execução, escolha Selecionar um cache de execução existente.
6. Selecione o cache na lista suspensa Executar ID do cache.

7. Para substituir o comportamento padrão do cache de execução, escolha o comportamento do cache para a execução. Para obter mais informações, consulte Comportamento do cache de execução.
8. Continue com a etapa 4 de Iniciar execução.

Configurando uma execução com cache de execução usando a CLI

Para iniciar uma execução que usa um cache de execução, adicione o parâmetro cache-id ao comando da CLI start-run. Opcionalmente, use o `cache-behavior` parâmetro para substituir o comportamento padrão que você configurou para o cache de execução. O exemplo a seguir mostra somente os campos de cache do comando:

```
aws omics start-run \
        ...
      --cache-id "xxxxxx"    \
      --cache-behavior  CACHE_ALWAYS
```

Se a operação for bem-sucedida, você receberá uma resposta sem campos de dados.

Casos de erro para executar caches

Para os cenários a seguir, HealthOmics talvez não armazene em cache as saídas da tarefa, mesmo para uma execução com o comportamento de cache definido como Cache always.

- Se a execução encontrar um erro antes que a primeira tarefa seja concluída com êxito, não há saídas de cache para exportar.
- Se o processo de exportação falhar, HealthOmics não salva as saídas da tarefa no local do cache do Amazon S3.
- Se a execução falhar devido a um filesystem out of space erro, o cache de chamadas não salvará nenhuma saída da tarefa.
- Se você cancelar uma execução, o cache de chamadas não salvará nenhuma saída de tarefa.
- Se a execução tiver um tempo limite de execução, o cache de chamadas não salvará nenhuma saída da tarefa, mesmo que você tenha configurado a execução para usar o cache em caso de falha.

Rastreamento de informações de cache de chamadas

Você pode acompanhar eventos de cache de chamadas (como executar ocorrências de cache) usando o console, a CLI ou os registros. CloudWatch

Tópicos

Rastreie os acessos ao cache usando o console

Na página de detalhes da execução de uma execução, a tabela Executar tarefas exibe as informações de ocorrência do Cache para cada tarefa. A tabela também inclui um link para a entrada de cache associada. Use o procedimento a seguir para visualizar as informações de ocorrência do cache para uma execução.

1. Abra o console do HealthOmics .
2. Se necessário, abra o painel de navegação esquerdo (≡). Selecione Execuções.
3. Na página Execuções, escolha a execução a ser inspecionada.
4. Na página de detalhes da execução, escolha a guia Executar tarefas para exibir a tabela de tarefas.
5. Se uma tarefa tiver um cache hit, a coluna Cache hit conterá um link para o local de execução da entrada do cache no Amazon S3.
6. Escolha o link para inspecionar a entrada do cache de execução.

Rastreie o armazenamento em cache de chamadas usando a CLI

Use o comando da CLI get-run para confirmar se a execução usou um cache de chamadas.

```
aws omics get-run --id 1234567
```

Na resposta, se o `cacheId` campo estiver definido, a execução usa esse cache.

Use o comando list-run-tasksCLI para recuperar o local dos dados do cache para cada tarefa em cache na execução.

```
aws omics list-run-tasks --id 1234567
```

Na resposta, se o campo CacheHit de uma tarefa for verdadeiro, o campo Cache3URI fornecerá a localização dos dados do cache para essa tarefa.

Você também pode usar o comando get-run-taskCLI para recuperar o local dos dados do cache para uma tarefa específica:

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

Monitore o armazenamento em cache de chamadas usando o CloudWatch Logs

HealthOmics cria registros de atividades de cache no grupo de `/aws/omics/WorkflowLog` CloudWatch registros. <cache_id><cache_uuid>Há um fluxo de log para cada cache de execução: runCache//.

Para execuções que usam o cache de chamadas, HealthOmics gera entradas de CloudWatch registros para esses eventos:

- criando uma entrada de cache (CACHE_ENTRY_CREATED)
- correspondendo a uma entrada de cache (CACHE_HIT)
- falha em corresponder a uma entrada de cache (CACHE_MISS)

Para obter mais informações sobre esses registros, consulteLogin CloudWatch.

Use a seguinte consulta do CloudWatch Insights no grupo de `/aws/omics/WorkflowLog` registros para retornar o número de acessos ao cache por execução desse cache:

```
filter @logStream like 'runCache/<CACHE_ID>/'
 fields @timestamp, @message
 filter logMessage like 'CACHE_HIT'
 parse "run: *," as run
 stats count(*) as cacheHits by run
```

Use a consulta a seguir para retornar o número de entradas de cache criadas por cada execução:

```
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
```

Compartilhamento de HealthOmics fluxos de trabalho

Como proprietário de um fluxo de trabalho privado, você pode compartilhar o fluxo de trabalho com Conta da AWS alguém da mesma região. Para compartilhar um fluxo de trabalho com mais de um Conta da AWS, você cria vários compartilhamentos do mesmo fluxo de trabalho.

Como proprietário, você pode revogar o acesso a um fluxo de trabalho compartilhado excluindo o compartilhamento.

> **Note**
>
> HealthOmics permite automaticamente que um fluxo de trabalho compartilhado acesse o repositório Amazon ECR enquanto o fluxo de trabalho está sendo executado na conta do assinante. Você não precisa conceder acesso adicional ao repositório para fluxos de trabalho compartilhados.

Quando você compartilha um fluxo de trabalho, o assinante pode usar qualquer uma das versões do fluxo de trabalho. Se você precisar de controle de acesso em nível de versão para um fluxo de trabalho compartilhado, recomendamos criar fluxos de trabalho separados em vez de usar versões de fluxo de trabalho.

Tópicos

- Inscrever-se em um fluxo de trabalho compartilhado
- Monitorando o status de um compartilhamento de fluxo de trabalho
- Compartilhando um fluxo de trabalho privado usando o console
- Compartilhando um fluxo de trabalho privado usando a CLI
- Aceitando um fluxo de trabalho compartilhado usando o console
- Executando um fluxo de trabalho compartilhado usando o console
- Executando um fluxo de trabalho compartilhado usando a API

Inscrever-se em um fluxo de trabalho compartilhado

Para assinar um fluxo de trabalho compartilhado, siga estas etapas gerais para aceitar e usar o fluxo de trabalho:

1. Use o console ou a API para aceitar o compartilhamento. Defina sua região atual para a mesma região da solicitação de compartilhamento.
   - Para encontrar a solicitação de compartilhamento no console, navegue até a página Todos os compartilhamentos de recursos e escolha a guia Compartilhado comigo.
2. Use o console ou a API para criar uma execução para o fluxo de trabalho compartilhado.
   - Para encontrar a página de detalhes do fluxo de trabalho no console, navegue até Compartilhado comigo (consulte a etapa 1) e escolha o link Recurso para o fluxo de trabalho compartilhado.
3. Você fornece seus próprios dados de entrada para o fluxo de trabalho.
4. O fluxo de trabalho compartilhado é executado em seu Conta da AWS.

Como assinante de um fluxo de trabalho compartilhado, o sistema impede que você execute as seguintes ações de fluxo de trabalho:

- Exportação de um fluxo de trabalho compartilhado
- Executando novamente o fluxo de trabalho compartilhado
  - Você cria uma nova execução para o fluxo de trabalho compartilhado.
- Compartilhando novamente o fluxo de trabalho.
- Atribuição de uma tag ao fluxo de trabalho.
- Excluindo o fluxo de trabalho.
  - Quando você não precisar mais do fluxo de trabalho, exclua o compartilhamento do fluxo de trabalho.

Consulte Compartilhamento de recursos entre contas em AWS HealthOmics para obter informações adicionais sobre o compartilhamento de recursos.

Monitorando o status de um compartilhamento de fluxo de trabalho

HealthOmics envia um evento EventBridge para cada alteração de status de um compartilhamento de fluxo de trabalho. Se você quiser receber notificações sobre mudanças de status específicas,

configure uma EventBridge regra para monitorar eventos de mudança de status do compartilhamento do fluxo de trabalho. Por exemplo:

- Você quer receber uma notificação sempre que receber uma solicitação de compartilhamento de fluxo de trabalho e sempre que um usuário revogar um compartilhamento de fluxo de trabalho.
- Depois de iniciar uma solicitação de compartilhamento de fluxo de trabalho, você deseja receber uma notificação quando o usuário aceitar ou recusar a solicitação.

Para obter detalhes sobre o uso de eventos, consulteUsando EventBridge com AWS HealthOmics.

Compartilhando um fluxo de trabalho privado usando o console

No console, você pode compartilhar um fluxo de trabalho privado com um Conta da AWS na mesma região do fluxo de trabalho.

Para compartilhar um fluxo de trabalho privado

1. Abra o console de HealthOmics .
2. Se necessário, abra o painel de navegação esquerdo (≡). Escolha fluxos de trabalho privados.
3. Na tabela Fluxos de trabalho na página Fluxos de trabalho privados, selecione o fluxo de trabalho a ser compartilhado e escolha Compartilhar.
4. No painel Detalhes do compartilhamento da página Compartilhar fluxo de trabalho, insira um nome descritivo para o compartilhamento e insira o nome Conta da AWS do assinante.
5. Escolha Compartilhar recurso. O console exibe compartilhamentos de recursos na página Todos os compartilhamentos de recursos.

O estado inicial do compartilhamento está pendente. Depois que o assinante aceita o compartilhamento, o estado muda para ativo.

Compartilhando um fluxo de trabalho privado usando a CLI

Use a operação da API create-share para criar um compartilhamento de fluxo de trabalho. O assinante principal é o Conta da AWS usuário que terá acesso ao fluxo de trabalho.

```
aws omics create-share \
  --resource-arn "arn:aws:omics:us-west-2:555555555555:workflow/123456" \
  --principal-subscriber "123456789012" \
```

```
  --name "my_Share-123"
```

Se a criação for bem-sucedida, você receberá uma resposta com o ID e o status do compartilhamento.

```
{
"shareId": "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a",
"name": "my_Share-123",
"status": "PENDING"
}
```

O compartilhamento permanece pendente até que o assinante o aceite usando a operação da `accept-share` API.

Consulte Compartilhamento de recursos entre contas em AWS HealthOmics para ver outros exemplos de uso da API.

Aceitando um fluxo de trabalho compartilhado usando o console

Você pode usar o console para aceitar um compartilhamento de fluxo de trabalho oferecido. Certifique-se de configurar o console na mesma região do fluxo de trabalho.

1. Abra o console de HealthOmics .
2. Se necessário, abra o painel de navegação esquerdo (≡). Escolha Todos os compartilhamentos de recursos e, em seguida, escolha a guia Compartilhado comigo.
3. Na tabela Recursos compartilhados comigo, selecione o compartilhamento do fluxo de trabalho e escolha Aceitar.

Depois de aceitar o fluxo de trabalho, escolha o link Recurso para o fluxo de trabalho compartilhado para ver seus detalhes.

Executando um fluxo de trabalho compartilhado usando o console

Depois de aceitar um compartilhamento de fluxo de trabalho, você pode iniciar uma execução no fluxo de trabalho.

1. Abra o console de HealthOmics .
2. Se necessário, abra o painel de navegação esquerdo (≡). Escolha Todos os compartilhamentos de recursos e, em seguida, escolha a guia Compartilhado comigo.

3. Na tabela Recursos compartilhados comigo, escolha o link Recurso para o fluxo de trabalho compartilhado.
4. Na página de detalhes do fluxo de trabalho, escolha Criar execução.

   O console abre a página Criar execução, com o tipo de fluxo de trabalho (compartilhado) e a ID do fluxo de trabalho pré-preenchidos.
5. Configure os campos restantes no formulário Criar execução. Para obter informações adicionais, consulte Iniciando uma execução usando o console.

Executando um fluxo de trabalho compartilhado usando a API

Use get-workflow para recuperar o ARN do fluxo de trabalho compartilhado.

```
aws omics get-workflow --id 1234567 \
--workflow-owner-id 55555555555
```

Ao executar o fluxo de trabalho, forneça a Conta da AWS ID do proprietário do fluxo de trabalho e o ARN do fluxo de trabalho compartilhado.

```
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
```

Fluxos de trabalho do Ready2Run em HealthOmics

Os fluxos de trabalho do Ready2Run são fluxos de trabalho pré-configurados publicados por editores terceirizados. Alguns editores, como a Sentieon Inc, oferecem fluxos de trabalho baseados em assinatura. Outros fluxos de trabalho do Ready2Run não exigem uma assinatura, e alguns fluxos de trabalho são de código aberto, como os fluxos de trabalho NF-Core.

Os fluxos de trabalho do Ready2Run são adequados aos seguintes cenários:

- Você quer se concentrar na análise da saída do pipeline e na geração de resultados, sem a necessidade de configurar a infraestrutura subjacente.
- Você deseja replicar seus resultados usando fluxos de trabalho estabelecidos.
- Como desenvolvedor de software, você deseja integrar seu aplicativo diretamente ao HealthOmics SDK.

HealthOmics suporta controle de versão para fluxos de trabalho do Ready2Run. Para um fluxo de trabalho do Ready2Run que oferece versões, você pode especificar o nome da versão ao iniciar uma execução.

Todos os fluxos de trabalho do Ready2Run fornecem registros, incluindo CloudWatch registros, que você pode usar para solucionar problemas.

> **Note**
>
> Os fluxos de trabalho do Sentieon Ready2Run são baseados em assinatura. Quando você executa um fluxo de trabalho do Sentieon Ready2Run pela primeira vez em uma conta, o Sentieon cria automaticamente uma licença de avaliação de duas semanas para sua. Conta da AWS A licença é válida para todos os fluxos de trabalho do Sentieon Ready2Run. Após o término do período de avaliação, você poderá solicitar uma licença permanente ou solicitar uma extensão da licença de avaliação. Para mais detalhes, consulte Subscribing to Sentieon Ready2Run workflows.

Tópicos

- Iniciando fluxos de trabalho do HealthOmics Ready2Run usando o console
- Iniciando fluxos de trabalho do HealthOmics Ready2Run usando a API

Fluxos de trabalho Ready2Run disponíveis em HealthOmics

A tabela a seguir lista os fluxos de trabalho do Ready2Run que estão disponíveis em. HealthOmics

Você pode fazer login no HealthOmicsconsole para ver informações detalhadas sobre esses fluxos de trabalho, incluindo parâmetros de entrada e diagramas de fluxo de trabalho. Para obter informações sobre preços dos fluxos de trabalho do Ready2Run, consulte Preços. HealthOmics

> **Note**
>
> Cada fluxo de trabalho do Ready2Run tem um tamanho máximo de arquivo de entrada. Esses tamanhos máximos de arquivo não são ajustáveis.

Nome do fluxo de trabalho	Publicador	É necessária uma assinatura?	Tamanho máximo do arquivo de entrada (GiB)	Tempo de execução estimado (HH:MM)
AlphaFold para 601-1200 resíduos	Google DeepMind	Não	1	11:15
AlphaFold para até 600 resíduos	Google DeepMind	Não	1	7:30
Bases2Fastq para 2x150	Biociências do Elemento	Não	1000	1:45
Bases2Fastq para 2x300	Biociências do Elemento	Não	1000	1:30
Bases2Fastq para 2x75	Biociências do Elemento	Não	500	0:45

Nome do fluxo de trabalho	Publicador	É necessária uma assinatura?	Tamanho máximo do arquivo de entrada (GiB)	Tempo de execução estimado (HH:MM)
ESMFold para até 800 resíduos	Metapesquisa	Não	1	0:15
GATK-BP fq2bam	Instituto Broad	Não	64	10 & 10
Linha germinativa GATK-BP bam2vcf para genoma 30x	Instituto Broad	Não	39	2:45
Linha germinativa GATK-BP fq2vcf para genoma 30x	Instituto Broad	Não	64	12:30
GATK-BP Somatic WES bam2vcf	Instituto Broad	Não	86	1:30
NVIDIA Parabricks BAM2 FQ2 BAM WGS para até 30X	Corporação NVIDIA	Não	80	1:39
NVIDIA Parabricks BAM2 FQ2 BAM WGS para até 50X	Corporação NVIDIA	Não	120	2:45

Nome do fluxo de trabalho	Publicador	É necessária uma assinatura?	Tamanho máximo do arquivo de entrada (GiB)	Tempo de execução estimado (HH:MM)
NVIDIA Parabricks BAM2 FQ2 BAM WGS para até 5X	Corporação NVIDIA	Não	20	0:18
NVIDIA Parabricks FQ2 BAM WGS para até 30X	Corporação NVIDIA	Não	71	1:00
NVIDIA Parabricks FQ2 BAM WGS para até 50X	Corporação NVIDIA	Não	137	1:45
NVIDIA Parabricks FQ2 BAM WGS para até 5X	Corporação NVIDIA	Não	13	0:15
NVIDIA Parabricks DeepVariant Germline WGS para até 30X	Corporação NVIDIA	Não	71	2:00
NVIDIA Parabricks DeepVariant Germline WGS para até 50X	Corporação NVIDIA	Não	137	3:30

Nome do fluxo de trabalho	Publicador	É necessária uma assinatura?	Tamanho máximo do arquivo de entrada (GiB)	Tempo de execução estimado (HH:MM)
NVIDIA Parabricks DeepVariant Germline WGS para até 5X	Corporação NVIDIA	Não	12	0:30
NVIDIA Parabricks HaplotypeCaller Germline WGS para até 30X	Corporação NVIDIA	Não	71	1:15
NVIDIA Parabricks HaplotypeCaller Germline WGS para até 50X	Corporação NVIDIA	Não	137	2:00
NVIDIA Parabricks HaplotypeCaller Germline WGS para até 5X	Corporação NVIDIA	Não	13	0:15
NVIDIA Parabricks Somatic Mutect2 WGS para até 50X	Corporação NVIDIA	Não	196	0:45
sc RNAseq com Kallisto BUStools	NF-núcleo	Não	119	1:30

Nome do fluxo de trabalho	Publicador	É necessária uma assinatura?	Tamanho máximo do arquivo de entrada (GiB)	Tempo de execução estimado (HH:MM)
1 colher de RNAseq chá com salmão frito	NF-núcleo	Não	119	2:30
sc RNAseq com STARsolo	NF-núcleo	Não	119	2:30
Sentieon Germline BAM WES para até 300x	Sentieon, Inc.	Sim	9	1:00
Sentieon Germline BAM WGS para até 32x	Sentieon, Inc.	Sim	18	1:30
Sentieon Germline FASTQ WES por até 100x	Sentieon, Inc.	Sim	5	0:45
Sentieon Germline FASTQ WES por até 300x	Sentieon, Inc.	Sim	26	2:00
Sentieon Germline FASTQ WGS por até 32x	Sentieon, Inc.	Sim	51	3:30

Nome do fluxo de trabalho	Publicador	É necessária uma assinatura?	Tamanho máximo do arquivo de entrada (GiB)	Tempo de execução estimado (HH:MM)
Sentieon LongRead para ONT	Sentieon, Inc.	Sim	25	1:30
Sentieon LongRead para PacBio HiFi	Sentieon, Inc.	Sim	58	4:00
Sentieon Somatic WES	Sentieon, Inc.	Sim	50	2:30
Sentieon Somatic WGS	Sentieon, Inc.	Sim	113	4:30
Ultima Genomics DeepVariant para até 40x	Ultima Genomics	Não	91	1:55

Quando você usa um fluxo de trabalho Ready2Run, seu fluxo de trabalho é pré-configurado e não pode ser editado. Ao contrário dos fluxos de trabalho privados, os fluxos de trabalho do Ready2Run não oferecem suporte ao seguinte:

- Aumentar o tamanho máximo do arquivo de entrada
- Alterar os recursos computacionais ou executar o armazenamento
- Alterando a definição do fluxo de trabalho ou os contêineres
- Adicionar execuções a um grupo de corridas
- Compartilhando o fluxo de trabalho

Se o editor tiver compartilhado o fluxo de trabalho do Ready2Run GitHub, você poderá criar seu próprio fluxo de trabalho privado com base no fluxo de trabalho do Ready2Run. A tabela a seguir fornece links para GitHub fluxos de trabalho de cada editor.

Publicador	Fluxos de trabalho em GitHub
Google DeepMind, Meta Pesquisa	Fluxos de trabalho para dobrar proteínas
Biociências do Elemento	Para obter informações, entre em contato com a Element Biosciences
Instituto Broad	Fluxos de trabalho do GATK
Corporação NVIDIA	Fluxos de trabalho do Parabricks
nf-core	Fluxos de trabalho NF-Core
Sentieon	Fluxos de trabalho do Sentieon
Ultima Genomics	Fluxos de trabalho da Ultima Genomics

Inscrevendo-se nos fluxos de trabalho do Sentieon Ready2Run

Os fluxos de trabalho do Sentieon Ready2Run são baseados em assinatura. Quando você executa um fluxo de trabalho do Sentieon Ready2Run pela primeira vez em uma conta, o Sentieon cria automaticamente uma licença de avaliação de duas semanas para sua. Conta da AWS A licença é válida para todos os fluxos de trabalho do Sentieon Ready2Run. Após o término do período de avaliação, você poderá solicitar uma licença permanente ou solicitar uma extensão da licença de avaliação.

Siga estas etapas para assinar os fluxos de trabalho do Sentieon Ready2Run:

- Encontre sua ID de usuário da AWS Canonical seguindo estas instruções.
- Envie um e-mail para o grupo de suporte da Sentieon (support@sentieon.com) para solicitar uma licença de software. Forneça seu ID de usuário AWS canônico no e-mail.

Iniciando fluxos de trabalho do HealthOmics Ready2Run usando o console

Usar fluxos de trabalho do Ready2Run no console é semelhante ao uso de um fluxo de trabalho privado. Uma diferença importante é que o editor do fluxo de trabalho fornece dados de amostra, para que você possa experimentar o fluxo de trabalho sem criar seus próprios dados.

Para usar um fluxo de trabalho Ready2Run no console

1. Abra o console de HealthOmics .
2. Se necessário, abra o painel de navegação esquerdo (≡). Escolha fluxos de trabalho do Ready2Run.
3. Na página Fluxos de trabalho do Ready2Run, escolha o fluxo de trabalho que você deseja usar. O console abre a página de detalhes desse fluxo de trabalho.
4. A guia de detalhes lista informações como nome, preço sugerido por execução, descrição, tipo de idioma do fluxo de trabalho, capacidade de armazenamento da execução, status, data de criação e parâmetros com descrições. A guia de detalhes também informa se o fluxo de trabalho requer uma assinatura.
5. Para usar o fluxo de trabalho, escolha Criar execução
6. Na página Especificar detalhes da execução, insira um nome de execução. Opcionalmente, você pode especificar a versão do fluxo de trabalho. Você também pode adicionar prioridade de execução à execução.
7. Insira ou selecione um local do Amazon S3 para a saída da execução.
8. Para o modo de retenção de metadados de execução, escolha se deseja reter ou remover os metadados de execução.
9. No painel Função de serviço, escolha se deseja usar uma função de serviço existente ou criar uma nova.
10. (Opcional) Adicione tags para ajudar a identificar e gerenciar sua corrida.
11. Escolha Próximo.
12. Na página Adicionar parâmetros, escolha uma das opções para adicionar os valores dos parâmetros de execução:

    - Selecione um arquivo de parâmetros (no formato JSON) em um local do Amazon S3.
    - Selecione um arquivo de parâmetros (no formato JSON) na sua unidade local.

- Insira manualmente os valores dos parâmetros.
- Execute o fluxo de trabalho com os dados de amostra do Ready2Run fornecidos pelo editor do fluxo de trabalho.

13. Se você fizer upload de um arquivo JSON, o console analisará o arquivo e executará a validação em linha. Em seguida, você pode atualizar manualmente os valores dos seus parâmetros conforme necessário.
14. Escolha Próximo.
15. Revise suas entradas e escolha Iniciar execução.

Iniciando fluxos de trabalho do HealthOmics Ready2Run usando a API

A maioria das operações de API se comporta de forma semelhante para fluxos de trabalho do Ready2Run e fluxos de trabalho privados.

Para retornar uma lista dos fluxos de trabalho disponíveis do Ready2Run, use list-workflows com o parâmetro definido como RUN. `type` READY2

```
aws omics list-workflows --type READY2RUN
```

Depois de identificar o fluxo de trabalho a ser executado a partir da resposta list-workflows, você pode usar get-workflow com o `--id` parâmetro para obter mais detalhes.

```
aws omics get-workflow --type READY2RUN --id workflow id
```

Para executar um fluxo de trabalho Ready2Run, você pode usar a operação de API start-run com o parâmetro de tipo de fluxo de trabalho definido como, conforme mostrado no exemplo a seguir `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 uma versão do fluxo de trabalho, use o parâmetro workflow-version, conforme mostrado neste exemplo.

```
aws-omics start-run \
  --workflow-type READY2RUN \
  ...
  --version-name '3.0.0'
```

Para monitorar sua execução, você pode usar a operação da API get-run, conforme mostrado.

```
aws-omics get-run \
  --id run id
```

HealthOmics armazenamento

Use o HealthOmics armazenamento para armazenar, recuperar, organizar e compartilhar dados genômicos com eficiência e baixo custo. HealthOmics o armazenamento compreende as relações entre diferentes objetos de dados, para que você possa definir quais conjuntos de leitura se originaram dos mesmos dados de origem. Isso fornece a proveniência dos dados.

Os dados armazenados no `ACTIVE` estado podem ser recuperados imediatamente. Os dados que não foram acessados por 30 dias ou mais são armazenados no `ARCHIVE` estado. Para acessar os dados arquivados, você pode reativá-los por meio das operações da API ou do console.

HealthOmics os armazenamentos de sequências são projetados para preservar a integridade do conteúdo dos arquivos. No entanto, a equivalência bit a bit dos arquivos de dados importados e dos arquivos exportados não é preservada devido à compactação durante a classificação em camadas ativa e de arquivamento.

Durante a ingestão, HealthOmics gera uma tag de entidade ou HealthOmics ETag, para possibilitar a validação da integridade do conteúdo de seus arquivos de dados. As partes de sequenciamento são identificadas e capturadas ETag no nível da fonte de um conjunto de leitura. O ETag cálculo não altera o arquivo real nem os dados genômicos. Depois que um conjunto de leitura é criado, ele não ETag deve mudar durante todo o ciclo de vida da fonte do conjunto de leitura. Isso significa que a reimportação do mesmo arquivo resulta no cálculo do mesmo ETag valor.

Tópicos

HealthOmics ETags e proveniência dos dados

A HealthOmics ETag (tag de entidade) é um hash do conteúdo ingerido em um armazenamento de sequências. Isso simplifica a recuperação e o processamento de dados, mantendo a integridade do conteúdo dos arquivos de dados ingeridos. ETag Isso reflete as alterações no conteúdo semântico do objeto, não em seus metadados. O tipo de conjunto de leitura e o algoritmo especificados determinam como o ETag é calculado. O ETag cálculo não altera o arquivo real nem os dados genômicos. Quando o esquema de tipo de arquivo do conjunto de leitura permite, o armazenamento de sequências atualiza os campos vinculados à proveniência dos dados.

Os arquivos têm uma identidade bit a bit e uma identidade semântica. A identidade bit a bit significa que os bits de um arquivo são idênticos, e uma identidade semântica significa que o conteúdo de um arquivo é idêntico. A identidade semântica é resistente a alterações de metadados e alterações de compressão, pois captura a integridade do conteúdo do arquivo.

Os conjuntos de leitura em armazenamentos HealthOmics sequenciais passam por compression/decompression ciclos e rastreamento de proveniência de dados durante todo o ciclo de vida de um objeto. Durante esse processamento, a identidade bit a bit de um arquivo ingerido pode mudar e espera-se que mude sempre que um arquivo for ativado; no entanto, a identidade semântica do arquivo é mantida. A identidade semântica é capturada como uma tag de HealthOmics entidade ou calculada durante ETag a ingestão do armazenamento de sequências e está disponível como metadados do conjunto de leitura.

Quando o esquema de tipo de arquivo do conjunto de leitura permite, os campos de atualizações do armazenamento de sequências são vinculados à proveniência dos dados. Para arquivos uBAM, BAM e CRAM, uma nova `Comment` tag `@CO` ou tag é adicionada ao cabeçalho. O comentário contém o ID do armazenamento da sequência e o carimbo de data/hora da ingestão.

Amazon S3 ETags

Ao acessar um arquivo usando o URI do Amazon S3, as operações da API do Amazon S3 também podem retornar valores do Amazon S3 e da soma de verificação. ETag Os valores do Amazon S3 ETag e da soma de verificação diferem do HealthOmics ETags porque representam a identidade bit a bit do arquivo. Para saber mais sobre metadados descritivos e objetos, consulte a documentação da API de objetos do Amazon S3. ETag Os valores do Amazon S3 podem mudar com cada ciclo de ativação de um conjunto de leitura e você pode usá-los para validar a leitura de um arquivo. No entanto, não armazene em cache ETag os valores do Amazon S3 para usar na validação da identidade do arquivo durante o ciclo de vida do arquivo, pois eles não permanecem consistentes.

Em contraste, o HealthOmics ETag permanece consistente durante todo o ciclo de vida do conjunto de leitura.

Como HealthOmics calcula ETags

O ETag é gerado a partir de um hash do conteúdo do arquivo ingerido. A família de ETag algoritmos é definida como padrão, mas pode ser configurada de forma diferente durante a criação do armazenamento de sequências. MD5up Quando o ETag é calculado, o algoritmo e os hashes calculados são adicionados ao conjunto de leitura. Os MD5 algoritmos suportados para tipos de arquivo são os seguintes.

- FASTQ_ MD5up — Calcula o MD5 hash de uma fonte de conjunto de leitura FASTQ completa e não compactada.
- BAM_ MD5up — Calcula o MD5 hash da seção de alinhamento de uma fonte de conjunto de leitura BAM ou UBAM não compactada, conforme representada no SAM, com base na referência vinculada, se houver uma disponível.
- CRAM_ MD5up — Calcula o MD5 hash da seção de alinhamento da fonte do conjunto de leitura CRAM não compactada, conforme representada no SAM, com base na referência vinculada.

> **Note**
>
> MD5 sabe-se que o hashing é vulnerável a colisões. Por causa disso, dois arquivos diferentes poderiam ter o mesmo ETag se tivessem sido fabricados para explorar a colisão conhecida.

Os algoritmos a seguir são compatíveis com a SHA256 família. Os algoritmos são calculados da seguinte forma:

- FASTQ_ SHA256up — Calcula o hash SHA-256 de uma fonte de conjunto de leitura FASTQ completa e não compactada.
- BAM_ SHA256up — Calcula o hash SHA-256 da seção de alinhamento de uma fonte de conjunto de leitura BAM ou UBAM não compactada, conforme representada no SAM, com base na referência vinculada, se houver uma disponível.
- CRAM_ SHA256up — Calcula o hash SHA-256 da seção de alinhamento de uma fonte de conjunto de leitura CRAM não compactada, conforme representada no SAM, com base na referência vinculada.

Os algoritmos a seguir são compatíveis com a SHA512 família. Os algoritmos são calculados da seguinte forma:

- FASTQ_ SHA512up — Calcula o hash SHA-512 de uma fonte de conjunto de leitura FASTQ completa e não compactada.
- BAM_ SHA512up — Calcula o hash SHA-512 da seção de alinhamento de uma fonte de conjunto de leitura BAM ou UBAM não compactada, conforme representada no SAM, com base na referência vinculada, se houver uma disponível.
- CRAM_ SHA512up — Calcula o hash SHA-512 da seção de alinhamento de uma fonte de conjunto de leitura CRAM não compactada, conforme representada no SAM, com base na referência vinculada.

Criação de uma loja HealthOmics de referência

Um armazenamento de referência em HealthOmics é um armazenamento de dados para o armazenamento de genomas de referência. Você pode ter uma única loja de referência em Conta da AWS cada região. Você pode criar um repositório de referência usando o console ou a CLI.

Tópicos

Criando um repositório de referência usando o console

Para criar um repositório de referências

1. Abra o console do HealthOmics .
2. Se necessário, abra o painel de navegação esquerdo (≡). Escolha Loja de referência.
3. Escolha genomas de referência nas opções de armazenamento de dados genômicos.
4. Você pode escolher um genoma de referência importado anteriormente ou importar um novo. Se você não importou um genoma de referência, escolha Importar genoma de referência no canto superior direito.
5. Na página Criar tarefa de importação de genoma de referência, escolha a opção Criação rápida ou Criação manual para criar um repositório de referência e, em seguida, forneça as seguintes informações.

- Nome do genoma de referência - Um nome exclusivo para esta loja.
- Descrição (opcional) - Uma descrição dessa loja de referência.
- Função do IAM - Selecione uma função com acesso ao seu genoma de referência.
- Referência do Amazon S3 - Selecione seu arquivo de sequência de referência em um bucket do Amazon S3.
- Tags (opcional) - forneça até 50 tags para esse repositório de referência.

Criando um repositório de referência usando a CLI

O exemplo a seguir mostra como criar um repositório de referência usando AWS CLI o. Você pode ter uma loja de referência por AWS região.

Os repositórios de referência suportam o armazenamento de arquivos FASTA com as extensões `.fasta.fa`,`.fas`,`.fsa`,,`.faa`,`.fna`,`.ffn`,`.frn`,, `.mpfa.seq`,`.txt`. A `bgzip` versão dessas extensões também é suportada.

No exemplo a seguir, *`reference store name`* substitua pelo nome que você escolheu para sua loja de referência.

```
aws omics create-reference-store --name "reference store name"
```

Você recebe uma resposta JSON com o ID e o nome da loja de referência, o ARN e a data e hora de quando sua loja de referência foi criada.

```
{
    "id": "3242349265",
    "arn": "arn:aws:omics:us-west-2:555555555555:referenceStore/3242349265",
    "name": "MyReferenceStore",
    "creationTime": "2022-07-01T20:58:42.878Z"
}
```

Você pode usar o ID do repositório de referência em AWS CLI comandos adicionais. Você pode recuperar a lista de repositórios de referência IDs vinculados à sua conta usando o list-reference-storescomando, conforme mostrado no exemplo a seguir.

```
aws omics list-reference-stores
```

Em resposta, você recebe o nome da sua loja de referência recém-criada.

```
{
    "referenceStores": [
        {
              "id": "3242349265",
              "arn": "arn:aws:omics:us-west-2:555555555555:referenceStore/3242349265",
              "name": "MyReferenceStore",
             "creationTime": "2022-07-01T20:58:42.878Z"
         }
     ]
}
```

Depois de criar um repositório de referência, você pode criar trabalhos de importação para carregar arquivos de referência genômica nele. Para fazer isso, você deve usar ou criar uma função do IAM para acessar os dados. Veja abaixo um exemplo de política .

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/*"
            ]
        }
    ]
}
```

Você também deve ter uma política de confiança semelhante ao exemplo a seguir.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": [
                   "omics.amazonaws.com"
                ]
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

Agora você pode importar um genoma de referência. Este exemplo usa o Genome Reference Consortium Human Build 38 (hg38), que é de acesso aberto e está disponível no Registry of Open Data em. AWS O bucket que hospeda esses dados está localizado no Leste dos EUA (Ohio). Para usar buckets em outras AWS regiões, você pode copiar os dados para um bucket do Amazon S3 hospedado em sua região. Use o AWS CLI comando a seguir para copiar o genoma para o seu bucket do Amazon S3.

```
aws s3 cp s3://broad-references/hg38/v0/Homo_sapiens_assembly38.fasta s3://amzn-s3-demo-bucket
```

Em seguida, você pode começar seu trabalho de importação. Substitua *reference store IDrole ARN*,, e *source file path* com sua própria entrada.

```
aws omics start-reference-import-job --reference-store-id reference store ID --role-arn role ARN --sources source file path
```

Depois que os dados forem importados, você receberá a seguinte resposta em JSON.

```
{
        "id": "7252016478",
        "referenceStoreId": "3242349265",
        "roleArn": "arn:aws:iam::111122223333:role/OmicsReferenceImport",
```

```
        "status": "CREATED",
        "creationTime": "2022-07-01T21:15:13.727Z"
}
```

Você pode monitorar o status de um trabalho usando o comando a seguir. No exemplo a seguir, substitua *reference store ID* e *job ID* pelo ID da loja de referência e pelo ID do trabalho sobre o qual você deseja saber mais.

```
aws omics get-reference-import-job --reference-store-id reference store ID --id job ID
```

Em resposta, você recebe uma resposta com os detalhes dessa loja de referência e seu status.

```
{
    "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"
        }
    ]
}
```

Você também pode encontrar a referência que foi importada listando suas referências e filtrando-as com base no nome da referência. *reference store ID*Substitua pelo ID da loja de referência e adicione um filtro opcional para restringir a lista.

```
aws omics list-references --reference-store-id reference store ID --filter
 name=MyReference
```

Em resposta, você recebe as seguintes informações.

```
{
    "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 saber mais sobre os metadados de referência, use a operação da get-reference-metadataAPI. No exemplo a seguir, *reference store ID* substitua pelo ID da loja de referência e *reference ID* pelo ID de referência sobre o qual você deseja saber mais.

```
aws omics get-reference-metadata --reference-store-id reference store ID --id reference
 ID
```

Você recebe as seguintes informações em resposta.

```
{
    "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
```

```
        }
    }
}
```

Você também pode baixar partes do arquivo de referência usando get-reference. No exemplo a seguir, *reference store ID* substitua pelo ID da loja de referência e *reference ID* pelo ID de referência do qual você deseja fazer o download.

```
aws omics get-reference --reference-store-id reference store ID --id reference ID --
part-number 1 outfile.fa
```

Criando um armazenamento HealthOmics de sequências

HealthOmics os armazenamentos de sequências suportam o armazenamento de arquivos genômicos nos formatos não alinhados de FASTQ (somente gzip) e. uBAM Ele também suporta os formatos alinhados de BAM e. CRAM

Os arquivos importados são armazenados como conjuntos de leitura. Você pode adicionar tags aos conjuntos de leitura e usar políticas do IAM para controlar o acesso aos conjuntos de leitura. Os conjuntos de leitura alinhados exigem um genoma de referência para alinhar as sequências genômicas, mas é opcional para conjuntos de leitura não alinhados.

Para armazenar conjuntos de leitura, primeiro você cria um armazenamento de sequências. Ao criar um armazenamento de sequências, você pode especificar um bucket opcional do Amazon S3 como um local alternativo e o local onde os registros de acesso do S3 são armazenados. O local alternativo é usado para armazenar todos os arquivos que não conseguem criar um conjunto de leitura durante um upload direto. Os locais alternativos estão disponíveis para lojas de sequências criadas após 15 de maio de 2023. Você especifica o local de fallback ao criar o armazenamento de sequências.

Você pode especificar até cinco chaves de tag do conjunto de leitura. Quando você cria ou atualiza um conjunto de leitura com uma chave de tag que corresponde a uma dessas chaves, as tags do conjunto de leitura são propagadas para o objeto Amazon S3 correspondente. As tags do sistema criadas por HealthOmics são propagadas por padrão.

Tópicos

- Criando um armazenamento de sequências usando o console

- Criando um armazenamento de sequências usando a CLI
- Atualizando um armazenamento de sequências
- Atualizando as tags do conjunto de leitura para um armazenamento de sequências
- Importação de arquivos genômicos

Criando um armazenamento de sequências usando o console

Para criar um repositório de sequências

1. Abra o console de HealthOmics .
2. Se necessário, abra o painel de navegação esquerdo (≡). Escolha Armazenamentos de sequências.
3. Na página Criar armazenamento de sequências, forneça as seguintes informações
   - Nome da loja de sequências - Um nome exclusivo para essa loja.
   - Descrição (opcional) - Uma descrição desse armazenamento de sequências.
4. Para a localização do Fallback no S3, especifique uma localização do Amazon S3. HealthOmics usa o local de fallback para armazenar qualquer arquivo que falhe na criação de um conjunto de leitura durante um upload direto. Você precisa conceder ao HealthOmics serviço acesso de gravação ao local de fallback do Amazon S3. Para visualizar um exemplo de política, consulte Configurar um local de fallback.

   Os locais alternativos não estão disponíveis para lojas de sequências criadas antes de 16 de maio de 2023.
5. (Opcional) Para chaves de tag do conjunto de leitura para propagação do S3, você pode inserir até cinco chaves do conjunto de leitura para propagar de um conjunto de leitura para os objetos do S3 subjacentes. Ao propagar tags de um conjunto de leitura para o objeto do S3, você pode conceder permissões de acesso ao S3 com base nas tags aos usuários and/or finais para ver as tags propagadas por meio da operação da API do Amazon S3. getObjectTagging
   a. Insira um valor-chave na caixa de texto. O console cria uma nova caixa de texto para adicionar a próxima chave.
   b. (Opcional) Escolha Remover para remover todas as chaves.
6. Em Criptografia de dados, selecione se você deseja que a criptografia de dados seja de propriedade e gerenciada por AWS ou use uma CMK gerenciada pelo cliente.

7. (Opcional) Em Acesso aos dados do S3, selecione se deseja criar uma nova função e política para acessar o armazenamento de sequências por meio do Amazon S3.
8. (Opcional) Para o registro de acesso do S3, selecione `Enabled` se você deseja que o Amazon S3 colete registros do registro de acesso.

   Para o local de registro de acesso no S3, especifique um local do Amazon S3 para armazenar os registros. Esse campo fica visível somente se você habilitou o registro de acesso do S3.
9. Tags (opcional) - forneça até 50 tags para esse armazenamento de sequências. Essas tags são separadas das tags do conjunto de leitura que são definidas durante a import/tag atualização do conjunto de leitura.

Depois de criar a loja, ela estará pronta paraImportação de arquivos genômicos.

Criando um armazenamento de sequências usando a CLI

No exemplo a seguir, *`sequence store name`* substitua pelo nome que você escolheu para seu armazenamento de sequências.

```
aws omics create-sequence-store --name sequence store name --fallback-location "s3://
amzn-s3-demo-bucket"
```

Você recebe a seguinte resposta em JSON, que inclui o número de identificação do seu armazenamento de sequências recém-criado.

```
{
    "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"
}
```

Você também pode visualizar todos os armazenamentos de sequências associados à sua conta usando o list-sequence-storescomando, conforme mostrado a seguir.

```
aws omics list-sequence-stores
```

Você recebe a seguinte resposta.

```
{
    "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"
        }
    ]
}
```

Você pode usar get-sequence-storepara saber mais sobre um armazenamento de sequências usando seu ID, conforme mostrado no exemplo a seguir:

```
aws omics get-sequence-store --id sequence store ID
```

Você recebe a seguinte resposta:

```
{
  "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"],
}
```

Após a criação, vários parâmetros da loja também podem ser atualizados. Isso pode ser feito por meio do console ou da `updateSequenceStore` operação da API.

Atualizando um armazenamento de sequências

Para atualizar um armazenamento de sequências, siga estas etapas:

1. Abra o console de HealthOmics .
2. Se necessário, abra o painel de navegação esquerdo (≡). Escolha Armazenamentos de sequências.
3. Escolha o armazenamento de sequências a ser atualizado.
4. No painel Detalhes, escolha Editar.
5. Na página Editar detalhes, você pode atualizar os seguintes campos:

   - Nome da loja de sequências - Um nome exclusivo para essa loja.
   - Descrição - Uma descrição desse armazenamento de sequências.
   - Local de fallback no S3, especifique um local do Amazon S3. HealthOmics usa o local de fallback para armazenar qualquer arquivo que falhe na criação de um conjunto de leitura durante um upload direto.
   - Chaves de tag do conjunto de leitura para propagação do S3 Você pode inserir até cinco chaves do conjunto de leitura para propagar para o Amazon S3.
   - (Opcional) Para o registro de acesso do S3, selecione `Enabled` se você deseja que o Amazon S3 colete registros do registro de acesso.

     Para o local de registro de acesso no S3, especifique um local do Amazon S3 para armazenar os registros. Esse campo fica visível somente se você habilitou o registro de acesso do S3.
   - Tags (opcional) - forneça até 50 tags para esse armazenamento de sequências.

Atualizando as tags do conjunto de leitura para um armazenamento de sequências

Para atualizar as tags do conjunto de leitura ou outros campos para um armazenamento de sequências, siga estas etapas:

1. Abra o console de HealthOmics .
2. Se necessário, abra o painel de navegação esquerdo (≡). Escolha Armazenamentos de sequências.
3. Escolha o repositório de sequências que você deseja atualizar.
4. Escolha a guia Detalhes.
5. Escolha Editar.
6. Adicione novas tags do conjunto de leitura ou exclua as tags existentes, conforme necessário.
7. Atualize o nome, a descrição, o local alternativo ou o acesso aos dados do S3, conforme necessário.
8. Escolha Salvar alterações.

Importação de arquivos genômicos

Para importar arquivos genômicos para um armazenamento de sequências, siga estas etapas:

Para importar um arquivo genômico

1. Abra o console de HealthOmics .
2. Se necessário, abra o painel de navegação esquerdo (≡). Escolha escolher Armazenamentos de sequências.
3. Na página Armazenamentos de sequências, escolha o repositório de sequências para o qual você deseja importar seus arquivos.
4. Na página de armazenamento de sequências individuais, escolha Importar arquivos genômicos.
5. Na página Especificar detalhes da importação, forneça as seguintes informações

   - Função do IAM - A função do IAM que pode acessar os arquivos genômicos no Amazon S3.
   - Genoma de referência - O genoma de referência para esses dados genômicos.
6. Na página Especificar manifesto de importação, especifique as informações a seguir: Arquivo de manifesto. O arquivo de manifesto é um arquivo JSON ou YAML que descreve informações

essenciais de seus dados genômicos. Para obter informações sobre o arquivo de manifesto, consulteImportação de conjuntos de leitura para um armazenamento de HealthOmics sequências.

7. Clique em Criar tarefa de importação.

Excluindo repositórios HealthOmics de referências e sequências

Tanto os armazenamentos de referência quanto os de sequência podem ser excluídos. Os armazenamentos de sequências só podem ser excluídos se não contiverem conjuntos de leitura, e os armazenamentos de referência só poderão ser excluídos se não contiverem referências. A exclusão de uma sequência ou armazenamento de referência também exclui todas as tags associadas a essa loja.

O exemplo a seguir mostra como excluir um repositório de referência usando AWS CLI o. Se a ação for bem-sucedida, você não receberá uma resposta. No exemplo a seguir, *reference store ID* substitua pelo ID da loja de referência.

```
aws omics delete-reference-store --id reference store ID
```

O exemplo a seguir mostra como excluir um armazenamento de sequências. Você não receberá uma resposta se a ação for bem-sucedida. No exemplo a seguir, *sequence store ID* substitua pelo seu ID de armazenamento de sequências.

```
aws omics delete-sequence-store --id sequence store ID
```

Você também pode excluir uma referência em um repositório de referência, conforme mostrado no exemplo a seguir. As referências só podem ser excluídas se não estiverem sendo usadas em um conjunto de leitura, armazenamento de variantes ou armazenamento de anotações. No exemplo a seguir, *reference store ID* substitua pelo ID da loja de referência e *reference ID* substitua pelo ID da referência que você deseja excluir.

```
aws omics delete-reference  --id reference ID --reference-store-id reference store ID
```

Importação de conjuntos de leitura para um armazenamento de HealthOmics sequências

Depois de criar seu armazenamento de sequências, crie trabalhos de importação para carregar conjuntos de leitura no armazenamento de dados. Você pode carregar seus arquivos de um bucket do Amazon S3 ou pode fazer upload diretamente usando as operações síncronas da API. Seu bucket do Amazon S3 deve estar na mesma região do seu armazenamento de sequências.

Você pode carregar qualquer combinação de conjuntos de leitura alinhados e não alinhados em seu armazenamento de sequências. No entanto, se algum dos conjuntos de leitura em sua importação estiver alinhado, você deverá incluir um genoma de referência.

Você pode reutilizar a política de acesso do IAM que você usou para criar o repositório de referência.

Os tópicos a seguir descrevem as principais etapas que você segue para importar um conjunto de leitura para seu armazenamento de sequências e, em seguida, obter informações sobre os dados importados.

Tópicos

Faça upload de arquivos para o Amazon S3

O exemplo a seguir mostra como mover arquivos para o bucket do 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
```

A amostra BAM e a CRAM usada neste exemplo requerem referências de genoma diferentes, Hg19 e. Hg38 Para saber mais ou acessar essas referências, consulte The Broad Genome References no Registro de Dados Abertos em AWS.

Criar um arquivo de manifesto

Você também deve criar um arquivo de manifesto em JSON para modelar o trabalho de importação `import.json` (veja o exemplo a seguir). Se você criar um armazenamento de sequências no console, não precisará especificar `sequenceStoreId` ou`roleARN`, portanto, seu arquivo de manifesto começa com a `sources` entrada.

API manifest

O exemplo a seguir importa três conjuntos de leitura usando a API: um FASTQBAM, um e umCRAM.

```
{
  "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

Esse código de exemplo é usado para importar um único conjunto de leitura usando o console.

```
[
  {
      "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, você pode carregar o arquivo de manifesto no formato YAML.

Iniciando o trabalho de importação

Para iniciar o trabalho de importação, use o AWS CLI comando a seguir.

```
aws omics start-read-set-import-job --cli-input-json file://import.json
```

Você recebe a seguinte resposta, que indica uma criação de emprego bem-sucedida.

```
{
  "id": "3660451514",
  "sequenceStoreId": "3936421177",
  "roleArn": "arn:aws:iam::111122223333:role/OmicsImport",
  "status": "CREATED",
  "creationTime": "2022-07-13T22:14:59.309Z"
}
```

Monitore o trabalho de importação

Depois que o trabalho de importação for iniciado, você poderá monitorar seu progresso com o comando a seguir. No exemplo a seguir, *`sequence store id`* substitua pela ID do armazenamento de sequências e *`job import ID`* substitua pela ID de importação.

```
aws omics get-read-set-import-job --sequence-store-id sequence store id --id job import
 ID
```

A seguir, são mostrados os status de todos os trabalhos de importação associados à ID de armazenamento de sequência especificada.


```
{
  "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"
        }
    ]
}
```

Encontre os arquivos de sequência importados

Depois que o trabalho for concluído, você poderá usar a operação da list-read-setsAPI para encontrar os arquivos de sequência importados. No exemplo a seguir, *`sequence store id`* substitua pelo seu ID de armazenamento de sequências.

```
aws omics list-read-sets --sequence-store-id sequence store id
```

Você recebe a seguinte resposta.


```
{
  "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"
            }
        }
    ]
}
```

Obtenha detalhes sobre um conjunto de leitura

Para ver mais detalhes sobre um conjunto de leitura, use a operação GetReadSetMetadatada API. No exemplo a seguir, *`sequence store id`* substitua pela ID do armazenamento de sequências e *`read set id`* substitua pela ID do conjunto de leitura.

```
aws omics get-read-set-metadata --sequence-store-id sequence store id --id read set id
```

Você recebe a seguinte resposta.

```
{
"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"
}
```

Baixe os arquivos de dados do conjunto de leitura

Você pode acessar os objetos de um conjunto de leitura ativo usando a operação de GetObject API do Amazon S3. O URI do objeto é retornado na resposta GetReadSetMetadatada API. Para obter mais informações, consulte Acessando conjuntos de HealthOmics leitura com o Amazon S3 URIs.

Como alternativa, use a operação HealthOmics GetReadSet da API. Você pode usar GetReadSet para baixar paralelamente baixando partes individuais. Essas peças são semelhantes às peças do Amazon S3. Veja a seguir um exemplo de como baixar a parte 1 de um conjunto de leitura. No exemplo a seguir, *`sequence store id`* substitua pela ID do armazenamento de sequências e *`read set id`* substitua pela ID do conjunto de leitura.

```
aws omics get-read-set --sequence-store-id sequence store id --id read set id  --part-
number 1 outfile.bam
```

Você também pode usar o Gerenciador HealthOmics de Transferências para baixar arquivos para um conjunto de HealthOmics referência ou leitura. Você pode baixar o HealthOmics Transfer Manager aqui. Para obter mais informações sobre como usar e configurar o Gerenciador de Transferências, consulte este GitHubRepositório.

Upload direto para um armazenamento HealthOmics de sequências

Recomendamos que você use o Gerenciador HealthOmics de Transferências para adicionar arquivos ao seu armazenamento de sequências. Para obter mais informações sobre como usar o Transfer Manager, consulte este GitHubRepositório. Você também pode carregar seus conjuntos de leitura diretamente para um armazenamento de sequências por meio das operações da API de upload direto.

Os conjuntos de leitura de upload direto existem primeiro no `PROCESSING_UPLOAD` estado. Isso significa que as partes do arquivo estão sendo carregadas no momento e você pode acessar os metadados do conjunto de leitura. Depois que as peças são carregadas e as somas de verificação são validadas, o conjunto de leitura se torna `ACTIVE` e se comporta da mesma forma que um conjunto de leitura importado.

Se o upload direto falhar, o status do conjunto de leitura será mostrado como`UPLOAD_FAILED`. Você pode configurar um bucket do Amazon S3 como um local alternativo para arquivos que falham no

upload. Os locais alternativos estão disponíveis para lojas de sequências criadas após 15 de maio de 2023.

Tópicos

Upload direto para um armazenamento de sequências usando o AWS CLI

Para começar, inicie um upload de várias partes. Você pode fazer isso usando o AWS CLI, conforme mostrado no exemplo a seguir.

Para direcionar o upload usando AWS CLI comandos

1. Crie as partes separando seus dados, conforme mostrado no exemplo a seguir.

```
split -b 100MiB SRR233106_1.filt.fastq.gz source1_part_
```

2. Depois que os arquivos de origem estiverem divididos em partes, crie um upload de conjunto de leitura com várias partes, conforme mostrado no exemplo a seguir. Substitua *sequence store ID* e os outros parâmetros pelo ID do armazenamento de sequências e outros 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"
```

   Você obtém os `uploadID` e outros metadados na resposta. Use o `uploadID` para a próxima etapa do processo de upload.

```
{
"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. Adicione seus conjuntos de leitura ao upload. Se o arquivo for pequeno o suficiente, você só precisará executar essa etapa uma vez. Para arquivos maiores, você executa essa etapa para cada parte do seu arquivo. Se você fizer o upload de uma nova peça usando um número de peça usado anteriormente, ele substituirá a peça carregada anteriormente.

   No exemplo a seguir, substitua *sequence store IDupload ID*, e os outros parâmetros pelos seus 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
   ```

   A resposta é uma ID que você pode usar para verificar se o arquivo carregado corresponde ao arquivo pretendido.

   ```
   {
   "checksum": "984979b9928ae8d8622286c4a9cd8e99d964a22d59ed0f5722e1733eb280e635"
   }
   ```

4. Continue carregando as partes do seu arquivo, se necessário. Para verificar se seus conjuntos de leitura foram carregados, use a operação de API list-read-set-upload-parts, conforme mostrado a seguir. No exemplo a seguir, substitua *sequence store ID upload ID*, e o *part source* por sua própria entrada.

   ```
   aws omics list-read-set-upload-parts \
    --sequence-store-id sequence store ID \
    --upload-id upload ID \
    --part-source SOURCE1
   ```

   A resposta retorna o número de conjuntos de leitura, o tamanho e a data e hora de quando foi atualizada mais recentemente.

```
{
"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 visualizar todos os uploads ativos de conjuntos de leitura de várias partes, use list-multipart-read-set-uploads, conforme mostrado a seguir. *`sequence store ID`*Substitua pelo ID do seu próprio armazenamento de sequências.

```
aws omics list-multipart-read-set-uploads --sequence-store-id
            sequence store ID
```

Essa API retorna apenas uploads de conjuntos de leitura em várias partes que estão em andamento. Depois que os conjuntos de leitura ingeridos `ACTIVE` terminarem, ou se o upload falhar, o upload não será retornado na resposta à API list-multipart-read-set-uploads. Para

visualizar conjuntos de leitura ativos, use a list-read-setsAPI. Um exemplo de resposta para list-multipart-read-set-uploads é mostrado a seguir.


```
{
"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. Depois de carregar todas as partes do seu arquivo, use complete-multipart-read-set-upload para concluir o processo de upload, conforme mostrado no exemplo a seguir. Substitua *sequence store IDupload ID*,, e o parâmetro para peças com seus próprios valores.

```
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"}]'
```

A resposta para complete-multipart-read-set-upload é o conjunto de leitura IDs para seus conjuntos de leitura importados.

```
{
"readSetId": "0000000001"
}
```

7. Para interromper o upload, use abort-multipart-read-set-upload com o ID de upload para finalizar o processo de upload. Substitua *sequence store ID* e *upload ID* por seus próprios valores de parâmetros.

```
aws omics abort-multipart-read-set-upload \
--sequence-store-id sequence store ID \
--upload-id upload ID
```

8. Depois que o upload for concluído, recupere seus dados do conjunto de leitura usando get-read-set, conforme mostrado a seguir. Se o upload ainda estiver sendo processado, get-read-setretornará metadados limitados e os arquivos de índice gerados não estarão disponíveis. Substitua *sequence store ID* e os outros parâmetros por sua própria entrada.

```
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 verificar os metadados, incluindo o status do seu upload, use a operação da get-read-set-metadataAPI.

```
aws omics get-read-set-metadata --sequence-store-id sequence store ID --id read set
 ID
```

A resposta inclui detalhes de metadados, como o tipo de arquivo, o ARN de referência, o número de arquivos e o tamanho das sequências. Também inclui o status. Os status possíveis são `PROCESSING_UPLOADACTIVE`, e. `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"
}
```


Configurar um local de fallback

Ao criar ou atualizar um armazenamento de sequências, você pode configurar um bucket do Amazon S3 como local alternativo para arquivos que falham no upload. As partes do arquivo desses

conjuntos de leitura são transferidas para o local de fallback. Os locais alternativos estão disponíveis para lojas de sequências criadas após 15 de maio de 2023.

Crie uma política de bucket do Amazon S3 para conceder acesso de HealthOmics gravação ao local de fallback do Amazon S3, conforme mostrado no exemplo a seguir:

```
{
    "Effect": "Allow",
    "Principal": {
        "Service": "omics.amazonaws.com"
    },
    "Action": "s3:PutObject",
    "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
}
```

Se o bucket do Amazon S3 para registros de fallback ou de acesso usar uma chave gerenciada pelo cliente, adicione as seguintes permissões à política de chaves:

```
 {
    "Sid": "Allow use of key",
    "Effect": "Allow",
    "Principal": {
        "Service": "omics.amazonaws.com"
    },
    "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey*"
    ],
    "Resource": "*"
}
```

Exportação de conjuntos de HealthOmics leitura para um bucket do Amazon S3

Você pode exportar conjuntos de leitura como um trabalho de exportação em lote para um bucket do Amazon S3. Para fazer isso, primeiro crie uma política do IAM que tenha acesso de gravação ao bucket, semelhante ao exemplo de política do IAM a seguir.

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"
  }
]
}
```

Depois que a política do IAM estiver em vigor, comece seu trabalho de exportação do conjunto de leitura. O exemplo a seguir mostra como fazer isso usando a operação da API start-read-

set-export-job. No exemplo a seguir, substitua todos os parâmetros*sequence store ID*, como*destination*,*role ARN*, e*sources*, pela sua 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
```

Você recebe a seguinte resposta com informações sobre o armazenamento da sequência de origem e o bucket 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"
}
```

Após o início do trabalho, você pode determinar seu status usando a operação da API get-read-set-export-job, conforme mostrado a seguir. Substitua o *sequence store ID* e *job ID* por seu ID de armazenamento de sequências e ID do trabalho, respectivamente.

```
aws omics get-read-set-export-job --id job-id --sequence-store-id sequence store ID
```

Você pode visualizar todos os trabalhos de exportação inicializados para um armazenamento de sequências usando a operação da API list-read-set-export-jobs, conforme mostrado a seguir. *sequence store ID*Substitua o pelo seu ID de armazenamento de sequências.

```
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"
  }
]
}
```

Além de exportar seus conjuntos de leitura, você também pode compartilhá-los usando o acesso ao Amazon URIs S3. Para saber mais, consulte Acessando conjuntos de HealthOmics leitura com o Amazon S3 URIs.

Acessando conjuntos de HealthOmics leitura com o Amazon S3 URIs

Você pode usar caminhos de URI do Amazon S3 para acessar seus conjuntos de leitura do armazenamento de sequências ativas.

Com o caminho de URI do Amazon S3, você pode usar as operações do Amazon S3 para listar, compartilhar e baixar seus conjuntos de leitura. O acesso por meio do S3 APIs acelera a colaboração e a integração de ferramentas, já que muitas ferramentas do setor já foram criadas para serem lidas a partir do S3. Além disso, você pode compartilhar o acesso ao S3 APIs com outras contas e fornecer acesso de leitura entre regiões aos dados.

HealthOmics não oferece suporte ao acesso URI do Amazon S3 a conjuntos de leitura arquivados. Quando você ativa um conjunto de leitura, ele é restaurado sempre para o mesmo caminho de URI.

Com os dados carregados nas HealthOmics lojas, como o URI do Amazon S3 é baseado nos pontos de acesso do Amazon S3, você pode se integrar diretamente às ferramentas padrão do setor que leem o Amazon S3, como as URIs seguintes:

- Aplicativos de análise visual, como o Integrative Genomics Viewer (IGV) ou o UCSC Genome Browser.
- Fluxos de trabalho comuns com extensões do Amazon S3, como CWL, WDL e Nextflow.
- Qualquer ferramenta que possa autenticar e ler a partir do ponto de acesso Amazon URIs S3 ou ler o Amazon S3 pré-assinado. URIs
- Utilitários do Amazon S3, como Mountpoint ou. CloudFront

O Amazon S3 Mountpoint possibilita que você use um bucket do Amazon S3 como um sistema de arquivos local. Para saber mais sobre o Mountpoint e instalá-lo para uso, consulte Mountpoint para Amazon S3.

CloudFront A Amazon é um serviço de rede de entrega de conteúdo (CDN) criado para alto desempenho, segurança e conveniência para desenvolvedores. Para saber mais sobre como usar a Amazon CloudFront, consulte a CloudFront documentação da Amazon. Para configurar um armazenamento CloudFront de sequências, entre em contato com a AWS HealthOmics equipe.

A conta raiz do proprietário dos dados está habilitada para as ações S3:GetObject, S3: e S3:List Bucket no prefixo de armazenamento de sequência. GetObjectTagging Para que um usuário na conta acesse os dados, você cria uma política do IAM e a anexa ao usuário ou à função. Para visualizar um exemplo de política, consulte Permissões para acesso a dados usando o Amazon S3 URIs.

Você pode usar as seguintes operações de API do Amazon S3 nos conjuntos de leitura ativos para listar e recuperar seus dados. Você pode acessar conjuntos de leitura arquivados por meio do Amazon URIs S3 depois que eles forem ativados.

- GetObject— Recupera um objeto do Amazon S3.
- HeadObject— A operação HEAD recupera metadados de um objeto sem retornar o objeto em si. Essa operação é útil se você quiser apenas os metadados de um objeto.
- ListObjects e ListObject v2 — Retorna alguns ou todos (até 1.000) dos objetos em um bucket.
- CopyObject— Cria uma cópia de um objeto que já está armazenado no Amazon S3. HealthOmicssuporta a cópia em um ponto de acesso do Amazon S3, mas não a gravação em um ponto de acesso.

HealthOmics os armazenamentos de sequências mantêm a identidade semântica dos arquivos por meio ETags de. Durante todo o ciclo de vida de um arquivo, o Amazon ETag S3, que é baseado na identidade bit a bit, pode mudar, HealthOmics ETag mas permanece o mesmo. Para saber mais, consulte HealthOmics ETags e proveniência dos dados.

Tópicos

- Estrutura de URI do Amazon S3 em armazenamento HealthOmics
- Usando IGV hospedado ou local para acessar conjuntos de leitura
- Usando Samtools ou HTSlib em HealthOmics
- Usando Mountpoint HealthOmics
- Usando CloudFront com HealthOmics

Estrutura de URI do Amazon S3 em armazenamento HealthOmics

Todos os arquivos com o Amazon S3 URIs têm tags `omics:subjectId` de `omics:sampleId` recursos. Você pode usar essas tags para compartilhar o acesso usando políticas do IAM por meio de um padrão como`"s3:ExistingObjectTag/omics:subjectId": "pattern desired"`.

A estrutura do arquivo é a seguinte:

`.../`*`account_id`*`/sequenceStore/`*`seq_store_id`*`/readSet/`*`read_set_id`*`/`*`files`*`.`

Para arquivos importados para armazenamentos de sequências do Amazon S3, o armazenamento de sequências tenta manter o nome da fonte original. Quando os nomes entram em conflito, o sistema anexa as informações do conjunto de leitura para garantir que os nomes dos arquivos sejam exclusivos. Por exemplo, para conjuntos de leitura fastq, se os dois nomes de arquivo forem iguais, para tornar os nomes exclusivos, `sourceX` é inserido antes de .fastq.gz ou .fq.gz. Para um upload direto, os nomes dos arquivos seguem os seguintes padrões:

- Para FASTQ— *`read_set_name`* _ .fastq.gz *`sourcex`*
- Para uBAM/BAM/CRAM —*`read_set_name`*. *`file extension`*com extensões de `.bam` ou`.cram`. Um exemplo é `NA193948.bam`.

Para conjuntos de leitura que são BAM ou CRAM, os arquivos de índice são gerados automaticamente durante o processo de ingestão. Para os arquivos de índice gerados, a extensão de índice adequada no final do nome do arquivo é aplicada. Tem o padrão *`<name of the Source the index is on>.<file index extension>.`* As extensões do índice são `.bai` ou`.crai`.

Usando IGV hospedado ou local para acessar conjuntos de leitura

IGV é um navegador de genoma usado para analisar arquivos BAM e CRAM. Ele requer o arquivo e o índice porque exibe apenas uma parte do genoma por vez. O IGV pode ser baixado e usado localmente, e há guias para criar um IGV hospedado na AWS. A versão pública da web não é suportada porque requer CORS.

O IGV local depende da AWS configuração local para acessar arquivos. Certifique-se de que a função usada nessa configuração tenha uma política anexada que habilite as GetObject permissões kms: Decrypt e s3: para o URI s3 dos conjuntos de leitura que estão sendo acessados. Depois disso, no IGV, você pode usar “Arquivo > carregar do URL” e colar o URI da fonte e do índice. Como alternativa, o presigned URLs pode ser gerado e usado da mesma maneira, o que ignorará a

configuração da AWS. Observe que o CORS não é compatível com o acesso ao URI do Amazon S3, portanto, solicitações que dependem do CORS não são suportadas.

O exemplo do AWS Hosted IGV depende do AWS Cognito para criar as configurações e permissões corretas dentro do ambiente. Certifique-se de que seja criada uma política que habilite as permissões KMS:DECRYPT e s3: GetObject para o URI do Amazon S3 dos conjuntos de leitura que estão sendo acessados e adicione essa política à função atribuída ao grupo de usuários do Cognito. Depois disso, no IGV, você pode usar “Arquivo > carregar do URL” e inserir o URI da fonte e do índice. Como alternativa, o presigned URLs pode ser gerado e usado da mesma maneira, o que ignora a configuração da AWS.

Observe que o armazenamento de sequências não aparecerá na guia “Amazon” porque ela exibe apenas buckets de sua propriedade na região em que o AWS perfil está configurado.

Usando Samtools ou HTSlib em HealthOmics

HTSlib é a biblioteca principal compartilhada por várias ferramentas, como Samtools, RSAMtools e outras. PySam Use a HTSlib versão 1.20 ou posterior para obter suporte contínuo para pontos de acesso do Amazon S3. Para versões mais antigas da HTSlib biblioteca, você pode usar as seguintes soluções alternativas:

- Defina a variável de ambiente para o host HTS Amazon S3 com:. `export HTS_S3_HOST="s3.region.amazonaws.com"`
- Gere uma URL pré-assinada para os arquivos que você deseja usar. Se um BAM ou CRAM estiver sendo usado, certifique-se de que um URL pré-assinado seja gerado tanto para o arquivo quanto para o índice. Depois disso, os dois arquivos podem ser usados com as bibliotecas.
- Use Mountpoint para montar o armazenamento de sequências ou ler o prefixo do conjunto no mesmo ambiente em que você está usando bibliotecas. HTSlib A partir daqui, os arquivos podem ser acessados usando caminhos de arquivo locais.

Usando Mountpoint HealthOmics

O Mountpoint for Amazon S3 é um cliente de arquivos simples e de alto rendimento para montar um bucket do Amazon S3 como um sistema de arquivos local. Com o Mountpoint for Amazon S3, seus aplicativos podem acessar objetos armazenados no Amazon S3 por meio de operações de arquivo, como abrir e ler. O Mountpoint for Amazon S3 traduz automaticamente essas operações em chamadas de API de objetos do Amazon S3, dando aos seus aplicativos acesso ao armazenamento elástico e à taxa de transferência do Amazon S3 por meio de uma interface de arquivo.

O Mountpoint pode ser instalado usando as instruções de instalação do Mountpoint. O Mountpoint usa o perfil da AWS que é local para a instalação e funciona em um nível de prefixo do Amazon S3. Certifique-se de que o perfil que está sendo usado tenha uma política que habilite as permissões s3:GetObject, s3: ListBucket e kms: Decrypt para o prefixo URI do Amazon S3 do (s) conjunto (s) de leitura ou armazenamento de sequências que está sendo acessado. Depois disso, o bucket pode ser montado usando o seguinte caminho:

```
mount-s3 access point arn local path to mount --prefix prefix to sequence store or read
 set --region region
```

Usando CloudFront com HealthOmics

CloudFront A Amazon é um serviço de rede de entrega de conteúdo (CDN) criado para oferecer alto desempenho, segurança e conveniência para desenvolvedores. Os clientes que desejam usar CloudFront devem trabalhar com a equipe de serviços para ativar a CloudFront distribuição. Trabalhe com sua equipe de contas para engajar a equipe HealthOmics de atendimento.

Ativando conjuntos de leitura em HealthOmics

Você pode ativar conjuntos de leitura que são arquivados com a operação da API start-read-set-activation-job ou por meio do AWS CLI, conforme mostrado no exemplo a seguir. Substitua o *sequence store ID* e *read set id* pelo ID do armazenamento de sequências e pelo conjunto de leitura 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
```

Você recebe uma resposta que contém as informações do trabalho de ativação, conforme mostrado a seguir.

```
{
    "id": "12345678",
    "sequenceStoreId": "1234567890",
    "status": "SUBMITTED",
    "creationTime": "2022-10-22T00:50:54.670000+00:00"
}
```

Depois que o trabalho de ativação for iniciado, você poderá monitorar seu progresso com a operação da API get-read-set-activation-job. Veja a seguir um exemplo de como usar o AWS CLI para verificar o status do seu trabalho de ativação. Substitua *job ID* e *sequence store ID* por seu ID de armazenamento de sequências e trabalho IDs, respectivamente.

```
aws omics get-read-set-activation-job --id job ID --sequence-store-id sequence store ID
```

A resposta resume o trabalho de ativação, conforme mostrado a seguir.

```
{
    "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."
        }
    ]
}
```

Você pode verificar o status de um trabalho de ativação com a operação get-read-set-metadatada API. Os status possíveis são `ACTIVE`ACTIVATING, e. `ARCHIVED` No exemplo a seguir, *sequence store ID* substitua pela ID do armazenamento de sequências e *read set ID* substitua pela ID do conjunto de leitura.

```
aws omics get-read-set-metadata --sequence-store-id sequence store ID --id read set ID
```

A resposta a seguir mostra que o conjunto de leitura está ativo.

```
{
    "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"
```

```
    }
}
```

Você pode visualizar todos os trabalhos de ativação do conjunto de leitura usando list-read-set-activation-jobs, conforme mostrado no exemplo a seguir. No exemplo a seguir, *sequence store ID* substitua pelo seu ID de armazenamento de sequências.

```
aws omics list-read-set-activation-jobs --sequence-store-id sequence store ID
```

Você recebe a seguinte resposta.

```
{
    "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 análises

> **Important**
>
> AWS HealthOmics lojas de variantes e lojas de anotações não estão mais abertas a novos clientes. Os clientes atuais podem continuar usando o serviço normalmente. Para obter mais informações, consulte AWS HealthOmics alteração na disponibilidade da loja de variantes e da loja de anotações.

HealthOmics a análise suporta o armazenamento e a análise de variantes e anotações genômicas. O Analytics fornece dois tipos de recursos de armazenamento: lojas de variantes e lojas de anotações. Você usa esses recursos para armazenar, transformar e consultar dados de variantes genômicas e dados de anotação. Depois de importar dados para um armazenamento de dados, você pode usar o Athena para realizar análises avançadas sobre os dados.

Você pode usar o HealthOmics console ou a API para criar e gerenciar lojas, importar dados e compartilhar dados analíticos da loja com colaboradores.

A variante armazena dados de suporte em formatos VCF, e a anotação armazena suporte TSV/CSV e formatos. GFF3 As coordenadas genômicas são representadas como intervalos semiabertos semicerrados baseados em zero. Quando seus dados estão no armazenamento de dados HealthOmics analíticos, o acesso aos arquivos VCF é gerenciado por meio AWS Lake Formation de. Em seguida, você pode consultar os arquivos VCF usando o Amazon Athena. As consultas devem usar o mecanismo de consulta Athena versão 3. Para ler mais sobre as versões do mecanismo de consulta do Athena, consulte a documentação do Amazon Athena.

Tópicos

- Consultando HealthOmics dados analíticos
- Compartilhamento HealthOmics de lojas de análise

Criação de lojas HealthOmics variantes

> **Important**
>
> AWS HealthOmics lojas de variantes e lojas de anotações não estão mais abertas a novos clientes. Os clientes atuais podem continuar usando o serviço normalmente. Para obter mais informações, consulte AWS HealthOmics alteração na disponibilidade da loja de variantes e da loja de anotações.

Os tópicos a seguir descrevem como criar armazenamentos de HealthOmics variantes usando o console e a API.

Tópicos

- Criação de um armazenamento de variantes usando o console
- Criação de um armazenamento de variantes usando a API

Criação de um armazenamento de variantes usando o console

Você pode criar um armazenamento de variantes usando o HealthOmics console.

1. Abra o console de HealthOmics .
2. Se necessário, abra o painel de navegação esquerdo (≡). Escolha lojas Variant.
3. Na página Criar loja de variantes, forneça as seguintes informações
   - Nome da loja variante: um nome exclusivo para essa loja.
   - Descrição (opcional) - Uma descrição desse armazenamento de variantes.
   - Genoma de referência - O genoma de referência para esse armazenamento de variantes.
   - Criptografia de dados - Escolha se você deseja que a criptografia de dados seja de propriedade e gerenciada por você AWS ou por você mesmo.
   - Tags (opcional) - forneça até 50 tags para esse armazenamento de variantes.
4. Escolha Criar loja de variantes.

Criação de um armazenamento de variantes usando a API

Use a operação de HealthOmics `CreateVariantStore` API para criar lojas de variantes. Você também pode realizar essa operação com AWS CLI o.

Para criar uma loja de variantes, você fornece um nome para a loja e o ARN de uma loja de referência. O armazenamento de variantes está pronto para ingerir dados quando seu status muda para PRONTO.

O exemplo a seguir usa o AWS CLI para criar um armazenamento de variantes.

```
aws omics create-variant-store --name myvariantstore \
    --reference referenceArn="arn:aws:omics:us-
west-2:555555555555:referenceStore/123456789/reference/5987565360"
```

Para confirmar a criação do seu armazenamento de variantes, você recebe a seguinte resposta.

```
{
    "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 saber mais sobre um armazenamento de variantes, use a get-variant-storeAPI.

```
aws omics get-variant-store --name myvariantstore
```

Você recebe a seguinte resposta.

```
{
    "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 todos os armazenamentos de variantes associados a uma conta, use a list-variant-storesAPI.

```
aws omics list-variant-stores
```

Você recebe uma resposta que lista todos os armazenamentos de variantes IDs, junto com seus status e outros detalhes, conforme mostrado no exemplo de resposta a seguir.

```
{
    "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
        }
    ]
}
```

Você também pode filtrar as respostas da list-variant-storesAPI com base em status ou outros critérios.

Os arquivos VCF importados para repositórios analíticos criados em ou após 15 de maio de 2023 definiram esquemas para anotações do Variant Effect Predictor (VEP). Isso facilita a consulta e a análise de dados VCF importados. A alteração não afeta as lojas criadas antes de 15 de maio de

2023, exceto se o `annotation fields` parâmetro estiver incluído na chamada da API ou da CLI. Para esses armazenamentos, o uso do `annotation fields` parâmetro fará com que a solicitação falhe.

Criação de trabalhos de importação de lojas HealthOmics variantes

> **Important**
>
> AWS HealthOmics lojas de variantes e lojas de anotações não estão mais abertas a novos clientes. Os clientes atuais podem continuar usando o serviço normalmente. Para obter mais informações, consulte AWS HealthOmics alteração na disponibilidade da loja de variantes e da loja de anotações.

O exemplo a seguir mostra como usar o AWS CLI para criar um trabalho de importação para um armazenamento 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"}
    ]
}
```

Para lojas criadas após 15 de maio de 2023, o exemplo a seguir mostra como adicionar o `--annotation-fields` parâmetro. Os campos de anotação são definidos com a importação.

```
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"
}
```

Use get-variant-import-jobpara verificar o status.

```
aws omics get-variant-import-job --job-id 08279950-a9e3-4cc3-9a3c-a574f9c9e229
```

Você receberá uma resposta JSON que mostra o status do seu trabalho de importação. As anotações VEP no VCF são analisadas em busca de informações armazenadas na coluna INFO como um par. ID/Value O ID padrão da coluna INFO de anotações do Ensembl Variant Effect Predictor é CSQ, mas você pode usar o `--annotation-fields` parâmetro para indicar um valor personalizado usado na coluna INFO. Atualmente, a análise é suportada para anotações VEP.

Para uma loja criada antes de 15 de maio de 2023 ou para arquivos VCF que não incluem a anotação VEP, a resposta não inclui nenhum campo de anotação.

```
{
    "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",
}
```

As anotações VEP que fazem parte dos arquivos VCF são armazenadas como um esquema predefinido com a estrutura a seguir. O campo extras pode ser usado para armazenar quaisquer campos VEP adicionais que não estejam incluídos no esquema padrão.

```
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>
    >>
>
```

A análise é executada com a abordagem de melhor esforço. Se a entrada do VEP não seguir as especificações padrão do VEP, ela não será analisada e a linha na matriz ficará vazia.

Para um novo armazenamento de variantes, a resposta para get-variant-import-jobincluiria os campos de anotação, conforme mostrado.

```
aws omics get-variant-import-job --job-id 08279950-a9e3-4cc3-9a3c-a574f9c9e229
```

Você recebe uma resposta JSON que mostra o status do seu trabalho de importação.

```
{
    "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"}
  }
}
```

Você pode usar list-variant-import-jobspara ver todos os trabalhos de importação e seus status.

```
aws omics list-variant-import-jobs --ids 7a1c67e3-b7f9-434d-817b-9c571fd63bea
```

A resposta contém as informações a seguir.

```
{
    "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"}
        }
    ]
  }
}
```

Se necessário, você pode cancelar um trabalho de importação com o comando a seguir.

```
aws omics cancel-variant-import-job
```

```
    --job-id edd7b8ce-xmpl-47e2-bc99-258cac95a508
```

Criação de HealthOmics lojas de anotações

> **Important**
>
> AWS HealthOmics lojas de variantes e lojas de anotações não estão mais abertas a novos clientes. Os clientes atuais podem continuar usando o serviço normalmente. Para obter mais informações, consulte AWS HealthOmics alteração na disponibilidade da loja de variantes e da loja de anotações.

Um armazenamento de anotações é um armazenamento de dados que representa um banco de dados de anotações, como um de um arquivo TSV, VCF ou GFF. Se o mesmo genoma de referência for especificado, os armazenamentos de anotações serão mapeados para o mesmo sistema de coordenadas dos armazenamentos de variantes durante uma importação. Os tópicos a seguir mostram como usar o HealthOmics console e AWS CLI criar e gerenciar repositórios de anotações.

Tópicos

Criando um repositório de anotações usando o console

Use o procedimento a seguir para criar repositórios de anotações com o HealthOmics console.

Para criar um repositório de anotações

1. Abra o console de HealthOmics .
2. Se necessário, abra o painel de navegação esquerdo (≡). Escolha Lojas de anotações.
3. Na página Armazenamentos de anotações, escolha Criar repositório de anotações.
4. Na página Criar repositório de anotações, forneça as seguintes informações
   - Nome da loja de anotações - Um nome exclusivo para essa loja.
   - Descrição (opcional) - Uma descrição desse genoma de referência.

- Formato de dados e detalhes do esquema - Selecione o formato do arquivo de dados e faça o upload da definição do esquema para essa loja.
- Genoma de referência - O genoma de referência para esta anotação.
- Criptografia de dados - Escolha se você deseja que a criptografia de dados seja de propriedade e gerenciada por você AWS ou por você mesmo.
- Tags (opcional) - forneça até 50 tags para esse repositório de anotações.

5. Escolha Criar repositório de anotações.

Criação de um armazenamento de anotações usando a API

O exemplo a seguir mostra como criar um armazenamento de anotações usando o. AWS CLI Para todas as operações AWS CLI e de API, você deve especificar o formato dos seus dados.

```
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
```

Você recebe a seguinte resposta para confirmar a criação do seu repositório de anotações.

```
{
           "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 saber mais sobre um armazenamento de anotações, use a get-annotation-storeAPI.

```
aws omics get-annotation-store --name my_annotation_store
```

Você recebe a seguinte resposta.

```
{
          "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 os repositórios de anotações associados a uma conta, use a operação da list-annotation-storesAPI.

```
aws omics list-annotation-stores
```

Você recebe uma resposta que lista todos os repositórios de anotações, junto com seus IDs status e outros detalhes, conforme mostrado no exemplo de resposta a seguir.

```
{
           "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"
               }
           ]
       }
```

Você também pode filtrar as respostas com base no status ou em outros critérios.

Criação de trabalhos de importação para lojas de HealthOmics anotações

> **Important**
>
> AWS HealthOmics lojas de variantes e lojas de anotações não estão mais abertas a novos clientes. Os clientes atuais podem continuar usando o serviço normalmente. Para obter mais informações, consulte AWS HealthOmics alteração na disponibilidade da loja de variantes e da loja de anotações.

Tópicos

Criação de um trabalho de importação de anotações usando a API

O exemplo a seguir mostra como usar o AWS CLI para iniciar um trabalho de importação de anotações.

```
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"}'
```

Os repositórios de anotações criados antes de 15 de maio de 2023 retornarão uma mensagem de erro se os campos de anotação forem incluídos. Eles não retornam a saída de nenhuma operação de API envolvida com trabalhos de importação do armazenamento de anotações.

Em seguida, você pode usar a operação da get-annotation-import-jobAPI e o `job ID` parâmetro para saber mais detalhes sobre o trabalho de importação de anotações.

```
aws omics get-annotation-import-job --job-id 9e4198fb-fa85-446c-9301-9b823a1a8ba8
```

Você recebe a seguinte resposta, incluindo os campos de anotação.

```
{
          "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 os trabalhos de importação do repositório de anotações, use. list-annotation-import-jobs

```
aws omics list-annotation-import-jobs --ids 9e4198fb-fa85-446c-9301-9b823a1a8ba8
```

A resposta inclui os detalhes e os status dos trabalhos de importação da loja de anotações.

```
{
          "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"}
        }
        ]
    }
```

Parâmetros adicionais para formatos TSV e VCF

Para os formatos TSV e VCF, há parâmetros adicionais que informam a API sobre como analisar sua entrada.

> **Important**
>
> Os dados de anotação CSV exportados com mecanismos de consulta retornam diretamente as informações da importação do conjunto de dados. Se os dados importados contiverem fórmulas ou comandos, o arquivo poderá estar sujeito à injeção de CSV. Portanto, arquivos exportados com mecanismos de consulta podem solicitar avisos de segurança. Para evitar atividades maliciosas, desative links e macros ao ler arquivos de exportação.

O analisador TSV também realiza operações básicas de bioinformática, como normalização à esquerda e padronização das coordenadas genômicas, listadas na tabela a seguir.

Tipo de formato	Description
Genérico	Arquivo de texto genérico. Sem informações genômicas.
`CHR_POS`	Posição inicial - 1, Adicione a posição final, que é igual `POS` a.
`CHR_POS_REF_ALT`	Contém informações de contagem, posição de 1 base, alelos ref e alt.
`CHR_START_END_REF_ALT_ONE_BASE`	Contém informações dos alelos contig, start, end, ref e alt. As coordenadas são baseadas em 1.

Tipo de formato	Description
CHR_START_END_ZERO_BASE	Contém as posições inicial, inicial e final. As coordenadas são baseadas em 0.
CHR_START_END_ONE_BASE	Contém as posições inicial, inicial e final. As coordenadas são baseadas em 1.
CHR_START_END_REF_ALT_ZERO_BASE	Contém informações dos alelos contig, start, end, ref e alt. As coordenadas são baseadas em 0.

Uma solicitação de armazenamento de anotações de importação de TSV se parece com o exemplo a seguir.

```
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"
        }
    }
}'
```

Criação de armazenamentos de anotações formatados em TSV

O exemplo a seguir cria um armazenamento de anotações usando um arquivo limitado por abas que contém um cabeçalho, linhas e comentários. As coordenadas são CHR_START_END_ONE_BASED e contêm o mapa HG19 genético da Sinopse do Mapa do Gene Humano do 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}]}'
```

Você pode importar arquivos com ou sem cabeçalho. Para indicar isso em uma solicitação de CLI, use`header=false`, conforme mostrado no exemplo de tarefa de importação a seguir.

```
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="#"}}'
```

O exemplo a seguir cria um armazenamento de anotações para um arquivo de cama. Um arquivo bed é um arquivo simples delimitado por tabulações. Neste exemplo, as colunas são cromossomo, início, fim e nome da região. As coordenadas são baseadas em zero e os dados não têm um cabeçalho.

```
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}]}'
```

Em seguida, você pode importar o arquivo bed para o armazenamento de anotações usando o seguinte comando da 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="#"}}'
```

O exemplo a seguir cria um armazenamento de anotações para um arquivo delimitado por abas que contém as primeiras colunas de um arquivo VCF, seguidas por colunas com informações de anotação. Ele contém as posições do genoma com informações sobre o cromossomo, alelos iniciais, de referência e alternativos, e contém um cabeçalho.

```
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}]}'
```

Em seguida, você importaria o arquivo para o armazenamento de anotações usando o seguinte comando da 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="#"}}'
```

O exemplo a seguir mostra como um cliente pode criar um repositório de anotações para um arquivo mim2gene. Um arquivo mim2gene fornece os links entre os genes no OMIM e outro identificador de gene. É delimitado por abas e contém comentários.

```
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}]}'
```

Em seguida, você pode importar dados para sua loja da seguinte maneira.

```
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="#"}}'
```

Iniciando trabalhos de importação formatados em VCF

Para arquivos VCF, há duas entradas adicionais `ignoreQualField` `eignoreFilterField`, que ignoram ou incluem esses parâmetros, conforme mostrado.

```
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
    }
   }'
```

Você também pode cancelar a importação de um repositório de anotações, conforme mostrado. Se o cancelamento for bem-sucedido, você não receberá uma resposta para essa AWS CLI chamada. No entanto, se o ID do trabalho de importação não for encontrado ou o trabalho de importação for concluído, você receberá uma mensagem de erro.

```
aws omics cancel-annotation-import-job --job-id edd7b8ce-xmpl-47e2-bc99-258cac95a508
```

> **Note**
>
> Seus metadados importam o histórico de trabalhos para get-annotation-import-job, get-variant-import-joblist-annotation-import-jobs, e list-variant-import-jobssão excluídos automaticamente após dois anos. Os dados da variante e da anotação importados não são excluídos automaticamente e permanecem nos seus armazenamentos de dados.

Criação de HealthOmics versões de armazenamento de anotações

> **Important**
>
> AWS HealthOmics lojas de variantes e lojas de anotações não estão mais abertas a novos clientes. Os clientes atuais podem continuar usando o serviço normalmente. Para obter mais informações, consulte AWS HealthOmics alteração na disponibilidade da loja de variantes e da loja de anotações.

Você pode criar novas versões de repositórios de anotações para coletar diferentes versões de seus bancos de dados de anotações. Isso ajuda você a organizar seus dados de anotação, que são atualizados regularmente.

Para criar uma nova versão de um repositório de anotações existente, use a create-annotation-store-versionAPI conforme mostrado no exemplo a seguir.

```
aws omics create-annotation-store-version \
     --name my_annotation_store \
```

```
    --version-name my_version
```

Você receberá a seguinte resposta com o ID da versão do armazenamento de anotações, confirmando que uma nova versão da sua anotação foi criada.

```
{
    "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 atualizar a descrição de uma versão do armazenamento de anotações, você pode usar update-annotation-store-versionpara adicionar atualizações a uma versão do armazenamento de anotações.

```
aws omics update-annotation-store-version \
    --name my_annotation_store \
    --version-name my_version \
    --description "New Description"
```

Você receberá a seguinte resposta, confirmando que a versão do armazenamento de anotações foi atualizada.

```
{
    "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 os detalhes de uma versão do armazenamento de anotações, use. get-annotation-store-version

```
aws omics get-annotation-store-version --name my_annotation_store --version-name
 my_version
```

Você receberá uma resposta com o nome da versão, status e outros detalhes.

```
{
     "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 visualizar todas as versões de um repositório de anotações, você pode usar list-annotation-store-versions, conforme mostrado no exemplo a seguir.

```
aws omics list-annotation-store-versions --name my_annotation_store
```

Você receberá uma resposta com as seguintes informações

```
{
  "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
    }
}
```

Se você não precisar mais de uma versão do armazenamento de anotações, poderá usá-la delete-annotation-store-versionspara excluir uma versão do armazenamento de anotações, conforme mostrado no exemplo a seguir.

```
aws omics delete-annotation-store-versions --name my_annotation_store --versions
 my_version
```

Se a versão da loja for excluída sem erros, você receberá a seguinte resposta.

```
{
  "errors": []
}
```

Se houver erros, você receberá uma resposta com os detalhes dos erros, conforme mostrado.

```
{
  "errors": [
    {
      "versionName": "my_version",
      "message": "Version with versionName: my_version was not found."
    }
  ]
}
```

Se você tentar excluir uma versão do armazenamento de anotações que tenha um trabalho de importação ativo, você receberá uma resposta com um erro, conforme mostrado.

```
{
```

```
  "errors": [
    {
      "versionName": "my_version",
      "message": "version has an inflight import running"
    }
  ]
}
```

Nesse caso, você pode forçar a exclusão da versão do armazenamento de anotações, conforme mostrado no exemplo a seguir.

```
aws omics delete-annotation-store-versions --name my_annotation_store --versions
 my_version --force
```

Excluindo lojas de HealthOmics análise

> **Important**
>
> AWS HealthOmics lojas de variantes e lojas de anotações não estão mais abertas a novos clientes. Os clientes atuais podem continuar usando o serviço normalmente. Para obter mais informações, consulte AWS HealthOmics alteração na disponibilidade da loja de variantes e da loja de anotações.

Quando você exclui uma variante ou um repositório de anotações, o sistema também exclui todos os dados importados nesse armazenamento e todas as tags associadas.

O exemplo a seguir mostra como excluir um armazenamento de variantes usando AWS CLI o. Se a ação for bem-sucedida, o status do armazenamento de variantes mudará para`DELETING`.

```
aws omics delete-variant-store --id <variant-store-id>
```

O exemplo a seguir mostra como excluir um repositório de anotações. Se a ação for bem-sucedida, o status do armazenamento de anotações mudará para. `DELETING` Os repositórios de anotações não podem ser excluídos se houver mais de uma versão.

```
aws omics delete-annotation-store --id <annotation-store-id>
```

Consultando HealthOmics dados analíticos

> **Important**
>
> AWS HealthOmics lojas de variantes e lojas de anotações não estão mais abertas a novos clientes. Os clientes atuais podem continuar usando o serviço normalmente. Para obter mais informações, consulte AWS HealthOmics alteração na disponibilidade da loja de variantes e da loja de anotações.

Você pode realizar consultas em seus armazenamentos de variantes usando o AWS Lake Formation Amazon Athena ou o Amazon EMR. Antes de executar qualquer consulta, conclua os procedimentos de configuração (descritos nas seções a seguir) para Lake Formation e Amazon Athena.

Para obter informações sobre o Amazon EMR, consulte Tutorial: Introdução ao Amazon EMR

Para lojas de variantes criadas após 26 de setembro de 2024, HealthOmics particiona a loja por ID de amostra. Esse particionamento significa que HealthOmics usa o ID de amostra para otimizar o armazenamento das informações da variante. As consultas que usam informações de amostra como filtros retornarão os resultados mais rapidamente, pois a consulta digitaliza menos dados.

HealthOmics usa amostra IDs como nomes de arquivo de partição. Antes de ingerir dados, verifique se o ID da amostra contém dados de PHI. Se isso acontecer, altere a ID da amostra antes de ingerir os dados. Para obter mais informações sobre qual conteúdo incluir e não incluir na amostra IDs, consulte as orientações na página da web de conformidade com a AWS HIPAA.

Tópicos

Configurando o Lake Formation para usar HealthOmics

> **Important**
>
> AWS HealthOmics lojas de variantes e lojas de anotações não estão mais abertas a novos clientes. Os clientes atuais podem continuar usando o serviço normalmente. Para obter mais

informações, consulte AWS HealthOmics alteração na disponibilidade da loja de variantes e da loja de anotações.

Antes de usar o Lake Formation para gerenciar armazenamentos de HealthOmics dados, execute os seguintes procedimentos de configuração do Lake Formation.

Tópicos

- Criando ou verificando administradores do Lake Formation
- Criação de links de recursos usando o console Lake Formation
- Configurando permissões para compartilhamentos AWS RAM de recursos

Criando ou verificando administradores do Lake Formation

Antes de criar um data lake no Lake Formation, você define um ou mais administradores.

Os administradores são usuários e funções com permissões para criar links de recursos. Você configura administradores de data lake por conta por região.

Crie um usuário administrador no console do Lake Formation

1. Abra o console do AWS Lake Formation: console do Lake Formation
2. Se o console exibir o painel Welcome to Lake Formation, escolha Começar.

   O Lake Formation adiciona você à tabela de administradores do Data Lake.
3. Caso contrário, no menu à esquerda, escolha Funções e tarefas administrativas.
4. Adicione outros administradores conforme necessário.

Criação de links de recursos usando o console Lake Formation

Para criar um recurso compartilhado que os usuários possam consultar, os controles de acesso padrão devem estar desativados. Para saber mais sobre como desativar os controles de acesso padrão, consulte Alteração das configurações de segurança padrão do seu data lake na documentação do Lake Formation. Você pode criar links de recursos individualmente ou em grupo, para poder acessar dados no Amazon Athena ou em outros AWS serviços (como o Amazon EMR).

Criar links de recursos no console do AWS Lake Formation e compartilhá-los com os usuários do HealthOmics Analytics

1. Abra o console do AWS Lake Formation: console do Lake Formation
2. Na barra de navegação principal, escolha Bancos de dados.
3. Na tabela Bancos de dados, selecione o banco de dados desejado.
4. No menu Criar, escolha Link do recurso.
5. Insira o nome do link do recurso. Se você planeja acessar o banco de dados a partir do Athena, insira um nome usando somente letras minúsculas (até 256 caracteres).
6. Escolha Criar.
7. O novo link do recurso agora está listado em Bancos de dados.

Conceda acesso ao recurso compartilhado usando o console do Lake Formation

Um administrador de banco de dados do Lake Formation pode conceder acesso ao recurso compartilhado usando o procedimento a seguir.

1. Abra o console do AWS Lake Formation: https://console.aws.amazon.com/lakeformation/
2. Na barra de navegação principal, escolha Bancos de dados.
3. Na página Bancos de dados, selecione o link do recurso que você criou anteriormente.
4. No menu Ações, escolha Conceder no alvo.
5. Na página Conceder permissões de dados, em Diretores, escolha usuários ou funções do IAM.
6. No menu suspenso Usuários ou funções do IAM, encontre o usuário ao qual você deseja conceder acesso.
7. Em seguida, em LF-Tags ou cartão de recursos do catálogo, selecione a opção Recursos do catálogo de dados nomeados.
8. No menu suspenso Tabelas opcionais, selecione Todas as tabelas ou a tabela que você criou anteriormente.
9. No cartão Permissões da tabela, em Permissões da tabela, escolha Descrever e selecionar.
10. Em seguida, escolha Grant.

Para visualizar as permissões do Lake Formation, escolha Permissões do Data lake no painel de navegação principal. A tabela mostra os bancos de dados e links de recursos disponíveis.

Configurando permissões para compartilhamentos AWS RAM de recursos

No console do AWS Lake Formation, visualize as permissões escolhendo Permissões do Data lake na barra de navegação principal. Na página Permissões de dados, você pode ver uma tabela que mostra os tipos de recursos, bancos de dados e **ARN** que está relacionada a um recurso compartilhado em Compartilhamento de recursos de RAM. Se você precisar aceitar um compartilhamento de recursos AWS Resource Access Manager (AWS RAM), AWS Lake Formation notificará você no console.

HealthOmics pode aceitar implicitamente os compartilhamentos AWS RAM de recursos durante a criação da loja. Para aceitar o compartilhamento AWS RAM de recursos, o usuário ou a função do IAM que chama as operações da `CreateAnnotationStore` API `CreateVariantStore` ou da API deve permitir as seguintes ações:

- `ram:GetResourceShareInvitations`- Esta ação permite HealthOmics encontrar os convites.
- `ram:AcceptResourceShareInvitation`- Esta ação permite HealthOmics aceitar o convite usando um token FAS.

Sem essas permissões, você vê um erro de autorização durante a criação da loja.

Aqui está um exemplo de política que inclui essas ações. Adicione essa política ao usuário ou função do IAM que aceita o compartilhamento AWS RAM de recursos.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "omics:*",
        "ram:AcceptResourceShareInvitation",
        "ram:GetResourceShareInvitations"
      ],
      "Resource": "*"
    }
  ]
}
```

Configurando o Athena para consultas

> **Important**
>
> AWS HealthOmics lojas de variantes e lojas de anotações não estão mais abertas a novos clientes. Os clientes atuais podem continuar usando o serviço normalmente. Para obter mais informações, consulte AWS HealthOmics alteração na disponibilidade da loja de variantes e da loja de anotações.

Você pode usar o Athena para consultar variantes e anotações. Antes de executar qualquer consulta, execute as seguintes tarefas de configuração:

Tópicos

Configurar um local de resultados de consulta usando o console Athena

Para configurar a localização dos resultados da consulta, siga estas etapas.

1. Abra o console Athena: console Athena
2. Na barra de navegação principal, escolha Editor de consultas.
3. No editor de consultas, escolha a guia Configurações e, em seguida, escolha Gerenciar.
4. Insira um prefixo S3 de um local para salvar o resultado da consulta.

Configurar um grupo de trabalho com o Athena engine v3

Para configurar um grupo de trabalho, siga estas etapas.

1. Abra o console Athena: console Athena
2. Na barra de navegação principal, escolha Grupos de trabalho e, em seguida, Criar grupo de trabalho.
3. Insira um nome para o grupo de trabalho.
4. Selecione Athena SQL como o tipo de mecanismo.
5. Em Atualizar mecanismo de consulta, selecione Manual.

6. Em Mecanismo de versão do Query, selecione Athena versão 3.
7. Escolha Create workgroup (Criar grupo de trabalho).

Executando consultas em lojas de HealthOmics variantes

> **Important**
>
> AWS HealthOmics lojas de variantes e lojas de anotações não estão mais abertas a novos clientes. Os clientes atuais podem continuar usando o serviço normalmente. Para obter mais informações, consulte AWS HealthOmics alteração na disponibilidade da loja de variantes e da loja de anotações.

Você pode realizar consultas na sua loja de variantes usando o Amazon Athena. Observe que as coordenadas genômicas nos armazenamentos de variantes e anotações são representadas como intervalos semiabertos semifechados com base em zero.

Execute uma consulta simples usando o console Athena

O exemplo a seguir mostra como executar uma consulta simples.

1. Abra o editor Athena Query: editor Athena Query
2. Em Grupo de trabalho, selecione o grupo de trabalho que você criou durante a configuração.
3. Verifique se a fonte de dados é AwsDataCatalog.
4. Em Database, selecione o link do recurso de banco de dados que você criou durante a configuração do Lake Formation.
5. Copie a consulta a seguir no Editor de consultas na guia Consulta 1:

```
SELECT * from omicsvariants limit 10
```

6. Para executar a consulta, escolha Executar. O console preenche a tabela de resultados com as primeiras 10 linhas da omicsvariants tabela.

Execute uma consulta complexa usando o console Athena

O exemplo a seguir mostra como executar uma consulta complexa. Para executar essa consulta, importe `ClinVar` para o armazenamento de anotações.

Execute uma consulta complexa

1. Abra o editor Athena Query: editor Athena Query
2. Em Grupo de trabalho, selecione o grupo de trabalho que você criou durante a configuração.
3. Verifique se a fonte de dados é AwsDataCatalog.
4. Em Database, selecione o link do recurso de banco de dados que você criou durante a configuração do Lake Formation.
5. Escolha a opção + no canto superior direito para criar uma nova guia de consulta chamada Consulta 2.
6. Copie a consulta a seguir no Editor de consultas na guia 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. Escolha Executar para começar a executar a consulta.

Compartilhamento HealthOmics de lojas de análise

> **Important**
>
> AWS HealthOmics lojas de variantes e lojas de anotações não estão mais abertas a novos clientes. Os clientes atuais podem continuar usando o serviço normalmente. Para obter mais informações, consulte AWS HealthOmics alteração na disponibilidade da loja de variantes e da loja de anotações.

Como proprietário de uma loja de variantes ou de uma loja de anotações, você pode compartilhar a loja com outras contas da AWS. O proprietário pode revogar o acesso ao recurso compartilhado excluindo o compartilhamento.

Como assinante de uma loja compartilhada, você primeiro aceita o compartilhamento. Em seguida, você pode definir fluxos de trabalho que usam a loja compartilhada. Os dados aparecem como uma tabela em Lake Formation AWS Glue e em Lake Formation.

Quando você não precisa mais acessar a loja, você exclui o compartilhamento.

Consulte Compartilhamento de recursos entre contas em AWS HealthOmics para obter informações adicionais sobre compartilhamento de recursos.

Criação de um compartilhamento na loja

Para criar um compartilhamento de loja, use a operação da API create-share. O assinante principal é Conta da AWS o usuário que assinará o compartilhamento. O exemplo a seguir cria um compartilhamento para um armazenamento de variantes. Para compartilhar uma loja com mais de uma conta, você cria vários compartilhamentos da mesma loja.

```
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"
```

Se a criação for bem-sucedida, você receberá uma resposta com o ID e o status do compartilhamento.

```
{
       "shareId": "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a",
       "name": "my_Share-123",
       "status": "PENDING"
  }
```

O compartilhamento permanece em estado pendente até que o assinante o aceite usando a operação da API accept-share.

Compartilhamento de recursos entre contas em AWS HealthOmics

Use o compartilhamento entre contas para compartilhar recursos com colaboradores sem criar cópias ou modificar as políticas de recursos do IAM. Os recursos a seguir oferecem suporte ao compartilhamento entre contas:

- HealthOmics lojas variantes
- HealthOmics lojas de anotações
- Fluxos de trabalho privados

O compartilhamento de um recurso inclui as seguintes etapas:

1. O proprietário do recurso cria um compartilhamento e especifica o ARN do recurso e o do assinante Conta da AWS pretendido. O compartilhamento de recursos permanece em estado pendente até que o assinante aceite o compartilhamento.
2. O assinante aceita o compartilhamento de recursos para ter acesso ao recurso. O compartilhamento de recursos passa para o estado de ativação.
3. O HealthOmics serviço fornece acesso ao recurso à conta do assinante.
4. O proprietário do recurso pode excluir o compartilhamento ou o assinante pode revogar seu acesso ao compartilhamento. O assinante não pode excluir o compartilhamento ou o recurso associado.

Tópicos

Criando um compartilhamento

Você pode usar a operação da API create-share para criar um compartilhamento. O assinante principal é Conta da AWS o usuário que assinará o recurso compartilhado. O exemplo a seguir cria um compartilhamento para um armazenamento 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"
```

Se a criação for bem-sucedida, você receberá uma resposta com o ID e o status do compartilhamento.

```
{
"shareId": "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a",
     "name": "my_Share-123",
     "status": "PENDING"
}
```

O compartilhamento permanece pendente até que o assinante o aceite usando a operação da accept-share API.

```
 aws omics accept-share \
    --share-id "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a"
```

Depois que o assinante aceita o compartilhamento, o compartilhamento passa para o estado ativo.

```
{
"status": "ACTIVATING"
}
```

Recuperar informações sobre um compartilhamento

Use a operação da API get-share para recuperar informações sobre o compartilhamento.

```
aws omics get-share --share-id "495c21bedc889d07d0ab69d710a6841e-
dd75ab7a1a9c384fa848b5bd8e5a7e0a"
```

A resposta da API inclui informações de metadados sobre o compartilhamento.

```
{
  "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"
    }
}
```

Veja as ações que você possui

Use a API list-shares para recuperar informações sobre cada um dos compartilhamentos que você possui.

```
aws omics list-shares  --resource-owner SELF
```

A resposta da API inclui os metadados de cada compartilhamento que você possui.

Exibir ações aceitas de outras contas

Use a API list-shares para ver todos os compartilhamentos que você aceitou de outras contas.

```
aws omics list-shares  --resource-owner OTHER
```

A resposta da API inclui os metadados de cada compartilhamento que você aceitou.

Excluir um compartilhamento

Use a API delete-share para excluir um compartilhamento depois que você não precisar mais dele.

```
aws omics delete-share \
    --share-id "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a"
```

Marcando recursos em HealthOmics

Tópicos

Aviso importante

HealthOmics protege os dados do cliente de acordo com as políticas do Modelo de Responsabilidade Compartilhada da AWS. Isso significa que todos os dados do cliente são criptografados em transição e em repouso. No entanto, nem todos os nomes inseridos pelo cliente para recursos, como armazenamentos de dados ou operações baseadas em tarefas, são criptografados. Eles nunca devem conter informações de identificação pessoal ou informações de saúde protegidas. Para obter mais informações, consulte Segurança na AWS HealthOmics.

Recursos de marcação HealthOmics

Você pode atribuir metadados aos seus recursos da AWS usando tags. Cada tag é um rótulo que consiste em um valor e uma chave definida pelo usuário. As tags podem ajudar você a gerenciar, identificar, organizar, pesquisar e filtrar recursos da .

Este tópico descreve as categorias de marcação normalmente usadas e as estratégias para ajudar você a implementar uma estratégia de marcação eficiente e consistente. As seções a seguir pressupõem conhecimento básico dos recursos da AWS, marcação, faturamento detalhado e. AWS Identity and Access Management

Cada tag da tem duas partes:

- Uma chave de tag (por exemplo CostCenter, Ambiente ou Projeto). As chaves de tag diferenciam maiúsculas de minúsculas

- Um valor de tag (por exemplo, 111122223333 ou Produção). Assim como as chaves de tag, os valores de tag diferenciam maiúsculas de minúsculas.

É possível usar tags para categorizar recursos por finalidade, proprietário, ambiente ou outros critérios. Para obter mais informações, consulte Estratégias de marcação da AWS.

Você pode adicionar, alterar ou remover tags de um recurso do console de serviço do recurso, da API de serviço ou do AWS CLI.

Para ativar a marcação, certifique-se de que TagResources está autorizado. Você pode autorizar TagResources anexando uma política do IAM, como no exemplo a seguir.

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": "*"
```

```
        }
    ]
}
```

Práticas recomendadas

Ao criar uma estratégia de marcação para os recursos da AWS, siga as melhores práticas:

- Não armazene Informações de Identificação Pessoal (PII), Informações de Saúde Protegidas (PHI) ou outras informações confidenciais em tags.
- Use um formato padronizado que diferencia maiúsculas de minúsculas para tags e aplique-o de forma consistente a todos os tipos de recursos.
- Considere as diretrizes de tags que oferecem suporte a diversas finalidades, como gerenciar o controle de acesso a recursos, o rastreamento de custos, a automação e a organização.
- Use ferramentas automatizadas para ajudar a gerenciar as tags de recursos. O AWS Resource Groups e a API Resource Groups Tagging permitem o controle programático de tags, possibilitando gerenciar, pesquisar e filtrar automaticamente tags e recursos.
- A marcação é mais eficaz quando você usa mais tags.
- As tags podem ser editadas ou modificadas conforme as necessidades do usuário mudam. No entanto, para atualizar as tags de controle de acesso, você também deve atualizar as políticas que fazem referência a essas tags para controlar o acesso aos seus recursos.

Requisitos de marcação

As tags têm os seguintes requisitos:

- As chaves não podem ser prefixadas com aws:.
- As chaves devem ser exclusivas por conjunto de tags.
- Uma chave deve ter entre 1 e 128 caracteres permitidos.
- Um valor deve ter entre 0 e 256 caracteres permitidos.
- Os valores não precisam ser exclusivos por conjunto de tags.
- Os caracteres permitidos para chaves e valores são letras Unicode, dígitos, espaço em branco e qualquer um dos seguintes símbolos: _ . : / = + - @.
- As chaves e os valores diferenciam letras maiúsculas de minúsculas.

Sequência armazena etiquetas de conjunto de leitura

Para armazenamentos de sequências, as tags criadas no conjunto de leitura estão no nível do recurso do conjunto de leitura. Os conjuntos de leitura também contêm objetos abaixo deles que podem ser acessados, pesquisados e restringidos usando o S3 APIs. Por padrão, a ID da amostra (omics:sampleID) e a ID do assunto (omics:subjectID) são adicionadas ao objeto.

Além disso, até cinco tags podem ser sincronizadas entre o conjunto de leitura e os objetos abaixo dele. A configuração para quais tags sincronizar é uma configuração em nível de loja definida durante a criação ou atualização da loja usando o `propogatedSetLevelTags` parâmetro.

Se já houver dados na loja, a atualização das chaves pode levar algum tempo. Durante essa atualização, HealthOmics altera o status da loja paraUpdating. Ao concluir, HealthOmics define o status da loja comoActive. Enquanto as tags estão se propagando, as permissões que dependem delas podem não ser aplicadas. As permissões serão aplicadas após a conclusão da propagação da tag.

Quando as tags são definidas ou atualizadas no conjunto de leitura, o sistema decide se deseja atualizar os objetos desse conjunto de leitura, com base na configuração do armazenamento.

Adicionar uma tag a um HealthOmics recurso

Adicionar tags a um recurso pode ajudar você a identificar e organizar seus recursos da AWS e gerenciar o acesso a eles. Primeiro, você adiciona uma ou mais tags (pares de valores-chave) a um recurso. Você pode usar até 50 tags por recurso. Também há restrições nos caracteres que você pode usar nos campos de chave e valor.

Depois de adicionar tags, você pode criar políticas do IAM para gerenciar o acesso ao AWS recurso com base nessas tags. Você pode usar o HealthOmics console ou o AWS CLI para adicionar tags a um recurso. Adicionar tags a um repositório pode afetar o acesso ele. Antes de adicionar uma tag a um armazenamento de dados, revise todas as políticas do IAM que possam usar tags para controlar o acesso a recursos, como armazenamentos de dados.

As etiquetas de serviço são geradas automaticamente para um assunto e uma ID de amostra para armazenamentos de sequências.

Siga estas etapas para usar o AWS CLI para adicionar uma tag a um HealthOmics recurso.
Por exemplo, para adicionar tags a um armazenamento de sequência enquanto ele está sendo criado, você usaria o seguinte comando no AWS CLI. O nome do armazenamento de sequência é

MySequenceStore, e as duas tags adicionadas com chaves são chave1 e chave2 com valores como valor1 e valor2, respectivamente :

```
aws omics create-sequence-store --name "MySequenceStore" --tags key1=value1,key2=value2
```

A saída não lista as tags. Ele retorna a seguinte resposta.

```
{
    "id": "6860403586",
    "referenceStoreId": "4889894479",
    "roleArn": "arn:aws:iam::555555555555:role/ImportTest",
    "status": "CREATED",
    "creationTime": "2022-07-21T01:19:07.194Z"
}
```

Para adicionar tags a um recurso existente, você executaria o comando de exemplo a seguir.

```
aws omics tag-resource --resource-arn arn:aws:omics:us-
west-2:555555555555:sequenceStore/2275234794 --tags key1=value1,key2=value2
```

Se houver êxito, o comando não retornará resposta nenhuma.

Listar as tags de um recurso

Siga estas etapas para usar o AWS CLI para ver uma lista das AWS tags de um HealthOmics recurso. Se não foram adicionadas tags, a lista retornará vazia.

No terminal ou na linha de comando, execute o list-tags-for-resource comando conforme mostrado no exemplo a seguir.

```
aws omics list-tags-for-resource --resource-arn arn:aws:omics:us-
west-2:555555555555:sequenceStore/2275234794
```

Você receberá uma lista de tags em resposta, no formato JSON.

```
 {
    "tags": {
        "key1": "value1",
```

```
        "key2": "value2"
    }
}
```

Removendo tags de um armazenamento de dados

Você pode remover uma ou mais tags associadas a um recurso. A remoção de uma tag não exclui a tag de outros recursos da AWS associados a essa tag.

No terminal ou na linha de comando, execute o comando untag-resource, especificando o Amazon Resource Name (ARN) do recurso do qual você deseja remover as tags e a chave da tag que você deseja remover.

```
aws omics untag-resource --resource-arn arn:aws:omics:us-
west-2:555555555555:sequenceStore/2275234794 --tag-keys key1,key2
```

Se for bem-sucedido, esse comando não retornará uma resposta. Para verificar as tags associadas ao recurso, execute o comando list-tags-for-resource.

Permissões do IAM para HealthOmics

Você pode usar AWS Identity and Access Management (IAM) para gerenciar o acesso à HealthOmics API e aos recursos, como lojas e fluxos de trabalho. Para usuários e aplicativos em sua conta que usam HealthOmics, você gerencia as permissões em uma política de permissões que pode ser aplicada aos usuários, grupos ou funções do IAM.

Para gerenciar permissões para usuários e aplicativos em suas contas, use as políticas que HealthOmics fornecem ou crie suas próprias. O HealthOmics console usa vários serviços para obter informações sobre a configuração e os acionadores da sua função. Você pode usar as políticas fornecidas no estado em que se encontram ou como ponto de partida para políticas mais restritivas.

HealthOmics usa funções de serviço do IAM para acessar outros serviços em seu nome. Por exemplo, você criaria ou escolheria uma função de serviço ao executar um fluxo de trabalho que lê dados do Amazon S3. Para alguns recursos, você também precisa configurar permissões em recursos em outros serviços. Analise esses requisitos antes de começar a trabalhar com HealthOmics

Para obter mais informações sobre o IAM, consulte O que é o IAM? no Manual do usuário do IAM.

Tópicos

Políticas de IAM baseadas em identidade para HealthOmics

Para conceder acesso aos usuários da sua conta HealthOmics, você usa políticas baseadas em identidade no AWS Identity and Access Management (IAM). As políticas baseadas em identidade podem ser aplicadas diretamente aos usuários do IAM ou aos grupos e funções do IAM associados a um usuário. Também é possível conceder a usuários em outra conta permissão para assumir uma função na conta e acessar os recursos do HealthOmics.

Para conceder permissão aos usuários para realizar ações em uma versão do fluxo de trabalho, você deve adicionar o fluxo de trabalho e a versão específica do fluxo de trabalho à lista de recursos.

A política do IAM a seguir permite que um usuário acesse todas as ações HealthOmics da API e transmita funções de serviço para HealthOmics o.

Example Política de usuário

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "omics:*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "iam:PassRole"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": "omics.amazonaws.com"
        }
      }
    }
  ]
}
```

Quando você usa HealthOmics, você também interage com outros AWS serviços. Para acessar esses serviços, use as políticas gerenciadas fornecidas por cada serviço. Para restringir o acesso a um subconjunto de recursos, você pode usar as políticas gerenciadas como ponto de partida para criar suas próprias políticas mais restritivas.

- AmazonS3 FullAccess — Acesso aos buckets e objetos do Amazon S3 usados por trabalhos.

- Amazon EC2 ContainerRegistryFullAccess — Acesso aos registros e repositórios do Amazon ECR para imagens de contêineres de fluxo de trabalho.
- AWSLakeFormationDataAdmin— Acesso aos bancos de dados e tabelas do Lake Formation criados por lojas de análise.
- ResourceGroupsandTagEditorFullAccess— Marque HealthOmics recursos com operações de API de HealthOmics marcação.

As políticas anteriores não permitem que um usuário crie funções do IAM. Para que um usuário com essas permissões execute um trabalho, um administrador deve criar a função de serviço que conceda HealthOmics permissão para acessar fontes de dados. Para obter mais informações, consulte Funções de serviço para AWS HealthOmics.

Defina permissões personalizadas do IAM para execuções

Você pode incluir qualquer fluxo de trabalho, execução ou grupo de execução referenciado pela `StartRun` solicitação em uma solicitação de autorização. Para fazer isso, liste a combinação desejada de fluxos de trabalho, execuções ou grupos de execução na política do IAM. Por exemplo, você pode limitar o uso de um fluxo de trabalho a uma execução específica ou a um grupo de execução. Você também pode especificar que um fluxo de trabalho seja usado somente com um grupo de execução.

Veja a seguir um exemplo de política do IAM que permite um único fluxo de trabalho com um único grupo de execução.

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/*"
            ]
        }
    ]
}
```

Funções de serviço para AWS HealthOmics

Uma função de serviço é uma função AWS Identity and Access Management (IAM) que concede permissões para que um AWS serviço acesse recursos em sua conta. Você fornece uma função de serviço AWS HealthOmics ao iniciar um trabalho de importação ou iniciar uma execução.

O HealthOmics console pode criar a função necessária para você. Se você usa a HealthOmics API para gerenciar recursos, crie a função de serviço usando o console do IAM. Para obter mais informações, consulte Criar uma função para delegar permissões a um AWS service (Serviço da AWS).

As funções de serviço devem ter a seguinte política de confiança.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "omics.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

A política de confiança permite que o HealthOmics serviço assuma a função.

Tópicos

- Exemplo de políticas de serviço do IAM
- CloudFormation Modelo de exemplo

Exemplo de políticas de serviço do IAM

Nesses exemplos, nomes de recursos e contas IDs são espaços reservados para você substituir por valores reais.

O exemplo a seguir mostra a política de uma função de serviço que você pode usar para iniciar uma execução. A política concede permissões para acessar o local de saída do Amazon S3, o grupo de registros do fluxo de trabalho e o contêiner do Amazon ECR para a execução.

> **Note**
>
> Se você estiver usando o cache de chamadas para a execução, adicione a localização do cache de execução do Amazon S3 como um recurso nas permissões do s3.

Example Política de função de serviço para iniciar uma execução

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/*"
            ]
        }
    ]
}
```

O exemplo a seguir mostra a política de uma função de serviço que você pode usar para um trabalho de importação de loja. A política concede permissões para acessar o local de entrada do Amazon S3.

Example Função de serviço para trabalho na Reference Store

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"
            ]
        }

    ]
}
```

CloudFormation Modelo de exemplo

O CloudFormation modelo de exemplo a seguir cria uma função de serviço que dá HealthOmics permissão para acessar buckets do Amazon S3 que têm nomes prefixados com`omics-`, e para fazer upload de registros de fluxo de trabalho.

Example Permissões de armazenamento de referência, Amazon S3 e 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"
                ]
              }
            }
          ]
        }
```

Permissões do Amazon ECR

Antes que o HealthOmics serviço possa executar um fluxo de trabalho em um contêiner do seu repositório privado do Amazon ECR, você cria uma política de recursos para o repositório. A política concede permissão para o HealthOmics serviço usar o contêiner. Você adiciona essa política de recursos a cada repositório privado referenciado pelo fluxo de trabalho.

> **Note**
>
> O repositório privado e o fluxo de trabalho devem estar na mesma região.

Se AWS contas diferentes forem proprietárias do fluxo de trabalho e do repositório, você precisará configurar as permissões entre contas.

Você não precisa conceder acesso adicional ao repositório para fluxos de trabalho compartilhados. No entanto, você pode criar políticas que permitam ou neguem o acesso de fluxos de trabalho específicos à imagem do contêiner.

Para usar o recurso de cache pull through do Amazon ECR, você precisa criar uma política de permissão de registro.

As seções a seguir descrevem como configurar as permissões de recursos do Amazon ECR para esses cenários. Para obter mais informações sobre permissões no Amazon ECR, consulte Permissões de registro privado no Amazon ECR.

Tópicos

Crie uma política de recursos para o repositório Amazon ECR

Crie uma política de recursos para permitir que o HealthOmics serviço execute um fluxo de trabalho usando um contêiner no repositório. A política concede permissão para que o responsável pelo HealthOmics serviço acesse as ações necessárias do Amazon ECR.

Siga estas etapas para criar a política:

1. Abra a página de repositórios privados no console do Amazon ECR e selecione o repositório ao qual você está concedendo acesso.
2. Na barra lateral de navegação, selecione Permissões.
3. Escolha Editar.
4. Escolha Editar política JSON.
5. Adicione a declaração de política a seguir e selecione Salvar.

   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": "*"
        }
    ]
}
```

Executando fluxos de trabalho com contêineres entre contas

Se AWS contas diferentes forem proprietárias do fluxo de trabalho e do contêiner, você precisará configurar as seguintes permissões entre contas:

1. Atualize a política do Amazon ECR para o repositório para conceder permissão explícita à conta proprietária do fluxo de trabalho.

2. Atualize a função de serviço da conta proprietária do fluxo de trabalho para conceder acesso à imagem do contêiner.

O exemplo a seguir demonstra uma política de recursos do Amazon ECR que concede acesso à conta proprietária do fluxo de trabalho.

Neste exemplo:

- ID da conta de fluxo de trabalho: 111122223333
- ID da conta do repositório do contêiner: 444455556666
- Nome do contêiner: 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 concluir a configuração, adicione a seguinte declaração de política à função de serviço da conta proprietária do fluxo de trabalho. A política concede permissão para que a função de serviço acesse a imagem do contêiner “samtools”. Certifique-se de substituir os números da conta, o nome do contêiner e a região pelos seus próprios valores.

```
{
    "Sid": "CrossAccountEcrRepoPolicy",
    "Effect": "Allow",
    "Action": ["ecr:BatchCheckLayerAvailability", "ecr:BatchGetImage",
 "ecr:GetDownloadUrlForLayer"],
    "Resource": "arn:aws:ecr:us-west-2:444455556666:repository/samtools"
}
```

Políticas do Amazon ECR para fluxos de trabalho compartilhados

> **Note**
>
> HealthOmics permite automaticamente que um fluxo de trabalho compartilhado acesse o repositório Amazon ECR na conta do proprietário do fluxo de trabalho, enquanto o fluxo de trabalho está sendo executado na conta do assinante. Você não precisa conceder acesso adicional ao repositório para fluxos de trabalho compartilhados. Para obter mais informações, consulte Compartilhamento de HealthOmics fluxos de trabalho.

Por padrão, o assinante não tem acesso ao repositório Amazon ECR para usar os contêineres subjacentes. Opcionalmente, você pode personalizar o acesso ao repositório Amazon ECR adicionando chaves de condição à política de recursos do repositório. As seções a seguir fornecem exemplos de políticas.

Restrinja o acesso a fluxos de trabalho específicos

Você pode listar fluxos de trabalho individuais em uma declaração de condição, para que somente esses fluxos de trabalho possam usar contêineres no repositório. A chave de SourceArncondição

especifica o ARN do fluxo de trabalho compartilhado. O exemplo a seguir concede permissão para o fluxo de trabalho especificado usar esse repositório.

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 o acesso a contas específicas

Você pode listar contas de assinantes em uma declaração condicional, para que somente essas contas tenham permissão para usar contêineres no repositório. A chave de SourceAccountcondição especifica a Conta da AWS do assinante. O exemplo a seguir concede permissão para a conta especificada usar esse repositório.

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"
        }
      }
    }
  ]
}
```

Você também pode negar permissões do Amazon ECR para assinantes específicos, conforme mostrado no exemplo de política a seguir.

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"
              }
            }
        }
    ]
}
```

As políticas do Amazon ECR passam pelo cache

Para usar o cache pull through do Amazon ECR, você cria uma política de permissão de registro. Você também cria um modelo de criação de repositório, que define as permissões para os repositórios criados pelo Amazon ECR pull through cache.

As seções a seguir incluem exemplos dessas políticas. Para obter mais informações sobre o cache pull through, consulte Sincronizar um registro upstream com um registro privado do Amazon ECR no Guia do usuário do Amazon Elastic Container Registry.

Política de permissão de registro

Para usar o cache pull through do Amazon ECR, crie uma política de permissão de registro. A política de permissões do registro fornece controle sobre as permissões de replicação e extração de cache.

Para a replicação entre contas, você deve permitir explicitamente Conta da AWS que cada uma delas possa replicar seus repositórios em seu registro.

Por padrão, quando você cria uma regra de cache pull through, qualquer diretor do IAM que tenha permissão para extrair imagens de um registro privado também pode usar a regra de cache pull through. Você pode usar as permissões do registro para ampliar ainda mais essas permissões para repositórios específicos.

Adicione uma política de permissão de registro à conta que possui a imagem do contêiner.

No exemplo a seguir, a política permite que o HealthOmics serviço crie repositórios para cada registro upstream e inicie pull requests upstream dos repositórios criados.

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/*"
            ]
        }
    ]
}
```

Modelo de criação de repositório

Para usar o cache pull through HealthOmics, o repositório Amazon ECR deve ter um modelo de criação de repositório. O modelo define as configurações dos repositórios privados criados para um registro upstream.

Cada modelo contém um prefixo de namespace de repositório, que o Amazon ECR usa para combinar novos repositórios com um modelo específico. Os modelos podem especificar a configuração para todas as configurações do repositório, incluindo políticas de acesso baseadas em recursos, imutabilidade de tags, criptografia e políticas de ciclo de vida. Para obter mais informações, consulte Modelos de criação de repositórios no Guia do usuário do Amazon Elastic Container Registry.

No exemplo a seguir, a política permite que o HealthOmics serviço inicie solicitações pull upstream dos repositórios upstream.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "PTCRepoCreationTemplate",
            "Effect": "Allow",
            "Principal": {
                "Service": "omics.amazonaws.com"
            },
            "Action": [
                "ecr:BatchGetImage",
                "ecr:GetDownloadUrlForLayer"
            ],
            "Resource": "*"
        }
    ]
}
```

Políticas para acesso entre contas ao Amazon ECR

Para acesso entre contas, o proprietário do repositório privado atualiza a política de permissão do registro e o modelo de criação do repositório para permitir o acesso à outra conta e à função de execução dessa conta.

Na política de permissão do registro, adicione uma declaração de política para permitir que a função de execução da outra conta acesse as ações do 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/*"
        }
    ]
}
```

No modelo de criação do repositório, adicione uma declaração de política para permitir que a função de execução da outra conta acesse as novas imagens do contêiner. Opcionalmente, você pode adicionar declarações condicionais para limitar o acesso a fluxos de trabalho 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"
                }
            }
        }
    ]
}
```

Adicione permissões para duas ações adicionais (CreateRepository e BatchImportUpstreamImage) na função de execução e especifique o recurso que a função de execução pode acessar.

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 Permissões de recursos

AWS HealthOmics cria e acessa recursos em outros serviços em seu nome quando você executa um trabalho ou cria uma loja. Em alguns casos, você precisa configurar permissões em outros serviços para acessar recursos ou permitir o acesso HealthOmics a eles.

Para obter permissões de recursos relacionadas ao Amazon ECR, consultePermissões do Amazon ECR.

Permissões do Lake Formation

Antes de usar os recursos de análise HealthOmics, defina as configurações padrão do banco de dados no Lake Formation.

Para configurar as permissões de recursos no Lake Formation

1. Abra a página de configurações do catálogo de dados no console do Lake Formation.
2. Desmarque os requisitos de controle de acesso do IAM para bancos de dados e tabelas em Permissões padrão para bancos de dados e tabelas recém-criados.
3. Escolha Salvar.

HealthOmics O Analytics aceita dados automaticamente se sua política de serviço tiver as permissões corretas de RAM, como no exemplo a seguir.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "omics:*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ram:AcceptResourceShareInvitation",
        "ram:GetResourceShareInvitations"
      ],
      "Resource": "*"
    }
  ]
}
```

Permissões para acesso a dados usando o Amazon S3 URIs

Você pode acessar dados do armazenamento de sequências usando operações de HealthOmics API ou operações de API do Amazon S3.

Para acesso à HealthOmics API, HealthOmics as permissões são gerenciadas por meio de uma política do IAM. No entanto, o S3 Access requer dois níveis de configuração: permissão explícita na política de acesso ao S3 da loja e uma política do IAM. Para saber mais sobre como usar políticas do IAM com HealthOmics, consulte Funções de serviço para HealthOmics.

Há três maneiras de compartilhar a capacidade de ler objetos usando o Amazon APIs S3:

1. Compartilhamento baseado em políticas — Esse compartilhamento requer habilitar o diretor do IAM na política de acesso do S3 e escrever uma política do IAM e anexá-la ao diretor do IAM. Consulte o próximo tópico para obter mais detalhes.
2. Pré-assinado URLs — Você também pode gerar um URL pré-assinado compartilhável para um arquivo no armazenamento de sequências. Para saber mais sobre a criação de pré-assinados URLs usando o Amazon S3, consulte Usando URLs pré-assinados na documentação do Amazon S3. A política de acesso do S3 do armazenamento de sequências oferece suporte a declarações para limitar os recursos de URL pré-assinados.
3. Funções assumidas — Crie uma função na conta do proprietário dos dados que tenha uma política de acesso que permita que os usuários assumam essa função.

Tópicos

Compartilhamento baseado em políticas

Se você acessar dados do armazenamento de sequência usando um URI direto do S3, HealthOmics fornece medidas de segurança aprimoradas para a política de acesso ao bucket do S3 associada.

As regras a seguir se aplicam às novas políticas de acesso do S3. Para as políticas existentes, as regras se aplicam na próxima atualização da política:

- As políticas de acesso do S3 oferecem suporte aos seguintes elementos de política
  - Versão, ID, Declaração, Sid, Efeito, Principal, Ação, Recurso, Condição
- As políticas de acesso do S3 oferecem suporte às seguintes chaves de condição:
  - s3:ExistingObjectTag/<key>, s3: prefixo, s3: versão da assinatura, s3: TlsVersion
  - As políticas também oferecem suporte ao aws: PrincipalArn com os seguintes operadores de condição: ArnEquals e ArnLike

Se você tentar adicionar ou atualizar uma política para incluir um elemento ou condição sem suporte, o sistema rejeitará a solicitação.

Tópicos

Política de acesso padrão do S3

Quando você cria um armazenamento de sequência, HealthOmics cria uma política de acesso padrão do S3 concedendo à conta raiz do proprietário do armazenamento de dados as seguintes permissões para todos os objetos acessíveis no armazenamento de sequência: S3:GetObject, S3 e S3GetObjectTagging:. ListBucket A política padrão criada é:

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/*"
        }
    ]
}
```

Personalizando a política de acesso

Se a política de acesso do S3 estiver em branco, nenhum acesso ao S3 será permitido. Se houver uma política existente e você precisar remover o acesso s3, use `deleteS3AccessPolicy` para remover todo o acesso.

Para adicionar restrições ao compartilhamento ou conceder acesso a outras contas, você pode atualizar a política usando a `PutS3AccessPolicy` API. As atualizações da política não podem ir além do prefixo do armazenamento de sequências ou das ações especificadas.

Política do IAM

Para permitir que um usuário ou principal do IAM acesse usando o Amazon S3 APIs, além da permissão na política de acesso do S3, uma política do IAM precisa ser criada e anexada ao principal para conceder acesso. Uma política que permite o acesso à API do Amazon S3 pode ser aplicada no nível do armazenamento de sequências ou no nível do conjunto de leitura. No nível do conjunto de leitura, a permissão pode ser restringida por meio do prefixo ou usando filtros de tag de recurso para padrões de ID de amostra ou assunto.

Se o armazenamento de sequências usar uma chave gerenciada pelo cliente (CMK), o principal também deverá ter direitos de usar a chave KMS para descriptografia. Para obter mais informações, consulte Acesso ao KMS entre contas no Guia do AWS Key Management Service desenvolvedor.

O exemplo a seguir dá ao usuário acesso a um armazenamento de sequências. Você pode ajustar o acesso com condições adicionais ou filtros baseados em 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/*"
        }
      }
    }
  ]
}
```

Controle de acesso com base em tags

Para usar o controle de acesso baseado em tags, o armazenamento de sequências deve primeiro ser atualizado para propagar as chaves de tag que serão usadas. Essa configuração é definida durante a criação ou atualização do armazenamento de sequências. Depois que as tags são propagadas, as condições da tag podem ser usadas para adicionar mais restrições. As restrições podem ser colocadas na política de acesso do S3 ou na política do IAM. A seguir está um exemplo de uma política de acesso do S3 baseada em guias que seria definida:

```
{
    "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"
        }
    }
}
```

Exemplo de restrição

Cenário: criar um compartilhamento em que o proprietário dos dados possa restringir a capacidade do usuário de baixar dados “retirados”.

Nesse cenário, um proprietário de dados (conta #111111111111) gerenciava um armazenamento de dados. Esse proprietário de dados compartilha os dados com uma ampla variedade de usuários terceirizados, incluindo um pesquisador (conta #999999999999). Como parte do gerenciamento

dos dados, o proprietário dos dados recebe periodicamente solicitações para retirar os dados dos participantes. Para gerenciar essa retirada, o proprietário dos dados primeiro restringe o acesso direto ao download ao receber a solicitação e, eventualmente, exclui os dados de acordo com seus requisitos.

Para atender a essa necessidade, o proprietário dos dados configura um armazenamento de sequência e cada conjunto de leitura recebe uma tag de “status” que será definida como “retirado” se a solicitação de retirada for recebida. Para dados com a tag definida com esse valor, eles querem garantir que nenhum usuário possa executar “getObject” nesse arquivo. Para fazer essa configuração, o proprietário dos dados precisará garantir que duas etapas sejam executadas.

Etapa 1. Para o armazenamento de sequências, certifique-se de que a tag de status esteja atualizada para ser propagada. Isso é feito adicionando a chave “status” `propogatedSetLevelTags` ao ligar `createSequenceStore` ou `updateSequenceStore.`

Etapa 2. Atualize a política de acesso s3 da loja para restringir getObject em objetos com a tag de status definida como retirada. Isso é feito atualizando a política de acesso às lojas usando a `PutS3AccesPolicy` API. A política a seguir permitiria que os clientes ainda vissem os arquivos retirados ao listar objetos, mas impediria que eles os acessassem:

- Declaração 1 (restrictedGetWithdrawal): A conta 999999999999 não pode recuperar objetos que foram retirados.
- Declaração 2 (ownerGetAll): A conta 111111111111, proprietária dos dados, pode recuperar todos os objetos, incluindo objetos que foram retirados.
- Declaração 3 (everyoneListAll): Todas as contas compartilhadas, 111111111111 e 9999999999, podem executar a operação em todo o prefixo. ListBucket

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/*"
                }
            }
        }
    ]
}
```

Segurança na AWS HealthOmics

A segurança na nuvem AWS é a maior prioridade. Como AWS cliente, você se beneficia de data centers e arquiteturas de rede criados para atender aos requisitos das organizações mais sensíveis à segurança.

A segurança é uma responsabilidade compartilhada entre você AWS e você. O Modelo de Responsabilidade Compartilhada descreve isso como segurança da nuvem e segurança na nuvem:

- Segurança da nuvem — AWS é responsável por proteger a infraestrutura que executa AWS os serviços no Nuvem AWS. AWS também fornece serviços que você pode usar com segurança. Auditores terceirizados testam e verificam regularmente a eficácia de nossa segurança como parte dos Programas de Conformidade Programas de AWS de . Para saber mais sobre os programas de conformidade que se aplicam à AWS HealthOmics, consulte AWS Serviços no escopo por programa de conformidade AWS .
- Segurança na nuvem — Sua responsabilidade é determinada pelo AWS serviço que você usa. Você também é responsável por outros fatores, incluindo a confidencialidade de seus dados, os requisitos da empresa e as leis e regulamentos aplicáveis.

Essa documentação ajuda você a entender como aplicar o modelo de responsabilidade compartilhada ao usar a AWS HealthOmics. Os tópicos a seguir mostram como configurar a AWS HealthOmics para atender aos seus objetivos de segurança e conformidade. Você também aprende a usar outros AWS serviços que ajudam você a monitorar e proteger seus HealthOmics recursos da AWS.

Tópicos

Proteção de dados em AWS HealthOmics

O modelo de responsabilidade AWS compartilhada de se aplica à proteção de dados na AWS HealthOmics. Conforme descrito neste modelo, AWS é responsável por proteger a infraestrutura global que executa todos os Nuvem AWS. Você é responsável por manter o controle sobre o conteúdo hospedado nessa infraestrutura. Você também é responsável pelas tarefas de configuração e gerenciamento de segurança dos Serviços da AWS que usa. Para obter mais informações sobre a privacidade de dados, consulte as Data Privacy FAQ. Para obter mais informações sobre a proteção de dados na Europa, consulte a postagem do blog AWS Shared Responsibility Model and RGPD no Blog de segurança da AWS .

Para fins de proteção de dados, recomendamos que você proteja Conta da AWS as credenciais e configure usuários individuais com AWS IAM Identity Center ou AWS Identity and Access Management (IAM). Dessa maneira, cada usuário receberá apenas as permissões necessárias para cumprir suas obrigações de trabalho. Recomendamos também que você proteja seus dados das seguintes formas:

- Use uma autenticação multifator (MFA) com cada conta.
- Use SSL/TLS para se comunicar com AWS os recursos. Exigimos TLS 1.2 e recomendamos TLS 1.3.
- Configure a API e o registro de atividades do usuário com AWS CloudTrail. Para obter informações sobre o uso de CloudTrail trilhas para capturar AWS atividades, consulte Como trabalhar com CloudTrail trilhas no Guia AWS CloudTrail do usuário.
- Use soluções de AWS criptografia, juntamente com todos os controles de segurança padrão Serviços da AWS.
- Use serviços gerenciados de segurança avançada, como o Amazon Macie, que ajuda a localizar e proteger dados sensíveis armazenados no Amazon S3.
- Se você precisar de módulos criptográficos validados pelo FIPS 140-3 ao acessar AWS por meio de uma interface de linha de comando ou de uma API, use um endpoint FIPS. Para obter mais informações sobre os endpoints FIPS disponíveis, consulte Federal Information Processing Standard (FIPS) 140-3.

É altamente recomendável que nunca sejam colocadas informações confidenciais ou sensíveis, como endereços de e-mail de clientes, em tags ou campos de formato livre, como um campo Nome. Isso inclui quando você trabalha com a AWS HealthOmics ou outra Serviços da AWS pessoa usando o console AWS CLI, a API ou AWS SDKs. Quaisquer dados inseridos em tags ou em campos

de texto de formato livre usados para nomes podem ser usados para logs de faturamento ou de diagnóstico. Se você fornecer um URL para um servidor externo, é fortemente recomendável que não sejam incluídas informações de credenciais no URL para validar a solicitação nesse servidor.

Criptografia em repouso

Tópicos

Para proteger dados confidenciais de clientes em repouso, AWS HealthOmics fornece criptografia por padrão usando uma chave do AWS Key Management Service (AWS KMS) de propriedade do serviço. As chaves gerenciadas pelo cliente também são suportadas. Para saber mais sobre a chave gerenciada pelo cliente, consulte Amazon Key Management Service.

Todos os armazenamentos de HealthOmics dados (armazenamento e análise) oferecem suporte ao uso de chaves gerenciadas pelo cliente. A configuração de criptografia não pode ser alterada após a criação de um armazenamento de dados. Se um armazenamento de dados estiver usando um Chave pertencente à AWS, ele será indicado como AWS_OWNED_KMS_KEY e você não verá a chave específica usada para criptografia em repouso.

Para HealthOmics fluxos de trabalho, as chaves gerenciadas pelo cliente não são suportadas pelo sistema de arquivos temporário; no entanto, todos os dados em repouso são criptografados automaticamente usando o algoritmo de criptografia de criptografia de bloco XTS-AES-256 para criptografar o sistema de arquivos. O usuário e a função do IAM usados para iniciar a execução de um fluxo de trabalho também devem ter acesso às AWS KMS chaves usadas para os buckets de entrada e saída do fluxo de trabalho. Os fluxos de trabalho não usam concessões e a AWS KMS criptografia é limitada aos buckets de entrada e saída do Amazon S3. A função do IAM usada para o fluxo de trabalho também APIs deve ter acesso às AWS KMS chaves usadas, bem como aos buckets de entrada e saída do Amazon S3. Você pode usar as funções e permissões do IAM para controlar o acesso ou AWS KMS as políticas. Para saber mais, consulte Autenticação e controle de acesso para AWS KMS.

Quando você usa AWS Lake Formation com o HealthOmics Analytics, todas as permissões de descriptografia associadas ao Lake Formation também são concedidas aos buckets de entrada e saída do Amazon S3. Mais informações sobre como AWS Lake Formation gerenciar permissões podem ser encontradas na AWS Lake Formation documentação.

HealthOmics O Analytics concede ao Lake Formation kms: Decrypt permissões para ler os dados criptografados em um bucket do Amazon S3. Contanto que você tenha permissões para consultar os dados por meio do Lake Formation, você poderá ler os dados criptografados. O acesso aos dados é controlado por meio do controle de acesso aos dados no Lake Formation, não por meio de uma política de chaves do KMS. Para saber mais, consulte as solicitações de serviço AWS integradas da AWS na documentação do Lake Formation.

Chaves pertencentes à AWS

Por padrão, HealthOmics usa Chaves pertencentes à AWS para criptografar automaticamente dados em repouso, pois esses dados podem conter informações confidenciais, como informações de identificação pessoal (PII) ou Informações de saúde protegidas (PHI). Chaves pertencentes à AWS não estão armazenados em sua conta. Elas fazem parte de uma coleção de chaves do KMS que a AWS possui e gerencia para uso em várias contas da AWS.

Os serviços da AWS podem ser usados Chaves pertencentes à AWS para proteger seus dados. Você não pode visualizar, gerenciar Chaves pertencentes à AWS, acessar ou auditar seu uso. No entanto, você não precisa fazer nenhum trabalho nem alterar nenhum programa para proteger as chaves que criptografam seus dados.

Não é cobrada uma taxa mensal ou uma taxa de uso pelo uso Chaves pertencentes à AWS, e elas não contam nas cotas do AWS KMS da sua conta. Para obter mais informações, consulte Chaves gerenciadas pela AWS.

Chaves gerenciadas pelo cliente

HealthOmics suporta o uso de chaves simétricas gerenciadas pelo cliente que você cria, possui e gerencia para adicionar uma segunda camada de criptografia sobre a criptografia existente de propriedade da AWS. Como você tem controle total dessa camada de criptografia, você pode realizar tarefas como:

- Estabelecer e manter políticas de chaves, políticas do IAM e concessões
- Rotacionar os materiais de chave de criptografia
- Habilitar e desabilitar políticas de chaves

- Adicionar etiquetas
- Criar réplicas de chaves
- Chaves de agendamento para exclusão

Você também pode usar CloudTrail para rastrear as solicitações HealthOmics enviadas AWS KMS em seu nome. Aplicam-se cobranças adicionais do AWS KMS . Para obter mais informações, consulte chaves gerenciadas pelo cliente.

Criar uma chave gerenciada pelo cliente

Você pode criar uma chave simétrica gerenciada pelo cliente usando o AWS Management Console ou o. AWS KMS APIs

Siga as etapas para criar chaves simétricas gerenciadas pelo cliente no Guia do desenvolvedor do AWS Key Management Service.

As políticas de chaves controlam o acesso à chave gerenciada pelo cliente. Cada chave gerenciada pelo cliente deve ter exatamente uma política de chave, que contém declarações que determinam quem pode usar a chave e como pode usá-la. Ao criar uma chave gerenciada pelo cliente, você pode especificar uma política de chaves. Para obter mais informações, consulte Gerenciamento do acesso às chaves gerenciadas pelo cliente no Guia do desenvolvedor do AWS Key Management Service.

Para usar uma chave gerenciada pelo cliente com seus recursos do HealthOmics Analytics, o responsável pela chamada exige kms: CreateGrant operations na política de chaves. Isso permite que o sistema use um token FAS para criar uma concessão para uma chave gerenciada pelo cliente que controla o acesso a uma chave KMS especificada. Essa chave dá ao usuário acesso às operações kms:grant necessárias. HealthOmics Consulte Uso de subsídios para obter mais informações.

Para HealthOmics análises, as seguintes operações de API devem ser permitidas para o responsável pela chamada:

- kms: CreateGrant adiciona concessões a uma chave gerenciada pelo cliente específica, o que permite o acesso às operações de concessão no HealthOmics Analytics.
- kms: DescribeKey fornece os detalhes da chave gerenciada pelo cliente necessários para validar a chave. Isso é necessário para todas as operações.

- kms: GenerateDataKey fornece acesso para criptografar recursos em repouso para todas as operações de gravação. Além disso, essa ação fornece detalhes da chave gerenciada pelo cliente que o serviço pode usar para validar se o chamador tem acesso para usar a chave.
- O KMS:Decrypt fornece acesso às operações de leitura ou pesquisa de recursos criptografados.

Para usar uma chave gerenciada pelo cliente com seus recursos HealthOmics de armazenamento, o responsável pelo HealthOmics serviço e o responsável pela chamada devem ser permitidos na política de chaves. Isso permite que o serviço valide se o chamador tem acesso à chave e usa o responsável pelo serviço para executar o gerenciamento da loja usando a chave gerenciada pelo cliente. Para HealthOmics armazenamento, a política de chaves do responsável pelo serviço deve permitir as seguintes operações de API:

- kms: DescribeKey fornece os detalhes da chave gerenciada pelo cliente necessários para validar a chave. Isso é necessário para todas as operações.
- kms: GenerateDataKey fornece acesso para criptografar recursos em repouso para todas as operações de gravação. Além disso, essa ação fornece detalhes da chave gerenciada pelo cliente que o serviço pode usar para validar se o chamador tem acesso para usar a chave.
- O KMS:Decrypt fornece acesso às operações de leitura ou pesquisa de recursos criptografados.

O exemplo a seguir mostra uma declaração de política que permite que um responsável pelo serviço crie e interaja com uma HealthOmics sequência ou um repositório de referência criptografado usando a chave gerenciada pelo cliente:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
{
    "Effect": "Allow",
    "Principal": {
        "Service": "omics.amazonaws.com"
    },
    "Action": [
        "kms:Decrypt",
        "kms:DescribeKey",
        "kms:Encrypt",
```

```
            "kms:GenerateDataKey*"
        ],
        "Resource": "*"
            }
        ]
}
```

O exemplo a seguir mostra uma política que cria permissões para um armazenamento de dados descriptografar dados de um bucket do 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"
                  ]
              }
            }
          }
        }
    ]
}
```

Permissões do IAM necessárias para usar uma chave gerenciada pelo cliente

Ao criar um recurso, como um armazenamento de dados com AWS KMS criptografia usando uma chave gerenciada pelo cliente, há permissões necessárias tanto para a política de chaves quanto para a política do IAM para o usuário ou a função do IAM.

Você pode usar a chave de ViaService condição kms: para limitar o uso da chave KMS somente às solicitações originadas de. HealthOmics

Para obter mais informações sobre políticas de chaves, consulte Habilitar políticas do IAM no Guia do desenvolvedor do AWS Key Management Service.

Tópicos

- Permissões da API Analytics
- Permissões da API de armazenamento
- Como HealthOmics usa subsídios no AWS KMS
- Monitorando suas chaves de criptografia para AWS HealthOmics

Permissões da API Analytics

Para HealthOmics análises, o usuário ou a função do IAM que cria as lojas deve ter as permissões kms:CreateGrant, kms:GenerateDataKey, kms: Decrypt e, além das kms: DescribeKey permissões necessárias. HealthOmics

Permissões da API de armazenamento

Para HealthOmics armazenamento APIs, o usuário ou a função do IAM que chama as seguintes operações de API exige as permissões listadas:

CreateReferenceStore, CreateSequenceStore

Para criar uma loja, o chamador do IAM deve ter `kms:DescribeKey` as permissões mais as HealthOmics permissões necessárias. O principal do HealthOmics serviço liga `kms:GenerateDataKeyWithoutPlaintext` para realizar verificações de validação de acesso para carregamento e acesso aos dados.

StartReadSetImportJob, StartReferenceImportJob

Para iniciar trabalhos de importação de dados, o chamador do IAM deve ter `kms:Decrypt` `kms:GenerateDataKey` permissões para a chave KMS na loja para a importação e `kms:Decrypt` permissões no bucket do Amazon S3 contendo os objetos a serem importados. Além disso, a função passada para a chamada deve ter `kms:Decrypt` permissões no bucket do Amazon S3 contendo os objetos a serem importados. O chamador do IAM também deve ter permissões para passar a função para o trabalho.

CreateMultipartReadSetUpload, UploadReadSetPart, CompleteMultipartReadSetUpload

Para concluir um upload de várias partes, o chamador do IAM deve ter `kms:Decrypt` e `kms:GenerateDataKey` criar, carregar e concluir o upload de várias partes.

StartReadSetExportJob

Para iniciar um trabalho de exportação de dados, o chamador do IAM deve ter `kms:Decrypt` permissão para que a chave KMS na loja exporte de `kms:GenerateDataKey` e `kms:Decrypt` permissões no bucket do Amazon S3 que recebe os objetos. Além disso, a função passada para a chamada deve ter `kms:Decrypt` permissões no bucket do Amazon S3 que recebe os objetos. O chamador do IAM também deve ter permissões para passar a função para o trabalho.

StartReadsetActivationJob

Para iniciar um trabalho de ativação do conjunto de leitura, o chamador do IAM deve ter `kms:GenerateDataKey` permissões `kms:Decrypt` e permissões para os objetos.

GetReference, GetReadSet

Para ler objetos da loja, o chamador do IAM precisa ter `kms:Decrypt` permissão para os objetos.

Conjunto de leitura S3 GetObject

Para ler objetos da loja usando a `GetObject` API do Amazon S3, o chamador do IAM deve ter `kms:Decrypt` permissão para os objetos. Defina essa permissão tanto para a chave gerenciada pelo cliente quanto para Chave pertencente à AWS as configurações.

Como HealthOmics usa subsídios no AWS KMS

HealthOmics O Analytics exige uma concessão para usar sua chave KMS gerenciada pelo cliente. Os subsídios não são necessários nem usados para HealthOmics fluxos de trabalho. HealthOmics O armazenamento usa a chave gerenciada pelo cliente diretamente do responsável pelo serviço, portanto, não use uma concessão. Quando você cria uma loja de análise criptografada com uma chave gerenciada pelo cliente, a HealthOmics análise cria uma concessão em seu nome enviando uma CreateGrantsolicitação para o AWS KMS. Os subsídios no AWS KMS são usados para dar HealthOmics acesso a uma chave do KMS em uma conta de cliente.

Não é recomendável revogar ou retirar os subsídios que o HealthOmics Analytics cria em seu nome. Se você revogar ou retirar a concessão que dá HealthOmics permissão para usar as chaves do AWS KMS em sua conta HealthOmics , não poderá acessar esses dados, criptografar novos recursos enviados para o armazenamento de dados ou descriptografá-los quando forem retirados.

Quando você revoga ou retira um subsídio HealthOmics, a alteração ocorre imediatamente. Para revogar os direitos de acesso, recomendamos que você exclua o armazenamento de dados em vez de revogar a concessão. Quando você exclui o armazenamento de dados, HealthOmics retira as concessões em seu nome.

Monitorando suas chaves de criptografia para AWS HealthOmics

Você pode usar CloudTrail para rastrear as solicitações AWS HealthOmics enviadas AWS KMS em seu nome ao usar uma chave gerenciada pelo cliente. As entradas de registro no CloudTrail registro mostram HealthOmics .amazonaws.com no campo UserAgent para distinguir claramente as solicitações feitas por. HealthOmics

Os exemplos a seguir são CloudTrail eventos para CreateGrant, GenerateDataKey, Decrypt e DescribeKey para monitorar AWS KMS operações chamadas por HealthOmics para acessar dados criptografados pela chave gerenciada pelo cliente.

O seguinte também mostra como usar CreateGrant para permitir que a HealthOmics análise acesse uma chave KMS fornecida pelo cliente, permitindo usar essa chave KMS HealthOmics para criptografar todos os dados do cliente em repouso.

Você não precisa criar seus próprios subsídios. HealthOmics cria uma concessão em seu nome enviando uma CreateGrant solicitação para o AWS KMS. As concessões AWS KMS são usadas para dar HealthOmics acesso a uma AWS KMS chave na conta de um 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"
}
```

O exemplo a seguir mostra como usar GenerateDataKey para garantir que o usuário tenha as permissões necessárias para criptografar dados antes de armazená-los.

```
{
    "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"
}
```


Saiba mais

Os recursos a seguir fornecem mais informações sobre criptografia de dados em repouso.

Para obter mais informações sobre os conceitos básicos do AWS Key Management Service, consulte a AWS KMS documentação.

Para obter mais informações sobre as melhores práticas de segurança na AWS KMS documentação.

Criptografia em trânsito

AWS HealthOmics usa o TLS 1.2+ para criptografar dados em trânsito pelos endpoints públicos e por meio de serviços de back-end.

Gerenciamento de identidade e acesso em HealthOmics

AWS Identity and Access Management (IAM) é uma ferramenta AWS service (Serviço da AWS) que ajuda o administrador a controlar com segurança o acesso aos AWS recursos. Os administradores do IAM controlam quem pode ser autenticado (conectado) e autorizado (tem permissões) para usar os recursos da AWS HealthOmics . O IAM é um AWS service (Serviço da AWS) que você pode usar sem custo adicional.

Tópicos

Público

A forma como você usa AWS Identity and Access Management (IAM) difere com base na sua função:

- Usuário do serviço: solicite permissões ao administrador caso você não consiga acessar os recursos (consulte Solução de problemas AWS HealthOmics de identidade e acesso).
- Administrador do serviço: determine o acesso do usuário e envie solicitações de permissão (consulte Como AWS HealthOmics funciona com o IAM).
- Administrador do IAM: escreva políticas para gerenciar o acesso (consulte Exemplos de políticas baseadas em identidade para AWS HealthOmics).

Autenticação com identidades

A autenticação é a forma como você faz login AWS usando suas credenciais de identidade. Você deve estar autenticado como usuário do IAM ou assumindo uma função do IAM. Usuário raiz da conta da AWS

Você pode fazer login como uma identidade federada usando credenciais de uma fonte de identidade como AWS IAM Identity Center (IAM Identity Center), autenticação de login único ou credenciais. Google/Facebook Para obter mais informações sobre como fazer login, consulte Como fazer login na Conta da AWS no Guia do usuário do Início de Sessão da AWS .

Para acesso programático, AWS fornece um SDK e uma CLI para assinar solicitações criptograficamente. Para obter mais informações, consulte AWS Signature Version 4 para solicitações de API no Guia do usuário do IAM.

Conta da AWS usuário root

Ao criar um Conta da AWS, você começa com uma identidade de login chamada usuário Conta da AWS raiz que tem acesso completo a todos Serviços da AWS os recursos. É altamente recomendável não usar o usuário-raiz em tarefas diárias. Para tarefas que exijam credenciais do usuário-raiz, consulte Tarefas que exijam credenciais do usuário-raiz no Guia do usuário do IAM.

Identidade federada

Como prática recomendada, exija que os usuários humanos usem a federação com um provedor de identidade para acessar Serviços da AWS usando credenciais temporárias.

Uma identidade federada é um usuário do seu diretório corporativo, provedor de identidade da web ou Directory Service que acessa Serviços da AWS usando credenciais de uma fonte de identidade. As identidades federadas assumem funções que oferecem credenciais temporárias.

Para gerenciamento de acesso centralizado, é recomendável usar o AWS IAM Identity Center. Para obter mais informações, consulte O que é o IAM Identity Center? no Guia do usuário do AWS IAM Identity Center .

Usuários e grupos do IAM

Usuário do IAM é uma identidade com permissões específicas para uma única pessoa ou aplicação. É recomendável usar credenciais temporárias, em vez de usuários do IAM com credenciais de longo prazo. Para obter mais informações, consulte Exigir que usuários humanos usem a federação com

um provedor de identidade para acessar AWS usando credenciais temporárias no Guia do usuário do IAM.

Um grupo do IAM especifica um conjunto de usuários do IAM e facilita o gerenciamento de permissões para grandes conjuntos de usuários. Para obter mais informações, consulte Casos de uso para usuários do IAM no Guia do usuário do IAM.

Perfis do IAM

Um perfil do IAM é uma identidade com permissões específicas que oferece credenciais temporárias. Você pode assumir uma função mudando de um usuário para uma função do IAM (console) ou chamando uma operação de AWS API AWS CLI ou. Para obter mais informações, consulte Métodos para assumir um perfil no Manual do usuário do IAM.

As funções do IAM são úteis para acesso de usuários federados, permissões temporárias de usuários do IAM, acesso entre contas, acesso entre serviços e aplicativos executados na Amazon. EC2 Consulte mais informações em Acesso a recursos entre contas no IAM no Guia do usuário do IAM.

Gerenciar o acesso usando políticas

Você controla o acesso AWS criando políticas e anexando-as a AWS identidades ou recursos. Uma política define permissões quando associada a uma identidade ou recurso. AWS avalia essas políticas quando um diretor faz uma solicitação. A maioria das políticas é armazenada AWS como documentos JSON. Para obter mais informações sobre documentos de política JSON, consulte Visão geral das políticas de JSON no Guia do usuário do IAM.

Usando políticas, os administradores especificam quem tem acesso ao quê definindo qual entidade principal pode realizar ações em quais recursos e sob quais condições.

Por padrão, usuários e perfis não têm permissões. Um administrador do IAM cria políticas do IAM e as adiciona a funções, que os usuários podem acabar assumindo. As políticas do IAM definem permissões, independentemente do método usado para realizar a operação.

Políticas baseadas em identidade

Políticas baseadas em identidade são documentos de política de permissões JSON anexados a uma identidade (usuário, grupo ou perfil). Essas políticas controlam quais ações as identidades podem realizar, em quais recursos e sob quais condições. Para saber como criar uma política baseada em

identidade, consulte Definir permissões personalizadas do IAM com as políticas gerenciadas pelo cliente no Guia do Usuário do IAM.

As políticas baseadas em identidade podem ser políticas em linha (incorporadas diretamente a uma única identidade) ou políticas gerenciadas (políticas independentes anexadas a várias identidades). Para saber como escolher entre uma política gerenciada e políticas em linha, consulte Escolher entre políticas gerenciadas e políticas em linha no Guia do usuário do IAM.

Políticas baseadas em recursos

Políticas baseadas em recursos são documentos de políticas JSON que você anexa a um recurso. Entre os exemplos estão políticas de confiança de perfil do IAM e políticas de bucket do Amazon S3. Em serviços compatíveis com políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. É necessário especificar uma entidade principal em uma política baseada em recursos.

Políticas baseadas em recursos são políticas em linha localizadas nesse serviço. Você não pode usar políticas AWS gerenciadas do IAM em uma política baseada em recursos.

Outros tipos de política

AWS oferece suporte a tipos de políticas adicionais que podem definir o máximo de permissões concedidas por tipos de políticas mais comuns:

- Limites de permissões: definem o número máximo de permissões que uma política baseada em identidade pode conceder a uma entidade do IAM. Para obter mais informações, consulte Limites de permissões para entidades do IAM no Guia do usuário do IAM.
- Políticas de controle de serviço (SCPs) — Especifique as permissões máximas para uma organização ou unidade organizacional em AWS Organizations. Para obter mais informações, consulte Políticas de controle de serviço no Guia do usuário do AWS Organizations .
- Políticas de controle de recursos (RCPs) — Defina o máximo de permissões disponíveis para recursos em suas contas. Para obter mais informações, consulte Políticas de controle de recursos (RCPs) no Guia AWS Organizations do usuário.
- Políticas de sessão: políticas avançadas passadas como um parâmetro durante a criação de uma sessão temporária para uma função ou um usuário federado. Para obter mais informações, consulte Políticas de sessão no Guia do usuário do IAM.

Vários tipos de política

Quando vários tipos de política são aplicáveis a uma solicitação, é mais complicado compreender as permissões resultantes. Para saber como AWS determinar se uma solicitação deve ser permitida quando vários tipos de políticas estão envolvidos, consulte Lógica de avaliação de políticas no Guia do usuário do IAM.

Como AWS HealthOmics funciona com o IAM

Antes de usar o IAM para gerenciar o acesso à AWS HealthOmics, saiba quais recursos do IAM estão disponíveis para uso com a AWS HealthOmics.

Recursos do IAM que você pode usar com AWS HealthOmics

Recurso do IAM	HealthOmics apoio
Políticas baseadas em identidade	Sim
Políticas baseadas em recurso	Não
Ações de políticas	Sim
Recursos de políticas	Sim
Chaves de condição de políticas	Não
ACLs	Não
ABAC (tags nas políticas)	Sim
Credenciais temporárias	Sim
Permissões de entidade principal	Sim
Perfis de serviço	Sim
Perfis vinculados a serviço	Não

Para ter uma visão de alto nível de como HealthOmics e outros AWS serviços funcionam com a maioria dos recursos do IAM, consulte AWS os serviços que funcionam com o IAM no Guia do usuário do IAM.

Prevenção do problema "confused deputy" entre serviços

O problema "confused deputy" é um problema de segurança em que uma entidade que não tem permissão para executar uma ação pode coagir uma entidade mais privilegiada a executar a ação. Em AWS, a falsificação de identidade entre serviços pode resultar no problema confuso do deputado. A personificação entre serviços pode ocorrer quando um serviço (o serviço de chamada) chama outro serviço (o serviço chamado). O serviço de chamada pode ser manipulado de modo a usar suas permissões para atuar nos recursos de outro cliente de uma forma na qual ele não deveria ter permissão para acessar. Para evitar isso, a AWS fornece ferramentas que ajudam você a proteger seus dados para todos os serviços com entidades principais de serviço que receberam acesso aos recursos em sua conta.

Recomendamos usar as chaves de contexto de condição `aws:SourceAccount`global `aws:SourceArn`e as chaves de contexto nas políticas de recursos para limitar as permissões que a AWS HealthOmics concede a outro serviço ao recurso.

Para evitar o problema confuso de delegado nas funções assumidas por HealthOmics, defina o valor de `aws:SourceArn` to `arn:aws:omics:`*`region`*`:`*`accountNumber`*`:*` na política de confiança da função. O caractere curinga (*) aplica a condição a todos os HealthOmics recursos.

A política de relacionamento de confiança a seguir concede HealthOmics acesso aos seus recursos e usa as `aws:SourceArn` chaves de contexto de condição `aws:SourceAccount` global para evitar o confuso problema adjunto. Use essa política ao criar uma função 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 baseadas em identidade para HealthOmics

Compatível com políticas baseadas em identidade: sim

As políticas baseadas em identidade são documentos de políticas de permissões JSON que podem ser anexados a uma identidade, como usuário do IAM, grupo de usuários ou perfil. Essas políticas controlam quais ações os usuários e perfis podem realizar, em quais recursos e em que condições. Para saber como criar uma política baseada em identidade, consulte Definir permissões personalizadas do IAM com as políticas gerenciadas pelo cliente no Guia do Usuário do IAM.

Com as políticas baseadas em identidade do IAM, é possível especificar ações e recursos permitidos ou negados, assim como as condições sob as quais as ações são permitidas ou negadas. Para saber mais sobre todos os elementos que podem ser usados em uma política JSON, consulte Referência de elemento de política JSON do IAM no Guia do usuário do IAM.

Exemplos de políticas baseadas em identidade para HealthOmics

Para ver exemplos de políticas HealthOmics baseadas em identidade da AWS, consulte. Exemplos de políticas baseadas em identidade para AWS HealthOmics

Políticas baseadas em recursos dentro HealthOmics

Compatibilidade com políticas baseadas em recursos: não

Políticas baseadas em recursos são documentos de políticas JSON que você anexa a um recurso. São exemplos de políticas baseadas em recursos as políticas de confiança de perfil do IAM e as políticas de bucket do Amazon S3. Em serviços compatíveis com políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. Para o atributo ao qual a política está anexada, a política define quais ações uma entidade principal especificado pode executar nesse atributo e em que condições. É necessário especificar uma entidade principal em uma política baseada em recursos. Os diretores podem incluir contas, usuários, funções, usuários federados ou. Serviços da AWS

Para permitir o acesso entre contas, é possível especificar uma conta inteira ou as entidades do IAM em outra conta como a entidade principal em uma política baseada em recursos. Consulte mais informações em Acesso a recursos entre contas no IAM no Guia do usuário do IAM.

Ações políticas para HealthOmics

Compatível com ações de políticas: sim

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

O elemento `Action` de uma política JSON descreve as ações que podem ser usadas para permitir ou negar acesso em uma política. Incluem ações em uma política para conceder permissões para executar a operação associada.

Para ver uma lista de HealthOmics ações, consulte Ações definidas pela AWS HealthOmics na Referência de autorização de serviço.

As ações de política HealthOmics usam o seguinte prefixo antes da ação:

```
omics
```

Para especificar várias ações em uma única declaração, separe-as com vírgulas.

```
"Action": [
      "omics:action1",
      "omics:action2"
         ]
```

Para ver exemplos de políticas HealthOmics baseadas em identidade da AWS, consulte. Exemplos de políticas baseadas em identidade para AWS HealthOmics

Recursos políticos para HealthOmics

Compatível com recursos de políticas: sim

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

O elemento de política JSON `Resource` especifica o objeto ou os objetos aos quais a ação se aplica. Como prática recomendada, especifique um recurso usando seu nome do recurso da Amazon (ARN). Para ações que não oferecem compatibilidade com permissões em nível de recurso, use um curinga (*) para indicar que a instrução se aplica a todos os recursos.

```
"Resource": "*"
```

Para ver uma lista dos tipos de HealthOmics recursos e seus ARNs, consulte Recursos definidos pela AWS HealthOmics na Referência de autorização de serviço. Para saber com quais ações você pode especificar o ARN de cada recurso, consulte Ações definidas pela AWS. HealthOmics

Para ver exemplos de políticas HealthOmics baseadas em identidade da AWS, consulte. Exemplos de políticas baseadas em identidade para AWS HealthOmics

Chaves de condição de política para HealthOmics

As chaves de condição da política não são suportadas no HealthOmics.

Listas de controle de acesso (ACLs) em HealthOmics

Suportes ACLs: Não

As listas de controle de acesso (ACLs) controlam quais diretores (membros da conta, usuários ou funções) têm permissões para acessar um recurso. ACLs são semelhantes às políticas baseadas em recursos, embora não usem o formato de documento de política JSON.

Controle de acesso baseado em atributos (ABAC) com HealthOmics

Compatível com ABAC (tags em políticas): sim

O controle de acesso por atributo (ABAC) é uma estratégia de autorização que define permissões com base em atributos chamados de tags. Você pode anexar tags a entidades e AWS recursos do IAM e, em seguida, criar políticas ABAC para permitir operações quando a tag do diretor corresponder à tag no recurso.

Para controlar o acesso baseado em tags, forneça informações sobre as tags no elemento de condição de uma política usando as `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` ou chaves de condição `aws:TagKeys`.

Se um serviço for compatível com as três chaves de condição para cada tipo de recurso, o valor será Sim para o serviço. Se um serviço for compatível com as três chaves de condição somente para alguns tipos de recursos, o valor será Parcial

Para obter mais informações sobre o ABAC, consulte Definir permissões com autorização do ABAC no Guia do usuário do IAM. Para visualizar um tutorial com etapas para configurar o ABAC, consulte Usar controle de acesso por atributo (ABAC) no Guia do usuário do IAM.

Para obter mais informações sobre recursos de marcação do HealthOmics, consulte Marcando recursos em HealthOmics.

O exemplo a seguir mostra como você pode escrever uma política do IAM negando acesso a um recurso sem uma tag específica.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "omics:*"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "Null": {
                  "aws:RequestTag/MyCustomTag": "true"
                }
```

```
            }
        }
    ]
}
```

Usando credenciais temporárias com HealthOmics

Compatível com credenciais temporárias: sim

As credenciais temporárias fornecem acesso de curto prazo aos AWS recursos e são criadas automaticamente quando você usa a federação ou troca de funções. AWS recomenda que você gere credenciais temporárias dinamicamente em vez de usar chaves de acesso de longo prazo. Para ter mais informações, consulte Credenciais de segurança temporárias no IAM e Serviços da Serviços da AWS que funcionam com o IAM no Guia do usuário do IAM.

Permissões principais entre serviços para HealthOmics

Compatibilidade com o recurso de encaminhamento de sessões de acesso (FAS): sim

As sessões de acesso direto (FAS) usam as permissões do principal chamando um AWS service (Serviço da AWS), combinadas com a solicitação AWS service (Serviço da AWS) de fazer solicitações aos serviços posteriores. Para obter detalhes da política ao fazer solicitações de FAS, consulte Encaminhamento de sessões de acesso.

Funções de serviço para HealthOmics

Compatível com perfis de serviço: sim

O perfil de serviço é um perfil do IAM que um serviço assume para executar ações em seu nome. Um administrador do IAM pode criar, modificar e excluir um perfil de serviço do IAM. Para obter mais informações, consulte Criar um perfil para delegar permissões a um AWS service (Serviço da AWS) no Guia do Usuário do IAM.

> **Warning**
>
> Alterar as permissões de uma função de serviço pode interromper HealthOmics a funcionalidade. Edite as funções de serviço somente quando HealthOmics fornecer orientação para fazer isso.

Funções vinculadas a serviços para HealthOmics

Compatível com perfis vinculados ao serviço: Não

Uma função vinculada ao serviço é um tipo de função de serviço vinculada a um. AWS service (Serviço da AWS) O serviço pode assumir o perfil de executar uma ação em seu nome. As funções vinculadas ao serviço aparecem em você Conta da AWS e são de propriedade do serviço. Um administrador do IAM pode visualizar, mas não editar as permissões para perfis vinculados ao serviço.

Para obter detalhes sobre como criar ou gerenciar perfis vinculados a serviços, consulte Serviços da AWS que funcionam com o IAM. Encontre um serviço na tabela que inclua um `Yes` na coluna Perfil vinculado ao serviço. Escolha o link Sim para visualizar a documentação do perfil vinculado a serviço desse serviço.

Exemplos de políticas baseadas em identidade para AWS HealthOmics

Por padrão, usuários e funções não têm permissão para criar ou modificar HealthOmics recursos da AWS. Para conceder permissão aos usuários para executar ações nos recursos que eles precisam, um administrador do IAM pode criar políticas do IAM.

Para aprender a criar uma política baseada em identidade do IAM ao usar esses documentos de política em JSON de exemplo, consulte Criar políticas do IAM (console) no Guia do usuário do IAM.

Para obter detalhes sobre ações e tipos de recursos definidos pela AWS HealthOmics, incluindo o formato de cada um dos tipos de recursos, consulte Ações, recursos e chaves de condição para a AWS HealthOmics na Referência de autorização de serviço. ARNs

Tópicos

- Práticas recomendadas de política
- Usar o console do HealthOmics
- Permitir que os usuários visualizem suas próprias permissões

Práticas recomendadas de política

As políticas baseadas em identidade determinam se alguém pode criar, acessar ou excluir HealthOmics recursos da AWS em sua conta. Essas ações podem incorrer em custos para

sua Conta da AWS. Ao criar ou editar políticas baseadas em identidade, siga estas diretrizes e recomendações:

- Comece com as políticas AWS gerenciadas e avance para as permissões de privilégios mínimos — Para começar a conceder permissões aos seus usuários e cargas de trabalho, use as políticas AWS gerenciadas que concedem permissões para muitos casos de uso comuns. Eles estão disponíveis no seu Conta da AWS. Recomendamos que você reduza ainda mais as permissões definindo políticas gerenciadas pelo AWS cliente que sejam específicas para seus casos de uso. Para obter mais informações, consulte Políticas gerenciadas pela AWS ou Políticas gerenciadas pela AWS para funções de trabalho no Guia do usuário do IAM.
- Aplique permissões de privilégio mínimo: ao definir permissões com as políticas do IAM, conceda apenas as permissões necessárias para executar uma tarefa. Você faz isso definindo as ações que podem ser executadas em recursos específicos sob condições específicas, também conhecidas como permissões de privilégio mínimo. Para obter mais informações sobre como usar o IAM para aplicar permissões, consulte Políticas e permissões no IAM no Guia do usuário do IAM.
- Use condições nas políticas do IAM para restringir ainda mais o acesso: é possível adicionar uma condição às políticas para limitar o acesso a ações e recursos. Por exemplo, é possível escrever uma condição de política para especificar que todas as solicitações devem ser enviadas usando SSL. Você também pode usar condições para conceder acesso às ações de serviço se elas forem usadas por meio de uma ação específica AWS service (Serviço da AWS), como CloudFormation. Para obter mais informações, consulte Elementos da política JSON do IAM: condição no Guia do usuário do IAM.
- Use o IAM Access Analyzer para validar suas políticas do IAM a fim de garantir permissões seguras e funcionais: o IAM Access Analyzer valida as políticas novas e existentes para que elas sigam a linguagem de política do IAM (JSON) e as práticas recomendadas do IAM. O IAM Access Analyzer oferece mais de cem verificações de política e recomendações práticas para ajudar a criar políticas seguras e funcionais. Para obter mais informações, consulte Validação de políticas do IAM Access Analyzer no Guia do Usuário do IAM.
- Exigir autenticação multifator (MFA) — Se você tiver um cenário que exija usuários do IAM ou um usuário root, ative Conta da AWS a MFA para obter segurança adicional. Para exigir MFA quando as operações de API forem chamadas, adicione condições de MFA às suas políticas. Para obter mais informações, consulte Configuração de acesso à API protegido por MFA no Guia do Usuário do IAM.

Para obter mais informações sobre as práticas recomendadas do IAM, consulte Práticas recomendadas de segurança no IAM no Guia do usuário do IAM.

Usar o console do HealthOmics

Para acessar o HealthOmics console da AWS, você deve ter um conjunto mínimo de permissões. Essas permissões devem permitir que você liste e visualize detalhes sobre os HealthOmics recursos da AWS em seu Conta da AWS. Caso crie uma política baseada em identidade mais restritiva que as permissões mínimas necessárias, o console não funcionará como pretendido para entidades (usuários ou perfis) com essa política.

Você não precisa permitir permissões mínimas do console para usuários que estão fazendo chamadas somente para a API AWS CLI ou para a AWS API. Em vez disso, permita o acesso somente a ações que correspondam à operação de API que estiverem tentando executar.

Permitir que os usuários visualizem suas próprias permissões

Este exemplo mostra como criar uma política que permita que os usuários do IAM visualizem as políticas gerenciadas e em linha anexadas a sua identidade de usuário. Essa política inclui permissões para concluir essa ação no console ou programaticamente usando a API AWS CLI ou 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 gerenciadas para AWS HealthOmics

Uma política AWS gerenciada é uma política autônoma criada e administrada por AWS. AWS as políticas gerenciadas são projetadas para fornecer permissões para muitos casos de uso comuns, para que você possa começar a atribuir permissões a usuários, grupos e funções.

Lembre-se de que as políticas AWS gerenciadas podem não conceder permissões de privilégio mínimo para seus casos de uso específicos porque elas estão disponíveis para uso de todos os AWS clientes. Recomendamos que você reduza ainda mais as permissões definindo as políticas gerenciadas pelo cliente que são específicas para seus casos de uso.

Você não pode alterar as permissões definidas nas políticas AWS gerenciadas. Se AWS atualizar as permissões definidas em uma política AWS gerenciada, a atualização afetará todas as identidades principais (usuários, grupos e funções) às quais a política está anexada. AWS é mais provável que atualize uma política AWS gerenciada quando uma nova AWS service (Serviço da AWS) for lançada ou novas operações de API forem disponibilizadas para serviços existentes.

Para mais informações, consulte Políticas gerenciadas pela AWS no Guia do usuário do IAM.

AWS política gerenciada: AmazonOmicsFullAccess

Você pode anexar a `AmazonOmicsFullAccess` política às suas identidades do IAM para dar a elas acesso HealthOmics total a.

Essa política concede permissões de acesso total a todas as HealthOmics ações. Quando você cria um repositório de anotações ou variantes, o Omics também fornece acesso a essa loja por meio de um convite de compartilhamento de recursos no console do Resource Access Manager (RAM). Para obter mais informações sobre convites para compartilhamento de recursos por meio do Lake Formation, consulte Compartilhamento de dados entre contas no Lake Formation. Para uma política administrativa do Omics, você também precisa das seguintes permissões para acessar seu bucket do 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 gerenciada: AmazonOmicsReadOnlyAccess

Você pode anexar a `AWSOmicsReadOnlyAccess` política às suas identidades do IAM quando quiser limitar as permissões dessa identidade ao acesso somente para leitura.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "omics:Get*",
                "omics:List*"
            ],
            "Resource": "*"
        }
    ]
}
```

HealthOmics atualizações nas políticas AWS gerenciadas

Veja detalhes sobre as atualizações das políticas AWS gerenciadas HealthOmics desde que esse serviço começou a rastrear essas alterações. Para receber alertas automáticos sobre alterações nessa página, assine o feed RSS na página Histórico do HealthOmics documento.

Alteração	Descrição	Data
AmazonOmicsFullAccess - Nova política adicionada	HealthOmics adicionou uma nova política para conceder ao usuário acesso total a todas as ações e recursos. Para saber mais, consulte AmazonOmicsFullAccess.	23 de fevereiro de 2023
HealthOmics começou a rastrear as alterações	HealthOmics começou a rastrear as mudanças em suas políticas AWS gerenciadas.	29 de novembro de 2022
AmazonOmicsReadOnlyAccess - Nova política adicionada	HealthOmics adicionou uma nova política que limita o acesso somente para leitura. Para saber mais, AmazonOmicsReadOnlyAccess.	29 de novembro de 2022

Solução de problemas AWS HealthOmics de identidade e acesso

Use as informações a seguir para ajudá-lo a diagnosticar e corrigir problemas comuns que você pode encontrar ao trabalhar com a AWS HealthOmics e o IAM.

Tópicos

- Não estou autorizado a realizar uma ação em HealthOmics
- Não estou autorizado a realizar iam: PassRole
- Quero permitir que pessoas fora da minha Conta da AWS acessem meus HealthOmics recursos

Não estou autorizado a realizar uma ação em HealthOmics

Se você receber uma mensagem de erro informando que não tem autorização para executar uma ação, suas políticas deverão ser atualizadas para permitir que você realize a ação.

O erro do exemplo a seguir ocorre quando o usuário do IAM `mateojackson` tenta usar o console para visualizar detalhes sobre um atributo *`my-example-widget`* fictício, mas não tem as permissões `omics:`*`GetWidget`* fictícias.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 omics:GetWidget on resource: my-example-widget
```

Nesse caso, a política do usuário `mateojackson` deve ser atualizada para permitir o acesso ao recurso *`my-example-widget`* usando a ação `omics:`*`GetWidget`*.

Se precisar de ajuda, entre em contato com seu AWS administrador. Seu administrador é a pessoa que forneceu suas credenciais de login.

Não estou autorizado a realizar iam: PassRole

Se você receber um erro informando que não está autorizado a realizar a `iam:PassRole` ação, suas políticas devem ser atualizadas para permitir que você passe uma função para a AWS HealthOmics.

Alguns Serviços da AWS permitem que você passe uma função existente para esse serviço em vez de criar uma nova função de serviço ou uma função vinculada ao serviço. Para fazer isso, é preciso ter permissões para passar o perfil para o serviço.

O exemplo de erro a seguir ocorre quando um usuário do IAM chamado `marymajor` tenta usar o console para realizar uma ação na AWS HealthOmics. No entanto, a ação exige que o serviço tenha permissões concedidas por um perfil de serviço. Mary não tem permissões para passar o perfil para o serviço.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

Nesse caso, as políticas de Mary devem ser atualizadas para permitir que ela realize a ação `iam:PassRole`.

Se precisar de ajuda, entre em contato com seu AWS administrador. Seu administrador é a pessoa que forneceu suas credenciais de login.

Quero permitir que pessoas fora da minha Conta da AWS acessem meus HealthOmics recursos

É possível criar um perfil que os usuários de outras contas ou pessoas fora da organização podem usar para acessar seus recursos. É possível especificar quem é confiável para assumir o perfil. Para serviços que oferecem suporte a políticas baseadas em recursos ou listas de controle de acesso (ACLs), você pode usar essas políticas para conceder às pessoas acesso aos seus recursos.

Para saber mais, consulte:

- Para saber se a AWS HealthOmics oferece suporte a esses recursos, consulteComo AWS HealthOmics funciona com o IAM.
- Para saber como fornecer acesso aos seus recursos em todos os Contas da AWS que você possui, consulte Como fornecer acesso a um usuário do IAM em outro Conta da AWS que você possui no Guia do usuário do IAM.
- Para saber como fornecer acesso aos seus recursos a terceiros Contas da AWS, consulte Como fornecer acesso Contas da AWS a terceiros no Guia do usuário do IAM.
- Para saber como conceder acesso por meio da federação de identidades, consulte Conceder acesso a usuários autenticados externamente (federação de identidades) no Guia do usuário do IAM.
- Para conhecer a diferença entre perfis e políticas baseadas em recurso para acesso entre contas, consulte Acesso a recursos entre contas no IAM no Guia do usuário do IAM.

Validação de conformidade para AWS HealthOmics

Auditores terceirizados avaliam a segurança e a conformidade AWS HealthOmics como parte de vários programas de AWS conformidade. Isso inclui HIPAA, FedRAMP e outros. A tabela a seguir mostra as certificações de conformidade do HealthOmics serviço.

Certificação	Link
HIPAA	Referência de serviços qualificados pela HIPAA

Certificação	Link
HiTrust-CSF	Estrutura de segurança comum da Health Information Trust Alliance
FedRAMP Moderado (Leste/Oeste)	Programa federal de gerenciamento de riscos e autorizações
ESTRELA ISO/CSA	Certificação ISO e CSA STAR
C5	Catálogo de controles de conformidade de computação em nuvem
DoD CC SRG IL2	Guia de requisitos de segurança de computação em nuvem do Departamento de Defesa
ENS High	Esquema Nacional de Segurança
FINMA	Autoridade Supervisora do Mercado Financeiro Suíço
É MAPA	Programa de avaliação e gerenciamento de segurança do sistema de informação
OSPAR	Relatório de auditoria do provedor de serviços terceirizado
PCI	Padrão de segurança de dados do setor de cartões de pagamento
Pinakes	Associação bancária CCI - Qualificação de terceiros
PiTuKri	Critérios para avaliar a segurança da informação dos serviços em nuvem
SOC 1,2,3	Controles do sistema e da organização

Para obter uma lista de todos os AWS serviços no escopo de programas de conformidade específicos, consulte AWS Services in Scope by Compliance Program . Para obter informações gerais, consulte Programas de conformidade da AWS.

Você pode baixar relatórios de auditoria de terceiros usando AWS Artifact. Para obter mais informações, consulte Baixar relatórios em AWS Artifact .

HealthOmics os armazenamentos de dados usam o ID de amostra para nomeação interna de arquivos e para marcar recursos. Antes de ingerir dados, verifique se o ID da amostra contém algum dado de PHI. Se isso acontecer, altere a ID da amostra antes de ingerir os dados. Para obter mais informações, consulte as orientações na página da web de conformidade com a AWS HIPAA.

Sua responsabilidade de conformidade ao usar AWS HealthOmics é determinada pela confidencialidade de seus dados, pelos objetivos de conformidade de sua empresa e pelas leis e regulamentações aplicáveis. AWS fornece os seguintes recursos para ajudar na conformidade:

- Guias de início rápido de segurança e compatibilidade: esses guias de implantação abordam as considerações de arquitetura e fornecem etapas para implantação de ambientes de linha de base focados em compatibilidade e segurança na AWS.
- Documento técnico sobre arquitetura para segurança e conformidade com a HIPAA — Este whitepaper descreve como as empresas podem usar para criar aplicativos compatíveis com a HIPAA. AWS
- AWS Recursos de https://aws.amazon.com/compliance/resources/ de conformidade — Essa coleção de pastas de trabalho e guias pode ser aplicada ao seu setor e local.
- Avaliação de recursos com regras no Guia do AWS Config Desenvolvedor — AWS Config avalia o quão bem suas configurações de recursos estão em conformidade com as práticas internas, as diretrizes do setor e os regulamentos.
- AWS Security Hub CSPM— Esse AWS serviço fornece uma visão abrangente do seu estado de segurança interno, AWS que ajuda você a verificar sua conformidade com os padrões e as melhores práticas do setor de segurança.

Resiliência em HealthOmics

A infraestrutura AWS global é construída em torno Regiões da AWS de zonas de disponibilidade. Regiões da AWS fornecem várias zonas de disponibilidade fisicamente separadas e isoladas, conectadas a redes de baixa latência, alta taxa de transferência e alta redundância. Com as zonas

de disponibilidade, é possível projetar e operar aplicações e bancos de dados que automaticamente executam o failover entre as zonas sem interrupção. As zonas de disponibilidade são altamente disponíveis, tolerantes a falhas e escaláveis que uma ou várias infraestruturas de data center tradicionais.

Para obter mais informações sobre zonas de disponibilidade Regiões da AWS e zonas de disponibilidade, consulte Infraestrutura AWS global.

Além da infraestrutura AWS global, a AWS HealthOmics oferece vários recursos para ajudar a apoiar suas necessidades de resiliência e backup de dados.

AWS HealthOmics e endpoints VPC de interface ()AWS PrivateLink

Você pode estabelecer uma conexão privada entre sua VPC e criar uma AWS HealthOmics interface VPC endpoint. Os endpoints de interface são alimentados por AWS PrivateLinkuma tecnologia que você pode usar para acessar de forma privada as operações de HealthOmics API sem um gateway de internet, dispositivo NAT, conexão VPN ou conexão do AWS Direct Connect. As instâncias na sua VPC não exigem endereços IP públicos para se comunicar com as operações HealthOmics da API. O tráfego entre sua VPC e o tráfego HealthOmics não sai da rede Amazon.

Cada endpoint de interface é representado por uma ou mais Interfaces de Rede Elástica nas sub-redes.

Para obter mais informações, consulte Interface VPC endpoints (AWS PrivateLink) no Guia do usuário da Amazon VPC.

As políticas de endpoint de VPC são compatíveis com todas as regiões HealthOmics , exceto Israel (Tel Aviv). Por padrão, o acesso total ao HealthOmics é permitido por meio do endpoint.

Considerações sobre HealthOmics VPC endpoints

Antes de configurar uma interface para o VPC endpoint HealthOmics, certifique-se de revisar as propriedades e limitações do endpoint da interface no Guia do usuário do Amazon VPC.

HealthOmics suporta fazer chamadas para todas as ações da API HealthOmics de armazenamento a partir da sua VPC.

As políticas de VPC endpoint não são suportadas HealthOmics por padrão, mas você pode criar um VPC endpoint para acesso total HealthOmics às operações de armazenamento. HealthOmics Para

mais informações, consulte Controlar o acesso a serviços com VPC endpoints no Guia do usuário da Amazon VPC.

Criar um endpoint da VPC de interface para o HealthOmics

Você pode criar um VPC endpoint para o HealthOmics serviço usando o console Amazon VPC ou o (). AWS Command Line Interface AWS CLI Para obter mais informações, consulte Criar um endpoint de interface no Guia do usuário da Amazon VPC.

Crie um VPC endpoint para HealthOmics usando os seguintes nomes de serviço:

- 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-comics

As regiões Leste dos EUA (Norte da Virgínia) e Oeste dos EUA (Oregon) oferecem suporte a endpoints AWS PrivateLink FIPS. Para essas regiões, você também pode usar os seguintes nomes de serviço:

- 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

Se você ativar o DNS privado para o endpoint, poderá fazer solicitações de API HealthOmics usando o nome DNS padrão para a região, por exemplo,. `omics.us-east-1.amazonaws.com`

Para obter mais informações, consulte Acessar um serviço por um endpoint de interface no Manual do Usuário do Amazon VPC.

Criando uma política de endpoint da VPC para o HealthOmics

É possível anexar uma política de endpoint ao endpoint da VPC que controla o acesso ao HealthOmics. Essa política especifica as seguintes informações:

- A entidade principal que pode executar ações
- As ações que podem ser executadas
- Os recursos nos quais as ações podem ser executadas

Para mais informações, consulte Controlar o acesso a serviços com VPC endpoints no Guia do usuário da Amazon VPC.

Exemplo: política de VPC endpoint para ações. HealthOmics

Veja a seguir um exemplo de uma política de endpoint para HealthOmics. Quando anexada a um endpoint, essa política concede acesso às HealthOmics ações para todos os diretores em todos os 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\":\"*\"}]}"
```

Considerações especiais para acessar conjuntos de leitura usando o Amazon S3 URIs

Para acessar conjuntos de leitura por meio do Amazon S3 URIs quando você estiver usando uma conexão privada, configure os endpoints da PrivateLink interface no armazenamento de sequências. Depois de configurá-los, os endpoints têm os seguintes formatos:

```
com.amazonaws.region.storage-omics
com.amazonaws.region.control-storage-omics
```

Para usar endpoints de gateway, siga o guia Endpoints de gateway para Amazon S3 para configurar seus endpoints de gateway. HealthOmics possui o bucket do Amazon S3, então você não precisa criar ou ajustar a política do bucket. Os endpoints do gateway dependem da política vinculada ao usuário ou à função que acessa os dados, mas você também pode configurar endpoints com políticas mais restritivas. Essas políticas podem incluir restrições de acesso com base no ARN do ponto de acesso do Amazon S3 e nas ações do Amazon S3.

Monitoramento da AWS HealthOmics

O monitoramento é uma parte importante da manutenção da confiabilidade, disponibilidade e desempenho da AWS HealthOmics e de suas outras AWS soluções. AWS fornece as seguintes ferramentas de monitoramento para monitorar a AWS HealthOmics, relatar quando algo está errado e realizar ações automáticas quando apropriado:

- A Amazon CloudWatch monitora seus AWS recursos e os aplicativos em que você executa AWS em tempo real. Você pode coletar e rastrear métricas, criar painéis personalizados e definir alarmes que o notificam ou que realizam ações quando uma métrica especificada atinge um limite definido. Por exemplo, você pode CloudWatch rastrear o uso da CPU ou outras métricas de suas EC2 instâncias da Amazon e iniciar automaticamente novas instâncias quando necessário. Para obter mais informações, consulte o Guia CloudWatch do usuário da Amazon.
- O Amazon CloudWatch Logs permite que você monitore, armazene e acesse seus arquivos de log de EC2 instâncias da Amazon e de outras fontes. CloudTrail CloudWatch Os registros podem monitorar as informações nos arquivos de log e notificá-lo quando determinados limites forem atingidos. É possível também arquivar seus dados de log em armazenamento resiliente. Para obter mais informações, consulte o Guia do usuário do Amazon CloudWatch Logs.
- O AWS CloudTrail captura chamadas de API e eventos relacionados feitos por sua conta da Conta da AWS ou em nome dela e entrega os arquivos de log a um bucket do Amazon S3 que você especificar. Você pode identificar quais usuários e contas chamaram AWS, o endereço IP de origem de onde as chamadas foram feitas e quando elas ocorreram. Para mais informações, consulte o Guia do usuário do AWS CloudTrail.
- EventBridgeA Amazon é um serviço de ônibus de eventos sem servidor que facilita a conexão de seus aplicativos com dados de várias fontes. EventBridge fornece um fluxo de dados em tempo real de seus próprios aplicativos, aplicativos Software-as-a-Service (SaaS) e AWS serviços e encaminha esses dados para destinos como o Lambda. Isso permite monitorar eventos que ocorram em serviços e criem arquiteturas orientadas a eventos. Para obter mais informações, consulte o Guia EventBridge do usuário da Amazon.

> **Note**
>
> Para atualizações de serviços, configure e monitore seu Personal Health Dashboard. Para obter mais informações sobre como gerenciar o painel, consulte Getting started with your AWS Health Dashboard.

Tópicos

Registro de acesso ao S3

Você pode monitorar o acesso à API do Amazon S3 para HealthOmics sequenciar dados do armazenamento usando os registros de acesso criados pela loja. Você pode usar CloudWatch para monitorar o acesso ao S3 a partir das operações da HealthOmics API. CloudWatch fornece visibilidade do acesso ao Amazon S3 proveniente de sua própria conta. Se você, como proprietário dos dados, compartilhar o acesso a uma conta de terceiros, o registro de acesso não estará disponível em CloudWatch. Em vez disso, use o S3 Access Log. da loja, que registra todo o acesso do S3 aos dados no bucket Amazon S3 configurado.

Configure os registros de acesso do S3 usando as operações da `UpdateSequenceStore` API `CreateSequenceStore` ou. Além disso, certifique-se de que o principal do HealthOmics serviço (`omics.amazonaws.com`) tenha `s3:PutObject` permissões para o prefixo S3 configurado.

> **Note**
>
> Os registros usam a configuração de criptografia padrão do bucket de destino. Se o bucket usar uma chave gerenciada pelo cliente, o responsável pelo serviço deverá ter acesso para usar a chave para gravação.

Para desativar o registro de acesso, use `UpdateSequenceStore` e defina a configuração do registro de acesso como em branco.

Monitoramento HealthOmics com CloudWatch métricas

Você pode monitorar HealthOmics o uso CloudWatch, que coleta dados brutos e os processa em métricas legíveis e quase em tempo real. Essas estatísticas são mantidas por 15 meses, de maneira

que você possa acessar informações históricas e ter uma perspectiva melhor de como o aplicativo web ou o serviço está se saindo. Você também pode definir alarmes que observam determinados limites e enviam notificações ou realizam ações quando esses limites são atingidos. Para obter mais informações, consulte o Guia CloudWatch do usuário da Amazon.

O AWS HealthOmics serviço relata as seguintes métricas no `AWS/Omics` namespace.

As métricas de contagem de chamadas da API são relatadas a seguir AWS HealthOmics APIs. Somente a dimensão Operação da API é relatada.

- Loja de referência e referência APIs —CreateReferenceStore, DeleteReferenceStore, StartReferenceImportJob
- Conjunto de armazenamento e leitura de sequências APIs — CreateSequenceStore DeleteSequenceStore,, StartReadSetImportJob, StartReadSetActivationJob, StartReadSetExportJob
- Loja de variantes APIs — CreateVariantStore, DeleteVariantStore, StartVariantImportJob, CancelVariantImportJob
- Loja de anotações APIs — CreateAnnotationStore,,, DeleteAnotationStore StartAnnotationImportJob CancelAnnotationImportJob
- Fluxo de trabalho, execução e grupo de execução APIs — CreateWorkflow DeleteWorkflow, StartRun, CancelRun, DeleteRun, CreateRunGroup, DeleteRunGroup

Visualizar métricas do *AWS HealthOmics*

CloudWatch as métricas para AWS HealthOmics podem ser visualizadas no CloudWatch console.

Para visualizar métricas (CloudWatch console)

1. Faça login no Console de Gerenciamento da AWS e abra o console do CloudWatch .
2. Escolha Métricas, escolha Todas as métricas e, em seguida, escolha AWS/Usage.
3. Serviço de filtro para AWS HealthOmics.
4. Escolha a dimensão, informe um nome de métrica e selecione Adicionar ao gráfico.
5. Escolha um valor para o intervalo de datas. A contagem da métrica para o intervalo de datas selecionado é exibida no gráfico.

Criando um alarme usando CloudWatch

Um CloudWatch alarme monitora uma única métrica durante um período de tempo especificado e executa uma ou mais ações: enviar uma notificação para um tópico do Amazon Simple Notification Service (Amazon SNS) ou política de Auto Scaling. A ação ou ações são baseadas no valor da métrica em relação a um determinado limite em vários períodos que você especifica. CloudWatch também pode enviar uma mensagem do Amazon SNS quando o alarme muda de estado.

CloudWatch os alarmes invocam ações somente quando o estado muda e persiste durante o período especificado.

Para visualizar métricas (CloudWatch console)

1. Faça login no Console de Gerenciamento da AWS e abra o console do CloudWatch .
2. Escolha Alarmes e, em seguida, Criar alarme.
3. Escolha AWS/Usage e, em seguida, escolha uma AWS HealthOmics métrica usando a dimensão Serviço.
4. Para Intervalo de tempo, escolha um intervalo de tempo para monitorar e, em seguida, selecione Avançar.
5. Preencha os campos Nome e Descrição.
6. Em Whenever, escolha **>=** e digite um valor máximo.
7. Se você quiser CloudWatch enviar um e-mail quando o estado do alarme for atingido, na seção Ações, para Sempre que este alarme for atingido, escolha Estado é ALARME. Em Enviar notificação para, escolha uma lista de endereçamento ou escolha Nova lista e crie uma nova lista de endereçamento.
8. Visualize o alarme na seção Prévia do alarme. Se você estiver satisfeito com o alarme, selecione Criar alarme.

Monitoramento HealthOmics com CloudWatch registros

HealthOmics gera uma variedade de registros para ajudar você a entender e solucionar problemas em suas execuções. Os registros estão disponíveis em dois lugares: CloudWatch e no Amazon S3.

Por padrão, as execuções têm o registro ativado. Opcionalmente, você pode desativar o registro `LogLevel = OFF` em log para uma execução configurando a startrun solicitação.

> **Note**
>
> Para atualizações de serviços, configure e monitore seu Personal Health Dashboard. Para obter mais informações sobre como gerenciar o painel, consulte Getting started with your AWS Health Dashboard.

Tópicos

Tipos de log para HealthOmics fluxos de trabalho

HealthOmics fornece os seguintes tipos de registros para fluxos de trabalho:

- Registros do mecanismo — Os mecanismos de fluxo de trabalho subjacentes (Nextflow, WDL e CWL) produzem registros do mecanismo para execuções. Esses registros podem ajudar você a solucionar problemas de definição do fluxo de trabalho.
- Executar registros de manifesto — Esses registros fornecem informações de alto nível sobre cada tarefa executada, como status da tarefa, hora de início, hora de parada e motivo da falha (se a tarefa falhar).

  Os registros de manifestos de execução também relatam estatísticas de utilização de recursos que podem ser úteis para entender as oportunidades de otimização de recursos. Essas estatísticas incluem:

  - Média de CPUs
  - CPUs máximos
  - CPUs reservadas
  - GPUs reservadas
  - memoryAverageGiB
  - memoryMaximumGiB

  - memoryReservedGiB
  - Segundos de execução
- Registros de execução — Os registros de execução fornecem o status geral da execução e o horário em que as tarefas individuais são iniciadas, executadas, interrompidas e concluídas. Os registros de execução também oferecem visibilidade das etapas de importação e exportação de arquivos.
- Registros de tarefas — Os registros de tarefas fornecem informações detalhadas sobre tarefas individuais em sua execução. As saídas em seu registro de tarefas dependem da definição da tarefa e de onde você usa as instruções de log em seu código. Se seus registros de tarefas não fornecerem o nível de insight de que você precisa, considere adicionar instruções de registro adicionais à sua definição de tarefas para produzir registros de tarefas mais detalhados.
- Executar registros de cache — Os registros de execução de cache fornecem o status geral dos caches de execução e o armazenamento em cache das saídas das tarefas. Os registros de execução do cache oferecem visibilidade dos acertos e erros do cache em cada execução que usa o cache.
- outputs.json — Para fluxos de trabalho de WDL e CWL, HealthOmics entrega um arquivo gerado pelo mecanismo, chamado, para seu bucket do Amazon S3 após `outputs.json` a conclusão da execução. Esses arquivos incluem uma lista e um mapa de todas as saídas para a execução.

Login CloudWatch

CloudWatch gera registros de fluxo de trabalho para execuções com falha e execuções bem-sucedidas. Todos os registros estão disponíveis para execuções com falha e execuções bem-sucedidas, exceto que os registros do mecanismo estão disponíveis somente para execuções com falha.

Você pode encontrar os registros do CloudWatch fluxo de trabalho no seguinte grupo de registros:`/aws/omics/WorkflowLog`. Além disso, a saída da operação da API get-run fornece o fluxo de registros ARNs para os CloudWatch registros do mecanismo e os registros de execução.

Por padrão, AWS mantém os CloudWatch registros indefinidamente. Você pode ajustar a política de retenção do grupo de registros para definir um período de retenção entre 10 anos e um dia.

A tabela a seguir fornece um resumo dos CloudWatch Logins HealthOmics. Todos os registros do fluxo de trabalho estão disponíveis para execuções bem-sucedidas e com falha, exceto que os registros do mecanismo estão disponíveis somente para execuções com falha.

Nome do log	Disponível em CloudWatch registros	Quando o registro está disponível	Formato de fluxo de log
Registros do motor	Sim, para execuções com falha	Após a conclusão da execução	executar/ /motor *runID*
Executar registros de manifestos	Sim	Após a conclusão da execução	manifesto/execução //*runIDrunUUID*
Registros de execução	Sim	Em tempo real	correr/ *runID*
Registros de tarefas	Sim	Em tempo real	executar/ /tarefa/ *runID taskID*
Executar registros de cache	Sim	Em tempo real	Execute Cache//*runCacheId runCacheUUID*
outputs.json (WDL e CWL)	Não	n/a	n/a

Logs no Amazon S3

Somente os registros do motor e o `outputs.json` arquivo são entregues ao Amazon S3.

Após a conclusão da execução, os registros do mecanismo são entregues ao seu bucket do S3 e ficam disponíveis indefinidamente até que você os exclua. Esses registros estão localizados no diretório de registros do URI de saída do S3 que você especificou para o fluxo de trabalho.

O caminho para o diretório de registros tem o seguinte formato:`s3://{user_provided_path}/logs/`.

A tabela a seguir fornece um resumo dos HealthOmics registros disponíveis em seu bucket do Amazon S3.

Nome do log	Disponível no Amazon S3	Quando o registro está disponível	Caminho do fluxo de log
Registros do motor	Sim	Após a conclusão da execução	s3:///logs/engine.log *user_provided_path*
outputs.json (WDL e CWL)	Sim	Após a conclusão da execução	s3://*user_provided_path* /*runID*/*runUUID*/logs/outputs.json
Execute registros de manifestos, registros de execução e registros de tarefas	Não	n/a	n/a

CloudWatch Registros interativos na CLI

Você pode visualizar interativamente os CloudWatch registros usando o comando Live Tail no modo interativo. Você pode acompanhar o progresso da execução em tempo real e definir até 5 palavras-chave para destacar nos registros:

```
aws logs start-live-tail  \
  --mode interactive  \
  --log-group-identifiers arn:aws:logs:region:account-ID:log-group:/aws/omics/
WorkflowLog
```

Para obter mais informações, consulte Start live tail na Referência de AWS CLI Comandos.

Acessando CloudWatch registros a partir do console

Para acessar os registros de uma execução, você pode vinculá-los diretamente na página de detalhes da execução no HealthOmics console.

1. Abra o console de HealthOmics .
2. Se necessário, abra o painel de navegação esquerdo (≡). Selecione Execuções.

3. Selecione a execução na tabela Execuções.
4. Na página de detalhes da execução, você pode escolher qualquer uma dessas ações:

   a. Em Resumo da execução, escolha Exibir registros de execução. O console abre os registros de execução no CloudWatch console.
   b. Em Resumo da execução, escolha Exibir registros no Amazon S3. O console abre a pasta de registros no console do Amazon S3.
   c. Em Executar tarefas, escolha Exibir registros, Exibir registros de execução ou Exibir registros de manifesto de execução de uma tarefa. O console abre os registros no CloudWatch console.

Você também pode navegar até os registros no CloudWatch console:

1. Abra o CloudWatch console https://console.aws.amazon.com/cloudwatch/.
2. No menu à esquerda, escolha Grupos de registros.
3. Selecione o grupo do `/aws/omics/WorkflowLog`.

   Se a lista de grupos de registros for longa, você poderá inserir ômicas na caixa de texto de pesquisa para restringir a lista.
4. Quando a página de detalhes do grupo de registros for aberta, escolha o fluxo de registros que você deseja visualizar. O console exibe os eventos desse fluxo de log.

Registrando chamadas de AWS HealthOmics API usando AWS CloudTrail

AWS HealthOmics é integrado com AWS CloudTrail, um serviço que fornece um registro das ações realizadas por um usuário, função ou AWS serviço em HealthOmics. CloudTrail captura todas as chamadas de API HealthOmics como eventos. As chamadas capturadas incluem chamadas do HealthOmics console e chamadas de código para as operações HealthOmics da API. Se você criar uma trilha, poderá habilitar a entrega contínua de CloudTrail eventos para um bucket do Amazon S3, incluindo eventos para. HealthOmics Se você não configurar uma trilha, ainda poderá ver os eventos mais recentes no CloudTrail console no Histórico de eventos. Usando as informações coletadas por CloudTrail, você pode determinar a solicitação que foi feita HealthOmics, o endereço IP do qual a solicitação foi feita, quem fez a solicitação, quando ela foi feita e detalhes adicionais.

Para saber mais sobre isso CloudTrail, consulte o Guia AWS CloudTrail do usuário.

HealthOmics informações em CloudTrail

CloudTrail é ativado no seu Conta da AWS quando você cria a conta. Quando a atividade ocorre em HealthOmics, essa atividade é registrada em um CloudTrail evento junto com outros eventos AWS de serviço no histórico de eventos. Você pode visualizar, pesquisar e baixar eventos recentes no seu Conta da AWS. Para obter mais informações, consulte Visualização de eventos com histórico de CloudTrail eventos.

Para um registro contínuo dos eventos em sua Conta da AWS, incluindo eventos para HealthOmics, crie uma trilha. Uma trilha permite CloudTrail entregar arquivos de log para um bucket do Amazon S3. Por padrão, quando você cria uma trilha no console, ela é aplicada a todas as Regiões da AWS. A trilha registra eventos de todas as regiões na AWS partição e entrega os arquivos de log ao bucket do Amazon S3 que você especificar. Além disso, você pode configurar outros AWS serviços para analisar e agir com base nos dados de eventos coletados nos CloudTrail registros. Para obter mais informações, consulte:

- Visão geral da criação de uma trilha
- CloudTrail serviços e integrações suportados
- Configurando notificações do Amazon SNS para CloudTrail
- Recebendo arquivos de CloudTrail log de várias regiões e Recebendo arquivos de CloudTrail log de várias contas

Todas HealthOmics as ações são registradas CloudTrail e documentadas na Referência da AWS HealthOmics API. Por exemplo, chamadas para o `CreateReferenceeStore` `StartVariantImportJob` e `CreateWorkflow` as ações geram entradas nos arquivos de CloudTrail log.

Cada entrada de log ou evento contém informações sobre quem gerou a solicitação. As informações de identidade ajudam a determinar o seguinte:

- Se a solicitação foi feita com as credenciais do usuário do IAM.
- Se a solicitação foi feita com credenciais de segurança temporárias de uma função ou de um usuário federado.
- Se a solicitação foi feita por outro AWS serviço.

Para obter mais informações, consulte Elemento userIdentity do CloudTrail .

Entendendo as entradas do arquivo de HealthOmics log

Uma trilha é uma configuração que permite a entrega de eventos como arquivos de log para um bucket do Amazon S3 que você especificar. CloudTrail os arquivos de log contêm uma ou mais entradas de log. Um evento representa uma única solicitação de qualquer fonte e inclui informações sobre a ação solicitada, a data e a hora da ação, os parâmetros da solicitação e assim por diante. CloudTrail os arquivos de log não são um rastreamento de pilha ordenado das chamadas públicas de API, portanto, eles não aparecem em nenhuma ordem específica.

O exemplo a seguir mostra uma entrada de CloudTrail registro que demonstra a CreateWorkflow ação.


```
{
    "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"
}
```


Usando EventBridge com AWS HealthOmics

HealthOmics envia eventos para a Amazon EventBridge quando os recursos mudam de status. Os recursos incluem trabalhos de importação, trabalhos de exportação, compartilhamentos de recursos, fluxos de trabalho, tarefas e execuções. Para cada tipo de recurso, há uma lista de mudanças de status que geram um evento.

Um barramento de eventos é um roteador que recebe eventos e os entrega aos destinos. Sua conta inclui um barramento de eventos padrão que recebe automaticamente eventos dos AWS serviços. Você pode criar ônibus de eventos personalizados adicionais.

Você cria EventBridge regras para especificar as ações a serem tomadas quando o barramento de eventos recebe eventos. Por exemplo, você pode criar uma regra que o notifique sobre as mudanças de status de um recurso.

Os cenários comuns para o uso de eventos incluem:

- Para monitorar quando um usuário compartilha um recurso com você ou revoga o compartilhamento.
- Para monitorar se uma execução falha ou é concluída com êxito.

Para obter mais informações sobre o uso EventBridge, consulte O que é a Amazon EventBridge?

Tópicos

Configurado EventBridge para HealthOmics

Antes de monitorar os EventBridge eventos, crie um EventBridge ônibus e crie regras para os eventos de interesse.

Configurar um EventBridge barramento

Você pode usar o barramento de eventos padrão para você Conta da AWS ou configurar um barramento de eventos personalizado. Para configurar um barramento de eventos personalizado, siga estas etapas:

1. Abra o EventBridge console: https://console.aws.amazon.com/events/.
2. No painel de navegação à esquerda, escolha Ônibus de eventos.
3. Selecione Create event bus (Criar barramento de eventos).
4. No formulário Criar barramento de eventos, insira um nome para o ônibus.
5. Escolha Criar para criar o barramento.

Crie uma EventBridge regra

O procedimento a seguir mostra como criar uma regra simples. Para obter mais informações sobre regras, consulte Regras em EventBridge.

1. Abra o EventBridge console: https://console.aws.amazon.com/events/.
2. Na barra de navegação à esquerda, selecione Rules (Regras).
3. Escolha Criar regra. O console abre o formulário Criar regra.
4. Em Definir detalhes da regra, forneça um nome para a regra.

    - Em Nome, insira um nome para o ônibus.

- Para Barramento de eventos, selecione o barramento para essa regra.
- Escolha Próximo.

5. Em Criar padrão de evento, em Origem do evento, selecione Eventos da AWS ou eventos de EventBridge parceiros.
6. Role para baixo até Padrão de eventos.
   a. Em Origem do evento, selecione os serviços da AWS.
   b. Para o serviço AWS, insira ômicas no filtro de texto e selecione AWS HealthOmicscomo serviço.
   c. Em Tipo de evento, selecione o evento de interesse (ou Todos os eventos).
   d. Escolha Próximo.
7. Em Selecionar alvo (s), selecione um alvo para o evento. Por exemplo, escolha o serviço da AWS, o grupo de CloudWatch registros escolhido e configure um grupo de registros.

   Para muitos tipos de destino, o EventBridge precisa de permissão para enviar eventos ao destino. O console cria essas permissões para você.
8. (Opcional) Em Configurar tags, associe as tags à regra.
9. Em Revisar e atualizar, revise a configuração e escolha Criar regra.

EventBridge eventos em HealthOmics

A tabela a seguir lista os eventos HealthOmics enviados para EventBridge e a lista de valores de status possíveis para o evento.

Nome do evento	Valores de status possíveis
Alteração do status do trabalho de importação de anotações	Enviado, em andamento, cancelado, concluído, com falha ou concluído com falhas
Alteração do status de compartilhamento do Annotation Store	Pendente, ativando, ativo, excluindo, excluído, falhou
Alteração do status do Annotation Store	Falha na criação, criação, atualização, atualização, exclusão, exclusão ou criação

Nome do evento	Valores de status possíveis
Alteração do status do trabalho de ativação do Read Set	Enviado, em andamento, concluído, com falha ou concluído com falhas
Read Set Export Job Status Change	Enviado, em andamento, concluído, com falha ou concluído com falhas
Read Set Import Job Status Change	Enviado, em andamento, concluído, com falha ou concluído com falhas
Alteração de status do conjunto de leitura	Processamento de upload, falha no upload, ativo, arquivado, ativado ou excluído
Alteração do status do trabalho de importação de referência	Enviado, em andamento, concluído, com falha ou concluído com falhas
Alteração do status de referência	Ativo ou excluído
Alteração do status do repositório de referência	Criado, atualizado, ativo ou excluído
Alteração do status de execução	Pendente, iniciando, em execução, parando, concluído, excluído, com falha ou cancelado
Alteração do status do Sequence Store	Criado, atualizado, ativo ou excluído
Alteração do status da tarefa	Pendente, iniciando, em execução, parando, concluído, excluído, com falha ou cancelado
Alteração do status do trabalho de importação de variantes	Enviado, em andamento, cancelado, concluído, com falha ou concluído com falhas
Alteração do status de compartilhamento da Variant Store	Pendente, ativando, ativo, excluindo, excluído, falhou
Alteração do status da loja de variantes	Falha na criação, criação, atualização, atualização, exclusão, exclusão ou criação
Alteração do status de compartilhamento do fluxo de	Pendente, ativando, ativo, excluindo, excluído, falhou

Nome do evento	Valores de status possíveis
Alteração do status do fluxo de	Sucesso na criação, falha na criação, sucesso na exclusão ou falha na exclusão

Estrutura de mensagens de evento

HealthOmics fornece o melhor esforço para enviar mensagens de eventos de mudança de estado para EventBridge. O evento é um objeto com estrutura JSON que também contém detalhes de metadados. Você pode usar os metadados como entrada para recriar o evento ou obter mais informações. Os eventos incluem os seguintes campos:

- `version`— Atualmente 0 (zero) para todos os eventos.
- `id`— Um UUID da versão 4 gerado para cada evento.
- `detail-type`— O tipo de evento que está sendo enviado.
- `account`— O Conta da AWS ID de 12 dígitos do proprietário do bucket.
- `source`— Identifica o serviço que gerou o evento.
- `time`— A hora em que o evento ocorreu.
- `region`— Identifica o Região da AWS do bucket.
- `resources`— Uma matriz JSON que contém o Amazon Resource Name (ARN) do bucket.
- `detail`— Um objeto JSON que contém informações sobre o evento.

Os eventos de execução incluem os seguintes campos:

- `uuid`— O identificador universalmente exclusivo para a corrida.
- `workflowId`— Identificador do fluxo de trabalho associado a essa execução.
- `workflowName`— Nome do fluxo de trabalho associado a essa execução.
- `runId`— Identificador de execução.
- `runName`— Nome da execução.
- `runOutputUri`— O URI para onde a execução gravará seus dados de saída.

Exemplos de mensagens de eventos

O exemplo a seguir é um evento para uma alteração no status de execução, mostrando os campos adicionais.


```
{
    "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"
    }
}
```


O exemplo a seguir é um evento para uma alteração no status da tarefa.


```
{
    "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"
    }
}
```

Veja a seguir um exemplo de um evento para uma alteração de status do conjunto de leitura.

```
{
  "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"
  }
}
```

Um evento semelhante é criado para um trabalho de importação de lojas 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"
    }
}
```

O seguinte é um evento para uma mudança no status do trabalho de importação.

```
{
    "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"
    }
}
```

Solução de problemas

Os tópicos a seguir podem ajudá-lo a solucionar problemas encontrados ao usar HealthOmics fluxos de trabalho e armazenamentos de dados.

Tópicos

Solucionar problemas com fluxos de trabalho

Tópicos

Como soluciono uma falha na execução?

Use a operação GetRunda API para recuperar o motivo da falha. Para obter mais informações, consulte Motivos de falha na execução.

Como soluciono problemas de uma tarefa que falhou?

Revise o código de erro da mensagem de falha da tarefa para entender a falha. Revise os registros da tarefa CloudWatch para ver as mensagens de registro detalhadas da tarefa. Se você não estiver recebendo mensagens de registro detalhadas, poderá revisar seu fluxo de trabalho para gerar instruções de registro adicionais. Para obter mais informações, consulte Monitoramento HealthOmics com CloudWatch registros.

Onde posso encontrar os registros do motor para corridas concluídas com sucesso?

HealthOmics publica registros somente CloudWatch para execuções com falha. Se uma execução for concluída com sucesso, HealthOmics entrega os registros do mecanismo para o seu bucket do Amazon S3. Para obter mais informações, consulte Logs no Amazon S3.

Como posso reduzir o tamanho do parâmetro de entrada para um fluxo de trabalho?

Você pode especificar até 50 KB de parâmetros de entrada para um fluxo de trabalho. Você pode usar importações de diretórios ou folhas de amostra para permanecer dentro dessa restrição de tamanho. Para obter mais informações, consulte Gerenciando o tamanho dos parâmetros de execução.

Por que minha corrida não está sendo concluída?

Se houver problemas com seu código e os processos não tiverem sido encerrados corretamente, sua execução poderá deixar de responder ou ficar “travada”. Para obter mais informações sobre como evitar e capturar execuções que não respondem, consulte. Orientação para corridas sem resposta

Solução de problemas de cache de chamadas

Os tópicos a seguir podem ajudá-lo a solucionar problemas que você encontra com o cache de chamadas.

Tópicos

- Por que minha execução não está sendo salva no cache?
- Por que uma tarefa não está usando a entrada de cache?
- Por que o cache de chamadas de uma tarefa está desativado?

Por que minha execução não está sendo salva no cache?

1. Verifique se a execução está configurada para usar um cache verificando o campo cacheID na resposta da operação da GetRun API. Usando a CLI, execute este comando:. `aws omics get-run –id <run_id>`

2. Se a execução for bem-sucedida, verifique se o comportamento do cache retornado na GetRun resposta é CACHE_ALWAYS. Se o comportamento do cache for definido como CACHE_ON_FAILURE, as execuções só serão salvas no cache quando falharem.

Por que uma tarefa não está usando a entrada de cache?

<cache_id><cache_uuid>No grupo de `/aws/omics/WorkflowLog` CloudWatch registros, abra o fluxo de registros para o cache de execução: runCache//.

1. Verifique se uma execução anterior criou uma entrada de cache para a tarefa que você esperava que fosse armazenada em cache. As execuções salvas no cache serão gravadas com uma mensagem de log de CACHE_ENTRY_CREATED.
2. Localize o log CACHE_MISS da tarefa e execute-o concluído. Se não houver entrada de registro, verifique se a execução foi configurada para usar o cache.
3. Se uma entrada de cache foi criada, verifique se a CPUs memória GPUs e o resumo do contêiner são idênticos para ambas as tarefas. O ARN da tarefa que criou a entrada de cache está na mensagem de log.
4. Se os requisitos de computação para ambas as tarefas corresponderem, verifique se as entradas não foram alteradas entre as tarefas. Para fazer isso, abra os registros do motor. Se a execução tiver um status de FALHA, os registros estarão no Cloudwatch Log Group/. aws/omics/WorkflowLog Caso contrário, os registros do mecanismo podem ser encontrados no diretório de saída da execução.

Por que o cache de chamadas de uma tarefa está desativado?

Verifique se a tarefa está configurada para desativar o armazenamento em cache usando os recursos do mecanismo de fluxo de trabalho:

- Para fluxos de trabalho da WDL: verifique se a tarefa tem volátil definido como `true` na seção meta
- Para fluxos de trabalho do Nextflow: verifique se a tarefa tem a diretiva de cache definida como `false`
- Para fluxos de trabalho do CWL: verifique se a tarefa tem EnableReuse definido como para o recurso `false` WorkReuse

Solução de problemas de armazenamento de dados

Tópicos

Por que o S3 está GetObject falhando no meu conjunto de leitura?

Mais comumente, a falha ocorre devido à falta de uma permissão. A permissão de leitura do S3 do armazenamento de sequências é uma configuração bidirecional que exige que a política de acesso do S3 do armazenamento de sequências permita o acesso e que o diretor do IAM tenha uma política anexada permitindo o acesso. Para obter mais detalhes sobre os requisitos da política, consulte Permissões para acesso a dados usando o Amazon S3 URIs. Verifique se as seguintes configurações estão em vigor:

- A política de acesso S3 do armazenamento de sequências permitiu explicitamente o acesso ao principal do IAM ou à raiz da conta do principal.
- Verifique se o diretor do IAM tem uma política que fornece permissão explícita para o recurso que está sendo acessado. Observe que a política principal do IAM deve usar o ARN do ponto de acesso e não o caminho baseado no alias do ponto de acesso ao definir permissões e que o ARN está na condição e não é usado para especificar um recurso.
- Se sua loja usa uma chave gerenciada pelo cliente (CMK-KMS), certifique-se de que o diretor do IAM tenha permissões de kms: descriptografia na chave. Consulte o guia de acesso entre contas do KMS para configurar o uso entre contas.

Se você tiver uma política que usa controles de acesso baseados em tags, verifique o seguinte:

- Certifique-se de que o armazenamento de sequências tenha concluído a sincronização das tags. Para isso, o status da loja precisa ser active ou nãoupdating.
- Certifique-se de que não haja erros de digitação na chave da tag ou no valor da chave no conjunto de leitura e na política.

Por que não consigo ver minha loja de anotações ou loja de variantes no Athena?

Em Lake Formation, não se esqueça de criar um link de recurso com base na loja que foi compartilhada com você. Depois de criar um link de recurso que você tenha permissão para acessar, a loja deverá estar visível no Athena. Para obter mais informações, consulte Configurando o Lake Formation para usar HealthOmics.

Por que não consigo acessar meu armazenamento de dados no Athena?

Se seu armazenamento de anotações ou variantes estiver visível, mas você estiver recebendo uma mensagem de erro informando que o acesso foi negado, verifique qual versão do mecanismo de consulta você está usando. Somente consultas executadas usando a versão 3 do mecanismo são suportadas. Para ler mais sobre as versões do mecanismo de consulta do Athena, consulte a documentação do Amazon Athena.

Solução de problemas com o Amazon Q CLI

O Amazon Q CLI pode ajudar a simplificar seu processo de solução de problemas ao:

- Analisar execuções de fluxo de trabalho e depurar falhas de tarefas
- Coletando registros e mensagens de erro relevantes
- Criação de casos de AWS Support com todos os registros de depuração necessários anexados
- Retira informações de identificação pessoal (PII) das informações enviadas ao Support AWS

Para obter mais informações sobre como usar o Amazon Q CLI AWS HealthOmics para solucionar problemas e criar casos de suporte, consulte o tutorial de IA generativa da HealthOmics Agentic em. GitHub

> **Warning**
>
> Ao trabalhar com o Amazon Q CLI, revise todo o conteúdo gerado e as ações propostas antes de continuar. Forneça feedback para melhorar a qualidade da resposta e atender aos requisitos do seu fluxo de trabalho. Para obter mais informações, consulte Considerações de segurança e melhores práticas para o Amazon Q.

Cotas para AWS HealthOmics

AWS preenche sua conta com valores padrão para as HealthOmics cotas. Salvo indicação em contrário, cada valor de cota é o valor máximo por região.

> **Important**
>
> Você pode solicitar aumentos na maioria das cotas de serviço e de API. Consulte os tópicos a seguir para obter detalhes.

Tópicos

HealthOmics cotas de serviço

A tabela abaixo lista as cotas HealthOmics de serviço, junto com seus valores padrão. Para ver as cotas atuais de cada região, abra o console Service Quotas.

> **Important**
>
> Você pode solicitar um aumento para uma cota ajustável usando o console Service Quotas.

Para obter mais informações sobre cotas de serviço, consulte Solicitando um aumento de cota no Guia do usuário de cotas de serviço. Para uma cota que não está disponível no console Service Quotas, use o formulário de aumento de cota.

Nome	Padrão	Ajustá	Description
Analytics - Máximo de armazenamentos de anotações	Cada região com suporte: 10	Sim	O número máximo de lojas de anotações na região atual AWS

Nome	Padrão	Ajustá	Description
Analytics - Máximo de trabalhos simultâneos de importação de variantes ou de armazenamento de anotações	Cada região compatível: 5	Sim	O número máximo de trabalhos de importaçã o simultâneos na região atual AWS
Analytics - Máximo de arquivos por trabalho de importação do armazenam ento de variantes	Cada região com suporte: 1.000	Sim	O número máximo de arquivos por tarefa de importação de variantes na AWS região atual
Analytics - Máximo de compartil hamentos por armazenamento de anotações	Cada região com suporte: 10	Sim	O número máximo de compartilhamentos por armazenamento de anotações na região atual AWS
Analytics - Máximo de compartil hamentos por armazenamento de variantes	Cada região com suporte: 10	Sim	O número máximo de compartilhamentos por loja de variantes na AWS região atual
Analytics - Tamanho máximo de cada arquivo em um trabalho de importação de variantes	Cada região compatível: 20 gigabytes	Sim	O tamanho máximo de um arquivo em uma tarefa de importação de variantes na AWS região atual
Analytics - Tamanho máximo de cada arquivo em um trabalho de importação de anotações	Cada região compatível: 20 gigabytes	Sim	O tamanho máximo de um arquivo em uma tarefa de importação de anotações na região atual AWS
Analytics - Máximo de armazenamentos de variantes	Cada região com suporte: 10	Sim	O número máximo de lojas de variantes na AWS região atual

Nome	Padrão	Ajustá	Description
Analytics - Máximo de versões por armazenamento de anotações	Cada região com suporte: 10	Sim	O número máximo de versões por armazenamento de anotações na região atual AWS
Configurações - Configurações máximas	Cada região com suporte: 10	Sim	O número máximo de configurações na AWS região atual.
Armazenamento - Máximo de trabalhos simultâneos de ativação de conjuntos de leitura	Cada região com suporte: 25	Sim	O número máximo de trabalhos de ativação de conjuntos de leitura simultâneos na região atual AWS
Armazenamento - Máximo de trabalhos simultâneos de exportação de sequências e armazenamentos de referência	Cada região compatível: 5	Sim	O número máximo de trabalhos de exportação simultâneos de uma sequência ou armazenamento de referência na região atual AWS
Armazenamento - Máximo de trabalhos simultâneos de importação de sequências ou de armazenamento de referência	Cada região compatível: 5	Sim	O número máximo de trabalhos de importação simultâneos para uma sequência ou armazenamento de referência na região atual AWS
Armazenamento - Máximo de conjuntos de leitura por armazenamento de sequência	Cada região com suporte: 1.000.000	Sim	O número máximo de conjuntos de leitura em um armazenamento de sequência na AWS região atual

Nome	Padrão	Ajustá	Description
Armazenamento - Máximo de referênci as por armazenamento de referência	Cada região compatível: 50	Sim	O número máximo de referências em um repositório de referência na AWS região atual
Armazenamento - Máximo de armazenamentos de sequência	Cada região compatível: 20	Sim	O número máximo de sequências armazenadas na AWS região atual
Fluxos de trabalho - máximo ativo GPUs	Cada região compatível: 12	Sim	O número máximo de ativos simultâneos GPUs na AWS região atual. Em us-east-1 e us-west-2, as solicitações de aumento de cota para valores de até 500 são aprovadas automaticamente.
Fluxos de trabalho - Máximo de execuções ativas simultâneas usando o armazenamento dinâmico de execuções	Cada região compatível: 50	Sim	O número máximo de execuções ativas usando o armazenamento dinâmico de execuções na AWS região atual. As solicitações de aumento de cota para valores até 200 são aprovadas automaticamente.

Nome	Padrão	Ajustá	Description
Fluxos de trabalho - Máximo de execuções ativas simultâneas usando armazenamento estático de execuções	Cada região com suporte: 10	Sim	O número máximo de execuções ativas usando armazenamento de execução estática na AWS região atual. As solicitações de aumento de cota para valores até 50 são automaticamente aprovadas.
Fluxos de trabalho - Máximo de tarefas simultâneas por execução	Cada região com suporte: 25	Sim	O número máximo de tarefas simultâneas em cada execução na AWS região atual. Em us-east-1 e us-west-2, as solicitações de aumento de cota para valores de até 100 são aprovadas automaticamente.
Fluxos de trabalho - Duração máxima da execução	Cada região suportada: 604.800 segundos	Sim	A duração máxima da execução do fluxo de trabalho na AWS região atual.
Fluxos de trabalho - Máximo de execuções (ativas ou inativas)	Cada região compatível: 100.000	Sim	O número máximo de execuções (ativas ou inativas) na AWS região atual.
Fluxos de trabalho - Máximo de compartilhamentos por fluxo de trabalho	Cada região compatível: 100	Sim	O número máximo de compartilhamentos por fluxo de trabalho na AWS região atual

Nome	Padrão	Ajustá	Description
Fluxos de trabalho - Capacidad e máxima de armazenamento de execução estática por execução	Cada região suportada: 9.600	Sim	A capacidade máxima de armazenamento de execução estática em gibibytes (GiB) para cada execução na região atual. AWS Em us-east-1 e no us-west-2, as solicitações de aumento de cota para valores de até 50.000 são automaticamente aprovadas.
Fluxos de trabalho - Fluxos de trabalho máximos	Cada região com suporte: 1.000	Sim	O número máximo de fluxos de trabalho na AWS região atual.
Fluxos de trabalho - Transações por segundo (TPS) para a operação StartRun	Cada região compatível: 1	Sim	O máximo de transações por segundo (TPS) para a StartRun operação na AWS região atual.

HealthOmics cotas de tamanho fixo

Além doHealthOmics cotas de serviço, HealthOmics inclui cotas com tamanhos fixos. Você não pode solicitar um aumento para esses valores.

Salvo indicação em contrário, cada cota lista o valor máximo por região.

Tópicos

- HealthOmics cotas de tamanho fixo do analytics
- HealthOmics cotas de tamanho fixo de armazenamento
- HealthOmics cotas de tamanho fixo do fluxo de trabalho
- HealthOmics Cotas de tamanho fixo do fluxo de trabalho Ready2Run

HealthOmics cotas de tamanho fixo do analytics

A tabela a seguir mostra os valores máximos suportados para cotas de análise. Esses valores não são ajustáveis.

Name (Nome)	Description	Máximo	Ajustável Sim/Não
Analytics - Máximo de arquivos por tarefa de importação do armazenamento de anotações	O número máximo de arquivos por tarefa de importação de anotações.	1	Não

HealthOmics cotas de tamanho fixo de armazenamento

A tabela a seguir mostra os valores máximos suportados para arquivos de armazenamento. Esses valores não são ajustáveis.

Name (Nome)	Description	Máximo	Ajustável Sim/Não
Armazenamento - Tamanho máximo da política de recursos de acesso S3	O tamanho máximo da política de recursos de acesso do S3	15 KB	Não
Armazenamento - Máximo de tags de nível de conjunto propagadas	O número máximo de chaves de tag de nível definido, por loja, que se propagam ao objeto S3	5	Não
Armazenamento - Máximo de conjuntos de leitura por tarefa de ativação	O número máximo de conjuntos de leitura por tarefa de ativação.	20	Não

Name (Nome)	Description	Máximo	Ajustável Sim/Não
Armazenamento - Máximo de conjuntos de leitura por tarefa de exportação	O número máximo de conjuntos de leitura por tarefa de exportação.	100	Não
Armazenamento - Máximo de conjuntos de leitura por tarefa de importação	O número máximo de conjuntos de leitura por tarefa de importação.	100	Não
Armazenamento - Máximo de lojas de referência	O número máximo de lojas de referência.	1	Não
Armazenamento - Tamanho máximo da peça para um upload direto	O tamanho máximo da peça para upload direto para um armazenamento de sequências.	100 MB	Não
Armazenamento - Máximo de partes no arquivo para upload direto	O número máximo de partes em um arquivo para upload direto para um armazenam ento de sequências.	10.000	Não
Armazenamento - Tamanho máximo de referência	O tamanho máximo de um arquivo de referência que pode ser importado para um repositório de referência.	15 GB	Não

Name (Nome)	Description	Máximo	Ajustável Sim/Não
Armazenamento - Tamanho máximo da fonte do conjunto de leitura	O tamanho máximo de um único arquivo de origem em um conjunto de leitura que pode ser importado para um armazenamento de sequência.	976 GB	Não

HealthOmics cotas de tamanho fixo do fluxo de trabalho

A tabela a seguir mostra os valores máximos suportados para cotas de fluxo de trabalho. Esses valores não são ajustáveis.

Name (Nome)	Description	Tamanho máximo	Ajustável Sim/Não
Fluxos de trabalho - Máximo de grupos de execução	O número máximo de grupos de execução.	1000	Não
Fluxos de trabalho - Máximo de caches de execução	O número máximo de caches de execução que você pode criar para uma conta. Uma ou mais execuções podem compartilhar o mesmo cache de execução. Não há cota para o número de execuções que HealthOmics podem	1000	Não

Name (Nome)	Description	Tamanho máximo	Ajustável Sim/Não
	ser armazenadas em cache por conta.		
Fluxos de trabalho - versões máximas do fluxo de trabalho	O número máximo de versões do fluxo de trabalho por fluxo de trabalho.	1000	Não
Fluxos de trabalho - tamanho do contêiner da instância de CPU	O tamanho máximo da imagem do contêiner para uma instância de CPU.	45 GiB	Não
Fluxos de trabalho - tamanho do contêiner da instância de GPU	O tamanho máximo da imagem do contêiner para uma instância de GPU.	95 GiB	Não
Instância de GPU / dev/shm memória compartilhada	A quantidade máxima de memória compartilhada por instância de GPU.	8 GB por GPU	Não
Fluxos de trabalho - Executar arquivo de parâmetros	O tamanho máximo de um arquivo de parâmetros de execução.	50.000 bytes	Não

Name (Nome)	Description	Tamanho máximo	Ajustável Sim/Não
Fluxos de trabalho - Arquivo de modelo de parâmetros do fluxo de trabalho	O número máximo de entradas e o tamanho máximo do arquivo para um arquivo de modelo de parâmetro s de fluxo de trabalho. Essa cota se aplica aos fluxos de trabalho que você cria usando o console ou a API.	1.000 entradas, 400 KB	Não
Fluxos de trabalho - Tamanho do arquivo de definição do fluxo de trabalho - API	O tamanho máximo do arquivo de definição do fluxo de trabalho quando você cria o fluxo de trabalho usando a operação de API ou um AWS SDK.	100 MB	Não
Fluxos de trabalho - Tamanho do arquivo de definição do fluxo de trabalho - Console (upload direto)	O tamanho máximo do arquivo de definição do fluxo de trabalho que você pode fornecer como upload direto ao criar o fluxo de trabalho usando o console.	4,4 MB	Não

Name (Nome)	Description	Tamanho máximo	Ajustável Sim/Não
Fluxos de trabalho - Tamanho do arquivo de definição do fluxo de trabalho - Console (upload do Amazon S3)	O tamanho máximo do arquivo de definição de fluxo de trabalho que você pode fornecer como upload do Amazon S3 ao criar o fluxo de trabalho usando o console.	100 MB	Não
Fluxos de trabalho - Tamanho do repositório	O tamanho máximo de um repositório de código externo.	1 GiB	Não
Fluxos de trabalho - Tamanho de arquivo individual do repositório	O tamanho máximo de um arquivo individual de um repositório de código externo.	100 MiB	Não
Fluxos de trabalho - tamanho do arquivo README	O tamanho máximo de um arquivo README.	500 KIB	Não

Para obter sugestões sobre como reduzir o tamanho do arquivo de parâmetros de execução, consulteGerenciando o tamanho dos parâmetros de execução.

HealthOmics Cotas de tamanho fixo do fluxo de trabalho Ready2Run

Cada fluxo de trabalho do Ready2Run tem um tamanho máximo de arquivo de entrada. Na tabela a seguir, as unidades de tamanho do arquivo estão listadas em Gibibytes (GiB). Esses tamanhos máximos de arquivo não são ajustáveis.

Nome do fluxo de trabalho Ready2Run	Tamanho máximo do arquivo de entrada (GiB)	Ajustável (Sim/Não)
AlphaFold para 601-1200 resíduos	1	Não
AlphaFold para até 600 resíduos	1	Não
Bases2Fastq para 2x150	1000	Não
Bases2Fastq para 2x300	1000	Não
Bases2Fastq para 2x75	500	Não
ESMFold para até 800 resíduos	1	Não
GATK-BP fq2bam	64	Não
Linha germinativa GATK-BP bam2vcf para genoma 30x	39	Não
Linha germinativa GATK-BP fq2vcf para genoma 30x	64	Não
GATK-BP Somatic WES bam2vcf	86	Não
NVIDIA Parabricks BAM2 FQ2 BAM WGS para até 30X	80	Não
NVIDIA Parabricks BAM2 FQ2 BAM WGS para até 50X	120	Não
NVIDIA Parabricks BAM2 FQ2 BAM WGS para até 5X	20	Não
NVIDIA Parabricks FQ2 BAM WGS para até 30X	71	Não

Nome do fluxo de trabalho Ready2Run	Tamanho máximo do arquivo de entrada (GiB)	Ajustável (Sim/Não)
NVIDIA Parabricks FQ2 BAM WGS para até 50X	137	Não
NVIDIA Parabricks FQ2 BAM WGS para até 5X	13	Não
NVIDIA Parabricks DeepVariant Germline WGS para até 30X	71	Não
NVIDIA Parabricks DeepVariant Germline WGS para até 50X	137	Não
NVIDIA Parabricks DeepVariant Germline WGS para até 5X	12	Não
NVIDIA Parabricks Haplotype Caller Germline WGS para até 30X	71	Não
NVIDIA Parabricks Haplotype Caller Germline WGS para até 50X	137	Não
NVIDIA Parabricks Haplotype Caller Germline WGS para até 5X	13	Não
NVIDIA Parabricks Somatic Mutect2 WGS para até 50X	196	Não
sc RNAseq com Kallisto BUStools	119	Não

Nome do fluxo de trabalho Ready2Run	Tamanho máximo do arquivo de entrada (GiB)	Ajustável (Sim/Não)
1 colher de RNAseq chá com salmão frito	119	Não
sc RNAseq com STARsolo	119	Não
Sentieon Germline BAM WES para até 300x	9	Não
Sentieon Germline BAM WGS para até 32x	18	Não
Sentieon Germline FASTQ WES por até 100x	5	Não
Sentieon Germline FASTQ WES por até 300x	26	Não
Sentieon Germline FASTQ WGS por até 32x	51	Não
Sentieon LongRead para ONT	25	Não
Sentieon LongRead para PacBio HiFi	58	Não
Sentieon Somatic WES	50	Não
Sentieon Somatic WGS	113	Não
Ultima Genomics DeepVariant por até 40x	91	Não

HealthOmics Cotas de API

HealthOmics tem as seguintes cotas relacionadas às operações da API. Onde indicado, a cota é ajustável. Para solicitar um aumento, use o formulário de aumento de cota.

Para cada operação de API listada, a cota é o máximo de transações por segundo (TPS) para essa operação de API em cada região.

Tópicos

Cotas gerais de API

A tabela a seguir lista as operações gerais de API que se aplicam a mais de uma categoria (armazenamento, fluxos de trabalho e análises).

Operação de API	TPS máximo padrão	Ajustável (Sim/Não)
AcceptShare, CreateShare, DeleteShare, GetShare, ListShares	1 TPS	Sim

Cotas da API de armazenamento

A tabela a seguir lista as operações da API de armazenamento.

Operação da API de armazenamento	TPS máximo padrão	Ajustável (Sim/Não)
CreateSequenceStore, UpdateSequenceStore, DeleteSequenceStore, CreateReferenceStore, DeleteReferenceStore	1 TPS	Sim
BatchDeleteReadSet, DeleteReference	1 TPS	Sim

Operação da API de armazenamento	TPS máximo padrão	Ajustável (Sim/Não)
CreateMultipartReadSetUpload, CompleteMultipartReadSetUpload, AbortMultipartReadSetUpload	1 TPS	Não
Obtém S3AccessPolicy, coloca AccessPolicy S3, exclui S3 AccessPolicy	1 TPS	Sim
GetReference	10 TPS	Sim
UploadReadSetPart	10 TPS	Sim
GetReadSet	30 TPS	Sim
GetSequenceStore, ListSequenceStores	5 TPS	Sim
GetReadSetMetadata, ListReadSets	5 TPS	Sim
StartReadSetImportJob, GetReadSetImportJob, ListReadSetImportJobs	5 TPS	Sim
StartReadSetExportJob, GetReadSetExportJob, ListReadSetExportJobs	5 TPS	Sim
ListReferenceStores	5 TPS	Sim
StartReferencetImportJob, GetReferenceImportJob, ListReferenceImportJobs	5 TPS	Sim

Operação da API de armazenamento	TPS máximo padrão	Ajustável (Sim/Não)
ListReferences, GetReferenceMetadata	5 TPS	Sim
StartReadsetActivationJob	5 TPS	Sim
ListReadsetActivationJobs, GetReadSetActivationJob	5 TPS	Sim
ListMultipartReadSetUploads, ListReadSetUploadParts	5 TPS	Sim
TagResource, UntagResource, ListTagsForResource	5 TPS	Sim

Cotas da API de fluxo de trabalho

A tabela a seguir lista as operações da API do fluxo de trabalho.

Operação da API de fluxo de trabalho	TPS máximo padrão	Ajustável (Sim/Não)
StartRun	1 TPS	Sim
CreateWorkflow	5 TPS	Sim
CancelRun, DeleteRun, GetRun, GetRunTask, ListRunTasks, ListRuns	10 TPS	Sim
CreateRunGroup, DeleteRunGroup, GetRunGroup, ListRunGroups, UpdateRunGroup	10 TPS	Sim

Operação da API de fluxo de trabalho	TPS máximo padrão	Ajustável (Sim/Não)
CreateRunCache, UpdateRunCache, DeleteRunCache, GetRunCache, ListRunCaches	10 TPS	Sim
DeleteWorkflow, GetWorkflow, ListWorkflows, UpdateWorkflow	10 TPS	Sim

Cotas da API Analytics

A tabela a seguir lista as operações da API de análise.

Operação da API Analytics	TPS máximo padrão	Ajustável (Sim/Não)
CreateVariantStore, DeleteVariantStore, GetVariantStore, ListVariantStores, UpdateVariantStore	1 TPS	Não
StartVariantImportJob, CancelVariantImportJob, GetVariantImportJob, ListVariantImportJobs	1 TPS	Não
CreateAnnotationStore, DeleteAnnotationStore, GetAnnotationStore, ListAnnotationStores, UpdateAnnotationStore	1 TPS	Não
StartAnnotationImportJob, ListAnnotationImportJobs, GetAnnotationImportJob, CancelAnnotationImportJob	1 TPS	Não

Histórico de documentos para o Guia HealthOmics do usuário

A tabela a seguir descreve as versões de documentação do HealthOmics.

Alteração	Descrição	Data
AWS HealthOmics lojas de variantes e lojas de anotações não estão mais abertas a novos clientes.	AWS HealthOmics lojas de variantes e lojas de anotações não estão mais abertas a novos clientes. Para obter mais informações, consulte Alteração de disponibilidade da loja de AWS HealthOmics variantes e da loja de anotações.	7 de novembro de 2025
AWS HealthOmics lojas variantes e lojas de anotações não estarão mais abertas a novos clientes a partir de 7 de novembro de 2025.	AWS HealthOmics lojas variantes e lojas de anotações não estarão mais abertas a novos clientes a partir de 7 de novembro de 2025. Se você quiser usar lojas de variantes ou lojas de anotações, cadastre-se antes dessa data. Os clientes atuais podem continuar usando o serviço normalmente. Para obter mais informações, consulte Alteração de disponibilidade da loja de AWS HealthOmics variantes e da loja de anotações.	7 de outubro de 2025
Novos atributos	HealthOmics adicionou suporte para fluxos de	28 de agosto de 2025

	trabalho para sincronizar um repositório privado do Amazon ECR com um registro upstream. Para saber mais, consulte Imagens de contêiner para fluxos de trabalho privados em HealthOmics.	
Novos recursos de README e integração de repositórios	Foi adicionado suporte para criar fluxos de trabalho a partir de repositórios de código externos e arquivos README.	24 de julho de 2025
Novos atributos	HealthOmics adicionou suporte para interpolação automática de parâmetros Nextflow. Para saber mais, consulte Arquivos de modelo de parâmetros para HealthOmics fluxos de trabalho.	27 de junho de 2025
Novos atributos	HealthOmics adicionou suporte para fluxos de trabalho para interpolar os parâmetros de execução de um arquivo de definição de fluxo de trabalho WDL. Para saber mais, consulte Arquivos de modelo de parâmetros para HealthOmics fluxos de trabalho.	30 de maio de 2025

Novos atributos	HealthOmics adicionou suporte para controle de versão do fluxo de trabalho. Para saber mais, consulte Controle de versão do fluxo de trabalho em. HealthOmics	18 de abril de 2025
Novos atributos	HealthOmics adicionou taxa de transferência elástica para armazenamento dinâmico de execução. Para saber mais, consulte Executar tipos de armazenamento em HealthOmics.	16 de abril de 2025
Novos atributos	HealthOmics adicionou controles de acesso baseados em atributos para locais do Sequence Store S3 e a capacidade de sincronizar até cinco tags de conjunto de leitura com um objeto do Sequence Store S3. Para saber mais, consulte Criação de um armazenamento HealthOmics de sequências.	22 de novembro de 2024
Novos atributos	HealthOmics adicionou suporte para cache de chamadas, também conhecido como currículo, para fluxos de trabalho privados. Para saber mais, consulte Cache de chamadas.	20 de novembro de 2024

Novos atributos	HealthOmics adicionou novos campos de API para ajudá-lo a mapear entre trabalhos de entrada de armazenamento de sequências e conjuntos de leitura.	29 de agosto de 2024
Novos atributos	HealthOmics adicionou suporte para gerenciar versões do Nextflow. Para saber mais, consulte as versões do Nextflow.	14 de agosto de 2024
Novos atributos	HealthOmics adicionou suporte para fluxos de trabalho compartilhados e armazenamento dinâmico de execução.	30 de abril de 2024
Novos atributos	HealthOmics adicionou suporte para o acesso do Amazon S3 a lojas de referência e sequência, além de suporte para. SHA256 ETags	15 de abril de 2024
Novos atributos	HealthOmics adicionou tags de entidade (ETags) para armazenamentos de sequências.	6 de outubro de 2023
Novos atributos	HealthOmics adicionou controle de versão do armazenamento de anotações e compartilhamento do armazenamento analítico.	15 de agosto de 2023

Novos atributos	HealthOmics adicionou Common Workflow Language (CWL) como linguagem suportada para HealthOmics fluxos de trabalho.	30 de junho de 2023
Novos atributos	HealthOmics adicionou novos fluxos de trabalho Ready2Run , suporte de GPU para fluxos de trabalho, análise de dados para armazenamentos de anotações, upload direto para armazenamento e integração com o. HealthOmics EventBrid ge	15 de maio de 2023
Nova política gerenciada	HealthOmics adicionou uma nova política gerenciada que fornece acesso total. Para saber mais, consulte as políticas gerenciadas da AWS.	23 de fevereiro de 2023
Nova política gerenciada	HealthOmics adicionou uma nova política gerenciada que limita o acesso somente para leitura. Para saber mais, consulte as políticas gerenciad as da AWS.	29 de novembro de 2022
Lançamento inicial	Versão inicial do Guia HealthOmics do usuário	29 de novembro de 2022

As traduções são geradas por tradução automática. Em caso de conflito entre o conteúdo da tradução e da versão original em inglês, a versão em inglês prevalecerá.