aws

User Guide

AWS HealthOmics

Version latest

AWS HealthOmics: User Guide

Table of Contents

Was ist AWS HealthOmics?

AWS HealthOmics ist ein HIPAA-fähiger Service, der klinische Diagnosetests, Arzneimittelforschung und landwirtschaftliche Forschung beschleunigt, indem er die komplexe Infrastruktur hinter Ihren bioinformatischen Workflows vollständig verwaltet. HealthOmics unterstützt branchenübliche Workflow-Sprachen (WDL, Nextflow, CWL) und skaliert die Bioinformatik-Infrastruktur nahtlos, um Daten aus Zehntausenden von Tests pro Tag zu unterstützen — und das alles mit vorhersehbaren Kosten pro Probe. HealthOmics bewältigt die technischen Komplexitäten wie die Verwaltung von Rechenressourcen und die Wartung von Workflow-Engines, sodass Sie sich voll und ganz auf wissenschaftliche Durchbrüche konzentrieren können.

Topics

Wichtiger Hinweis

HealthOmics ist nur für die Übertragung, Speicherung, Formatierung oder Anzeige von Daten sowie für die Bereitstellung von Infrastruktur- und Konfigurationsunterstützung für die Verwaltung von Workflows vorgesehen. HealthOmics ist kein Ersatz für professionelle medizinische Beratung, Diagnose oder Behandlung und ist nicht dazu bestimmt, Krankheiten oder Gesundheitsprobleme zu heilen, zu behandeln, zu mildern, zu verhindern oder zu diagnostizieren. Sie sind dafür verantwortlich, im Rahmen der Verwendung von Produkten von Drittanbietern, die als Grundlage für klinische Entscheidungen dienen AWS HealthOmics, Untersuchungen am Menschen durchzuführen, auch in Verbindung mit Produkten von Drittanbietern.

HealthOmics features

Primäre Anwendungsfälle für: HealthOmics

- Klinische Diagnostik — Erstellen und skalieren Sie Workflows für diagnostische Tests mit vorhersehbaren Kosten und einer vollständig verwalteten Infrastruktur, die mit Ihrem Testvolumen wächst.
- Wirkstoffforschung — Beschleunigen Sie die therapeutische Forschung, indem Sie biologische Grundlagenmodelle in großem Maßstab orchestrieren und so eine schnelle Iteration mit Millionen potenzieller Kandidaten ermöglichen.
- Agrarforschung — Verbessern Sie Merkmale von Nutzpflanzen wie Trockenheitstoleranz und Schädlingsresistenz durch KI-gestützte Arbeitsabläufe, die die Ernährungssicherheit und die landwirtschaftliche Produktivität verbessern.

Die wichtigsten Vorteile von HealthOmics:

- Skalierbarkeit — Skalieren Sie Workflows auf mehr als 100.000 gleichzeitige Anwendungen, CPUs um Zehntausende von Tests täglich zu unterstützen — ganz ohne Infrastrukturmanagement und mit vorhersehbaren Kosten pro Probe.
- Konzentrieren Sie sich auf die Wissenschaft, nicht auf die Infrastruktur — Verwenden Sie vertraute Workflow-Sprachen und kümmern sich APIs dabei AWS automatisch um die Orchestrierung der Infrastruktur und das Datenmanagement hinter den Kulissen.
- Einhaltung der Vorschriften — Umfassende Prüfprotokolle, Nachverfolgung der Datenherkunft und HIPAA-konforme Infrastruktur, die für klinische Arbeitsabläufe konzipiert ist out-of-the-box — all das unterstützt die Entwicklung von Lösungen, die den gesetzlichen Anforderungen entsprechen.

HealthOmics besteht aus drei Hauptkomponenten:

- HealthOmics Workflows — Führen Sie bioinformatische Berechnungen auf einer automatisch bereitgestellten und skalierten Infrastruktur durch.
- HealthOmics Speicher — Speichern und teilen Sie Petabytes an Genomdaten effizient und zu niedrigen Kosten pro Gigabase.
- HealthOmics Analytik — Bereiten Sie Genomikdaten für multimodale und multimodale Analysen vor.

Verwenden Sie diese Komponenten unabhängig voneinander oder kombinieren Sie sie zu einer Lösung. end-to-end

HealthOmics Konzepte

Dieses Thema behandelt Definitionen für wichtige Konzepte und Begriffe, die spezifisch für dieses Handbuch sind HealthOmics, um Ihnen das Verständnis der in diesem Handbuch HealthOmics verwendeten Terminologie zu erleichtern.

Themen

Arbeitsabläufe

Mit HealthOmics Workflows können Sie Ihre Genomdaten verarbeiten und analysieren.

- Workflow — Die allgemeine Definition eines durchgängigen Prozesses, einschließlich Parametern und Verweisen auf Tools. Workflow-Definitionen können als WDL, Nextflow oder CWL ausgedrückt werden. Jeder erstellte Workflow hat eine eindeutige Kennung.
- Ausführen — Ein einziger Aufruf eines Workflows. Ein einzelner Lauf verwendet Ihre definierten Eingabedaten und erzeugt eine Ausgabe. Jeder erstellte Lauf hat eine eindeutige Kennung.
- Aufgabe — Die einzelnen Prozesse innerhalb eines Laufs. HealthOmics Workflows verwenden diese definierten Rechenspezifikationen, um Ihre Aufgabe auszuführen. Jede Aufgabe hat eine eindeutige Kennung.
- Ausführungsgruppe — Eine Gruppe von Läufen, für die Sie die maximale vCPU, die maximale Dauer oder die maximale Anzahl gleichzeitiger Läufe festlegen können, um die pro Lauf verwendeten Rechenressourcen zu begrenzen. Sie können Prioritäten für Ihre Läufe innerhalb einer Ausführungsgruppe angeben und konfigurieren. Sie können beispielsweise angeben, dass ein Lauf mit hoher Priorität vor einem Lauf mit niedrigerer Priorität ausgeführt wird, wodurch eine Prioritätswarteschlange entsteht. Es ist optional, eine Ausführungsgruppe zu verwenden, und jede Ausführungsgruppe hat eine eindeutige Kennung.

Speicher

Der Datenspeicher ist aufgeteilt in Sequenzspeicher für Ihre Genomsequenzen und zugehörige Informationen und einen Referenzspeicher für all Ihre Referenzgenome. Die folgenden Begriffe beschreiben die Implementierungen, die spezifisch für sind. HealthOmics

- Sequenzspeicher — Ein Datenspeicher für die Speicherung von Genomikdateien. Sie können einen oder mehrere Sequenzspeicher darin HealthOmics haben. Zugriffsberechtigungen und AWS KMS Verschlüsselung können für einen Sequenzspeicher festgelegt werden, um zu kontrollieren, wer Zugriff auf die Daten hat.
- Lesesatz — Ein Lesesatz ist eine Abstraktion von genomischen Lesevorgängen, die in den Formaten FASTQ, BAM oder CRAM gespeichert werden. Lesesätze können in Sequenzspeicher importiert und mit Metadaten annotiert werden. Mithilfe der attributebasierten Zugriffskontrolle (ABAC) können Sie Lesesätzen Berechtigungen zuweisen.
- Referenz — Eine Genomreferenz wird bei Lesevorgängen verwendet, um zu ermitteln, welcher Stelle in einem Genom ein bestimmter Lesevorgang oder eine Gruppe von Lesevorgängen zugeordnet ist. Diese sind im FASTA-Format und werden im Referenzspeicher gespeichert.
- Referenzspeicher — Ein Datenspeicher für die Speicherung von Referenzgenomen. Sie können in jedem Konto und jeder Region einen einzigen Referenzspeicher einrichten.

Analysen

Sie können Ihre Genomdaten mit HealthOmics Analytics transformieren und analysieren. Erstellen Sie einen Variantenspeicher oder einen Annotationsspeicher, um zusätzliche Informationen für Ihre Abfragen aufzunehmen.

- Variantenspeicher — Datenspeicher, der Variantendaten auf Populationsebene speichert. Variantenspeicher unterstützen sowohl GvCF (Genomic Variant Call Format) als auch VCF-Eingaben.
- Annotationsspeicher — Ein Datenspeicher, der eine Annotationsdatenbank darstellt, z. B. eine aus einer TSV/CSV-, VCF- oder General Feature Format () -Datei. GFF3 Annotationsspeicher werden während eines Imports demselben Koordinatensystem zugeordnet wie Variantenspeicher.

Zugehörige Services

Die folgenden Dienste funktionieren mit HealthOmics.

- Amazon Elastic Container Registry — Jeder private Workflow verwendet ein Amazon ECR-Image (in einem privaten Amazon ECR-Repository), um alle ausführbaren Dateien, Bibliotheken und Skripten zu enthalten, die für die Ausführung des Workflows erforderlich sind.
- Amazon Simple Storage Service — Amazon S3 bietet Dateispeicher für Store- und Workflow-Daten.

- AWS Lake Formation — Lake Formation verwaltet den Datenzugriff auf Ihre Analytics-Datenspeicher.
- Amazon Athena — Verwenden Sie Athena, um Abfragen in Ihren Variant-Shops durchzuführen.
- Amazon SageMaker AI — Verwenden Sie SageMaker KI, um HealthOmics Aufgaben mit Jupyter-Notebooks auszuführen.
- GitHub connections— Verwenden Sie Verbindungen, um Ihre externen Code-Repositorys mit Ihren Workflows zu verbinden. HealthOmics

Wie greife ich zu HealthOmics

Sie können über die Managementkonsole, CLI SDKs oder API auf AWS HealthOmics Funktionen zugreifen.

- AWS Managementkonsole — Stellt eine Weboberfläche bereit, über die Sie darauf zugreifen können HealthOmics.
- AWS Command Line Interface (AWS CLI) — Stellt Befehle für eine Vielzahl von AWS Diensten bereit, darunter Windows AWS HealthOmics, MacOS und Linux, und wird unter diesen unterstützt. Weitere Informationen zur Installation von finden Sie unter AWS Command Line Interface. AWS CLI
- AWS SDKs — AWS stellt SDKs (Software Development Kits) bereit, die aus Bibliotheken und Beispielcode für verschiedene Programmiersprachen und Plattformen (einschließlich Java, Python, Ruby, .NET, iOS und Android) bestehen. SDKs Sie bieten eine bequeme Möglichkeit, sie HealthOmics programmgesteuert zu verwenden. Weitere Informationen finden Sie im AWS SDK Developer Center.
- AWS API — Sie können API-Operationen verwenden, um HealthOmics programmgesteuert darauf zuzugreifen und diese zu verwalten. Weitere Informationen finden Sie in der HealthOmics -API-Referenz.

Regionen und Endpunkte für AWS HealthOmics

Eine vollständige Liste der Regionen und Endpunkte finden Sie in der AWS Allgemeinen Referenz.

Zusätzlich zu den AWS Regionen, die standardmäßig aktiv sind, gibt es auch Opt-in-Regionen, die aktiviert werden müssen. Weitere Informationen zur Aktivierung oder Deaktivierung einer Region

findest du im Leitfaden zur Kontoverwaltung unter „Geben Sie an, welche AWS Regionen Ihr AWS Konto verwenden kann“.

Weitere Informationen

In diesen Workshops und Tutorials erfährst du mehr darüber HealthOmics :

- HealthOmics Workshop — Workshop von HealthOmics Ende zu Ende
- AWS Ressourcen zur Genomik — Öffentliche Amazon ECR-Repositorien zum Thema Genomik
- Python-Tutorials — Jupyter-Notebook-Tutorials zu den GitHub Themen HealthOmics Speicher, Analysen und Workflows

Machen Sie sich mit zusätzlichen HealthOmics Tools vertraut, die Folgendes bieten: AWS

- WDL Linter — HealthOmics Linter für WDL
- Nextflow-Linter — Linter für Nextflow HealthOmics
- HealthOmics Amazon ECR-Hilfstool — Amazon ECR-Hilfstool für HealthOmics
- HealthOmics Tools on GitHub — Tools für die Arbeit mit HealthOmics (Transfermanager, URI-Parser, Omics-Wiederholung, Run-Analyzer).

AWS HealthOmics Änderung der Verfügbarkeit von Variantenspeicher und Annotationsspeicher

Nach reiflicher Überlegung haben wir beschlossen, AWS HealthOmics Variantengeschäfte und Annotations-Stores ab dem 7. November 2025 für Neukunden zu schließen. Vorhandene Kunden können den Service weiterhin wie gewohnt verwenden.

Im folgenden Abschnitt werden Migrationsoptionen beschrieben, die Ihnen helfen sollen, Ihre Variantengeschäfte und Analytics-Shops auf neue Lösungen umzustellen. Wenn Sie Fragen oder Bedenken haben, erstellen Sie unter support.console.aws.amazon.com eine Support-Anfrage.

Themen

Überblick über die Migrationsoptionen

Die folgenden Migrationsoptionen bieten eine Alternative zur Verwendung von Variantenspeichern und Annotationsspeichern:

1. Verwenden Sie die von HealthOmics -bereitgestellte Referenzimplementierung der ETL-Logik.

   Verwenden Sie S3-Tabellen-Buckets als Speicher und nutzen Sie weiterhin bestehende AWS Analysedienste.
2. Erstellen Sie eine Lösung mit einer Kombination vorhandener AWS Dienste.

   Für ETL können Sie benutzerdefinierte Glue-ETL-Jobs schreiben oder Open-Source-HAIL- oder GLOW-Code auf EMR verwenden, um Variantendaten zu transformieren.

   Verwenden Sie S3-Tabellen-Buckets als Speicher und nutzen Sie weiterhin bestehende Analysedienste AWS

3. Wählen Sie einen AWS Partner aus, der eine Variante und eine Alternative zum Speichern von Anmerkungen anbietet.

Migrationsoptionen für ETL-Logik

Beachten Sie die folgenden Migrationsoptionen für die ETL-Logik:

1. HealthOmics stellt die aktuelle ETL-Logik des Variantenspeichers als HealthOmics Referenzworkflow bereit. Sie können die Engine dieses Workflows verwenden, um genau den gleichen ETL-Prozess für Variantendaten wie den Variantenspeicher zu starten, jedoch mit voller Kontrolle über die ETL-Logik.

   Dieser Referenz-Workflow ist auf Anfrage erhältlich. Um Zugriff zu beantragen, erstellen Sie einen Support-Fall unter support.console.aws.amazon.com.
2. Um Variantendaten zu transformieren, können Sie benutzerdefinierte Glue-ETL-Jobs schreiben oder Open-Source-HAIL- oder GLOW-Code auf EMR verwenden.

Migrationsoptionen für Speicher

Als Ersatz für einen vom Service gehosteten Datenspeicher können Sie Amazon S3 S3-Tabellen-Buckets verwenden, um ein benutzerdefiniertes Tabellenschema zu definieren. Weitere Informationen zu Tabellen-Buckets finden Sie unter Tabellen-Buckets im Amazon S3 S3-Benutzerhandbuch.

Sie können Tabellen-Buckets für vollständig verwaltete Iceberg-Tabellen in Amazon S3 verwenden.

Sie können eine Support-Anfrage stellen und das HealthOmics Team bitten, die Daten aus Ihrem Varianten- oder Annotationsspeicher in den von Ihnen konfigurierten Amazon S3 S3-Tabellen-Bucket zu migrieren.

Nachdem Ihre Daten in den Amazon S3 S3-Tabellen-Bucket gefüllt wurden, können Sie Ihre Variantenspeicher und Annotationsspeicher löschen. Weitere Informationen finden Sie unter HealthOmics Analytics-Stores löschen.

Analysen

Verwenden Sie für Datenanalysen weiterhin AWS Analysedienste wie Amazon Athena, Amazon EMR, Amazon Redshift oder Amazon Quick Suite.

AWS Partner

Sie können mit einem AWS Partner zusammenarbeiten, der anpassbares ETL, Tabellenschemas, integrierte Abfrage- und Analysetools sowie Benutzeroberflächen für die Interaktion mit Daten bereitstellt.

Beispiele

Die folgenden Beispiele zeigen, wie Sie Tabellen erstellen, die sich zum Speichern von VCF- und GVCF-Daten eignen.

Athena DDL

Sie können das folgende DDL-Beispiel in Athena verwenden, um eine Tabelle zu erstellen, die zum Speichern von VCF- und GVCF-Daten in einer einzigen Tabelle geeignet ist. Dieses Beispiel entspricht nicht exakt der Variantenspeicherstruktur, eignet sich aber gut für einen generischen Anwendungsfall.

Erstellen Sie Ihre eigenen Werte für DATABASE_NAME und TABLE_NAME, wenn Sie die Tabelle erstellen.

```
 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'
);
```

Erstellen Sie Tabellen mit Python (ohne Athena)

Das folgende Python-Codebeispiel zeigt, wie die Tabellen ohne Verwendung von Athena erstellt werden.

```
 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)
```

Einrichten HealthOmics

Registrieren Sie sich zur Einrichtung für einen AWS-Konto, erstellen Sie einen Administratorbenutzer und verwalten Sie den Zugriff für weitere Benutzer auf sichere Weise. AWS HealthOmics

Themen

Melden Sie sich für eine an AWS-Konto

Wenn Sie noch keine haben AWS-Konto, führen Sie die folgenden Schritte aus, um eine zu erstellen.

Um sich für eine anzumelden AWS-Konto

1. Öffnen Sie https://portal.aws.amazon.com/billing/die Anmeldung.
2. Folgen Sie den Online-Anweisungen.

   Während der Anmeldung erhalten Sie einen Telefonanruf oder eine Textnachricht und müssen einen Verifizierungscode über die Telefontasten eingeben.

   Wenn Sie sich für eine anmelden AWS-Konto, Root-Benutzer des AWS-Kontoswird eine erstellt. Der Root-Benutzer hat Zugriff auf alle AWS-Services und Ressourcen des Kontos. Als bewährte Sicherheitsmethode weisen Sie einem Administratorbenutzer Administratorzugriff zu und verwenden Sie nur den Root-Benutzer, um Aufgaben auszuführen, die Root-Benutzerzugriff erfordern.

AWS sendet Ihnen nach Abschluss des Anmeldevorgangs eine Bestätigungs-E-Mail. Sie können Ihre aktuellen Kontoaktivitäten jederzeit einsehen und Ihr Konto verwalten, indem Sie zu https://aws.amazon.com/gehen und Mein Konto auswählen.

Erstellen eines Benutzers mit Administratorzugriff

Nachdem Sie sich für einen angemeldet haben AWS-Konto, sichern Sie Ihren Root-Benutzer des AWS-Kontos AWS IAM Identity Center, aktivieren und erstellen Sie einen Administratorbenutzer, sodass Sie den Root-Benutzer nicht für alltägliche Aufgaben verwenden.

Sichern Sie Ihre Root-Benutzer des AWS-Kontos

1. Melden Sie sich AWS-Managementkonsolealls Kontoinhaber an, indem Sie Root-Benutzer auswählen und Ihre AWS-Konto E-Mail-Adresse eingeben. Geben Sie auf der nächsten Seite Ihr Passwort ein.

   Hilfe bei der Anmeldung mit dem Root-Benutzer finden Sie unter Anmelden als Root-Benutzer im AWS-Anmeldung Benutzerhandbuch zu.

2. Aktivieren Sie die Multi-Faktor-Authentifizierung (MFA) für den Root-Benutzer.

   Anweisungen finden Sie unter Aktivieren eines virtuellen MFA-Geräts für Ihren AWS-Konto Root-Benutzer (Konsole) im IAM-Benutzerhandbuch.

Erstellen eines Benutzers mit Administratorzugriff

1. Aktivieren Sie das IAM Identity Center.

   Anweisungen finden Sie unter Aktivieren AWS IAM Identity Center im AWS IAM Identity Center Benutzerhandbuch.

2. Gewähren Sie einem Administratorbenutzer im IAM Identity Center Benutzerzugriff.

   Ein Tutorial zur Verwendung von IAM-Identity-Center-Verzeichnis als Identitätsquelle finden Sie IAM-Identity-Center-Verzeichnis im Benutzerhandbuch unter Benutzerzugriff mit der Standardeinstellung konfigurieren.AWS IAM Identity Center

Anmelden als Administratorbenutzer

- Um sich mit Ihrem IAM-Identity-Center-Benutzer anzumelden, verwenden Sie die Anmelde-URL, die an Ihre E-Mail-Adresse gesendet wurde, als Sie den IAM-Identity-Center-Benutzer erstellt haben.

Hilfe bei der Anmeldung mit einem IAM Identity Center-Benutzer finden Sie im AWS-Anmeldung Benutzerhandbuch unter Anmeldung beim AWS Access-Portal.

Weiteren Benutzern Zugriff zuweisen

1. Erstellen Sie im IAM-Identity-Center einen Berechtigungssatz, der den bewährten Vorgehensweisen für die Anwendung von geringsten Berechtigungen folgt.

   Anweisungen hierzu finden Sie unter Berechtigungssatz erstellen im AWS IAM Identity Center Benutzerhandbuch.
2. Weisen Sie Benutzer einer Gruppe zu und weisen Sie der Gruppe dann Single Sign-On-Zugriff zu.

   Eine genaue Anleitung finden Sie unter Gruppen hinzufügen im AWS IAM Identity Center Benutzerhandbuch.

Erstellen Sie IAM-Berechtigungen für HealthOmics

Um sie zu verwenden HealthOmics, konfigurieren Sie die folgenden IAM-Berechtigungen:

- Identitätsbasierte IAM-Richtlinien, auf die Benutzer in Ihrem Konto zugreifen können. HealthOmics
- Eine IAM-Servicerolle für den Zugriff HealthOmics auf Ressourcen in Ihrem Namen.
- Berechtigungen in anderen Diensten (wie Lake Formation und Amazon ECR) für Ihre Benutzer und den HealthOmics Service für den Zugriff auf Ressourcen.

Weitere Informationen zur Konfiguration von IAM-Berechtigungen für finden Sie HealthOmics unter. IAM-Berechtigungen für HealthOmics

Connect mit externen Code-Repositorys her

Mit AWS HealthOmics können Sie Ihre Workflows mithilfe von Git-basierten Repositorys verwalten. AWS CodeConnections HealthOmics verwendet diese Verbindung, um auf Ihre Quellcode-Repositorys zuzugreifen.

Bevor Sie mit externen Code-Repositorys arbeiten, folgen Sie zunächst der Anleitung zum Einrichten von Verbindungen. AWS CodeConnections Stellen Sie sicher, dass Sie die richtigen IAM-Richtlinien

und -Berechtigungen für Ihr AWS Konto erstellt haben. Eine Liste der unterstützten Git-Anbieter und weitere Informationen findest du unter Für welche Drittanbieter kann ich Verbindungen erstellen? .

Eine Verbindung herstellen

Um eine Verbindung mit Ihrem bevorzugten Repository-Anbieter herzustellen, folgen Sie der Anleitung Verbindung erstellen.

Verwenden von Amazon Q CLI mit HealthOmics

Amazon Q CLI ermöglicht Interaktionen in natürlicher Sprache mit AWS HealthOmics, sodass Sie komplexe genomische Workflows und Analyseaufgaben mithilfe von Konversationsbefehlen ausführen können. Um Amazon Q CLI zu verwenden, müssen Sie IAM-Berechtigungen für HealthOmics und andere Dienste (wie CloudWatch Amazon ECR oder Amazon S3) konfigurieren, damit Amazon Q auf deren Ressourcen zugreifen kann.

Das HealthOmics Generative KI-Tutorial von Agentic bietet step-by-step Anleitungen zur Konfiguration von Kontextdateien und zur Aktivierung von Amazon Q CLI zur Erstellung, Ausführung und Optimierung Ihrer AWS HealthOmics Workflows.

Erste Schritte mit HealthOmics

Stellen Sie zunächst sicher HealthOmics, dass Sie Ihre IAM-Berechtigungen und -Rollen für HealthOmics ordnungsgemäß eingerichtet haben.

Verwenden eines Ready2Run-Workflows in der Konsole HealthOmics

Die folgende Übung zeigt, wie Sie einen Ready2Run-Workflow verwenden. Ein Ready2Run-Workflow ist mit den Parametern und Toolreferenzen vorkonfiguriert, die Sie zur Ausführung des Workflows benötigen. Der Workflow-Herausgeber stellt Beispieldaten bereit, sodass Sie keine eigenen Daten erstellen müssen.

1. Öffnen Sie die HealthOmics -Konsole.
2. Wählen Sie oben links den Navigationsbereich (**>**) und wählen Sie Ready2Run-Workflows aus.
3. Wählen Sie auf der Seite Ready2Run-Workflows den Workflow aus. ESMFold for up to 800 residues Die Konsole öffnet die Detailseite für diesen Workflow.
4. Die Registerkarte „Details“ enthält Informationen zum Workflow. Um den Workflow auszuprobieren, wählen Sie oben rechts auf der Seite Start run aus.
5. Geben Sie auf der Seite „Ausführungsdetails angeben“ einen Namen für die Ausführung ein.
6. Geben Sie einen Amazon S3 S3-Speicherort für die Run-Ausgabe ein oder wählen Sie ihn aus.
7. Wählen Sie für den Aufbewahrungsmodus für Run-Metadaten aus, ob Runmeta-Daten beibehalten oder entfernt werden sollen.
8. Wählen Sie im Bereich „Servicerolle“ die Option Neue Servicerolle erstellen und verwenden aus.
9. Wählen Sie Weiter aus.
10. Wählen Sie auf der Seite Parameterwerte hinzufügen die Option Workflow mit Ready2Run-Testdaten ausführen aus.
11. Wählen Sie Weiter aus.
12. Überprüfen Sie Ihre Eingaben und wählen Sie dann Start run aus.

Beispielaufforderungen für Amazon Q CLI

Amazon Q CLI kann genomische Workflows und Analyseaufgaben AWS HealthOmics mithilfe von Befehlen in natürlicher Sprache ausführen. Die folgenden Beispielaufforderungen ermöglichen es Ihnen, Workflows zu erstellen, Läufe zu verwalten und genomische Daten zu analysieren. Weitere Informationen und Beispielaufforderungen dazu finden Sie im Agentic HealthOmics Generative HealthOmics AI-Tutorial unter. GitHub

- „Erstellen Sie eine WDL 1.1-Workflow-Datei, auf der `main.wdl` diese ausgeführt werden soll. HealthOmics Der Workflow verwendet ein Referenzgenom als Eingabe und Paare von Fastq-Dateien. Es indexiert das Referenzgenom mithilfe von BWA und ordnet dann jedes Paar von Fastq-Dateien der Referenz zu. Schließlich werden alle gemappten BAMs zu einer einzigen BAM-Datei zusammengeführt und diese Datei und ihr BAI-Index ausgegeben.“
- „Den Workflow verpacken und erstellen in HealthOmics“
- „Aktualisieren Sie die Datei inputs.json, um echte Dateien aus meinem Amazon S3 S3-Bucket zu verwenden`omics-my-bucket-with-genome-data`" (Geben Sie einen bestimmten Amazon S3 S3-Bucket-Speicherort an, oder lassen Sie Amazon Q die Suche durchführen)
- „Finden Sie geeignete Container in meinen Amazon ECR-Repositorys und aktualisieren Sie inputs.json, um diese zu verwenden“
- „Suchen oder erstellen Sie eine geeignete IAM-Rolle, die Sie bei der Ausführung des Workflows verwenden können“
- „Einen Run-Cache für meinen Workflow erstellen“
- „Führen Sie den Workflow aus in HealthOmics“
- „Überprüfen Sie den Status des Laufs“

> **Warning**
>
> Wenn Sie mit Amazon Q CLI arbeiten, überprüfen Sie alle generierten Inhalte und vorgeschlagenen Maßnahmen, bevor Sie fortfahren. Geben Sie Feedback, um die Antwortqualität zu verbessern und die Anforderungen Ihres Workflows zu erfüllen. Weitere Informationen finden Sie unter Sicherheitsüberlegungen und bewährte Methoden für Amazon Q.

Private Workflows in HealthOmics

Verwenden Sie private Workflows, wenn Sie Ihre eigene Workflow-Definition erstellen möchten. Die Workflow-Definition spezifiziert Informationen über den Workflow und definiert die Workflow-Aufgaben. Ein Lauf ist ein einzelner Aufruf eines Workflows, und eine Aufgabe ist ein einzelner Prozess innerhalb des Laufs.

HealthOmics unterstützt Workflow-Definitionen, die Sie in Workflow Description Language (WDL), Common Workflow Language (CWL) oder Nextflow erstellen.

HealthOmics Workflows bieten die folgenden optionalen Funktionen:

- Run groups— Sie können private Workflows zu einer Ausführungsgruppe hinzufügen, um die Computenutzung zu kontrollieren. Eine Ausführungsgruppe ist eine Sammlung von Workflow-Ausführungen, die sich eine Reihe von Ressourcenlimits teilen, z. B. die maximale Anzahl gleichzeitiger Läufe und die maximale Ausführungsdauer. Sie legen diese Grenzwerte fest, um die Rechenressourcen zu steuern, die die Ausführungsgruppe verbraucht.
- Call caching— Sie können einen Aufruf-Cache verwenden, um Aufgabenausgaben zu speichern und wiederzuverwenden, was zu kürzeren Ausführungsdauern und Rechenkosten spart.
- Sharing workflows— Sie können Ihre privaten Workflows mit anderen AWS-Konten in derselben Region teilen.
- Workflow versions— Sie können Versionen eines privaten Workflows erstellen. Die Workflow-Versionierung bietet Benutzern die Möglichkeit, zu wählen, wann sie mit der Nutzung aktualisierter Funktionen beginnen möchten. Workflow-Versionen sind unveränderlich und bieten den gleichen Grad an Datenherkunft wie Workflows.

Informationen zur Konfiguration von IAM-Berechtigungen für Workflows finden Sie unter. IAM-Berechtigungen für HealthOmics

Vollständige Beispiele für die Verwendung HealthOmics privater Workflows finden Sie in den HealthOmics Github-Tutorials oder im umfassenden AWS-Workshop-Tutorial für HealthOmics.

Themen

- HealthOmics Run-Gruppen verwenden
- Aufruf-Caching für Läufe HealthOmics
- HealthOmics Workflows teilen

Private Workflows erstellen in HealthOmics

Private Workflows hängen von einer Vielzahl von Ressourcen ab, die Sie vor der Erstellung des Workflows erstellen und konfigurieren:

- Workflow definition file:Eine inWDL, Nextflow oder geschriebene Workflow-DefinitionsdateiCWL. Die Workflow-Definition spezifiziert die Eingaben und Ausgaben für Läufe, die den Workflow verwenden. Sie enthält auch Spezifikationen für die Ausführungen und Ausführungsaufgaben für Ihren Workflow, einschließlich der Rechen- und Speicheranforderungen. Die Workflow-Definitionsdatei muss `.zip` das Format haben. Weitere Informationen finden Sie unter Workflow-Definitionsdateien.
  - Sie können Amazon Q CLI verwenden, um Ihre Workflow-Definitionsdateien in WDL, Nextflow und CWL zu erstellen und zu validieren. Weitere Informationen finden Sie unter Beispielaufforderungen für Amazon Q CLI und im HealthOmics Agentic Generative AI-Tutorial unter. GitHub
- (Optional) Parameter template file:Eine Parameter-Vorlagendatei, die in geschrieben wurde. JSON Erstellen Sie die Datei, um die Ausführungsparameter zu definieren, oder HealthOmics generiert die Parametervorlage für Sie. Weitere Informationen finden Sie unter Parametervorlagendateien für HealthOmics Workflows.
- Amazon ECR container images:Erstellen Sie ein privates Amazon ECR-Repository für den Workflow. Erstellen Sie Container-Images im privaten Repository oder synchronisieren Sie den Inhalt einer unterstützten Upstream-Registrierung mit Ihrem privaten Amazon ECR-Repository.
- (Optional) Sentieon licenses:Fordern Sie eine Sentieon Lizenz an, um die Sentieon Software in privaten Workflows zu verwenden.

Optional können Sie die Workflow-Definition vor oder nach der Erstellung des Workflows überprüfen. Das linter Thema beschreibt die verfügbaren Linter in. HealthOmics

Themen

- HealthOmics Workflow-Integration mit Git-basierten Repositorys
- Workflow-Definitionsdateien in HealthOmics

- Parametervorlagendateien für HealthOmics Workflows
- Container-Images für private Workflows
- HealthOmics Workflow-README-Dateien
- Sentieon-Lizenzen für private Workflows anfordern
- Der Arbeitsablauf blinkt ein HealthOmics
- HealthOmics Workflow-Operationen

HealthOmics Workflow-Integration mit Git-basierten Repositorys

Wenn Sie einen Workflow (oder eine Workflow-Version) erstellen, geben Sie eine Workflow-Definition an, um Informationen über den Workflow, die Ausführungen und Aufgaben anzugeben. HealthOmics kann die Workflow-Definition als ZIP-Archiv (lokal oder in einem Amazon S3 S3-Bucket gespeichert) oder aus einem unterstützten Git-basierten Repository abrufen.

Die HealthOmics Integration mit Git-basierten Repositorys ermöglicht die folgenden Funktionen:

- Direkte Workflow-Erstellung aus öffentlichen, privaten und selbstverwalteten Instanzen.
- Integration von Workflow-README-Dateien und Parametervorlagen aus Repositorys.
- Support für GitHub GitLab, und Bitbucket-Repositorys.

Durch die Verwendung eines Git-basierten Repositorys vermeiden Sie die manuellen Schritte des Herunterladens von Workflow-Definitionsdateien und Eingabeparameter-Vorlagendateien, der Erstellung eines ZIP-Archivs und der anschließenden Bereitstellung des Archivs in S3. Dies vereinfacht die Workflow-Erstellung für Szenarien wie die folgenden Beispiele:

1. Sie möchten schnell mit einem gängigen Open-Source-Workflow wie nf-core loslegen. HealthOmicsruft automatisch alle Workflow-Definitions- und Eingabeparameter-Vorlagendateien aus dem NF-Core-Repository ab GitHub und verwendet diese Dateien, um Ihren neuen Workflow zu erstellen.
2. Sie verwenden einen öffentlichen Workflow von GitHub, und einige neue Updates sind verfügbar. Sie können ganz einfach eine neue HealthOmics Workflow-Version erstellen, indem Sie die aktualisierte Workflow-Definition GitHub als Quelle verwenden. Benutzer Ihres Workflows können zwischen dem ursprünglichen Workflow und der neuen Workflow-Version wählen, die Sie erstellt haben.

3. Ihr Team baut eine eigene Pipeline auf, die nicht öffentlich ist. Sie speichern Ihren Code in einem privaten Git-Repository und verwenden diese Workflow-Definition für Ihre HealthOmics Workflows. Das Team aktualisiert die Workflow-Definition häufig als Teil eines iterativen Workflow-Entwicklungszyklus. Sie können ganz einfach neue Workflow-Versionen nach Bedarf aus Ihrem privaten Repository erstellen.

Themen

Unterstützte Git-basierte Repositorys

HealthOmics unterstützt öffentliche und private Repositorys für die folgenden Git-basierten Anbieter:

- GitHub
- GitLab
- Bitbucket

HealthOmics unterstützt selbstverwaltete Repositorys für die folgenden Git-basierten Anbieter:

- GitHubEnterpriseServer
- GitLabSelfManaged

HealthOmics unterstützt die Verwendung von kontoübergreifenden Verbindungen für GitHub, GitLab und Bitbucket. Richten Sie gemeinsame Berechtigungen über den AWS Resource Access Manager ein. Ein Beispiel finden Sie im CodePipeline Benutzerhandbuch unter Gemeinsame Verbindungen.

Konfigurieren Sie Verbindungen zu externen Code-Repositorys

Connect Ihre Workflows mithilfe von AWS mit Git-basierten Repositorys. CodeConnection HealthOmics verwendet diese Verbindung, um auf Ihre Quellcode-Repositorys zuzugreifen.

> **Note**
>
> Der CodeConnections AWS-Service ist in der Region il-central-1 nicht verfügbar. Konfigurieren Sie für diese Region den Dienst us-east-1, um Workflows oder Workflow-Versionen aus einem Repository zu erstellen.

Eine Verbindung erstellen

Bevor Sie Verbindungen herstellen können, folgen Sie den Anweisungen unter Verbindungen einrichten im Developer Console Tools-Benutzerhandbuch.

Um eine Verbindung herzustellen, folgen Sie den Anweisungen unter Verbindung erstellen im Developer Console Tools-Benutzerhandbuch.

Konfigurieren Sie die Autorisierung für die Verbindung

Sie müssen die Verbindung mithilfe des OAuth Flow des Anbieters autorisieren. Stellen Sie sicher, dass der Verbindungsstatus lautet, `AVAILABLE` bevor Sie ihn verwenden.

Beispiele finden Sie im Blogbeitrag How To Create an AWS HealthOmics Workflows from Content in Git.

Zugreifen auf selbstverwaltete Repositorys

Um Verbindungen zu einem GitLab selbstverwalteten Repository einzurichten, verwenden Sie beim Erstellen eines Hosts ein persönliches Administrator-Zugriffstoken. Bei der darauffolgenden Verbindungsherstellung wird mit dem Konto des Kunden auf Oauth zugegriffen.

Das folgende Beispiel richtet eine Verbindung zu einem selbstverwalteten Repository ein GitLab :

1. Richten Sie den Zugriff auf das Personal Access Token eines Admin-Benutzers ein.

   Informationen zum Einrichten eines PAT in einem GitLab selbstverwalteten Repository finden Sie unter Persönliche Zugriffstoken in GitLab Docs.
2. Erstellen eines Hosts

   a. Navigieren Sie zu CodePipeline>Einstellungen>Verbindungen.
   b. Wählen Sie die Registerkarte Hosts und dann Create Host.
   c. Konfigurieren Sie die folgenden Felder:

- Geben Sie einen Namen des Hosts ein
- Wählen Sie als Anbietertyp GitLab Self Managed
- Geben Sie die Host-URL ein
- Geben Sie die VPC-Informationen ein, wenn der Host in einer VPC definiert ist

d. Wählen Sie Create Host, wodurch der Host im Status PENDING erstellt wird.

e. Um die Einrichtung abzuschließen, wählen Sie Host einrichten.

f. Geben Sie das Personal Access Token (PAT) eines Admin-Benutzers ein und wählen Sie dann Weiter.

3. Stellen Sie die Verbindung her

a. Wählen Sie auf der Registerkarte Verbindungen die Option Verbindungen erstellen aus.

b. Wählen Sie als Anbietertyp die Option GitLab Selbstverwaltet aus.

c. Geben Sie unter Verbindungseinstellungen > Verbindungsnamen eingeben die Host-URL ein, die Sie zuvor erstellt haben.

d. Wenn auf Ihre GitLab selbstverwaltete Instanz nur über eine VPC zugegriffen werden kann, konfigurieren Sie die VPC-Details.

e. Wählen Sie „Ausstehende Verbindung aktualisieren“. Das modale Fenster leitet Sie zur GitLab Anmeldeseite weiter.

f. Geben Sie den Benutzernamen und das Passwort für das Kundenkonto ein und schließen Sie den Autorisierungsprozess ab.

g. Wählen Sie für die erstmalige Einrichtung Authorize AWS Connector for Gitlab Self Managed.

Kontingente im Zusammenhang mit externen Code-Repositorys

Für die HealthOmics Integration mit externen Code-Repositorys gibt es eine maximale Größe für ein Repository, jede Repository-Datei und jede README-Datei. Details hierzu finden Sie unter HealthOmics Workflow-Kontingente mit fester Größe.

Erforderliche IAM-Berechtigungen

Fügen Sie Ihrer identitätsbasierten IAM-Richtlinie die folgenden Aktionen hinzu:

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

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

Workflow-Definitionsdateien in HealthOmics

Sie verwenden eine Workflow-Definition, um Informationen über den Workflow, die Läufe und die Aufgaben in den Läufen anzugeben. Sie erstellen Workflow-Definitionen in einer oder mehreren Dateien mithilfe einer Workflow-Definitionssprache. HealthOmics unterstützt in WDL, Nextflow oder CWL geschriebene Workflow-Definitionen.

HealthOmics unterstützt die folgenden Optionen für WDL-Workflow-Definitionen:

- WDL — Stellt eine spezifikationskonforme WDL-Engine bereit.
- WDL lenient — Konzipiert für Workflows, die von Cromwell migriert wurden. Es unterstützt Kundenrichtlinien von Cromwell und einige nichtkonforme Logiken. Details hierzu finden Sie unter Implizite Typkonvertierung in WDL Lenient.

Informationen zu den einzelnen Workflow-Sprachen finden Sie in den sprachspezifischen Abschnitten weiter unten.

In der Workflow-Definition geben Sie die folgenden Arten von Informationen an:

- Language version— Die Sprache und Version der Workflow-Definition.
- Compute and memory— Die Rechen- und Speicheranforderungen für Aufgaben im Workflow.
- Inputs— Speicherort der Eingaben für die Workflow-Aufgaben. Weitere Informationen finden Sie unter HealthOmics Eingaben ausführen.
- Outputs— Speicherort für die von den Aufgaben generierten Ausgaben.
- Task resources— Rechen- und Speicheranforderungen für jede Aufgabe.
- Accelerators— andere Ressourcen, die für die Aufgaben erforderlich sind, z. B. Beschleuniger.

Themen

- Aufgabenressourcen in einer HealthOmics Workflow-Definition
- Aufgabenbeschleuniger in einer Workflow-Definition HealthOmics
- Besonderheiten der WDL-Workflow-Definition
- Einzelheiten der Nextflow-Workflow-Definition
- Besonderheiten der CWL-Workflow-Definition
- Beispiele für Workflow-Definitionen

HealthOmics Anforderungen an die Workflow-Definition

Die HealthOmics Workflow-Definitionsdateien müssen die folgenden Anforderungen erfüllen:

- Aufgaben müssen input/output Parameter, Amazon ECR-Container-Repositorys und Laufzeitspezifikationen wie Speicher- oder CPU-Zuweisung definieren.
- Stellen Sie sicher, dass Ihre IAM-Rollen über die erforderlichen Berechtigungen verfügen.
  - Ihr Workflow hat Zugriff auf Eingabedaten aus AWS Ressourcen wie Amazon S3.
  - Ihr Workflow hat bei Bedarf Zugriff auf externe Repository-Services.
- Deklarieren Sie die Ausgabedateien in der Workflow-Definition. Um Dateien für Zwischenläufe an den Ausgabespeicherort zu kopieren, deklarieren Sie sie als Workflow-Ausgaben.
- Die Eingabe- und Ausgabespeicherorte müssen sich in derselben Region wie der Workflow befinden.
- HealthOmics Die Eingaben für den Speicher-Workflow müssen den `ACTIVE` Status haben. HealthOmics importiert keine Eingaben mit einem `ARCHIVED` Status, wodurch der Workflow fehlschlägt. Informationen zu Amazon S3 S3-Objekteingaben finden Sie unterHealthOmics Eingaben ausführen.
- Ein main Speicherort des Workflows ist optional, wenn Ihr ZIP-Archiv entweder eine einzelne Workflow-Definition oder eine Datei mit dem Namen „main“ enthält.
  - Beispielpfad: `workflow-definition/main-file.wdl`
- Bevor Sie einen Workflow von Amazon S3 oder Ihrem lokalen Laufwerk aus erstellen, erstellen Sie ein ZIP-Archiv mit den Workflow-Definitionsdateien und allen Abhängigkeiten, z. B. Unterworkflows.
- Wir empfehlen Ihnen, Amazon ECR-Container im Workflow als Eingabeparameter für die Validierung der Amazon ECR-Berechtigungen zu deklarieren.

Zusätzliche Überlegungen zu Nextflow:

- /bin

  Nextflow-Workflow-Definitionen können einen /bin-Ordner mit ausführbaren Skripten enthalten. Dieser Pfad hat sowohl lesenden als auch ausführbaren Zugriff auf Aufgaben. Aufgaben, die auf diesen Skripten basieren, sollten einen Container verwenden, der mit den entsprechenden Skriptinterpretern erstellt wurde. Es empfiehlt sich, den Interpreter direkt aufzurufen. Zum Beispiel:

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

- includeConfig

  Auf NextFlow basierende Workflow-Definitionen können nextflow.config-Dateien enthalten, die dabei helfen, Parameterdefinitionen zu abstrahieren oder Ressourcenprofile zu verarbeiten. Um die Entwicklung und Ausführung von Nextflow-Pipelines in mehreren Umgebungen zu unterstützen, verwenden Sie eine HealthOmics -spezifische Konfiguration, die Sie der globalen Konfiguration mithilfe der IncludeConfig-Direktive hinzufügen. Um die Portabilität zu gewährleisten, konfigurieren Sie den Workflow mithilfe des folgenden Codes so, dass er die Datei nur einbezieht, wenn er darauf ausgeführt HealthOmics wird:

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

- Reports

  HealthOmics unterstützt keine vom Modul generierten Dag-, Trace- und Ausführungsberichte. Sie können Alternativen zu den Trace- und Ausführungsberichten mithilfe einer Kombination aus GetRun und GetRunTask API-Aufrufen generieren.

Zusätzliche Überlegungen zu CWL:

- Container image uri interpolation

HealthOmics ermöglicht, dass die DockerPull-Eigenschaft von ein DockerRequirement Inline-Javascript-Ausdruck ist. Zum Beispiel:

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

Auf diese Weise können Sie das Container-Image URIs als Eingabeparameter für den Workflow angeben.

- Javascript expressions

  Javascript-Ausdrücke müssen `strict mode` konform sein.

- Operation process

  HealthOmics unterstützt keine CWL-Operationsprozesse.

Versionsunterstützung für HealthOmics Workflow-Definitionssprachen

HealthOmics unterstützt Workflow-Definitionsdateien, die in Nextflow, WDL oder CWL geschrieben wurden. Die folgenden Abschnitte enthalten Informationen zur HealthOmics Versionsunterstützung für diese Sprachen.

Themen

- Unterstützung für WDL-Versionen
- Unterstützung für CWL-Versionen
- Unterstützung für die Nextflow-Version

Unterstützung für WDL-Versionen

HealthOmics unterstützt die Versionen 1.0, 1.1 und die Entwicklungsversion der WDL-Spezifikation.

Jedes WDL-Dokument muss eine Versionsanweisung enthalten, aus der hervorgeht, welcher Version (Haupt- und Nebenversion) der Spezifikation es entspricht. Weitere Informationen zu Versionen finden Sie unter WDL-Versionierung

Die Versionen 1.0 und 1.1 der WDL-Spezifikation unterstützen den Typ nicht. `Directory` Um den `Directory` Typ für Eingaben oder Ausgaben zu verwenden, setzen Sie die Version development in der ersten Zeile der Datei auf:

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

Unterstützung für CWL-Versionen

HealthOmics unterstützt die Versionen 1.0, 1.1 und 1.2 der CWL-Sprache.

Sie können die Sprachversion in der CWL-Workflow-Definitionsdatei angeben. Weitere Informationen zu CWL finden Sie im CWL-Benutzerhandbuch

Unterstützung für die Nextflow-Version

HealthOmics unterstützt drei stabile Versionen von Nextflow. Nextflow veröffentlicht in der Regel alle sechs Monate eine stabile Version. HealthOmics unterstützt die monatlichen „Edge“ - Veröffentlichungen nicht.

HealthOmics unterstützt in jeder Version veröffentlichte Funktionen, jedoch keine Vorschaufunktionen.

Unterstützte Versionen

HealthOmics unterstützt die folgenden Nextflow-Versionen:

- Nextflow v22.04.01 DSL 1 und DSL 2
- Nextflow v23.10.0 DSL 2 (Standard)
- Nextflow v24.10.8 DSL 2

Folgen Sie der Nextflow-Upgrade-Anleitung, um Ihren Workflow auf die neueste unterstützte Version (v24.10.8) zu migrieren.

Bei der Migration von Nextflow v23 zu v24 gibt es einige wichtige Änderungen, wie in den folgenden Abschnitten des Nextflow-Migrationsleitfadens beschrieben:

- Die wichtigsten Änderungen in 24.04
- Die wichtigsten Änderungen in 24.10

Nextflow-Versionen erkennen und verarbeiten

HealthOmics erkennt die von Ihnen angegebene DSL-Version und die Nextflow-Version. Basierend auf diesen Eingaben wird automatisch die beste Nextflow-Version ermittelt, die am besten ausgeführt werden kann.

DSL-Version

HealthOmics erkennt die angeforderte DSL-Version in Ihrer Workflow-Definitionsdatei. Sie können beispielsweise Folgendes angeben:`nextflow.enable.dsl=2`.

HealthOmics unterstützt standardmäßig DSL 2. Es bietet Abwärtskompatibilität mit DSL 1, sofern dies in Ihrer Workflow-Definitionsdatei angegeben ist.

- Wenn Sie DSL 2 angeben, HealthOmics wird Nextflow v23.10.0 ausgeführt, sofern Sie Nextflow v22.04.0 oder v24.10.8 nicht angeben.
- Wenn Sie DSL 1 angeben, wird Nextflow v22.04 ausgeführt (die einzige unterstützte Version, auf der DSL 1 HealthOmics ausgeführt wird). DSL1
- Wenn Sie keine DSL-Version angeben oder die DSL-Informationen aus irgendeinem Grund nicht analysieren HealthOmics können (z. B. Syntaxfehler in Ihrer Workflow-Definitionsdatei), wird HealthOmics standardmäßig DSL 2 verwendet und Nextflow v23.10.0 ausgeführt.
- Informationen zum Upgrade Ihres Workflows von DSL 1 auf DSL 2, um die neuesten Versionen und Softwarefunktionen von Nextflow zu nutzen, finden Sie unter Migration von DSL 1.

Nextflow-Versionen

HealthOmics erkennt die angeforderte Nextflow-Version in der Nextflow-Konfigurationsdatei (nextflow.config), wenn Sie diese Datei angeben. Wir empfehlen, dass Sie die `nextflowVersion` Klausel am Ende der Datei hinzufügen, um unerwartete Überschreibungen durch die enthaltenen Konfigurationen zu vermeiden. Weitere Informationen finden Sie unter Nextflow-Konfiguration.

Sie können eine Nextflow-Version oder einen Versionsbereich mit der folgenden Syntax angeben:

```
// 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 verarbeitet die Nextflow-Versionsinformationen wie folgt:

- Wenn Sie = eine genaue Version angeben, die HealthOmics unterstützt, HealthOmics verwendet diese Version.
- Wenn Sie ! eine exakte Version oder einen Bereich von Versionen angeben, die nicht unterstützt werden, HealthOmics löst dies eine Ausnahme aus und die Ausführung schlägt fehl. Erwägen Sie die Verwendung dieser Option, wenn Sie bei Versionsanfragen strikt vorgehen und schnell scheitern möchten, wenn die Anfrage Versionen enthält, die nicht unterstützt werden.
- Wenn Sie einen Versionsbereich angeben, wird die neueste unterstützte Version in diesem Bereich HealthOmics verwendet, es sei denn, der Bereich umfasst v24.10.8. In diesem Fall wird einer früheren HealthOmics Version der Vorzug gegeben. Wenn der Bereich beispielsweise sowohl v23.10.0 als auch v24.10.8 abdeckt, wählen Sie v23.10.0. HealthOmics
- Wenn es keine angeforderte Version gibt oder wenn die angeforderten Versionen nicht gültig sind oder aus irgendeinem Grund nicht analysiert werden können:
  - Wenn Sie DSL 1 angegeben haben, HealthOmics wird Nextflow v22.04 ausgeführt.
  - Andernfalls wird Nextflow v23.10.0 ausgeführt HealthOmics .

Sie können die folgenden Informationen über die Nextflow-Version abrufen, die für jeden Lauf HealthOmics verwendet wurde:

- Die Run-Logs enthalten Informationen über die tatsächliche Nextflow-Version, die für den Lauf HealthOmics verwendet wurde.
- HealthOmics fügt Warnungen in die Run-Logs ein, wenn es keine direkte Übereinstimmung mit der von Ihnen angeforderten Version gibt oder wenn eine andere Version als die von Ihnen angegebene verwendet werden musste.
- Die Antwort auf den GetRun API-Vorgang enthält ein Feld (`engineVersion`) mit der tatsächlichen Nextflow-Version, die für den Lauf HealthOmics verwendet wurde. Zum Beispiel:

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

Rechen- und Speicheranforderungen für HealthOmics Aufgaben

HealthOmics führt Ihre privaten Workflow-Aufgaben in einer Omics-Instanz aus. HealthOmics bietet eine Vielzahl von Instanztypen für verschiedene Arten von Aufgaben. Jeder Instance-Typ hat eine feste Speicher- und vCPU-Konfiguration (und eine feste GPU-Konfiguration für beschleunigte Recheninstanztypen). Die Kosten für die Nutzung einer Omics-Instanz variieren je nach Instanztyp. Einzelheiten finden Sie auf der Seite mit der HealthOmics Preisgestaltung.

Für Aufgaben in einem Workflow geben Sie den erforderlichen Arbeitsspeicher und v CPUs in der Workflow-Definitionsdatei an. Wenn eine Workflow-Aufgabe ausgeführt wird, HealthOmics ordnet die kleinste Omics-Instanz zu, die den angeforderten Speicher aufnehmen kann, und v. CPUs Wenn eine Aufgabe beispielsweise 64 GiB Speicher und 8 V benötigtCPUs, HealthOmics wählt aus`omics.r.2xlarge`.

Wir empfehlen Ihnen, die Instance-Typen zu überprüfen und die angeforderte V CPUs - und Speichergröße so einzustellen, dass sie der Instanz entsprechen, die Ihren Anforderungen am besten entspricht. Der Task-Container verwendet die Zahl von v CPUs und die Speichergröße, die Sie in Ihrer Workflow-Definitionsdatei angeben, auch wenn der Instance-Typ über zusätzliche V CPUs und zusätzlichen Speicher verfügt.

Die folgende Liste enthält zusätzliche Informationen zu vCPU und Speicherzuweisung:

- Bei der Zuweisung von Container-Ressourcen handelt es sich um feste Grenzwerte. Wenn für eine Aufgabe nicht genügend Arbeitsspeicher zur Verfügung steht oder wenn versucht wird, zusätzliche V zu verwendenCPUs , generiert die Aufgabe ein Fehlerprotokoll und wird beendet.
- Wenn Sie keine Rechen- oder Speicheranforderungen angeben, HealthOmics wählt eine Konfiguration mit 1 vCPU omics.c.large und 1 GiB Arbeitsspeicher aus und verwendet diese standardmäßig.
- Die Mindestkonfiguration, die Sie anfordern können, ist 1 vCPU und 1 GiB Arbeitsspeicher.
- Wenn Sie vCPUs, memory oder GPUs that angeben, das die unterstützten Instance-Typen überschreitet, HealthOmics wird eine Fehlermeldung ausgegeben und der Workflow schlägt bei den Validierungen fehl
- Wenn Sie Brucheinheiten angeben, wird auf die nächste HealthOmics Ganzzahl aufgerundet.

- HealthOmics reserviert eine kleine Menge an Speicher (5%) für Verwaltungs- und Protokollierungsagenten, sodass der Anwendung in der Aufgabe möglicherweise nicht immer die gesamte Speicherzuweisung zur Verfügung steht.
- HealthOmics passt die Instanztypen an die von Ihnen angegebenen Rechen- und Speicheranforderungen an und verwendet möglicherweise eine Mischung aus Hardwaregenerationen. Aus diesem Grund kann es zu geringfügigen Abweichungen bei den Ausführungszeiten von Aufgaben für dieselbe Aufgabe kommen.

Diese Themen enthalten Einzelheiten zu den HealthOmics unterstützten Instance-Typen.

Themen

> **Note**
>
> Erhöhen Sie bei standardmäßigen, rechen- und speicheroptimierten Instances die Größe der Instance-Bandbreite, wenn die Instance einen höheren Durchsatz benötigt. Bei Amazon EC2 EC2-Instances mit weniger als 16 vCPUs (Größe 4xl und kleiner) kann es zu Durchsatzsteigerungen kommen. Weitere Informationen zum Durchsatz von Amazon EC2 EC2-Instances finden Sie unter Verfügbare Amazon EC2 EC2-Instance-Bandbreite.

Standard-Instance-Typen

Bei Standard-Instance-Typen zielen die Konfigurationen auf ein ausgewogenes Verhältnis von Rechenleistung und Arbeitsspeicher ab.

HealthOmics unterstützt die Instances 32xlarge und 48xlarge in den folgenden Regionen: USA West (Oregon) und USA Ost (Nord-Virginia).

Instance	Anzahl von v CPUs	Arbeitsspeicher
omics.m.large	2	8 GiB

Instance	Anzahl von v CPUs	Arbeitsspeicher
omics.m.x groß	4	16 GiB
omics.m.2 x groß	8	32 GiB
omics.m.4xgroß	16	64 GiB
omics.m.8xgroß	32	128 GiB
omics.m. 12 x groß	48	192 GiB
omics.m.16x groß	64	256 GiB
omics.m.24x groß	96	384 GiB
omics.m.32x groß	128	512 GiB
omics.m.48x groß	192	768 GiB

Für die Datenverarbeitung optimierte Instanzen

Bei rechenoptimierten Instance-Typen verfügen die Konfigurationen über mehr Rechenleistung und weniger Arbeitsspeicher.

HealthOmics unterstützt die Instances 32xlarge und 48xlarge in den folgenden Regionen: USA West (Oregon) und USA Ost (Nord-Virginia).

Instance	Anzahl von v CPUs	Arbeitsspeicher
omics.c.large	2	4 GiB
omics.c.xlarge	4	8 GiB
omics.c.2 x groß	8	16 GiB
omics.c.4xlarge	16	32 GiB
omics.c.8xlarge	32	64 GiB

Instance	Anzahl von v CPUs	Arbeitsspeicher
omics.c. 12 x groß	48	96 GiB
omics.c. 16 x groß	64	128 GiB
omics.c.24xlarge	96	192 GiB
omics.c.32xlarge	128	256 GiB
omics.c.48xlarge	192	384 GiB

Für den Arbeitsspeicher optimierte Instanzen

Bei speicheroptimierten Instance-Typen verfügen die Konfigurationen über weniger Rechenleistung und mehr Arbeitsspeicher.

HealthOmics unterstützt die Instances 32xlarge und 48xlarge in den folgenden Regionen: USA West (Oregon) und USA Ost (Nord-Virginia).

Instance	Anzahl von v CPUs	Arbeitsspeicher
omics.r.large	2	16 GiB
omic s.r.x groß	4	32 GiB
omic s.r.2 x groß	8	64 GiB
omic s.r.4 x groß	16	128 GiB
omic s.r.8 x groß	32	256 GiB
omic s.r.12 x groß	48	384 GiB
omic s.r.16 x groß	64	512 GiB
omic s.r.24 x groß	96	768 GiB
omic s.r.32 x groß	128	1024 GiB

Instance	Anzahl von v CPUs	Arbeitsspeicher
omic s.r.48 x groß	192	1536 GiB

Instanzen für beschleunigte Datenverarbeitung

Sie können optional GPU-Ressourcen für jede Aufgabe in einem Workflow angeben, sodass der Aufgabe eine Instanz für HealthOmics beschleunigtes Rechnen zugewiesen wird. Informationen zum Angeben der GPU-Informationen in der Workflow-Definitionsdatei finden Sie unter. Aufgabenbeschleuniger in einer Workflow-Definition HealthOmics

Wenn Sie eine GPU angeben, die mehrere Instanztypen unterstützt, HealthOmics wählt Sie den Instanztyp je nach Verfügbarkeit aus. Wenn beide Instance-Typen verfügbar sind HealthOmics , wird die kostengünstigere Instanz bevorzugt.

G4-Instances werden in der Region Israel (Tel Aviv) nicht unterstützt. G5-Instances werden in der Region Asien-Pazifik (Singapur) nicht unterstützt.

Themen

- Instance-Typen G6 und G6e
- G4- und G5-Instanzen

Instance-Typen G6 und G6e

HealthOmics unterstützt die folgenden G6-Instance-Konfigurationen für beschleunigtes Rechnen. Alle omics.g6-Instances verwenden Nvidia L4 oder Nvidia L4 A10G. GPUs

HealthOmics unterstützt die G6- und G6e-Instances in den folgenden Regionen: USA West (Oregon) und USA Ost (Nord-Virginia).

Instance	Anzahl von v CPUs	Arbeitssp eicher	Anzahl von GPUs	GPU-Arbei tsspeicher	
omics.g6. xlarge	4	16 GiB	1	24 GiB	

Instance	Anzahl von v CPUs	Arbeitssp eicher	Anzahl von GPUs	GPU-Arbei tsspeicher	
omics.g 6.2 x groß	8	32 GiB	1	24 GiB	
omics.g 6.4x groß	16	64 GiB	1	24 GiB	
omics.g 6.8 x groß	32	128 GiB	1	24 GiB	
omics.g 6.12 x groß	48	192 GiB	4	96 GiB	
omics.g 6.16 x groß	64	256 GiB	1	24 GiB	
omics.g 6.24x groß	96	384 GiB	4	96 GiB	

Alle omics.g6e-Instanzen verwenden Nvidia L40s. GPUs

Instance	Anzahl von v CPUs	Arbeitssp eicher	Anzahl von GPUs	GPU-Arbei tsspeicher	
omics.g6e .xlarge	4	32 GiB	1	48 GiB	
omics.g6e .2xlarge	8	64 GiB	1	48 GiB	
omics.g6e.4x groß	16	128 GiB	1	48 GiB	
omics.g6e .8xgroß	32	256 GiB	1	48 GiB	

Instance	Anzahl von v CPUs	Arbeitssp eicher	Anzahl von GPUs	GPU-Arbei tsspeicher	
omics.g6e .12xlarge	48	384 GiB	4	192 GiB	
omics.g6e .16x groß	64	512 GiB	1	48 GiB	
omics.g6e .24x groß	96	768 GiB	4	192 GiB	

G4- und G5-Instanzen

HealthOmics unterstützt die folgenden G4- und G5-Instance-Konfigurationen für beschleunigtes Rechnen.

Alle omics.g5-Instances verwenden Nvidia L4 A10G, Nvidia Tesla A10G oder Nvidia Tesla T4 A10G. GPUs

Instance	Anzahl von v CPUs	Arbeitssp eicher	Anzahl von GPUs	GPU-Arbei tsspeicher	
omics.g5. xlarge	4	16 GiB	1	24 GiB	
omics.g5.2 x groß	8	32 GiB	1	24 GiB	
omics.g 5.4 x groß	16	64 GiB	1	24 GiB	
omics.g 5.8x groß	32	128 GiB	1	24 GiB	
omics.g 5.12 x groß	48	192 GiB	4	96 GiB	

Instance	Anzahl von v CPUs	Arbeitssp eicher	Anzahl von GPUs	GPU-Arbei tsspeicher	
omics.g 5.16 x groß	64	256 GiB	1	24 GiB	
omics.g 5.24x groß	96	384 GiB	4	96 GiB	

Alle omics.g4dn-Instances verwenden Nvidia Tesla T4 oder Nvidia Tesla T4 A10G. GPUs

Instance	Anzahl von v CPUs	Arbeitssp eicher	Anzahl von GPUs	GPU-Arbei tsspeicher	
omics.g4d n.xlarge	4	16 GiB	1	16 GiB	
omics.g4d n.2xlarge	8	32 GiB	1	16 GiB	
omics.g4d n.4xgroß	16	64 GiB	1	16 GiB	
omics.g4d n.8xgroß	32	128 GiB	1	16 GiB	
omics.g4d n.12xlarge	48	192 GiB	4	64 GiB	
omics.g4d n.16xlarge	64	256 GiB	1	24 GiB	

Aufgabenausgaben in einer HealthOmics Workflow-Definition

Sie geben Aufgabenausgaben in der Workflow-Definition an. Verwirft standardmäßig alle zwischengeschalteten Aufgabendateien, HealthOmics wenn der Workflow abgeschlossen ist. Um eine Zwischendatei zu exportieren, definieren Sie sie als Ausgabe.

Wenn Sie das Aufruf-Caching verwenden, werden die Ausgaben der Aufgaben im Cache HealthOmics gespeichert, einschließlich aller Zwischendateien, die Sie als Ausgaben definieren.

Die folgenden Themen enthalten Beispiele für Aufgabendefinitionen für jede der Workflow-Definitionssprachen.

Themen

- Aufgabenausgaben für WDL
- Task-Ausgaben für Nextflow
- Aufgabenausgaben für CWL

Aufgabenausgaben für WDL

Für in WDL geschriebene Workflow-Definitionen definieren Sie Ihre Ausgaben im outputs Workflow-Bereich der obersten Ebene.

HealthOmics

Themen

- Aufgabenausgabe für STDOUT
- Aufgabenausgabe für STDERR
- Ausgabe der Aufgabe in eine Datei
- Ausgabe der Aufgabe in ein Array von Dateien

Aufgabenausgabe für STDOUT

In diesem Beispiel wird eine Aufgabe mit dem Namen erstelltSayHello, die den STDOUT-Inhalt in die Aufgabenausgabedatei zurückgibt. Die stdout WDL-Funktion erfasst den STDOUT-Inhalt (in diesem Beispiel die Eingabezeichenfolge Hello World!) in einer Datei. stdout_file

Da Protokolle für den gesamten STDOUT-Inhalt HealthOmics erstellt werden, wird die Ausgabe zusammen mit anderen CloudWatch STDERR-Protokollierungsinformationen für die Aufgabe auch in Logs angezeigt.

```
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()
    }
}
```

Aufgabenausgabe für STDERR

In diesem Beispiel wird eine Aufgabe mit dem Namen erstelltSayHello, die den STDERR-Inhalt in die Aufgabenausgabedatei zurückgibt. Die stderr WDL-Funktion erfasst den STDERR-Inhalt (in diesem Beispiel die Eingabezeichenfolge Hello World!) in einer Datei. stderr_file

Da Protokolle für den gesamten STDERR-Inhalt HealthOmics erstellt werden, wird die Ausgabe zusammen mit anderen CloudWatch STDERR-Protokollierungsinformationen für die Aufgabe in Logs angezeigt.

```
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()
```

```
    }
}
```

Ausgabe der Aufgabe in eine Datei

In diesem Beispiel erstellt die SayHello Aufgabe zwei Dateien (message.txt und info.txt) und deklariert diese Dateien explizit als die benannten Ausgaben (message_file und 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"
    }
}
```

Ausgabe der Aufgabe in ein Array von Dateien

In diesem Beispiel generiert die `GenerateGreetings` Aufgabe ein Array von Dateien als Aufgabenausgabe. Die Aufgabe generiert dynamisch eine Begrüßungsdatei für jedes Mitglied des Eingabearrays`names`. Da die Dateinamen erst zur Laufzeit bekannt sind, verwendet die Ausgabedefinition die WDL-Funktion glob (), um alle Dateien auszugeben, die dem Muster entsprechen. `*_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")
    }
 }
```

Task-Ausgaben für Nextflow

Definieren Sie für in Nextflow geschriebene Workflow-Definitionen eine publishDir-Direktive, um Aufgabeninhalte in Ihren Amazon S3 S3-Ausgabe-Bucket zu exportieren. Setzen Sie den Wert publishDir auf. `/mnt/workflow/pubdir`

HealthOmics Um Dateien nach Amazon S3 exportieren zu können, müssen sich die Dateien in diesem Verzeichnis befinden.

Wenn eine Aufgabe eine Gruppe von Ausgabedateien erzeugt, die als Eingaben für eine nachfolgende Aufgabe verwendet werden sollen, empfehlen wir, diese Dateien in einem Verzeichnis zu gruppieren und das Verzeichnis als Aufgabenausgabe auszugeben. Das Aufzählen jeder einzelnen Datei kann zu einem I/O-Engpass im zugrunde liegenden Dateisystem führen. Zum Beispiel:

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

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

```
    ...
}
```

Aufgabenausgaben für CWL

Bei in CWL geschriebenen Workflow-Definitionen können Sie die Aufgabenausgaben mithilfe von `CommandLineTool` Aufgaben angeben. Die folgenden Abschnitte zeigen Beispiele für `CommandLineTool` Aufgaben, die verschiedene Arten von Ausgaben definieren.

Themen

Aufgabenausgabe für STDOUT

In diesem Beispiel wird eine `CommandLineTool` Aufgabe erstellt, die den STDOUT-Inhalt in einer Textausgabedatei mit dem Namen wiedergibt. output.txt Wenn Sie beispielsweise die folgende Eingabe angeben, lautet die resultierende Aufgabenausgabe Hello World! in der output.txt Datei.

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

Die `outputs` Direktive gibt an, dass der Ausgabename example_out und der Typ lauten`stdout`. Damit eine Downstream-Aufgabe die Ausgabe dieser Aufgabe verarbeitet, würde sie die Ausgabe als bezeichnen`example_out`.

Da Protokolle für den gesamten STDERR- und STDOUT-Inhalt HealthOmics erstellt werden, wird die Ausgabe zusammen mit anderen CloudWatch STDERR-Protokollierungsinformationen für die Aufgabe auch in Logs angezeigt.

```
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
```

Aufgabenausgabe für STDERR

In diesem Beispiel wird eine `CommandLineTool` Aufgabe erstellt, die den STDERR-Inhalt in einer Textausgabedatei mit dem Namen wiedergibt. stderr.txt Die Aufgabe ändert die `baseCommand` so, dass sie in STDERR (statt STDOUT) `echo` schreibt.

Die `outputs` Direktive gibt an, dass der Ausgabename stderr_out und der Typ lautet. `stderr`

Da Protokolle für den gesamten STDERR- und STDOUT-Inhalt HealthOmics erstellt werden, wird die Ausgabe zusammen mit anderen CloudWatch STDERR-Protokollierungsinformationen für die Aufgabe in Logs angezeigt.

```
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
```

Ausgabe der Aufgabe in eine Datei

In diesem Beispiel wird eine `CommandLineTool` Aufgabe erstellt, die aus den Eingabedateien ein komprimiertes Tar-Archiv erstellt. Sie geben den Namen des Archivs als Eingabeparameter an (archive_name).

Die outputs Direktive gibt an, dass der `archive_file` Ausgabetyp ist`File`, und sie verwendet einen Verweis auf den Eingabeparameter`archive_name`, um eine Bindung an die Ausgabedatei herzustellen.

```
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
```

Ausgabe der Aufgabe in ein Array von Dateien

In diesem Beispiel erstellt die `CommandLineTool` Aufgabe mithilfe des `touch` Befehls ein Array von Dateien. Der Befehl verwendet die Zeichenketten im `files-to-create` Eingabeparameter, um die Dateien zu benennen. Der Befehl gibt ein Array von Dateien aus. Das Array enthält alle Dateien im Arbeitsverzeichnis, die dem `glob` Muster entsprechen. In diesem Beispiel wird ein Platzhaltermuster („*“) verwendet, das allen Dateien entspricht.

```
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
```

Aufgabenressourcen in einer HealthOmics Workflow-Definition

Definieren Sie in der Workflow-Definition für jede Aufgabe Folgendes:

- Das Container-Image für die Aufgabe. Weitere Informationen finden Sie unter Container-Images für private Workflows.

- Die Anzahl CPUs und der für die Aufgabe benötigte Speicher. Weitere Informationen finden Sie unter Rechen- und Speicheranforderungen für HealthOmics Aufgaben.

HealthOmics ignoriert alle Speicherspezifikationen pro Aufgabe. HealthOmics stellt Ausführungsspeicher bereit, auf den alle Aufgaben in der Ausführung zugreifen können. Weitere Informationen finden Sie unter Speichertypen in HealthOmics Workflows ausführen.

WDL

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

Bei einem WDL-Workflow werden bis zu zwei HealthOmics Wiederholungsversuche für eine Aufgabe versucht, die aufgrund von Dienstfehlern fehlschlägt (die API-Anfrage gibt einen 5XX-HTTP-Statuscode zurück). Weitere Informationen zu Wiederholungsversuchen von Aufgaben finden Sie unter. Die Aufgabe wird erneut versucht

Sie können das Wiederholungsverhalten deaktivieren, indem Sie die folgende Konfiguration für die Aufgabe in der WDL-Definitionsdatei angeben:

```
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
```

Aufgabenbeschleuniger in einer Workflow-Definition HealthOmics

In der Workflow-Definition können Sie optional die GPU-Beschleuniger-Spezifikation für eine Aufgabe angeben. HealthOmics unterstützt die folgenden Werte für die Beschleunigerspezifikation zusammen mit den unterstützten Instanztypen:

Beschleuniger-Spezifikation	Instanztypen von Healthomics				
nvidia-tesla-t4	G4				
nvidia-tesla-t4-a 10 g	G4 und G5				
nvidia-tesla-a10 g	G5				
nvidia-l4-a 10 g	G5 und G6				
nvidia-l4	G6				
nvidia-l40	G6e				

Wenn Sie einen Beschleunigertyp angeben, der mehrere Instance-Typen unterstützt, HealthOmics wählt der Instance-Typ auf der Grundlage der verfügbaren Kapazität aus. Wenn beide Instance-Typen verfügbar sind HealthOmics , wird die kostengünstigere Instanz bevorzugt.

Einzelheiten zu den Instance-Typen finden Sie unterInstanzen für beschleunigte Datenverarbeitung.

Im folgenden Beispiel gibt die Workflow-Definition `nvidia-l4` als Accelarator an:

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"
```

Besonderheiten der WDL-Workflow-Definition

Die folgenden Themen enthalten Einzelheiten zu den Typen und Anweisungen, die für WDL-Workflow-Definitionen in verfügbar sind. HealthOmics

Themen

Implizite Typkonvertierung in WDL Lenient

HealthOmics unterstützt die implizite Typkonvertierung in der Datei input.json und der Workflow-Definition. Um implizite Typumwandlung zu verwenden, geben Sie bei der Erstellung des Workflows für die Workflow-Engine den Wert WDL Lenient an. WDL Lenient wurde für Workflows entwickelt, die von Cromwell migriert wurden. Es unterstützt Cromwell-Richtlinien von Kunden und einige nichtkonforme Logiken.

WDL Lenient unterstützt die Typkonvertierung für die folgenden Elemente in der Liste der eingeschränkten Ausnahmen von WDL:

- Float wird in Int umgewandelt, wobei der Zwang zu keinem Genauigkeitsverlust führt (z. B. 1,0 entspricht 1).
- Von String zu Int/Float, wobei der Zwang zu keinem Genauigkeitsverlust führt.
- Ordnen Sie [W, X] dem Array [Pair [Y, Z]] zu, falls W gegen Y und X gegen Z erzwingbar ist.
- Ordnen Sie [Paar [W, X]] an, um [Y, Z] zuzuordnen, falls W gegen Y und X gegen Z erzwingbar ist (z. B. 1,0 entspricht 1).

Um implizites Typ-Casting zu verwenden, geben Sie die Workflow-Engine als WDL_LENIENT an, wenn Sie den Workflow oder die Workflow-Version erstellen.

In der Konsole heißt der Workflow-Engine-Parameter Language. In der API heißt der Workflow-Engine-Parameter Engine. Für weitere Informationen siehe Erstellen Sie einen privaten Workflow oder Workflow-Version erstellen.

Namespace-Definition in input.json

HealthOmics unterstützt vollständig qualifizierte Variablen in input.json. Wenn Sie beispielsweise zwei Eingabevariablen mit den Namen number1 und number2 im Workflow deklarieren: SumWorkflow

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

Sie können sie als vollqualifizierte Variablen in input.json verwenden:

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

Primitive Typen in WDL

Die folgende Tabelle zeigt, wie Eingaben in WDL den entsprechenden primitiven Typen zugeordnet werden. HealthOmics bietet eingeschränkte Unterstützung für Typenzwang, daher empfehlen wir, explizite Typen festzulegen.

Primitive Typen

WDL-Typ	JSON-Typ	Beispiel WDL	Beispiel für einen JSON-Schlüssel und -Wert	Hinweise
`Boolean`	`boolean`	`Boolean b`	`"b": true`	Der Wert muss in Kleinbuchstaben geschrieben werden und darf keine

WDL-Typ	JSON-Typ	Beispiel WDL	Beispiel für einen JSON-Schlüssel und -Wert	Hinweise
				Anführungszeichen enthalten.
`Int`	`integer`	`Int i`	`"i": 7`	Darf nicht in Anführungszeichen gesetzt werden.
`Float`	`number`	`Float f`	`"f": 42.2`	Darf nicht in Anführungszeichen stehen.
`String`	`string`	`String s`	`"s": "characters"`	JSON-Zeichenfolgen, die eine URI sind, müssen einer zu importierenden WDL-Datei zugeordnet werden.

WDL-Typ	JSON-Typ	Beispiel WDL	Beispiel für einen JSON-Schlüssel und -Wert	Hinweise
`File`	`string`	`File f`	`"f": "s3://amzn-s3-demo-bucket1/path/to/file"`	Amazon S3 und HealthOmics Speicher URIs werden importiert, solange die für den Workflow bereitgestellte IAM-Rolle Lesezugriff auf diese Objekte hat. Andere URI-Schemas werden nicht unterstützt (wie `file://` `https://`, und `ftp://`). Die URI muss ein Objekt angeben. Es kann kein Verzeichnis sein, was bedeutet, dass es nicht mit einem enden kann `/`.

WDL-Typ	JSON-Typ	Beispiel WDL	Beispiel für einen JSON-Schlüssel und -Wert	Hinweise
`Directory`	`string`	`Directory d`	`"d": "s3://bucket/path/"`	Der `Directory` Typ ist nicht in WDL 1.0 oder 1.1 enthalten, daher müssen Sie ihn zum Header der WDL-Datei hinzufüge `nversion development` . Die URI muss eine Amazon S3 S3-URI sein und ein Präfix haben, das mit einem '/' endet. Der gesamte Inhalt des Verzeichnisses wird rekursiv als einziger Download in den Workflow kopiert. Der `Directory` sollte nur Dateien enthalten, die sich auf den Workflow beziehen.

Komplexe Typen in WDL

Die folgende Tabelle zeigt, wie Eingaben in WDL den entsprechenden komplexen JSON-Typen zugeordnet werden. Komplexe Typen in WDL sind Datenstrukturen, die aus primitiven Typen bestehen. Datenstrukturen wie Listen werden in Arrays umgewandelt.

Komplexe Typen

Typ WDL	JSON-Typ	Beispiel WDL	Beispiel für einen JSON-Schlüssel und -Wert	Hinweise
`Array`	`array`	`Array[Int] nums`	`“nums": [1, 2, 3]`	Die Mitglieder des Arrays müssen dem Format des WDL-Arraytyps folgen.
`Pair`	`object`	`Pair[String, Int] str_to_i`	`“str_to_i": {"left": "0", "right": 1}`	Jeder Wert des Paares muss das JSON-Format des entsprechenden WDL-Typs verwenden.
`Map`	`object`	`Map[Int, String] int_to_string`	`"int_to_string": { 2: "hello", 1: "goodbye" }`	Jeder Eintrag in der Map muss das JSON-Format des entsprechenden WDL-Typs verwenden.
`Struct`	`object`	`struct SampleBamAndIndex { String sample_name`	`"b_and_i": { "sample_name": "NA12878",`	Die Namen der Strukturmitglieder müssen exakt mit den Namen der JSON-Obje

Typ WDL	JSON-Typ	Beispiel WDL	Beispiel für einen JSON-Schlüssel und -Wert	Hinweise
		`File bam File bam_index } SampleBamAndIndex b_and_i`	`"bam": "s3://amzn-s3-demo-bucket1/NA12878.bam", "bam_index": "s3://amzn-s3-demo-bucket1/NA12878.bam.bai" }`	ktschlüssel übereinstimmen. Jeder Wert muss das JSON-Format des entsprechenden WDL-Typs verwenden.
`Object`	–	–	–	Der `Object` WDL-Typ ist veraltet und sollte `Struct` in jedem Fall durch ersetzt werden.

Richtlinien in WDL

HealthOmics unterstützt die folgenden Direktiven in allen WDL-Versionen, die HealthOmics sie unterstützen.

GPU-Ressourcen konfigurieren

HealthOmics unterstützt Laufzeitattribute acceleratorType und acceleratorCount mit allen unterstützten GPU-Instanzen. HealthOmics unterstützt auch Aliase mit dem Namen gpuType undgpuCount, die dieselbe Funktionalität wie ihre Accelerator-Gegenstücke haben. Wenn die WDL-Definition beide Direktiven enthält, HealthOmics verwendet die Beschleunigerwerte.

Das folgende Beispiel zeigt, wie diese Direktiven verwendet werden:

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

Konfigurieren Sie die Aufgabenwiederholung bei Servicefehlern

HealthOmics unterstützt bis zu zwei Wiederholungen für eine Aufgabe, die aufgrund von Dienstfehlern fehlgeschlagen ist (5XX-HTTP-Statuscodes). Sie können die maximale Anzahl von Wiederholungen (1 oder 2) konfigurieren und Wiederholungen aufgrund von Servicefehlern deaktivieren. Standardmäßig werden maximal zwei Wiederholungen HealthOmics versucht.

Das folgende Beispiel legt fest`preemptible`, dass Wiederholungsversuche bei Dienstfehlern deaktiviert werden:

```
{
  preemptible: 0
}
```

Weitere Hinweise zu Wiederholungsversuchen von Aufgaben finden Sie unter HealthOmics. Die Aufgabe wird erneut versucht

Konfigurieren Sie die Aufgabenwiederholung bei fehlendem Arbeitsspeicher

HealthOmics unterstützt Wiederholungsversuche für eine Aufgabe, die fehlgeschlagen ist, weil ihr nicht genügend Speicherplatz zur Verfügung stand (Container-Exitcode 137, 4XX HTTP-Statuscode). HealthOmics verdoppelt die Speichermenge für jeden Wiederholungsversuch.

Standardmäßig versucht es bei dieser Art von Fehler HealthOmics nicht erneut. Verwenden Sie die `maxRetries` Direktive, um die maximale Anzahl von Wiederholungen anzugeben.

Im folgenden Beispiel wird der `maxRetries` Wert auf 3 gesetzt, sodass HealthOmics maximal vier Versuche unternommen werden, die Aufgabe abzuschließen (der erste Versuch plus drei Wiederholungen):

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

> **Note**
>
> Für die Wiederholung der Aufgabe bei fehlendem Arbeitsspeicher sind GNU Findutils 4.2.3+ erforderlich. Der Standard-Image-Container enthält dieses Paket HealthOmics . Wenn Sie in Ihrer WDL-Definition ein benutzerdefiniertes Bild angeben, stellen Sie sicher, dass das Bild GNU Findutils 4.2.3+ enthält.

Konfigurieren Sie Rückgabecodes

Das ReturnCodes-Attribut bietet einen Mechanismus zur Angabe eines Rückgabecodes oder einer Reihe von Rückgabecodes, die auf eine erfolgreiche Ausführung einer Aufgabe hinweisen. Das WDL-Modul berücksichtigt die Rückgabecodes, die Sie im Runtime-Abschnitt der WDL-Definition angeben, und legt den Aufgabenstatus entsprechend fest.

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

HealthOmics unterstützt auch einen Alias namens continueOnReturnCode, der dieselben Funktionen wie ReturnCodes hat. Wenn Sie beide Attribute angeben, wird der ReturnCodes-Wert HealthOmics verwendet.

Aufgaben-Metadaten in WDL

HealthOmics unterstützt die folgenden Metadatenoptionen für WDL-Aufgaben.

Deaktivieren Sie das Caching auf Aufgabenebene mit dem Attribut volatile

Mit dem flüchtigen Attribut können Sie das Caching von Anrufen für bestimmte Aufgaben in Ihrem WDL-Workflow deaktivieren. Wenn eine Aufgabe als flüchtig markiert ist, wird sie immer ausgeführt und verwendet niemals zwischengespeicherte Ergebnisse, selbst wenn das Caching für die Ausführung aktiviert ist.

Fügen Sie das flüchtige Attribut zum Metabereich Ihrer Aufgabendefinition hinzu:

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

```
    input {
        String input_file
    }

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

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

Beispiel für eine WDL-Workflow-Definition

Die folgenden Beispiele zeigen private Workflow-Definitionen für die Konvertierung von `CRAM` zu `BAM` in WDL. Der `CRAM BAM` To-Workflow definiert zwei Aufgaben und verwendet Tools aus dem `genomes-in-the-cloud` Container, der im Beispiel gezeigt wird und öffentlich verfügbar ist.

Das folgende Beispiel zeigt, wie der Amazon ECR-Container als Parameter eingebunden wird. Auf diese Weise können HealthOmics Sie die Zugriffsberechtigungen für Ihren Container überprüfen, bevor der Run gestartet wird.

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

Das folgende Beispiel zeigt, wie Sie angeben, welche Dateien in Ihrem Lauf verwendet werden sollen, wenn sich die Dateien in einem Amazon S3 S3-Bucket befinden.

```
{
    "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"
}
```

Wenn Sie Dateien aus einem Sequenzspeicher angeben möchten, geben Sie dies wie im folgenden Beispiel gezeigt an, indem Sie den URI für den Sequenzspeicher verwenden.

```
{
    "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"
}
```

Anschließend können Sie Ihren Workflow in WDL definieren, wie im folgenden Beispiel gezeigt.

```
 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}"
    }
}
```

Einzelheiten der Nextflow-Workflow-Definition

HealthOmics unterstützt DSL1 Nextflow und. DSL2 Details hierzu finden Sie unter Unterstützung für die Nextflow-Version.

Nextflow DSL2 basiert auf der Programmiersprache Groovy, sodass Parameter dynamisch sind und Typenzwang nach den gleichen Regeln wie Groovy möglich ist. Parameter und Werte, die von der JSON-Eingabe bereitgestellt werden, sind in der Parameters () -Map des Workflows verfügbar. `params`

Themen

- Verwenden Sie die Plug-ins NF-Schema und NF-Validation
- Speicher angeben URIs
- Nextflow-Direktiven
- Aufgabeninhalt exportieren

Verwenden Sie die Plug-ins NF-Schema und NF-Validation

> **Note**
>
> Zusammenfassung der Unterstützung für Plugins HealthOmics :
>
> - v22.04 — keine Unterstützung für Plugins
> - v23.10 — unterstützt und `nf-schema` `nf-validation`
> - v24.10 — unterstützt `nf-schema`

HealthOmics bietet die folgende Unterstützung für Nextflow-Plugins:

- Für Nextflow v23.10 ist das Plugin nf-validation @1 HealthOmics .1.1 vorinstalliert.
- Für Nextflow v23.10 und höher wird das Plugin nf-schema @2 .3.0 vorinstalliert. HealthOmics
- Sie können während einer Workflow-Ausführung keine zusätzlichen Plugins abrufen. HealthOmics ignoriert alle anderen Plugin-Versionen, die Sie in der `nextflow.config` Datei angeben.
- Für Nextflow v24 und höher `nf-schema` ist dies die neue Version des veralteten Plugins. `nf-validation` Weitere Informationen finden Sie unter nf-schema im Nextflow-Repository. GitHub

Speicher angeben URIs

Wenn ein Amazon S3 oder HealthOmics URI verwendet wird, um eine Nextflow-Datei oder ein Nextflow-Pfadobjekt zu erstellen, stellt es das entsprechende Objekt für den Workflow zur Verfügung, sofern Lesezugriff gewährt wird. Die Verwendung von Präfixen oder Verzeichnissen ist für Amazon S3 URIs zulässig. Beispiele finden Sie unter Amazon S3 S3-Eingabeparameterformate.

HealthOmics unterstützt teilweise die Verwendung von Glob Patterns in Amazon S3 URIs oder HealthOmics Storage URIs. Verwenden Sie Glob-Muster in der Workflow-Definition für die Erstellung von Or-Kanälen`path`. `file` Informationen zum erwarteten Verhalten und zu den genauen Fällen finden Sie unterNextflow-Behandlung von Glob-Mustern in Amazon S3 S3-Eingaben.

Nextflow-Direktiven

Sie konfigurieren Nextflow-Direktiven in der Nextflow-Konfigurationsdatei oder Workflow-Definition. Die folgende Liste zeigt die Rangfolge, in der die HealthOmics Konfigurationseinstellungen angewendet werden, von der niedrigsten zur höchsten Priorität:

1. Globale Konfiguration in der Konfigurationsdatei.

2. Aufgabenbereich der Workflow-Definition.
3. Aufgabenspezifische Selektoren in der Konfigurationsdatei.

Themen

Strategie zur Wiederholung von Aufgaben unter Verwendung von **`errorStrategy`**

Verwenden Sie die `errorStrategy` Direktive, um die Strategie für Aufgabenfehler zu definieren. Wenn eine Aufgabe mit einer Fehleranzeige (einem Exit-Status ungleich Null) zurückkehrt, wird die Aufgabe standardmäßig angehalten und die gesamte HealthOmics Ausführung beendet. Wenn Sie `errorStrategy` auf festlegen, wird HealthOmics versuch`retry`, die fehlgeschlagene Aufgabe erneut zu versuchen. Informationen zur Erhöhung der Anzahl der Wiederholungen finden Sie unter. Versuche, Aufgaben erneut zu versuchen, verwenden `maxRetries`

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

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

Informationen darüber, wie mit Wiederholungsversuchen HealthOmics von Aufgaben während einer Ausführung umgegangen wird, finden Sie unter. Die Aufgabe wird erneut versucht

Versuche, Aufgaben erneut zu versuchen, verwenden **`maxRetries`**

Führt standardmäßig HealthOmics keine Wiederholungsversuche für eine fehlgeschlagene Aufgabe durch, oder versucht einen erneuten Versuch, wenn Sie dies konfigurieren. `errorStrategy` Um die maximale Anzahl von Wiederholungen zu erhöhen, legen `errorStrategy` Sie die maximale Anzahl von Wiederholungen fest `retry` und konfigurieren Sie sie mithilfe der Direktive. `maxRetries`

Im folgenden Beispiel wird die maximale Anzahl von Wiederholungen in der globalen Konfiguration auf 3 festgelegt.

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

Das folgende Beispiel zeigt, wie `maxRetries` im Aufgabenbereich der Workflow-Definition festgelegt wird.

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

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

Das folgende Beispiel zeigt, wie eine aufgabenspezifische Konfiguration in der Nextflow-Konfigurationsdatei auf der Grundlage der Namens- oder Labelselektoren angegeben wird.

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

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

Deaktivieren Sie die Wiederholung von Aufgaben mit **`omicsRetryOn5xx`**

HealthOmics Unterstützt für Nextflow v23 und v24 Aufgabenwiederholungen, wenn die Aufgabe aufgrund von Dienstfehlern fehlgeschlagen ist (5XX-HTTP-Statuscodes). Standardmäßig werden bis zu zwei Wiederholungen einer fehlgeschlagenen Aufgabe HealthOmics versucht.

Sie können so konfigurieren`omicsRetryOn5xx`, dass die Wiederholung von Aufgaben bei Dienstfehlern deaktiviert wird. Weitere Informationen zur Wiederholung von Aufgaben finden Sie unter HealthOmics. Die Aufgabe wird erneut versucht

Im folgenden Beispiel wird die globale Konfiguration so konfiguriert`omicsRetryOn5xx`, dass die Aufgabenwiederholung deaktiviert wird.

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

Das folgende Beispiel zeigt, wie die Konfiguration `omicsRetryOn5xx` im Aufgabenbereich der Workflow-Definition erfolgt.

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

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

Das folgende Beispiel zeigt, wie eine aufgabenspezifische Konfiguration in der Nextflow-Konfigurationsdatei auf der Grundlage der Namens- oder Labelauswahl festgelegt `omicsRetryOn5xx` wird.

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

    withName: 'myTask' {
```

```
        omicsRetryOn5xx = false
    }
}
```

Dauer der Aufgabe unter Verwendung der Direktive **`time`**

HealthOmics stellt ein einstellbares Kontingent bereit (siehe HealthOmics Servicekontingenten), um die maximale Dauer eines Laufs anzugeben. Für Nextflow v23- und v24-Workflows können Sie mithilfe der Nextflow-Direktive auch die maximale Aufgabendauer angeben. `time`

Bei der Entwicklung neuer Workflows hilft Ihnen die Festlegung der maximalen Aufgabendauer dabei, außer catch geratene Aufgaben und lang andauernde Aufgaben zu erkennen.

Weitere Informationen zur Nextflow-Zeitdirektive finden Sie unter Zeitdirektive in der Nextflow-Referenz.

HealthOmics bietet die folgende Unterstützung für die Nextflow-Zeitdirektive:

1. HealthOmics unterstützt eine Granularität von 1 Minute für die Zeitdirektive. Sie können einen Wert zwischen 60 Sekunden und dem Wert für die maximale Laufzeit angeben.
2. Wenn Sie einen Wert unter 60 eingeben, wird HealthOmics dieser auf 60 Sekunden aufgerundet. Bei Werten über 60 wird auf die nächste Minute HealthOmics abgerundet.
3. Wenn der Workflow Wiederholungsversuche für eine Aufgabe unterstützt, versucht er die Aufgabe HealthOmics erneut, wenn das Timeout überschritten wird.
4. Wenn bei einer Aufgabe das Timeout überschritten wird (oder bei der letzten Wiederholung), wird die Aufgabe HealthOmics abgebrochen. Dieser Vorgang kann eine Dauer von ein bis zwei Minuten haben.
5. Bei Zeitüberschreitung der Aufgabe werden die Ausführung und der Aufgabenstatus auf Fehlgeschlagen gesetzt und die anderen Aufgaben in der Ausführung abgebrochen (für Aufgaben mit dem Status „Gestartet“, „Ausstehend“ oder „Wird ausgeführt“). HealthOmics HealthOmics exportiert die Ausgaben von Aufgaben, die vor dem Timeout abgeschlossen wurden, an den von Ihnen angegebenen S3-Ausgabespeicherort.
6. Die Zeit, die eine Aufgabe im Status „Ausstehend“ verbringt, wird nicht auf die Dauer der Aufgabe angerechnet.
7. Wenn die Ausführung Teil einer Ausführungsgruppe ist und das Timeout der Ausführungsgruppe vor Ablauf des Task-Timers abläuft, gehen Ausführung und Task in den Status Fehlgeschlagen über.

Geben Sie die Timeoutdauer mit einer oder mehreren der folgenden Einheiten an:`ms`,`s`, `mh`, oderd.

Das folgende Beispiel zeigt, wie die globale Konfiguration in der Nextflow-Konfigurationsdatei angegeben wird. Es legt ein globales Timeout von 1 Stunde und 30 Minuten fest.

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

Das folgende Beispiel zeigt, wie eine Zeitanweisung im Aufgabenbereich der Workflow-Definition angegeben wird. In diesem Beispiel wird ein Timeout von 3 Tagen, 5 Stunden und 4 Minuten festgelegt. Dieser Wert hat Vorrang vor dem globalen Wert in der Konfigurationsdatei, hat jedoch keinen Vorrang vor einer aufgabenspezifischen Zeitanweisung für `my_label` in der Konfigurationsdatei.

```
process myTask {
    label 'my_label'
    time '3d5h4m'

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

Das folgende Beispiel zeigt, wie aufgabenspezifische Zeitdirektiven in der Nextflow-Konfigurationsdatei auf der Grundlage der Namens- oder Labelselektoren angegeben werden. In diesem Beispiel wird ein globaler Task-Timeout-Wert von 30 Minuten festgelegt. Es legt einen Wert von 2 Stunden für eine Aufgabe `myTask` und einen Wert von 3 Stunden für Aufgaben mit Bezeichnung `my_label` fest. Bei Aufgaben, die dem Selektor entsprechen, haben diese Werte Vorrang vor dem globalen Wert und dem Wert in der Workflow-Definition.

```
process {
    time = '30m'

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

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

```
    }
}
```

Aufgabeninhalt exportieren

Definieren Sie für in Nextflow geschriebene Workflows eine PublishDir-Direktive, um Aufgabeninhalte in Ihren Amazon S3 S3-Ausgabe-Bucket zu exportieren. Wie im folgenden Beispiel gezeigt, setzen Sie den Wert publishDir auf. `/mnt/workflow/pubdir` Um Dateien nach Amazon S3 zu exportieren, müssen sich die Dateien in diesem Verzeichnis befinden.

```
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
  """
}
```

Besonderheiten der CWL-Workflow-Definition

Workflows, die in Common Workflow Language (CWL) geschrieben wurden, bieten ähnliche Funktionen wie Workflows, die in WDL und Nextflow geschrieben wurden. Sie können Amazon S3 oder HealthOmics Storage URIs als Eingabeparameter verwenden.

Wenn Sie die Eingabe in einer SecondaryFile in einem Unter-Workflow definieren, fügen Sie dieselbe Definition im Haupt-Workflow hinzu.

HealthOmics Workflows unterstützen keine Betriebsprozesse. Weitere Informationen zu Betriebsprozessen in CWL-Workflows finden Sie in der CWL-Dokumentation.

Es hat sich bewährt, für jeden Container, den Sie verwenden, einen separaten CWL-Workflow zu definieren. Wir empfehlen, den DockerPull-Eintrag nicht mit einer festen Amazon ECR-URI fest zu codieren.

Themen

- Konvertieren Sie die zu verwendenden CWL-Workflows HealthOmics

- Deaktivieren Sie die Aufgabenwiederholung mit omicsRetryOn5xx
- Einen Workflow-Schritt wiederholen
- Führen Sie Aufgaben mit mehr Arbeitsspeicher erneut aus
- Beispiele

Konvertieren Sie die zu verwendenden CWL-Workflows HealthOmics

Um eine bestehende CWL-Workflow-Definition zur Verwendung zu konvertieren HealthOmics, nehmen Sie die folgenden Änderungen vor:

- Ersetzen Sie alle Docker-Container URIs durch Amazon URIs ECR.
- Stellen Sie sicher, dass alle Workflow-Dateien im Haupt-Workflow als Eingabe deklariert sind und dass alle Variablen explizit definiert sind.
- Stellen Sie sicher, dass der gesamte JavaScript Code Strict-Mode-konform ist.

Deaktivieren Sie die Aufgabenwiederholung mit **`omicsRetryOn5xx`**

HealthOmics unterstützt Aufgabenwiederholungen, wenn die Aufgabe aufgrund von Dienstfehlern fehlgeschlagen ist (5XX-HTTP-Statuscodes). Standardmäßig werden bis zu zwei Wiederholungen einer HealthOmics fehlgeschlagenen Aufgabe versucht. Weitere Hinweise zur Wiederholung von Aufgaben finden Sie unter HealthOmics. Die Aufgabe wird erneut versucht

Um die Wiederholung von Aufgaben bei Servicefehlern zu deaktivieren, konfigurieren Sie die `omicsRetryOn5xx` Anweisung in der Workflow-Definition. Sie können diese Direktive unter Anforderungen oder Hinweisen definieren. Wir empfehlen, die Direktive als Hinweis für die Portabilität hinzuzufügen.

```
requirements:
  ResourceRequirement:
    omicsRetryOn5xx: false

hints:
  ResourceRequirement:
    omicsRetryOn5xx: false
```

Anforderungen haben Vorrang vor Hinweisen. Wenn eine Aufgabenimplementierung einen Ressourcenbedarf in Hinweisen enthält, der auch durch Anforderungen in einem umschließenden Workflow bereitgestellt wird, haben die einschließenden Anforderungen Vorrang.

Wenn dieselbe Aufgabenanforderung auf verschiedenen Ebenen des Workflows vorkommt, wird der spezifischste Eintrag von HealthOmics verwendet `requirements` (oder`hints`, falls es keine Einträge in gibt). `requirements` Die folgende Liste zeigt die Rangfolge, in der die HealthOmics Konfigurationseinstellungen angewendet werden, von der niedrigsten zur höchsten Priorität:

- Workflow-Ebene
- Schrittebene
- Aufgabenbereich der Workflow-Definition

Das folgende Beispiel zeigt, wie die `omicsRetryOn5xx` Direktive auf verschiedenen Ebenen des Workflows konfiguriert wird. In diesem Beispiel hat die Anforderung auf Workflow-Ebene Vorrang vor den Hinweisen auf Workflow-Ebene. Die Anforderungskonfigurationen auf Aufgaben- und Schrittebene haben Vorrang vor den Hinweiskonfigurationen.

```
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
```

Einen Workflow-Schritt wiederholen

HealthOmics unterstützt die Schleife eines Workflow-Schritts. Sie können Schleifen verwenden, um Workflow-Schritte wiederholt auszuführen, bis eine bestimmte Bedingung erfüllt ist. Dies ist nützlich für iterative Prozesse, bei denen Sie eine Aufgabe mehrmals wiederholen müssen oder bis ein bestimmtes Ergebnis erreicht ist.

Hinweis: Für die Loop-Funktionalität ist CWL Version 1.2 oder höher erforderlich. Workflows, die CWL-Versionen vor 1.2 verwenden, unterstützen keine Loop-Operationen.

Um Loops in Ihrem CWL-Workflow zu verwenden, definieren Sie eine Loop-Anforderung. Das folgende Beispiel zeigt die Konfiguration der Loop-Anforderungen:

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

Das `loopWhen` Feld steuert, wann die Schleife endet. In diesem Beispiel wird die Schleife fortgesetzt, solange der Zähler unter dem Maximalwert liegt. Das `loop` Feld definiert, wie Eingabeparameter zwischen den Iterationen aktualisiert werden. Das `loopSource` gibt an, welche Ausgabe der vorherigen Iteration in die nächste Iteration einfließen wird. Das `outputMethod` Feld, das auf gesetzt ist, `last` gibt nur die Ausgabe der letzten Iteration zurück.

Führen Sie Aufgaben mit mehr Arbeitsspeicher erneut aus

HealthOmics unterstützt die automatische Wiederholung fehlgeschlagener out-of-memory Aufgaben. Wenn eine Aufgabe mit dem Code 137 (out-of-memory) beendet HealthOmics wird, wird eine neue Aufgabe mit erhöhter Speicherzuweisung auf der Grundlage des angegebenen Multiplikators erstellt.

> **Note**
>
> HealthOmics wiederholt out-of-memory Fehler bis zu dreimal oder bis die Speicherzuweisung 1536 GiB erreicht, je nachdem, welcher Grenzwert zuerst erreicht wird.

Das folgende Beispiel zeigt, wie Wiederholungen konfiguriert werden: out-of-memory

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

Wenn eine Aufgabe aufgrund von fehlschlägt out-of-memory, HealthOmics berechnet die Speicherzuweisung beim erneuten Versuch anhand der Formel:. `previous_run_memory × memoryRetryMultiplier` Wenn im obigen Beispiel die Aufgabe mit 4096 MB Arbeitsspeicher fehlschlägt, verwendet der Wiederholungsversuch 4096 × 2,5 = 10.240 MB Arbeitsspeicher.

Der `memoryRetryMultiplier` Parameter steuert, wie viel zusätzlicher Speicher für Wiederholungsversuche reserviert werden soll:

- Standardwert: Wenn Sie keinen Wert angeben, wird der Standardwert verwendet 2 (verdoppelt den Speicher)
- Gültiger Bereich: Muss eine positive Zahl größer als sein. 1 Ungültige Werte führen zu einem 4XX-Validierungsfehler
- Effektiver Mindestwert: Werte zwischen 1 und `1.5` werden automatisch erhöht, `1.5` um eine sinnvolle Speichererweiterung sicherzustellen und übermäßige Wiederholungsversuche zu verhindern

Beispiele

Im Folgenden finden Sie ein Beispiel für einen in CWL geschriebenen Workflow.

```
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
```

Die folgende Datei definiert die `copy.cwl` Aufgabe.

```


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)"
```

Im Folgenden finden Sie ein Beispiel für einen in CWL geschriebenen Workflow mit einer GPU-Anforderung.

```
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: []
```

Beispiele für Workflow-Definitionen

Das folgende Beispiel zeigt dieselbe Workflow-Definition in WDL, Nextflow und 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
```

Parametervorlagendateien für HealthOmics Workflows

Parametervorlagen definieren die Eingabeparameter für einen Workflow. Sie können Eingabeparameter definieren, um Ihren Workflow flexibler und vielseitiger zu gestalten. Sie können beispielsweise einen Parameter für den Amazon S3 S3-Speicherort der Referenzgenomdateien definieren. Parametervorlagen können über einen Git-basierten Repository-Dienst oder Ihr lokales

Laufwerk bereitgestellt werden. Benutzer können den Workflow dann mit verschiedenen Datensätzen ausführen.

Sie können die Parametervorlage für Ihren Workflow oder HealthOmics die Parametervorlage für Sie erstellen.

Die Parametervorlage ist eine JSON-Datei. In der Datei ist jeder Eingabeparameter ein benanntes Objekt, das mit dem Namen der Workflow-Eingabe übereinstimmen muss. Wenn Sie einen Lauf starten und nicht Werte für alle erforderlichen Parameter angeben, schlägt der Lauf fehl.

Das Eingabeparameterobjekt umfasst die folgenden Attribute:

- description— Dieses erforderliche Attribut ist eine Zeichenfolge, die von der Konsole auf der Startrun-Seite angezeigt wird. Diese Beschreibung wird auch als Run-Metadaten gespeichert.
- optional— Dieses optionale Attribut gibt an, ob der Eingabeparameter optional ist. Wenn Sie das optional Feld nicht angeben, ist der Eingabeparameter erforderlich.

Die folgende Beispielparametervorlage zeigt, wie die Eingabeparameter angegeben werden.

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

Generieren von Parametervorlagen

HealthOmics generiert die Parametervorlage, indem die Workflow-Definition analysiert wird, um Eingabeparameter zu ermitteln. Wenn Sie eine Parametervorlagendatei für einen Workflow bereitstellen, überschreiben die Parameter in Ihrer Datei die in der Workflow-Definition erkannten Parameter.

Es gibt geringfügige Unterschiede zwischen der Parsing-Logik der CWL-, WDL- und Nextflow-Engines, wie in den folgenden Abschnitten beschrieben.

Themen

Parametererkennung für CWL

In der CWL-Workflow-Engine geht die Parsing-Logik von den folgenden Annahmen aus:

- Alle unterstützten Typen, für die ein Nullwert zulässig ist, sind als optionale Eingabeparameter gekennzeichnet.
- Alle unterstützten Typen, die nicht Null enthalten, werden als erforderliche Eingabeparameter gekennzeichnet.
- Alle Parameter mit Standardwerten sind als optionale Eingabeparameter gekennzeichnet.
- Die Beschreibungen werden aus dem `label` Abschnitt der `main` Workflow-Definition extrahiert. Wenn nicht angegeben, `label` ist die Beschreibung leer (eine leere Zeichenfolge).

Die folgenden Tabellen zeigen Beispiele für die CWL-Interpolation. Für jedes Beispiel lautet der Parametername. `x` Wenn der Parameter erforderlich ist, müssen Sie einen Wert für den Parameter angeben. Wenn der Parameter optional ist, müssen Sie keinen Wert angeben.

Diese Tabelle zeigt Beispiele für die CWL-Interpolation für primitive Typen.

Eingabe	Beispiel für Eingabe/Ausgabe	Erforderlich
`x:` `  type: int`	1 oder 2 oder...	Ja

`x:`
`  type: int`
`  default: 2`
	Der Standardwert ist 2. Gültige Eingabe ist 1 oder 2 oder...	Nein

Eingabe	Beispiel für Eingabe/Ausgabe	Erforderlich
`x:` `  type: int?`	Gültige Eingabe ist Keine oder 1 oder 2 oder...	Nein

`x:`
`  type: int?`
`  default: 2`
	Der Standardwert ist 2. Gültige Eingabe ist Keine oder 1 oder 2 oder...	Nein

Die folgende Tabelle enthält Beispiele für die CWL-Interpolation für komplexe Typen. Ein komplexer Typ ist eine Sammlung primitiver Typen.

Eingabe	Beispiel für Eingabe/Ausgabe	Erforderlich

`x:`
`  type: array`
`  items: int`
	[] oder [1,2,3]	Ja

`x:`
`  type: array?`
`  items: int`
	Keiner oder [] oder [1,2,3]	Nein

`x:`
`  type: array`
`  items: int?`
	[] oder [Keine, 3, Keine]	Ja

`x:`
`  type: array?`
`  items: int?`
	[Keine] oder Keine oder [1,2,3] oder [Keine, 3] aber nicht []	Nein

Parametererkennung für WDL

In der WDL-Workflow-Engine geht die Parsing-Logik von den folgenden Annahmen aus:

- Alle unterstützten Typen, für die NULL-Werte zulässig sind, sind als optionale Eingabeparameter gekennzeichnet.
- Für unterstützte Typen, die keine NULL-Werte zulassen:
  - Jede Eingabevariable mit der Zuweisung von Literalen oder Ausdrücken ist als optionale Parameter gekennzeichnet. Zum Beispiel:

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

  - Wenn den Eingabeparametern keine Werte oder Ausdrücke zugewiesen wurden, werden sie als erforderliche Parameter markiert.
- Beschreibungen werden aus `parameter_meta` der `main` Workflow-Definition extrahiert. Wenn nicht angegeben, `parameter_meta` ist die Beschreibung leer (eine leere Zeichenfolge). Weitere Informationen finden Sie in der WDL-Spezifikation für Parameter-Metadaten.

Die folgenden Tabellen enthalten Beispiele für WDL-Interpolation. Für jedes Beispiel lautet der Parametername. x Wenn der Parameter erforderlich ist, müssen Sie einen Wert für den Parameter angeben. Wenn der Parameter optional ist, müssen Sie keinen Wert angeben.

Diese Tabelle zeigt Beispiele für WDL-Interpolation für primitive Typen.

Eingabe	Beispiel für Eingabe/Ausgabe	Erforderlich
Ganzzahl x	1 oder 2 oder...	Ja
Int x = 2	2	Nein
Ganzzahl x = 1+2	3	Nein
Ganzzahl x = y+z	y+z	Nein
Int? x	Keiner oder 1 oder 2 oder...	Ja
Int? x = 2	Keiner oder 2	Nein
Int? x = 1+2	Keiner oder 3	Nein
Int? x = y+z	Keiner oder y+z	Nein

Die folgende Tabelle zeigt Beispiele für die WDL-Interpolation für komplexe Typen. Ein komplexer Typ ist eine Sammlung primitiver Typen.

Eingabe	Beispiel für Eingabe/Ausgabe	Erforderlich		
Array [Int] x	[1,2,3] oder []	Ja		
Reihe [Int] + x	[1], aber nicht []	Ja		
Array [Int]? x	Keiner oder [] oder [1,2,3]	Nein		
Array [Int?] x	[] oder [Keine, 3, Keine]	Ja		
Array [Int?] =? x	[Keine] oder Keine oder [1,2,3] oder [Keine, 3] aber nicht []	Nein		
Strukturbeispiel {Zeichenfolge a, Int y} später in den Eingaben: Beispiel mySample	`String a = mySample.a Int y = mySample.y`	Ja		
Strukturbeispiel {Zeichenfolge a, Int y} später in den Eingaben:	`if (defined( mySample)) { String a = mySample.a Int y = mySample.y`	Nein		

Eingabe	Beispiel für Eingabe/Ausgabe	Erforderlich		
Beispiel? Meine Probe	}			

Parametererkennung für Nextflow

HealthOmics Generiert für Nextflow die Parametervorlage durch Analysieren der Datei. `nextflow_schema.json` Wenn die Workflow-Definition keine Schemadatei enthält, wird die Haupt-Workflow-Definitionsdatei HealthOmics analysiert.

Themen

- Analysieren der Schemadatei
- Analysieren der Hauptdatei
- Verschachtelte Parameter
- Beispiele für Nextflow-Interpolation

Analysieren der Schemadatei

Damit das Parsen korrekt funktioniert, stellen Sie sicher, dass die Schemadatei die folgenden Anforderungen erfüllt:

- Die Schemadatei hat einen Namen `nextflow_schema.json` und befindet sich in demselben Verzeichnis wie die Haupt-Workflow-Datei.
- Die Schemadatei ist gültiges JSON, wie in einem der folgenden Schemas definiert:
  - Json-Schema. org/draft/2020-12/schema.
  - Json-Schema. org/draft-07/schema.

HealthOmics analysiert die `nextflow_schema.json` Datei, um die Parametervorlage zu generieren:

- Extrahiert allesproperties, was im Schema definiert ist.
- Schließt die Eigenschaft eindescription, sofern sie für die Eigenschaft verfügbar ist.

- Identifiziert, ob jeder Parameter optional oder erforderlich ist, basierend auf dem required Feld der Eigenschaft.

Das folgende Beispiel zeigt eine Definitionsdatei und die generierte Parameterdatei.

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

Die generierte Parametervorlage:

```
{
    "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
    }
}
```

Analysieren der Hauptdatei

Wenn die Workflow-Definition keine `nextflow_schema.json` Datei enthält, wird die Haupt-Workflow-Definitionsdatei HealthOmics analysiert.

HealthOmics analysiert die `params` Ausdrücke, die in der Workflow-Definitionsdatei und in der `nextflow.config` Datei gefunden wurden. Alle `params` mit Standardwerten sind als optional gekennzeichnet.

Beachten Sie die folgenden Anforderungen, damit das Parsen korrekt funktioniert:

- HealthOmics analysiert nur die Haupt-Workflow-Definitionsdatei. Um sicherzustellen, dass alle Parameter erfasst werden, empfehlen wir, eine Verbindung zu allen params Untermodulen und importierten Workflows herzustellen.
- Die Konfigurationsdatei ist optional. Wenn Sie eine definieren, geben Sie ihr einen Namen `nextflow.config` und platzieren Sie sie im selben Verzeichnis wie die Haupt-Workflow-Definitionsdatei.

Das folgende Beispiel zeigt eine Definitionsdatei und die generierte Parametervorlage.

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

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

Die generierte Parametervorlage:

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

HealthOmics Sammelt bei Standardwerten, die in nextflow.config definiert sind, `params` Zuweisungen und Parameter, die darin deklariert sind`params {}`, wie im folgenden Beispiel gezeigt. `params`Muss in Zuweisungsanweisungen auf der linken Seite der Anweisung stehen.

```
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}"
```

Die generierte Parametervorlage:

```
{
    // 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
    }
}
```

Verschachtelte Parameter

Beides `nextflow_schema.json` und `nextflow.config` erlaubt verschachtelte Parameter. Für die HealthOmics Parametervorlage sind jedoch nur die Parameter der obersten Ebene erforderlich. Wenn Ihr Workflow einen verschachtelten Parameter verwendet, müssen Sie ein JSON-Objekt als Eingabe für diesen Parameter angeben.

Verschachtelte Parameter in Schemadateien

HealthOmics überspringt verschachtelte Elemente params beim Parsen einer Datei. `nextflow_schema.json` Wenn Sie beispielsweise die folgende Datei definieren: `nextflow_schema.json`

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

HealthOmics ignoriert `input_file` und `input_num` wenn es die Parametervorlage generiert:

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

Wenn Sie diesen Workflow ausführen, HealthOmics erwartet eine `input.json` Datei, die der folgenden ähnelt:

```
{
```

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

Verschachtelte Parameter in Konfigurationsdateien

HealthOmics sammelt keine params in einer `nextflow.config` Datei verschachtelten Dateien und überspringt sie beim Parsen. Wenn Sie beispielsweise die folgende Datei definieren: `nextflow.config`

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

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

HealthOmics ignoriert `params.nested.beta` und `params.group.delta` wenn es die Parametervorlage generiert:

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

Beispiele für Nextflow-Interpolation

Die folgende Tabelle zeigt Beispiele für die Nextflow-Interpolation für Parameter in der Hauptdatei.

Parameter	Erforderlich
params.input_file	Ja
params.input_file = "s3://bucket/data.json“	Nein
params.nested.input_file	N/A
params.nested.input_file = "s3://bucket/data.json“	N/A

Die folgende Tabelle zeigt Beispiele für die Nextflow-Interpolation für Parameter in der Datei. `nextflow.config`

Parameter	Erforderlich
`params.input_file = "s3://bucket/data.json"`	Nein
`params { input_file = "s3://bucket/data.json" }`	Nein
`params { nested { input_file = "s3://bucket/data.json" } }`	N/A
`input_file = params.input_file`	N/A

Container-Images für private Workflows

HealthOmics unterstützt Container-Images, die in privaten Amazon ECR-Repositorys gehostet werden. Sie können Container-Images erstellen und sie in das private Repository hochladen. Sie können Ihre private Amazon ECR-Registrierung auch als Pull-Through-Cache verwenden, um den Inhalt von Upstream-Registrierungen zu synchronisieren.

Ihr Amazon ECR-Repository muss sich in derselben AWS Region befinden wie das Konto, das den Service aufruft. Eine andere Person AWS-Konto kann Eigentümer des Container-Images sein, sofern das Quell-Image-Repository die entsprechenden Berechtigungen bietet. Weitere Informationen finden Sie unter Richtlinien für den kontenübergreifenden Zugriff auf Amazon ECR.

Wir empfehlen Ihnen, Ihr Amazon ECR-Container-Image URIs als Parameter in Ihrem Workflow zu definieren, damit der Zugriff verifiziert werden kann, bevor der Lauf beginnt. Es macht es auch einfacher, einen Workflow in einer neuen Region auszuführen, indem der Parameter Region geändert wird.

> **Note**
>
> HealthOmics unterstützt keine ARM-Container und unterstützt keinen Zugriff auf öffentliche Repositorys.

Informationen zur Konfiguration von IAM-Berechtigungen für den Zugriff HealthOmics auf Amazon ECR finden Sie unter. HealthOmics Berechtigungen für Ressourcen

Themen

Synchronisieren mit Container-Registrierungen von Drittanbietern

Sie können Amazon ECR Pull-Through-Cache-Regeln verwenden, um Repositorys in einer unterstützten Upstream-Registry mit Ihren privaten Amazon ECR-Repositorys zu synchronisieren.

Weitere Informationen finden Sie unter Synchronisieren einer Upstream-Registrierung im Amazon ECR-Benutzerhandbuch.

Der Pull-Through-Cache erstellt automatisch das Image-Repository in Ihrer privaten Registrierung, wenn Sie den Cache erstellen, und er synchronisiert sich automatisch mit dem zwischengespeicherten Image, wenn Änderungen am Upstream-Image vorgenommen werden.

HealthOmics unterstützt den Pull-Through-Cache für die folgenden Upstream-Registrierungen:

- Amazon ECR Public
- Registrierung für Kubernetes-Container-Images
- Quay
- Docker Hub
- Microsoft Azure Container Registry
- GitHub Container-Registrierung
- GitLab Container-Registrierung

HealthOmics unterstützt keinen Pull-Through-Cache für ein privates Amazon ECR-Upstream-Repository.

Zu den Vorteilen der Verwendung des Amazon ECR Pull-Through-Cache gehören:

1. Sie müssen Container-Images nicht manuell zu Amazon ECR migrieren oder Updates aus dem Drittanbieter-Repository synchronisieren.
2. Workflows greifen auf die synchronisierten Container-Images in Ihrem privaten Repository zu. Dies ist zuverlässiger als das Herunterladen von Inhalten zur Laufzeit aus einer öffentlichen Registrierung.
3. Da Amazon ECR Pull-Through-Caches eine vorhersehbare URI-Struktur verwenden, kann der HealthOmics Service die private Amazon ECR-URI automatisch der Upstream-Registrierungs-URI zuordnen. Sie müssen die URI-Werte in der Workflow-Definition nicht aktualisieren und ersetzen.

Themen

Pull-Through-Cache konfigurieren

Amazon ECR bietet eine Registrierung für Sie AWS-Konto in jeder Region. Stellen Sie sicher, dass Sie die Amazon ECR-Konfiguration in derselben Region erstellen, in der Sie den Workflow ausführen möchten.

In den folgenden Abschnitten werden die Konfigurationsaufgaben für den Pull-Through-Cache beschrieben.

Aufgaben zur Konfiguration

- Erstellen Sie eine Pull-Through-Cache-Regel
- Registrierungsberechtigungen für die Upstream-Registrierung
- Vorlagen für die Erstellung von Reposit
- Den Workflow erstellen

Erstellen Sie eine Pull-Through-Cache-Regel

Erstellen Sie eine Amazon ECR-Pull-Through-Cache-Regel für jede Upstream-Registrierung, die Bilder enthält, die Sie zwischenspeichern möchten. Eine Regel spezifiziert eine Zuordnung zwischen einer Upstream-Registrierung und dem privaten Amazon ECR-Repository.

Für eine Upstream-Registrierung, die eine Authentifizierung erfordert, geben Sie Ihre Anmeldeinformationen mithilfe von AWS Secrets Manager ein.

> **Note**
>
> Ändern Sie eine Pull-Through-Cache-Regel nicht, während ein aktiver Lauf das private Repository verwendet. Der Lauf könnte fehlschlagen oder, was noch wichtiger ist, dazu führen, dass Ihre Pipeline unerwartete Bilder verwendet.

Weitere Informationen finden Sie unter Erstellen einer Pull-Through-Cache-Regel im Amazon Elastic Container Registry-Benutzerhandbuch.

Erstellen Sie mithilfe der Konsole eine Pull-Through-Cache-Regel

Gehen Sie mit der Amazon ECR-Konsole wie folgt vor, um den Pull-Through-Cache zu konfigurieren:

1. Öffnen Sie die Amazon ECR-Konsole: https://console.aws.amazon.com /ecr

2. Erweitern Sie im linken Menü unter Private Registrierung die Option Funktionen und Einstellungen. Wählen Sie dann Pull through Cache aus.
3. Wählen Sie auf der Seite Pull-Through-Cache die Option Regel hinzufügen aus.
4. Wählen Sie im Bereich Upstream-Registrierung die Upstream-Registrierung aus, die mit Ihrer privaten Registrierung synchronisiert werden soll, und klicken Sie dann auf Weiter.
5. Wenn für die Upstream-Registrierung eine Authentifizierung erforderlich ist, öffnet die Konsole eine neue Seite, auf der Sie das SageMaker KI-Geheimnis angeben, das Ihre Anmeldeinformationen enthält. Wählen Sie Weiter aus.
6. Wählen Sie unter Namespaces angeben im Bereich Cache-Namespace aus, ob die privaten Repositorys mit einem bestimmten Repository-Präfix oder ohne Präfix erstellt werden sollen. Wenn Sie ein Präfix verwenden möchten, geben Sie den Präfixnamen im Feld Cache-Repository-Präfix an.
7. Wählen Sie im Bereich Upstream-Namespace aus, ob aus Upstream-Repositorys mit einem bestimmten Repository-Präfix oder ohne Präfix abgerufen werden soll. Wenn Sie ein Präfix verwenden möchten, geben Sie den Präfixnamen im Feld Upstream-Repository-Präfix an.

   Das Namespace-Beispielfenster zeigt ein Beispiel für eine Pull-Anfrage, eine Upstream-URL und die URL des Cache-Repositorys, das erstellt wird.
8. Wählen Sie Weiter aus.
9. Überprüfen Sie die Konfiguration und wählen Sie Create, um die Regel zu erstellen.

Weitere Informationen finden Sie unter Eine Pull-Through-Cache-Regel erstellen (AWS Management Console).

Erstellen Sie mit der CLI eine Pull-Through-Cache-Regel

Verwenden Sie den Amazon create-pull-through-cache-rule ECR-Befehl, um eine Pull-Through-Cache-Regel zu erstellen. Für Upstream-Registrierungen, die eine Authentifizierung erfordern, speichern Sie die Anmeldeinformationen in einem Secrets Manager Manager-Geheimnis.

Die folgenden Abschnitte enthalten Beispiele für jede unterstützte Upstream-Registrierung.

Für Amazon ECR Public

Im folgenden Beispiel wird eine Pull-Through-Cache-Regel für die öffentliche Registrierung von Amazon ECR erstellt. Es gibt ein Repository-Präfix von `ecr-public`, was dazu führt, dass jedes

Repository, das mit der Pull-Through-Cache-Regel erstellt wurde, das Benennungsschema von `ecr-public/`*`upstream-repository-name`* hat.

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

Für Kubernetes Container Registry

Im folgenden Beispiel wird eine Pull-Through-Cache-Regel für das öffentliche Kubernetes-Registry erstellt. Es gibt ein Repository-Präfix von `kubernetes`, was dazu führt, dass jedes Repository, das mit der Pull-Through-Cache-Regel erstellt wurde, das Benennungsschema von `kubernetes/`*`upstream-repository-name`* hat.

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

Für Quay

Im folgenden Beispiel wird eine Pull-Through-Cache-Regel für die öffentliche Registrierung von Quay erstellt. Es gibt ein Repository-Präfix von `quay`, was dazu führt, dass jedes Repository, das mit der Pull-Through-Cache-Regel erstellt wurde, das Benennungsschema von `quay/`*`upstream-repository-name`* hat.

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

Für Docker Hub

Im folgenden Beispiel wird eine Pull-Through-Cache-Regel für die Docker-Hub-Registrierung von Quay erstellt. Es gibt ein Repository-Präfix von `docker-hub`, was dazu führt, dass jedes Repository, das mit der Pull-Through-Cache-Regel erstellt wurde, das Benennungsschema von `docker-hub/`*`upstream-repository-name`* hat. Sie müssen den vollständigen Amazon-Ressourcennamen (ARN) des Secrets mit Ihren Anmeldeinformationen für Docker Hub angeben.

```
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
```

Für Container Registry GitHub

Im folgenden Beispiel wird eine Pull-Through-Cacheregel für die GitHub Container Registry erstellt. Es gibt ein Repository-Präfix von `github`, was dazu führt, dass jedes Repository, das mit der Pull-Through-Cache-Regel erstellt wurde, das Benennungsschema von `github/`*`upstream-repository-name` hat. Sie müssen den vollständigen Amazon-Ressourcennamen (ARN) des Geheimnisses angeben, das Ihre Anmeldeinformationen für die GitHub Container Registry enthält.

```
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
```

Für Microsoft Azure Container Registry

Im folgenden Beispiel wird eine Pull-Through-Cacheregel für die Microsoft Azure Container Registry erstellt. Es gibt ein Repository-Präfix von `azure`, was dazu führt, dass jedes Repository, das mit der Pull-Through-Cache-Regel erstellt wurde, das Benennungsschema von `azure/`*`upstream-repository-name` hat. Sie müssen den vollständigen Amazon-Ressourcennamen (ARN) des Secrets mit Ihren Anmeldeinformationen für die Microsoft Azure Container Registry angeben.

```
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
```

Für GitLab Container Registry

Im folgenden Beispiel wird eine Pull-Through-Cacheregel für die GitLab Container Registry erstellt. Es gibt ein Repository-Präfix von `gitlab`, was dazu führt, dass jedes Repository, das mit der

Pull-Through-Cache-Regel erstellt wurde, das Benennungsschema von `gitlab/`*`upstream-repository-name`* hat. Sie müssen den vollständigen Amazon-Ressourcennamen (ARN) des Geheimnisses angeben, das Ihre Anmeldeinformationen für die GitLab Container Registry enthält.

```
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
```

Weitere Informationen finden Sie unter Erstellen einer Pull-Through-Cache-Regel (CLI) im Amazon ECR-Benutzerhandbuch.

Sie können den get-run-task CLI-Befehl verwenden, um Informationen über das Container-Image abzurufen, das für eine bestimmte Aufgabe verwendet wird:

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

Die Ausgabe enthält die folgenden Informationen über das Container-Image:

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

Registrierungsberechtigungen für die Upstream-Registrierung

Verwenden Sie Registrierungsberechtigungen, um die Verwendung des Pull-Through-Cache und das Abrufen der Container-Images in die private Amazon ECR-Registrierung zu ermöglichen HealthOmics . Fügen Sie der Registrierung eine Amazon ECR-Registrierungsrichtlinie hinzu, die die in Läufen verwendeten Container bereitstellt.

Die folgende Richtlinie erteilt dem HealthOmics Service die Erlaubnis, Repositorys mit den angegebenen Pull-Through-Cache-Präfixen zu erstellen und Upstream-Pulls in diese Repositorys zu initiieren.

1. Öffnen Sie in der Amazon ECR-Konsole das linke Menü, erweitern Sie unter Private Registrierung den Eintrag Registrierungsberechtigungen. Wählen Sie dann Kontoauszug generieren aus.
2. Wählen Sie oben rechts JSON aus. Geben Sie eine Richtlinie ein, die der folgenden ähnelt:

   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/*"
            ]
        }
    ]
}
```

Vorlagen für die Erstellung von Reposit

Um das Pull-Through-Caching in verwenden zu können HealthOmics, muss das Amazon ECR-Repository über eine Vorlage für die Repository-Erstellung verfügen. Die Vorlage definiert Konfigurationseinstellungen für den Fall, dass Sie oder Amazon ECR ein privates Repository für eine Upstream-Registrierung erstellen.

Jede Vorlage enthält ein Repository-Namespace-Präfix, das Amazon ECR verwendet, um neue Repositorys einer bestimmten Vorlage zuzuordnen. Vorlagen spezifizieren die Konfiguration für alle Repository-Einstellungen, einschließlich ressourcenbasierter Zugriffsrichtlinien, Tag-Unveränderlichkeit, Verschlüsselung und Lebenszyklus-Richtlinien.

Weitere Informationen finden Sie unter Vorlagen für die Erstellung von Repositorys im Amazon Elastic Container Registry User Guide.

So erstellen Sie eine Vorlage für die Erstellung eines Repositorys:

1. Öffnen Sie in der Amazon ECR-Konsole das linke Menü unter Private Registrierung, erweitern Sie Funktionen und Einstellungen. Wählen Sie dann Vorlagen zur Erstellung von Repositorys aus.
2. Wählen Sie Create template (Vorlage erstellen) aus.
3. Wählen Sie in den Vorlagendetails die Option Pull through cache aus.
4. Wählen Sie aus, ob diese Vorlage auf ein bestimmtes Präfix oder auf alle Repositorys angewendet werden soll, die keiner anderen Vorlage entsprechen.

   Wenn Sie Ein bestimmtes Präfix wählen, geben Sie den Namespace-Präfixwert im Feld Präfix ein. Sie haben dieses Präfix bei der Erstellung der PTC-Regel angegeben.
5. Wählen Sie Weiter aus.
6. Geben Sie auf der Seite Konfiguration zur Repository-Erstellung hinzufügen die Repository-Berechtigungen ein. Verwenden Sie eine der Beispiel-Richtlinienanweisungen oder geben Sie eine ein, die dem folgenden Beispiel ähnelt:

   JSON

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

7. Optional können Sie Repository-Einstellungen wie Lebenszyklusrichtlinien und Tags hinzufügen. Amazon ECR wendet diese Regeln auf alle Container-Images an, die für den Pull-Through-Cache erstellt wurden und das angegebene Präfix verwenden.
8. Wählen Sie Weiter aus.
9. Überprüfen Sie die Konfiguration und wählen Sie Weiter.

Den Workflow erstellen

Wenn Sie einen neuen Workflow oder eine neue Workflow-Version erstellen, überprüfen Sie die Registrierungszuordnungen und aktualisieren Sie sie bei Bedarf. Details hierzu finden Sie unter Erstellen Sie einen privaten Workflow.

Registrierungszuordnungen

Sie definieren Registrierungszuordnungen, um Präfixe in Ihrer privaten Amazon ECR-Registrierung und den Namen der Upstream-Registrierung zuzuordnen.

Weitere Informationen zu Amazon ECR-Registrierungszuordnungen finden Sie unter Erstellen einer Pull-Through-Cache-Regel in Amazon ECR.

Das folgende Beispiel zeigt Registrierungszuordnungen zu Docker Hub, Quay und Amazon ECR Public.

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

Bildzuordnungen

Sie definieren Bildzuordnungen, die zwischen den Bildnamen, wie sie in Ihren privaten Amazon ECR-Workflows definiert sind, und den Bildnamen in der Upstream-Registrierung zugeordnet werden.

Sie können Image-Zuordnungen mit Registern verwenden, die den Pull-Through-Cache unterstützen. Sie können Image-Zuordnungen auch für Upstream-Registrierungen verwenden, die den Pull-Through-Cache HealthOmics nicht unterstützen. Sie müssen die Upstream-Registrierung manuell mit Ihrem privaten Repository synchronisieren.

Weitere Informationen zu Amazon ECR-Image-Zuordnungen finden Sie unter Erstellen einer Pull-Through-Cache-Regel in Amazon ECR.

Das folgende Beispiel zeigt Zuordnungen von privaten Amazon ECR-Images zu einem öffentlichen Genomik-Image und dem neuesten Ubuntu-Image.

```
{
    "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",
        }
    ]
}
```

Allgemeine Überlegungen zu Amazon ECR-Container-Images

- Architektur

  HealthOmics unterstützt x86_64-Container. Wenn Ihr lokaler Computer ARM-basiert ist, z. B. Apple Mac, verwenden Sie einen Befehl wie den folgenden, um ein x86_64-Container-Image zu erstellen:

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

- Entrypoint und Shell

HealthOmics Workflow-Engines fügen Bash-Skripten als Befehlsüberschreibung in die Container-Images ein, die von Workflow-Aufgaben verwendet werden. Daher sollten Container-Images ohne einen angegebenen ENTRYPOINT erstellt werden, sodass eine Bash-Shell die Standardeinstellung ist.

- Bereitgestellte Pfade

  Ein gemeinsam genutztes Dateisystem wird in Container-Aufgaben unter /tmp eingehängt. Alle Daten oder Tools, die an diesem Ort in das Container-Image integriert sind, werden überschrieben.

  Die Workflow-Definition ist für Aufgaben über einen schreibgeschützten Mount unter /mnt/workflow verfügbar.

- Größe des Images

  Die maximalen Bildgrößen von Containern finden Sie unter[HealthOmics Workflow-Kontingente mit fester Größe](#).

Umgebungsvariablen für HealthOmics Workflows

HealthOmics stellt Umgebungsvariablen bereit, die Informationen über den Workflow enthalten, der im Container ausgeführt wird. Sie können die Werte dieser Variablen in der Logik Ihrer Workflow-Aufgaben verwenden.

Alle HealthOmics Workflow-Variablen beginnen mit dem `AWS_WORKFLOW_` Präfix. Dieses Präfix ist ein geschütztes Umgebungsvariablenpräfix. Verwenden Sie dieses Präfix nicht für Ihre eigenen Variablen in Workflow-Containern.

HealthOmics stellt die folgenden Workflow-Umgebungsvariablen bereit:

AWS_REGION

Diese Variable ist die Region, in der der Container ausgeführt wird.

AWS_WORKFLOW_AUSFÜHREN

Diese Variable ist der Name des aktuellen Laufs.

AWS_WORKFLOW_RUN_ID

Diese Variable ist die Lauf-ID des aktuellen Laufs.

AWS_WORKFLOW_RUN_UUID

Diese Variable ist die Run-UUID des aktuellen Laufs.

AWS_WORKFLOW_AUFGABE

Diese Variable ist der Name der aktuellen Aufgabe.

AWS_WORKFLOW_TASK_ID

Diese Variable ist die Aufgaben-ID der aktuellen Aufgabe.

AWS_WORKFLOW_TASK_UUID

Diese Variable ist die Task-UUID der aktuellen Aufgabe.

Das folgende Beispiel zeigt typische Werte für jede Umgebungsvariable:

```
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
```

Verwenden von Java in Amazon ECR-Container-Images

Wenn eine Workflow-Aufgabe eine Java-Anwendung wie GATK verwendet, sollten Sie die folgenden Speicheranforderungen für den Container berücksichtigen:

- Java-Anwendungen verwenden Stack-Speicher und Heap-Speicher. Standardmäßig ist der maximale Heap-Speicher ein Prozentsatz des gesamten verfügbaren Speichers im Container. Dieser Standard hängt von der jeweiligen JVM-Distribution und der JVM-Version ab. Konsultieren Sie daher die entsprechende Dokumentation für Ihre JVM oder legen Sie das maximale Heap-Speicherlimit explizit mithilfe von Java-Befehlszeilenoptionen (wie `-Xmx`) fest.
- Legen Sie den maximalen Heap-Speicher nicht auf 100% der Speicherzuweisung des Containers fest, da der JVM-Stack auch Speicher benötigt. Speicher ist auch für den JVM-Garbage-Collector und alle anderen Betriebssystemprozesse erforderlich, die im Container ausgeführt werden.
- Einige Java-Anwendungen, wie GATK, können native Methodenaufrufe oder andere Optimierungen wie Speicherzuordnungsdateien verwenden. Diese Techniken erfordern

Speicherzuweisungen, die „außerhalb des Heaps“ erfolgen und nicht durch den JVM-Parameter für maximale Heap-Anzahl gesteuert werden.

Wenn Sie wissen (oder vermuten), dass Ihre Java-Anwendung Off-Heap-Speicher zuweist, stellen Sie sicher, dass Ihre Aufgabenspeicherzuweisung die Anforderungen an Off-Heap-Speicher berücksichtigt.

Wenn diese Off-Heap-Zuweisungen dazu führen, dass dem Container nicht mehr genügend Speicher zur Verfügung steht, wird normalerweise kein OutOfMemory Java-Fehler angezeigt, da die JVM diesen Speicher nicht kontrolliert.

Hinzufügen von Aufgabeneingaben zu einem Amazon ECR-Container-Image

Fügen Sie alle ausführbaren Dateien, Bibliotheken und Skripten, die für die Ausführung einer Workflow-Aufgabe erforderlich sind, in das Amazon ECR-Image ein, das für die Ausführung der Aufgabe verwendet wird.

Es hat sich bewährt, die Verwendung von Skripten, Binärdateien und Bibliotheken zu vermeiden, die sich außerhalb eines Task-Container-Images befinden. Dies ist besonders wichtig, wenn `nf-core` Workflows verwendet werden, die ein `bin` Verzeichnis als Teil des Workflow-Pakets verwenden. Dieses Verzeichnis wird zwar für die Workflow-Aufgabe verfügbar sein, es ist jedoch als schreibgeschütztes Verzeichnis bereitgestellt. Erforderliche Ressourcen in diesem Verzeichnis sollten in das Task-Image kopiert und zur Laufzeit oder beim Erstellen des für die Aufgabe verwendeten Container-Images verfügbar gemacht werden.

Die maximale Größe des HealthOmics unterstützten Container-Images finden HealthOmics Workflow-Kontingente mit fester Größe Sie unter.

HealthOmics Workflow-README-Dateien

Sie können eine README.md-Datei hochladen, die Anweisungen, Diagramme und wichtige Informationen für Ihren Arbeitsablauf enthält. Jede Workflow-Version unterstützt eine README-Datei, die Sie jederzeit aktualisieren können.

Zu den README-Anforderungen gehören:

- Die README-Datei muss im Markdown-Format (.md) vorliegen
- Maximale Dateigröße: 500 KiB

Themen

Verwenden Sie eine vorhandene README-Datei

READMEs Aus Git-Repositorys exportierte Dateien enthalten relative Links, die normalerweise außerhalb des Repositorys nicht funktionieren. HealthOmics Die Git-Integration konvertiert diese automatisch in absolute Links für das korrekte Rendern in der Konsole, sodass keine manuellen URL-Updates erforderlich sind.

Bei READMEs Importen aus Amazon S3 oder lokalen Laufwerken müssen Bilder und Links entweder öffentlich verwendet werden URLs oder ihre relativen Pfade müssen aktualisiert werden, damit sie korrekt gerendert werden.

> **Note**
>
> Bilder müssen öffentlich gehostet werden, damit sie in der HealthOmics Konsole angezeigt werden können. Bilder, die in GitHub Enterprise Server oder GitLab Self-Managed Repositorys gespeichert sind, können nicht gerendert werden.

Bedingungen für das Rendern

Die HealthOmics Konsole interpoliert öffentlich zugängliche Bilder und Links mithilfe absoluter Pfade. Um URLs aus privaten Repositorys rendern zu können, muss der Benutzer Zugriff auf das Repository haben. Für GitHub Enterprise Server oder GitLab Self-Managed Repositorys, die benutzerdefinierte Domänen verwenden, können relative Links HealthOmics nicht aufgelöst oder Bilder gerendert werden, die in diesen privaten Repositorys gespeichert sind.

Die folgende Tabelle zeigt die Markdown-Elemente, die von der README-Ansicht der AWS Konsole unterstützt werden.

Element	AWS Konsole
Benachrichtigungen	Ja, aber ohne Icons

Element	AWS Konsole
Abzeichen	Ja
Grundlegende Textformatierung	Ja
Codeblöcke	Ja, hat aber keine Syntaxhervorhebungs - und Kopierschaltflächenfunktionen
Zusammenklappbare Abschnitte	Ja
Überschriften	Ja
Bildformate	Ja
Bild (anklickbar)	Ja
Zeilenumbrüche	Ja
Meerjungfrau-Diagramm	Kann nur das Diagramm öffnen, die Position des Diagramms verschieben und Code kopieren
Angebote	Ja
Tiefgestellt und hochgestellt	Ja
Tabellen	Ja, unterstützt aber keine Textausrichtung
Textausrichtung	Ja

Bild und Link werden verwendet URLs

Strukturieren Sie je nach Quellenanbieter Ihre Basis URLs für Seiten und Bilder in den folgenden Formaten.

- `{username}`: Der Benutzername, unter dem das Repository gehostet wird.
- `{repo}`: Der Name des Repositorys.
- `{ref}`: Die Quellreferenz (Branch, Tag und Commit-ID).
- `{path}`: Der Dateipfad zur Seite oder zum Bild im Repository.

Quellanbieter	Seiten-URL	Bild-URL

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

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

GitHubGitLab, und Bitbucket unterstützt sowohl Seiten als auch Bilder URLs , die auf ein öffentliches Repository verweisen. Die folgende Tabelle zeigt, wie die einzelnen Quellenanbieter das Rendern von Bildern und Links URLs für private Repositorys unterstützen.

Unterstützung für private Repositorys		
Quellenanbieter	Seiten-URL	Bild-URL
GitHub	Nur mit Zugriff auf das Repository	Nein
GitLab	Nur mit Zugriff auf das Repository	Nein
Bitbucket	Nur mit Zugriff auf das Repository	Nein

Sentieon-Lizenzen für private Workflows anfordern

Wenn Ihr privater Workflow die Sentieon-Software verwendet, benötigen Sie eine Senieon-Lizenz. Gehen Sie wie folgt vor, um eine Lizenz für die Sentieon-Software anzufordern und einzurichten:

- Fordern Sie eine Sentieon-Lizenz an
  - Senden Sie eine E-Mail an die Sentieon-Supportgruppe (support@sentieon.com), um eine Softwarelizenz anzufordern.
    - Geben Sie in der AWS E-Mail Ihre Canonical Benutzer-ID an.
    - Finden Sie Ihre AWS Canonical-Benutzer-ID, indem Sie diesen Anweisungen folgen.
- Aktualisieren Sie Ihre HealthOmics Servicerolle, um ihr Zugriff auf den Sentieon-Lizenzserver-Proxy und den Sentieon Omics Bucket in Ihrer Region zu gewähren. Das folgende Beispiel gewährt Zugriff auf. `us-east-1` Falls erforderlich, ersetzen Sie diesen Text durch Ihre Region.

  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/*"
            ]
        }
    ]
 }
```

- Generieren Sie eine AWS Support-Anfrage, um Zugriff auf den Sentieon-Lizenzserver-Proxy zu erhalten.
  - Um einen Support-Fall zu erstellen, navigieren Sie zu support.console.aws.amazon.com.
  - Geben Sie im Support-Fall Ihre Region und Ihre Region an AWS-Konto . Ihr Konto wurde der Zulassungsliste für den Lizenzserver-Proxy hinzugefügt.

- Erstellen Sie Ihren privaten Workflow mithilfe des Sentieon-Containers und des Sentieon-Lizenzskripts.
  - Weitere Anweisungen zur Verwendung von Sentieon-Tools in privaten Workflows finden Sie unter Sentieon-Amazon-Omics unter. GitHub
- Die Sentieon-Softwareversion 202112.07 und höher unterstützen den Lizenzserver-Proxy. HealthOmics Um Sentieon-Softwareversionen vor 202112.07 zu verwenden, wenden Sie sich an den Sentieon-Support.

Der Arbeitsablauf blinkt ein HealthOmics

Nachdem Sie einen Workflow erstellt haben, empfehlen wir, dass Sie einen Linter für den Workflow ausführen, bevor Sie mit der ersten Ausführung beginnen. Der Linter erkennt Fehler, die dazu führen können, dass der Lauf fehlschlägt.

Bei WDL HealthOmics wird automatisch ein Linter ausgeführt, wenn Sie den Workflow erstellen. Die Linterausgabe ist im `statusMessage` Feld der Antwort verfügbar. get-workflow Verwenden Sie den folgenden CLI-Befehl, um die Statusausgabe abzurufen (verwenden Sie die Workflow-ID des WDL-Workflows, den Sie erstellt haben):

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

HealthOmics stellt Links bereit, die Sie auf der Workflow-Definition ausführen können, bevor Sie den Workflow erstellen. Führen Sie diese Linter auf vorhandenen Pipelines aus, zu denen Sie migrieren. HealthOmics

- WDL— Ein öffentliches Amazon ECR-Image zum Ausführen eines WDL-Linters.
- Nextflow— Ein öffentliches Amazon ECR-Image zur Ausführung von Linter-Regeln für Nextflow. Sie können auf den Quellcode für diesen Linter von zugreifen. GitHub
- CWL— nicht verfügbar

HealthOmics Workflow-Operationen

Um einen privaten Workflow zu erstellen, benötigen Sie:

- Workflow definition file:Eine inWDL, Nextflow oder geschriebene Workflow-DefinitionsdateiCWL. Die Workflow-Definition spezifiziert die Eingaben und Ausgaben für Läufe, die den Workflow verwenden. Sie enthält auch Spezifikationen für die Ausführungen und Ausführungsaufgaben für Ihren Workflow, einschließlich der Rechen- und Speicheranforderungen. Die Workflow-Definitionsdatei muss `.zip` das Format haben. Weitere Informationen finden Sie unter Workflow-Definitionsdateien in HealthOmics.
  - Sie können Amazon Q CLI verwenden, um Ihre Workflow-Definitionsdateien in WDL, Nextflow und CWL zu erstellen und zu validieren. Weitere Informationen finden Sie unter Beispielaufforderungen für Amazon Q CLI und im HealthOmics Agentic Generative AI-Tutorial unter. GitHub
- (Optional) Parameter template file:Eine Parameter-Vorlagendatei, die in geschrieben wurde. JSON Erstellen Sie die Datei, um die Ausführungsparameter zu definieren, oder HealthOmics generiert die Parametervorlage für Sie. Weitere Informationen finden Sie unter Parametervorlagendateien für HealthOmics Workflows.
- Amazon ECR container images:Erstellen Sie private Amazon ECR-Repositorys für jeden Container, der im Workflow verwendet wird. Erstellen Sie Container-Images für den Workflow und speichern Sie sie in einem privaten Repository, oder synchronisieren Sie den Inhalt einer unterstützten Upstream-Registrierung mit Ihrem privaten ECR-Repository.
- (Optional) Sentieon licenses:Fordern Sie eine Sentieon Lizenz an, um die Sentieon Software in privaten Workflows zu verwenden.

Für Workflow-Definitionsdateien, die größer als 4 MiB (gezippt) sind, wählen Sie bei der Workflow-Erstellung eine der folgenden Optionen:

- Laden Sie in einen Amazon Simple Storage Service-Ordner hoch und geben Sie den Speicherort an.
- Laden Sie in ein externes Repository hoch (maximale Größe 1 GiB) und geben Sie die Repository-Details an.

Nachdem Sie einen Workflow erstellt haben, können Sie die folgenden Workflow-Informationen mit dem `UpdateWorkflow` Vorgang aktualisieren:

- Name
- Description
- Standard-Speichertyp

- Standardspeicherkapazität (mit Workflow-ID)
- README.md-Datei

Um andere Informationen im Workflow zu ändern, erstellen Sie einen neuen Workflow oder eine neue Workflow-Version.

Verwenden Sie die Workflow-Versionierung, um Ihre Workflows zu organisieren und zu strukturieren. Versionen helfen Ihnen auch dabei, die Einführung iterativer Workflow-Updates zu verwalten. Weitere Informationen zu Versionen erhalten Sie unter Workflow-Version erstellen.

Themen

Erstellen Sie einen privaten Workflow

Erstellen Sie einen Workflow mit der HealthOmics Konsole, AWS CLI-Befehlen oder einem der AWS SDKs.

> **Note**
>
> Nehmen Sie keine persönlich identifizierbaren Informationen (PII) in Workflow-Namen auf. Diese Namen sind in CloudWatch Protokollen sichtbar.

Wenn Sie einen Workflow erstellen, HealthOmics weist er dem Workflow einen Universally Unique Identifier (UUID) zu. Die Workflow-UUID ist eine GUID (Globally Unique Identifier), die für alle Workflows und Workflow-Versionen eindeutig ist. Aus Gründen der Datenherkunft empfehlen wir, die Workflow-UUID zu verwenden, um Workflows eindeutig zu identifizieren.

Wenn Ihre Workflow-Aufgaben externe Tools (ausführbare Dateien, Bibliotheken oder Skripts) verwenden, bauen Sie diese Tools in ein Container-Image ein. Sie haben die folgenden Optionen für das Hosten des Container-Images:

- Hosten Sie das Container-Image in der privaten ECR-Registrierung. Voraussetzungen für diese Option:
  - Erstellen Sie ein privates ECR-Repository oder wählen Sie ein vorhandenes Repository aus.
  - Konfigurieren Sie die ECR-Ressourcenrichtlinie wie unter beschrieben. Amazon-ECR-Berechtigungen
  - Laden Sie Ihr Container-Image in das private Repository hoch.
- Synchronisieren Sie das Container-Image mit dem Inhalt einer unterstützten Registrierung eines Drittanbieters. Voraussetzungen für diese Option:
  - Konfigurieren Sie in der privaten ECR-Registrierung eine Pull-Through-Cache-Regel für jede Upstream-Registrierung. Weitere Informationen finden Sie unter Bildzuordnungen.
  - Konfigurieren Sie die ECR-Ressourcenrichtlinie wie unter beschrieben. Amazon-ECR-Berechtigungen
  - Erstellen Sie Vorlagen für die Repository-Erstellung. Die Vorlage definiert Einstellungen für den Zeitpunkt, zu dem Amazon ECR das private Repository für eine Upstream-Registrierung erstellt.
  - Erstellen Sie Präfixzuordnungen, um Container-Image-Referenzen in der Workflow-Definition den ECR-Cache-Namespaces neu zuzuordnen.

Wenn Sie einen Workflow erstellen, geben Sie eine Workflow-Definition an, die Informationen über den Workflow, die Ausführungen und die Aufgaben enthält. HealthOmics kann die Workflow-Definition als lokal oder in einem Amazon S3 S3-Bucket gespeichertes ZIP-Archiv oder aus einem unterstützten Git-basierten Repository abrufen.

Themen

- Einen Workflow mithilfe der Konsole erstellen
- Einen Workflow mit der CLI erstellen
- Einen Workflow mithilfe eines SDK erstellen

Einen Workflow mithilfe der Konsole erstellen

Schritte zum Erstellen eines Workflows

1. Öffnen Sie die HealthOmics -Konsole.
2. Öffnen Sie bei Bedarf den linken Navigationsbereich (±1). Wählen Sie Private Workflows aus.
3. Wählen Sie auf der Seite Private Workflows die Option Workflow erstellen aus.

4. Geben Sie auf der Seite Workflow definieren die folgenden Informationen ein:

   1. Workflow-Name: Ein eindeutiger Name für diesen Workflow. Wir empfehlen, Workflow-Namen festzulegen, um Ihre Läufe in der AWS HealthOmics Konsole und in den CloudWatch Protokollen zu organisieren.
   2. Beschreibung (optional): Eine Beschreibung dieses Workflows.

5. Geben Sie im Bereich Workflow-Definition die folgenden Informationen ein:

   1. Workflow-Sprache (optional): Wählen Sie die Spezifikationssprache des Workflows aus. Andernfalls HealthOmics bestimmt die Sprache anhand der Workflow-Definition.
   2. Wählen Sie unter Workflow-Definitionsquelle, ob Sie den Definitionsordner aus einem Git-basierten Repository, einem Amazon S3 S3-Speicherort oder von einem lokalen Laufwerk importieren möchten.

      a. Für den Import aus einem Repository-Service:

      > **Note**
      >
      > HealthOmics unterstützt öffentliche und private Repositorys fürGitHub,GitLab,, BitbucketGitHub self-managed,GitLab self-managed.

      i. Wählen Sie eine Verbindung, um Ihre AWS Ressourcen mit dem externen Repository zu verbinden. Informationen zum Herstellen einer Verbindung finden Sie unterConnect mit externen Code-Repositorys her.

      > **Note**
      >
      > Kunden in der TLV Region müssen eine Verbindung in der Region IAD (us-east-1) herstellen, um einen Workflow zu erstellen.

      ii. Geben Sie unter Vollständige Repository-ID Ihre Repository-ID als Benutzername/Repository-Name ein. Stellen Sie sicher, dass Sie Zugriff auf die Dateien in diesem Repository haben.

      iii. Geben Sie im Feld Quellverweis (optional) eine Repository-Quellenreferenz ein (Branch-, Tag- oder Commit-ID). HealthOmics verwendet den Standardzweig, wenn kein Quellverweis angegeben ist.

iv. Geben Sie im Feld Dateimuster ausschließen die Dateimuster ein, um bestimmte Ordner, Dateien oder Erweiterungen auszuschließen. Dies hilft bei der Verwaltung der Datengröße beim Import von Repository-Dateien. Es gibt maximal 50 Muster, und die Muster müssen der Glob-Pattern-Syntax folgen. Zum Beispiel:

A. `tests/`

B. `*.jpeg`

C. `large_data.zip`

b. Für Select Definition Folder aus S3:

i. Geben Sie den Amazon S3 S3-Speicherort ein, der den komprimierten Workflow-Definitionsordner enthält. Der Amazon S3 S3-Bucket muss sich in derselben Region wie der Workflow befinden.

ii. Wenn Ihr Konto nicht Eigentümer des Amazon S3 S3-Buckets ist, geben Sie die Konto-ID des Bucket-Besitzers in die AWS Konto-ID des S3-Bucket-Besitzers ein. Diese Informationen sind erforderlich, um die Inhaberschaft des Buckets verifizieren zu HealthOmics können.

c. Für Wählen Sie den Definitionsordner aus einer lokalen Quelle aus:

i. Geben Sie den Speicherort des komprimierten Workflow-Definitionsordners auf dem lokalen Laufwerk ein.

3. Haupt-Workflow-Definitionsdateipfad (optional): Geben Sie den Dateipfad vom komprimierten Workflow-Definitionsordner oder Repository zur `main` Datei ein. Dieser Parameter ist nicht erforderlich, wenn der Workflow-Definitionsordner nur eine Datei enthält oder wenn die Hauptdatei den Namen „main“ hat.

6. Wählen Sie im Bereich README-Datei (optional) die Quelle der README-Datei aus und geben Sie die folgenden Informationen ein:

- Geben Sie für Import aus einem Repository-Service im Feld README-Dateipfad den Pfad zur README-Datei innerhalb des Repositorys ein.
- Geben Sie unter Datei aus S3 auswählen in der README-Datei in S3 den Amazon S3 S3-URI für die README-Datei ein.
- Für Datei aus einer lokalen Quelle auswählen: Wählen Sie in README — optional die Option Datei auswählen, um die hochzuladende Markdown-Datei (.md) auszuwählen.

7. Geben Sie im Bereich „Konfiguration des Standardlaufspeichers“ den Standardspeichertyp und die Kapazität für Läufe an, die diesen Workflow verwenden:

1. Speichertyp ausführen: Wählen Sie aus, ob statischer oder dynamischer Speicher als Standard für den temporären Laufspeicher verwendet werden soll. Die Standardeinstellung ist statischer Speicher.
2. Laufspeicherkapazität (optional): Für den statischen Laufspeichertyp können Sie die Standardmenge an Laufspeicher eingeben, die für diesen Workflow erforderlich ist. Der Standardwert für diesen Parameter ist 1200 GiB. Sie können diese Standardwerte überschreiben, wenn Sie einen Lauf starten.

8. Tags (optional): Sie können diesem Workflow bis zu 50 Tags zuordnen.
9. Wählen Sie Weiter aus.
10. Wählen Sie auf der Seite Workflow-Parameter hinzufügen (optional) die Parameterquelle aus:

    1. Bei Aus Workflow-Definitionsdatei analysieren: HealthOmics Analysiert automatisch die Workflow-Parameter aus der Workflow-Definitionsdatei.
    2. Verwenden Sie für Parametervorlage aus dem Git-Repository bereitstellen den Pfad zur Parametervorlagendatei aus Ihrem Repository.
    3. Laden Sie für „JSON-Datei aus lokaler Quelle auswählen“ eine JSON Datei aus einer lokalen Quelle hoch, die die Parameter angibt.
    4. Geben Sie für Workflow-Parameter manuell eingeben die Parameternamen und Beschreibungen manuell ein.

11. Im Bereich Parametervorschau können Sie die Parameter für diese Workflow-Version überprüfen oder ändern. Wenn Sie die JSON Datei wiederherstellen, gehen alle lokalen Änderungen verloren, die Sie vorgenommen haben.
12. Wählen Sie Weiter aus.
13. Auf der Seite zur Neuzuweisung von Container-URI können Sie im Bereich Zuordnungsregeln Regeln für die URI-Zuordnung für Ihren Workflow definieren.

    Wählen Sie für Quelle der Zuordnungsdatei eine der folgenden Optionen aus:

    - Keine — Keine Zuordnungsregeln erforderlich.
    - JSON-Datei aus S3 auswählen — Geben Sie den S3-Speicherort für die Zuordnungsdatei an.
    - JSON-Datei aus einer lokalen Quelle auswählen — Geben Sie den Speicherort der Mapping-Datei auf Ihrem lokalen Gerät an.
    - Zuordnungen manuell eingeben — Geben Sie die Registry- und Image-Zuordnungen im Bereich „Zuordnungen“ ein.

14. In der Konsole wird der Bereich „Zuordnungen“ angezeigt. Wenn Sie eine Mapping-Quelldatei ausgewählt haben, zeigt die Konsole die Werte aus der Datei an.

    a. In Registrierungszuordnungen können Sie die Zuordnungen bearbeiten oder Zuordnungen hinzufügen (maximal 20 Registrierungszuordnungen).

       Jede Registrierungszuweisung enthält die folgenden Felder:

       - Upstream-Registrierungs-URL — Die URI der Upstream-Registrierung.
       - ECR-Repository-Präfix — Das Repository-Präfix, das im privaten Amazon ECR-Repository verwendet werden soll.
       - (Optional) Upstream-Repository-Präfix — Das Präfix des Repositorys in der Upstream-Registrierung.
       - (Optional) ECR-Konto-ID — Konto-ID des Kontos, dem das Upstream-Container-Image gehört.

    b. In Image-Zuordnungen können Sie die Image-Zuordnungen bearbeiten oder Zuordnungen hinzufügen (maximal 100 Image-Zuordnungen).

       Jede Image-Zuordnung enthält die folgenden Felder:

       - Quellbild — Gibt den URI des Quellbilds in der Upstream-Registrierung an.
       - Zielbild — Gibt den URI des entsprechenden Images in der privaten Amazon ECR-Registrierung an.

15. Wählen Sie Weiter aus.
16. Überprüfen Sie die Workflow-Konfiguration und wählen Sie dann Workflow erstellen.

Einen Workflow mit der CLI erstellen

Wenn sich Ihre Workflow-Dateien und die Parameter-Vorlagendatei auf Ihrem lokalen Computer befinden, können Sie mit dem folgenden CLI-Befehl einen Workflow erstellen.

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

Der `create-workflow` Vorgang gibt die folgende Antwort zurück:

```
{
  "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"
}
```

Optionale Parameter, die beim Erstellen eines Workflows verwendet werden können

Sie können jeden der optionalen Parameter angeben, wenn Sie einen Workflow erstellen. Einzelheiten zur Syntax finden Sie CreateWorkflowin der HealthOmics AWS-API-Referenz.

Themen

- Geben Sie den Amazon S3 S3-Speicherort der Workflow-Definition an
- Verwenden Sie die Workflow-Definition aus einem Git-basierten Repository
- Geben Sie eine Readme-Datei an
- mainGeben Sie die Definitionsdatei an
- Geben Sie den Speichertyp für die Ausführung an
- Geben Sie die GPU-Konfiguration an
- Konfigurieren Sie die Parameter für die Zuordnung des Pull-Through-C

Geben Sie den Amazon S3 S3-Speicherort der Workflow-Definition an

Wenn sich Ihre Workflow-Definitionsdatei in einem Amazon S3 S3-Ordner befindet, geben Sie den Speicherort mithilfe des `definition-uri` Parameters an, wie im folgenden Beispiel gezeigt. Wenn Ihr Konto nicht Eigentümer des Amazon S3 S3-Buckets ist, geben Sie die AWS-Konto ID des Besitzers an.

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

Verwenden Sie die Workflow-Definition aus einem Git-basierten Repository

Um die Workflow-Definition aus einem unterstützten Git-basierten Repository zu verwenden, verwenden Sie den `definition-repository` Parameter in Ihrer Anfrage. Geben Sie keinen anderen `definition` Parameter an, da eine Anfrage fehlschlägt, wenn sie mehr als eine Eingabequelle enthält.

Der `definition-respository` Parameter enthält die folgenden Felder:

- connectionArn— ARN der Code-Verbindung, die Ihre AWS-Ressourcen mit dem externen Repository verbindet.
- fullRepositoryId— Geben Sie die Repository-ID als ein`owner-name/repo-name`. Stellen Sie sicher, dass Sie Zugriff auf die Dateien in diesem Repository haben.
- sourceReference(Optional) — Geben Sie einen Repository-Referenztyp (BRANCH, TAG oder COMMIT) und einen Wert ein.

  HealthOmics verwendet den neuesten Commit für den Standard-Branch, wenn Sie keinen Quellverweis angeben.
- excludeFilePatterns(Optional) — Geben Sie die Dateimuster ein, um bestimmte Ordner, Dateien oder Erweiterungen auszuschließen. Dies hilft bei der Verwaltung der Datengröße beim Import von Repository-Dateien. Geben Sie maximal 50 Muster an. Die Muster müssen der Glob-Pattern-Syntax folgen. Zum Beispiel:
  - `tests/`
  - `*.jpeg`
  - `large_data.zip`

Wenn Sie die Workflow-Definition aus einem Git-basierten Repository angeben, verwenden Sie diese Option, `parameter-template-path` um die Parameter-Vorlagendatei anzugeben. Wenn Sie diesen Parameter nicht angeben, HealthOmics wird der Workflow ohne Parametervorlage erstellt.

Das folgende Beispiel zeigt die Parameter, die sich auf Inhalte aus einem Git-basierten privaten Repository beziehen:

```
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" \
```

Weitere Beispiele finden Sie im Blogbeitrag So erstellen Sie HealthOmics AWS-Workflows aus Inhalten in Git.

Geben Sie eine Readme-Datei an

Sie können den Speicherort der README-Datei mit einem der folgenden Parameter angeben:

- readme-markdown— Zeichenketteneingabe oder eine Datei auf Ihrem lokalen Computer.
- readme-uri— Die URI einer auf S3 gespeicherten Datei.
- readme-path — Der Pfad zur README-Datei im Repository.

Verwenden Sie den Readme-Pfad nur in Verbindung mit definition-respository. Wenn Sie keinen README-Parameter angeben, wird die README.md-Datei auf Stammebene in das Repository HealthOmics importiert (falls vorhanden).

Die folgenden Beispiele zeigen, wie der Speicherort der README-Datei mithilfe des Readme-Pfads und der Readme-URI angegeben wird.

```
# 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"
```

Weitere Informationen finden Sie unter HealthOmics Workflow-README-Dateien.

mainGeben Sie die Definitionsdatei an

Wenn Sie mehrere Workflow-Definitionsdateien einbeziehen, verwenden Sie den `main` Parameter, um die Hauptdefinitionsdatei für Ihren Workflow anzugeben.

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

Geben Sie den Speichertyp für die Ausführung an

Sie können den Standard-Laufspeichertyp (DYNAMIC oder STATIC) und die Run-Speicherkapazität (erforderlich für statischen Speicher) angeben. Weitere Hinweise zu Laufspeichertypen finden Sie unterSpeichertypen in HealthOmics Workflows ausführen.

```
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  \
```

Geben Sie die GPU-Konfiguration an

Verwenden Sie den Accelerators-Parameter, um einen Workflow zu erstellen, der auf einer Accelerated-Compute-Instance ausgeführt wird. Das folgende Beispiel zeigt, wie der Parameter verwendet wird. `accelerators` Sie geben die GPU-Konfiguration in der Workflow-Definition an. Siehe Instanzen für beschleunigte Datenverarbeitung.

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

Konfigurieren Sie die Parameter für die Zuordnung des Pull-Through-C

Wenn Sie die Amazon ECR Pull-Through-Cache-Zuordnungsfunktion verwenden, können Sie die Standardzuordnungen überschreiben. Weitere Informationen zu den Container-Einrichtungsparametern finden Sie unter. Container-Images für private Workflows

Im folgenden Beispiel `mappings.json` enthält die Datei diesen Inhalt:

```
{
    "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"
        }
    ]
}
```

Geben Sie die Zuordnungsparameter im Befehl create-workflow an:

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

```
    ...
```

Sie können auch den S3-Speicherort der Datei mit den Zuordnungsparametern angeben:

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

Einen Workflow mithilfe eines SDK erstellen

Sie können einen Workflow mit einem der erstellen SDKs. Das folgende Beispiel zeigt, wie ein Workflow mit dem Python-SDK erstellt wird.

```
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={ ... }
)
```

Einen privaten Workflow aktualisieren

Sie können einen Workflow über die HealthOmics Konsole, AWS CLI-Befehle oder einen der folgenden aktualisieren AWS SDKs.

> **Note**
>
> Nehmen Sie keine persönlich identifizierbaren Informationen (PII) in Workflow-Namen auf. Diese Namen sind in CloudWatch Protokollen sichtbar.

Themen

- Aktualisierung eines Workflows mithilfe der Konsole

- Aktualisieren eines Workflows mit der CLI
- Aktualisierung eines Workflows mithilfe eines SDK

Aktualisierung eines Workflows mithilfe der Konsole

Schritte zum Aktualisieren eines Workflows

1. Öffnen Sie die HealthOmics -Konsole.
2. Öffnen Sie bei Bedarf den linken Navigationsbereich (←). Wählen Sie Private Workflows.
3. Wählen Sie auf der Seite Private Workflows den Workflow aus, der aktualisiert werden soll.
4. Auf der Workflow-Seite:
   - Wenn der Workflow Versionen hat, stellen Sie sicher, dass Sie die Standardversion auswählen.
   - Wählen Sie in der Liste Aktionen die Option Ausgewählte Bearbeitung aus.
5. Auf der Seite „Workflow bearbeiten“ können Sie jeden der folgenden Werte ändern:
   - Name des Workflows.
   - Beschreibung des Workflows.
   - Der standardmäßige Run-Speichertyp für den Workflow.
   - Die standardmäßige Run-Speicherkapazität (wenn der Run-Speichertyp statischer Speicher ist). Weitere Informationen zur standardmäßigen Run-Speicherkonfiguration finden Sie unterEinen Workflow mithilfe der Konsole erstellen.
6. Wählen Sie Änderungen speichern, um die Änderungen zu übernehmen.

Aktualisieren eines Workflows mit der CLI

Wie im folgenden Beispiel gezeigt, können Sie den Namen und die Beschreibung des Workflows aktualisieren. Sie können auch den Standard-Laufspeichertyp (STATIC oder DYNAMIC) und die Run-Speicherkapazität (für den statischen Speichertyp) ändern. Weitere Informationen zu Laufspeichertypen finden Sie unterSpeichertypen in HealthOmics Workflows ausführen.

```
aws omics update-workflow    \
  --id 1234567    \
  --name my_workflow      \
  --description "updated workflow"    \
```

```
  --storage-type 'STATIC'    \
  --storage-capacity 1200
```

Sie erhalten keine Antwort auf die `update-workflow` Anfrage.

Aktualisierung eines Workflows mithilfe eines SDK

Sie können einen Workflow mit einem der aktualisieren SDKs.

Das folgende Beispiel zeigt, wie ein Workflow mit dem Python-SDK aktualisiert wird.

```
import boto3

omics = boto3.client('omics')

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

Löschen Sie einen privaten Workflow

Wenn Sie einen Workflow nicht mehr benötigen, können Sie ihn mit der HealthOmics Konsole, AWS CLI-Befehlen oder einem der folgenden Befehle löschen AWS SDKs. Sie können einen Workflow löschen, der die folgenden Kriterien erfüllt:

- Sein Status ist AKTIV oder FEHLGESCHLAGEN.
- Es hat keine aktiven Aktien.
- Sie haben alle Workflow-Versionen gelöscht.

Das Löschen eines Workflows hat keine Auswirkungen auf laufende Läufe, die den Workflow verwenden.

Themen

Löschen eines Workflows mithilfe der Konsole

Um einen Workflow zu löschen

1. Öffnen Sie die HealthOmics -Konsole.
2. Öffnen Sie bei Bedarf den linken Navigationsbereich (←). Wählen Sie Private Workflows aus.
3. Wählen Sie auf der Seite Private Workflows den zu löschenden Workflow aus.
4. Wählen Sie auf der Workflow-Seite in der Liste Aktionen die Option Ausgewählte löschen aus.
5. Geben Sie im Workflow-Modal „Löschen“ „Bestätigen“ ein, um den Löschvorgang zu bestätigen.
6. Wählen Sie Löschen aus.

Löschen eines Workflows mit der CLI

Das folgende Beispiel zeigt, wie Sie den AWS CLI Befehl verwenden können, um einen Workflow zu löschen. Um das Beispiel auszuführen, ersetzen Sie den *`workflow id`* durch die ID des Workflows, den Sie löschen möchten.

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

HealthOmics sendet keine Antwort auf die `delete-workflow` Anfrage.

Löschen eines Workflows mithilfe eines SDK

Sie können einen Workflow mit einem der löschen SDKs.

Das folgende Beispiel zeigt, wie ein Workflow mit dem Python-SDK gelöscht wird.

```
import boto3

omics = boto3.client('omics')

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

Überprüfen Sie den Workflow-Status

Nachdem Sie Ihren Workflow erstellt haben, können Sie den Status überprüfen und weitere Details des Workflows mithilfe von get-workflow anzeigen, wie in der Abbildung gezeigt.

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

Die Antwort enthält Workflow-Details, einschließlich des Status, wie in der Abbildung gezeigt.

```
{
    "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"
  }
```

Sie können mit diesem Workflow einen Lauf starten, nachdem der Status zu gewechselt ist`ACTIVE`.

Referenzieren von Genomdateien aus einer Workflow-Definition

Auf ein HealthOmics Referenzspeicherobjekt kann mit einem URI wie dem folgenden verwiesen werden. Verwenden Sie Ihre eigenen*account ID*,*reference store ID*, und *reference ID* wo angegeben.

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

Für einige Workflows sind `SOURCE` sowohl die als auch die `INDEX` Dateien für das Referenzgenom erforderlich. Der vorherige URI ist die Standardkurzform und wird standardmäßig auf die SOURCE-Datei gesetzt. Um eine der beiden Dateien anzugeben, können Sie die lange URI-Form wie folgt verwenden.

```
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
```

Die Verwendung eines Sequence-Readesets hätte ein ähnliches Muster, wie hier gezeigt.

```
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
```

Einige Lesesätze, z. B. solche, die auf FASTQ basieren, können gepaarte Lesesätze enthalten. In den folgenden Beispielen werden sie als SOURCE1 und SOURCE2 bezeichnet. Formate wie BAM und CRAM enthalten nur eine SOURCE1 Datei. Einige Lesesätze enthalten INDEX-Dateien wie `bai` ODER-Dateien. `crai` Der vorhergehende URI ist die Standardkurzform und verwendet standardmäßig die SOURCE1 Datei. Um die genaue Datei oder den Index anzugeben, können Sie die lange URI-Form wie folgt verwenden.

```
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
```

Im Folgenden finden Sie ein Beispiel für eine JSON-Eingabedatei, die zwei Omics Storage URIs verwendet.

```
{
   "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>"
}
```

Verweisen Sie auf die JSON-Eingabedatei in, AWS CLI indem Sie sie `--inputs file://<input_file.json>` zu Ihrer Start-Run-Anforderung hinzufügen.

Workflow-Versionierung in HealthOmics

Wenn Sie Änderungen an einem Workflow vornehmen müssen, können Sie entweder einen neuen Workflow oder eine neue Workflow-Version erstellen. Versionen sind unveränderlich, mit Ausnahme der zulässigen Konfigurationsänderungen, die sich nicht auf die Ausführungslogik auswirken.

Workflow-Versionen bieten die folgenden Vorteile:

- Versionen bilden eine logische Gruppe von Workflows, die miteinander verwandt sind. Sie können jeder Workflow-Version einen benutzerdefinierten Namen hinzufügen, um sie einfacher zu verwalten (insbesondere bei einem Workflow mit einer großen Anzahl von Versionen).
- Sie können mehrere Versionen eines Workflows gleichzeitig ausführen.
- Alle Versionen eines Workflows verwenden dieselbe Workflow-ID und denselben Basis-ARN, wodurch das Pipeline-Management vereinfacht werden kann, nachdem Sie einen Workflow geändert haben.
- Workflow-Versionen bieten dieselbe Datenherkunft wie Workflows. Versionen sind unveränderlich und erstellen HealthOmics einen eindeutigen ARN für jede Workflow-Version. Die Version ARN enthält die Workflow-ID und den Versionsnamen, wie im folgenden Beispiel gezeigt:

  ```
  arn:aws:omics:us-west-2:123456789012:workflow/1234567/version/
  myUniqueVersionName
  ```
- Wenn Sie Eigentümer eines gemeinsam genutzten Workflows sind, können Sie den Workflow aktualisieren, ohne die Abonnenten zu stören (die die vorherige Version weiterhin verwenden können). Abonnenten können auf alle Workflow-Versionen zugreifen. Wenn Sie eine neue Version erstellen, müssen Sie den Workflow nicht erneut teilen.
- Wenn Sie eine Workflow-Ausführung starten, können Sie die Workflow-Version angeben.
  - Benutzer können sich dafür entscheiden, für Produktionsläufe eine stabile Version zu verwenden und die neueste Version für einen Testlauf auszuprobieren.
  - Benutzer können zur vorherigen Version eines Workflows zurückkehren, wenn sie auf Probleme mit der neuen Version stoßen.
  - Abonnenten eines gemeinsam genutzten Workflows können wählen, welche Version verwendet werden soll.

Themen

Standard-Workflow-Version

HealthOmics Behandelt den ursprünglichen Workflow als Standardversion, nachdem Sie eine oder mehrere Versionen eines Workflows erstellt haben. Wenn Sie einen Lauf starten, können Sie optional eine Workflow-Version für den Lauf angeben. Wenn Sie beim Starten eines Laufs keine Version angeben, HealthOmics verwendet die Standardversion.

HealthOmics Zeigt in der Konsole den ursprünglichen Workflow mit der Bezeichnung Standardversion an. Die Konsole verwendet dieses Label erst, nachdem Sie eine oder mehrere Workflow-Versionen erstellt haben. Der ursprüngliche Workflow bleibt immer die Standardversion. Sie können keine andere Version als Standardversion zuweisen.

Sie können die Standardversion eines Workflows nicht löschen, wenn dem Workflow andere Versionen zugeordnet sind. Weitere Informationen finden Sie unter Löschen Sie einen privaten Workflow.

Workflow-Version erstellen

Wenn Sie eine neue Version eines Workflows erstellen, müssen Sie die Konfigurationswerte für die neue Version angeben. Es erbt keine Konfigurationswerte aus dem Workflow.

Wenn Sie die Version erstellen, geben Sie einen Versionsnamen an, der für diesen Workflow eindeutig ist. Sie können den Namen nicht ändern, nachdem die Version HealthOmics erstellt wurde.

Der Versionsname muss mit einem Buchstaben oder einer Zahl beginnen und kann Groß- und Kleinbuchstaben, Zahlen, Bindestriche, Punkte und Unterstriche enthalten. Die maximale Länge beträgt 64 Zeichen. Sie können beispielsweise ein einfaches Benennungsschema wie Version1, Version2, Version3 verwenden. Sie können Ihre Workflow-Versionen auch Ihren eigenen internen Versionierungskonventionen wie 2.7.0, 2.7.1, 2.7.2 zuordnen.

Verwenden Sie optional das Feld für die Versionsbeschreibung, um Anmerkungen zu dieser Version hinzuzufügen. Beispiel: Fix for syntax error in workflow definition.

> **Note**
>
> Nehmen Sie keine persönlich identifizierbaren Informationen (PII) in den Versionsnamen auf. Versionsnamen werden im ARN der Workflow-Version angezeigt.

HealthOmics weist der Workflow-Version einen eindeutigen ARN zu. Der ARN ist aufgrund der Kombination aus Workflow-ID und Versionsname eindeutig.

> **Warning**
>
> Nachdem Sie eine Workflow-Version gelöscht haben, HealthOmics können Sie den Versionsnamen für eine andere Workflow-Version wiederverwenden. Es hat sich bewährt, Versionsnamen nicht wiederzuverwenden. Wenn Sie einen Namen wiederverwenden, haben der Workflow und jede Version eine eindeutige UUID, die Sie als Herkunft verwenden können.

Themen

Erstellen Sie mithilfe der Konsole eine Workflow-Version

Schritte zum Erstellen einer Workflow-Version

1. Öffnen Sie die HealthOmics -Konsole.
2. Öffnen Sie bei Bedarf den linken Navigationsbereich (•). Wählen Sie Private Workflows aus.
3. Wählen Sie auf der Seite Private Workflows den Workflow für die neue Version aus.
4. Wählen Sie auf der Seite mit den Workflow-Details die Option Neue Version erstellen aus.
5. Geben Sie auf der Seite Version erstellen die folgenden Informationen ein:
    1. Versionsname: Geben Sie einen Namen für die Workflow-Version ein, der für den gesamten Workflow eindeutig ist.
    2. Versionsbeschreibung (optional): Sie können das Beschreibungsfeld verwenden, um Hinweise zu dieser Version hinzuzufügen.
6. Geben Sie im Bereich Workflow-Definition die folgenden Informationen ein:
    1. Workflow-Sprache (optional): Wählen Sie die Spezifikationssprache für die Workflow-Version aus. Andernfalls HealthOmics bestimmt die Sprache anhand der Workflow-Definition.

2. Wählen Sie unter Workflow-Definitionsquelle, ob Sie den Definitionsordner aus einem Git-basierten Repository, einem Amazon S3 S3-Speicherort oder von einem lokalen Laufwerk importieren möchten.

   a. Für den Import aus einem Repository-Service:

   > **Note**
   >
   > HealthOmics unterstützt öffentliche und private Repositorys fürGitHub,GitLab,, BitbucketGitHub self-managed,GitLab self-managed.

   i. Wählen Sie eine Verbindung, um Ihre AWS Ressourcen mit dem externen Repository zu verbinden. Informationen zum Herstellen einer Verbindung finden Sie unterConnect mit externen Code-Repositorys her.

   > **Note**
   >
   > Kunden in der TLV Region müssen eine Verbindung in der Region IAD (us-east-1) herstellen, um einen Workflow zu erstellen.

   ii. Geben Sie unter Vollständige Repository-ID Ihre Repository-ID als Benutzername/Repository-Name ein. Stellen Sie sicher, dass Sie Zugriff auf die Dateien in diesem Repository haben.

   iii. Geben Sie im Feld Quellverweis (optional) eine Repository-Quellenreferenz ein (Branch-, Tag- oder Commit-ID). HealthOmics verwendet den Standardzweig, wenn kein Quellverweis angegeben ist.

   iv. Geben Sie im Feld Dateimuster ausschließen die Dateimuster ein, um bestimmte Ordner, Dateien oder Erweiterungen auszuschließen. Dies hilft bei der Verwaltung der Datengröße beim Import von Repository-Dateien. Es gibt maximal 50 Muster, und die Muster müssen der Glob-Pattern-Syntax folgen. Zum Beispiel:

      A. `tests/`

      B. `*.jpeg`

      C. `large_data.zip`

   b. Für Select Definition Folder aus S3:

i. Geben Sie den Amazon S3 S3-Speicherort ein, der den komprimierten Workflow-Definitionsordner enthält. Der Amazon S3 S3-Bucket muss sich in derselben Region wie der Workflow befinden.

ii. Wenn Ihr Konto nicht Eigentümer des Amazon S3 S3-Buckets ist, geben Sie die Konto-ID des Bucket-Besitzers in die AWS Konto-ID des S3-Bucket-Besitzers ein. Diese Informationen sind erforderlich, um die Inhaberschaft des Buckets verifizieren zu HealthOmics können.

c. Für Wählen Sie den Definitionsordner aus einer lokalen Quelle aus:

i. Geben Sie den Speicherort des komprimierten Workflow-Definitionsordners auf dem lokalen Laufwerk ein.

3. Haupt-Workflow-Definitionsdateipfad (optional): Geben Sie den Dateipfad vom komprimierten Workflow-Definitionsordner oder Repository zur `main` Datei ein. Dieser Parameter ist nicht erforderlich, wenn der Workflow-Definitionsordner nur eine Datei enthält oder wenn die Hauptdatei den Namen „main“ hat.

7. Wählen Sie im Bereich README-Datei (optional) die Quelle der README-Datei aus und geben Sie die folgenden Informationen ein:

- Geben Sie für Import aus einem Repository-Service im Feld README-Dateipfad den Pfad zur README-Datei innerhalb des Repositorys ein.
- Geben Sie unter Datei aus S3 auswählen in der README-Datei in S3 den Amazon S3 S3-URI für die README-Datei ein.
- Für Datei aus einer lokalen Quelle auswählen: Wählen Sie in README — optional die Option Datei auswählen, um die hochzuladende Markdown-Datei (.md) auszuwählen.

8. Geben Sie im Bereich „Konfiguration des Standardlaufspeichers“ den Standardspeichertyp und die Kapazität für Läufe an, die diesen Workflow verwenden:

1. Speichertyp ausführen: Wählen Sie aus, ob statischer oder dynamischer Speicher als Standard für den temporären Laufspeicher verwendet werden soll. Die Standardeinstellung ist statischer Speicher.
2. Laufspeicherkapazität (optional): Für den statischen Laufspeichertyp können Sie die Standardmenge an Laufspeicher eingeben, die für diesen Workflow erforderlich ist. Der Standardwert für diesen Parameter ist 1200 GiB. Sie können diese Standardwerte überschreiben, wenn Sie einen Lauf starten.

9. Tags (optional): Sie können dieser Workflow-Version bis zu 50 Tags zuordnen.

10. Wählen Sie Weiter aus.

11. Wählen Sie auf der Seite Workflow-Parameter hinzufügen (optional) die Parameterquelle aus:

    1. Bei Aus Workflow-Definitionsdatei analysieren: HealthOmics Analysiert automatisch die Workflow-Parameter aus der Workflow-Definitionsdatei.
    2. Verwenden Sie für Parametervorlage aus dem Git-Repository bereitstellen den Pfad zur Parametervorlagendatei aus Ihrem Repository.
    3. Laden Sie für „JSON-Datei aus lokaler Quelle auswählen“ eine JSON Datei aus einer lokalen Quelle hoch, die die Parameter angibt.
    4. Geben Sie für Workflow-Parameter manuell eingeben die Parameternamen und Beschreibungen manuell ein.

12. Im Bereich Parametervorschau können Sie die Parameter für diese Workflow-Version überprüfen oder ändern. Wenn Sie die JSON Datei wiederherstellen, gehen alle lokalen Änderungen verloren, die Sie vorgenommen haben.
13. Auf der Seite zur Neuzuweisung von Container-URI können Sie im Bereich Zuordnungsregeln Regeln für die URI-Zuordnung für Ihren Workflow definieren.

    Wählen Sie für Quelle der Zuordnungsdatei eine der folgenden Optionen aus:

    - Keine — Keine Zuordnungsregeln erforderlich.
    - JSON-Datei aus S3 auswählen — Geben Sie den S3-Speicherort für die Zuordnungsdatei an.
    - JSON-Datei aus einer lokalen Quelle auswählen — Geben Sie den Speicherort der Mapping-Datei auf Ihrem lokalen Gerät an.
    - Zuordnungen manuell eingeben — Geben Sie die Registry- und Image-Zuordnungen im Bereich „Zuordnungen“ ein.

14. In der Konsole wird der Bereich „Zuordnungen“ angezeigt. Wenn Sie eine Mapping-Quelldatei ausgewählt haben, zeigt die Konsole die Werte aus der Datei an.

    a. In Registrierungszuordnungen können Sie die Zuordnungen bearbeiten oder Zuordnungen hinzufügen (maximal 20 Registrierungszuordnungen).

       Jede Registrierungszuweisung enthält die folgenden Felder:

       - Upstream-Registrierungs-URL — Die URI der Upstream-Registrierung.
       - ECR-Repository-Präfix — Das Repository-Präfix, das im privaten Amazon ECR-Repository verwendet werden soll.

- (Optional) Upstream-Repository-Präfix — Das Präfix des Repositorys in der Upstream-Registrierung.
- (Optional) ECR-Konto-ID — Konto-ID des Kontos, dem das Upstream-Container-Image gehört.

b. In Image-Zuordnungen können Sie die Image-Zuordnungen bearbeiten oder Zuordnungen hinzufügen (maximal 100 Image-Zuordnungen).

Jede Image-Zuordnung enthält die folgenden Felder:

- Quellbild — Gibt den URI des Quellbilds in der Upstream-Registrierung an.
- Zielbild — Gibt den URI des entsprechenden Images in der privaten Amazon ECR-Registrierung an.

15. Wählen Sie Weiter aus.
16. Überprüfen Sie die Versionskonfiguration und wählen Sie dann Version erstellen.

Wenn die Version erstellt wurde, kehrt die Konsole zur Workflow-Detailseite zurück und zeigt die neue Version in der Tabelle Workflows und Versionen an.

Erstellen Sie eine Workflow-Version mit der CLI

Sie können mithilfe der `CreateWorkflowVersion` API-Operation eine Workflow-Version erstellen. HealthOmics Verwendet für optionale Parameter die folgenden Standardwerte:

Parameter	Standard
Engine	Wird anhand der Workflow-Definition bestimmt
Speichertyp	STATIC
Speicherkapazität (für statischen Speicher)	1200 GiB
Main (Haupt)	Wird anhand des Inhalts des Workflow-Definitionsordners ermittelt. Details hierzu finden Sie unter HealthOmics Anforderungen an die Workflow-Definition.
Accelerators	Keine

Parameter	Standard
Tags (Markierungen)	Keine

Das folgende CLI-Beispiel erstellt eine Workflow-Version mit statischem Speicher als Standard-Run-Storage:

```
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
```

Wenn sich Ihre Workflow-Definitionsdatei in einem Amazon S3 S3-Ordner befindet, geben Sie den Speicherort mit dem `definition-uri` Parameter anstelle von ein`definition-zip`. Weitere Informationen finden Sie CreateWorkflowVersionin der HealthOmics AWS-API-Referenz.

Sie erhalten die folgende Antwort auf die `create-workflow-version` Anfrage.

```
{
  "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"
}
```

Erstellen Sie eine Workflow-Version mit einem SDK

Sie können einen Workflow mit einem der erstellen SDKs.

Das folgende Beispiel zeigt, wie eine Workflow-Version mit dem Python-SDK erstellt wird.

```
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={ ... }
)
```

Überprüfen Sie den Status einer Workflow-Version

Nachdem Sie Ihre Workflow-Version erstellt haben, können Sie den Status überprüfen und weitere Details des Workflows get-workflow-versionwie in der Abbildung gezeigt anzeigen.

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

In der Antwort erhalten Sie Ihre Workflow-Details, einschließlich des Status, wie in der Abbildung dargestellt.

```
{
  "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"
}
```

Bevor Sie einen Lauf mit dieser Workflow-Version starten können, muss der Status auf geändert werden`ACTIVE`.

Eine Workflow-Version aktualisieren

Sie können die Beschreibung und die standardmäßige Run-Storage-Konfiguration für eine private Workflow-Version aktualisieren. Um andere Informationen in der Workflow-Version zu ändern, erstellen Sie eine neue Version.

Themen

Aktualisieren Sie eine Workflow-Version mithilfe der Konsole

Um eine Workflow-Version zu aktualisieren

1. Öffnen Sie die HealthOmics -Konsole.
2. Öffnen Sie bei Bedarf den linken Navigationsbereich (←). Wählen Sie Private Workflows aus.
3. Wählen Sie auf der Seite Private Workflows den Workflow aus.
4. Wählen Sie auf der Workflow-Seite die Workflow-Version aus, die Sie aktualisieren möchten, und wählen Sie in der Liste Aktionen die Option Ausgewählte Bearbeitung aus.
   - Wenn Sie die Standardversion wählen, öffnet die Konsole die Seite Workflow bearbeiten. Weitere Informationen finden Sie unter Einen privaten Workflow aktualisieren.
   - Wenn Sie eine benutzerdefinierte Version wählen, öffnet die Konsole die Seite Version bearbeiten.
5. Geben Sie auf der Seite Version bearbeiten die folgenden Informationen ein
   - Versionsbeschreibung (optional) — Eine Beschreibung dieser Version.
6. Geben Sie im Bereich „Konfiguration des Standardlaufspeichers“ die folgenden Standardwerte für Läufe an, die diese Workflow-Version verwenden. Sie können die Standardwerte überschreiben, wenn Sie einen Lauf starten:
   - Wählen Sie für den Speichertyp Run die Option Statisch oder Dynamisch aus.
   - Wählen Sie für statischen Run-Speicher die Standardmenge an Run-Speicherkapazität für Läufe aus, die diese Workflow-Version verwenden. Der Standardwert für diesen Parameter ist 1200 GiB.

7. Wählen Sie Änderungen speichern aus.

Die Konsole kehrt zur Workflow-Detailseite zurück und zeigt ein Seitenbanner mit der aktualisierten Workflow-Version an.

Aktualisieren Sie eine Workflow-Version mit der CLI

Sie können Parameter für eine Workflow-Version mit dem folgenden CLI-Befehl aktualisieren. Die Kombination aus Workflow-ID und Versionsname identifiziert die Version eindeutig.

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

Sie erhalten keine Antwort auf die `update-workflow-version` Anfrage.

Aktualisieren Sie eine Workflow-Version mithilfe eines SDK

Sie können eine Workflow-Version mit einem der aktualisieren SDKs. Das folgende Python-SDK-Beispiel zeigt, wie der Speichertyp und die Beschreibung für eine Workflow-Version aktualisiert werden.

```
import boto3

omics = boto3.client('omics')

response = omics.update_workflow_version(
   workflowID=1234567,
   versionName='3.0.0',
   storageType='DYNAMIC',
   description='new version description'
)
```

Löschen Sie eine Workflow-Version

Sie können eine benutzerdefinierte Workflow-Version mit der Konsole, CLI oder einer der SDKs folgenden löschen. Das Löschen einer Workflow-Version hat keine Auswirkungen auf laufende Läufe, die die Workflow-Version verwenden.

Sie können das nicht löschenStandard-Workflow-Version. Sie löschen alle benutzerdefinierten Versionen und anschließend den Workflow.

Themen

- Löschen Sie eine Workflow-Version mithilfe der Konsole
- Löschen Sie eine Workflow-Version mit der CLI
- Löschen Sie eine Workflow-Version mithilfe eines SDK

Löschen Sie eine Workflow-Version mithilfe der Konsole

Um eine Workflow-Version zu löschen

1. Öffnen Sie die HealthOmics -Konsole.
2. Öffnen Sie bei Bedarf den linken Navigationsbereich (←). Wählen Sie Private Workflows aus.
3. Wählen Sie auf der Seite Private Workflows den Workflow aus.
4. Wählen Sie auf der Workflow-Seite die Workflow-Version aus, die Sie löschen möchten, und wählen Sie in der Liste Aktionen die Option Ausgewählte löschen aus.
5. Geben Sie im Modal „Workflow-Version löschen“ „Bestätigen“ ein, um den Löschvorgang zu bestätigen.
6. Wählen Sie Löschen aus.

In der Konsole wird ein Seitenbanner mit der gelöschten Workflow-Version angezeigt.

Löschen Sie eine Workflow-Version mit der CLI

Sie können eine benutzerdefinierte Workflow-Version mit dem folgenden CLI-Befehl löschen. Die Kombination aus Workflow-ID und Versionsname identifiziert die Version eindeutig.

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

Sie erhalten keine Antwort auf die `delete-workflow-version` Anfrage.

Löschen Sie eine Workflow-Version mithilfe eines SDK

Sie können einen Workflow mit einem der löschen SDKs.

Das folgende Beispiel zeigt, wie ein Workflow mit dem Python-SDK gelöscht wird.

```
import boto3

omics = boto3.client('omics')

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

HealthOmics Läufe verwenden

Nachdem Sie einen Workflow erstellt haben, können Sie Läufe mithilfe des Workflows starten.

Wenn Sie einen Lauf starten, HealthOmics wird temporärer Ausführungsspeicher zugewiesen, den die Workflow-Engine während der Ausführung verwenden kann. Um die Datenisolierung und Sicherheit zu gewährleisten HealthOmics , wird der Speicher zu Beginn jedes Laufs bereitgestellt und am Ende des Laufs wird die Bereitstellung aufgehoben.

HealthOmics stellt mehrere Kontingente für Workflow-Ausführungen und -Aufgaben bereit. Die Standardwerte sind bewusst konservativ, um unerwartete Kostenüberschreitungen zu vermeiden. Sie können eine Erhöhung dieser Kontingente beantragen. Weitere Informationen finden Sie unter HealthOmics Servicekontingenten.

Wenn Sie einen Lauf starten, HealthOmics weist er dem Lauf eine Lauf-ID und eine Lauf-UUID zu. Läufe in einem Konto haben einen eindeutigen Lauf. IDs Der gelöschte Lauf wird jedoch HealthOmics wiederverwendet IDs, sodass ein Lauf und ein gelöschter Lauf dieselbe Lauf-ID haben können. Außerdem ist es selten, aber möglich, dass ein gemeinsam genutzter Workflow dieselbe Lauf-ID wie ein Lauf in Ihrem Konto hat.

Dabei run uuid handelt es sich um eine GUID (Globally Unique Identifier), die Sie verwenden können, um Läufe zwischen Konten zu identifizieren oder um zwischen zwei Läufen in Ihrem Konto zu unterscheiden, die dieselbe Run-ID haben.

> **Note**
>
> Aus Gründen der Datenherkunft empfehlen wir, die run uuid zur eindeutigen Identifizierung von Läufen zu verwenden. Dies run uuid ist auch die beste Kennung, um eine

Verbindung zu Ihrem internen Laborinformationsmanagementsystem (LIMs) oder Ihrem Probenverfolgungssystem herzustellen.

Sie können Amazon Q CLI verwenden, um Ihre Läufe zu optimieren und die Laufleistung zu analysieren. Weitere Informationen finden Sie unter Beispielaufforderungen für Amazon Q CLI und im HealthOmics Agentic Generative AI-Tutorial unter. GitHub

Themen

Speichertypen in HealthOmics Workflows ausführen

Wenn Sie einen Lauf starten, wird temporärer HealthOmics Ausführungsspeicher zugewiesen, den die Workflow-Engine während des Laufs verwenden kann. HealthOmicsstellt den temporären Run-Speicher als Dateisystem bereit.

Für einen bestimmten Workflow oder eine Workflow-Ausführung können Sie zwischen dynamischem oder statischem Run-Speicher wählen. HealthOmics Stellt standardmäßig DYNAMISCHEN Run-Speicher bereit.

> **Note**
>
> Für die Nutzung von Run-Speicher fallen Gebühren für Ihr Konto an. Preisinformationen zu statischem und dynamischem Laufspeicher finden Sie unter HealthOmicsPreise.

Die folgenden Abschnitte enthalten Informationen, die Sie bei der Entscheidung, welcher Run-Speichertyp verwendet werden soll, berücksichtigen sollten.

Dynamischer Run-Speicher

Wir empfehlen die Verwendung von dynamischem Run-Storage für die meisten Läufe, einschließlich Läufe, für die schnellere Startzeiten erforderlich sind, für Läufe, bei denen Sie den Speicherbedarf nicht im Voraus kennen, und für iterative Entwicklungstestzyklen.

Sie müssen den benötigten Speicherplatz oder den Durchsatz für den Lauf nicht abschätzen. HealthOmics skaliert die Speichergröße je nach Auslastung des Dateisystems während des Laufs dynamisch nach oben oder unten. HealthOmics skaliert außerdem dynamisch den Durchsatz entsprechend den Anforderungen des Workflows. Eine Ausführung schlägt nie aufgrund eines Fehlers „Nicht genügend Speicherplatz für das Dateisystem“ fehl.

Dynamischer Run-Speicher bietet eine schnellere provisioning/deprovisioning Zeit als statischer Run-Speicher. Eine schnellere Einrichtung ist für die meisten Workflows von Vorteil und auch während development/test Zyklen von Vorteil.

Nach Abschluss der Ausführung (Erfolgspfad oder Fehlschlagpfad) gibt die GetRun-API-Operation den vom Lauf maximal belegten Speicherplatz im Feld StorageCapacity zurück. Sie finden diese Informationen auch in den Run-Manifest-Protokollen in der omics Protokollgruppe. Bei einem dynamischen Speicherlauf, der innerhalb von 2 Stunden abgeschlossen wird, ist der maximale Speicherwert möglicherweise nicht verfügbar.

Für den dynamischen Run-Speicher stellt der Lauf ein Dateisystem bereit, das das NFS-Protokoll verwendet. NFS behandelt CREATE-, DELETE- und RENAME-Dateioperationen als nicht idempotent, was gelegentlich zu Wettlaufbedingungen für diese Operationen führen kann, die Ihr Code ordnungsgemäß verarbeiten muss. Ihr Code sollte beispielsweise nicht fehlschlagen, wenn er versucht, eine Datei zu löschen, die nicht existiert. Vor der Einführung von Dynamic Run Storage empfehlen wir, Ihren Workflow-Code so anzupassen, dass er auch nicht-idempotenten Dateioperationen standhält. Siehe Codebeispiele für den sicheren Umgang mit nicht-idempotenten Vorgängen.

Codebeispiele für den sicheren Umgang mit nicht-idempotenten Vorgängen

Das folgende Python-Beispiel zeigt, wie eine Datei gelöscht werden kann, ohne dass ein Fehler auftritt, wenn die Datei nicht existiert.

```
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")
```

Die folgenden Beispiele verwenden die Bash-Shell. Um eine Datei sicher zu entfernen, auch wenn sie nicht existiert, verwenden Sie:

```
rm -f my_file
```

Um eine Datei sicher zu verschieben (umzubenennen), führen Sie den Befehl move nur aus, wenn die Datei im aktuellen Verzeichnis `old_name` vorhanden ist.

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

Verwenden Sie den folgenden Befehl, um ein Verzeichnis zu erstellen:

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

Statisch ausgeführter Speicher

Für den statischen Run-Speicher stellt der Run ein Dateisystem bereit, das das Lustre-Protokoll verwendet. Dieses Protokoll ist standardmäßig resistent gegen nicht-idempotente Dateioperationen. Sie müssen Ihren Workflow-Code nicht anpassen, um nicht-idempotente Dateioperationen zu verarbeiten.

HealthOmics weist eine feste Menge an Run-Speicher zu. Sie geben diesen Wert an, wenn Sie den Lauf starten. Der Standard-Run-Speicher ist 1200 GiB, wenn Sie keinen Wert angeben. Wenn Sie in der StartRun API-Anfrage einen Wert für die Speichergröße angeben, rundet das System den Wert auf das nächste Vielfache von 1200 GiB auf. Wenn diese Speichergröße nicht verfügbar ist, wird sie auf das nächste Vielfache von 2400 GiB aufgerundet.

Stellt für statischen Run-Speicher HealthOmics die folgenden Durchsatzwerte bereit:

- Basisdurchsatz von 200 MB/s pro TiB bereitgestellter Speicherkapazität.
- Burst-Durchsatz von bis zu 1300 MB/s pro TiB bereitgestellter Speicherkapazität.

Wenn die angegebene Speichergröße zu niedrig ist, schlägt die Ausführung fehl und es wird der Fehler Nicht genügend Speicherplatz für das Dateisystem angezeigt. Static Run Storage eignet sich gut für vorhersehbare Workflows mit bekannten Speicheranforderungen.

Static Run Storage eignet sich für große, Burst-Workloads mit hoher Aufgabenparallelität (z. B. eine große Menge parallel verarbeiteter RNASeq Samples). Es bietet einen höheren Dateisystemdurchsatz pro GiB und niedrigere Kosten pro GiB als Dynamic Run Storage.

Berechnung des erforderlichen statischen Laufspeichers

Ein Workflow benötigt zusätzliche Kapazität, wenn er statischen Laufspeicher verwendet (im Vergleich zu dynamischem Laufspeicher), da die Basisdateisysteminstallation 7% der Kapazität des statischen Dateisystems beansprucht.

Wenn Sie einen dynamischen Run-Storage-Workflow ausführen, um den maximalen Speicherplatz zu messen, der von der Ausführung verwendet wird, verwenden Sie die folgende Berechnung, um die Mindestmenge an statischem Speicher zu ermitteln:

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

Beispiel:

```
    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.
```

Daher sind 1200 GiB (die Mindestkapazität für statischen Laufspeicher) für diesen Lauf ausreichend.

Retentionsmodus für HealthOmics Läufe ausführen

HealthOmics Archiviert nach Abschluss eines Laufs die Metadaten des Laufs unter CloudWatch. Speichert die Laufdaten standardmäßig auf unbestimmte Zeit, sofern Sie die CloudWatch

Aufbewahrungsrichtlinie nicht ändern. CloudWatch Run-Ausgaben werden ebenfalls in Amazon S3 gespeichert, bis Sie sie löschen.

Eine der einstellbaren HealthOmics Servicekontingenten Optionen ist die maximum number of runs (active and inactive) in einer Region. HealthOmics speichert Run-Metadaten für bis zu dieser Anzahl von Durchläufen zur Verwendung durch die Konsole und API-Operationen (ListRuns und GetRun). Wenn Sie einen Lauf starten, können Sie den Parameter für den Ausführungsmodus so einstellen, dass er das Aufbewahrungsverhalten für den Lauf angibt. Der Parameter unterstützt die Werte REMOVE und RETAIN.

Wenn bei einem neuen Lauf, bei dem der Aufbewahrungsmodus auf REMOVE gesetzt ist, HealthOmics versucht wird, den Lauf hinzuzufügen, nachdem er bereits die maximale Anzahl von Durchläufen gespeichert hat, werden automatisch die Metadaten für den ältesten Lauf entfernt, für den der REMOVE-Modus festgelegt wurde. Diese Entfernung hat keine Auswirkungen auf die in CloudWatch oder Amazon S3 gespeicherten Daten.

RETAIN ist der Standardwert für den Run-Aufbewahrungsmodus. Bei Läufen in diesem Modus löscht das System die Ausführungsmetadaten nicht. Wenn die maximale Anzahl von Läufen HealthOmics erreicht ist, die alle auf RETAIN gesetzt sind, können Sie keine weiteren Läufe erstellen, bis Sie einige Läufe gelöscht haben.

Wenn Sie planen, einen Stapel mit mehr als der maximalen Anzahl von Durchläufen gleichzeitig auszuführen, stellen Sie sicher, dass der Aufbewahrungsmodus für die Ausführung auf REMOVE eingestellt ist. Andernfalls schlägt der Batch fehl, wenn HealthOmics versucht wird, den nächsten Lauf nach dem Maximum zu starten.

Zusätzliche Überlegungen zur Verwendung des REMOVE-Aufbewahrungsmodus:

- Wenn Sie REMOVE zum ersten Mal als Aufbewahrungsmodus verwenden, sollten Sie erwägen, einen oder mehrere Läufe zu löschen, die den RETIN-Modus verwenden, um Steckplätze freizugeben. Wenn Sie weitere REMOVE-Läufe starten, wird das automatische Entfernen übernommen, sodass genügend Steckplätze für neue Läufe verfügbar sind.
- Wenn Sie einen archivierten Lauf (oder eine Reihe von Läufen) erneut ausführen möchten, verwenden Sie das HealthOmics Rerun-CLI-Tool. Weitere Informationen und Beispiele zur Verwendung dieses Tools finden Sie unter Omics rerun im Tools-Repository. HealthOmics GitHub
- Wir empfehlen, für jeden Lauf einen eindeutigen Namen zu konfigurieren. Nach dem HealthOmics Entfernen eines Laufs können Sie die Konsole oder API nicht mehr verwenden, um den Namen oder die Lauf-ID des Laufs zu finden. Sie können sie jedoch verwenden, CloudWatch um nach

dem Namen der Ausführung zu suchen. Verwenden Sie daher eindeutige Namen, um die besten Suchergebnisse zu erzielen.

- Sie können den CloudWatch start-query Befehl verwenden, um Informationen zu einem archivierten Lauf abzurufen. Wenn der Runname nicht eindeutig ist, gibt die Abfrage möglicherweise mehrere Manifeste zurück. Die Parameter Startzeit und Endzeit definieren den Zeitraum für die Suche.

```
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>
```

Der start-query Befehl gibt eine Abfrage-ID zurück. Wenn die Abfrage-ID an den get-query-results Befehl übergeben wird, werden die Abfrageergebnisse zurückgegeben.

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

HealthOmics Eingaben ausführen

Wenn in der Workflow-Definition Eingabedateien für den Workflow oder die Workflow-Aufgaben angegeben sind, werden HealthOmics die Dateien auf einem Scratch-Volume bereitgestellt, das für die Workflow-Ausführung vorgesehen ist. Diese Eingabedateien sind schreibgeschützt, wodurch verhindert wird, dass Aufgaben potenzielle Eingaben für andere Aufgaben im Workflow ändern. Bei Verzeichnisimporten sind die Verzeichnisse ebenfalls schreibgeschützt.

Viele Genomikanwendungen gehen davon aus, dass sich die Indexdateien zusammen mit den Sequenzdateien befinden (z. B. eine `bai` Begleitdatei für eine Datei). `bam` Um Indexdateien einzubeziehen, geben Sie sie als Aufgabeneingaben in der Workflow-Definition an.

Themen

- Größe der Ausführungsparameter verwalten
- Amazon S3 S3-Eingabeparameterformate
- Status des Amazon S3 S3-Eingabearchivs

Größe der Ausführungsparameter verwalten

Wenn Sie einen Lauf starten, geben Sie die Eingaben für die Ausführung im JSON-Objekt oder in der JSON-Datei mit den Ausführungsparametern an. Sie können bis zu 50 KB an Ausführungsparametern für den Workflow angeben. Sie können die folgenden Techniken verwenden, um diese Größenbeschränkung einzuhalten:

- Verwenden Sie Verzeichnisimporte

  Um eine große Anzahl von Eingabedateien anzugeben, geben Sie einen Parameter als Amazon S3 S3-Speicherort an, der alle Dateien enthält, anstatt für jeden Dateispeicherort einen Parameter anzugeben. Weitere Informationen finden Sie im nächsten Thema (Amazon S3 S3-Eingabeparameterformate).
- Verwenden Sie ein Musterblatt

  Ein Musterblatt ist eine CSV- oder TSV-Datei mit einer Spalte für die Adresse fastq.gz (oder zwei für gepaarte Lesevorgänge) und zusätzlichen Spalten für Metadaten wie Probennamen. Sie geben das Musterblatt als Eingabeparameter für den Lauf an und nicht als Parameter für jede Eingabedatei.

  Ihr Workflow definiert, wie Ihr Musterblatt den Datenstrukturen im Workflow zugeordnet wird. Sie könnten zwar Code für Musterblätter in WDL und CWL schreiben, sie sind jedoch häufiger in. NextFlow Ein Beispiel finden Sie im Beispielblatt auf der GitHub nf-core-Website.

Amazon S3 S3-Eingabeparameterformate

Für einen Eingabeparameter, der einen Amazon S3 S3-Speicherort akzeptiert, kann der Parameter den Speicherort einer Datei oder eines ganzen Dateiverzeichnisses angeben. Die Verwendung eines Verzeichnisses hat die folgenden Vorteile:

- Komfort — Sie geben den Verzeichnisnamen als Parameter an. Sie listen nicht jeden Dateinamen auf.
- Kompaktheit — Die maximale Dateigröße des Eingabeparameters beträgt 50 KB. Wenn Sie eine lange Liste von Eingabedateinamen angeben, können Sie dieses Maximum überschreiten.

Amazon S3 ist ein flaches Objektspeichersystem und unterstützt daher keine Verzeichnisse. Sie gruppieren Dateien in einem „Verzeichnis“, indem Sie jeder Datei dasselbe Objektschlüsselpräfix

geben. Weitere Informationen zu Amazon S3 S3-Objektschlüsselpräfixen finden Sie unter Objekte mithilfe von Präfixen organisieren.

HealthOmics interpretiert den Wert des Eingabeparameters wie folgt:

- Wenn der Amazon S3 S3-Standort nicht mit einem Schrägstrich endet oder das Glob-Muster verwendet, wird HealthOmics erwartet, dass der Parameterwert der Schlüssel für ein Amazon S3 S3-Objekt ist.

  Sie geben beispielsweise an, file1.fastq `s3://myfiles/runs/inputs/a/file1.fastq` einzugeben
- Wenn der Amazon S3 S3-Standort mit einem Schrägstrich endet, wird der Parameterwert als Amazon S3 S3-Präfix HealthOmics interpretiert. Es lädt alle Amazon S3 S3-Objekte mit diesem Präfix.

  Sie können beispielsweise angeben, dass alle Objekte geladen werden `s3://myfiles/runs/inputs/a/` sollen, deren Schlüssel mit diesem Präfix beginnen.
- Für Nextflow wird HealthOmics teilweise das Glob-Muster für Amazon S3 URIs in Eingabeparametern unterstützt.

  Sie können beispielsweise angeben, dass alle .gz-Dateien eingegeben werden sollen`"s3://myfiles/runs/inputs/a/*.gz"`, deren Schlüssel mit diesem Präfix beginnen.

Nextflow-Behandlung von Glob-Mustern in Amazon S3 S3-Eingaben

Glob-Muster	HealthOmics Verhalten abgleichen	Hinweise
s3://bucket/directory/*.txt	`.txt`Entspricht allen Objekten in beliebiger Tiefe unter dem Präfix s3://bucket/directory/. For example, matches s3://bucket/directory/abc.txt or s3://bucket/directory/subDir/123.txt usw.	
s3://bucket/directory/**/*.txt	`.txt`Entspricht allen Objekten in beliebiger Tiefe unter dem	In S3 `**` entspricht dies`*`.

Glob-Muster	HealthOmics Verhalten abgleichen	Hinweise
	Präfix s3://bucket/directory/. For example, matches s3://bucket/directory/abc.txt or s3://bucket/directory/subDir/123.txt usw.	
s3://bucket/directory/{a,b}.txt	s3://bucket/directory/a.txt, s3://bucket/directory/b.txt	
s3://bucket/directory/?. txt	Findet Objekte im Präfixstamm, deren Dateiname aus einem einzelnen Zeichen besteht, gefolgt von `.txt`. Es entspricht beispielsweise 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://bucket/directory/0.txt, s3://bucket/directory/1.txt, ... ,s3://bucket/directory/9.txt	
s3://bucket/directory/[0-9].txt	s3://bucket/directory/1.txt, s3://bucket/directory/2.txt, s3://bucket/directory/3.txt	
s3://bucket/directory/[0-9].txt	s3://bucket/directory/b.txt, s3://bucket/directory/c.txt, ... ,s3://bucket/directory/Y.txt	

Sprachspezifische Behandlung von Doppelschrägstrichen in Amazon S3 S3-Eingaben

HealthOmics behält das native Engine-Verhalten für jede Workflow-Engine bei, wenn doppelte Schrägstriche in Amazon S3 verarbeitet werden URIs, sodass Sie bei der Migration zu keine

Änderungen an Ihren Workflows vornehmen müssen. HealthOmics In den folgenden Abschnitten wird beschrieben, wie jede Engine mit verschiedenen Szenarien umgeht.

WDL

Wenn der Eingabeparameter einen doppelten Schrägstrich in der Mitte oder am Ende des URI enthält, behält die WDL-Engine den doppelten Schrägstrich bei.

Eingabeparameter	Erwarteter Standort	
s3://myfiles/runs/inputs//file1.fastq	s3://1.fastq myfiles/runs/inputs//file	
s3:////myfiles/runs/inputs	s3://myfiles/runs/inputs//	

Nächster Ablauf

Wenn der Eingabeparameter einen doppelten Schrägstrich in der Mitte des URI enthält, behält die Nextflow-Engine den doppelten Schrägstrich bei. Bei einem doppelten Schrägstrich am Ende der URI löst die Nextflow-Engine ihn in einen einzigen Schrägstrich auf.

Eingabeparameter	Erwarteter Standort	
s3://myfiles/runs/inputs//file1.fastq	s3://1.fastq myfiles/runs/inputs//file	
s3://myfiles//runs/inputs//*.gz	s3://myfiles//runs/inputs//*.gz	
s3://myfiles//runs/inputs//	s3://myfiles//runs/inputs/	

CWL

Wenn der Eingabeparameter einen doppelten Schrägstrich in der Mitte oder am Ende des URI enthält, behält die CWL-Engine den doppelten Schrägstrich bei.

Eingabeparameter	Erwarteter Standort	
s3://myfiles// runs/inputs//file 1.fastq	s3://myfiles// 1.fastq runs/inpu ts//file	
s3://myfiles//runs/inputs//	s3://myfiles//runs/inputs//	

Status des Amazon S3 S3-Eingabearchivs

HealthOmics kann Amazon S3 S3-Objekte, die S3 liefert, in Echtzeit abrufen. Für Objekte, die sich in den folgenden archivierten Speicherzuständen befinden, restore die Objekte, für die sie verfügbar gemacht werden sollen HealthOmics:

- Flexible Retrieval- oder Deep Archive-Speicherklassen in Amazon S3 Glacier.
- Archived Access oder Deep Archive Access-Stufen im Rahmen von Intelligent Tiering.

Informationen zum Wiederherstellen von Objekten finden Sie unter Wiederherstellen eines archivierten Objekts im Amazon S3 S3-Benutzerhandbuch.

Lebenszyklus in einem HealthOmics Workflow ausführen

Sie können den Fortschritt eines Laufs verfolgen, indem Sie den Ausführungsstatus überwachen. HealthOmics aktualisiert den Ausführungsstatus, während ein Lauf seinen Lebenszyklus durchläuft.

Sie können den Ausführungsstatus mit einer der folgenden Methoden abrufen:

- Die HealthOmics Konsole zeigt den Status jedes Laufs auf der Runs Seite an.
- Der GetRun API-Vorgang gibt den aktuellen Ausführungsstatus zurück.
- Sie können den Ausführungsstatus mithilfe von EventBridge Ereignissen überwachen. Weitere Informationen finden Sie unter Verwenden EventBridge mit AWS HealthOmics.

Themen

Werte für den Ausführungsstatus

Wenn Sie einen Lauf starten, HealthOmics wird der Ausführungsstatus auf gesetztPending. Wenn der Lauf seinen Lebenszyklus durchläuft, HealthOmics aktualisiert er den Statuswert, sodass er seinen aktuellen Fortschritt wiedergibt.

> **Note**
>
> Für einen anderen Laufstatus als „Wird ausgeführt“ fallen für Sie keine Gebühren an. Details finden Sie im nächsten Abschnitt.

HealthOmics unterstützt die folgenden Werte für den Ausführungsstatus:

Ausstehend

Der Lauf befindet sich in der Warteschlange und wartet darauf, gestartet zu werden. Läufe verbleiben in der Regel für einen kurzen Zeitraum im Status Ausstehend, bevor sie beginnen.

- Läufe können länger im Status „Ausstehend“ verbleiben, wenn Sie viele Aufträge gleichzeitig einreichen.
- Läufe bleiben auch dann im Status Ausstehend, wenn Ihr Konto die maximale Anzahl gleichzeitiger Ausführungen erreicht hat.
- Ein Lauf verbleibt im Status Ausstehend, wenn der Lauf Teil einer Ausführungsgruppe ist, die einen ihrer Ressourcenmaximalwerte erreicht hat.
- Sie können die Ausführungsprioritäten so anpassen, dass bestimmte Läufe in der Warteschlange vor anderen gestartet werden. Weitere Informationen zur Ausführungspriorität finden Sie unterPriorität ausführen.

Wird gestartet

HealthOmics erstellt den Lauf und stellt die für den Lauf erforderlichen Ressourcen bereit (z. B. den temporären Run-Speicher und den Engine-Knoten).

- HealthOmics stellt zu Beginn des Laufs temporären Run-Speicher bereit und deprovisioniert den Run-Speicher, wenn der Lauf gestoppt wird.

In Ausführung

Ein Lauf verbleibt während des Importvorgangs, der Verarbeitung der einzelnen Aufgaben und des Exportvorgangs im Status Wird ausgeführt.

- HealthOmics importiert die Eingabedateien in das temporäre Run-Speicherdateisystem. Die Eingabedateien sind schreibgeschützt, um zu verhindern, dass Aufgaben die Eingaben für andere Aufgaben in einem Workflow ändern.
- Exportiert beim HealthOmics Dateiexport die Ausgabedateien aus dem Run-Speicherdateisystem an den S3-Speicherort.
- HealthOmics übermittelt die Ausführungs- und Aufgabenprotokolle CloudWatch in Echtzeit, solange der Ausführungsstatus „Wird ausgeführt“ lautet. Weitere Informationen finden Sie unter Meldet sich an CloudWatch.

Wird angehalten

Nach Abschluss des Exportvorgangs wechselt die Ausführung in den Status Stopp.

- HealthOmics beendet die Bereitstellung aller Ressourcen (einschließlich des Run-Speicherdateisystems und des Engine-Knotens).

Completed

Nach HealthOmics Abschluss der Deprovisionierung der Ressource wechselt der Rechenlauf in den Status Abgeschlossen.

- HealthOmics hat alle ausgeführten Aufgaben abgeschlossen und die Ausgabedaten fehlerfrei exportiert.
- Die Run-Ausgaben sind im angegebenen Amazon S3 S3-URI-Ausgabespeicherort verfügbar. HealthOmicsGeneriert für WDL und CWL eine Datei mit einer Zusammenfassung der Run-Ausgabe, die Informationen über die enthält. HealthOmics Ausgaben ausführen
- Die endgültigen Ausführungsmanifestprotokolle und Engine-Protokolle (falls zutreffend) sind unter verfügbar. CloudWatch
- Bei Läufen, die Aufgabenwiederholungen unterstützen, kann ein Lauf mit dem Status Abgeschlossen eine oder mehrere fehlgeschlagene Aufgaben enthalten. Solange eine Aufgabenwiederholung für jede fehlgeschlagene Aufgabe erfolgreich war, wird die Ausführung in den HealthOmics Status Abgeschlossen überführt. HealthOmics weist jedem Wiederholungsversuch eine neue Aufgaben-ID zu, sodass die Ausführung die Aufgaben IDs für die fehlgeschlagenen Versuche und den abgeschlossenen Versuch umfasst.

Fehlgeschlagen

HealthOmics ist auf einen oder mehrere Fehler gestoßen und es konnten nicht alle ausgeführten Aufgaben abgeschlossen werden.

- Ein fehlgeschlagener Lauf wechselt in den Status Stopp, während die Ressourcen HealthOmics nicht bereitgestellt werden.

Abgebrochen

Ein Benutzer hat eine Anfrage zum Abbruch des Laufs initiiert.

- HealthOmics stoppt alle laufenden Aufgaben und hebt die Bereitstellung aller Ressourcen auf.
- HealthOmics exportiert keine Laufausgabedaten, wenn ein Benutzer einen Lauf abbricht. Sie haben keinen Zugriff auf Zwischendateien für einen abgebrochenen Lauf.
- Für Ihr Konto fallen Gebühren für die Aufgaben und Ressourcen an, die der Lauf während des Status Running vor dem Abbruch verbraucht hat.
- Es fallen keine Gebühren an, wenn Sie einen Lauf mit dem Status „Ausstehend“ oder „Wird gestartet“ stornieren.

Die Aufgabe wird erneut versucht

HealthOmics unterstützt Aufgabenwiederholungen für Aufgaben, die aufgrund von Dienstfehlern fehlschlagen (5XX-HTTP-Statuscodes).

Wenn alle Aufgaben in der Ausführung irgendwann abgeschlossen werden, auch wenn sie Wiederholungen erforderten, wird die Ausführung auf HealthOmics Abgeschlossen umgestellt. HealthOmics weist jedem Wiederholungsversuch eine neue Aufgaben-ID zu, sodass die Ausführung die Aufgaben IDs für die fehlgeschlagenen Versuche und den abgeschlossenen Versuch umfasst.

Das standardmäßige Wiederholungsverhalten hängt davon ab, welche Definitionssprache der Workflow verwendet. Die Standardeinstellung für Nextflow ist keine Wiederholungen. Bei WDL und CWL werden bis zu zwei Wiederholungen einer fehlgeschlagenen Aufgabe versucht, aber Sie können die Aufgabenwiederholung für bestimmte Aufgaben oder für alle Aufgaben in einem Workflow deaktivieren. HealthOmics Die Wiederholung von Aufgaben ist nützlich, um zeitweise auftretende Servicefehler zu beheben. Sie könnten jedoch erwägen, eine Aufgabe zu deaktivieren, die idempotent ist.

Spezifische Informationen zu den einzelnen Workflow-Definitionssprachen finden Sie in den folgenden Themen:

- WDL — Konfigurieren Sie das Verhalten beim erneuten Versuch von Aufgaben in der Workflow-Definition. Siehe Verhalten beim erneuten Versuch von WDL-Aufgaben konfigurieren.
- Nextflow — Konfigurieren Sie das Verhalten beim erneuten Versuch von Aufgaben in der Nextflow-Konfigurationsdatei oder in der Workflow-Definition. Siehe Verhalten bei Wiederholungsversuchen von Nextflow-Aufgaben konfigurieren.

- CWL — Konfigurieren Sie das Verhalten beim erneuten Versuch von Aufgaben in der Workflow-Definition. Siehe Verhalten beim erneuten Versuch von CWL-Aufgaben konfigurieren.

Auswirkungen des Ausführungsstatus auf die Preisgestaltung

Für Ihr Konto können Gebühren anfallen, solange der Run-Status „Wird ausgeführt“ lautet. Während eines anderen Laufstatus fallen für Sie keine Gebühren an. Beispielsweise fallen keine Gebühren für Ressourcen an, wenn der Lauf gestartet oder gestoppt wird.

Ein Lauf mit dem Status Wird ausgeführt hat folgende Auswirkungen auf die Abrechnung:

- Für Ihr Konto fallen Gebühren für die Nutzung des Laufspeicher-Dateisystems an, solange der Ausführungsstatus „Wird ausgeführt“ lautet. Informationen zu den Laufspeichertypen finden Sie unter. Speichertypen in HealthOmics Workflows ausführen
- Für Ihr Konto fallen Gebühren für die Ausführung von Aufgaben an, die auf den Rechen- und Speicherressourcen basieren, die Sie für jede Aufgabe in der Workflow-Definition angegeben haben, und auf der Dauer der Aufgabe. Weitere Informationen finden Sie unter Rechen- und Speicheranforderungen für HealthOmics Aufgaben.
- Für jede Aufgabe gilt ein Mindestabrechnungsschwellenwert von einer Minute. Wenn Sie eine Aufgabe für weniger als eine Minute ausführen, fällt eine Gebühr für die Nutzung von mindestens einer Minute an. Wenn möglich, gruppieren Sie kleine Aufgaben, um die Kosten zu optimieren. Durch das Gruppieren von Aufgaben wird auch die Laufzeit reduziert, da vermieden wird, dass mehrere aufeinanderfolgende Aufgaben gleichzeitig ausgeführt werden.

Weitere Informationen zur Preisgestaltung finden Sie unter HealthOmics Preisgestaltung. HealthOmics

HealthOmics Ausgaben ausführen

Wenn ein WDL- oder CWL-Lauf abgeschlossen ist, enthalten die Ausgaben eine Ausgabeübersichtsdatei (im JSON-Format), in der alle durch den Lauf erzeugten Ausgaben aufgeführt sind. Sie können die Ausgabe-Zusammenfassungsdatei für folgende Zwecke verwenden:

- Ermitteln Sie programmgesteuert die Ausgabedateien, die durch den Lauf generiert wurden.
- Stellen Sie sicher, dass der Lauf alle erwarteten Ausgaben erzeugt hat.

Themen

- Führen Sie die Ausgabezusammenfassung für WDL aus
- Führen Sie die Ausgabezusammenfassung für CWL aus

Führen Sie die Ausgabezusammenfassung für WDL aus

Wenn ein WDL-Lauf abgeschlossen ist, HealthOmics wird eine Ausgabezusammenfassungsdatei mit dem Namen erstellt. output.json

Für jede Ausgabe des Workflows gibt es ein entsprechendes key/value Paar in der Datei. Der Schlüssel enthält den Workflow-Namen und den Ausgabenamen im folgenden Format:`WorkflowName.output_name`. Bei einer Dateiausgabe ist der Wert ein S3-URI, der auf den Ausgabeort in S3 verweist, an dem die Datei gespeichert ist. Bei einer Array [File] -Ausgabe ist der Wert ein Array von S3 URIs.

Das folgende Beispiel zeigt die output.json Datei für einen Workflow mit dem NamenBWAMappingWorkflow.

```
{
  "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"
  ]
}
```

Wenn der Workflow Ausgaben erzeugt, die keine Dateitypen haben (wie String, Int, Float oder Bool), ist der Feldwert ein JSON-Primitiv. Zum Beispiel:

```
{
  "MyWorkflow.my_int_ouput": 1,
  "MyWorkflow.my_bool_output": false,
    ...
}
```

Führen Sie die Ausgabezusammenfassung für CWL aus

Wenn ein CWL-Lauf abgeschlossen ist, HealthOmics wird eine Ausgabezusammenfassungsdatei mit dem folgenden Namen outputs.json erstellt:

```
{my-S3outputpath}/{runId}/{run-uuid}/logs/outputs.json
```

Die Ausgabe-Zusammenfassungsdatei enthält eine Liste von Ausgaben. Jede Ausgabe ist ein key/value Paar, wobei der Schlüssel der Name der Ausgabe ist. Der Wert ist ein Objekt, das die folgenden Eigenschaften enthält:

- location — Der vollqualifizierte Pfad zur Ausgabedatei
- basename — Der Dateiname-Teil des Pfads
- class — Der Typ der Ausgabe, normalerweise Datei
- Größe — Die Größe der Datei in Byte

Im folgenden Beispiel enthält die Datei output.json eine Liste mit zwei Ausgabedateien.

```
{
  "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
  }
```

```
}
```

Gründe für Fehler beim Ausführen

Wenn ein Lauf fehlschlägt, verwenden Sie den GetRunAPI-Vorgang, um den Grund für den Fehler abzurufen.

Überprüfen Sie den Grund für den Fehler, um herauszufinden, warum der Lauf fehlgeschlagen ist. In der folgenden Tabelle sind die einzelnen Fehlerursachen zusammen mit einer Beschreibung des Fehlers aufgeführt.

Failure reason (Fehlerursache)	Fehlerbeschreibung
ASSUME_ROLE_FAILED	HealthOmics hat keine Erlaubnis, die Rolle zu übernehmen. Geben Sie den HealthOmics Prinzipal in der Vertrauensstellung für die Rolle an.
CONTAINER_ERROR KANN NICHT GESTARTET WERDEN	Workflow-Aufgabe:, ID: Container mit Bild: konnte nicht gestartet *name* werden. *ID image name* Stellen Sie sicher, dass das Bild gültig ist, und versuchen Sie es erneut.
CANNOT_START_CONTAINER_SIZE_ERROR	Workflow-Aufgabe:, ID: Container mit Bild: konnte nicht gestartet werden. *name ID image name* Stellen Sie sicher, dass die Bildgröße weniger als 45 GiB (95 GiB für eine GPU-Instanz) beträgt, und versuchen Sie es erneut.
ECR_PERMISSION_ERROR	HealthOmics hat keine Berechtigung, auf die Bild-URI zuzugreifen. Vergewissern Sie sich, dass das private Amazon ECR-Repository existiert und dem HealthOmics Service Principal Zugriff gewährt wurde.
EXPORT_FAILED	Der Export ist fehlgeschlagen. Stellen Sie sicher, dass der Ausgabe-Bucket vorhanden ist und dass die Run-Rolle über Schreibberechtigungen für den Bucket verfügt.
FILE_SYSTEM_OUT_OF_SPACE	Das Dateisystem hat nicht genug Speicherplatz. Erhöhen Sie die Größe des Dateisystems und führen Sie es erneut aus.

Failure reason (Fehlerursache)	Fehlerbeschreibung
IMAGE_VERIFICATION_FAILURE	Das Bild konnte nicht verifiziert werden. *image name* Um das Problem zu beheben, versuchen Sie, das Bild abzurufen und es dann erneut in Ihr ECR-Repository zu übertragen.
IMPORT_FAILED	Der Import ist fehlgeschlagen. Vergewissern Sie sich, dass die Eingabedatei vorhanden ist und dass die Ausführungsrolle auf die Eingabe zugreifen kann.
INACTIVE_OMICS_STORAGE_RESOURCE	Die Speicher-URI befindet sich nicht im Status ACTIVE. HealthOmics Aktivieren Sie das Leseset und versuchen Sie es erneut. Weitere Informationen zur Aktivierung von Lesesätzen finden Sie unterAktivierung von Readsets in HealthOmics.
INPUT_URI_NOT_FOUND	Die angegebene URI existiert nicht:. *uri* Überprüfen Sie, ob der URI-Pfad existiert, und stellen Sie sicher, dass die Rolle auf das Objekt zugreifen kann.
INSTANCE_RESERVATION_FAILED	Die Instanzkapazität reicht nicht aus, um den Workflow-Lauf abzuschließen. Warten Sie und versuchen Sie erneut, den Workflow auszuführen.
UNGÜLTIGER_ECR_IMAGE_URI	Die Amazon ECR-Image-URI-Struktur ist nicht gültig. Geben Sie eine gültige URI ein und versuchen Sie es erneut.
INVALID_TASK_RESOURCE_VALUE	Die angeforderte GPU, CPU oder Arbeitsspeicher ist entweder zu hoch für die verfügbare Rechenkapazität oder sie unterschreitet den Mindestwert von 1 für die Aufgabe. *ID*
INVALID_URI_INPUT	Die URI-Struktur ist nicht gültig. *uri* Überprüfen Sie die URI-Struktur und versuchen Sie es erneut.
MODIFIED_INPUT_RESOURCE	Der angegebene URI *uri* wurde nach dem Start des Laufs geändert. Versuchen Sie den Lauf erneut.

Failure reason (Fehlerursache)	Fehlerbeschreibung
OUT_OF_MEMORY_ERROR	Für die Workflow-Aufgabe ist nicht genügend Arbeitsspeicher verfügbar. *ID* Erhöhen Sie den Speicherwert in der Workflow-Definition und versuchen Sie die Ausführung erneut.
RUN_TASK_FAILED	Die Ausführung ist fehlgeschlagen, weil die Aufgabe fehlgeschlagen ist. Verwenden Sie den GetRunTaskAPI-Vorgang und den Amazon CloudWatch Logs-Stream, um den Aufgabenfehler zu debuggen.
RUN_TIMED_OUT	Timeout nach Minuten ausführen. *number*
SERVICE_ERROR	Im Dienst ist ein vorübergehender Fehler aufgetreten. Versuchen Sie erneut, den Workflow auszuführen.
TASK_TIMED_OUT	Das *id* Zeitlimit für die Aufgabe wurde nach Sekunden überschritten. *number*
UNSUPPORTED_INPUT_SIZE	Die Gesamtgröße der Eingabe ist zu hoch. Verringern Sie die Eingabegröße und versuchen Sie es erneut.
WORKFLOW_RUN_FAILED	Die Workflow-Ausführung ist fehlgeschlagen. Überprüfen Sie den CloudWatch Logstream der Log-Engine:*ID*, um den Fehler zu debuggen.
WORKFLOW_VER_VALIDATION_FAILED	HealthOmics unterstützt die angeforderte Nextflow-Version nicht: --. *version* Die neueste unterstützte Version ist*version*. Ändern Sie Ihre Nextflow-Version auf eine unterstützte Version und versuchen Sie es erneut.
UNSUPPORTED_GPU_INSTANCE_TYPE	Der angeforderte Instanztyp wird in nicht unterstützt. *Region* Versuchen Sie den Lauf erneut mit einem GPU-Instanztyp, der in dieser Region unterstützt wird. Verfügbare Instanztypen sind*GPU instance types*.

Hinweise für nicht reagierende Läufe

Bei der Entwicklung neuer Workflows kann es vorkommen, dass Läufe oder bestimmte Aufgaben „hängen bleiben“ oder „hängen“, wenn es Probleme mit Ihrem Code gibt und Aufgaben Prozesse nicht ordnungsgemäß beenden können. Dies kann schwierig sein, Fehler zu beheben und catch, da es normal ist, dass Aufgaben über einen längeren Zeitraum ausgeführt werden. Folgen Sie den empfohlenen bewährten Methoden in den folgenden Abschnitten, um nicht reagierende Ausführungen zu verhindern und zu identifizieren.

Bewährte Methoden zur Vermeidung nicht reagierender Läufe

- Stellen Sie sicher, dass Sie alle in Ihrem Aufgabencode geöffneten Dateien schließen. Das Öffnen zu vieler Dateien kann gelegentlich zu Threading-Problemen innerhalb der Workflow-Engines führen.
- Hintergrundprozesse, die durch eine Workflow-Aufgabe erstellt wurden, sollten beendet werden, wenn die Aufgabe beendet wird. Wenn ein Hintergrundprozess jedoch nicht ordnungsgemäß beendet wird, müssen Sie diesen Prozess in Ihrem Aufgabencode explizit beenden.
- Stellen Sie sicher, dass Ihre Prozesse nicht wiederholt werden, ohne sie zu beenden. Dies kann dazu führen, dass die Ausführung nicht reagiert, und zur Behebung des Problems ist eine Änderung Ihres Workflow-Definitionscodes erforderlich.
- Stellen Sie eine angemessene Speicher- und CPU-Zuweisung für Ihre Aufgaben bereit. Analysieren Sie die CloudWatch Protokolle oder verwenden Sie die Führen Sie Analyzer aus Daten der erfolgreich abgeschlossenen Läufe Ihres Workflows, um sicherzustellen, dass Sie über eine optimale Rechenzuweisung verfügen. Verwenden Sie den Run `headroom` Analyzer-Parameter, um zusätzlichen Spielraum einzubeziehen und sicherzustellen, dass die Prozesse über ausreichende Ressourcen verfügen, um sie abzuschließen. Rechnen Sie mindestens 5% des zugewiesenen Speichers und der CPU mit ein, um Hintergrundprozesse im Betriebssystem zu berücksichtigen.
  - Erhöhen Sie außerdem die Größe der Instance-Bandbreite, wenn die Instance einen höheren Durchsatz benötigt. Bei EC2 Amazon-Instances mit weniger als 16 v CPUs (Größe 4xl und kleiner) kann es zu Durchsatzspitzen kommen. Weitere Informationen zum EC2 Amazon-Instance-Durchsatz finden Sie unter EC2 Verfügbare Instance-Bandbreite von Amazon.
- Stellen Sie sicher, dass Sie die richtige Dateisystemgröße für Ihre Läufe verwenden. Bei nicht reagierenden Läufen, die statischen Laufspeicher verwenden, sollten Sie erwägen, die Speicherzuweisung für statische Durchläufe zu erhöhen, um einen höheren I/O-Durchsatz und eine höhere Speicherkapazität im Dateisystem zu ermöglichen. Analysieren Sie das Ausführungsmanifest, um den maximalen Speicherplatz im Dateisystem zu ermitteln, und ermitteln Sie mit dem Run Analyzer, ob die Dateisystemzuweisung erhöht werden muss.

Bewährte Methoden zur Erkennung nicht reagierender Läufe

- Verwenden Sie bei der Entwicklung neuer Workflows eine Ausführungsgruppe, bei der das maximale Laufzeitlimit festgelegt ist, um außer Kontrolle geratenen Code abzufangen. Wenn ein Lauf beispielsweise 1 Stunde dauern sollte, ordnen Sie ihn einer Ausführungsgruppe zu, bei der nach 2 oder 3 Stunden (oder einem anderen Zeitraum, je nach Anwendungsfall) ein Timeout auftritt, um ungenutzte Jobs abzufangen. Wenden Sie außerdem einen Puffer an, um Abweichungen bei den Verarbeitungszeiten zu berücksichtigen.
- Richten Sie eine Reihe von Ausführungsgruppen mit unterschiedlichen maximalen Laufzeitlimits ein. Sie könnten beispielsweise einer Ausführungsgruppe, die die Läufe nach einigen Stunden beendet, kurze Läufe zuweisen, und einer Gruppe mit langen Ausführungen, die Läufe nach einigen Tagen beenden, basierend auf Ihrer erwarteten Workflow-Dauer.
- HealthOmics hat ein standardmäßiges Service-Limit für die maximale Ausführungsdauer von 604.800 Sekunden oder 7 Tagen, das durch eine Anfrage im Quotas-Tool angepasst werden kann. Fordern Sie eine Erhöhung des Servicelimits für dieses Kontingent nur an, wenn Sie Läufe mit einer Dauer von etwa einer Woche haben. Wenn Sie eine Mischung aus kurzen und langen Läufen haben und keine Ausführungsgruppen verwenden, sollten Sie erwägen, die Läufe mit langer Laufzeit in einem separaten Konto mit einem höheren Dienstlimit für die maximale Ausführungsdauer abzulegen.
- Suchen Sie in den CloudWatch Protokollen nach Aufgaben, von denen Sie vermuten, dass sie nicht reagieren. Wenn eine Aufgabe normalerweise reguläre Protokollanweisungen ausgibt und dies über einen längeren Zeitraum nicht getan hat, ist die Aufgabe wahrscheinlich hängen geblieben oder eingefroren.

Was tun, wenn eine Ausführung nicht reagiert

- Stornieren Sie den Lauf, um zusätzliche Kosten zu vermeiden.
- Prüfen Sie in den Task-Protokollen, ob Prozesse nicht korrekt beendet werden konnten.
- Untersuchen Sie die Motorprotokolle, um abnormales Motorverhalten zu identifizieren.
- Vergleichen Sie die Aufgaben- und Engine-Protokolle des nicht reagierenden Laufs mit denen identischer, erfolgreich abgeschlossener Läufe. Auf diese Weise können Sie alle Unterschiede ermitteln, die möglicherweise zu dem nicht reagierenden Verhalten geführt haben.
- Wenn Sie nicht in der Lage sind, die Ursache zu ermitteln, melden Sie eine Support-Anfrage und fügen Sie Folgendes hinzu:

- ARN des festgefahrenen Laufs und ARN eines identischen Laufs, der erfolgreich abgeschlossen wurde.
- Engine-Protokolle (verfügbar, sobald der Lauf abgebrochen wurde oder fehlschlägt)
- Aufgabenprotokolle für die nicht reagierende Aufgabe. Zur Fehlerbehebung benötigen wir nicht für alle Aufgaben im Workflow Aufgabenprotokolle.

Aufgabenlebenszyklus in einem HealthOmics Lauf

Eine Aufgabe ist ein einzelner Prozess innerhalb eines Laufs. HealthOmics ordnet jede Aufgabe in Ihrem Workflow einem Omics Computing-Instanztyp zu, der am besten zu den für die Aufgabe benötigten Ressourcen passt. Sie geben die erforderlichen Ressourcen in der Workflow-Definition an. Weitere Informationen finden Sie unterRechen- und Speicheranforderungen für HealthOmics Aufgaben.

HealthOmics stellt temporären Ausführungsspeicher zur Verfügung, den die Aufgabe verwenden kann. HealthOmics kopiert die Eingabedateien für die Aufgabe als schreibgeschützte Dateien in den temporären Ausführungsspeicher. HealthOmics stellt symbolische Links bereit, sodass die Aufgabe vom Arbeitsverzeichnis aus auf die Eingabedateien zugreifen kann. Die Aufgabe hat nur Zugriff auf die Dateien, die Sie in der Workflow-Definitionsdatei deklarieren.

Werte für den Aufgabenstatus

Sie können den Fortschritt einer Aufgabe verfolgen, indem Sie den Aufgabenstatus überwachen. Wenn Sie eine Ausführung starten, wird der Aufgabenstatus Pending für jede Aufgabe in der Ausführung auf HealthOmics festgelegt. Wenn die Aufgabe gestartet wird und ihren Lebenszyklus durchläuft, wird der Statuswert HealthOmics aktualisiert, sodass er ihren aktuellen Fortschritt wiedergibt.

Sie können den Aufgabenstatus mit einer der folgenden Methoden abrufen:

- Die HealthOmics Konsole zeigt den Status jeder Aufgabe in einer Ausführung auf der Run details Seite an.
- Der GetRunTask API-Vorgang gibt den Aufgabenstatus zurück.
- Sie können den Aufgabenstatus mithilfe von EventBridge Ereignissen überwachen. Weitere Informationen finden Sie unter Verwenden EventBridge mit AWS HealthOmics.

Sie können den aktuellen Status einer Aufgabe mithilfe der GetRunTask API-Operation abrufen. Die HealthOmics Konsole zeigt den Status für jede Aufgabe in einer Ausführung auf der Run details Seite an.

HealthOmics unterstützt die folgenden Statuswerte für Aufgaben:

Ausstehend

Ihre Aufgabe befindet sich in der Warteschlange und wartet darauf, gestartet zu werden. Aufgaben bleiben für einen kurzen Zeitraum ausstehend, bevor sie beginnen.

- Aufgaben bleiben ausstehend, auch wenn Ihr Konto die maximale Anzahl gleichzeitiger Aufgaben erreicht hat.
- Aufgaben bleiben ausstehend, wenn die Ausführung Teil einer Ausführungsgruppe ist, deren Ressourcenmaximalwerte erreicht wurden.
- Sie können die Ausführungsprioritäten so anpassen, dass bestimmte Läufe in der Warteschlange und ihre Aufgaben vor anderen Läufen in der Warteschlange beginnen. Weitere Informationen zur Ausführungspriorität finden Sie unter Priorität ausführen

Wird gestartet

HealthOmics erstellt die Aufgabe und stellt die für die Aufgabe erforderlichen Ressourcen bereit, z. B. den Workflow-Aufgabenknoten.

In Ausführung

Der Aufgabenstatus ist Wird ausgeführt, während die Aufgabe bearbeitet HealthOmics wird.

Wird angehalten

Nach Abschluss der Aufgabenverarbeitung und dem Export der Ausgabedaten wechselt die Aufgabe in den Status Beendet.

- HealthOmics beendet die Bereitstellung des Workflow-Aufgabenknotens.

Completed

HealthOmics hat die Bearbeitung der Aufgabe abgeschlossen und die Ausgabedaten in das Run-Storage-Dateisystem übertragen.

Fehlgeschlagen

HealthOmics ist bei der Bearbeitung der Aufgabe auf einen Fehler gestoßen und wurde nicht abgeschlossen.

- Die Aufgabe wechselt in den Status Stopp (die HealthOmics Bereitstellung der Ressourcen wird aufgehoben) und anschließend in den Status Fehlgeschlagen.
- Wenn es sich bei dem Fehler um einen Dienstfehler handelt (HTTP-Statuscode 5XX) und der Workflow Wiederholungsversuche für diese Aufgabe unterstützt, wird HealthOmics versucht, die Aufgabe erneut zu verarbeiten. HealthOmics weist dem Wiederholungsversuch eine neue Aufgaben-ID zu.

Abgebrochen

HealthOmics stoppt die Aufgabe nach einer vom Benutzer initiierten Aufforderung, die Ausführung abzubrechen.

- Die Aufgabe wechselt in den Status Beendet (Bereitstellung der Ressourcen HealthOmics wird aufgehoben) und anschließend in den Status Abgebrochen.

Problembehandlung bei Workflow-Aufgaben

Im Folgenden finden Sie bewährte Methoden und Überlegungen zur Problembehandlung bei Ihren Aufgaben.

- Aufgabenprotokolle hängen von der Aufgabe `STDERR` ab `STDOUT` und werden von ihr erstellt. Wenn die in der Aufgabe verwendete Anwendung keines dieser Elemente erzeugt, wird es kein Aufgabenprotokoll geben. Verwenden Sie Anwendungen im `verbose` Modus, um das Debuggen zu unterstützen.
- Verwenden Sie den Bash-Befehl, um die Befehle, die in einer Aufgabe ausgeführt werden, zusammen mit ihren interpolierten Werten anzuzeigen. `set -x` Auf diese Weise können Sie feststellen, ob die Aufgabe die richtigen Eingaben verwendet, und ermitteln, wo Fehler die Ausführung der Aufgabe möglicherweise verhindert haben.
- Verwenden Sie den `echo` Befehl, um die Werte von Variablen in `STDOUT` oder auszugebenSTDERR. Auf diese Weise können Sie überprüfen, ob sie wie erwartet festgelegt wurden.
- Verwenden Sie Befehle wie `ls -l <name_of_input_file>`, um zu überprüfen, ob Eingaben vorhanden sind und die erwartete Größe haben. Wenn dies nicht der Fall ist, könnte dies auf ein Problem mit einer vorherigen Aufgabe hinweisen, die aufgrund eines Fehlers leere Ausgaben erzeugt hat.
- Verwenden Sie den Befehl `df -Ph . | awk 'NR==2 {print $4}'` in einem Aufgabenskript, um den Speicherplatz zu ermitteln, der derzeit für die Aufgabe verfügbar ist, und um Situationen

zu identifizieren, in denen Sie den Workflow möglicherweise mit zusätzlicher Speicherzuweisung ausführen müssen.

Wenn Sie einen der oben genannten Befehle in ein Taskskript aufnehmen, wird davon ausgegangen, dass der Task-Container auch diese Befehle enthält und dass sie sich in `path` der Container-Umgebung befinden.

Führen Sie die Optimierung für einen privaten HealthOmics Workflow durch

Sie können Läufe im Hinblick auf Gesamtkosten, Gesamtlaufzeit oder eine Kombination aus beidem optimieren. HealthOmics stellt Daten und Tools bereit, die Sie bei Entscheidungen zur Laufoptimierung unterstützen. Die Run-Optimierung gilt nicht für Ready2Run-Workflows, da Sie keine Kontrolle darüber haben, wie der Service die Ressourcenbereitstellung für diese Workflows verwaltet.

Der erste Schritt besteht darin, den aktuellen Ressourcenverbrauch und die Kosten für die laufenden Aufgaben zu verstehen und anschließend Methoden zur Optimierung der Ausführungskosten und der Leistung anzuwenden.

Themen

Führen Sie Analyzer aus

HealthOmics bietet ein Open-Source-Tool namens Run Analyzer. Dieses Tool extrahiert Informationen zur Ressourcennutzung auf Aufgabenebene für einen Lauf und schlägt Optimierungsmöglichkeiten in Bezug auf Kosten und Rechenleistung vor.

> **Note**
>
> Run Analyzer schätzt die Aufgabenkosten und potenziellen Kosteneinsparungen auf der Grundlage der AWS Listenpreise zum Zeitpunkt der Ausführung des Tools. Beurteilen Sie die Optimierungsempfehlungen und implementieren Sie diejenigen, die für Ihre Anwendungsfälle

sinnvoll sind. Testen Sie die Optimierungen, die Sie verwenden, um sicherzustellen, dass sie für Ihren Lauf funktionieren.

Run Analyzer führt die folgenden Aufgaben aus:

- Wertet Speicher- und Rechenengpässe aus.
- Identifiziert Aufgaben, bei denen zu viel Arbeitsspeicher oder CPU zur Verfügung steht, und empfiehlt neue Instanzgrößen, mit denen die Kosten gesenkt werden können.
- Berechnet Kostenschätzungen für einzelne Aufgaben und berechnet die potenziellen Kosteneinsparungen, wenn Sie die Empfehlungen anwenden.
- Bietet Ihnen eine Zeitleistenansicht der Aufgaben, sodass Sie die Aufgabenabhängigkeiten und die Verarbeitungsreihenfolge überprüfen können. Die Zeitleiste hilft Ihnen auch dabei, Aufgaben mit langer Laufzeit zu identifizieren.
- Enthält Empfehlungen zur Dateisystemgröße für den Run-Speicher.
- Zeigt die Bereitstellungszeiten für Aufgaben an, sodass Sie Bereiche identifizieren können, in denen große Containerlasten die Bereitstellungszeit verlangsamen könnten.
- Das Tool enthält einen Eingabeparameter (Headroom), mit dem Sie die Aggressivität der Optimierungsempfehlungen steuern können.

Die folgenden Abschnitte enthalten spezifische Vorschläge für die Verwendung von Run Analyzer zur Optimierung von Läufen.

Ermitteln Sie die Betriebskosten

Sie können die folgenden Methoden und Richtlinien verwenden, um die Betriebskosten zu ermitteln:

- Gehen Sie wie folgt vor, um die gesamten Betriebskosten für einen Abrechnungszeitraum anzuzeigen:

  1. Öffnen Sie die Billing and Cost Management-Konsole und wählen Sie Rechnungen aus.
  2. Erweitern Sie unter Gebühren nach Service den Eintrag Omics.
  3. Erweitern Sie die Region und sehen Sie sich dann die Kosten für all Ihre Läufe an, aufgeschlüsselt nach Omics-Instanztyp, Run-Speichertyp und Ready2Run-Workflow.
- Gehen Sie wie folgt vor, um einen Kostenbericht zu erstellen, der Informationen für jeden Lauf enthält:

1. Öffnen Sie die Billing and Cost Management-Konsole und wählen Sie Datenexporte.
2. Wählen Sie Erstellen, um einen neuen Datenexport zu erstellen.
3. Geben Sie einen Exportnamen für den Datenexport ein. Behalten Sie die Standardwerte der anderen Felder bei, um einen CUR-Bericht (Kosten und Nutzung) zu erstellen.
4. Wählen Sie unter Zeitgranularität die Option stündlich oder täglich aus.
5. Führen Sie unter Speichereinstellungen für den Datenexport die folgenden Konfigurationsschritte durch:

   a. Konfigurieren Sie einen Amazon S3 S3-Bucket für den Datenexport.

   b. Wählen Sie für die Dateiversionierung aus, ob die bestehende Exportdatei überschrieben oder jedes Mal eine neue Datei erstellt werden soll.

   Das System generiert den ersten Bericht innerhalb der nächsten 24 Stunden und die nachfolgenden Berichte einmal täglich.
6. Weitere Informationen zum Erstellen des Datenexports finden Sie unter Erstellen von Datenexporten im Benutzerhandbuch für AWS Datenexporte.

- Du kannst deine Läufe taggen, um die Kosten nach Kategorien, z. B. nach Team oder Projekt, zu überwachen und zu optimieren. Wenn Sie Tags verwenden, gehen Sie wie folgt vor, um die Kosten für die einzelnen Durchläufe nach Tag-Kategorien anzuzeigen:

  1. Öffnen Sie die Billing and Cost Management-Konsole und wählen Sie Cost Explorer.
  2. Wählen Sie unter Berichtsparameter > Gruppieren nach die Option Tag als Dimension und wählen Sie den gewünschten Tagnamen aus.
- Um die Ressourcennutzung für Aufgaben zu sehen, schauen Sie sich das Run-Manifest an, in CloudWatch dem Sie angemeldet sind. Weitere Informationen finden Sie unter Überwachung HealthOmics mit CloudWatch Protokollen.
- Verwenden Sie das Führen Sie Analyzer aus Tool, um Informationen zur Nutzung der Aufgabenressourcen für einen Lauf zu extrahieren.

Ermitteln Sie die Nutzung der Laufzeit

Sie können die folgenden Methoden verwenden, um die Laufzeitnutzung zu untersuchen:

- Auf der Seite „Ausführungen“ der Konsole können Sie die Gesamtlaufzeit für einen Lauf anzeigen.

- Auf der Seite mit den Ausführungsdetails können Sie die folgenden Elemente anzeigen:
  - Die Gesamtlaufzeit für einen Lauf anzeigen.
  - Zeigen Sie die Laufzeit für jede Aufgabe in der Ausführung an.
  - Wählen Sie einen der Links, um die Protokolle in Amazon S3 oder die Run-Logs oder Run-Manifest-Logs einzusehen CloudWatch.
- Wählen Sie in der Liste „Aufgaben ausführen“ den Link „Protokolle anzeigen“ für eine Aufgabe, in der Sie die Aufgabenprotokolle anzeigen möchten CloudWatch.
- Die Antwort auf den `listRuns` API-Vorgang umfasst die Start- und Endzeit der Ausführung, sodass Sie die Gesamtlaufzeit berechnen können.
- Das Führen Sie Analyzer aus Tool zeigt die Dauer der Aufgaben in einer Zeitleistenansicht an. Dieses Tool bietet eine visuelle Darstellung der Reihenfolge der Aufgabenverarbeitung, die Sie der erwarteten Reihenfolge zuordnen können.

Methoden zur Optimierung von Läufen

HealthOmics stellt Ressourcen für die Datenbereitstellung (wie Datenimporte und Datenexporte) automatisch bereit, verwaltet und optimiert. HealthOmics startet auch die Workflow-Engine für Ihren Workflow und führt sie aus. Sie können jedoch die Startzeiten der Ausführung, die Startzeiten von Aufgaben und die Gesamtlaufzeit von Aufgaben beeinflussen, indem Sie verschiedene Ausführungskonfigurationen festlegen. Ihr allgemeiner Ansatz bei der Workflow-Definition und -Gestaltung wirkt sich auch auf die Laufzeit der Aufgaben aus. In der folgenden Liste werden Faktoren beschrieben, die sich auf die Ausführung und die Aufgabenleistung auswirken können:

Speichertyp ausführen

Der Speichertyp „Run“ wirkt sich auf die Laufleistung und die Bereitstellungszeit aus. Dynamic Run Storage bietet eine schnellere Bereitstellung und es geht nie der Arbeitsspeicher aus, da er dynamisch mit Ihren Laufspeicheranforderungen skaliert wird. Dynamic Run Storage eignet sich auch gut für Workflows in der Entwicklung, bei denen Sie häufig einen Workflow starten und beenden können, um Probleme zu beheben.

Statischer Run-Speicher erfordert längere Bereitstellungszeiten für das Dateisystem, kann aber einige Läufe schneller abschließen, in der Regel, wenn die Läufe eine hohe Aufgabenparallelität aufweisen oder mehr als 9,6 TiB Dateisystemkapazität erfordern. Static Run Storage eignet sich gut für Workflows mit langer Laufzeit und hohen Anforderungen. I/O

Um Ihnen bei der Bewertung der Kosten und der Leistung der einzelnen Speichertypen für einen bestimmten Lauf zu helfen, können Sie A/B-Tests durchführen, um herauszufinden, welcher Laufspeichertyp die bessere Leistung bietet. Erwägen Sie außerdem, dynamischen Run-Storage für Ihre Entwicklungszyklen zu verwenden und anschließend statischen Run-Speicher für Produktionsläufe in großem Maßstab zu verwenden.

Weitere Informationen zu Run-Storage-Typen Speichertypen in HealthOmics Workflows ausführen

Statischer Speicher mit übermäßiger Bereitstellung

Wenn die Berechnung Ihrer Workflow-Aufgaben durch 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 Engpässe eingeschränkt wird, kann eine Erhöhung der Dateisystemgröße zur Verkürzung der Task-Laufzeit die Gesamtkosten senken.

Reduzieren Sie die Größe von Container-Images

HealthOmics Lädt beim Start jeder Aufgabe den Container, den Sie für die Aufgabe angegeben haben. Das Laden größerer Container dauert länger. Optimieren Sie Ihre Container so klein wie möglich, um die Effizienz beim Starten neuer Aufgaben zu verbessern. Wenn Sie Ihren Containern große Datensätze hinzufügen, sollten Sie erwägen, die Datensätze in S3 zu speichern und Ihren Workflow die Daten aus S3 importieren zu lassen. Die maximal HealthOmics unterstützten Containergrößen finden Sie unter. HealthOmics Workflow-Kontingente mit fester Größe

Aufgabengröße

Sie können kleine, aufeinanderfolgende Aufgaben zu einer einzigen Aufgabe zusammenfassen, um Zeit bei der Aufgabenbereitstellung zu sparen. Außerdem wird HealthOmics eine Mindestdauer von einer Minute für Aufgaben berechnet, sodass die Kombination von Aufgaben die Kosten senken kann. Im Rahmen der kombinierten Aufgabe können Sie möglicherweise Unix-Pipes verwenden, um die I/O Kosten für das Serialisieren und Deserialisieren von Dateien zu vermeiden.

Komprimierung von Dateien

Vermeiden Sie eine übermäßige Komprimierung von Workflow-Zwischendateien. Die meisten Genomikformate verwenden die Komprimierung „Gzip“ oder „Block Gzip“. Das Dekomprimieren der Task-Eingabedatei und das erneute Komprimieren der Task-Ausgabedatei können einen großen Prozentsatz der gesamten CPU-Auslastung der Aufgabe beanspruchen. Bei

einigen Genomikanwendungen können Sie die Komprimierungsstufe bei der Serialisierung von Ausgaben festlegen. Durch die Reduzierung der Komprimierungsstufe können Sie die CPU-Zeit reduzieren, obwohl größere Dateien die Zeit erhöhen, die für das Schreiben auf die Festplatte aufgewendet wird. Abhängig von der Aufgabe und der Anwendung können Sie die optimale Komprimierungsstufe für Zwischendateien finden, die zu der kürzesten Laufzeit führen. Wir empfehlen, dass Sie sich zunächst auf die Aufgaben mit den größten Ausgabedateien konzentrieren. Eine Komprimierungsstufe von 2 eignet sich gut für mehrere Szenarien. Sie können mit dieser Stufe für Ihren Anwendungsfall beginnen und die Ergebnisse vergleichen, indem Sie andere Komprimierungsstufen ausprobieren.

Anzahl der Threads

Wenn Sie in Ihrer Aufgabendefinition Threads angeben, setzen Sie die Anzahl der Threads auf den gleichen Wert wie die Anzahl der angeforderten CPUs v.

Geben Sie Rechenleistung und Arbeitsspeicher an

Wenn Sie in Ihrer Aufgabe keine Speicher- oder Rechenressourcen angeben, HealthOmics weist Sie den kleinsten Instanztyp (`omics.c.large`) als Standard zu. Geben Sie Ihre Speicher- und Rechenanforderungen explizit HealthOmics an, wenn Sie einen größeren Instance-Typ zuweisen möchten.

HealthOmics weist die Anzahl der von Ihnen angeforderten V-CPUs, Speicher- und GPU-Ressourcen zu. Wenn Sie beispielsweise nach 15V CPUs und 33GiB fragen, wird eine HealthOmics omics.m.4xl-Instanz (16 V, 64 GB) für Ihre Aufgabe zugewiesenCPUs, aber Ihre Aufgabe kann nur 15 V und 33GiB verwenden. CPUs Daher empfehlen wir, dass Sie v- und Speicherressourcen anfordern, die einer Omics-Instanz entsprechen. CPUs

Mehrere Proben in einem Durchlauf stapeln

Da die Dateisystembereitstellung zu Beginn des Laufs einige Zeit in Anspruch nimmt, können Sie Bereitstellungszeit sparen, indem Sie mehrere Samples in einem Lauf stapeln. Berücksichtigen Sie die folgenden Faktoren, bevor Sie sich für diesen Ansatz entscheiden:

- Eine einzige fehlerhafte Probe kann dazu führen, dass ein Workflow fehlschlägt, sodass die Anzahl fehlgeschlagener Workflows durch das Stapeln von Proben erhöht werden kann. Wenn Sie nicht sicher sind, ob Ihr Workflow in den meisten Fällen erfolgreich sein wird, könnte ein Durchlauf pro Probe der bessere Ansatz sein.
- HealthOmics weist einem Lauf ein Speicherdateisystem für den gesamten Workflow zu. Achten Sie bei einem Probenstapel darauf, eine ausreichend große Menge an Laufspeicher anzugeben, um alle Proben verarbeiten zu können.

- Es gibt eine maximale Menge an Laufspeicher pro Workflow, sodass die Anzahl der Proben, die Sie dem Stapel hinzufügen können, möglicherweise eingeschränkt wird.
- Die Mindestgröße des Laufspeichers beträgt 1,2 TiB, sodass die Batchverarbeitung die Kosten senken kann, wenn der Workflow viel weniger Speicherplatz als das Minimum für jede Probe benötigt.
- Der Run-Speicher kann mehrere gleichzeitige Verbindungen verarbeiten, sodass es nicht zu Engpässen kommen sollte, wenn mehrere Aufgaben denselben Run-Speicher verwenden. I/O
- Jeder Lauf hat seinen eigenen Satz von Tags. Wenn Sie Workflows mit Informationen für die Budgetierung oder Nachverfolgung kennzeichnen, ist es möglicherweise besser, separate Läufe zu verwenden.
- IAM-Rollen gelten für den gesamten Lauf. Jeder Benutzer hat Zugriff auf alle Daten für eine Reihe von Proben. Durch die Trennung von Workflows können Sie detailliertere Berechtigungen verwenden.
- HealthOmics legt Kontingente auf Kontoebene für die maximale Anzahl gleichzeitiger Workflows und die maximale Anzahl gleichzeitiger Aufgaben in einem Workflow fest. Informationen darüber, wie Sie eine Erhöhung dieser Kontingente beantragen können, finden Sie unter. HealthOmics Servicekontingenten

Verwenden Sie Parameter für Container-Images

Parametrisieren Sie Ihre Container-Images, anstatt sie URIs in den Workflow einzubetten. Wenn es sich um Ausführungsparameter handelt, wird HealthOmics überprüft, ob die Ausführung Zugriff auf Ihre Container hat, bevor die Ausführung beginnt. Andernfalls schlägt die Aufgabe während der Ausführung fehl, wenn Ihnen Gebühren für abgeschlossene Aufgaben entstanden sind. Da es sich um parametrisierte Eingaben handelt, wird außerdem eine Prüfsumme im Ausführungsmanifest HealthOmics generiert, wodurch die Herkunft der Ausführung verbessert wird.

Verwenden Sie einen Linter

Verwenden Sie einen Linter, um häufig auftretende Workflow-Fehler zu finden, bevor Sie einen neuen Workflow ausführen. Weitere Informationen finden Sie unter Der Arbeitsablauf blinkt ein HealthOmics.

Wird verwendet EventBridge , um Probleme zu kennzeichnen

Verwenden Sie EventBridge maßgeschneiderte Benachrichtigungen, catch Anomalien zu erkennen, die für Ihre Geschäftslogik spezifisch sind.

Verwenden Sie Sequenzspeicher

Erwägen Sie die Verwendung eines Sequenzspeichers für Ihre Quelldaten, um Speicherkosten zu sparen. Weitere Informationen finden Sie im HealthOmics Blogbeitrag Omics-Daten kostengünstig in beliebiger Größenordnung speichern.

Auswirkung der Dateigrößenabweichung zwischen den Durchläufen

Benutzer entwerfen und testen Läufe häufig mit einem kleinen Satz von Testdaten und stoßen dann bei Produktionsläufen auf eine Vielzahl von Daten mit erheblichen Abweichungen in der Dateigröße. Achten Sie darauf, diese Varianz zu berücksichtigen, wenn Sie den Lauf optimieren.

In der folgenden Liste werden Optimierungsempfehlungen für Situationen beschrieben, in denen die Dateigrößen stark variieren:

Variieren Sie die Dateigrößen in Ihren Testdaten

Versuchen Sie, während der Entwicklung Testdaten zu verwenden, die eine repräsentative Varianz aufweisen.

Verwenden Sie Run Analyzer

Verwenden Sie das Tool Run Analyzer für eine Vielzahl von Stichproben, um Abweichungen bei den Datengrößen zu berücksichtigen.

Sie können den Run Analyzer verwenden, um die Varianz zwischen Durchläufen in Ihren Produktionsdatenstichproben zu verstehen. Verwenden Sie `--batch` den Modus in Run Analyzer, um Statistiken für einen Stapel von Durchläufen zu generieren und die maximalen Rechenressourcen zu analysieren, die für die Behandlung von Ausreißern in Ihren Datensätzen erforderlich sind.

Sie können Run Analyzer beispielsweise eine vollständige Datenflusszelle im Batch-Modus zur Verfügung stellen, um die maximale vCPU- und Speicherauslastung für die gesamte Flusszelle zu ermitteln.

Reduzieren Sie die Größenvarianz der Eingabe-Datasets

Wenn Sie eine hohe Varianz bei den Stichprobenumfängen feststellen, können Sie die Stichproben vorab teilen HealthOmics und für jeden Stapel unterschiedliche Dateisystemgrößen auswählen, um bei den laufenden Speicherkosten zu sparen.

Verwenden Sie in WDL die `size` Funktion, um die Ressourcenzuweisung für einzelne Aufgaben für große und kleine Stichproben aufzuteilen. Wenden Sie diese Strategie auf Ihre teuersten Aufgaben an, um die größtmögliche Wirkung zu erzielen.

Verwenden Sie in Nextflow bedingte Ressourcen für die stufenweise Zuweisung von Ressourcen auf der Grundlage der Dateigröße oder des Dateinamens. Weitere Informationen finden Sie unter Bedingte Prozessressourcen auf der GitHub Nextflow-Website.

Optimiere nicht zu früh

Finalisieren Sie Ihren Workflow-Code und Ihre Workflow-Logik, bevor Sie in umfangreiche Maßnahmen zur Leistungsoptimierung investieren. Eine Änderung Ihres Codes kann erhebliche Auswirkungen auf die benötigten Ressourcen haben. Wenn Sie einen Lauf zu früh im Entwicklungsprozess optimieren, kann dies zu einer Überoptimierung führen oder Sie müssen möglicherweise erneut optimieren, falls sich die Workflow-Definition später ändert.

Führen Sie das Run Analyzer-Tool regelmäßig erneut aus

Wenn Sie im Laufe der Zeit Änderungen an Ihrer Workflow-Definition vornehmen oder wenn sich Ihre Stichprobenvarianz ändert, führen Sie das Tool Run Analyzer regelmäßig aus, damit Sie weitere Optimierungen vornehmen können.

Methoden zur Optimierung der Parallelität von Ressourcen

HealthOmics bietet die folgenden Funktionen, mit denen Sie die Kosten kontrollieren und verwalten können, wenn die Verarbeitung in großem Maßstab ausgeführt wird:

- Verwenden Sie Ausführungsgruppen, um Ihre Kosten und den Ressourcenverbrauch zu kontrollieren. Sie können in der Ausführungsgruppe Höchstwerte für die Anzahl gleichzeitiger Läufe, v CPUs GPUs, und die Gesamtlaufzeit pro Aufgabe festlegen. Wenn separate Teams oder Gruppen dasselbe Konto verwenden, können Sie für jedes Team eine separate Ausführungsgruppe erstellen. Sie können die Ressourcennutzung und die Kosten pro Team kontrollieren und die Höchstwerte für die Laufgruppe konfigurieren. Weitere Informationen finden Sie unter HealthOmics Run-Gruppen verwenden.
- Während der Entwicklung können Sie eine separate Ausführungsgruppe mit niedrigeren Maximalwerten konfigurieren, um außer Kontrolle geratene Aufgaben abzufangen.
- Service Quotas tragen auch dazu bei, Ihr Konto vor übermäßigen Ressourcenanforderungen zu schützen. Informationen zu Service Quotas, einschließlich der Beantragung von Kontingentwerterhöhungen, finden Sie unter HealthOmics Servicekontingenten

Operationen ausführen in HealthOmics

Sie können einen Lauf starten, erneut ausführen, klonen, abbrechen oder löschen:

- Start— HealthOmics erstellt einen neuen Lauf mit den von Ihnen angegebenen Konfigurationseinstellungen und startet dann den Lauf.
- Rerun— HealthOmics erstellt einen neuen Lauf, der ein Duplikat des von Ihnen angegebenen Laufs ist. Sie können einen gelöschten Lauf mit dem HealthOmics rerun Tool erneut ausführen.
- Clone— Sie können einen vorhandenen Lauf mithilfe der Konsole klonen. Die Konsole öffnet die Seite „Clone-Run“ und füllt die Konfigurationsfelder anhand der Werte aus dem vorhandenen Lauf vorab aus. Sie können die Werte nach Bedarf ändern und den geklonten Lauf starten.
- Cancel— Sie können einen Lauf abbrechen, der noch nicht abgeschlossen ist. Wenn Sie einen Lauf abbrechen, HealthOmics werden keine der Laufausgaben gespeichert.
- Delete— Sie können abgeschlossene Läufe manuell löschen oder den Aufbewahrungsmodus so einstellen, HealthOmics dass die ältesten Läufe automatisch gelöscht werden. Weitere Informationen zum Aufbewahrungsmodus finden Sie unterthe section called “Führen Sie den Aufbewahrungsmodus aus”.

Themen

Starte einen Lauf in HealthOmics

Wenn Sie einen Lauf starten, geben Sie die Ressourcen an, die für HealthOmics die Verwendung während des Laufs zugewiesen werden.

Geben Sie den Speichertyp und die Speichermenge für den Lauf an (für statischen Speicher). Um Datenisolierung und Sicherheit zu gewährleisten HealthOmics , wird der Speicher zu Beginn jedes Laufs bereitgestellt und am Ende des Laufs deprovisioniert. Weitere Informationen finden Sie unter Speichertypen in HealthOmics Workflows ausführen.

Geben Sie einen Amazon S3 S3-Speicherort für die Ausgabedateien an. Wenn Sie eine große Anzahl von Workflows gleichzeitig ausführen, verwenden Sie URIs für jeden Workflow eine separate Amazon S3 S3-Ausgabe, um Bucket-Throttling zu vermeiden. Weitere Informationen finden Sie unter Objekte mithilfe von Präfixen organisieren im Amazon S3-Benutzerhandbuch und Speicherverbindungen horizontal skalieren im Whitepaper Optimizing Amazon S3 Performance.

Sie können auch die Ausführungspriorität angeben. Wie sich die Priorität auf den Lauf auswirkt, hängt davon ab, ob der Lauf einer Ausführungsgruppe zugeordnet ist. Weitere Informationen finden Sie unter Priorität ausführen.

Wenn ein Workflow über eine oder mehrere Versionen verfügt, können Sie beim Starten des Laufs eine Version angeben. Wenn Sie keine Version angeben, HealthOmics wird die Standard-Workflow-Version gestartet.

Wenn Sie die HealthOmics API verwenden, können Sie für jeden Lauf eine eindeutige Anforderungs-ID angeben. Die Anforderungs-ID ist ein Idempotenz-Token, das zur Identifizierung doppelter Anfragen HealthOmics verwendet wird und die Ausführung nur einmal startet.

> **Note**
>
> Sie geben eine IAM-Dienstrolle an, wenn Sie einen Lauf starten. Optional kann die Konsole die Servicerolle für Sie erstellen. Weitere Informationen finden Sie unter Servicerollen für AWS HealthOmics.

Themen

HealthOmics Parameter ausführen

Wenn Sie einen Lauf starten, geben Sie die Eingaben für die Ausführung in der JSON-Datei mit den Ausführungsparametern an, oder Sie können die Parameterwerte inline eingeben. Hinweise zur Verwaltung der Größe der JSON-Datei mit den Ausführungsparametern finden Sie unterGröße der Ausführungsparameter verwalten.

HealthOmics unterstützt die folgenden JSON-Typen für Parameterwerte.

JSON-Typ	Beispiel für Schlüssel und Wert	Hinweise
boolesch	„b“: wahr	Der Wert steht nicht in Anführungszeichen und ist ausschließlich in Kleinbuch staben geschrieben.
Ganzzahl	„i“ :7	Der Wert steht nicht in Anführungszeichen.
Zahl	„f“: 42,3	Der Wert steht nicht in Anführungszeichen.
Zeichenfolge	„s“: „Zeichen“	Der Wert steht in Anführung szeichen. Verwenden Sie den Zeichenfolgentyp für Textwerte und URIs. Das URI-Ziel muss dem erwarteten Eingabetyp entsprechen.
Array	„a“: [1,2,3]	Der Wert steht nicht in Anführungszeichen. Array-Mit glieder müssen jeweils den durch den Eingabeparameter definierten Typ haben.
object	„o“: {"links“ :"a“, „rechts“ :1}	In WDL wird das Objekt WDL Pair, Map oder Struct zugeordnet

Einen Lauf über die Konsole starten

Um einen Lauf zu starten

1. Öffnen Sie die HealthOmics -Konsole.

2. Öffnen Sie bei Bedarf den linken Navigationsbereich (±1). Wählen Sie Läufe.
3. Wählen Sie auf der Seite „Läufe“ die Option Ausführung starten aus.
4. Geben Sie im Bereich mit den Ausführungsdetails die folgenden Informationen ein

   - Workflow-Quelle — Wählen Sie Eigener Workflow oder Geteilter Workflow aus.
   - Workflow-ID — Die Workflow-ID, die diesem Lauf zugeordnet ist.
   - Workflow-Version (optional) — Wählen Sie eine Workflow-Version aus, die für diesen Lauf verwendet werden soll. Wenn Sie keine Version auswählen, verwendet der Lauf die Workflow-Standardversion.
   - Laufname — Ein eindeutiger Name für diesen Lauf.
   - Ausführungspriorität (optional) — Die Priorität dieses Laufs. Höhere Zahlen geben eine höhere Priorität an, und die Aufgaben mit der höchsten Priorität werden zuerst ausgeführt.
   - Speichertyp ausführen — Geben Sie hier den Speichertyp an, um den für den Workflow angegebenen Standardspeichertyp für die Ausführung zu überschreiben. Statischer Speicher weist dem Lauf eine feste Speichermenge zu. Dynamischer Speicher wird je nach Bedarf für jede Aufgabe in der Ausführung nach oben oder unten skaliert.
   - Laufspeicherkapazität — Geben Sie für statischen Laufspeicher die Speichermenge an, die für die Ausführung benötigt wird. Dieser Eintrag überschreibt die für den Workflow angegebene Standardspeichermenge für die Ausführung.
   - Wählen Sie das S3-Ausgabeziel aus — Der S3-Speicherort, an dem die Run-Ausgaben gespeichert werden.
   - Konto-ID des Output-Bucket-Besitzers (optional) — Wenn Ihr Konto nicht Eigentümer des Output-Buckets ist, geben Sie die AWS-Konto ID des Bucket-Besitzers ein. Diese Informationen sind erforderlich, damit der Besitzer des Buckets verifiziert werden HealthOmics kann.
   - Aufbewahrungsmodus für Metadaten ausführen — Wählen Sie aus, ob die Metadaten für alle Läufe beibehalten werden sollen oder ob das System die Metadaten der ältesten Ausführung entfernen soll, wenn Ihr Konto die maximale Anzahl von Durchläufen erreicht. Weitere Informationen finden Sie unter Retentionsmodus für HealthOmics Läufe ausführen.
5. Unter Servicerolle können Sie eine bestehende Servicerolle verwenden oder eine neue erstellen.
6. (Optional) Für Tags können Sie dem Lauf bis zu 50 Tags zuweisen.
7. Wählen Sie Weiter aus.

8. Geben Sie auf der Seite Parameterwerte hinzufügen die Ausführungsparameter an. Sie können entweder eine JSON-Datei hochladen, die die Parameter spezifiziert, oder die Werte manuell eingeben.
9. Wählen Sie Weiter aus.
10. Im Bereich „Gruppe ausführen“ können Sie optional eine Ausführungsgruppe für diesen Lauf angeben. Weitere Informationen finden Sie unter HealthOmics Run-Gruppen verwenden.
11. Im Bereich „Cache ausführen“ können Sie optional einen Run-Cache für diesen Lauf angeben. Weitere Informationen finden Sie unter Konfiguration eines Laufs mit Run-Cache mithilfe der Konsole.
12. Wählen Sie Review and start run (Überprüfen und Testlauf starten).
13. Nachdem Sie die Ausführungskonfiguration überprüft haben, wählen Sie Start run aus.

Einen Lauf mithilfe der API starten

Verwenden Sie den API-Vorgang start-run, um einen Lauf zu erstellen und zu starten.

Im folgenden Beispiel werden die Workflow-ID und die Servicerolle angegeben. In diesem Beispiel wird der Aufbewahrungsmodus auf festgelegt`REMOVE`. Weitere Hinweise zum Aufbewahrungsmodus finden Sie unterRetentionsmodus für HealthOmics Läufe ausführen.

```
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
```

Als Antwort erhalten Sie die folgende Ausgabe. Das `uuid` ist einzigartig für den Lauf und `outputUri` kann zusammen mit verwendet werden, um zu verfolgen, wohin die Ausgabedaten geschrieben wurden.

```
{
    "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"
}
```

Fügen Sie eine Parameterdatei hinzu

Wenn die Parametervorlage für einen Workflow erforderliche Parameter deklariert, können Sie eine lokale JSON-Datei mit den Eingaben bereitstellen, wenn Sie eine Workflow-Ausführung starten. Die JSON-Datei enthält den genauen Namen der einzelnen Eingabeparameter und einen Wert für den Parameter.

Verweisen Sie auf die JSON-Eingabedatei in`--parameters file://<input_file.json>`, AWS CLI indem Sie sie zu Ihrer `start-run` Anfrage hinzufügen. Weitere Hinweise zu Ausführungsparametern finden Sie unterHealthOmics Eingaben ausführen.

Geben Sie eine Anforderungs-ID an

Sie können `requestId` für jeden Lauf eine eindeutige Nummer angeben. Die Anforderungs-ID ist ein Idempotenz-Token, das HealthOmics verwendet wird, um doppelte Anfragen abzufangen. Es wird kein Lauf gestartet, wenn die Anforderungs-ID ein Duplikat eines vorherigen Laufs ist.

Wenn Sie Infrastruktur (wie Lambda-Funktionen oder Step-Funktionen) für die Orchestrierung von Runstarts verwenden, empfiehlt es sich, für jede StartRun Anfrage eine eindeutige Anforderungs-ID anzugeben. Dadurch wird sichergestellt, dass, wenn Ihre Infrastruktur versehentlich einen Lauf startet, den sie bereits gestartet hat, nicht der doppelte Lauf gestartet HealthOmics wird. Wenn die Infrastruktur beispielsweise versucht, sich nach einem Upstream-Fehler zu erholen, wird möglicherweise ein Skript erneut ausgeführt, das versucht, Läufe zu starten, bei denen es sich um doppelte Anfragen handelt.

Wählen Sie eine Workflow-Version

Sie können eine Workflow-Version für den Lauf angeben. Wenn Sie keine Version angeben, HealthOmics startet der Lauf mit der Standard-Workflow-Version.

```
aws omics start-run
     --workflow-id workflow id \
      ...
     --workflow-version-name '1.2.1'
```

Überschreiben Sie den Speichertyp für die Ausführung

Sie können den Standard-Laufspeichertyp überschreiben, der im Workflow festgelegt wurde.

```
aws omics start-run
       --workflow-id workflow id \
```

```
        ...
       --storage-type STATIC
       --storage-capacity 2400
```

Führen Sie einen GPU-Workflow aus

Sie können auch eine GPU-Workflow-ID angeben, wie im folgenden Beispiel gezeigt:

```
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
```

Informationen über einen Lauf abrufen

Sie können die ID in der Antwort mit der Get-Run-API verwenden, um den Status eines Laufs zu überprüfen, wie in der Abbildung gezeigt.

```
aws omics get-run --id run id
```

Die Antwort auf diesen API-Vorgang gibt Ihnen Auskunft über den Status der Workflow-Ausführung. Mögliche Status sind`PENDING`, `STARTING``RUNNING`, und`COMPLETED`. Wenn ein Lauf ausgeführt wird`COMPLETED`, finden Sie eine Ausgabedatei, die `outfile.txt` in Ihrem Amazon S3 S3-Ausgabe-Bucket aufgerufen wird, in einem Ordner, der nach der Lauf-ID benannt ist.

Der API-Vorgang get-run gibt auch andere Details zurück, z. B. ob es sich bei dem Workflow um `Ready2Run` oder handelt`PRIVATE`, die Workflow-Engine und Accelerator-Details. Das folgende Beispiel zeigt die Antwort von get-run für die Ausführung eines privaten Workflows, beschrieben in WDL mit einem GPU-Beschleuniger und ohne der Ausführung zugewiesene Tags.

```
{
    "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": {}
}
```

Sie können den Status aller Ausführungen mit dem API-Vorgang „List-Runs“ anzeigen, wie in der Abbildung gezeigt.

```
aws omics list-runs
```

Verwenden Sie die API, um alle Aufgaben zu sehen, die für einen bestimmten Lauf abgeschlossen wurden. list-run-tasks

```
aws omics list-run-tasks --id task ID
```

Verwenden Sie die get-run-task API, um die Details einer bestimmten Aufgabe abzurufen.

```
aws omics get-run-task --id <run_id> --task-id task ID
```

Nach Abschluss der Ausführung werden die Metadaten CloudWatch unter dem Stream an gesendet**`manifest/run/<run ID>/<run UUID>`**.

Das Folgende ist ein Beispiel für das Manifest.

```
{
    "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"
  },
  ...
```

Run-Metadaten werden nicht gelöscht, wenn sie nicht in den CloudWatch Protokollen vorhanden sind.

Führen Sie einen Run in erneut aus HealthOmics

Verwenden Sie für Läufe, die Sie noch nicht gelöscht haben, die Konsole oder API, um den Lauf erneut auszuführen. Verwenden Sie für Läufe, die Sie gelöscht haben, das Tool. HealthOmics rerun

Themen

- Führen Sie einen Lauf mit der Konsole erneut aus
- Führen Sie einen Lauf mithilfe der API erneut aus
- Verwenden Sie das Rerun-Tool

Führen Sie einen Lauf mit der Konsole erneut aus

Gehen Sie von der Konsole aus wie folgt vor, um einen Lauf erneut auszuführen:

1. Öffnen Sie die HealthOmics -Konsole.
2. Öffnen Sie bei Bedarf den linken Navigationsbereich (!). Wählen Sie Läufe aus.
3. Wählen Sie auf der Seite „Läufe“ den Lauf aus, der erneut ausgeführt werden soll.
4. Wählen Sie im Aktionsmenü über der Tabelle die Option Erneut ausführen aus.

Führen Sie einen Lauf mithilfe der API erneut aus

Verwenden Sie den StartRun API-Vorgang, um einen vorhandenen Lauf erneut auszuführen. Geben Sie die folgenden erforderlichen Eingaben ein:

- Eine Dienstrolle ARN (`roleArn`).
- Die ID des zu duplizierenden Laufs (`runId`).
- Ein Amazon S3 S3-Speicherort, an dem die Ausführung die Ausführungsausgaben (`outputUri`) speichert.

```
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
```

Verwenden Sie das Rerun-Tool

Für einen gelöschten Lauf können Sie das HealthOmics rerun Tool herunterladen und verwenden, um den Lauf erneut auszuführen. Das Tool ruft Ausführungsinformationen aus dem CloudWatch Logs-Manifest ab. Laden Sie das rerun Tool aus dem HealthOmics GitHub Tool-Repository herunter.

Das folgende Beispiel zeigt, wie das rerun Tool verwendet wird.

```
aws-healthomics-rerun 9876543
```

Wenn der Lauf in existiert CloudWatch, erhalten Sie eine Antwort, die der folgenden Beispielausgabe ähnelt. Wenn der Workflow nicht mehr existiert, erhalten Sie eine Fehlermeldung.

```
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": {}
}
```

Einen Run in klonen HealthOmics

Sie können einen vorhandenen Lauf mithilfe der HealthOmics Konsole klonen. Beim Klonen wird ein neuer Lauf erstellt, der die Konfigurationswerte des geklonten Laufs verwendet. Sie können diese Standardwerte ändern und weitere optionale Eingaben hinzufügen.

1. Öffnen Sie die HealthOmics -Konsole.
2. Öffnen Sie bei Bedarf den linken Navigationsbereich (•). Wählen Sie Läufe.
3. Wählen Sie auf der Seite „Runs“ den Lauf aus, den Sie klonen möchten.
4. Wählen Sie im Aktionsmenü über der Tabelle die Option Ausführung klonen aus. Die Konsole öffnet das Formular zum Ausführen von Klonen. Das Formular ist identisch mit Start run, außer dass die Konsole das Formular mit allen relevanten Werten aus dem geklonten Lauf füllt.

   Die Konsole erstellt eine neue Run-ID für den Run-Clone und fügt diese Run-ID als Suffix zum Runnamen hinzu.

   Beim Durchblättern der Formularseiten können Sie die Konfigurationswerte nach Bedarf anpassen.
5. Nachdem Sie die Ausführungskonfiguration überprüft haben, wählen Sie Start run aus.

Einen Run in stornieren HealthOmics

Sie können einen Lauf stornieren, wenn er den Status`PENDING`, `STARTINGRUNNING`, oder hat`STOPPING`.

> **Note**
>
> Wenn Sie einen Lauf abbrechen, HealthOmics werden keine der Laufausgaben gespeichert.

Gehen Sie von der Konsole aus wie folgt vor, um einen Lauf abzubrechen:

1. Öffnen Sie die HealthOmics -Konsole.
2. Öffnen Sie bei Bedarf den linken Navigationsbereich (•). Wählen Sie Läufe aus.
3. Wählen Sie auf der Seite „Läufe“ den Lauf aus, den Sie abbrechen möchten.
4. Die Konsole öffnet die Seite mit den Ausführungsdetails. Wählen Sie im Statusbanner oben auf der Seite die Option Ausführung beenden aus.
5. Geben Sie Bestätigen ein, um den Lauf zu beenden.

Verwenden Sie den API-Vorgang, um einen Lauf mithilfe der CancelRun API abzubrechen.

Das folgende Beispiel zeigt, wie Sie einen Lauf mit dem abbrechen AWS CLI . Um das Beispiel auszuführen, ersetzen Sie das *run id* durch die ID des Laufs, den Sie abbrechen möchten. Bei Erfolg erfolgt keine Antwort.

```
aws omics cancel-run --id run id
```

Lösche einen Run in HealthOmics

Wenn Sie einen Lauf nicht mehr benötigen, können Sie ihn mithilfe der AWS CLI API oder der Konsole löschen. Sie können einen Lauf löschen, wenn er den Status `COMPLETED` oder hat`CANCELED`.

Gehen Sie von der Konsole aus wie folgt vor, um einen Lauf zu löschen:

1. Öffnen Sie die HealthOmics -Konsole.
2. Öffnen Sie bei Bedarf den linken Navigationsbereich (•). Wählen Sie Läufe.
3. Wählen Sie auf der Seite „Läufe“ einen oder mehrere Läufe aus, die Sie löschen möchten.
4. Wählen Sie im Aktionsmenü über der Tabelle die Option Löschen aus.
5. Geben Sie in der modalen Form confirm ein, um das Löschen zu bestätigen.

Der folgende AWS CLI Befehl löscht einen Lauf. Um das Beispiel auszuführen, ersetzen Sie das *run id* durch die ID des Laufs, den Sie löschen möchten. Es erfolgt keine Antwort, wenn der Lauf erfolgreich gelöscht wurde.

```
aws omics delete-run --id run id
```

HealthOmics Run-Gruppen verwenden

Sie können optional eine Ausführungsgruppe erstellen, um die Rechenressourcen für die Läufe, die Sie der Gruppe hinzufügen, zu begrenzen. Run-Groups können Ihnen dabei helfen:

- Stellen Sie Ihre Läufe in eine Warteschlange, damit Sie die Dienstlimits nicht überschreiten.
- Halten Sie ungenutzte Aufgaben fest, indem Sie eine maximale Ausführungsdauer festlegen.
- Verwalte die Priorität jedes Laufs so, dass die wichtigsten Läufe zuerst abgeschlossen werden.

Wenn Sie die maximale Anzahl gleichzeitiger vCPU, GPU oder Läufe festlegen, werden ausgeführte Aufgaben in eine Warteschlange gestellt, wenn die maximale Anzahl erreicht ist. Wenn Sie eine maximale Ausführungsdauer festlegen, schlägt die Ausführung fehl, wenn sie die maximale Dauer überschreitet.

Verwenden Sie die Einstellung für die Ausführungspriorität, um die Priorität innerhalb einer Ausführungsgruppe festzulegen.

Dienstlimits haben Vorrang vor Grenzwerten für Ausführungsgruppen. Wenn Sie beispielsweise ein Maximum für Ausführungsgruppen auf einen höheren Wert als Ihr Dienstmaximum in einer Region festlegen, wird das Dienstmaximum HealthOmics angewendet.

Themen

Priorität ausführen

Sie können die Ausführungspriorität verwenden, um die Priorität von Läufen in einer Ausführungsgruppe festzulegen.

Wenn mehrere Läufe dieselbe Priorität haben, hat der Lauf, der zuerst gestartet wurde, die höhere Priorität.

Sie können auch eine Priorität für einen Lauf festlegen, der sich nicht in einer Ausführungsgruppe befindet. Die Priorität wird mit den Prioritäten aller anderen Läufe verglichen, die sich nicht in einer Ausführungsgruppe befinden

Sie legen die Ausführungspriorität fest, wenn Sie den Lauf starten. Weitere Informationen finden Sie unter Starte einen Lauf in HealthOmics.

Erstellen Sie mit der Konsole eine Ausführungsgruppe

Um eine Ausführungsgruppe zu erstellen

1. Öffnen Sie die HealthOmics -Konsole.
2. Öffnen Sie bei Bedarf den linken Navigationsbereich (±1). Wählen Sie Gruppen ausführen.
3. Wählen Sie auf der Seite „Run-Gruppen“ die Option Run-Gruppe erstellen aus.
4. Geben Sie auf der Detailseite zur Ausführungsgruppe erstellen die folgenden Informationen ein

   - Name der Ausführungsgruppe — Ein eindeutiger Name für diese Ausführungsgruppe.
   - Max. vCPU für gleichzeitige Läufe — Die maximale Anzahl von vCPUs , die gleichzeitig über alle aktiven Läufe in der Ausführungsgruppe ausgeführt werden können.
   - Max GPUs — Die maximale Anzahl davon GPUs , die gleichzeitig auf allen aktiven Läufen in der Ausführungsgruppe ausgeführt werden können.
   - Max. Laufzeit (Minuten) pro Lauf — Die maximale Laufzeit für jeden Lauf (in Minuten). Wenn ein Lauf die maximale Laufzeit überschreitet, schlägt der Lauf automatisch fehl.
   - Max. Anzahl gleichzeitiger Läufe — Die maximale Anzahl von Läufen, die gleichzeitig ausgeführt werden können.
5. (optional) Sie können der Ausführungsgruppe bis zu 50 Tags hinzufügen.
6. Wählen Sie „Ausführungsgruppe erstellen“.

Erstellen Sie eine Ausführungsgruppe mit der CLI

Um eine Ausführungsgruppe zu erstellen, verwenden Sie den create-run-groupAPI-Vorgang, um eine Ausführungsgruppe mit dem Namen zu erstellen`TestRunGroup`. Im folgenden Beispiel werden maximal 20 CPUs, 10 GPUs, 5 Läufe und eine maximale Ausführungsdauer von 600 Minuten festgelegt.

```
aws omics create-run-group --name TestRunGroup \
--max-cpus 20 \
--max-gpus 10 \
--max-duration 600 \
--max-runs 5
```

Die Antwort aus diesem API-Vorgang enthält die ID der neu erstelltenRunGroup.

```
{
    "arn": "arn:aws:omics:us-west-2:12345678901:runGroup/2839621",
    "id": "2839621",
    "tags": {}
}
```

Um zusätzliche Informationen über die Run-Gruppe zu erhalten, verwenden Sie diese ID zusammen mit dem get-run-groupAPI-Vorgang, wie im folgenden Beispiel gezeigt.

```
aws omics get-run-group --id run group id
```

Die Antwort umfasst die Grenzwerteinstellungen für die Ausführungsgruppe und die zugewiesenen Tags.

```
{
    "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
}
```

Sie können den list-run-groupAPI-Vorgang auch verwenden, um alle erstellten Ausführungsgruppen anzuzeigen.

```
aws omics list-run-groups
```

Löschen Sie eine Ausführungsgruppe mithilfe der Konsole

Sie können eine Ausführungsgruppe löschen, wenn dieser Ausführungsgruppe keine Läufe mit dem StatusPENDING, STARTINGRUNNING, oder zugeordnet sindSTOPPING.

Gehen Sie wie folgt vor, um eine Ausführungsgruppe zu löschen.

1. Öffnen Sie die HealthOmics -Konsole.
2. Öffnen Sie bei Bedarf den linken Navigationsbereich (±1). Wählen Sie Gruppen ausführen.
3. Wählen Sie auf der Seite „Gruppen ausführen“ die Ausführungsgruppe aus, die Sie löschen möchten, und klicken Sie im Feld xx auf Löschen.

Löschen Sie eine Ausführungsgruppe mit der CLI

Sie können eine Ausführungsgruppe löschen, wenn dieser Ausführungsgruppe mit dem StatusPENDING, STARTINGRUNNING, oder keine Läufe zugeordnet sindSTOPPING.

Das folgende Beispiel zeigt, wie Sie die verwenden können AWS CLI , um eine Ausführungsgruppe zu löschen. Sie werden keine Antwort erhalten. Um das Beispiel auszuführen, ersetzen Sie das *run group id* durch die ID der Ausführungsgruppe, die Sie löschen möchten.

```
aws omics delete-run-group --id run group id
```

Aufruf-Caching für Läufe HealthOmics

AWS HealthOmics unterstützt das Zwischenspeichern von Anrufen, auch bekannt als Resume, für private Workflows. Beim Aufruf-Caching werden die Ausgaben abgeschlossener Workflow-Aufgaben nach Abschluss eines Laufs gespeichert. Nachfolgende Ausführungen können die Aufgabenausgaben aus dem Cache verwenden, anstatt die Aufgabenausgaben erneut zu berechnen. Das Aufruf-Caching reduziert die Nutzung von Rechenressourcen, was zu kürzeren Ausführungsdauern und zu Einsparungen bei den Rechenkosten führt.

Nach Abschluss der Ausführung können Sie auf die zwischengespeicherten Aufgabenausgabedateien zugreifen. Für das erweiterte Debugging und die Problembehandlung von Aufgaben können Sie zwischengespeicherte Aufgabendateien zwischenspeichern, indem Sie diese Dateien in der Workflow-Definition als Aufgabenausgaben angeben.

Sie können die Zwischenspeicherung von Aufrufen verwenden, um die abgeschlossenen Aufgabenergebnisse fehlgeschlagener Ausführungen zu speichern. Die nächste Ausführung beginnt mit der letzten erfolgreich abgeschlossenen Aufgabe, anstatt die abgeschlossenen Aufgaben erneut zu berechnen.

Wenn HealthOmics kein passender Cache-Eintrag für eine Aufgabe gefunden wird, schlägt die Ausführung nicht fehl. HealthOmics berechnet die Aufgabe und die von ihr abhängigen Aufgaben neu.

Informationen zur Behebung von Problemen mit der Zwischenspeicherung von Anrufen finden Sie unter. Behebung von Problemen beim Zwischenspeichern von Anrufen

Themen

So funktioniert das Caching von Anrufen

Um Call Caching zu verwenden, erstellen Sie einen Run-Cache und konfigurieren ihn so, dass er über einen zugehörigen Amazon S3 S3-Speicherort für die zwischengespeicherten Daten verfügt. Wenn Sie einen Lauf starten, geben Sie den Run-Cache an. Ein Run-Cache ist nicht für einen einzigen Workflow vorgesehen. Läufe aus mehreren Workflows können denselben Cache verwenden.

Während der Exportphase eines Laufs exportiert das System die abgeschlossenen Aufgabenausgaben an den Amazon S3 S3-Speicherort. Um zwischengeschaltete Aufgabendateien zu exportieren, deklarieren Sie diese Dateien in der Workflow-Definition als Aufgabenausgaben. Das Aufruf-Caching speichert auch intern Metadaten und erstellt eindeutige Hashes für jeden Cache-Eintrag.

Für jede Aufgabe in einem Lauf erkennt die Workflow-Engine, ob es für diese Aufgabe einen passenden Cache-Eintrag gibt. Wenn es keinen passenden Cache-Eintrag gibt, HealthOmics

berechnet die Aufgabe. Wenn es einen passenden Cache-Eintrag gibt, ruft die Engine die zwischengespeicherten Ergebnisse ab.

HealthOmics Verwendet zum Abgleichen von Cache-Einträgen den Hashing-Mechanismus, der in den systemeigenen Workflow-Engines enthalten ist. HealthOmicserweitert diese vorhandenen Hash-Implementierungen um HealthOmics Variablen wie S3-ETags und ECR-Container-Digests.

HealthOmics unterstützt das Caching von Aufrufen für diese Workflow-Sprachversionen:

- Die WDL-Versionen 1.0, 1.1 und die Entwicklungsversion
- Nextflow-Versionen 23.10 und 24.10
- Alle CWL-Versionen

> **Note**
>
> HealthOmics unterstützt kein Aufruf-Caching für Ready2Run-Workflows.

Themen

Modell der geteilten Verantwortung

Die Benutzer sind gemeinsam dafür verantwortlich, AWS zu bestimmen, ob Aufgaben und Läufe sich für das Call-Caching eignen. Das Aufruf-Caching erzielt die besten Ergebnisse, wenn alle Aufgaben idempotent sind (wiederholte Ausführungen einer Aufgabe mit denselben Eingaben führen zu denselben Ergebnissen).

Wenn eine Aufgabe jedoch nicht deterministische Elemente enthält (wie Zufallszahlengenerationen oder Systemzeit), können wiederholte Ausführungen der Aufgabe mit denselben Eingaben zu unterschiedlichen Ausgaben führen. Dies kann sich auf folgende Weise auf die Effektivität des Caching von Anrufen auswirken:

- Wenn ein Cache-Eintrag HealthOmics verwendet wird (der durch einen vorherigen Lauf erstellt wurde), der nicht mit der Ausgabe identisch ist, die die Ausführung der Aufgabe für den aktuellen Lauf erzeugen würde, kann der Lauf zu anderen Ergebnissen führen als derselbe Lauf ohne Zwischenspeicherung.
- HealthOmics findet möglicherweise keinen passenden Cache-Eintrag für eine Aufgabe, der entsprechen sollte, weil die Aufgabenausgabe nicht deterministisch ist. Wenn der gültige Cache-Eintrag nicht gefunden wird, wird die Aufgabe bei der Ausführung unnötig neu berechnet, wodurch die Kosteneinsparung durch die Verwendung von Aufruf-Caching beeinträchtigt wird.

Im Folgenden sind bekannte Verhaltensweisen von Aufgaben aufgeführt, die zu nicht deterministischen Ergebnissen führen können, die sich auf die Ergebnisse der Zwischenspeicherung von Aufrufen auswirken:

- Verwendung von Zufallszahlengeneratoren.
- Abhängigkeit von der Systemzeit.
- Verwendung von Parallelität (Rennbedingungen können zu Leistungsabweichungen führen).
- Abrufen von lokalen oder entfernten Dateien, die über die in den Eingabeparametern der Aufgabe angegebenen hinausgehen.

Weitere Szenarien, die zu nicht deterministischem Verhalten führen können, finden Sie unter Nicht deterministische Prozesseingaben auf der Nextflow-Dokumentationsseite.

Wenn Sie vermuten, dass eine Aufgabe nicht deterministische Ergebnisse erzeugt, sollten Sie die Funktionen der Workflow-Engine verwenden, um zu vermeiden, dass bestimmte Aufgaben zwischengespeichert werden, die nicht deterministisch sind. Anweisungen, wie Sie das Caching für einzelne Aufgaben in jeder unterstützten Workflow-Sprache deaktivieren können, finden Sie unter. Engine-spezifische Caching-Funktionen

Wir empfehlen Ihnen, Ihre spezifischen Anforderungen an den Arbeitsablauf und die Aufgaben gründlich zu überprüfen, bevor Sie das Caching von Anrufen in Umgebungen aktivieren, in denen ineffektives Caching von Anrufen oder andere als erwartete Ergebnisse ein Risiko darstellen können. Beispielsweise sollten die potenziellen Einschränkungen von Call Caching bei der Entscheidung, ob Call Caching für klinische Anwendungsfälle geeignet ist, sorgfältig abgewogen werden.

Anforderungen an das Zwischenspeichern von Aufgaben

HealthOmics speichert Aufgabenausgaben für Aufgaben, die die folgenden Anforderungen erfüllen:

- Die Aufgabe muss einen Container definieren. HealthOmics speichert keine Ausgaben für eine Aufgabe ohne Container im Cache.
- Die Aufgabe muss eine oder mehrere Ausgaben erzeugen. Sie geben die Ausgaben der Aufgaben in der Workflow-Definition an.
- Die Workflow-Definition darf keine dynamischen Werte verwenden. Wenn Sie beispielsweise einen Parameter mit einem Wert an eine Aufgabe übergeben, der bei jedem Lauf erhöht wird, werden die Ausgaben der Aufgabe HealthOmics nicht zwischengespeichert.

> **Note**
>
> Wenn mehrere Aufgaben in einer Ausführung dasselbe Container-Image verwenden, wird HealthOmics für alle diese Aufgaben dieselbe Image-Version bereitgestellt. Nach HealthOmics dem Abrufen des Images werden alle Aktualisierungen des Container-Images für die Dauer der Ausführung ignoriert. Dieser Ansatz bietet eine vorhersehbare und konsistente Benutzererfahrung und verhindert potenzielle Probleme, die sich aus Aktualisierungen des Container-Images ergeben könnten, die während der Ausführung bereitgestellt werden.

Führen Sie die Cache-Leistung aus

Wenn Sie das Zwischenspeichern von Aufrufen für einen Lauf aktivieren, stellen Sie möglicherweise die folgenden Auswirkungen auf die Ausführungsleistung fest:

- HealthOmics Speichert während der ersten Ausführung die Cache-Daten für Aufgaben in der Ausführung. Bei diesem Lauf kann es zu längeren Exportzeiten kommen, da das Aufruf-Caching die Menge der Exportdaten erhöht.
- Wenn Sie in nachfolgenden Ausführungen einen Lauf aus dem Cache wieder aufnehmen, kann dies die Anzahl der Verarbeitungsschritte und die Laufzeit verringern.
- Wenn Sie sich auch dafür entscheiden, Zwischendateien als Ausgaben zu deklarieren, können Ihre Exportzeiten sogar noch länger sein, da diese Daten ausführlicher sein können.

Ereignisse zur Aufbewahrung und Invalidierung von Daten zwischenspeichern

Der Hauptzweck eines Run-Caches besteht darin, die Berechnung der Aufgaben während der Ausführung zu optimieren. Wenn es einen gültigen passenden Cache-Eintrag für eine Aufgabe gibt,

wird der Cache-Eintrag HealthOmics verwendet, anstatt die Aufgabe neu zu berechnen. Andernfalls wird das Standardverhalten des Dienstes HealthOmics wiederhergestellt, bei dem die Aufgabe und die von ihr abhängigen Aufgaben neu berechnet werden. Bei diesem Ansatz führen Cache-Fehler nicht dazu, dass die Ausführung fehlschlägt.

Es wird empfohlen, die Größe des Run-Caches zu verwalten. Im Laufe der Zeit sind Cacheeinträge aufgrund von Workflow-Engine- oder HealthOmics Service-Updates oder aufgrund von Änderungen, die Sie an der Ausführung oder den Ausführungsaufgaben vorgenommen haben, möglicherweise nicht mehr gültig. Die folgenden Abschnitte enthalten zusätzliche Informationen.

Themen

Manifeste Versionsupdates und Datenaktualität

In regelmäßigen Abständen führt der HealthOmics Dienst möglicherweise neue Funktionen oder Workflow-Engine-Updates ein, die einige oder alle Run-Cache-Einträge ungültig machen. In diesem Fall kann es bei Ihren Läufen zu einem einmaligen Cachefehler kommen.

HealthOmics erstellt für jeden Cache-Eintrag eine JSON-Manifestdatei. Für Läufe, die nach dem 12. Februar 2025 gestartet wurden, enthält die Manifestdatei einen Versionsparameter. Wenn ein Service-Update Cache-Einträge ungültig macht, wird die Versionsnummer HealthOmics erhöht, sodass Sie die Legacy-Cache-Einträge identifizieren können, die entfernt werden müssen.

Das folgende Beispiel zeigt eine Manifestdatei, deren Version auf 2 gesetzt ist:

```
{
     "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,
}
```

Bei Läufen mit Cacheeinträgen, die nicht mehr gültig sind, erstellen Sie den Cache neu, um neue gültige Einträge zu erstellen. Führen Sie für jeden Lauf die folgenden Schritte aus:

1. Starten Sie den Lauf einmal, wobei die Cache-Aufbewahrung auf CACHE ALWAYS eingestellt ist. Bei diesem Lauf werden die neuen Cache-Einträge erstellt.
2. Setzen Sie für nachfolgende Läufe die Cache-Aufbewahrung auf die vorherige Einstellung (CACHE ALWAYS oder CACHE ON FAILURE).

Um Cache-Einträge zu bereinigen, die nicht mehr gültig sind, können Sie diese Cache-Einträge aus dem Amazon S3 S3-Cache-Bucket löschen. HealthOmics verwendet diese Cache-Einträge niemals wieder. Wenn Sie sich dafür entscheiden, ungültige Einträge beizubehalten, hat dies keine Auswirkungen auf Ihre Läufe.

> **Note**
>
> Beim Aufruf-Caching werden die Ausgabedaten der Aufgaben an dem für den Cache angegebenen Amazon S3 S3-Speicherort gespeichert, wodurch Ihnen Gebühren entstehen. AWS-Konto

Verhalten beim Ausführen des Caches

Sie können das Verhalten beim Ausführen des Caches festlegen, um die Taskausgaben für fehlgeschlagene Läufe (Cache on Failure) oder für alle Läufe (Cache immer) zu speichern. Wenn Sie einen Run-Cache erstellen, legen Sie das Standard-Cache-Verhalten für alle Läufe fest, die diesen Cache verwenden. Sie können das Standardverhalten überschreiben, wenn Sie einen Lauf starten.

Cache on failureist nützlich, wenn Sie einen Workflow debuggen, der fehlschlägt, nachdem mehrere Aufgaben erfolgreich abgeschlossen wurden. Die nachfolgende Ausführung wird mit der letzten erfolgreich abgeschlossenen Aufgabe fortgesetzt, wenn alle vom Hash berücksichtigten eindeutigen Variablen mit der vorherigen Ausführung identisch sind.

Cache alwaysist nützlich, wenn Sie eine Aufgabe in einem Workflow aktualisieren, der erfolgreich abgeschlossen wurde. Wir empfehlen, dass Sie die folgenden Schritte ausführen:

1. Erstellen Sie einen neuen Lauf. Stellen Sie das Cache-Verhalten auf Immer zwischenspeichern ein und starten Sie den Lauf.
2. Nachdem die Ausführung abgeschlossen ist, aktualisieren Sie die Aufgabe im Workflow und starten Sie eine neue Ausführung mit dem Verhalten „Immer zwischenspeichern“. Dieser Lauf verarbeitet die aktualisierte Aufgabe und alle nachfolgenden Aufgaben, die von der aktualisierten Aufgabe abhängig sind. Alle anderen Aufgaben verwenden die zwischengespeicherten Ergebnisse.
3. Wiederholen Sie Schritt 2 nach Bedarf, bis die Entwicklung für die aktualisierte Aufgabe abgeschlossen ist.
4. Verwenden Sie die aktualisierte Aufgabe bei future Läufen nach Bedarf. Denken Sie daran, nachfolgende Läufe im Fehlerfall auf Cache umzustellen, wenn Sie beabsichtigen, für diese Läufe neue oder andere Eingaben zu verwenden.

> **Note**
>
> Wir empfehlen den Modus Immer zwischenspeichern, wenn Sie denselben Testdatensatz verwenden, jedoch nicht für mehrere Läufe. Wenn Sie diesen Modus für eine große Anzahl von Durchläufen festlegen, kann das System große Datenmengen nach Amazon S3 exportieren, was zu erhöhten Exportzeiten und Speicherkosten führt.

Steuern Sie die Größe des Run-Caches

HealthOmics löscht oder archiviert keine Run-Cache-Daten und wendet auch keine Amazon S3 S3-Bereinigungsregeln für die Verwaltung der Cache-Daten an. Wir empfehlen Ihnen, regelmäßige Cache-Bereinigungen durchzuführen, um Amazon S3 S3-Speicherkosten zu sparen und die Größe Ihres Run-Caches überschaubar zu halten. Sie können Dateien direkt löschen oder retention/ replication Datenrichtlinien für den Run-Cache-Bucket festlegen.

Sie können beispielsweise eine Amazon S3 S3-Lebenszyklusrichtlinie so konfigurieren, dass Objekte nach 90 Tagen ablaufen, oder Sie können die Cache-Daten am Ende jedes Entwicklungsprojekts manuell bereinigen.

Die folgenden Informationen können Ihnen bei der Verwaltung der Cache-Datengröße helfen:

- Sie können sehen, wie viele Daten sich im Cache befinden, indem Sie Amazon S3 überprüfen. HealthOmics überwacht oder berichtet nicht über die Cachegröße.
- Wenn Sie einen gültigen Cache-Eintrag löschen, schlägt die nachfolgende Ausführung nicht fehl. HealthOmics berechnet die Aufgabe und ihre abhängigen Aufgaben neu.
- Wenn Sie Cache-Namen oder Verzeichnisstrukturen so ändern, HealthOmics dass kein passender Eintrag für eine Aufgabe gefunden wird, HealthOmics berechnet die Aufgabe neu.

Wenn Sie überprüfen müssen, ob ein Cache-Eintrag noch gültig ist, überprüfen Sie die Versionsnummer des Cache-Manifests. Weitere Informationen finden Sie unter Manifeste Versionsupdates und Datenaktualität.

Einen Run-Cache erstellen

Wenn Sie einen Run-Cache erstellen, geben Sie einen Amazon S3 S3-Speicherort für die Cache-Daten an. Auf diese Daten muss sofort zugegriffen werden können. Beim Aufruf-Caching werden keine in Glacier archivierten Objekte abgerufen (z. B. GFR- und GDA-Speicherklassen).

Wenn der Amazon S3 S3-Bucket für die Cache-Daten einem anderen gehört AWS-Konto, geben Sie diese Konto-ID an, wenn Sie den Run-Cache erstellen.

Einen Run-Cache mithilfe der Konsole erstellen

Gehen Sie von der Konsole aus wie folgt vor, um einen Run-Cache zu erstellen.

1. Öffnen Sie die HealthOmics -Konsole.
2. Öffnen Sie bei Bedarf den linken Navigationsbereich (±1). Wählen Sie „Caches ausführen“.
3. Wählen Sie auf der Seite „Caches ausführen“ die Option Run-Cache erstellen aus.
4. Konfigurieren Sie auf der Seite „Run-Cache erstellen“ im Bereich „Details zum Ausführungs-Cache“ die folgenden Felder:

    a. Geben Sie einen Namen für den Run-Cache ein.

    b. (Optional) Geben Sie eine Beschreibung ein.

    c. Geben Sie einen S3-Speicherort für die zwischengespeicherte Ausgabe ein. Wählen Sie einen Bucket in derselben Region wie Ihr Workflow aus.

    d. (Optional) Geben Sie den Namen AWS-Konto des Bucket-Besitzers ein, um die Inhaberschaft des Buckets zu überprüfen. Wenn Sie keinen Wert eingeben, ist der Standardwert Ihre Konto-ID.

e. Konfigurieren Sie unter Cache-Verhalten das Standardverhalten (ob Ausgaben für fehlgeschlagene Läufe oder für alle Läufe zwischengespeichert werden sollen). Wenn Sie einen Lauf starten, können Sie optional das Standardverhalten überschreiben.

5. (Optional) Ordnen Sie dem Run-Cache ein oder mehrere Tags zu.
6. Wählen Sie „Run-Cache erstellen“. Die Konsole zeigt den neuen Run-Cache in der Tabelle Run-Caches an.

Einen Run-Cache mit der CLI erstellen

Verwenden Sie den create-run-cacheCLI-Befehl, um einen Run-Cache zu erstellen. Das Standard-Cache-Verhalten ist`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"
```

Wenn die Erstellung erfolgreich ist, erhalten Sie eine Antwort mit den folgenden Feldern.

```
{
  "arn": "string",
  "id": "string",
  "status": "ACTIVE"
  "tags": {}
  }
```

Einen Run-Cache aktualisieren

Sie können den Cache-Namen, die Beschreibung, die Tags oder das Cache-Verhalten ändern, aber nicht den S3-Speicherort für den Cache.

Einen Run-Cache mithilfe der Konsole aktualisieren

Gehen Sie von der Konsole aus wie folgt vor, um einen Run-Cache zu aktualisieren.

1. Öffnen Sie die HealthOmics -Konsole.

2. Öffnen Sie bei Bedarf den linken Navigationsbereich (±1). Wählen Sie „Caches ausführen“.
3. Wählen Sie in der Tabelle Run Caches den Run-Cache aus, der aktualisiert werden soll, und klicken Sie dann auf Bearbeiten.
4. Im Bereich „Details zum Ausführungs-Cache“ können Sie die Felder „Name“, „Beschreibung“ und „Cache-Verhalten“ aktualisieren.
5. (Optional) Ordnen Sie dem Run-Cache ein oder mehrere neue Tags zu oder entfernen Sie vorhandene Tags.
6. Wählen Sie „Run-Cache speichern“.

Aktualisierung eines Run-Caches mit der CLI

Verwenden Sie den update-run-cacheCLI-Befehl, um einen Run-Cache zu aktualisieren.

```
aws omics update-run-cache \
      --name "workflow 123 run cache" \
      --id "workflow id" \
      --description "my run cache" \
      --cache-behavior "CACHE_ALWAYS"
```

Wenn das Update erfolgreich ist, erhalten Sie eine Antwort ohne Datenfelder.

Löschen eines Run-Caches

Sie können einen Run-Cache löschen, wenn ihn keine aktiven Läufe verwenden. Wenn Läufe den Run-Cache verwenden, warten Sie, bis die Läufe abgeschlossen sind, oder Sie können die Läufe abbrechen.

Durch das Löschen eines Run-Caches werden die Ressource und ihre Metadaten entfernt, die Daten in Amazon S3 werden jedoch nicht gelöscht. Nachdem Sie den Cache gelöscht haben, können Sie ihn nicht erneut anhängen oder ihn für nachfolgende Läufe verwenden.

Die zwischengespeicherten Daten verbleiben zu Ihrer Prüfung in Amazon S3. Sie können alte Cache-Daten mithilfe von Delete Standard-S3-Vorgängen entfernen. Alternativ können Sie eine Amazon S3 S3-Lebenszyklusrichtlinie erstellen, um zwischengespeicherte Daten, die Sie nicht mehr verwenden, ablaufen zu lassen.

Löschen eines Run-Caches mithilfe der Konsole

Gehen Sie von der Konsole aus wie folgt vor, um einen Run-Cache zu löschen.

1. Öffnen Sie die HealthOmics -Konsole.
2. Öffnen Sie bei Bedarf den linken Navigationsbereich (±1). Wählen Sie „Caches ausführen“.
3. Wählen Sie in der Tabelle Run Caches den Run-Cache aus, den Sie löschen möchten.
4. Wählen Sie im Tabellenmenü „Caches ausführen“ die Option Löschen aus.
5. Speichern Sie im modalen Dialog den Amazon S3 S3-Cache-Datenlink zum future Nachschlagen und bestätigen Sie dann, dass Sie den Run-Cache löschen möchten.

   Sie können den Amazon S3 S3-Link verwenden, um die zwischengespeicherten Daten zu überprüfen, aber Sie können die Daten nicht erneut mit einem anderen Run-Cache verknüpfen. Löschen Sie die Cache-Daten, wenn Sie die Inspektion abgeschlossen haben.

Löschen eines Run-Caches mit der CLI

Verwenden Sie den delete-run-cacheCLI-Befehl, um einen Run-Cache zu löschen.

```
aws omics delete-run-cache \
      --id "my cache id"
```

Wenn das Löschen erfolgreich ist, erhalten Sie eine Antwort ohne Datenfelder.

Inhalt eines Run-Caches

HealthOmics organisiert Ihren Run-Cache mit der folgenden Struktur in Ihrem S3-Bucket:

```
s3://{cache.S3location}/{cache.uuid}/runID/taskID/{cacheentry.uuid}/
```

Die cache.uuid ist die weltweit eindeutige ID für den Cache. Die cacheentry.uuid ist die weltweit eindeutige UUID für eine zwischengespeicherte Aufgabe. HealthOmics weist die UUIDs Caches und Aufgaben zu.

Für alle Workflow-Engines enthält der Cache die folgenden Dateien:

- Die {cacheentryuuid}.json Datei — HealthOmics erstellt diese Manifestdatei, die Informationen über den Cache enthält, einschließlich einer Liste aller Elemente im Cache und der Cache-Version.
- Task-Ausgabedateien — Jede Task-Ausgabe besteht aus einer oder mehreren Dateien, wie von der Aufgabe definiert.

Für einen Workflow, der Nextflow verwendet, erstellt die Nextflow-Engine diese zusätzlichen Dateien im Cache:

- Die command.out Datei — Diese Datei enthält den Standardinhalt zur Aufgabenausführung.
- Die .exitcode Datei — Diese Datei enthält den Exit-Code der Aufgabe (eine Ganzzahl).

> **Note**
>
> Wenn Sie zur erweiterten Problembehandlung auf zwischengeschaltete Aufgabendateien in Ihrem Run-Cache zugreifen möchten, deklarieren Sie diese Dateien in der Workflow-Definition als Aufgabenausgaben.

Engine-spezifische Caching-Funktionen

HealthOmics versucht, eine konsistente Implementierung von Call Caching in allen Workflow-Engines bereitzustellen. Je nachdem, wie jede Workflow-Engine mit bestimmten Fällen umgeht, gibt es einige Unterschiede:

- Nextflow
  - Das Zwischenspeichern zwischen verschiedenen Nextflow-Versionen ist nicht garantiert. Wenn Sie beispielsweise eine Aufgabe in Version 23.10.0 ausführen und anschließend dieselbe Aufgabe in Version 24.10.8 ausführen, HealthOmics könnten Sie den zweiten Lauf als Cache-Fehler betrachten.
  - Sie können das Caching für einzelne Aufgaben deaktivieren, indem Sie die Cache-Direktive verwenden. false Informationen zu dieser Direktive finden Sie in den Prozessen in der Nextflow-Spezifikation.
  - HealthOmics verwendet den Nextflow-Lenient-Modus, unterstützt aber keinen Deep-Caching-Modus.
  - Beim Caching wird jedes einzelne S3-Objekt ausgewertet, wenn Sie ein Glob-Muster im S3-Pfad zu den Eingaben für eine Aufgabe verwenden. Wenn Sie ein neues Objekt hinzufügen, werden nur die Aufgaben HealthOmics neu berechnet, die das neue Objekt verwenden.
  - HealthOmics speichert keine Wiederholungsversuche von Aufgaben im Cache. Dieses Verhalten entspricht dem Standardverhalten von Nextflow.
- WDL

  - HealthOmics unterstützt den neuen Verzeichnistyp für Eingaben, wenn Sie die Entwicklungsversion des WDL-Workflows verwenden. Bei der Zwischenspeicherung von Aufrufen werden alle Aufgaben, die das Verzeichnis eingeben, HealthOmics neu berechnet, wenn sich ein Objekt im Verzeichnis ändert.
  - HealthOmics unterstützt Caching auf Aufgabenebene, aber kein Caching auf Workflow-Ebene.
  - Sie können das Caching für einzelne Aufgaben deaktivieren, indem Sie das Volatile-Attribut verwenden. Weitere Informationen finden Sie unter Deaktivieren Sie das Caching auf Aufgabenebene mit dem Attribut volatile.
- CWL
  - Konstante Ausgaben von Aufgaben sind in den Manifesten nicht explizit sichtbar. HealthOmics speichert konstante Ausgaben als Zwischendateien im Cache.
  - Sie können das Caching für einzelne Aufgaben mithilfe der WorkReuseFunktion steuern.

Verwenden Sie den Run-Cache

Standardmäßig verwenden Läufe keinen Run-Cache. Um einen Cache für die Ausführung zu verwenden, geben Sie den Run-Cache und das Verhalten des Run-Caches an, wenn Sie die Ausführung starten.

Nach Abschluss eines Laufs können Sie die Konsole, CloudWatch Protokolle oder API-Operationen verwenden, um Cache-Treffer zu verfolgen oder Cache-Probleme zu beheben. Details dazu finden Sie unter Informationen zum Anruf-Caching werden nachverfolgt und Behebung von Problemen beim Zwischenspeichern von Anrufen.

Wenn eine oder mehrere Aufgaben in einem Lauf nicht deterministische Ausgaben generieren, empfehlen wir dringend, für die Ausführung kein Aufruf-Caching zu verwenden oder diese speziellen Aufgaben vom Caching auszuschließen. Weitere Informationen finden Sie unter Modell der geteilten Verantwortung.

> **Note**
>
> Sie geben eine IAM-Dienstrolle an, wenn Sie einen Lauf starten. Um Call Caching verwenden zu können, benötigt die Service-Rolle die Erlaubnis, auf den Amazon S3 S3-Standort Run Cache zuzugreifen. Weitere Informationen finden Sie unter Servicerollen für AWS HealthOmics.

Sie können Amazon Q CLI verwenden, um Ihre Run-Cache-Daten zu analysieren und zu verwalten. Weitere Informationen finden Sie unter Beispielaufforderungen für Amazon Q CLI und im HealthOmics Agentic Generative AI-Tutorial unter. GitHub

Themen

- Konfiguration eines Laufs mit Run-Cache mithilfe der Konsole
- Konfiguration eines Laufs mit Run-Cache über die CLI
- Fehlerfälle bei Run-Caches
- Informationen zum Anruf-Caching werden nachverfolgt

Konfiguration eines Laufs mit Run-Cache mithilfe der Konsole

Von der Konsole aus konfigurieren Sie den Run-Cache für einen Lauf, wenn Sie den Lauf starten.

1. Öffnen Sie die HealthOmics -Konsole.
2. Öffnen Sie bei Bedarf den linken Navigationsbereich (±1). Wählen Sie Läufe aus.
3. Wählen Sie auf der Seite „Läufe“ den Lauf aus, den Sie starten möchten.
4. Wählen Sie Start run und führen Sie die Schritte 1 und 2 von Run starten aus, wie unter beschriebenEinen Lauf über die Konsole starten.
5. Wählen Sie in Schritt 3 von Start Run die Option Existierenden Run-Cache auswählen aus.
6. Wählen Sie den Cache aus der Dropdownliste „Cache-ID ausführen“ aus.
7. Um das Standardverhalten beim Ausführen des Cache zu überschreiben, wählen Sie das Cache-Verhalten für den Lauf aus. Weitere Informationen finden Sie unter Verhalten beim Ausführen des Caches.
8. Fahren Sie mit Schritt 4 von Start Run fort.

Konfiguration eines Laufs mit Run-Cache über die CLI

Um einen Lauf zu starten, der einen Run-Cache verwendet, fügen Sie dem CLI-Befehl start-run den Parameter cache-id hinzu. Verwenden Sie optional den `cache-behavior` Parameter, um das Standardverhalten zu überschreiben, das Sie für den Run-Cache konfiguriert haben. Das folgende Beispiel zeigt nur die Cache-Felder für den Befehl:

```
aws omics start-run \
        ...
```

```
    --cache-id "xxxxxx"    \
    --cache-behavior  CACHE_ALWAYS
```

Wenn der Vorgang erfolgreich ist, erhalten Sie eine Antwort ohne Datenfelder.

Fehlerfälle bei Run-Caches

In den folgenden Szenarien werden Taskausgaben HealthOmics möglicherweise nicht zwischengespeichert, auch nicht bei einer Ausführung, bei der das Cacheverhalten auf Immer zwischenspeichern eingestellt ist.

- Wenn bei der Ausführung ein Fehler auftritt, bevor die erste Aufgabe erfolgreich abgeschlossen wurde, können keine Cache-Ausgaben exportiert werden.
- Wenn der Exportvorgang fehlschlägt, werden die Aufgabenausgaben HealthOmics nicht im Amazon S3 S3-Cache gespeichert.
- Wenn der Lauf aufgrund eines filesystem out of space Fehlers fehlschlägt, speichert das Aufruf-Caching keine Aufgabenausgaben.
- Wenn Sie einen Lauf abbrechen, speichert das Aufruf-Caching keine Aufgabenausgaben.
- Wenn bei der Ausführung ein Timeout auftritt, speichert das Aufruf-Caching keine Aufgabenausgaben, auch wenn Sie den Lauf so konfiguriert haben, dass bei einem Fehler der Cache verwendet wird.

Informationen zum Anruf-Caching werden nachverfolgt

Sie können Call-Caching-Ereignisse (z. B. Run-Cache-Treffer) mithilfe der Konsole, der CLI oder CloudWatch Logs verfolgen.

Themen

- Cache-Treffer mithilfe der Konsole verfolgen
- Verfolgen Sie das Anruf-Caching mit der CLI
- Verfolgen Sie das Caching von Anrufen mithilfe von Protokollen CloudWatch

Cache-Treffer mithilfe der Konsole verfolgen

Auf der Seite mit den Ausführungsdetails für einen Lauf werden in der Tabelle Aufgaben ausführen Informationen zu Cache-Treffern für jede Aufgabe angezeigt. Die Tabelle enthält auch einen Link zum

zugehörigen Cache-Eintrag. Gehen Sie wie folgt vor, um die Cache-Trefferinformationen für einen Lauf anzuzeigen.

1. Öffnen Sie die HealthOmics -Konsole.
2. Öffnen Sie bei Bedarf den linken Navigationsbereich (±1). Wählen Sie Läufe aus.
3. Wählen Sie auf der Seite „Läufe“ den Lauf aus, den Sie überprüfen möchten.
4. Wählen Sie auf der Seite mit den Ausführungsdetails die Registerkarte Aufgaben ausführen, um die Tabelle mit den Aufgaben anzuzeigen.
5. Wenn eine Aufgabe einen Cache-Treffer hat, enthält die Spalte Cache-Treffer einen Link zum Speicherort des Run-Cache-Eintrags in Amazon S3.
6. Wählen Sie den Link, um den Run-Cache-Eintrag zu überprüfen.

Verfolgen Sie das Anruf-Caching mit der CLI

Überprüfen Sie mit dem CLI-Befehl get-run, ob der Lauf einen Aufrufcache verwendet hat.

```
aws omics get-run --id 1234567
```

Wenn das `cacheId` Feld in der Antwort gesetzt ist, verwendet die Ausführung diesen Cache.

Verwenden Sie den list-run-tasksCLI-Befehl, um den Speicherort der Cache-Daten für jede zwischengespeicherte Aufgabe in der Ausführung abzurufen.

```
aws omics list-run-tasks --id 1234567
```

Wenn das Feld CacheHit für eine Aufgabe den Wert true hat, gibt das Feld caches3URI in der Antwort den Speicherort der Cache-Daten für diese Aufgabe an.

Sie können auch den get-run-taskCLI-Befehl verwenden, um den Cache-Datenspeicherort für eine bestimmte Aufgabe abzurufen:

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

Verfolgen Sie das Caching von Anrufen mithilfe von Protokollen CloudWatch

HealthOmics erstellt Cache-Aktivitätsprotokolle in der `/aws/omics/WorkflowLog` CloudWatch Protokollgruppe. <cache_id><cache_uuid>Für jeden Run-Cache gibt es einen Log-Stream: RunCache//.

HealthOmics Generiert für Läufe, die Aufruf-Caching verwenden, CloudWatch Log-Einträge für die folgenden Ereignisse:

- einen Cache-Eintrag erstellen (CACHE_ENTRY_CREATED)
- Abgleichen eines Cache-Eintrags (CACHE_HIT)
- konnte mit einem Cache-Eintrag nicht übereinstimmen (CACHE_MISS)

Weitere Hinweise zu diesen Protokollen finden Sie unter. Meldet sich an CloudWatch

Verwenden Sie die folgende CloudWatch Insights-Abfrage für die `/aws/omics/WorkflowLog` Protokollgruppe, um die Anzahl der Cache-Treffer pro Lauf für diesen Cache zurückzugeben:

```
filter @logStream like 'runCache/<CACHE_ID>/'
 fields @timestamp, @message
 filter logMessage like 'CACHE_HIT'
 parse "run: *," as run
 stats count(*) as cacheHits by run
```

Verwenden Sie die folgende Abfrage, um die Anzahl der Cache-Einträge zurückzugeben, die bei jedem Lauf erstellt wurden:

```
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
```

HealthOmics Workflows teilen

Als Besitzer eines privaten Workflows können Sie den Workflow mit einem AWS-Konto in derselben Region teilen. Um einen Workflow mit mehr als einem zu teilen AWS-Konto, erstellen Sie mehrere Shares desselben Workflows.

Als Besitzer können Sie den Zugriff auf einen geteilten Workflow widerrufen, indem Sie den Share löschen.

> **Note**
>
> HealthOmics ermöglicht einem gemeinsam genutzten Workflow automatisch den Zugriff auf das Amazon ECR-Repository, während der Workflow im Konto des Abonnenten ausgeführt wird. Sie müssen keinen zusätzlichen Repository-Zugriff für gemeinsam genutzte Workflows gewähren.

Wenn Sie einen Workflow teilen, kann der Abonnent jede der Workflow-Versionen verwenden. Wenn Sie für einen gemeinsamen Workflow eine Zugriffskontrolle auf Versionsebene benötigen, empfehlen wir, separate Workflows zu erstellen, anstatt Workflow-Versionen zu verwenden.

Themen

Einen gemeinsamen Workflow abonnieren

Um einen gemeinsamen Workflow zu abonnieren, folgen Sie diesen allgemeinen Schritten, um den Workflow zu akzeptieren und zu verwenden:

1. Verwenden Sie die Konsole oder API, um das Teilen zu akzeptieren. Lege für deine aktuelle Region dieselbe Region fest wie die Teilungsanfrage.
   - Um die Freigabeanfrage in der Konsole zu finden, navigieren Sie zur Seite Alle gemeinsam genutzten Ressourcen und wählen Sie dann den Tab Für mich freigegeben.
2. Verwenden Sie die Konsole oder API, um eine Ausführung für den gemeinsam genutzten Workflow zu erstellen.
   - Um die Seite mit den Workflow-Details in der Konsole zu finden, navigieren Sie zu Für mich freigegeben (siehe Schritt 1) und wählen Sie dann den Link Ressource für den gemeinsam genutzten Workflow aus.

3. Sie geben Ihre eigenen Eingabedaten für den Workflow an.
4. Der gemeinsam genutzte Workflow läuft in Ihrem AWS-Konto.

Als Abonnent eines gemeinsamen Workflows blockiert das System Sie daran, die folgenden Workflow-Aktionen auszuführen:

- Exportieren eines gemeinsamen Workflows
- Den gemeinsamen Workflow erneut ausführen
  - Sie erstellen eine neue Ausführung für den gemeinsamen Workflow.
- Den Workflow erneut teilen.
- Dem Workflow ein Tag zuweisen.
- Den Workflow löschen.
  - Wenn Sie den Workflow nicht mehr benötigen, löschen Sie den Workflow-Share.

Weitere Informationen Kontoübergreifende gemeinsame Nutzung von Ressourcen in AWS HealthOmics zur gemeinsamen Nutzung von Ressourcen finden Sie unter.

Status einer Workflow-Freigabe überwachen

HealthOmics sendet bei jeder Statusänderung einer Workflow-Freigabe ein Ereignis an. EventBridge Wenn Sie Benachrichtigungen über bestimmte Statusänderungen erhalten möchten, richten Sie eine EventBridge Regel zur Überwachung von Statusänderungsereignissen für Workflow-Freigaben ein. Zum Beispiel:

- Sie möchten jedes Mal, wenn Sie eine Anfrage zur Workflow-Freigabe erhalten, und jedes Mal, wenn ein Benutzer eine Workflow-Freigabe widerruft, eine Benachrichtigung erhalten.
- Nachdem Sie eine Anfrage zur Workflow-Freigabe initiiert haben, möchten Sie eine Benachrichtigung erhalten, wenn der Benutzer die Anfrage akzeptiert oder ablehnt.

Einzelheiten zur Verwendung von Ereignissen finden Sie unterVerwenden EventBridge mit AWS HealthOmics.

Einen privaten Workflow über die Konsole teilen

Von der Konsole aus können Sie einen privaten Workflow mit einem teilen, der sich AWS-Konto in derselben Region wie der Workflow befindet.

Um einen privaten Workflow zu teilen

1. Öffnen Sie die HealthOmics -Konsole.
2. Öffnen Sie bei Bedarf den linken Navigationsbereich (←). Wählen Sie Private Workflows.
3. Wählen Sie in der Tabelle Workflows auf der Seite Private Workflows den Workflow aus, den Sie teilen möchten, und wählen Sie Teilen aus.
4. Geben Sie auf der Workflow-Seite „Teilen“ im Bereich „Freigabedetails“ einen aussagekräftigen Namen für die gemeinsame Nutzung und den Namen AWS-Konto des Abonnenten ein.
5. Wählen Sie Ressource teilen aus. Die Konsole zeigt Ressourcenfreigaben auf der Seite Alle Ressourcenfreigaben an.

Der ursprüngliche Status der Freigabe ist ausstehend. Nachdem der Abonnent die Aktie akzeptiert hat, wechselt der Status in „Aktiv“.

Einen privaten Workflow mit der CLI teilen

Verwenden Sie den API-Vorgang create-share, um eine Workflow-Freigabe zu erstellen. Der Hauptabonnent ist AWS-Konto der Benutzer, der Zugriff auf den Workflow erhält.

```
aws omics create-share \
  --resource-arn "arn:aws:omics:us-west-2:555555555555:workflow/123456" \
  --principal-subscriber "123456789012" \
  --name "my_Share-123"
```

Wenn die Erstellung erfolgreich ist, erhalten Sie eine Antwort mit der Share-ID und dem Status.

```
{
"shareId": "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a",
"name": "my_Share-123",
"status": "PENDING"
}
```

Die Freigabe verbleibt im Status „Ausstehend“, bis der Abonnent sie mithilfe des `accept-share` API-Vorgangs akzeptiert.

Weitere Beispiele Kontoübergreifende gemeinsame Nutzung von Ressourcen in AWS HealthOmics zur API-Nutzung finden Sie unter.

Akzeptieren eines gemeinsamen Workflows mithilfe der Konsole

Sie können die Konsole verwenden, um eine angebotene Workflow-Freigabe anzunehmen. Stellen Sie sicher, dass für die Konsole dieselbe Region wie für den Workflow eingestellt ist.

1. Öffnen Sie die HealthOmics -Konsole.
2. Öffnen Sie bei Bedarf den linken Navigationsbereich (•). Wählen Sie „Alle Ressourcenfreigaben“ und anschließend die Registerkarte „Für mich freigegeben“.
3. Wählen Sie in der Tabelle Mit mir gemeinsam genutzte Ressourcen die Workflow-Freigabe aus und klicken Sie dann auf Akzeptieren.

Nachdem Sie den Workflow akzeptiert haben, wählen Sie den Link Ressource für den gemeinsam genutzten Workflow aus, um dessen Details anzuzeigen.

Einen gemeinsamen Workflow mithilfe der Konsole ausführen

Nachdem Sie eine Workflow-Freigabe akzeptiert haben, können Sie eine Ausführung des Workflows starten.

1. Öffnen Sie die HealthOmics -Konsole.
2. Öffnen Sie bei Bedarf den linken Navigationsbereich (←). Wählen Sie „Alle Ressourcenfreigaben“ und anschließend die Registerkarte „Für mich freigegeben“.
3. Wählen Sie in der Tabelle Mit mir gemeinsam genutzte Ressourcen den Link Ressource für den gemeinsam genutzten Workflow aus.
4. Wählen Sie auf der Seite mit den Workflow-Details die Option Ausführung erstellen aus.

   Die Konsole öffnet die Seite „Ausführung erstellen“, auf der der Workflowtyp (gemeinsam genutzt) und die Workflow-ID vorausgefüllt sind.
5. Konfigurieren Sie die verbleibenden Felder im Formular „Ausführung erstellen“. Weitere Informationen finden Sie unter Einen Lauf über die Konsole starten.

Einen gemeinsamen Workflow mithilfe der API ausführen

Verwenden Sie get-workflow, um den ARN des gemeinsam genutzten Workflows abzurufen.

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

```
--workflow-owner-id 55555555555
```

Wenn Sie den Workflow ausführen, geben Sie die AWS-Konto ID des Workflow-Besitzers und den ARN des gemeinsam genutzten Workflows an.

```
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
```

Ready2Run-Workflows in HealthOmics

Ready2Run-Workflows sind vorkonfigurierte Workflows, die von Drittanbietern veröffentlicht wurden. Einige Herausgeber, wie Sentieon Inc., bieten Workflows auf Abonnementbasis an. Für andere Ready2Run-Workflows ist kein Abonnement erforderlich, und einige Workflows sind Open Source, wie z. B. die NF-Core-Workflows.

Ready2Run-Workflows eignen sich gut für die folgenden Szenarien:

- Sie möchten sich auf die Analyse der Pipeline-Ergebnisse und die Generierung von Ergebnissen konzentrieren, ohne die zugrunde liegende Infrastruktur einrichten zu müssen.
- Sie möchten Ihre Ergebnisse mithilfe etablierter Workflows replizieren.
- Als Softwareentwickler möchten Sie Ihre Anwendung direkt in das HealthOmics SDK integrieren.

HealthOmics unterstützt die Versionierung für Ready2Run-Workflows. Für einen Ready2Run-Workflow, der Versionen anbietet, können Sie den Versionsnamen angeben, wenn Sie einen Lauf starten.

Alle Ready2Run-Workflows bieten Protokolle, einschließlich CloudWatch Protokolle, die Sie zur Fehlerbehebung verwenden können.

> **Note**
>
> Sentieon Ready2Run-Workflows basieren auf Abonnements. Wenn Sie einen Sentieon Ready2Run-Workflow zum ersten Mal in einem Konto ausführen, erstellt Sentieon automatisch eine zweiwöchige Testlizenz für Ihren. AWS-Konto Die Lizenz ist für alle Sentieon Ready2Run-Workflows gültig. Nach Ablauf des Testzeitraums können Sie eine permanente Lizenz oder eine Verlängerung der Evaluierungslizenz beantragen. Details dazu finden Sie unter Subscribing to Sentieon Ready2Run workflows.

Themen

Verfügbare Ready2Run-Workflows in HealthOmics

In der folgenden Tabelle sind die Ready2Run-Workflows aufgeführt, die in verfügbar sind. HealthOmics

Sie können sich bei der HealthOmicsKonsole anmelden, um detaillierte Informationen zu diesen Workflows, einschließlich Eingabeparametern und Workflow-Diagrammen, einzusehen. Preisinformationen zu Ready2Run-Workflows finden Sie unter HealthOmics Preise.

> **Note**
>
> Jeder Ready2Run-Workflow hat eine maximale Eingabedateigröße. Diese maximalen Dateigrößen sind nicht einstellbar.

Workflow-Name	Publisher	Abonnement erforderlich?	Maximale Größe der Eingabedatei (GiB)	Geschätzte Laufzeit (HH:MM)
AlphaFold für 601-1200 Reste	Google DeepMind	Nein	1	11:15
AlphaFold für bis zu 600 Reste	Google DeepMind	Nein	1	7:30
Bases2Fastq für 2x150	Element Biowissenschaften	Nein	1000	1:45
Bases2Fastq für 2x300	Element Biowissenschaften	Nein	1000	1:30
Bases2Fastq für 2x75	Element Biowissenschaften	Nein	500	0:45

Workflow-Name	Publisher	Abonnement erforderlich?	Maximale Größe der Eingabedatei (GiB)	Geschätzte Laufzeit (HH:MM)
ESMFold für bis zu 800 Rückstände	Meta-Forschung	Nein	1	0:15
GATK-BP fq2bam	Breites Institut	Nein	64	10:10
GATK-BP-Keimbahn bam2vcf für das 30-fache Genom	Breites Institut	Nein	39	2:45
GATK-BP-Keimbahn fq2vcf für das 30-fache Genom	Breites Institut	Nein	64	12:30
GATK-BP Somatisches WES bam2vcf	Breites Institut	Nein	86	1:30
NVIDIA Parabricks BAM2 FQ2 BAM WGS für bis zu 30X	NVIDIA Corporation	Nein	80	1:39
NVIDIA Parabricks BAM2 FQ2 BAM WGS für bis zu 50X	NVIDIA Corporation	Nein	120	2:45

Workflow-Name	Publisher	Abonnement erforderlich?	Maximale Größe der Eingabedatei (GiB)	Geschätzte Laufzeit (HH:MM)
NVIDIA Parabricks BAM2 FQ2 BAM WGS für bis zu 5X	NVIDIA Corporation	Nein	20	0:18
NVIDIA Parabricks FQ2 BAM WGS für bis zu 30X	NVIDIA Corporation	Nein	71	1:00
NVIDIA Parabricks FQ2 BAM WGS für bis zu 50X	NVIDIA Corporation	Nein	137	1:45
NVIDIA Parabricks FQ2 BAM WGS für bis zu 5X	NVIDIA Corporation	Nein	13	0:15
NVIDIA Parabricks Germline DeepVariant WGS für bis zu 30X	NVIDIA Corporation	Nein	71	2:00
NVIDIA Parabricks Germline DeepVariant WGS für bis zu 50X	NVIDIA Corporation	Nein	137	3:30

Workflow-Name	Publisher	Abonnement erforderlich?	Maximale Größe der Eingabedatei (GiB)	Geschätzte Laufzeit (HH:MM)
NVIDIA Parabricks Germline DeepVariant WGS für bis zu 5X	NVIDIA Corporation	Nein	12	0:30
NVIDIA Parabricks Germline HaplotypeCaller WGS für bis zu 30X	NVIDIA Corporation	Nein	71	1:15
NVIDIA Parabricks Germline HaplotypeCaller WGS für bis zu 50X	NVIDIA Corporation	Nein	137	2:00
NVIDIA Parabricks Germline HaplotypeCaller WGS für bis zu 5X	NVIDIA Corporation	Nein	13	0:15
NVIDIA Parabricks Somatic Mutect2 WGS für bis zu 50X	NVIDIA Corporation	Nein	196	0:45

Workflow-Name	Publisher	Abonnement erforderlich?	Maximale Größe der Eingabedatei (GiB)	Geschätzte Laufzeit (HH:MM)
sc RNAseq mit Kallisto BUStools	NF-Kern	Nein	119	1:30
Sc RNAseq mit Lachs in Alevin-Fry	NF-Kern	Nein	119	2:30
sc RNAseq mit STARsolo	NF-Kern	Nein	119	2:30
Sention Germline BAM WES für bis zu 300x	Sentieon, Inc.	Ja	9	1:00
Sentieon Germline BAM WGS für bis zu 32x	Sentieon, Inc.	Ja	18	1:30
Sention Germline FASTQ WES für bis zu 100x	Sentieon, Inc.	Ja	5	0:45
Sention Germline FASTQ WES für bis zu 300x	Sentieon, Inc.	Ja	26	2:00
Sentieon Germline FASTQ WGS für bis zu 32x	Sentieon, Inc.	Ja	51	3:30

Workflow-Name	Publisher	Abonnement erforderlich?	Maximale Größe der Eingabedatei (GiB)	Geschätzte Laufzeit (HH:MM)
Sention für ONT LongRead	Sentieon, Inc.	Ja	25	1:30
Sention LongRead für PacBio HiFi	Sentieon, Inc.	Ja	58	4:00
Sentieon Somatic WES	Sentieon, Inc.	Ja	50	2:30
Sention Somatic WGS	Sentieon, Inc.	Ja	113	4:30
Ultima Genomics DeepVariant für bis zu 40x	Ultima Genomics	Nein	91	1:55

Wenn Sie einen Ready2Run-Workflow verwenden, ist Ihr Workflow vorkonfiguriert und kann nicht bearbeitet werden. Im Gegensatz zu privaten Workflows unterstützen Ready2Run-Workflows Folgendes nicht:

- Erhöhung der maximalen Größe der Eingabedatei
- Änderung der Rechenressourcen oder des Laufspeichers
- Änderung der Workflow-Definition oder der Container
- Hinzufügen von Läufen zu einer Ausführungsgruppe
- Den Workflow teilen

Wenn der Herausgeber den Ready2Run-Workflow freigegeben hat GitHub, können Sie Ihren eigenen privaten Workflow auf der Grundlage des Ready2Run-Workflows erstellen. Die folgende Tabelle enthält Links zu GitHub Workflows für jeden Herausgeber.

Publisher	Workflows auf GitHub
Google DeepMind, Metarecherche	Arbeitsabläufe zur Proteinfaltung
Element-Biowissenschaften	Für Informationen wenden Sie sich bitte an Element Biosciences
Breites Institut	GATK-Workflows
NVIDIA Corporation	Arbeitsabläufe bei Parabricks
NF-Kern	NF-Core-Workflows
Empfindung	Arbeitsabläufe bei Sentieon
Ultima Genomik	Arbeitsabläufe bei Ultima Genomics

Abonnieren Sie die Sentieon Ready2Run-Workflows

Sentieon Ready2Run-Workflows basieren auf Abonnements. Wenn Sie einen Sentieon Ready2Run-Workflow zum ersten Mal in einem Konto ausführen, erstellt Sentieon automatisch eine zweiwöchige Testlizenz für Ihren. AWS-Konto Die Lizenz ist für alle Sentieon Ready2Run-Workflows gültig. Nach Ablauf des Testzeitraums können Sie eine permanente Lizenz oder eine Verlängerung der Evaluierungslizenz beantragen.

Gehen Sie wie folgt vor, um die Sentieon Ready2Run-Workflows zu abonnieren:

- Finden Sie Ihre AWS Canonical Benutzer-ID, indem Sie diesen Anweisungen folgen.
- Senden Sie eine E-Mail an die Sentieon-Supportgruppe (support@sentieon.com), um eine Softwarelizenz anzufordern. Geben Sie in der AWS E-Mail Ihre Canonical Benutzer-ID an.

HealthOmics Ready2Run-Workflows über die Konsole starten

Die Verwendung von Ready2Run-Workflows in der Konsole ähnelt der Verwendung eines privaten Workflows. Ein wesentlicher Unterschied besteht darin, dass der Workflow-Herausgeber Beispieldaten bereitstellt, sodass Sie den Workflow ausprobieren können, ohne Ihre eigenen Daten zu erstellen.

Um einen Ready2Run-Workflow in der Konsole zu verwenden

1. Öffnen Sie die HealthOmics -Konsole.
2. Öffnen Sie bei Bedarf den linken Navigationsbereich (•). Wählen Sie Ready2Run-Workflows.
3. Wählen Sie auf der Seite Ready2Run-Workflows den Workflow aus, den Sie verwenden möchten. Die Konsole öffnet die Detailseite für diesen Workflow.
4. Auf der Registerkarte „Details“ werden Informationen wie Name, Listenpreis pro Lauf, Beschreibung, Workflow-Sprachtyp, Laufspeicherkapazität, Status, Erstellungsdatum und Parameter mit Beschreibungen aufgeführt. Auf der Registerkarte „Details“ erfahren Sie auch, ob für den Workflow ein Abonnement erforderlich ist.
5. Um den Workflow zu verwenden, wählen Sie Create run
6. Geben Sie auf der Seite „Ausführungsdetails angeben“ einen Namen für den Lauf ein. Optional können Sie die Workflow-Version angeben. Sie können dem Lauf auch eine Ausführungspriorität hinzufügen.
7. Geben Sie einen Amazon S3 S3-Speicherort für die Run-Ausgabe ein oder wählen Sie ihn aus.
8. Wählen Sie für den Aufbewahrungsmodus für Metadaten ausführen aus, ob Ausführungsmetadaten beibehalten oder entfernt werden sollen.
9. Wählen Sie im Bereich „Servicerolle“ aus, ob Sie eine bestehende Servicerolle verwenden oder eine neue erstellen möchten.
10. (Optional) Fügen Sie Tags hinzu, um Ihren Lauf leichter identifizieren und verwalten zu können.
11. Wählen Sie Weiter aus.
12. Wählen Sie auf der Seite „Parameter hinzufügen“ eine der Optionen aus, um die Werte der Ausführungsparameter hinzuzufügen:

    - Wählen Sie eine Parameterdatei (im JSON-Format) von einem Amazon S3 S3-Speicherort aus.
    - Wählen Sie eine Parameterdatei (im JSON-Format) von Ihrem lokalen Laufwerk aus.
    - Geben Sie die Parameterwerte manuell ein.
    - Führen Sie den Workflow mit den vom Workflow-Herausgeber bereitgestellten Ready2Run-Beispieldaten aus.
13. Wenn Sie eine JSON-Datei hochladen, analysiert die Konsole die Datei und führt eine Inline-Validierung durch. Anschließend können Sie die Werte Ihrer Parameter nach Bedarf manuell aktualisieren.
14. Wählen Sie Weiter aus.

15. Überprüfen Sie Ihre Eingaben und wählen Sie dann Start run.

HealthOmics Ready2Run-Workflows mithilfe der API starten

Die meisten API-Operationen verhalten sich bei Ready2Run-Workflows und privaten Workflows ähnlich.

Um eine Liste der verfügbaren Ready2Run-Workflows zurückzugeben, verwenden Sie list-workflows, wobei der Parameter auf RUN gesetzt ist. `type` READY2

```
aws omics list-workflows --type READY2RUN
```

Nachdem Sie anhand der Antwort auf die Liste der Workflows den auszuführenden Workflow identifiziert haben, können Sie get-workflow mit dem Parameter verwenden, um weitere Details abzurufen. `--id`

```
aws omics get-workflow --type READY2RUN --id workflow id
```

Um einen Ready2Run-Workflow auszuführen, können Sie den API-Vorgang start-run verwenden, wobei der Parameter workflow-type auf gesetzt ist, wie im folgenden Beispiel gezeigt 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
```

Um eine Workflow-Version anzugeben, verwenden Sie den Parameter workflow-version, wie in diesem Beispiel gezeigt.

```
aws-omics start-run \
  --workflow-type READY2RUN \
  ...
  --version-name '3.0.0'
```

Um Ihren Lauf zu überwachen, können Sie den API-Vorgang get-run verwenden, wie in der Abbildung gezeigt.

```
aws-omics get-run \
  --id run id
```

HealthOmics Speicher

Verwenden Sie HealthOmics Speicher, um Genomdaten effizient und kostengünstig zu speichern, abzurufen, zu organisieren und gemeinsam zu nutzen. HealthOmics Storage versteht die Beziehungen zwischen verschiedenen Datenobjekten, sodass Sie definieren können, welche Lesesätze aus denselben Quelldaten stammen. Auf diese Weise erhalten Sie Informationen zur Herkunft der Daten.

Daten, die im `ACTIVE` Status gespeichert sind, können sofort abgerufen werden. Daten, auf die seit 30 Tagen oder länger nicht zugegriffen wurde, werden im `ARCHIVE` Status gespeichert. Um auf archivierte Daten zuzugreifen, können Sie sie über die API-Funktionen oder die Konsole reaktivieren.

HealthOmics Sequenzspeicher dienen dazu, die Inhaltsintegrität von Dateien zu wahren. Die bitweise Äquivalenz von importierten Datendateien und exportierten Dateien wird jedoch aufgrund der Komprimierung beim aktiven und beim Archiv-Tiering nicht beibehalten.

HealthOmics Generiert während der Aufnahme ein Entitäts-Tag oder HealthOmics ETag, um die Überprüfung der Inhaltsintegrität Ihrer Datendateien zu ermöglichen. Teile der Sequenzierung werden auf der Quellebene eines ETag Lesesatzes identifiziert und erfasst. Die ETag Berechnung ändert nichts an der tatsächlichen Datei oder den Genomdaten. Nachdem ein Lesesatz erstellt wurde, ETag sollte er sich während des gesamten Lebenszyklus der Lesesatzquelle nicht ändern. Das bedeutet, dass der erneute Import derselben Datei dazu führt, dass derselbe ETag Wert berechnet wird.

Themen

HealthOmics ETags und Herkunft der Daten

Ein HealthOmics ETag (Entity-Tag) ist ein Hash des aufgenommenen Inhalts in einem Sequenzspeicher. Dies vereinfacht das Abrufen und Verarbeiten von Daten und gewährleistet gleichzeitig die Inhaltsintegrität der aufgenommenen Datendateien. Dies ETag spiegelt Änderungen am semantischen Inhalt des Objekts wider, nicht an seinen Metadaten. Der angegebene Lesesatztyp und der Algorithmus bestimmen, wie der berechnet ETag wird. Die ETag Berechnung ändert nichts an der tatsächlichen Datei oder den Genomdaten. Wenn das Dateitypschema des Lesesatzes dies zulässt, aktualisiert der Sequenzspeicher Felder, die mit der Herkunft der Daten verknüpft sind.

Dateien haben eine bitweise Identität und eine semantische Identität. Die bitweise Identität bedeutet, dass die Bits einer Datei identisch sind, und eine semantische Identität bedeutet, dass der Inhalt einer Datei identisch ist. Semantische Identität ist widerstandsfähig gegenüber Änderungen an Metadaten und Komprimierung, da sie die Inhaltsintegrität der Datei erfasst.

Lesesätze in HealthOmics Sequenzspeichern durchlaufen compression/decompression Zyklen und die Herkunft der Daten wird während des gesamten Lebenszyklus eines Objekts nachverfolgt. Während dieser Verarbeitung kann sich die bitweise Identität einer aufgenommenen Datei ändern, und es wird erwartet, dass sie sich bei jeder Aktivierung einer Datei ändert. Die semantische Identität der Datei bleibt jedoch erhalten. Die semantische Identität wird als HealthOmics Entitäts-Tag erfasst, oder ETag sie wird während der Aufnahme des Sequenzspeichers berechnet und ist als Readset-Metadaten verfügbar.

Wenn das Dateitypschema des Lesesatzes dies zulässt, werden die Felder für die Aktualisierung des Sequenzspeichers mit der Herkunft der Daten verknüpft. Bei UBam-, BAM- und CRAM-Dateien wird der Kopfzeile ein neues `@CO` OR-Tag hinzugefügt. `Comment` Der Kommentar enthält die Sequenzspeicher-ID und den Zeitstempel der Aufnahme.

Amazon S3 ETags

Beim Zugriff auf eine Datei über den Amazon S3-URI können Amazon S3 S3-API-Operationen auch Amazon S3 ETag - und Prüfsummenwerte zurückgeben. Die Amazon S3 ETag - und Prüfsummenwerte unterscheiden sich von den HealthOmics ETags , weil sie die bitweise Identität der Datei darstellen. Weitere Informationen zu beschreibenden Metadaten und Objekten finden Sie in der Amazon S3 Object API-Dokumentation. Amazon S3 ETag S3-Werte können sich mit jedem Aktivierungszyklus eines Lesesets ändern, und Sie können sie verwenden, um das Lesen einer Datei zu validieren. Speichern Sie Amazon S3 ETag S3-Werte jedoch nicht im Cache, um sie während des Lebenszyklus der Datei für die Überprüfung der Dateiidentität zu verwenden, da sie nicht konsistent

bleiben. Im Gegensatz dazu HealthOmics ETag bleiben sie während des gesamten Lebenszyklus des Lesesets konsistent.

Wie HealthOmics berechnet ETags

Das ETag wird aus einem Hash des aufgenommenen Dateiinhalts generiert. Die ETag Algorithmusfamilie ist MD5up standardmäßig auf eingestellt, kann aber bei der Erstellung des Sequenzspeichers anders konfiguriert werden. Wenn der berechnet ETag wird, werden der Algorithmus und die berechneten Hashes dem Lesesatz hinzugefügt. Die unterstützten MD5 Algorithmen für Dateitypen lauten wie folgt.

- FASTQ_ MD5up — Berechnet den MD5 Hash einer unkomprimierten, vollständigen FASTQ-Leseset-Quelle.
- BAM_ MD5up — Berechnet den MD5 Hash des Alignment-Abschnitts einer unkomprimierten BAM- oder UBAM-Readeset-Quelle, wie sie im SAM dargestellt wird, auf der Grundlage der verlinkten Referenz, sofern eine verfügbar ist.
- CRAM_ MD5up — Berechnet den MD5 Hash des Alignment-Abschnitts der unkomprimierten CRAM-Lesesatz-Quelle, wie er im SAM dargestellt wird, auf der Grundlage der verknüpften Referenz.

> **Note**
>
> MD5 Hashing ist bekanntermaßen anfällig für Kollisionen. Aus diesem Grund könnten zwei verschiedene Dateien dasselbe haben, ETag wenn sie so hergestellt wurden, dass sie die bekannte Kollision ausnutzen.

Die folgenden Algorithmen werden für die SHA256 Familie unterstützt. Die Algorithmen werden wie folgt berechnet:

- FASTQ_ SHA256up — Berechnet den SHA-256-Hash einer unkomprimierten, vollständigen FASTQ-Leset-Quelle.
- BAM_ SHA256up — Berechnet den SHA-256-Hash des Alignment-Abschnitts einer unkomprimierten BAM- oder UBAM-Readeset-Quelle, wie sie im SAM dargestellt wird, auf der Grundlage der verlinkten Referenz, sofern eine verfügbar ist.

- CRAM_ SHA256up — Berechnet den SHA-256-Hash des Alignment-Abschnitts einer unkomprimierten CRAM-Lesesatz-Quelle, wie er im SAM dargestellt wird, auf der Grundlage der verknüpften Referenz.

Die folgenden Algorithmen werden für die Familie unterstützt. SHA512 Die Algorithmen werden wie folgt berechnet:

- FASTQ_ SHA512up — Berechnet den SHA-512-Hash einer unkomprimierten, vollständigen FASTQ-Leset-Quelle.
- BAM_ SHA512up — Berechnet den SHA-512-Hash des Alignment-Abschnitts einer unkomprimierten BAM- oder UBAM-Readeset-Quelle, wie sie im SAM dargestellt wird, auf der Grundlage der verlinkten Referenz, sofern eine verfügbar ist.

- CRAM_ SHA512up — Berechnet den SHA-512-Hash des Alignment-Abschnitts einer unkomprimierten CRAM-Lesesatz-Quelle, wie er im SAM dargestellt wird, auf der Grundlage der verknüpften Referenz.

Einen HealthOmics Referenzspeicher erstellen

Ein Referenzspeicher in HealthOmics ist ein Datenspeicher für die Speicherung von Referenzgenomen. Sie können in jeder AWS-Konto Region einen einzigen Referenzspeicher haben. Sie können mit der Konsole oder der CLI einen Referenzspeicher erstellen.

Themen

Einen Referenzspeicher mithilfe der Konsole erstellen

So erstellen Sie einen Referenzspeicher

1. Öffnen Sie die HealthOmics -Konsole.
2. Öffnen Sie bei Bedarf den linken Navigationsbereich (•). Wählen Sie Referenzspeicher.
3. Wählen Sie aus den Speicheroptionen für Genomics-Daten die Option Referenzgenome aus.

4. Sie können entweder ein zuvor importiertes Referenzgenom auswählen oder ein neues importieren. Wenn Sie kein Referenzgenom importiert haben, wählen Sie oben rechts Referenzgenom importieren aus.
5. Wählen Sie auf der Auftragsseite Referenzgenom-Import erstellen entweder die Option Schnellerstellung oder Manuelles Erstellen aus, um einen Referenzspeicher zu erstellen, und geben Sie dann die folgenden Informationen ein.

   - Referenzgenomname — Ein eindeutiger Name für diesen Speicher.
   - Beschreibung (optional) — Eine Beschreibung dieses Referenzspeichers.
   - IAM-Rolle — Wählen Sie eine Rolle mit Zugriff auf Ihr Referenzgenom aus.
   - Referenz von Amazon S3 — Wählen Sie Ihre Referenzsequenzdatei in einem Amazon S3 S3-Bucket aus.
   - Tags (optional) — Stellen Sie bis zu 50 Tags für diesen Referenzspeicher bereit.

Erstellen eines Referenzspeichers mit der CLI

Das folgende Beispiel zeigt Ihnen, wie Sie mit dem einen Referenzspeicher erstellen AWS CLI. Sie können ein Referenzgeschäft pro AWS Region einrichten.

Referenzspeicher unterstützen das Speichern von FASTA-Dateien mit den Erweiterungen`.fasta`,,`.fa`,`.fas`,`.fsa`,`.faa`,`.fna`,`.ffn`,,`.frn`, `.mpfa.seq`,`.txt`. Die `bgzip` Version dieser Erweiterungen wird ebenfalls unterstützt.

Ersetzen Sie es im folgenden Beispiel *`reference store name`* durch den Namen, den Sie für Ihr Referenzgeschäft gewählt haben.

```
aws omics create-reference-store --name "reference store name"
```

Sie erhalten eine JSON-Antwort mit der ID und dem Namen des Referenzspeichers, dem ARN und dem Zeitstempel, zu dem Ihr Referenzspeicher erstellt wurde.

```
{
    "id": "3242349265",
    "arn": "arn:aws:omics:us-west-2:555555555555:referenceStore/3242349265",
    "name": "MyReferenceStore",
    "creationTime": "2022-07-01T20:58:42.878Z"
}
```

Sie können die Referenzspeicher-ID in zusätzlichen AWS CLI Befehlen verwenden. Sie können die Liste der mit Ihrem Konto IDs verknüpften Referenzspeicher mithilfe des list-reference-storesBefehls abrufen, wie im folgenden Beispiel gezeigt.

```
aws omics list-reference-stores
```

Als Antwort erhalten Sie den Namen Ihres neu erstellten Referenzgeschäfts.

```
{
    "referenceStores": [
        {
             "id": "3242349265",
             "arn": "arn:aws:omics:us-west-2:555555555555:referenceStore/3242349265",
             "name": "MyReferenceStore",
            "creationTime": "2022-07-01T20:58:42.878Z"
         }
     ]
}
```

Nachdem Sie einen Referenzspeicher erstellt haben, können Sie Importaufträge erstellen, um genomische Referenzdateien in diesen Speicher zu laden. Dazu müssen Sie eine IAM-Rolle für den Zugriff auf die Daten verwenden oder eine IAM-Rolle erstellen. Es folgt eine Beispielrichtlinie .

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/*"
            ]
        }
    ]
```

```
}
```

Sie müssen außerdem über eine Vertrauensrichtlinie verfügen, die dem folgenden Beispiel ähnelt.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": [
                   "omics.amazonaws.com"
                ]
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

Sie können jetzt ein Referenzgenom importieren. In diesem Beispiel wird das Genome Reference Consortium Human Build 38 (hg38) verwendet, das frei zugänglich ist und im Registry of Open Data unter AWS verfügbar ist. Der Bucket, der diese Daten hostet, befindet sich in US East (Ohio). Um Buckets in anderen AWS Regionen zu verwenden, können Sie die Daten in einen Amazon S3 S3-Bucket kopieren, der in Ihrer Region gehostet wird. Verwenden Sie den folgenden AWS CLI Befehl, um das Genom in Ihren Amazon S3 S3-Bucket zu kopieren.

```
aws s3 cp s3://broad-references/hg38/v0/Homo_sapiens_assembly38.fasta s3://amzn-s3-
demo-bucket
```

Anschließend können Sie mit Ihrem Importjob beginnen. Ersetzen Sie *reference store IDrole ARN*, und *source file path* durch Ihre eigene Eingabe.

```
aws omics start-reference-import-job --reference-store-id reference store ID --role-
arn role ARN --sources source file path
```

Nachdem die Daten importiert wurden, erhalten Sie die folgende Antwort in JSON.

```
{
        "id": "7252016478",
        "referenceStoreId": "3242349265",
        "roleArn": "arn:aws:iam::111122223333:role/OmicsReferenceImport",
        "status": "CREATED",
        "creationTime": "2022-07-01T21:15:13.727Z"
}
```

Sie können den Status eines Jobs mit dem folgenden Befehl überwachen. Ersetzen Sie im folgenden Beispiel *reference store ID* und *job ID* durch Ihre Referenzspeicher-ID und die Job-ID, über die Sie mehr erfahren möchten.

```
aws omics get-reference-import-job --reference-store-id reference store ID --id job ID
```

Als Antwort erhalten Sie eine Antwort mit den Details zu diesem Referenzspeicher und seinem 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"
        }
    ]
}
```

Sie können die importierte Referenz auch finden, indem Sie Ihre Referenzen auflisten und sie anhand des Referenznamens filtern. *reference store ID*Ersetzen Sie es durch Ihre Referenzshop-ID und fügen Sie einen optionalen Filter hinzu, um die Liste einzugrenzen.

```
aws omics list-references --reference-store-id reference store ID --filter
 name=MyReference
```

Als Antwort erhalten Sie die folgenden Informationen.

```
{
    "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"
        }
    ]
}
```

Verwenden Sie den get-reference-metadataAPI-Vorgang, um mehr über die Referenzmetadaten zu erfahren. Ersetzen Sie es im folgenden Beispiel *`reference store ID`* durch Ihre Referenzspeicher-ID und *`reference ID`* durch die Referenz-ID, über die Sie mehr erfahren möchten.

```
aws omics get-reference-metadata --reference-store-id reference store ID --id reference
 ID
```

Als Antwort erhalten Sie die folgenden Informationen.

```
{
    "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
        }
    }
}
```

Sie können Teile der Referenzdatei auch mithilfe von get-reference herunterladen. Im folgenden Beispiel *reference store ID* ersetzen Sie es durch Ihre Referenzspeicher-ID und *reference ID* durch die Referenz-ID, von der Sie herunterladen möchten.

```
aws omics get-reference --reference-store-id reference store ID --id reference ID --
part-number 1 outfile.fa
```

Einen HealthOmics Sequenzspeicher erstellen

HealthOmics Sequenzspeicher unterstützen das Speichern von Genomdateien in den unausgerichteten Formaten FASTQ (nur Gzip) und. uBAM Es unterstützt auch die ausgerichteten Formate von und. BAM CRAM

Importierte Dateien werden als Lesesätze gespeichert. Sie können Tags zu Lesesätzen hinzufügen und den Zugriff auf Lesesätze mithilfe von IAM-Richtlinien steuern. Aligned Read-Sets erfordern ein Referenzgenom, um genomische Sequenzen aufeinander abzustimmen. Für nicht ausgerichtete Read-Sets ist dies jedoch optional.

Um Lesesätze zu speichern, erstellen Sie zunächst einen Sequenzspeicher. Wenn Sie einen Sequenzspeicher erstellen, können Sie einen optionalen Amazon S3 S3-Bucket als Ausweichort und den Ort angeben, an dem S3-Zugriffsprotokolle gespeichert werden. Der Fallback-Speicherort wird zum Speichern von Dateien verwendet, für die während eines direkten Uploads kein Lesesatz erstellt werden kann. Ausweichspeicherorte sind für Sequenzspeicher verfügbar, die nach dem 15. Mai 2023 erstellt wurden. Sie geben den Fallback-Speicherort an, wenn Sie den Sequenzspeicher erstellen.

Sie können bis zu fünf Read-Set-Tag-Schlüssel angeben. Wenn Sie einen Lesesatz mit einem Tag-Schlüssel erstellen oder aktualisieren, der einem dieser Schlüssel entspricht, werden die Read-Set-Tags an das entsprechende Amazon S3 S3-Objekt weitergegeben. System-Tags, die von erstellt wurden, HealthOmics werden standardmäßig weitergegeben.

Themen

Einen Sequenzspeicher mithilfe der Konsole erstellen

Um einen Sequenzspeicher zu erstellen

1. Öffnen Sie die HealthOmics -Konsole.
2. Öffnen Sie bei Bedarf den linken Navigationsbereich (←). Wählen Sie Sequence Stores.
3. Geben Sie auf der Seite Sequenzspeicher erstellen die folgenden Informationen ein

   - Name des Sequenzspeichers — Ein eindeutiger Name für diesen Speicher.
   - Beschreibung (optional) — Eine Beschreibung dieses Sequenzspeichers.
4. Geben Sie für Fallback-Standort in S3 einen Amazon S3 S3-Standort an. HealthOmics verwendet den Ausweichspeicherort zum Speichern von Dateien, für die beim direkten Upload kein Lesesatz erstellt werden kann. Sie müssen dem HealthOmics Service Schreibzugriff auf den Amazon S3 S3-Fallback-Standort gewähren. Eine Beispielrichtlinie finden Sie unter Konfigurieren Sie einen Fallback-Standort.

   Ausweichstandorte sind für Sequenzspeicher, die vor dem 16. Mai 2023 erstellt wurden, nicht verfügbar.
5. (Optional) Für Read-Set-Tag-Schlüssel für die S3-Weitergabe können Sie bis zu fünf Read-Set-Schlüssel eingeben, um sie von einem Read-Set an die zugrunde liegenden S3-Objekte weiterzugeben. Durch die Weitergabe von Tags aus einem Lesesatz an das S3-Objekt können Sie and/or Endbenutzern S3-Zugriffsberechtigungen auf der Grundlage von Tags gewähren, um die weitergegebenen Tags über den Amazon S3 getObjectTagging S3-API-Vorgang zu sehen.

   a. Geben Sie einen Schlüsselwert in das Textfeld ein. Die Konsole erstellt ein neues Textfeld, um den nächsten Schlüssel hinzuzufügen.

   b. (Optional) Wählen Sie Entfernen, um alle Schlüssel zu entfernen.

6. Wählen Sie unter Datenverschlüsselung aus, ob die Datenverschlüsselung Eigentum und Verwaltung durch einen vom Kunden verwalteten CMK sein soll AWS oder ob ein vom Kunden verwaltetes CMK verwendet werden soll.
7. (Optional) Wählen Sie unter S3-Datenzugriff aus, ob Sie eine neue Rolle und Richtlinie für den Zugriff auf den Sequenzspeicher über Amazon S3 erstellen möchten.
8. (Optional) Wählen Sie für die S3-Zugriffsprotokollierung aus, `Enabled` ob Amazon S3 Zugriffsprotokolldatensätze sammeln soll.

   Geben Sie für den Speicherort für die Zugriffsprotokollierung in S3 einen Amazon S3 S3-Speicherort an, an dem die Protokolle gespeichert werden sollen. Dieses Feld ist nur sichtbar, wenn Sie die S3-Zugriffsprotokollierung aktiviert haben.
9. Tags (optional) — Stellen Sie bis zu 50 Tags für diesen Sequenzspeicher bereit. Diese Tags unterscheiden sich von den Read-Set-Tags, die während der import/tag Aktualisierung des Lesesatzes gesetzt werden

Nachdem Sie den Shop erstellt haben, ist er bereit fürGenomdateien importieren.

Erstellen eines Sequenzspeichers mit der CLI

Ersetzen Sie es im folgenden Beispiel *`sequence store name`* durch den Namen, den Sie für Ihren Sequenzspeicher gewählt haben.

```
aws omics create-sequence-store --name sequence store name --fallback-location "s3://
amzn-s3-demo-bucket"
```

Sie erhalten die folgende Antwort in JSON, die die ID-Nummer für Ihren neu erstellten Sequenzspeicher enthält.

```
{
    "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"
}
```

Sie können auch alle mit Ihrem Konto verknüpften Sequenzspeicher mithilfe des list-sequence-storesBefehls anzeigen, wie im Folgenden gezeigt.

```
aws omics list-sequence-stores
```

Sie erhalten die folgende Antwort.

```
{
    "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"
        }
    ]
}
```

Sie können get-sequence-storedamit mehr über einen Sequenzspeicher erfahren, indem Sie dessen ID verwenden, wie im folgenden Beispiel gezeigt:

```
aws omics get-sequence-store --id sequence store ID
```

Sie erhalten die folgende Antwort:

```
{
  "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"],
}
```

Nach der Erstellung können auch mehrere Speicherparameter aktualisiert werden. Dies kann über die Konsole oder die `updateSequenceStore` API-Operation erfolgen.

Aktualisierung eines Sequenzspeichers

Gehen Sie folgendermaßen vor, um einen Sequenzspeicher zu aktualisieren:

1. Öffnen Sie die HealthOmics -Konsole.
2. Öffnen Sie bei Bedarf den linken Navigationsbereich (•). Wählen Sie Sequence Stores.
3. Wählen Sie den Sequenzspeicher aus, der aktualisiert werden soll.
4. Wählen Sie im Bereich „Details“ die Option „Bearbeiten“.
5. Auf der Seite „Details bearbeiten“ können Sie die folgenden Felder aktualisieren:

   - Name des Sequenzspeichers — Ein eindeutiger Name für diesen Shop.
   - Beschreibung — Eine Beschreibung dieses Sequenzspeichers.
   - Fallback-Standort in S3, geben Sie einen Amazon S3 S3-Standort an. HealthOmics verwendet den Ausweichspeicherort zum Speichern von Dateien, für die beim direkten Upload kein Lesesatz erstellt werden kann.
   - Read-Set-Tag-Schlüssel für die S3-Weitergabe Sie können bis zu fünf Read-Set-Schlüssel für die Weitergabe an Amazon S3 eingeben.
   - (Optional) Wählen Sie für die S3-Zugriffsprotokollierung aus, `Enabled` ob Amazon S3 Zugriffsprotokolldatensätze sammeln soll.

     Geben Sie für den Speicherort für die Zugriffsprotokollierung in S3 einen Amazon S3 S3-Speicherort an, an dem die Protokolle gespeichert werden sollen. Dieses Feld ist nur sichtbar, wenn Sie die S3-Zugriffsprotokollierung aktiviert haben.
   - Tags (optional) — Stellen Sie bis zu 50 Tags für diesen Sequenzspeicher bereit.

Read-Set-Tags für einen Sequenzspeicher werden aktualisiert

Gehen Sie wie folgt vor, um Read-Set-Tags oder andere Felder für einen Sequenzspeicher zu aktualisieren:

1. Öffnen Sie die HealthOmics -Konsole.
2. Öffnen Sie bei Bedarf den linken Navigationsbereich (←). Wählen Sie Sequence Stores.
3. Wählen Sie den Sequenzspeicher aus, den Sie aktualisieren möchten.
4. Wählen Sie die Registerkarte Details.
5. Wählen Sie Bearbeiten aus.
6. Fügen Sie nach Bedarf neue Lesesatz-Tags hinzu oder löschen Sie vorhandene Tags.
7. Aktualisieren Sie nach Bedarf den Namen, die Beschreibung, den Ausweichort oder den S3-Datenzugriff.
8. Wählen Sie Änderungen speichern aus.

Genomdateien importieren

Gehen Sie folgendermaßen vor, um genomische Dateien in einen Sequenzspeicher zu importieren:

Um eine Genomikdatei zu importieren

1. Öffnen Sie die HealthOmics -Konsole.
2. Öffnen Sie bei Bedarf den linken Navigationsbereich (•). Wählen Sie „Sequenzspeicher auswählen“.
3. Wählen Sie auf der Seite Sequenzspeicher den Sequenzspeicher aus, in den Sie Ihre Dateien importieren möchten.
4. Wählen Sie auf der Seite mit den einzelnen Sequenzspeichern die Option Genomische Dateien importieren aus.
5. Geben Sie auf der Seite Importdetails angeben die folgenden Informationen ein

   - IAM-Rolle — Die IAM-Rolle, die auf die genomischen Dateien in Amazon S3 zugreifen kann.
   - Referenzgenom — Das Referenzgenom für diese Genomdaten.
6. Geben Sie auf der Seite Importmanifest angeben die folgende Informations-Manifestdatei an. Die Manifestdatei ist eine JSON- oder YAML-Datei, die wichtige Informationen Ihrer Genomdaten

beschreibt. Hinweise zur Manifestdatei finden Sie unter. Lesesätze in einen HealthOmics Sequenzspeicher importieren

7. Klicken Sie auf Importjob erstellen.

HealthOmics Referenz- und Sequenzspeicher löschen

Sowohl Referenz- als auch Sequenzspeicher können gelöscht werden. Sequenzspeicher können nur gelöscht werden, wenn sie keine Lesesätze enthalten, und Referenzspeicher können nur gelöscht werden, wenn sie keine Verweise enthalten. Durch das Löschen einer Sequenz oder eines Referenzspeichers werden auch alle mit diesem Speicher verknüpften Tags gelöscht.

Das folgende Beispiel zeigt, wie Sie einen Referenzspeicher mithilfe von löschen. AWS CLI Wenn die Aktion erfolgreich ist, erhalten Sie keine Antwort. Ersetzen Sie es im folgenden Beispiel *reference store ID* durch Ihre Referenz-Store-ID.

```
aws omics delete-reference-store --id reference store ID
```

Das folgende Beispiel zeigt Ihnen, wie Sie einen Sequenzspeicher löschen. Sie erhalten keine Antwort, wenn die Aktion erfolgreich ist. Im folgenden Beispiel ersetzen Sie es *sequence store ID* durch Ihre Sequenzspeicher-ID.

```
aws omics delete-sequence-store --id sequence store ID
```

Sie können auch eine Referenz in einem Referenzspeicher löschen, wie im folgenden Beispiel gezeigt. Verweise können nur gelöscht werden, wenn sie nicht in einem Lesesatz, Variantenspeicher oder Annotationsspeicher verwendet werden. Ersetzen Sie im folgenden Beispiel durch *reference store ID* Ihre Referenzspeicher-ID und *reference ID* ersetzen Sie sie durch die ID der Referenz, die Sie löschen möchten.

```
aws omics delete-reference  --id reference ID --reference-store-id reference store ID
```

Lesesätze in einen HealthOmics Sequenzspeicher importieren

Nachdem Sie Ihren Sequenzspeicher erstellt haben, erstellen Sie Importaufträge, um Lesesätze in den Datenspeicher hochzuladen. Sie können Ihre Dateien aus einem Amazon S3 S3-Bucket

hochladen oder Sie können sie direkt hochladen, indem Sie die synchronen API-Operationen verwenden. Ihr Amazon S3 S3-Bucket muss sich in derselben Region wie Ihr Sequence Store befinden.

Sie können eine beliebige Kombination aus ausgerichteten und nicht ausgerichteten Lesesätzen in Ihren Sequenzspeicher hochladen. Wenn jedoch einer der Lesesätze in Ihrem Import ausgerichtet ist, müssen Sie ein Referenzgenom angeben.

Sie können die IAM-Zugriffsrichtlinie, die Sie zur Erstellung des Referenzspeichers verwendet haben, wiederverwenden.

In den folgenden Themen werden die wichtigsten Schritte beschrieben, die Sie ausführen, um ein Leseset in Ihren Sequenzspeicher zu importieren und anschließend Informationen zu den importierten Daten abzurufen.

Themen

Dateien auf Amazon S3 hochladen

Das folgende Beispiel zeigt, wie Sie Dateien in Ihren Amazon S3 S3-Bucket verschieben.

```
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
```

Die in diesem Beispiel CRAM verwendete Probe BAM erfordert unterschiedliche Genomreferenzen, Hg19 undHg38. Weitere Informationen oder Zugriff auf diese Referenzen finden Sie unter The Broad Genome References in the Registry of Open Data unter AWS.

Erstellen einer Manifestdatei

Sie müssen außerdem eine Manifestdatei in JSON erstellen, in der Sie den Importauftrag modellieren können `import.json` (siehe das folgende Beispiel). Wenn Sie in der Konsole einen Sequenzspeicher erstellen, müssen Sie das `sequenceStoreId` oder nicht angeben`roleARN`, sodass Ihre Manifestdatei mit der `sources` Eingabe beginnt.

API manifest

Im folgenden Beispiel werden drei Lesesätze mithilfe der API importiert: eins FASTQBAM, eins und einsCRAM.

```
{
  "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

Dieser Beispielcode wird verwendet, um einen einzelnen Lesesatz mithilfe der Konsole zu importieren.

```
[
  {
      "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"
    }
]
```

Alternativ können Sie die Manifestdatei im YAML-Format hochladen.

Der Importjob wird gestartet

Verwenden Sie den folgenden AWS CLI Befehl, um den Importjob zu starten.

```
aws omics start-read-set-import-job --cli-input-json file://import.json
```

Sie erhalten die folgende Antwort, die darauf hinweist, dass der Job erfolgreich erstellt wurde.

```
{
  "id": "3660451514",
  "sequenceStoreId": "3936421177",
  "roleArn": "arn:aws:iam::111122223333:role/OmicsImport",
  "status": "CREATED",
  "creationTime": "2022-07-13T22:14:59.309Z"
}
```

Überwachen Sie den Importauftrag

Nach dem Start des Importauftrags können Sie seinen Fortschritt mit dem folgenden Befehl überwachen. Ersetzen Sie im folgenden Beispiel *`sequence store id`* durch Ihre Sequenzspeicher-ID und dann *`job import ID`* durch die Import-ID.

```
aws omics get-read-set-import-job --sequence-store-id sequence store id --id job import
 ID
```

Im Folgenden werden die Status aller Importaufträge angezeigt, die der angegebenen Sequenzspeicher-ID zugeordnet sind.

```
{
  "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"
      }
  ]
}
```

Suchen Sie die importierten Sequenzdateien

Nach Abschluss des Jobs können Sie den list-read-setsAPI-Vorgang verwenden, um die importierten Sequenzdateien zu finden. Im folgenden Beispiel ersetzen Sie es *`sequence store id`* durch Ihre Sequenzspeicher-ID.

```
aws omics list-read-sets --sequence-store-id sequence store id
```

Sie erhalten die folgende Antwort.


```
{
  "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"
            }
        }
    ]
}
```

Rufen Sie Details zu einem Lesesatz ab

Verwenden Sie die GetReadSetMetadataAPI-Operation, um weitere Details zu einem Lesesatz anzuzeigen. Ersetzen Sie im folgenden Beispiel *`sequence store id`* durch Ihre Sequenzspeicher-ID und dann *`read set id`* durch Ihre Leseset-ID.

```
aws omics get-read-set-metadata --sequence-store-id sequence store id --id read set id
```

Sie erhalten die folgende Antwort.

```
{
"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"
}
```


Laden Sie die Readset-Datendateien herunter

Sie können mithilfe der Amazon S3 GetObject S3-API-Operation auf die Objekte für einen aktiven Lesesatz zugreifen. Die URI für das Objekt wird in der GetReadSetMetadataAPI-Antwort

zurückgegeben. Weitere Informationen finden Sie unter Zugreifen auf HealthOmics Lesesätze mit Amazon S3 URIs.

Verwenden Sie alternativ den HealthOmics GetReadSet API-Vorgang. Sie können GetReadSet das parallel Herunterladen verwenden, indem Sie einzelne Teile herunterladen. Diese Teile ähneln Amazon S3 S3-Teilen. Im Folgenden finden Sie ein Beispiel dafür, wie Sie Teil 1 aus einem Lesesatz herunterladen können. Im folgenden Beispiel *`sequence store id`* ersetzen Sie es durch Ihre Sequenzspeicher-ID und *`read set id`* ersetzen Sie es durch Ihre Lesesatz-ID.

```
aws omics get-read-set --sequence-store-id sequence store id --id read set id  --part-
number 1 outfile.bam
```

Sie können den HealthOmics Transfer Manager auch verwenden, um Dateien als HealthOmics Referenz oder Lesesatz herunterzuladen. Sie können den HealthOmics Transfer Manager hier herunterladen. Weitere Informationen zur Verwendung und Einrichtung des Transfer Managers finden Sie in diesem GitHubRepository.

Direkter Upload in einen HealthOmics Sequenzspeicher

Wir empfehlen, dass Sie den HealthOmics Transfer Manager verwenden, um Dateien zu Ihrem Sequenzspeicher hinzuzufügen. Weitere Informationen zur Verwendung von Transfer Manager finden Sie in diesem GitHubRepository. Sie können Ihre Lesesätze auch direkt über die API-Operationen für den direkten Upload in einen Sequenzspeicher hochladen.

Direkte Upload-Lesätze existieren zuerst im `PROCESSING_UPLOAD` Status. Das bedeutet, dass die Teile der Datei gerade hochgeladen werden und Sie auf die Metadaten des Lesesatzes zugreifen können. Nachdem die Teile hochgeladen und die Prüfsummen validiert wurden, wird der Lesesatz zu einem importierten Lesesatz `ACTIVE` und verhält sich genauso wie ein importierter Lesesatz.

Wenn der direkte Upload fehlschlägt, wird der Status des Lesesatzes als angezeigt. `UPLOAD_FAILED` Sie können einen Amazon S3 S3-Bucket als Fallback-Speicherort für Dateien konfigurieren, die nicht hochgeladen werden können. Ausweichspeicherorte sind für Sequenzspeicher verfügbar, die nach dem 15. Mai 2023 erstellt wurden.

Themen

- Direkter Upload in einen Sequenzspeicher mit dem AWS CLI
- Konfigurieren Sie einen Fallback-Standort

Direkter Upload in einen Sequenzspeicher mit dem AWS CLI

Starten Sie zunächst einen mehrteiligen Upload. Sie können dies tun, indem Sie die verwenden AWS CLI, wie im folgenden Beispiel gezeigt.

Zum direkten Upload mithilfe von AWS CLI Befehlen

1. Erstellen Sie die Teile, indem Sie Ihre Daten trennen, wie im folgenden Beispiel gezeigt.

```
split -b 100MiB SRR233106_1.filt.fastq.gz source1_part_
```

2. Nachdem Ihre Quelldateien in Teilen vorliegen, erstellen Sie einen mehrteiligen Lesesatz-Upload, wie im folgenden Beispiel gezeigt. Ersetzen Sie *sequence store ID* und die anderen Parameter durch Ihre Sequenzspeicher-ID und andere Werte.

```
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"
```

   Sie erhalten die `uploadID` und andere Metadaten in der Antwort. Verwenden Sie die `uploadID` für den nächsten Schritt des Upload-Vorgangs.

```
{
"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. Fügen Sie Ihre Lesesätze zum Upload hinzu. Wenn Ihre Datei klein genug ist, müssen Sie diesen Schritt nur einmal ausführen. Bei größeren Dateien führen Sie diesen Schritt für jeden

Teil Ihrer Datei aus. Wenn Sie ein neues Teil unter Verwendung einer zuvor verwendeten Artikelnummer hochladen, überschreibt es das zuvor hochgeladene Teil.

Im folgenden Beispiel ersetzen Sie *sequence store IDupload ID*, und die anderen Parameter durch Ihre Werte.

```
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
```

Die Antwort ist eine ID, anhand derer Sie überprüfen können, ob die hochgeladene Datei mit der von Ihnen beabsichtigten Datei übereinstimmt.

```
{
"checksum": "984979b9928ae8d8622286c4a9cd8e99d964a22d59ed0f5722e1733eb280e635"
}
```

4. Fahren Sie gegebenenfalls mit dem Hochladen der Teile Ihrer Datei fort. Um zu überprüfen, ob Ihre Lesesätze hochgeladen wurden, verwenden Sie den API-Vorgang list-read-set-upload-parts, wie im Folgenden gezeigt. Im folgenden Beispiel ersetzen Sie *sequence store ID upload ID*, und the *part source* durch Ihre eigene Eingabe.

```
aws omics list-read-set-upload-parts \
 --sequence-store-id sequence store ID \
 --upload-id upload ID \
 --part-source SOURCE1
```

Die Antwort gibt die Anzahl der Lesesätze, die Größe und den Zeitstempel der letzten Aktualisierung zurück.

```
{
"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. Um alle aktiven Uploads von mehrteiligen Lesesätzen anzuzeigen, verwenden Sie list-multipart-read-set-uploads, wie im Folgenden gezeigt. Ersetzen Sie es *`sequence store ID`* durch die ID für Ihren eigenen Sequenzspeicher.

```
aws omics list-multipart-read-set-uploads --sequence-store-id
            sequence store ID
```

Diese API gibt nur mehrteilige Read-Set-Uploads zurück, die gerade ausgeführt werden. Sobald die aufgenommenen Lesesätze vorhanden sind **`ACTIVE`** oder wenn der Upload fehlgeschlagen ist, wird der Upload in der Antwort an die -uploads-API nicht zurückgegeben. list-multipart-read-set Verwenden Sie die API, um aktive Lesesätze anzuzeigen. list-read-sets Im Folgenden wird ein Beispiel für eine Antwort für list-multipart-read-set-uploads gezeigt.

```
{
"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. Nachdem Sie alle Teile Ihrer Datei hochgeladen haben, verwenden Sie complete-multipart-read-set-upload, um den Upload-Vorgang abzuschließen, wie im folgenden Beispiel gezeigt. Ersetzen Sie *sequence store IDupload ID*, und den Parameter für Teile durch Ihre eigenen Werte.

```
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"}]'
```

Die Antwort für complete-multipart-read-set-upload ist der Lesesatz IDs für Ihre importierten Lesesätze.

```
{
"readSetId": "0000000001"
}
```

7. Um den Upload zu beenden, verwenden Sie abort-multipart-read-set-upload zusammen mit der Upload-ID, um den Upload-Vorgang zu beenden. Ersetzen Sie *sequence store ID* und *upload ID* durch Ihre eigenen Parameterwerte.

```
aws omics abort-multipart-read-set-upload \
--sequence-store-id sequence store ID \
--upload-id upload ID
```

8. Nachdem der Upload abgeschlossen ist, rufen Sie Ihre Daten aus dem Lesesatz ab get-read-set, indem Sie, wie im Folgenden gezeigt, verwenden. Wenn der Upload noch in Bearbeitung ist, werden nur begrenzte Metadaten get-read-setzurückgegeben und die generierten Indexdateien sind nicht verfügbar. Ersetzen Sie *sequence store ID* und die anderen Parameter durch Ihre eigene Eingabe.

```
aws omics get-read-set
 --sequence-store-id sequence store ID \
 --id read set ID \
 --file SOURCE1 \
 --part-number 1 myfile.fastq.gz
```

9. Verwenden Sie die get-read-set-metadataAPI-Operation, um die Metadaten, einschließlich des Status Ihres Uploads, zu überprüfen.

```
aws omics get-read-set-metadata --sequence-store-id sequence store ID --id read set
 ID
```

Die Antwort enthält Metadatendetails wie den Dateityp, den Referenz-ARN, die Anzahl der Dateien und die Länge der Sequenzen. Sie enthält auch den Status. Mögliche Status sind `PROCESSING_UPLOADACTIVE`, und`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"
}
```

Konfigurieren Sie einen Fallback-Standort

Wenn Sie einen Sequenzspeicher erstellen oder aktualisieren, können Sie einen Amazon S3 S3-Bucket als Ausweichort für Dateien konfigurieren, die nicht hochgeladen werden können. Die Dateiteile für diese Lesesätze werden an den Ersatzspeicherort übertragen. Ausweichspeicherorte sind für Sequenzspeicher verfügbar, die nach dem 15. Mai 2023 erstellt wurden.

Erstellen Sie eine Amazon S3 S3-Bucket-Richtlinie, um HealthOmics Schreibzugriff auf den Amazon S3 S3-Fallback-Speicherort zu gewähren, wie im folgenden Beispiel gezeigt:

```
{
    "Effect": "Allow",
```

```
    "Principal": {
        "Service": "omics.amazonaws.com"
    },
    "Action": "s3:PutObject",
    "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
}
```

Wenn der Amazon S3 S3-Bucket für Fallback- oder Zugriffsprotokolle einen vom Kunden verwalteten Schlüssel verwendet, fügen Sie der Schlüsselrichtlinie die folgenden Berechtigungen hinzu:

```
 {
    "Sid": "Allow use of key",
    "Effect": "Allow",
    "Principal": {
        "Service": "omics.amazonaws.com"
    },
    "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey*"
    ],
    "Resource": "*"
}
```

Exportieren von HealthOmics Lesesätzen in einen Amazon S3 S3-Bucket

Sie können Lesesätze als Batch-Exportjob in einen Amazon S3 S3-Bucket exportieren. Erstellen Sie dazu zunächst eine IAM-Richtlinie, die Schreibzugriff auf den Bucket hat, ähnlich dem folgenden Beispiel für eine IAM-Richtlinie.

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"
  }
]
}
```

Sobald die IAM-Richtlinie eingerichtet ist, beginnen Sie mit dem Exportauftrag für Lesesätze. Das folgende Beispiel zeigt Ihnen, wie Sie dies mithilfe der API-Operation start-read-set-export-job tun können. Ersetzen Sie im folgenden Beispiel alle Parameter wie*sequence store ID*, *destination role ARNsources*, und durch Ihre Eingabe.

```
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
```

Sie erhalten die folgende Antwort mit Informationen zum Ursprungssequenzspeicher und zum Amazon S3-Ziel-Bucket.

```
{
"id": <job-id>,
"sequenceStoreId": <sequence-store-id>,
"destination": <destination-s3-uri>,
"status": "SUBMITTED",
"creationTime": "2022-10-22T01:33:38.079000+00:00"
}
```

Nach dem Start des Jobs können Sie seinen Status mithilfe der API-Operation get-read-set-export-job ermitteln, wie im Folgenden dargestellt. Ersetzen Sie das *sequence store ID* und *job ID* durch Ihre Sequenzspeicher-ID bzw. Ihre Job-ID.

```
aws omics get-read-set-export-job --id job-id --sequence-store-id sequence store ID
```

Sie können alle Exportaufträge anzeigen, die für einen Sequenzspeicher initialisiert wurden, indem Sie den API-Vorgang list-read-set-export-jobs verwenden, wie im Folgenden gezeigt. Ersetzen Sie das *sequence store ID* durch Ihre Sequenzspeicher-ID.

```
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"
  }
]
}
```

Sie können Ihre Lesesätze nicht nur exportieren, sondern sie auch über den Amazon S3 S3-Zugriff teilen URIs. Weitere Informationen hierzu finden Sie unter Zugreifen auf HealthOmics Lesesätze mit Amazon S3 URIs.

Zugreifen auf HealthOmics Lesesätze mit Amazon S3 URIs

Sie können Amazon S3 S3-URI-Pfade verwenden, um auf Ihre Active Sequence Store-Lesesätze zuzugreifen.

Mit dem Amazon S3 S3-URI-Pfad können Sie Amazon S3 S3-Operationen verwenden, um Ihre Lesesätze aufzulisten, zu teilen und herunterzuladen. Der Zugriff über S3 APIs beschleunigt die Zusammenarbeit und die Integration von Tools, da viele Industrietools bereits für das Lesen aus S3 entwickelt wurden. Darüber hinaus können Sie den Zugriff auf das S3 APIs mit anderen Konten teilen und regionsübergreifenden Lesezugriff auf Daten gewähren.

HealthOmics unterstützt keinen Amazon S3 S3-URI-Zugriff auf archivierte Lesesätze. Wenn Sie einen Lesesatz aktivieren, wird er jedes Mal auf demselben URI-Pfad wiederhergestellt.

Da die Amazon S3-URI auf Amazon S3 S3-Zugriffspunkten basiert, können Sie Daten direkt in branchenübliche Tools integrieren, die Amazon S3 lesen URIs, wie z. B. die folgenden: HealthOmics

- Visuelle Analyseanwendungen wie Integrative Genomics Viewer (IGV) oder UCSC Genome Browser.
- Allgemeine Workflows mit Amazon S3 S3-Erweiterungen wie CWL, WDL und Nextflow.
- Jedes Tool, das sich authentifizieren und vom Access Point Amazon S3 aus lesen URIs oder vorsigniertes Amazon S3 lesen kann. URIs
- Amazon S3 S3-Dienstprogramme wie Mountpoint oder. CloudFront

Amazon S3 Mountpoint ermöglicht es Ihnen, einen Amazon S3 S3-Bucket als lokales Dateisystem zu verwenden. Weitere Informationen zu Mountpoint und zur Installation zur Verwendung finden Sie unter Mountpoint for Amazon S3.

Amazon CloudFront ist ein Content Delivery Network (CDN) -Service, der auf hohe Leistung, Sicherheit und Entwicklerkomfort ausgelegt ist. Weitere Informationen zur Verwendung von Amazon CloudFront finden Sie in der CloudFront Amazon-Dokumentation. Wenden Sie sich CloudFront an das AWS HealthOmics Team, um einen Sequenzspeicher einzurichten.

Das Root-Konto des Datenbesitzers ist für die Aktionen S3:GetObject, S3: GetObjectTagging und S3:List Bucket für das Sequenzspeicherpräfix aktiviert. Damit ein Benutzer im Konto auf die Daten zugreifen kann, erstellen Sie eine IAM-Richtlinie und fügen sie dem Benutzer oder der Rolle hinzu. Eine Beispielrichtlinie finden Sie unter Berechtigungen für den Datenzugriff mit Amazon S3 URIs.

Sie können die folgenden Amazon S3 S3-API-Operationen für die aktiven Lesesätze verwenden, um Ihre Daten aufzulisten und abzurufen. Sie können über Amazon S3 auf archivierte Lesesätze zugreifen, URIs nachdem sie aktiviert wurden.

- GetObject— Ruft ein Objekt von Amazon S3 ab.
- HeadObject— Die HEAD-Operation ruft Metadaten von einem Objekt ab, ohne das Objekt selbst zurückzugeben. Diese Operation ist nützlich, wenn Sie nur die Metadaten eines Objekts benötigen.
- ListObjects und ListObject v2 — Gibt einige oder alle (bis zu 1.000) Objekte in einem Bucket zurück.
- CopyObject— Erstellt eine Kopie eines Objekts, das bereits in Amazon S3 gespeichert ist. HealthOmicsunterstützt das Kopieren auf einen Amazon S3 S3-Zugriffspunkt, aber nicht das Schreiben auf einen Zugriffspunkt.

HealthOmics Sequenzspeicher behalten die semantische Identität von Dateien ETags durchgehend bei. Während des Lebenszyklus einer Datei kann sich der Amazon S3 ETag, der auf einer bitweisen Identität basiert, ändern, die HealthOmics ETag bleibt jedoch gleich. Weitere Informationen hierzu finden Sie unter HealthOmics ETags und Herkunft der Daten.

Themen

- Amazon S3 S3-URI-Struktur im HealthOmics Speicher
- Verwenden von gehostetem oder lokalem IGV für den Zugriff auf Lesesätze
- Mit Samtools oder in HTSlib HealthOmics
- Mit Mountpoint HealthOmics
- Verwenden CloudFront mit HealthOmics

Amazon S3 S3-URI-Struktur im HealthOmics Speicher

Alle Dateien mit Amazon S3 URIs haben `omics:subjectId` `omics:sampleId` Ressourcen-Tags. Sie können diese Tags verwenden, um den Zugriff gemeinsam zu nutzen, indem Sie IAM-Richtlinien nach einem Muster wie `"s3:ExistingObjectTag/omics:subjectId": "pattern desired"` verwenden.

Die Dateistruktur sieht wie folgt aus:

`.../`*`account_id`*`/sequenceStore/`*`seq_store_id`*`/readSet/`*`read_set_id`*`/`*`files`*`.`

Bei Dateien, die aus Amazon S3 in Sequenzspeicher importiert wurden, versucht der Sequenzspeicher, den ursprünglichen Quellnamen beizubehalten. Wenn die Namen miteinander in Konflikt geraten, hängt das System Lesesatzinformationen an, um sicherzustellen, dass die Dateinamen eindeutig sind. Wenn zum Beispiel bei Fastq-Lesets beide Dateinamen identisch sind, `sourceX` wird, um die Namen eindeutig zu machen, vor .fastq.gz oder .fq.gz eingefügt. Bei einem direkten Upload folgen die Dateinamen den folgenden Mustern:

- Für FASTQ— *`read_set_name`* _ .fastq.gz *`sourcex`*
- uBAM/BAM/CRAMFür *`read_set_name`* —. *`file extension`*mit Erweiterungen von `.bam` oder`.cram`. Ein Beispiel ist `NA193948.bam`.

Bei Lesesätzen, bei denen es sich um BAM oder CRAM handelt, werden Indexdateien während des Aufnahmevorgangs automatisch generiert. Für die generierten Indexdateien wird die richtige Indexerweiterung am Ende des Dateinamens angewendet. Sie hat das Muster *`<name of the Source the index is on>.<file index extension>.`* Die Indexerweiterungen sind `.bai` oder`.crai`.

Verwenden von gehostetem oder lokalem IGV für den Zugriff auf Lesesätze

IGV ist ein Genombrowser, der zur Analyse von BAM- und CRAM-Dateien verwendet wird. Er benötigt sowohl die Datei als auch den Index, da jeweils nur ein Teil des Genoms angezeigt wird. IGV kann heruntergeladen und lokal verwendet werden, und es gibt Anleitungen zur Erstellung eines von AWS gehosteten IGV. Die öffentliche Webversion wird nicht unterstützt, da sie CORS benötigt.

Lokales IGV ist für den Zugriff auf Dateien auf die lokale AWS Konfiguration angewiesen. Stellen Sie sicher, dass der in dieser Konfiguration verwendeten Rolle eine Richtlinie zugewiesen ist, die kms: Decrypt- und s3: GetObject -Berechtigungen für die s3-URI der Lesesätze, auf die zugegriffen wird, aktiviert. Danach können Sie in IGV „Datei > Aus URL laden“ verwenden und den URI für die Quelle und den Index einfügen. Alternativ URLs kann Presigned auf dieselbe Weise generiert und verwendet werden, wodurch die AWS-Konfiguration umgangen wird. Beachten Sie, dass CORS beim Amazon S3 S3-URI-Zugriff nicht unterstützt wird, sodass Anfragen, die auf CORS basieren, nicht unterstützt werden.

Das Beispiel AWS Hosted IGV stützt sich auf AWS Cognito, um die richtigen Konfigurationen und Berechtigungen innerhalb der Umgebung zu erstellen. Stellen Sie sicher, dass eine Richtlinie erstellt wurde, die die GetObject Berechtigungen KMS:Decrypt und s3: für die Amazon S3 S3-URI der Lesesätze aktiviert, auf die zugegriffen wird, und fügen Sie diese Richtlinie der Rolle hinzu, die dem Cognito-Benutzerpool zugewiesen ist. Danach können Sie in IGV „Datei > Von URL laden“

verwenden und den URI für die Quelle und den Index eingeben. Alternativ URLs kann Presigned auf dieselbe Weise generiert und verwendet werden, wodurch die AWS-Konfiguration umgangen wird.

Beachten Sie, dass der Sequenzspeicher nicht auf der Registerkarte „Amazon“ angezeigt wird, da dort nur Buckets angezeigt werden, die Ihnen in der Region gehören, in der das AWS Profil konfiguriert ist.

Mit Samtools oder in HTSlib HealthOmics

HTSlib ist die Kernbibliothek, die von mehreren Tools wie Samtools, RSAMTools und anderen gemeinsam genutzt wird. PySam Verwenden Sie HTSlib Version 1.20 oder höher, um nahtlose Unterstützung für Amazon S3 Access Points zu erhalten. Für ältere Versionen der HTSlib Bibliothek können Sie die folgenden Problemumgehungen verwenden:

- Legen Sie die Umgebungsvariable für den HTS Amazon S3 S3-Host fest mit:`export HTS_S3_HOST="s3.region.amazonaws.com"`.
- Generieren Sie eine vorsignierte URL für die Dateien, die Sie verwenden möchten. Wenn ein BAM oder CRAM verwendet wird, stellen Sie sicher, dass sowohl für die Datei als auch für den Index eine vorsignierte URL generiert wird. Danach können beide Dateien mit den Bibliotheken verwendet werden.
- Verwenden Sie Mountpoint, um das Sequenzspeicher- oder Lesesatzpräfix in derselben Umgebung zu mounten, in der Sie Bibliotheken verwenden HTSlib . Von hier aus kann über lokale Dateipfade auf die Dateien zugegriffen werden.

Mit Mountpoint HealthOmics

Mountpoint for Amazon S3 ist ein einfacher Dateiclient mit hohem Durchsatz zum Mounten eines Amazon S3 S3-Buckets als lokales Dateisystem. Mit Mountpoint für Amazon S3 können Ihre Anwendungen über Dateioperationen wie Öffnen und Lesen auf in Amazon S3 gespeicherte Objekte zugreifen. Mountpoint for Amazon S3 übersetzt diese Operationen automatisch in Amazon S3-Objekt-API-Aufrufe, sodass Ihre Anwendungen über eine Dateischnittstelle auf den elastischen Speicher und den Durchsatz von Amazon S3 zugreifen können.

Mountpoint kann mithilfe der Mountpoint-Installationsanweisungen installiert werden. Mountpoint verwendet das AWS-Profil, das für die Installation lokal ist und auf einer Amazon S3 S3-Präfixebene funktioniert. Stellen Sie sicher, dass das verwendete Profil über eine Richtlinie verfügt, die die Berechtigungen s3:GetObject, s3: ListBucket und kms: Decrypt für das Amazon S3 S3-URI-Präfix der

Lesesätze oder Sequenzspeicher, auf die zugegriffen wird, aktiviert. Danach kann der Bucket mithilfe des folgenden Pfads bereitgestellt werden:

```
mount-s3 access point arn local path to mount --prefix prefix to sequence store or read
 set --region region
```

Verwenden CloudFront mit HealthOmics

Amazon CloudFront ist ein Content Delivery Network (CDN) -Service, der auf hohe Leistung, Sicherheit und Entwicklerkomfort ausgelegt ist. Kunden, die diesen Service nutzen möchten, CloudFront müssen mit dem Serviceteam zusammenarbeiten, um den CloudFront Vertrieb zu aktivieren. Arbeiten Sie mit Ihrem Account-Team zusammen, um das HealthOmics Serviceteam zu engagieren.

Aktivierung von Readsets in HealthOmics

Sie können Lesesätze aktivieren, die archiviert wurden, mit der API-Operation start-read-set-activation-job oder über AWS CLI, wie im folgenden Beispiel gezeigt. Ersetzen Sie das *sequence store ID* und *read set id* durch Ihre Sequenzspeicher-ID und Ihren Lesesatz 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
```

Sie erhalten eine Antwort, die die Informationen zum Aktivierungsauftrag enthält, wie im Folgenden dargestellt.

```
{
    "id": "12345678",
    "sequenceStoreId": "1234567890",
    "status": "SUBMITTED",
    "creationTime": "2022-10-22T00:50:54.670000+00:00"
}
```

Nach dem Start des Aktivierungsauftrags können Sie seinen Fortschritt mit dem API-Vorgang get-read-set-activation-job überwachen. Im Folgenden finden Sie ein Beispiel dafür, wie Sie den verwenden können, AWS CLI um den Status Ihres Aktivierungsauftrags zu überprüfen. Ersetzen Sie *job ID* und *sequence store ID* durch Ihre Sequenzspeicher-ID bzw. Ihren Job IDs.

```
aws omics get-read-set-activation-job --id job ID --sequence-store-id sequence store ID
```

Die Antwort fasst den Aktivierungsjob zusammen, wie im Folgenden dargestellt.

```
{
    "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."
        }
    ]
}
```

Sie können den Status eines Aktivierungsauftrags mit dem get-read-set-metadataAPI-Vorgang überprüfen. Mögliche Status sind `ACTIVEACTIVATING`, und`ARCHIVED`. Im folgenden Beispiel *`sequence store ID`* ersetzen Sie es durch Ihre Sequenzspeicher-ID und *`read set ID`* ersetzen Sie es durch Ihre Lesesatz-ID.

```
aws omics get-read-set-metadata --sequence-store-id sequence store ID --id read set ID
```

Die folgende Antwort zeigt Ihnen, dass der Lesesatz aktiv ist.

```
{
    "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"
    }
}
```

Sie können alle Aufträge zur Aktivierung von Lesesätzen mithilfe von list-read-set-activation-jobs anzeigen, wie im folgenden Beispiel gezeigt. Im folgenden Beispiel ersetzen Sie es *sequence store ID* durch Ihre Sequenzspeicher-ID.

```
aws omics list-read-set-activation-jobs --sequence-store-id sequence store ID
```

Sie erhalten die folgende Antwort.

```
{
    "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 Analytik

> **Important**
>
> AWS HealthOmics Variantenspeicher und Annotationsspeicher stehen Neukunden nicht mehr zur Verfügung. Vorhandene Kunden können den Service weiterhin wie gewohnt verwenden. Weitere Informationen finden Sie unter AWS HealthOmics Änderung der Verfügbarkeit von Variantenspeicher und Annotationsspeicher.

HealthOmics analytics unterstützt die Speicherung und Analyse genomischer Varianten und Annotationen. Analytics bietet zwei Arten von Speicherressourcen: Variantenspeicher und Annotationsspeicher. Sie verwenden diese Ressourcen, um genomische Variantendaten und Annotationsdaten zu speichern, zu transformieren und abzufragen. Nachdem Sie Daten in einen Datenspeicher importiert haben, können Sie Athena verwenden, um erweiterte Analysen der Daten durchzuführen.

Sie können die HealthOmics Konsole oder API verwenden, um Geschäfte zu erstellen und zu verwalten, Daten zu importieren und analytische Speicherdaten mit Mitarbeitern zu teilen.

Variant speichert Unterstützungsdaten in VCF-Formaten, und Annotation speichert Unterstützung TSV/CSV und Formate. GFF3 Genomische Koordinaten werden als auf Null basierende, halbgeschlossene, halboffene Intervalle dargestellt. Wenn sich Ihre Daten im HealthOmics Analysedatenspeicher befinden, wird der Zugriff auf die VCF-Dateien über verwaltet. AWS Lake Formation Anschließend können Sie die VCF-Dateien mithilfe von Amazon Athena abfragen. Abfragen müssen die Athena Query Engine Version 3 verwenden. Weitere Informationen zu den Versionen der Athena-Abfrage-Engine finden Sie in der Amazon Athena Athena-Dokumentation.

Themen

- Abfragen von HealthOmics Analysedaten
- HealthOmics Analytics-Shops teilen

HealthOmics Variantenspeicher erstellen

> **Important**
>
> AWS HealthOmics Variantenspeicher und Annotationsspeicher stehen Neukunden nicht mehr zur Verfügung. Vorhandene Kunden können den Service weiterhin wie gewohnt verwenden. Weitere Informationen finden Sie unter AWS HealthOmics Änderung der Verfügbarkeit von Variantenspeicher und Annotationsspeicher.

In den folgenden Themen wird beschrieben, wie HealthOmics Variantenspeicher mithilfe der Konsole und der API erstellt werden.

Themen

- Einen Variantenspeicher mithilfe der Konsole erstellen
- Einen Variantenspeicher mithilfe der API erstellen

Einen Variantenspeicher mithilfe der Konsole erstellen

Sie können mit der HealthOmics Konsole einen Variantenspeicher erstellen.

1. Öffnen Sie die HealthOmics -Konsole.
2. Öffnen Sie bei Bedarf den linken Navigationsbereich (±1). Wählen Sie Variant Stores aus.
3. Geben Sie auf der Seite Variantenspeicher erstellen die folgenden Informationen ein

   - Name des Variantenspeichers — Ein eindeutiger Name für diesen Shop.
   - Beschreibung (optional) — Eine Beschreibung dieses Variantenspeichers.
   - Referenzgenom — Das Referenzgenom für diesen Variantenspeicher.
   - Datenverschlüsselung — Wählen Sie aus, ob die Datenverschlüsselung von Ihnen AWS oder von Ihnen selbst verwaltet werden soll.
   - Tags (optional) — Geben Sie bis zu 50 Tags für diesen Variantenspeicher an.
4. Wählen Sie Variantenspeicher erstellen aus.

Einen Variantenspeicher mithilfe der API erstellen

Verwenden Sie den HealthOmics `CreateVariantStore` API-Betrieb, um Variantenspeicher zu erstellen. Sie können diesen Vorgang auch mit dem ausführen AWS CLI.

Um einen Variantenspeicher zu erstellen, geben Sie einen Namen für den Speicher und den ARN eines Referenzspeichers an. Der Variantenspeicher ist bereit, Daten aufzunehmen, wenn sich sein Status auf READY ändert.

Im folgenden Beispiel wird der verwendet AWS CLI , um einen Variantenspeicher zu erstellen.

```
aws omics create-variant-store --name myvariantstore \
    --reference referenceArn="arn:aws:omics:us-
west-2:555555555555:referenceStore/123456789/reference/5987565360"
```

Um die Erstellung Ihres Variantenspeichers zu bestätigen, erhalten Sie die folgende Antwort.

```
{
    "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"
}
```

Verwenden Sie die get-variant-storeAPI, um mehr über einen Variantenshop zu erfahren.

```
aws omics get-variant-store --name myvariantstore
```

Sie erhalten die folgende Antwort.

```
{
    "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
}
```

Verwenden Sie die list-variant-storesAPI, um alle mit einem Konto verknüpften Variantenshops anzuzeigen.

```
aws omics list-variant-stores
```

Sie erhalten eine Antwort, in der alle Variantenshops zusammen mit ihrem IDs Status und anderen Details aufgeführt sind, wie in der folgenden Beispielantwort dargestellt.

```
{
    "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
        }
    ]
}
```

Sie können die Antworten für die list-variant-storesAPI auch nach Status oder anderen Kriterien filtern.

VCF-Dateien, die in Analysespeicher importiert wurden, die am oder nach dem 15. Mai 2023 erstellt wurden, enthalten definierte Schemas für VEP-Anmerkungen (Variant Effect Predictor).

Dies erleichtert das Abfragen und Analysieren importierter VCF-Daten. Die Änderung wirkt sich nicht auf Stores aus, die vor dem 15. Mai 2023 erstellt wurden, es sei denn, der `annotation fields` Parameter ist im API- oder CLI-Aufruf enthalten. Für diese Shops führt die Verwendung des `annotation fields` Parameters dazu, dass die Anfrage fehlschlägt.

Importjobs für HealthOmics Variantenspeicher erstellen

> **Important**
>
> AWS HealthOmics Variantenspeicher und Annotationsspeicher stehen neuen Kunden nicht mehr zur Verfügung. Vorhandene Kunden können den Service weiterhin wie gewohnt verwenden. Weitere Informationen finden Sie unter AWS HealthOmics Änderung der Verfügbarkeit von Variantenspeicher und Annotationsspeicher.

Das folgende Beispiel zeigt, wie Sie mit AWS CLI dem einen Importjob für einen Variantenspeicher erstellen können.

```
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"}
    ]
}
```

Für Geschäfte, die nach dem 15. Mai 2023 erstellt wurden, zeigt das folgende Beispiel, wie der `--annotation-fields` Parameter hinzugefügt wird. Die Annotationsfelder werden beim Import definiert.

```
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"
}
```

Wird verwendet get-variant-import-job, um den Status zu überprüfen.

```
aws omics get-variant-import-job --job-id 08279950-a9e3-4cc3-9a3c-a574f9c9e229
```

Sie erhalten eine JSON-Antwort, die den Status Ihres Importauftrags anzeigt. VEP-Anmerkungen in der VCF werden nach Informationen geparst, die in der INFO-Spalte als Paar gespeichert sind. ID/Value Die Standard-ID für die INFO-Spalte der Anmerkungen zu Ensembl Variant Effect Predictor ist CSQ, aber Sie können den `--annotation-fields` Parameter verwenden, um einen benutzerdefinierten Wert anzugeben, der in der INFO-Spalte verwendet wird. Das Parsen wird derzeit für VEP-Anmerkungen unterstützt.

Für einen Shop, der vor dem 15. Mai 2023 erstellt wurde, oder für VCF-Dateien, die keine VEP-Anmerkung enthalten, enthält die Antwort keine Kommentarfelder.

```
{
    "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",
```

```
}
```

Die VEP-Anmerkungen, die Teil von VCF-Dateien sind, werden als vordefiniertes Schema mit der folgenden Struktur gespeichert. Das Feld extras kann verwendet werden, um alle zusätzlichen VEP-Felder zu speichern, die nicht im Standardschema enthalten sind.

```
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>
    >>
>
```

Die Analyse erfolgt nach bestem Wissen und Gewissen. Wenn der VEP-Eintrag nicht den VEP-Standardspezifikationen entspricht, wird er nicht analysiert und die Zeile im Array ist leer.

Bei einem neuen Variantenspeicher get-variant-import-jobwürde die Antwort für die Annotationsfelder enthalten, wie in der Abbildung gezeigt.

```
aws omics get-variant-import-job --job-id 08279950-a9e3-4cc3-9a3c-a574f9c9e229
```

Sie erhalten eine JSON-Antwort, die den Status Ihres Importauftrags anzeigt.

```
{
    "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"}
  }
}
```

Sie können list-variant-import-jobssie verwenden, um alle Importaufträge und deren Status zu sehen.

```
aws omics list-variant-import-jobs --ids 7a1c67e3-b7f9-434d-817b-9c571fd63bea
```

Die Antwort enthält Informationen wie folgt.

```
{
    "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"}
        }
    ]
  }
```

```
}
```

Bei Bedarf können Sie einen Importauftrag mit dem folgenden Befehl abbrechen.

```
aws omics cancel-variant-import-job
     --job-id edd7b8ce-xmpl-47e2-bc99-258cac95a508
```

HealthOmics Annotationsspeicher erstellen

> **Important**
>
> AWS HealthOmics Variantenspeicher und Annotationsspeicher stehen neuen Kunden nicht mehr zur Verfügung. Vorhandene Kunden können den Service weiterhin wie gewohnt verwenden. Weitere Informationen finden Sie unter AWS HealthOmics Änderung der Verfügbarkeit von Variantenspeicher und Annotationsspeicher.

Ein Annotationsspeicher ist ein Datenspeicher, der eine Annotationsdatenbank darstellt, z. B. eine Datenbank aus einer TSV-, VCF- oder GFF-Datei. Wenn dasselbe Referenzgenom angegeben wird, werden Annotationsspeicher während eines Imports demselben Koordinatensystem zugeordnet wie Variantenspeicher. In den folgenden Themen wird gezeigt, wie Sie die HealthOmics Konsole verwenden und AWS CLI Annotationsspeicher erstellen und verwalten.

Themen

- Erstellen eines Annotationsspeichers mithilfe der Konsole
- Einen Annotationsspeicher mithilfe der API erstellen

Erstellen eines Annotationsspeichers mithilfe der Konsole

Gehen Sie wie folgt vor, um Annotationsspeicher mit der HealthOmics Konsole zu erstellen.

Um einen Annotationsspeicher zu erstellen

1. Öffnen Sie die HealthOmics -Konsole.
2. Öffnen Sie bei Bedarf den linken Navigationsbereich (±1). Wählen Sie Annotation Stores.
3. Wählen Sie auf der Seite Annotationsspeicher die Option Annotationsspeicher erstellen aus.

4. Geben Sie auf der Seite Annotationsspeicher erstellen die folgenden Informationen ein

   - Name des Speichers für Anmerkungen — Ein eindeutiger Name für diesen Speicher.
   - Beschreibung (optional) — Eine Beschreibung dieses Referenzgenoms.
   - Datenformat und Schemadetails — Wählen Sie das Datendateiformat aus und laden Sie die Schemadefinition für diesen Speicher hoch.
   - Referenzgenom — Das Referenzgenom für diese Annotation.
   - Datenverschlüsselung — Wählen Sie aus, ob die Datenverschlüsselung von Ihnen AWS oder von Ihnen selbst verwaltet werden soll.
   - Tags (optional) — Stellen Sie bis zu 50 Tags für diesen Annotationsspeicher bereit.
5. Wählen Sie „Annotationsspeicher erstellen“.

Einen Annotationsspeicher mithilfe der API erstellen

Das folgende Beispiel zeigt, wie Sie einen Annotationsspeicher mit dem erstellen AWS CLI. Für alle AWS CLI API-Operationen müssen Sie das Format Ihrer Daten angeben.

```
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
```

Sie erhalten die folgende Antwort, um die Erstellung Ihres Annotationsspeichers zu bestätigen.

```
{
           "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"
       }
```

Verwenden Sie die get-annotation-storeAPI, um mehr über einen Annotationsspeicher zu erfahren.

```
aws omics get-annotation-store --name my_annotation_store
```

Sie erhalten die folgende Antwort.

```
{
          "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
      }
```

Verwenden Sie den list-annotation-storesAPI-Vorgang, um alle mit einem Konto verknüpften Annotationsspeicher anzuzeigen.

```
aws omics list-annotation-stores
```

Sie erhalten eine Antwort, in der alle Annotationsspeicher zusammen mit ihrem IDs Status und anderen Details aufgeführt sind, wie in der folgenden Beispielantwort dargestellt.

```
{
           "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"
            }
        ]
    }
```

Sie können Antworten auch nach Status oder anderen Kriterien filtern.

Importjobs für HealthOmics Annotationsspeicher erstellen

> **Important**
>
> AWS HealthOmics Variantenspeicher und Annotationsspeicher stehen neuen Kunden nicht mehr zur Verfügung. Vorhandene Kunden können den Service weiterhin wie gewohnt verwenden. Weitere Informationen finden Sie unter AWS HealthOmics Änderung der Verfügbarkeit von Variantenspeicher und Annotationsspeicher.

Themen

Einen Job zum Importieren von Anmerkungen mithilfe der API erstellen

Das folgende Beispiel zeigt, wie Sie den verwenden AWS CLI , um einen Importjob für Anmerkungen zu starten.

```
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"}'
```

Annotationsspeicher, die vor dem 15. Mai 2023 erstellt wurden, geben eine Fehlermeldung zurück, wenn die Annotationsfelder enthalten sind. Sie geben keine Ausgabe für API-Operationen zurück, die mit Importaufträgen für Annotationsspeicher verbunden sind.

Sie können dann den get-annotation-import-jobAPI-Vorgang und den `job ID` Parameter verwenden, um weitere Informationen zum Importjob für Anmerkungen zu erhalten.

```
aws omics get-annotation-import-job --job-id 9e4198fb-fa85-446c-9301-9b823a1a8ba8
```

Sie erhalten die folgende Antwort, einschließlich der Annotationsfelder.

```
{
          "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"}
       }
```

Um alle Importaufträge für den Annotationsspeicher anzuzeigen, verwenden Sie list-annotation-import-jobs.

```
aws omics list-annotation-import-jobs --ids 9e4198fb-fa85-446c-9301-9b823a1a8ba8
```

Die Antwort enthält die Details und den Status Ihrer Importaufträge für den Annotationsspeicher.

```
{
          "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"}
        }
        ]
    }
```

Zusätzliche Parameter für die Formate TSV und VCF

Für die Formate TSV und VCF gibt es zusätzliche Parameter, die die API darüber informieren, wie Ihre Eingabe analysiert werden soll.

> **Important**
>
> CSV-Annotationsdaten, die mit Abfrage-Engines exportiert werden, geben direkt Informationen aus dem Datensatzimport zurück. Wenn die importierten Daten Formeln oder Befehle enthalten, wird die Datei möglicherweise einer CSV-Injektion unterzogen. Daher können mit Abfrage-Engines exportierte Dateien zu Sicherheitswarnungen führen. Um böswillige Aktivitäten zu vermeiden, deaktivieren Sie Links und Makros beim Lesen von Exportdateien.

Der TSV-Parser führt auch grundlegende bioinformatische Operationen durch, wie die Normalisierung der linken Seite und die Standardisierung von genomischen Koordinaten, die in der folgenden Tabelle aufgeführt sind.

Typ des Formats	Description
Generisch	Generische Textdatei. Keine genomischen Informationen.
`CHR_POS`	Startposition - 1, Endposition hinzufügen, die identisch `POS` ist mit.
`CHR_POS_REF_ALT`	Enthält Informationen zu den Allelen Contig, 1-Basen-Position, Ref und Alt.

Typ des Formats	Description
CHR_START_END_REF_ALT_ONE_BASE	Enthält Contig-, Start-, End-, Ref- und Alt-Allel informationen. Die Koordinaten basieren auf Eins.
CHR_START_END_ZERO_BASE	Enthält Zähl-, Start- und Endpositionen. Die Koordinaten basieren auf 0.
CHR_START_END_ONE_BASE	Enthält Längs-, Start- und Endpositionen. Die Koordinaten basieren auf 1.
CHR_START_END_REF_ALT_ZERO_BASE	Enthält Contig-, Start-, End-, Ref- und Alt-Allel informationen. Die Koordinaten basieren auf 0.

Eine Anforderung für den TSV-Import eines Annotationsspeichers sieht wie das folgende Beispiel aus.

```
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"
        }
    }
}'
```

Annotationsspeicher im TSV-Format erstellen

Im folgenden Beispiel wird mithilfe einer tabulatorbeschränkten Datei, die eine Kopfzeile, Zeilen und Kommentare enthält, ein Annotationsspeicher erstellt. Die Koordinaten lautenCHR_START_END_ONE_BASED, und sie enthält die HG19 Genkarte aus der Synopsis of the Human Gene Map der 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}]}'
```

Sie können Dateien mit oder ohne Header importieren. Um dies in einer CLI-Anforderung anzugeben, verwenden Sie`header=false`, wie im folgenden Beispiel für einen Importjob gezeigt.

```
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="#"}}'
```

Im folgenden Beispiel wird ein Speicher für Anmerkungen für eine BED-Datei erstellt. Eine Bettdatei ist eine einfache tabulatorgetrennte Datei. In diesem Beispiel lauten die Spalten Chromosom, Start, Ende und Regionsname. Die Koordinaten basieren auf Null, und die Daten haben keinen Header.

```
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}]}'
```

Anschließend können Sie die Bed-Datei mit dem folgenden CLI-Befehl in den Annotationsspeicher importieren.

```
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="#"}}'
```

Im folgenden Beispiel wird ein Annotationsspeicher für eine tabulatorgetrennte Datei erstellt, die die ersten Spalten einer VCF-Datei enthält, gefolgt von Spalten mit Annotationsinformationen. Es enthält Genompositionen mit Informationen zu den Chromosomen-, Start-, Referenz- und alternativen Allelen sowie eine Überschrift.

```
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}]}'
```

Anschließend würden Sie die Datei mit dem folgenden CLI-Befehl in den Annotationsspeicher importieren.

```
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="#"}}'
```

Das folgende Beispiel zeigt, wie ein Kunde einen Annotationsspeicher für eine mim2gene-Datei erstellen kann. Eine mim2Gene-Datei stellt die Verbindungen zwischen den Genen in OMIM und einem anderen Genidentifikator bereit. Sie ist tabulatorgetrennt und enthält Kommentare.

```
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}]}'
```

Anschließend können Sie Daten wie folgt in Ihren Shop importieren.

```
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="#"}}'
```

Importaufträge im VCF-Format werden gestartet

Für VCF-Dateien gibt es zwei zusätzliche Eingaben, `ignoreQualField` und, die diese Parameter ignorieren oder einschließen`ignoreFilterField`, wie in der Abbildung gezeigt.

```
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
    }
   }'
```

Sie können einen Import eines Annotationsspeichers auch abbrechen, wie hier gezeigt. Wenn die Stornierung erfolgreich ist, erhalten Sie keine Antwort auf diesen AWS CLI Anruf. Wenn die Importauftrags-ID jedoch nicht gefunden wird oder der Importauftrag abgeschlossen ist, erhalten Sie eine Fehlermeldung.

```
aws omics cancel-annotation-import-job --job-id edd7b8ce-xmpl-47e2-bc99-258cac95a508
```

> **Note**
>
> Ihre Metadaten importieren den Jobverlauf für get-annotation-import-jobget-variant-import-job, list-annotation-import-jobs, und list-variant-import-jobswerden nach zwei Jahren automatisch gelöscht. Die importierten Varianten- und Annotationsdaten werden nicht automatisch gelöscht und verbleiben in Ihren Datenspeichern.

HealthOmics Annotation Store-Versionen erstellen

> **Important**
>
> AWS HealthOmics Variantenspeicher und Annotationsspeicher stehen neuen Kunden nicht mehr zur Verfügung. Vorhandene Kunden können den Service weiterhin wie gewohnt verwenden. Weitere Informationen finden Sie unter AWS HealthOmics Änderung der Verfügbarkeit von Variantenspeicher und Annotationsspeicher.

Sie können neue Versionen von Annotationsspeichern erstellen, um verschiedene Versionen Ihrer Annotationsdatenbanken zu sammeln. Dies hilft Ihnen bei der Organisation Ihrer Annotationsdaten, die regelmäßig aktualisiert werden.

Verwenden Sie die create-annotation-store-versionAPI, um eine neue Version eines vorhandenen Annotationsspeichers zu erstellen, wie im folgenden Beispiel gezeigt.

```
aws omics create-annotation-store-version \
     --name my_annotation_store \
     --version-name my_version
```

Sie erhalten die folgende Antwort mit der Versions-ID des Annotationsspeichers, die bestätigt, dass eine neue Version Ihrer Annotation erstellt wurde.

```
{
     "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"
}
```

Um die Beschreibung einer Annotation Store-Version zu aktualisieren, können Sie damit Aktualisierungen update-annotation-store-versionzu einer Annotation Store-Version hinzufügen.

```
aws omics update-annotation-store-version \
    --name my_annotation_store \
    --version-name my_version \
    --description "New Description"
```

Sie erhalten die folgende Antwort, in der bestätigt wird, dass die Annotation Store-Version aktualisiert wurde.

```
{
     "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"
}
```

Um die Details einer Annotation Store-Version anzuzeigen, verwenden Sie get-annotation-store-version.

```
aws omics get-annotation-store-version --name my_annotation_store --version-name
 my_version
```

Sie erhalten eine Antwort mit dem Versionsnamen, dem Status und anderen Details.

```
{
     "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
    }
```

Um alle Versionen eines Annotationsspeichers anzuzeigen, können Sie list-annotation-store-versions, wie im folgenden Beispiel gezeigt, Folgendes verwenden.

```
aws omics list-annotation-store-versions --name my_annotation_store
```

Sie erhalten eine Antwort mit den folgenden Informationen

```
{
  "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
    }
}
```

Wenn Sie eine Version des Annotationsspeichers nicht mehr benötigen, können Sie delete-annotation-store-versionssie verwenden, um eine Version des Annotationsspeichers zu löschen, wie im folgenden Beispiel gezeigt.

```
aws omics delete-annotation-store-versions --name my_annotation_store --versions
 my_version
```

Wenn die Store-Version ohne Fehler gelöscht wird, erhalten Sie die folgende Antwort.

```
{
  "errors": []
}
```

Wenn Fehler auftreten, erhalten Sie eine Antwort mit den Einzelheiten der Fehler, wie in der Abbildung dargestellt.

```
{
  "errors": [
    {
      "versionName": "my_version",
      "message": "Version with versionName: my_version was not found."
    }
  ]
}
```

Wenn Sie versuchen, eine Annotation Store-Version zu löschen, die über einen aktiven Importauftrag verfügt, erhalten Sie eine Antwort mit einem Fehler, wie hier gezeigt.

```
{
  "errors": [
    {
      "versionName": "my_version",
      "message": "version has an inflight import running"
    }
  ]
}
```

In diesem Fall können Sie das Löschen der Annotation Store-Version erzwingen, wie im folgenden Beispiel gezeigt.

```
aws omics delete-annotation-store-versions --name my_annotation_store --versions
 my_version --force
```

HealthOmics Analytics-Stores löschen

> **Important**
>
> AWS HealthOmics Variantenspeicher und Annotationsspeicher stehen neuen Kunden nicht mehr zur Verfügung. Vorhandene Kunden können den Service weiterhin wie gewohnt verwenden. Weitere Informationen finden Sie unter AWS HealthOmics Änderung der Verfügbarkeit von Variantenspeicher und Annotationsspeicher.

Wenn Sie einen Varianten- oder Annotationsspeicher löschen, löscht das System auch alle importierten Daten in diesem Speicher und alle zugehörigen Tags.

Das folgende Beispiel zeigt, wie Sie einen Variantenspeicher mit dem AWS CLI löschen. Wenn die Aktion erfolgreich ist, wechselt der Status des Variantenspeichers zu`DELETING`.

```
aws omics delete-variant-store --id <variant-store-id>
```

Das folgende Beispiel zeigt, wie ein Annotationsspeicher gelöscht wird. Wenn die Aktion erfolgreich ist, wechselt der Status des Annotationsspeichers zu`DELETING`. Annotationsspeicher können nicht gelöscht werden, wenn mehr als eine Version vorhanden ist.

```
aws omics delete-annotation-store --id <annotation-store-id>
```

Abfragen von HealthOmics Analysedaten

> **Important**
>
> AWS HealthOmics Variantenspeicher und Annotationsspeicher stehen neuen Kunden nicht mehr zur Verfügung. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter AWS HealthOmics Änderung der Verfügbarkeit von Variantenspeicher und Annotationsspeicher.

Sie können Abfragen für Ihre Variantenshops mit AWS Lake Formation Amazon Athena oder Amazon EMR durchführen. Bevor Sie Abfragen ausführen, müssen Sie die (in den folgenden Abschnitten beschriebenen) Einrichtungsverfahren für Lake Formation und Amazon Athena abschließen.

Informationen zu Amazon EMR finden Sie unter Tutorial: Erste Schritte mit Amazon EMR

HealthOmics Partitionieren Sie bei Variantenspeichern, die nach dem 26. September 2024 erstellt wurden, den Speicher nach der Proben-ID. Diese Partitionierung bedeutet, dass die Proben-ID HealthOmics verwendet wird, um die Speicherung der Varianteninformationen zu optimieren. Abfragen, die Probeninformationen als Filter verwenden, geben schneller Ergebnisse zurück, da die Abfrage weniger Daten scannt.

HealthOmics verwendet Beispiel IDs als Partitionsdateinamen. Bevor Sie Daten aufnehmen, überprüfen Sie, ob die Proben-ID PHI-Daten enthält. Ist dies der Fall, ändern Sie die Proben-ID, bevor Sie die Daten aufnehmen. Weitere Informationen darüber, welche Inhalte in die Stichprobe aufgenommen werden sollen und welche nicht IDs, finden Sie in den Anleitungen auf der Webseite zur AWS HIPAA-Konformität.

Themen

Konfiguration von Lake Formation für die Verwendung HealthOmics

> **Important**
>
> AWS HealthOmics Variantenspeicher und Annotationsspeicher stehen Neukunden nicht mehr zur Verfügung. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter AWS HealthOmics Änderung der Verfügbarkeit von Variantenspeicher und Annotationsspeicher.

Bevor Sie Lake Formation zur Verwaltung von HealthOmics Datenspeichern verwenden, führen Sie die folgenden Lake Formation Formation-Konfigurationsverfahren durch.

Themen

Lake Formation-Administratoren erstellen oder überprüfen

Bevor Sie in Lake Formation einen Data Lake erstellen können, definieren Sie einen oder mehrere Administratoren.

Administratoren sind Benutzer und Rollen mit Berechtigungen zum Erstellen von Ressourcenlinks. Sie richten Data Lake-Administratoren pro Konto und Region ein.

Erstellen Sie einen Admin-Benutzer in der Lake Formation Formation-Konsole

1. Öffnen Sie die AWS Lake Formation Formation-Konsole: Lake Formation Formation-Konsole
2. Wenn auf der Konsole das Fenster Willkommen bei Lake Formation angezeigt wird, wählen Sie Get started.

   Lake Formation fügt Sie der Data Lake-Administratortabelle hinzu.
3. Andernfalls wählen Sie im linken Menü die Option Administrative Rollen und Aufgaben aus.
4. Fügen Sie nach Bedarf weitere Administratoren hinzu.

Erstellen von Ressourcenlinks mit der Lake Formation Formation-Konsole

Um eine gemeinsam genutzte Ressource zu erstellen, die Benutzer abfragen können, müssen die Standardzugriffskontrollen deaktiviert sein. Weitere Informationen zum Deaktivieren der Standardzugriffskontrollen finden Sie unter Ändern der Standardsicherheitseinstellungen für Ihren Data Lake in der Lake Formation Formation-Dokumentation. Sie können Ressourcenlinks einzeln oder als Gruppe erstellen, sodass Sie auf Daten in Amazon Athena oder anderen AWS Diensten (wie Amazon EMR) zugreifen können.

Ressourcenlinks in der AWS Lake Formation Formation-Konsole erstellen und mit HealthOmics Analytics-Benutzern teilen

1. Öffnen Sie die AWS Lake Formation Formation-Konsole: Lake Formation Formation-Konsole
2. Wählen Sie in der primären Navigationsleiste Datenbanken aus.
3. Wählen Sie in der Tabelle Datenbanken die gewünschte Datenbank aus.
4. Wählen Sie im Menü Erstellen die Option Ressourcenlink aus.
5. Geben Sie einen Namen für den Ressourcenlink ein. Wenn Sie von Athena aus auf die Datenbank zugreifen möchten, geben Sie einen Namen ein, der nur Kleinbuchstaben (bis zu 256 Zeichen) verwendet.
6. Wählen Sie Erstellen aus.
7. Der neue Ressourcenlink ist jetzt unter Datenbanken aufgeführt.

Gewähren Sie mithilfe der Lake Formation Formation-Konsole Zugriff auf die gemeinsam genutzte Ressource

Ein Lake Formation Formation-Datenbankadministrator kann mithilfe des folgenden Verfahrens Zugriff auf die gemeinsam genutzte Ressource gewähren.

1. Öffnen Sie die AWS Lake Formation Formation-Konsole: https://console.aws.amazon.com/lakeformation/
2. Wählen Sie in der primären Navigationsleiste Datenbanken aus.
3. Wählen Sie auf der Seite Datenbanken den Ressourcenlink aus, den Sie zuvor erstellt haben.
4. Wählen Sie im Menü Aktionen die Option Auf Ziel gewähren aus.
5. Wählen Sie auf der Seite Datenberechtigungen gewähren unter Principals die Option IAM-Benutzer oder -Rollen aus.

6. Suchen Sie im Dropdownmenü IAM-Benutzer oder -Rollen nach dem Benutzer, dem Sie Zugriff gewähren möchten.
7. Wählen Sie als Nächstes unter der Karte LF-Tags oder Katalogressourcen die Option Benannte Datenkatalogressourcen aus.
8. Wählen Sie im Dropdownmenü „Tabellen optional“ die Option Alle Tabellen oder die Tabelle aus, die Sie zuvor erstellt haben.
9. Wählen Sie auf der Karte Tabellenberechtigungen unter Tabellenberechtigungen die Optionen Beschreiben und Auswählen aus.
10. Wählen Sie als Nächstes Grant aus.

Um die Lake Formation Formation-Berechtigungen anzuzeigen, wählen Sie im primären Navigationsbereich Data Lake-Berechtigungen aus. Die Tabelle zeigt die verfügbaren Datenbanken und Ressourcenlinks.

Berechtigungen für gemeinsam genutzte AWS RAM Ressourcen konfigurieren

Sehen Sie sich in der AWS Lake Formation Formation-Konsole die Berechtigungen an, indem Sie in der primären Navigationsleiste Data Lake-Berechtigungen auswählen. Auf der Seite mit den Datenberechtigungen können Sie unter RAM Resource Share eine Tabelle mit den Ressourcentypen und Datenbanken einsehen, **ARN** die sich auf eine gemeinsam genutzte Ressource bezieht. Wenn Sie eine Ressourcenfreigabe AWS Resource Access Manager (AWS RAM) akzeptieren müssen, werden Sie in der Konsole AWS Lake Formation darüber informiert.

HealthOmics kann die AWS RAM Ressourcenfreigaben während der Shop-Erstellung implizit akzeptieren. Um die AWS RAM Ressourcenfreigabe zu akzeptieren, muss der IAM-Benutzer oder die IAM-Rolle, die die `CreateVariantStore` oder `CreateAnnotationStore` API-Operationen aufruft, die folgenden Aktionen zulassen:

- `ram:GetResourceShareInvitations`- Diese Aktion ermöglicht es HealthOmics , die Einladungen zu finden.
- `ram:AcceptResourceShareInvitation`- Diese Aktion ermöglicht es HealthOmics , die Einladung mithilfe eines FAS-Tokens anzunehmen.

Ohne diese Berechtigungen wird bei der Erstellung des Shops ein Autorisierungsfehler angezeigt.

Hier ist ein Beispiel für eine Richtlinie, die diese Aktionen beinhaltet. Fügen Sie diese Richtlinie dem IAM-Benutzer oder der IAM-Rolle hinzu, die die AWS RAM Ressourcenfreigabe akzeptiert.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "omics:*",
        "ram:AcceptResourceShareInvitation",
        "ram:GetResourceShareInvitations"
      ],
      "Resource": "*"
    }
  ]
}
```

Konfiguration von Athena für Abfragen

> **Important**
>
> AWS HealthOmics Variantenspeicher und Annotationsspeicher stehen neuen Kunden nicht mehr zur Verfügung. Vorhandene Kunden können den Service weiterhin wie gewohnt verwenden. Weitere Informationen finden Sie unter AWS HealthOmics Änderung der Verfügbarkeit von Variantenspeicher und Annotationsspeicher.

Sie können Athena verwenden, um Varianten und Anmerkungen abzufragen. Bevor Sie Abfragen ausführen, führen Sie die folgenden Einrichtungsaufgaben aus:

Themen

- Konfigurieren Sie einen Speicherort für Abfrageergebnisse mit der Athena-Konsole
- Konfiguration einer Arbeitsgruppe mit Athena Engine v3

Konfigurieren Sie einen Speicherort für Abfrageergebnisse mit der Athena-Konsole

Gehen Sie folgendermaßen vor, um einen Speicherort für Abfrageergebnisse zu konfigurieren.

1. Öffnen Sie die Athena-Konsole: Athena-Konsole
2. Wählen Sie in der Hauptnavigationsleiste den Abfrage-Editor aus.
3. Wählen Sie im Abfrage-Editor die Registerkarte Einstellungen und anschließend Verwalten aus.
4. Geben Sie ein S3-Präfix eines Speicherorts ein, um das Abfrageergebnis zu speichern.

Konfiguration einer Arbeitsgruppe mit Athena Engine v3

Gehen Sie folgendermaßen vor, um eine Arbeitsgruppe zu konfigurieren.

1. Öffnen Sie die Athena-Konsole: Athena-Konsole
2. Wählen Sie in der Hauptnavigationsleiste Arbeitsgruppen und dann Arbeitsgruppe erstellen aus.
3. Geben Sie einen Namen für die Arbeitsgruppe ein.
4. Wählen Sie Athena SQL als Engine-Typ aus.
5. Wählen Sie unter Abfrage-Engine aktualisieren die Option Manuell aus.
6. Wählen Sie unter Query Version Engine die Option Athena Version 3 aus.
7. Wählen Sie Create workgroup (Arbeitsgruppe erstellen) aus.

Abfragen für HealthOmics Variantenspeicher ausführen

> **Important**
>
> AWS HealthOmics Variantenspeicher und Annotationsspeicher stehen neuen Kunden nicht mehr zur Verfügung. Vorhandene Kunden können den Service weiterhin wie gewohnt verwenden. Weitere Informationen finden Sie unter AWS HealthOmics Änderung der Verfügbarkeit von Variantenspeicher und Annotationsspeicher.

Sie können mit Amazon Athena Abfragen in Ihrem Variantenshop durchführen. Beachten Sie, dass genomische Koordinaten in Varianten- und Annotationsspeichern als auf Null basierende, halbgeschlossene, halboffene Intervalle dargestellt werden.

Führen Sie eine einfache Abfrage mit der Athena-Konsole aus

Das folgende Beispiel zeigt, wie eine einfache Abfrage ausgeführt wird.

1. Öffnen Sie den Athena Query Editor: Athena Query editor

2. Wählen Sie unter Arbeitsgruppe die Arbeitsgruppe aus, die Sie während der Installation erstellt haben.
3. Stellen Sie sicher, dass die Datenquelle AwsDataCatalog
4. Wählen Sie für Datenbank den Datenbankressourcen-Link aus, den Sie während des Lake Formation-Setups erstellt haben.
5. Kopieren Sie die folgende Abfrage in den Abfrage-Editor auf der Registerkarte Abfrage 1:

```
SELECT * from omicsvariants limit 10
```

6. Wählen Sie dann Ausführen aus, um die Abfrage auszuführen. Die Konsole füllt die Ergebnistabelle mit den ersten 10 Zeilen der omicsvariants Tabelle.

Führen Sie eine komplexe Abfrage mit der Athena-Konsole aus

Das folgende Beispiel zeigt, wie eine komplexe Abfrage ausgeführt wird. Um diese Abfrage auszuführen, importieren Sie `ClinVar` sie in den Annotationsspeicher.

Führen Sie eine komplexe Abfrage aus

1. Öffnen Sie den Athena Query Editor: Athena Query editor
2. Wählen Sie unter Arbeitsgruppe die Arbeitsgruppe aus, die Sie während der Installation erstellt haben.
3. Stellen Sie sicher, dass die Datenquelle AwsDataCatalog
4. Wählen Sie für Datenbank den Datenbankressourcen-Link aus, den Sie während des Lake Formation-Setups erstellt haben.
5. Wählen Sie + oben rechts, um eine neue Abfrageregisterkarte mit dem Namen Query 2 zu erstellen.
6. Kopieren Sie die folgende Abfrage in den Abfrage-Editor auf der Registerkarte Abfrage 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. Wählen Sie Ausführen, um mit der Ausführung der Abfrage zu beginnen.

HealthOmics Analytics-Shops teilen

> **Important**
>
> AWS HealthOmics Variantenspeicher und Annotationsspeicher stehen Neukunden nicht mehr zur Verfügung. Vorhandene Kunden können den Service weiterhin wie gewohnt verwenden. Weitere Informationen finden Sie unter AWS HealthOmics Änderung der Verfügbarkeit von Variantenspeicher und Annotationsspeicher.

Als Besitzer eines Variantenspeichers oder eines Annotationsspeichers können Sie den Shop mit anderen AWS-Konten teilen. Der Eigentümer kann den Zugriff auf die gemeinsam genutzte Ressource widerrufen, indem er die gemeinsame Nutzung löscht.

Als Abonnent eines gemeinsam genutzten Shops akzeptieren Sie zunächst die Freigabe. Anschließend können Sie Workflows definieren, die den gemeinsamen Speicher verwenden. Die Daten werden sowohl in Lake Formation als auch in AWS Glue Form einer Tabelle angezeigt.

Wenn Sie keinen Zugriff mehr auf den Shop benötigen, löschen Sie den Share.

Weitere Informationen Kontoübergreifende gemeinsame Nutzung von Ressourcen in AWS HealthOmics zur gemeinsamen Nutzung von Ressourcen finden Sie unter.

Eine Shop-Freigabe erstellen

Verwenden Sie den API-Vorgang create-share, um ein Store-Share zu erstellen. Der Hauptabonnent ist AWS-Konto der Benutzer, der die Aktie abonnieren wird. Im folgenden Beispiel wird eine Aktie für einen Variantenspeicher erstellt. Um einen Shop mit mehr als einem Konto zu teilen, erstellen Sie mehrere Shares desselben Shops.

```
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"
```

Wenn die Erstellung erfolgreich ist, erhältst du eine Antwort mit der Share-ID und dem Status.

```
{
       "shareId": "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a",
       "name": "my_Share-123",
       "status": "PENDING"
  }
```

Die Freigabe bleibt so lange im Status „Ausstehend“, bis der Abonnent sie mithilfe des API-Vorgangs „Accept-Share“ akzeptiert.

Kontoübergreifende gemeinsame Nutzung von Ressourcen in AWS HealthOmics

Verwenden Sie die kontoübergreifende gemeinsame Nutzung, um Ressourcen mit anderen zu teilen, ohne Kopien zu erstellen oder IAM-Ressourcenrichtlinien zu ändern. Die folgenden Ressourcen unterstützen die kontenübergreifende gemeinsame Nutzung:

- HealthOmics Variantenspeicher
- HealthOmics Speicher für Anmerkungen
- Private Workflows

Das Teilen einer Ressource umfasst die folgenden Schritte:

1. Der Ressourcenbesitzer erstellt eine Freigabe und gibt den ARN der Ressource und den AWS-Konto des beabsichtigten Abonnenten an. Die Ressourcenfreigabe verbleibt im Status „Ausstehend“, bis der Abonnent die Freigabe akzeptiert.
2. Der Abonnent akzeptiert die gemeinsame Nutzung der Ressource, um Zugriff auf die Ressource zu erhalten. Die gemeinsame Nutzung der Ressource geht in den aktivierenden Status über.
3. Der HealthOmics Dienst bietet dem Abonnentenkonto Zugriff auf die Ressource.
4. Der Eigentümer der Ressource kann die Freigabe löschen, oder der Abonnent kann seinen Zugriff auf die Freigabe widerrufen. Der Abonnent kann den Share oder die zugehörige Ressource nicht löschen.

Themen

Einen Share erstellen

Sie können den API-Vorgang create-share verwenden, um eine gemeinsame Nutzung zu erstellen. Der Hauptabonnent ist AWS-Konto der Benutzer, der die gemeinsam genutzte Ressource abonnieren wird. Im folgenden Beispiel wird ein Share für einen Variantenspeicher erstellt.

```
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"
```

Wenn die Erstellung erfolgreich ist, erhalten Sie eine Antwort mit der Share-ID und dem Status.

```
{
"shareId": "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a",
     "name": "my_Share-123",
     "status": "PENDING"
}
```

Die Freigabe verbleibt im Status „Ausstehend“, bis der Abonnent sie mithilfe des accept-share API-Vorgangs akzeptiert.

```
 aws omics accept-share \
    --share-id "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a"
```

Nachdem der Abonnent die Freigabe akzeptiert hat, wechselt die Freigabe in den Status Aktiv.

```
{
"status": "ACTIVATING"
}
```

Ruft Informationen über eine Aktie ab

Verwenden Sie den API-Vorgang get-share, um Informationen über die Freigabe abzurufen.

```
aws omics get-share --share-id "495c21bedc889d07d0ab69d710a6841e-
dd75ab7a1a9c384fa848b5bd8e5a7e0a"
```

Die API-Antwort enthält Metadateninformationen zur Freigabe.

```
{
  "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"
    }
}
```

Sehen Sie sich die Aktien an, die Sie besitzen

Verwenden Sie die List-Shares-API, um Informationen zu den einzelnen Aktien abzurufen, die Sie besitzen.

```
aws omics list-shares  --resource-owner SELF
```

Die API-Antwort enthält die Metadaten für jede Aktie, die Sie besitzen.

Akzeptierte Shares von anderen Konten anzeigen

Verwenden Sie die List-Shares-API, um alle Shares anzuzeigen, die Sie von anderen Konten akzeptiert haben.

```
aws omics list-shares  --resource-owner OTHER
```

Die API-Antwort enthält die Metadaten für jeden Share, den Sie akzeptiert haben.

Löschen Sie eine Aktie

Verwenden Sie die Delete-Share-API, um eine Freigabe zu löschen, wenn Sie sie nicht mehr benötigen.

```
aws omics delete-share \
    --share-id "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a"
```

Ressourcen taggen in HealthOmics

Themen

Wichtiger Hinweis

HealthOmics schützt Kundendaten gemäß den Richtlinien des AWS-Modells für gemeinsame Verantwortung. Das bedeutet, dass alle Kundendaten sowohl bei der Übertragung als auch bei der Speicherung verschlüsselt werden. Allerdings sind nicht alle vom Kunden eingegebenen Namen für Ressourcen wie Datenspeicher oder auftragsbasierte Vorgänge verschlüsselt. Sie sollten niemals persönlich identifizierbare Informationen oder geschützte Gesundheitsinformationen enthalten. Weitere Informationen finden Sie unter Sicherheit in AWS HealthOmics.

Ressourcen taggen HealthOmics

Sie können Ihren AWS-Ressourcen mithilfe von Tags Metadaten zuweisen. Jedes Tag ist ein Label, das aus einem benutzerdefinierten Schlüssel und Wert besteht. Mit Tags können Sie Ressourcen verwalten, identifizieren, organisieren, suchen und filtern.

In diesem Thema werden häufig verwendete Tagging-Kategorien und -Strategien beschrieben, mit denen Sie eine konsistente und effektive Tagging-Strategie implementieren können. In den folgenden Abschnitten werden Grundkenntnisse zu AWS-Ressourcen, Tagging, detaillierter Abrechnung und AWS Identity and Access Management vorausgesetzt.

Jedes -Tag besteht aus zwei Teilen:

- Ein Tag-Schlüssel (zum Beispiel CostCenter „Umgebung“ oder „Projekt“). Bei Tag-Schlüsseln wird zwischen Groß- und Kleinschreibung unterschieden.

- Ein Tag-Wert (zum Beispiel 111122223333 oder Production). Wie bei Tag-Schlüsseln wird auch bei Tag-Werten zwischen Groß- und Kleinschreibung unterschieden.

Sie können Tags verwenden, um Ressourcen nach Zweck, Eigentümer, Umgebung oder anderen Kriterien zu kategorisieren. Weitere Informationen finden Sie unter AWS Tagging-Strategien.

Sie können Tags für eine Ressource über die Servicekonsole, die Service-API oder die AWS CLI hinzufügen, ändern oder entfernen.

Um das Tagging zu aktivieren, stellen Sie sicher, dass TagResources es autorisiert ist. Sie können die Autorisierung durchführen, TagResources indem Sie eine IAM-Richtlinie wie im folgenden Beispiel anhängen.

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": "*"
        }
    ]
}
```

Best Practices

Beachten Sie bei der Erstellung einer Tagging-Strategie für AWS-Ressourcen die folgenden bewährten Methoden:

- Speichern Sie keine personenbezogenen Daten (PII), geschützte Gesundheitsinformationen (PHI) oder andere sensible Informationen in Tags.
- Verwenden Sie für Tags ein standardisiertes Format, bei dem die Groß-/Kleinschreibung beachtet wird, und wenden Sie es konsistent für alle Ressourcentypen an.
- Verwenden Sie Tag-Richtlinien, die mehrere Zwecke unterstützen, wie die Verwaltung der Ressourcenzugriffskontrolle, Kostenverfolgung, Automatisierung und Organisation.
- Verwenden Sie automatisierte Tools, um Ressourcen-Tags zu verwalten. AWS-Ressourcengruppen und die Resource Groups Tagging API ermöglichen die programmatische Steuerung von Tags und ermöglichen so die automatische Verwaltung, Suche und Filterung von Tags und Ressourcen.
- Tagging ist effektiver, wenn Sie mehr Tags verwenden.
- Tags können bearbeitet oder geändert werden, wenn sich die Benutzeranforderungen ändern. Um Zugriffskontroll-Tags zu aktualisieren, müssen Sie jedoch auch die Richtlinien aktualisieren, die auf diese Tags verweisen, um den Zugriff auf Ihre Ressourcen zu kontrollieren.

Anforderungen zum Markieren

Für Tags gelten zwei Anforderungen:

- Schlüsseln darf nicht das Präfix aws: vorangestellt werden.
- Schlüssel müssen in einem Tag-Satz eindeutig sein.
- Schlüssel müssen zwischen 1 und 128 Zeichen lang sein.
- Ein Wert muss zwischen 0 und 256 Zeichen haben.
- Werte müssen nicht pro Tagsatz eindeutig sein.

- Zulässige Zeichen für Schlüssel und Werte sind Unicode-Buchstaben, Ziffern, Leerzeichen sowie die folgenden Sonderzeichen: _ . : / = + - @.
- Bei Schlüsseln und Werten wird die Groß-/Kleinschreibung berücksichtigt.

Sequenzspeicher, gelesene Satz-Tags

Bei Sequenzspeichern befinden sich die auf dem Lesesatz erstellten Tags auf der Ressourcenebene des Lesesatzes. Lesesätze enthalten auch Objekte unter ihnen, auf die mit S3 zugegriffen, gesucht und eingeschränkt werden kann APIs. Standardmäßig werden die Proben-ID (omics:SampleID) und die Betreff-ID (omics:SubjectID) dem Objekt hinzugefügt.

Darüber hinaus können bis zu fünf Tags zwischen dem Lesesatz und den Objekten darunter synchronisiert werden. Bei der Konfiguration, für die Tags synchronisiert werden sollen, handelt es sich um eine Konfiguration auf Filialebene, die während der Erstellung oder Aktualisierung des Shops mithilfe des `propogatedSetLevelTags` Parameters festgelegt wurde.

Wenn sich bereits Daten im Store befinden, kann die Aktualisierung der Schlüssel einige Zeit in Anspruch nehmen. Während dieses Updates HealthOmics ändert sich der Status des Speichers aufUpdating. HealthOmics Setzt den Speicherstatus nach Abschluss aufActive. Während die Tags verbreitet werden, können Berechtigungen, die sich auf die Tags stützen, möglicherweise nicht durchgesetzt werden. Berechtigungen werden durchgesetzt, nachdem die Tag-Weitergabe abgeschlossen ist.

Wenn Tags für den Lesesatz gesetzt oder aktualisiert werden, entscheidet das System auf der Grundlage der Speicherkonfiguration, ob die Objekte für diesen Lesesatz aktualisiert werden sollen.

Hinzufügen eines Tags zu einer HealthOmics Ressource

Das Hinzufügen von Tags zu einer Ressource kann Ihnen helfen, Ihre AWS-Ressourcen zu identifizieren und zu organisieren und den Zugriff darauf zu verwalten. Zunächst fügen Sie einer Ressource ein oder mehrere Tags (Schlüssel-Wert-Paare) hinzu. Sie können bis zu 50 Tags pro Ressource verwenden. Es gibt auch Einschränkungen in Bezug auf die Zeichen, die Sie in den Schlüssel- und Wertfeldern verwenden können.

Nachdem Sie Tags hinzugefügt haben, können Sie IAM-Richtlinien erstellen, um den Zugriff auf die AWS Ressource auf der Grundlage dieser Tags zu verwalten. Sie können die HealthOmics Konsole oder die verwenden AWS CLI , um einer Ressource Tags hinzuzufügen. Das Hinzufügen von Tags zu einem Repository kann Auswirkungen auf den Zugriff auf dieses Repository haben. Bevor Sie

einem Datenspeicher ein Tag hinzufügen, überprüfen Sie alle IAM-Richtlinien, die möglicherweise Tags verwenden, um den Zugriff auf Ressourcen wie Datenspeicher zu steuern.

Service-Tags werden sowohl für einen Betreff als auch für eine Proben-ID für Sequenzspeicher automatisch generiert.

Gehen Sie wie folgt vor AWS CLI , um einer HealthOmics Ressource ein Tag hinzuzufügen. Um beispielsweise Tags zu einem Sequenzspeicher hinzuzufügen, während dieser erstellt wird, würden Sie den folgenden Befehl in der verwenden AWS CLI. Der Name des Sequenzspeichers lautet MySequenceStore, und die beiden hinzugefügten Tags mit Schlüsseln sind key1 und key2 mit Werten wie Wert1 bzw. Wert2 :

```
aws omics create-sequence-store --name "MySequenceStore" --tags key1=value1,key2=value2
```

Die Ausgabe listet die Tags nicht auf. Sie gibt die folgende Antwort zurück.

```
{
    "id": "6860403586",
    "referenceStoreId": "4889894479",
    "roleArn": "arn:aws:iam::555555555555:role/ImportTest",
    "status": "CREATED",
    "creationTime": "2022-07-21T01:19:07.194Z"
}
```

Um einer vorhandenen Ressource Tags hinzuzufügen, würden Sie den folgenden Beispielbefehl ausführen.

```
aws omics tag-resource --resource-arn arn:aws:omics:us-
west-2:555555555555:sequenceStore/2275234794 --tags key1=value1,key2=value2
```

Bei Erfolg gibt dieser Befehl keine Antwort zurück.

Tags für eine Ressource auflisten

Gehen Sie wie folgt vor AWS CLI , um mit dem eine Liste der AWS Tags für eine HealthOmics Ressource anzuzeigen. Wenn keine Tags hinzugefügt wurden, ist die zurückgegebene Liste leer.

Führen Sie den Befehl im Terminal oder in der list-tags-for-resource Befehlszeile aus, wie im folgenden Beispiel gezeigt.

```
aws omics list-tags-for-resource --resource-arn arn:aws:omics:us-
west-2:555555555555:sequenceStore/2275234794
```

Als Antwort erhalten Sie eine Liste von Tags im JSON-Format.

```
 {
    "tags": {
        "key1": "value1",
        "key2": "value2"
    }
}
```

Tags aus einem Datenspeicher entfernen

Sie können ein oder mehrere mit einer Ressource verknüpfte Tags entfernen. Durch das Entfernen eines Tags wird das Tag nicht aus anderen AWS-Ressourcen gelöscht, die mit diesem Tag verknüpft sind.

Führen Sie im Terminal oder in der Befehlszeile den Befehl untag-resource aus und geben Sie den Amazon-Ressourcennamen (ARN) der Ressource an, von der Sie Tags entfernen möchten, und den Tag-Schlüssel des Tags, das Sie entfernen möchten.

```
aws omics untag-resource --resource-arn arn:aws:omics:us-
west-2:555555555555:sequenceStore/2275234794 --tag-keys key1,key2
```

Bei Erfolg gibt dieser Befehl keine Antwort zurück. Um die der Ressource zugeordneten Tags zu überprüfen, führen Sie den Befehl list-tags-for-resource aus.

IAM-Berechtigungen für HealthOmics

Sie können AWS Identity and Access Management (IAM) verwenden, um den Zugriff auf die HealthOmics API und Ressourcen wie Shops und Workflows zu verwalten. Für Benutzer und Anwendungen in Ihrem Konto, die diese verwenden HealthOmics, verwalten Sie die Berechtigungen in einer Berechtigungsrichtlinie, die Sie auf IAM-Benutzer, -Gruppen oder -Rollen anwenden können.

Um die Berechtigungen für Benutzer und Anwendungen in Ihren Konten zu verwalten, verwenden Sie die Richtlinien, die HealthOmics Ihnen zur Verfügung stehen, oder schreiben Sie Ihre eigenen. Die HealthOmics Konsole verwendet mehrere Dienste, um Informationen über die Konfiguration und die Auslöser Ihrer Funktion abzurufen. Sie können die bereitgestellten Richtlinien unverändert oder als Ausgangspunkt für restriktivere Richtlinien verwenden.

HealthOmics verwendet IAM-Dienstrollen, um in Ihrem Namen auf andere Dienste zuzugreifen. Sie würden beispielsweise eine Servicerolle erstellen oder auswählen, wenn Sie einen Workflow ausführen, der Daten aus Amazon S3 liest. Für einige Funktionen müssen Sie auch Berechtigungen für Ressourcen in anderen Diensten konfigurieren. Überprüfen Sie diese Anforderungen, bevor Sie mit der Arbeit beginnen HealthOmics

Weitere Informationen zu IAM finden Sie unter Was ist IAM? im IAM-Benutzerhandbuch.

Themen

Identitätsbasierte IAM-Richtlinien für HealthOmics

Um Benutzern in Ihrem Konto Zugriff auf zu gewähren HealthOmics, verwenden Sie identitätsbasierte Richtlinien in AWS Identity and Access Management (IAM). Identitätsbasierte Richtlinien können direkt für IAM-Benutzer oder für IAM-Gruppen und -Rollen gelten, die einem Benutzer zugeordnet sind. Sie können auch Benutzern in einem anderen Konto die Berechtigung erteilen, eine Rolle in Ihrem Konto zu übernehmen und auf Ihre HealthOmics-Ressourcen zuzugreifen.

Um Benutzern die Erlaubnis zu erteilen, Aktionen an einer Workflow-Version auszuführen, müssen Sie den Workflow und die spezifische Workflow-Version zur Ressourcenliste hinzufügen.

Die folgende IAM-Richtlinie ermöglicht einem Benutzer den Zugriff auf alle HealthOmics API-Aktionen und die Übergabe von Servicerollen an HealthOmics.

Example Richtlinie für Benutzer:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "omics:*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "iam:PassRole"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": "omics.amazonaws.com"
        }
      }
    }
  ]
}
```

Wenn Sie diese Dienste verwenden HealthOmics, interagieren Sie auch mit anderen AWS Diensten. Um auf diese Dienste zuzugreifen, verwenden Sie die verwalteten Richtlinien, die von den einzelnen Diensten bereitgestellt werden. Um den Zugriff auf eine Teilmenge von Ressourcen zu beschränken, können Sie die verwalteten Richtlinien als Ausgangspunkt verwenden, um Ihre eigenen restriktiveren Richtlinien zu erstellen.

- AmazonS3 FullAccess — Zugriff auf Amazon S3-Buckets und Objekte, die von Jobs verwendet werden.
- Amazon EC2 ContainerRegistryFullAccess — Zugriff auf Amazon ECR-Registries und Repositorys für Workflow-Container-Images.
- AWSLakeFormationDataAdmin— Zugriff auf Lake Formation Formation-Datenbanken und -Tabellen, die von Analytics-Stores erstellt wurden.
- ResourceGroupsandTagEditorFullAccess— Taggen Sie HealthOmics Ressourcen mit HealthOmics Tagging-API-Vorgängen.

Die oben genannten Richtlinien erlauben es einem Benutzer nicht, IAM-Rollen zu erstellen. Damit ein Benutzer mit diesen Berechtigungen einen Job ausführen kann, muss ein Administrator die Servicerolle erstellen, die Zugriff auf Datenquellen gewährt HealthOmics . Weitere Informationen finden Sie unter Servicerollen für AWS HealthOmics.

Definieren Sie benutzerdefinierte IAM-Berechtigungen für Läufe

Sie können jeden Workflow, jede Ausführung oder jede Ausführungsgruppe, auf die in der `StartRun` Anfrage verwiesen wird, in eine Autorisierungsanfrage aufnehmen. Führen Sie dazu die gewünschte Kombination von Workflows, Läufen oder Ausführungsgruppen in der IAM-Richtlinie auf. Sie können beispielsweise die Verwendung eines Workflows auf einen bestimmten Lauf oder eine bestimmte Ausführungsgruppe beschränken. Sie können auch angeben, dass ein Workflow nur mit einer Ausführungsgruppe verwendet werden soll.

Im Folgenden finden Sie ein Beispiel für eine IAM-Richtlinie, die einen einzelnen Workflow mit einer einzigen Ausführungsgruppe ermöglicht.

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/*"
            ]
        }
    ]
}
```

Servicerollen für AWS HealthOmics

Eine Servicerolle ist eine AWS Identity and Access Management (IAM) -Rolle, die einem AWS Service Berechtigungen für den Zugriff auf Ressourcen in Ihrem Konto gewährt. Sie stellen eine Servicerolle bereit AWS HealthOmics , wenn Sie einen Importjob oder eine Ausführung starten.

Die HealthOmics Konsole kann die erforderliche Rolle für Sie erstellen. Wenn Sie die HealthOmics API zur Verwaltung von Ressourcen verwenden, erstellen Sie die Servicerolle mithilfe der IAM-

Konsole. Weitere Informationen finden Sie unter Eine Rolle erstellen, um Berechtigungen an eine zu delegieren. AWS-Service

Für Servicerollen muss die folgende Vertrauensrichtlinie gelten.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "omics.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Die Vertrauensrichtlinie ermöglicht es dem HealthOmics Dienst, die Rolle zu übernehmen.

Themen

- Beispiel für IAM-Dienstrichtlinien
- Beispiel für CloudFormation eine Vorlage

Beispiel für IAM-Dienstrichtlinien

In diesen Beispielen IDs sind Ressourcennamen und Konten Platzhalter, die Sie durch tatsächliche Werte ersetzen müssen.

Das folgende Beispiel zeigt die Richtlinie für eine Servicerolle, die Sie zum Starten eines Laufs verwenden können. Die Richtlinie gewährt Berechtigungen für den Zugriff auf den Amazon S3 S3-Ausgabespeicherort, die Workflow-Protokollgruppe und den Amazon ECR-Container für den Lauf.

> **Note**
>
> Wenn Sie Call Caching für den Lauf verwenden, fügen Sie den Amazon S3 S3-Speicherort für den Run-Cache als Ressource zu den S3-Berechtigungen hinzu.

Example Richtlinie für Servicerollen zum Starten eines Laufs

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/*"
            ]
        }
    ]
}
```

Das folgende Beispiel zeigt die Richtlinie für eine Servicerolle, die Sie für einen Store-Importjob verwenden können. Die Richtlinie gewährt Berechtigungen für den Zugriff auf den Amazon S3-Eingabespeicherort.

Example Servicerolle für den Job „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"
            ]
        }

    ]
}
```

Beispiel für CloudFormation eine Vorlage

Die folgende CloudFormation Beispielvorlage erstellt eine Servicerolle, die den HealthOmics Zugriff auf Amazon S3 S3-Buckets mit einem `omics-` Präfix und das Hochladen von Workflow-Protokollen ermöglicht.

Example Berechtigungen für Referenzspeicher, Amazon S3 und 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"
              ]
            }
          }
        ]
      }
```

Amazon-ECR-Berechtigungen

Bevor der HealthOmics Service einen Workflow in einem Container aus Ihrem privaten Amazon ECR-Repository ausführen kann, erstellen Sie eine Ressourcenrichtlinie für das Repository. Die Richtlinie erteilt dem HealthOmics Service die Erlaubnis, den Container zu verwenden. Sie fügen diese Ressourcenrichtlinie jedem privaten Repository hinzu, auf das der Workflow verweist.

> **Note**
>
> Das private Repository und der Workflow müssen sich in derselben Region befinden.

Wenn der Workflow und das Repository von unterschiedlichen AWS Konten verwaltet werden, müssen Sie kontenübergreifende Berechtigungen konfigurieren.

Sie müssen keinen zusätzlichen Repository-Zugriff für gemeinsam genutzte Workflows gewähren. Sie können jedoch Richtlinien erstellen, die bestimmten Workflows den Zugriff auf das Container-Image erlauben oder verweigern.

Um die Amazon ECR-Pull-Through-Cache-Funktion verwenden zu können, müssen Sie eine Registrierungsberechtigungsrichtlinie erstellen.

In den folgenden Abschnitten wird beschrieben, wie Amazon ECR-Ressourcenberechtigungen für diese Szenarien konfiguriert werden. Weitere Informationen zu Berechtigungen in Amazon ECR finden Sie unter Private Registrierungsberechtigungen in Amazon ECR.

Themen

- Erstellen Sie eine Ressourcenrichtlinie für das Amazon ECR-Repository
- Workflows mit kontenübergreifenden Containern ausführen
- Amazon ECR-Richtlinien für gemeinsam genutzte Workflows
- Richtlinien für den Amazon ECR-Pull-Through-Cache

Erstellen Sie eine Ressourcenrichtlinie für das Amazon ECR-Repository

Erstellen Sie eine Ressourcenrichtlinie, damit der HealthOmics Service einen Workflow mithilfe eines Containers im Repository ausführen kann. Die Richtlinie gewährt dem HealthOmics Service Principal die Erlaubnis, auf die erforderlichen Amazon ECR-Aktionen zuzugreifen.

Gehen Sie wie folgt vor, um die Richtlinie zu erstellen:

1. Öffnen Sie die Seite mit den privaten Repositorys in der Amazon ECR-Konsole und wählen Sie das Repository aus, auf das Sie Zugriff gewähren möchten.
2. Wählen Sie in der Seitenleisten-Navigation die Option Berechtigungen aus.
3. Wählen Sie Bearbeiten aus.
4. Wählen Sie Richtlinien-JSON bearbeiten aus.
5. Fügen Sie die folgende Richtlinienerklärung hinzu und wählen Sie dann Speichern aus.

   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": "*"
        }
    ]
}
```

Workflows mit kontenübergreifenden Containern ausführen

Wenn der Workflow und der Container von unterschiedlichen AWS Konten verwaltet werden, müssen Sie die folgenden kontoübergreifenden Berechtigungen konfigurieren:

1. Aktualisieren Sie die Amazon ECR-Richtlinie für das Repository, um dem Konto, das für den Workflow verantwortlich ist, ausdrücklich die Erlaubnis zu erteilen.
2. Aktualisieren Sie die Servicerolle für das Konto, dem der Workflow gehört, um ihm Zugriff auf das Container-Image zu gewähren.

Das folgende Beispiel zeigt eine Amazon ECR-Ressourcenrichtlinie, die Zugriff auf das Konto gewährt, dem der Workflow gehört.

In diesem Beispiel:

- Workflow-Konto-ID: 111122223333
- Konto-ID des Container-Repositorys: 444455556666
- Name des Containers: 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": "*"
        }
    ]
}
```

Um die Einrichtung abzuschließen, fügen Sie der Servicerolle des Accounts, dem der Workflow gehört, die folgende Richtlinienanweisung hinzu. Die Richtlinie gewährt der Servicerolle die Erlaubnis, auf das Container-Image „samtools“ zuzugreifen. Achten Sie darauf, die Kontonummern, den Container-Namen und die Region durch Ihre eigenen Werte zu ersetzen.

```
{
    "Sid": "CrossAccountEcrRepoPolicy",
    "Effect": "Allow",
    "Action": ["ecr:BatchCheckLayerAvailability", "ecr:BatchGetImage",
 "ecr:GetDownloadUrlForLayer"],
    "Resource": "arn:aws:ecr:us-west-2:444455556666:repository/samtools"
}
```

Amazon ECR-Richtlinien für gemeinsam genutzte Workflows

> **Note**
>
> HealthOmics ermöglicht einem gemeinsam genutzten Workflow automatisch den Zugriff auf das Amazon ECR-Repository im Konto des Workflow-Besitzers, während der Workflow im Konto des Abonnenten ausgeführt wird. Sie müssen keinen zusätzlichen Repository-Zugriff für gemeinsam genutzte Workflows gewähren. Weitere Informationen finden Sie unter HealthOmics Workflows teilen.

Standardmäßig haben Abonnenten keinen Zugriff auf das Amazon ECR-Repository, um die zugrunde liegenden Container zu verwenden. Optional können Sie den Zugriff auf das Amazon ECR-

Repository anpassen, indem Sie der Ressourcenrichtlinie des Repositorys Bedingungsschlüssel hinzufügen. Die folgenden Abschnitte enthalten Beispielrichtlinien.

Beschränken Sie den Zugriff auf bestimmte Workflows

Sie können einzelne Workflows in einer Bedingungserklärung auflisten, sodass nur diese Workflows Container im Repository verwenden können. Der SourceArnBedingungsschlüssel gibt den ARN des gemeinsam genutzten Workflows an. Das folgende Beispiel erteilt dem angegebenen Workflow die Erlaubnis, dieses Repository zu verwenden.

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"
          }
       }
     }
  ]
}
```

Beschränken Sie den Zugriff auf bestimmte Konten

Sie können Abonnentenkonten in einer Bedingungserklärung auflisten, sodass nur diese Konten berechtigt sind, Container im Repository zu verwenden. Der SourceAccountBedingungsschlüssel

gibt den AWS-Konto des Abonnenten an. Das folgende Beispiel erteilt dem angegebenen Konto die Erlaubnis, dieses Repository zu verwenden.

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"
        }
      }
    }
  ]
}
```

Sie können Amazon ECR-Berechtigungen auch bestimmten Abonnenten verweigern, wie in der folgenden Beispielrichtlinie dargestellt.

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"
              }
            }
        }
    ]
}
```

Richtlinien für den Amazon ECR-Pull-Through-Cache

Um den Amazon ECR-Pull-Through-Cache zu verwenden, erstellen Sie eine Registrierungsberechtigungsrichtlinie. Sie erstellen auch eine Repository-Erstellungsvorlage, die die Berechtigungen für die Repositorys definiert, die vom Amazon ECR-Pull-Through-Cache erstellt wurden.

Die folgenden Abschnitte enthalten Beispiele für diese Richtlinien. Weitere Informationen zum Pull-Through-Cache finden Sie unter Synchronisieren einer Upstream-Registrierung mit einer privaten Amazon ECR-Registrierung im Amazon Elastic Container Registry-Benutzerhandbuch.

Richtlinie für Registrierungsberechtigungen

Um den Amazon ECR-Pull-Through-Cache zu verwenden, erstellen Sie eine Registrierungsberechtigungsrichtlinie. Die Richtlinie für Registrierungsberechtigungen ermöglicht die Kontrolle über Replikations- und Pull-Through-Cache-Berechtigungen.

Für die kontenübergreifende Replikation müssen Sie ausdrücklich zulassen, AWS-Konto dass jeder Benutzer seine Repositorys in Ihre Registrierung replizieren kann.

Wenn Sie eine Pull-Through-Cache-Regel erstellen, kann jeder IAM-Prinzipal, der über die Berechtigung zum Abrufen von Bildern aus einer privaten Registrierung verfügt, standardmäßig auch

die Pull-Through-Cacheregel verwenden. Sie können Registrierungsberechtigungen verwenden, um diese Berechtigungen weiter auf bestimmte Repositorys einzuschränken.

Fügen Sie dem Konto, dem das Container-Image gehört, eine Registrierungsberechtigungsrichtlinie hinzu.

Im folgenden Beispiel ermöglicht die Richtlinie dem HealthOmics Dienst, Repositorys für jede Upstream-Registrierung zu erstellen und Upstream-Pull-Anfragen von den erstellten Repositorys aus zu initiieren.

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/*"
            ]
        }
    ]
}
```

Vorlage für die Erstellung eines Repository

Um den Pull-Through-Cache in verwenden zu können HealthOmics, muss das Amazon ECR-Repository über eine Vorlage für die Repository-Erstellung verfügen. Die Vorlage definiert die Konfigurationseinstellungen für die privaten Repositorys, die für eine Upstream-Registrierung erstellt wurden.

Jede Vorlage enthält ein Repository-Namespace-Präfix, das Amazon ECR verwendet, um neue Repositorys einer bestimmten Vorlage zuzuordnen. In Vorlagen kann die Konfiguration für alle Repository-Einstellungen festgelegt werden, einschließlich ressourcenbasierter Zugriffsrichtlinien, Unveränderlichkeit von Tags, Verschlüsselung und Lebenszyklusrichtlinien. Weitere Informationen finden Sie unter Vorlagen für die Erstellung von Repositorys im Amazon Elastic Container Registry User Guide.

Im folgenden Beispiel ermöglicht die Richtlinie dem HealthOmics Service, Upstream-Pull-Anfragen von den Upstream-Repositorys zu initiieren.

JSON

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

Richtlinien für den kontenübergreifenden Zugriff auf Amazon ECR

Für den kontoübergreifenden Zugriff aktualisiert der Eigentümer des privaten Repositorys die Registrierungsberechtigungsrichtlinie und die Vorlage für die Erstellung des Repositorys, um den Zugriff für das andere Konto und die Ausführungsrolle dieses Kontos zu ermöglichen.

Fügen Sie in der Registrierungsberechtigungsrichtlinie eine Richtlinienerklärung hinzu, um der Ausführungsrolle des anderen Kontos den Zugriff auf die Amazon ECR-Aktionen zu ermöglichen:

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/*"
        }
    ]
}
```

Fügen Sie in der Vorlage für die Repository-Erstellung eine Richtlinienerklärung hinzu, damit die Run-Rolle des anderen Accounts auf die neuen Container-Images zugreifen kann. Optional können Sie Bedingungsanweisungen hinzufügen, um den Zugriff auf bestimmte Workflows zu beschränken:

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"
                }
            }
        }
    ]
}
```

Fügen Sie Berechtigungen für zwei zusätzliche Aktionen (CreateRepository und BatchImportUpstreamImage) in der Ausführungsrolle hinzu und geben Sie die Ressource an, auf die die Ausführungsrolle zugreifen kann.

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 Berechtigungen für Ressourcen

AWS HealthOmics erstellt in Ihrem Namen Ressourcen in anderen Diensten und greift auf diese zu, wenn Sie einen Job ausführen oder einen Shop erstellen. In einigen Fällen müssen Sie

Berechtigungen in anderen Diensten konfigurieren, um auf Ressourcen zuzugreifen oder den Zugriff darauf HealthOmics zu ermöglichen.

Informationen zu Ressourcenberechtigungen im Zusammenhang mit Amazon ECR finden Sie unterAmazon-ECR-Berechtigungen.

Lake-Formation-Berechtigungen

Bevor Sie Analysefunktionen in verwenden HealthOmics, konfigurieren Sie die Standard-Datenbankeinstellungen in Lake Formation.

So konfigurieren Sie Ressourcenberechtigungen in Lake Formation

1. Öffnen Sie die Seite mit den Datenkatalogeinstellungen in der Lake Formation Formation-Konsole.
2. Deaktivieren Sie die IAM-Zugriffskontrollanforderungen für Datenbanken und Tabellen unter Standardberechtigungen für neu erstellte Datenbanken und Tabellen.
3. Wählen Sie Speichern.

HealthOmics Analytics akzeptiert auto Daten, wenn Ihre Servicerichtlinie über die richtigen RAM-Berechtigungen verfügt, wie im folgenden Beispiel.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "omics:*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ram:AcceptResourceShareInvitation",
        "ram:GetResourceShareInvitations"
```

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

Berechtigungen für den Datenzugriff mit Amazon S3 URIs

Sie können mithilfe von HealthOmics API-Operationen oder Amazon S3 S3-API-Vorgängen auf Sequenzspeicherdaten zugreifen.

Für den HealthOmics API-Zugriff werden die HealthOmics Berechtigungen über eine IAM-Richtlinie verwaltet. Für S3 Access sind jedoch zwei Konfigurationsebenen erforderlich: die ausdrückliche Genehmigung in der S3-Zugriffsrichtlinie des Stores und eine IAM-Richtlinie. Weitere Informationen zur Verwendung von IAM-Richtlinien mit HealthOmics finden Sie unter Servicerollen für. HealthOmics

Es gibt drei Möglichkeiten, die Fähigkeit, Objekte mit Amazon S3 zu lesen, gemeinsam zu nutzen APIs:

1. Richtlinienbasiertes Teilen — Für diese gemeinsame Nutzung muss der IAM-Prinzipal sowohl in der S3-Zugriffsrichtlinie aktiviert als auch eine IAM-Richtlinie geschrieben und an den IAM-Prinzipal angehängt werden. Weitere Informationen finden Sie im nächsten Thema.
2. Vorsigniert URLs — Sie können auch eine gemeinsam nutzbare vorsignierte URL für eine Datei im Sequenzspeicher generieren. Weitere Informationen zum Erstellen von Presigned URLs mit Amazon S3 finden Sie unter Using presigned URLs in der Amazon S3 S3-Dokumentation. Die S3-Zugriffsrichtlinie für Sequence Store unterstützt Anweisungen zur Einschränkung der Funktionen vorsignierter URLs.
3. Angenommene Rollen — Erstellen Sie eine Rolle innerhalb des Kontos des Datenbesitzers, die über eine Zugriffsrichtlinie verfügt, die es Benutzern ermöglicht, diese Rolle anzunehmen.

Themen

Auf Richtlinien basierendes Teilen

Wenn Sie über eine direkte S3-URI auf Sequenzspeicherdaten zugreifen, HealthOmics werden erweiterte Sicherheitsmaßnahmen für die zugehörige S3-Bucket-Zugriffsrichtlinie bereitgestellt.

Die folgenden Regeln gelten für neue S3-Zugriffsrichtlinien. Für bestehende Richtlinien gelten die Regeln, wenn Sie die Richtlinie das nächste Mal aktualisieren:

- Die S3-Zugriffsrichtlinien unterstützen die folgenden Richtlinienelemente
  - Version, ID, Aussage, Sid, Wirkung, Prinzip, Aktion, Ressource, Zustand
- Die S3-Zugriffsrichtlinien unterstützen die folgenden Bedingungsschlüssel:
  - s3:ExistingObjectTag/<key>, s3: prefix, s3: signatureversion, s3: TlsVersion
  - Richtlinien unterstützen auch aws: PrincipalArn mit den folgenden Bedingungsoperatoren: und ArnEquals ArnLike

Wenn Sie versuchen, eine Richtlinie hinzuzufügen oder zu aktualisieren, sodass sie ein Element oder eine Bedingung enthält, die nicht unterstützt wird, lehnt das System die Anfrage ab.

Themen

Standardmäßige S3-Zugriffsrichtlinie

Wenn Sie einen Sequenzspeicher erstellen, HealthOmics wird eine standardmäßige S3-Zugriffsrichtlinie erstellt, die dem Root-Konto des Datenspeicher-Besitzers die folgenden Berechtigungen für alle zugänglichen Objekte im Sequenzspeicher gewährt: S3: GetObjectGetObjectTagging, S3 und S3:ListBucket. Die erstellte Standardrichtlinie lautet:

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/*"
        }
    ]
}
```

Anpassen der Zugriffsrichtlinie

Wenn die S3-Zugriffsrichtlinie leer ist, ist kein S3-Zugriff zulässig. Wenn es eine bestehende Richtlinie gibt und Sie den S3-Zugriff entfernen müssen, verwenden Sie diese Option, `deleteS3AccessPolicy` um den gesamten Zugriff zu entfernen.

Um Einschränkungen für das Teilen hinzuzufügen oder anderen Konten Zugriff zu gewähren, können Sie die Richtlinie mithilfe der `PutS3AccessPolicy` API aktualisieren. Aktualisierungen der Richtlinie dürfen nicht über das Präfix für den Sequenzspeicher oder die angegebenen Aktionen hinausgehen.

IAM-Richtlinie

Um einem Benutzer oder IAM-Prinzipal mithilfe von Amazon S3 Zugriff zu gewähren APIs, muss zusätzlich zu den Berechtigungen in der S3-Zugriffsrichtlinie eine IAM-Richtlinie erstellt und an den Principal angehängt werden, um Zugriff zu gewähren. Eine Richtlinie, die den Zugriff auf die Amazon S3 S3-API ermöglicht, kann auf Sequenzspeicher-Ebene oder auf Leseset-Ebene angewendet werden. Auf der Ebene des Lesesatzes können die Berechtigungen entweder durch das Präfix oder mithilfe von Ressourcen-Tag-Filtern für Proben- oder Betreff-ID-Muster eingeschränkt werden.

Wenn der Sequenzspeicher einen vom Kunden verwalteten Schlüssel (CMK) verwendet, muss der Principal auch über die Rechte verfügen, den KMS-Schlüssel für die Entschlüsselung zu verwenden. Weitere Informationen finden Sie unter Kontoübergreifender KMS-Zugriff im Entwicklerhandbuch. AWS Key Management Service

Das folgende Beispiel gewährt einem Benutzer Zugriff auf einen Sequenzspeicher. Sie können den Zugriff mit zusätzlichen Bedingungen oder ressourcenbasierten Filtern verfeinern.

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/*"
        }
      }
    }
  ]
}
```

Tag-basierte Zugriffskontrolle

Um die tagbasierte Zugriffskontrolle verwenden zu können, muss der Sequenzspeicher zunächst aktualisiert werden, um die zu verwendenden Tagschlüssel weiterzugeben. Diese Konfiguration wird bei der Erstellung oder Aktualisierung des Sequenzspeichers festgelegt. Sobald die Tags weitergegeben wurden, können Tag-Bedingungen verwendet werden, um weitere Einschränkungen hinzuzufügen. Die Einschränkungen können in der S3-Zugriffsrichtlinie oder in der IAM-Richtlinie festgelegt werden. Im Folgenden finden Sie ein Beispiel für eine tabulatorbasierte S3-Zugriffsrichtlinie, die festgelegt werden würde:

```
{
    "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"
        }
    }
}
```

Beispiel für eine Einschränkung

Szenario: Erstellung einer Freigabe, bei der der Dateneigentümer die Möglichkeit eines Benutzers einschränken kann, „zurückgezogene“ Daten herunterzuladen.

In diesem Szenario verwaltete ein Datenbesitzer (Konto #111111111111) einen Datenspeicher. Dieser Dateneigentümer gibt die Daten an eine Vielzahl von Drittbenutzern weiter, darunter auch an einen Forscher (Konto #999999999999). Im Rahmen der Datenverwaltung erhält der Dateneigentümer regelmäßig Anfragen zum Widerruf der Daten eines Teilnehmers. Um diesen Widerruf zu verwalten, schränkt der Dateneigentümer bei Erhalt der Anfrage zunächst den direkten Download-Zugriff ein und löscht die Daten schließlich gemäß seinen Anforderungen.

Um diesem Bedarf gerecht zu werden, richtet der Dateneigentümer einen Sequenzspeicher ein und jeder Lesesatz erhält eine Markierung für „Status“, die auf „zurückgezogen“ gesetzt wird, wenn die Auszahlungsanforderung eingeht. Bei Daten, bei denen das Tag auf diesen Wert gesetzt ist, soll sichergestellt werden, dass kein Benutzer „GetObject“ für diese Datei ausführen kann. Um dieses Setup durchzuführen, muss der Dateneigentümer sicherstellen, dass zwei Schritte unternommen werden.

Schritt 1. Stellen Sie für den Sequenzspeicher sicher, dass das Status-Tag aktualisiert wird, damit es weitergegeben werden kann. Dies erfolgt durch Hinzufügen des Schlüssels „Status“ in die Befehlszeile `propogatedSetLevelTags` beim Aufrufen oder `createSequenceStore` `updateSequenceStore.`

Schritt 2. Aktualisieren Sie die S3-Zugriffsrichtlinie des Stores, um GetObject auf Objekte zu beschränken, deren Status-Tag auf „zurückgezogen“ gesetzt ist. Dies geschieht, indem die Zugriffsrichtlinie des Shops mithilfe der `PutS3AccesPolicy` API aktualisiert wird. Die folgende Richtlinie würde es Kunden ermöglichen, die zurückgezogenen Dateien weiterhin zu sehen, wenn sie Objekte anbieten, sie aber daran hindern, darauf zuzugreifen:

- Aussage 1 (restrictedGetWithdrawal): Das Konto 999999999999 kann keine zurückgezogenen Objekte abrufen.
- Aussage 2 (ownerGetAll): Konto 111111111111, der Datenbesitzer, kann alle Objekte abrufen, auch Objekte, die zurückgezogen wurden.
- Aussage 3 (everyoneListAll): Alle gemeinsam genutzten Konten, 111111111111 und 999999999999, können den Vorgang für das gesamte Präfix ausführen. 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/*"
                }
            }
        }
    ]
}
```

Sicherheit in AWS HealthOmics

Cloud-Sicherheit AWS hat höchste Priorität. Als AWS Kunde profitieren Sie von Rechenzentren und Netzwerkarchitekturen, die darauf ausgelegt sind, die Anforderungen der sicherheitssensibelsten Unternehmen zu erfüllen.

Sicherheit ist eine gemeinsame AWS Verantwortung von Ihnen und Ihnen. Das Modell der geteilten Verantwortung beschreibt dies als Sicherheit der Cloud selbst und Sicherheit in der Cloud:

- Sicherheit der Cloud — AWS ist verantwortlich für den Schutz der Infrastruktur, auf der AWS Dienste in der ausgeführt AWS Cloud werden. AWS bietet Ihnen auch Dienste, die Sie sicher nutzen können. Externe Prüfer testen und verifizieren regelmäßig die Wirksamkeit unserer Sicherheitsmaßnahmen im Rahmen der AWS . Weitere Informationen zu den Compliance-Programmen, die für AWS gelten HealthOmics, finden Sie unter AWS Services im Bereich nach Compliance-Programm AWS .
- Sicherheit in der Cloud — Ihre Verantwortung richtet sich nach dem AWS Service, den Sie nutzen. Sie sind auch für andere Faktoren verantwortlich, etwa für die Vertraulichkeit Ihrer Daten, für die Anforderungen Ihres Unternehmens und für die geltenden Gesetze und Vorschriften.

Diese Dokumentation hilft Ihnen zu verstehen, wie Sie das Modell der gemeinsamen Verantwortung bei der Nutzung von AWS anwenden können HealthOmics. In den folgenden Themen erfahren Sie, wie Sie AWS konfigurieren HealthOmics , um Ihre Sicherheits- und Compliance-Ziele zu erreichen. Sie lernen auch, wie Sie andere AWS Services nutzen können, die Sie bei der Überwachung und Sicherung Ihrer HealthOmics AWS-Ressourcen unterstützen.

Topics

Datenschutz in AWS HealthOmics

Das AWS Modell der gilt für den Datenschutz in AWS HealthOmics. Wie in diesem Modell beschrieben, AWS ist verantwortlich für den Schutz der globalen Infrastruktur, auf der alle Systeme laufen AWS Cloud. Sie sind dafür verantwortlich, die Kontrolle über Ihre in dieser Infrastruktur gehosteten Inhalte zu behalten. Sie sind auch für die Sicherheitskonfiguration und die Verwaltungsaufgaben für die von Ihnen verwendeten AWS-Services verantwortlich. Weitere Informationen zum Datenschutz finden Sie unter Häufig gestellte Fragen zum Datenschutz. Informationen zum Datenschutz in Europa finden Sie im Blog-Beitrag AWS -Modell der geteilten Verantwortung und in der DSGVO im AWS -Sicherheitsblog.

Aus Datenschutzgründen empfehlen wir, dass Sie AWS-Konto Anmeldeinformationen schützen und einzelne Benutzer mit AWS IAM Identity Center oder AWS Identity and Access Management (IAM) einrichten. So erhält jeder Benutzer nur die Berechtigungen, die zum Durchführen seiner Aufgaben erforderlich sind. Außerdem empfehlen wir, die Daten mit folgenden Methoden schützen:

- Verwenden Sie für jedes Konto die Multi-Faktor-Authentifizierung (MFA).
- Wird verwendet SSL/TLS , um mit AWS Ressourcen zu kommunizieren. Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Richten Sie die API und die Protokollierung von Benutzeraktivitäten mit ein AWS CloudTrail. Informationen zur Verwendung von CloudTrail Pfaden zur Erfassung von AWS Aktivitäten finden Sie unter Arbeiten mit CloudTrail Pfaden im AWS CloudTrail Benutzerhandbuch.
- Verwenden Sie AWS Verschlüsselungslösungen zusammen mit allen darin enthaltenen Standardsicherheitskontrollen AWS-Services.
- Verwenden Sie erweiterte verwaltete Sicherheitsservices wie Amazon Macie, die dabei helfen, in Amazon S3 gespeicherte persönliche Daten zu erkennen und zu schützen.
- Wenn Sie für den Zugriff AWS über eine Befehlszeilenschnittstelle oder eine API FIPS 140-3-validierte kryptografische Module benötigen, verwenden Sie einen FIPS-Endpunkt. Weitere Informationen über verfügbare FIPS-Endpunkte finden Sie unter Federal Information Processing Standard (FIPS) 140-3.

Wir empfehlen dringend, in Freitextfeldern, z. B. im Feld Name, keine vertraulichen oder sensiblen Informationen wie die E-Mail-Adressen Ihrer Kunden einzugeben. Dies gilt auch, wenn Sie mit AWS HealthOmics oder anderen AWS-Services über die Konsole AWS CLI, API oder arbeiten AWS SDKs. Alle Daten, die Sie in Tags oder Freitextfelder eingeben, die für Namen verwendet werden, können für Abrechnungs- oder Diagnoseprotokolle verwendet werden. Wenn Sie eine URL für einen externen

Server bereitstellen, empfehlen wir dringend, keine Anmeldeinformationen zur Validierung Ihrer Anforderung an den betreffenden Server in die URL einzuschließen.

Verschlüsselung im Ruhezustand

Themen

Zum Schutz vertraulicher Kundendaten im Speicher wird standardmäßig eine Verschlüsselung mit einem serviceeigenen AWS Key Management Service (AWS KMS) -Schlüssel bereitgestellt. AWS HealthOmics Kundenverwaltete Schlüssel werden ebenfalls unterstützt. Weitere Informationen zu vom Kunden verwalteten Schlüsseln finden Sie unter Amazon Key Management Service.

Alle HealthOmics Datenspeicher (Storage und Analytics) unterstützen die Verwendung von kundenverwalteten Schlüsseln. Die Verschlüsselungskonfiguration kann nicht geändert werden, nachdem ein Datenspeicher erstellt wurde. Wenn ein Datenspeicher einen verwendet AWS-eigener Schlüssel, wird er als gekennzeichnet AWS_OWNED_KMS_KEY und der spezifische Schlüssel, der für die Verschlüsselung verwendet wird, wird im Ruhezustand nicht angezeigt.

Bei HealthOmics Workflows werden vom Kunden verwaltete Schlüssel vom temporären Dateisystem nicht unterstützt. Alle Daten im Ruhezustand werden jedoch automatisch mit dem Blockchiffrierverschlüsselungsalgorithmus XTS-AES-256 verschlüsselt, um das Dateisystem zu verschlüsseln. Der IAM-Benutzer und die Rolle, die zum Starten einer Workflow-Ausführung verwendet wurden, müssen auch Zugriff auf die Schlüssel haben, die für Workflow-Eingabe- und -Ausgabe-Buckets verwendet werden. AWS KMS Workflows verwenden keine Zuschüsse, und die AWS KMS Verschlüsselung ist auf Eingabe- und Ausgabe-Amazon S3-Buckets beschränkt. Die IAM-Rolle, die sowohl für den Workflow verwendet wird, APIs muss auch Zugriff auf die verwendeten AWS KMS Schlüssel sowie auf die Eingabe- und Ausgabe-Buckets von Amazon S3 haben. Sie können entweder IAM-Rollen und -Berechtigungen verwenden, um den Zugriff oder Richtlinien zu kontrollieren. AWS KMS Weitere Informationen finden Sie unter Authentifizierung und Zugriffskontrolle für AWS KMS.

Wenn Sie HealthOmics Analytics verwenden AWS Lake Formation , werden alle Entschlüsselungsberechtigungen, die mit der Lake Formation verknüpft sind, auch für die Eingabe- und Ausgabe-Amazon S3-Buckets erteilt. Weitere Informationen zur AWS Lake Formation Verwaltung von Berechtigungen finden Sie in der AWS Lake Formation Dokumentation.

HealthOmics Analytics gewährt Lake Formation kms: Decrypt Berechtigungen zum Lesen der verschlüsselten Daten in einem Amazon S3 S3-Bucket. Solange Sie berechtigt sind, die Daten über Lake Formation abzufragen, können Sie die verschlüsselten Daten lesen. Der Zugriff auf die Daten wird durch die Datenzugriffskontrolle in Lake Formation gesteuert, nicht durch eine KMS-Schlüsselrichtlinie. Weitere Informationen finden Sie in den AWS integrierten AWS-Serviceanfragen in der Lake Formation Formation-Dokumentation.

AWS-eigene Schlüssel

Standardmäßig werden Daten im Ruhezustand automatisch verschlüsselt, da diese Daten vertrauliche Informationen wie personenbezogene Daten (PII) oder geschützte Gesundheitsinformationen (PHI) enthalten können. HealthOmics AWS-eigene Schlüssel AWS-eigene Schlüssel sind nicht in Ihrem Konto gespeichert. Sie sind Teil einer Sammlung von KMS-Schlüsseln, die AWS besitzt und verwaltet, um sie in mehreren AWS-Konten zu verwenden.

AWS-Services können AWS-eigene Schlüssel zum Schutz Ihrer Daten verwendet werden. Sie können ihre Verwendung nicht einsehen, verwalten AWS-eigene Schlüssel, darauf zugreifen oder sie überprüfen. Sie müssen jedoch keine Arbeit verrichten oder Programme ändern, um die Schlüssel zu schützen, mit denen Ihre Daten verschlüsselt werden.

Ihnen wird keine monatliche Gebühr oder Nutzungsgebühr für die Nutzung berechnet AWS-eigene Schlüssel, und sie werden auch nicht auf die AWS KMS KMS-Kontingente für Ihr Konto angerechnet. Weitere Informationen finden Sie unter Von AWS verwaltete Schlüssel.

Kundenseitig verwaltete Schlüssel

HealthOmics unterstützt die Verwendung von symmetrischen, vom Kunden verwalteten Schlüsseln, die Sie erstellen, besitzen und verwalten, um eine zweite Verschlüsselungsebene gegenüber der bestehenden AWS-eigenen Verschlüsselung hinzuzufügen. Da Sie die volle Kontrolle über diese Verschlüsselungsebene haben, können Sie beispielsweise folgende Aufgaben ausführen:

- Einrichtung und Pflege wichtiger Richtlinien, IAM-Richtlinien und Zuschüsse
- Kryptographisches Material mit rotierendem Schlüssel
- Aktivieren und Deaktivieren wichtiger Richtlinien

- Hinzufügen von -Tags
- Erstellen von Schlüsselaliasen
- Schlüssel für das Löschen von Schlüsseln planen

Sie können es auch verwenden CloudTrail , um die Anfragen nachzuverfolgen, die in Ihrem Namen HealthOmics AWS KMS an gesendet werden. Es AWS KMS fallen zusätzliche Gebühren an. Weitere Informationen finden Sie unter Vom Kunden verwaltete Schlüssel.

Einen vom Kunden verwalteten Schlüssel erstellen

Sie können einen symmetrischen, vom Kunden verwalteten Schlüssel mithilfe der AWS-Managementkonsole oder der AWS KMS APIs erstellen.

Folgen Sie den Schritten zur Erstellung symmetrischer kundenverwalteter Schlüssel im AWS Key Management Service Developer Guide.

Schlüsselrichtlinien steuern den Zugriff auf den vom Kunden verwalteten Schlüssel. Jeder vom Kunden verwaltete Schlüssel muss über genau eine Schlüsselrichtlinie verfügen, die aussagt, wer den Schlüssel wie verwenden kann. Wenn Sie einen vom Kunden verwalteten Schlüssel erstellen, können Sie eine Schlüsselrichtlinie angeben. Weitere Informationen finden Sie unter Verwaltung des Zugriffs auf vom Kunden verwaltete Schlüssel im AWS Key Management Service Developer Guide.

Um einen vom Kunden verwalteten Schlüssel mit Ihren HealthOmics Analytics-Ressourcen zu verwenden, benötigt der aufrufende Principal die CreateGrant Operationen kms: in der Schlüsselrichtlinie. Auf diese Weise kann das System mithilfe eines FAS-Tokens einen Zuschuss für einen vom Kunden verwalteten Schlüssel gewähren, der den Zugriff auf einen bestimmten KMS-Schlüssel steuert. Dieser Schlüssel gewährt einem Benutzer Zugriff auf die erforderlichen kms:grant-Operationen. HealthOmics Weitere Informationen finden Sie unter Grants verwenden.

Für HealthOmics Analysen müssen die folgenden API-Operationen für den aufrufenden Prinzipal zulässig sein:

- kms: CreateGrant fügt einem bestimmten vom Kunden verwalteten Schlüssel Zuschüsse hinzu, wodurch der Zugriff auf Grant-Operationen in HealthOmics Analytics ermöglicht wird.
- kms: DescribeKey stellt die vom Kunden verwalteten Schlüsseldetails bereit, die zur Validierung des Schlüssels erforderlich sind. Dies ist für alle Operationen erforderlich.

- kms: GenerateDataKey bietet Zugriff auf verschlüsselte Ressourcen im Ruhezustand für alle Schreibvorgänge. Außerdem stellt diese Aktion vom Kunden verwaltete Schlüsselinformationen bereit, anhand derer der Dienst überprüfen kann, ob der Anrufer Zugriff auf den Schlüssel hat.
- kms:Decrypt bietet Zugriff auf Lese- oder Suchvorgänge für verschlüsselte Ressourcen.

Um einen vom Kunden verwalteten Schlüssel mit Ihren HealthOmics Speicherressourcen zu verwenden, müssen der HealthOmics Service Principal und der Calling Principal in der Schlüsselrichtlinie zugelassen sein. Auf diese Weise kann der Service überprüfen, ob der Anrufer Zugriff auf den Schlüssel hat, und den Service Principal verwenden, um die Geschäftsverwaltung mithilfe des vom Kunden verwalteten Schlüssels auszuführen. Für die HealthOmics Speicherung muss die Schlüsselrichtlinie für den Service Principal die folgenden API-Operationen zulassen:

- kms: DescribeKey stellt die vom Kunden verwalteten Schlüsseldetails bereit, die zur Validierung des Schlüssels erforderlich sind. Dies ist für alle Operationen erforderlich.
- kms: GenerateDataKey bietet Zugriff auf verschlüsselte Ressourcen im Ruhezustand für alle Schreibvorgänge. Außerdem stellt diese Aktion vom Kunden verwaltete Schlüsselinformationen bereit, anhand derer der Dienst überprüfen kann, ob der Anrufer Zugriff auf den Schlüssel hat.
- kms:Decrypt bietet Zugriff auf Lese- oder Suchvorgänge für verschlüsselte Ressourcen.

Das folgende Beispiel zeigt eine Richtlinienanweisung, die es einem Dienstprinzipal ermöglicht, eine HealthOmics Sequenz oder einen Referenzspeicher zu erstellen und mit diesem zu interagieren, der mit dem vom Kunden verwalteten Schlüssel verschlüsselt ist:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
{
    "Effect": "Allow",
    "Principal": {
        "Service": "omics.amazonaws.com"
    },
    "Action": [
        "kms:Decrypt",
        "kms:DescribeKey",
        "kms:Encrypt",
```

```
            "kms:GenerateDataKey*"
        ],
        "Resource": "*"
            }
        ]
}
```

Das folgende Beispiel zeigt eine Richtlinie, die Berechtigungen für einen Datenspeicher zum Entschlüsseln von Daten aus einem Amazon S3 S3-Bucket erstellt.

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"
                  ]
              }
            }
          }
        ]
}
```

Erforderliche IAM-Berechtigungen für die Verwendung eines vom Kunden verwalteten Schlüssels

Beim Erstellen einer Ressource wie eines Datenspeichers mit AWS KMS Verschlüsselung mithilfe eines vom Kunden verwalteten Schlüssels sind Berechtigungen sowohl für die Schlüsselrichtlinie als auch für die IAM-Richtlinie für den IAM-Benutzer oder die IAM-Rolle erforderlich.

Sie können den ViaService Bedingungsschlüssel kms: verwenden, um die Verwendung des KMS-Schlüssels auf Anfragen zu beschränken, die von stammen. HealthOmics

Weitere Informationen zu wichtigen Richtlinien finden Sie unter Enabling IAM-Policies im AWS Key Management Service Developer Guide.

Themen

- Analytics-API-Berechtigungen
- Speicher-API-Berechtigungen
- Wie HealthOmics werden Zuschüsse in AWS KMS verwendet
- Überwachen Sie Ihre Verschlüsselungsschlüssel für AWS HealthOmics

Analytics-API-Berechtigungen

Für HealthOmics Analysen muss der IAM-Benutzer oder die IAM-Rolle, die die Stores erstellt, über die Berechtigungen kms:CreateGrant, kms:GenerateDataKey, kms: Decrypt und die erforderlichen kms: DescribeKey HealthOmics Berechtigungen verfügen.

Speicher-API-Berechtigungen

Für die HealthOmics Speicherung APIs benötigt der IAM-Benutzer oder die IAM-Rolle, die die folgenden API-Operationen aufruft, die aufgeführten Berechtigungen:

CreateReferenceStore, CreateSequenceStore

Um einen Store zu erstellen, muss der IAM-Aufrufer über Berechtigungen und die `kms:DescribeKey` erforderlichen Berechtigungen verfügen. HealthOmics Der HealthOmics Dienstprinzipal ruft auf`kms:GenerateDataKeyWithoutPlaintext`, um Zugriffsprüfungen für das Laden und den Zugriff auf Daten durchzuführen.

StartReadSetImportJob, StartReferenceImportJob

Um Datenimportaufträge zu starten, muss `kms:Decrypt` der IAM-Aufrufer über `kms:GenerateDataKey` Berechtigungen für den KMS-Schlüssel im Store für den Import sowie über `kms:Decrypt` Berechtigungen für den Amazon S3 S3-Bucket verfügen, der die zu importierenden Objekte enthält. Darüber hinaus muss die an den Aufruf übergebene Rolle über `kms:Decrypt` Berechtigungen für den Amazon S3 S3-Bucket verfügen, der die zu importierenden Objekte enthält. Der IAM-Aufrufer muss außerdem über die Berechtigungen verfügen, um die Rolle an den Job weiterzugeben.

CreateMultipartReadSetUpload, UploadReadSetPart, CompleteMultipartReadSetUpload

Um einen mehrteiligen Upload abzuschließen, muss `kms:Decrypt` der IAM-Aufrufer den mehrteiligen Upload erstellen, hochladen und abschließen können. `kms:GenerateDataKey`

StartReadSetExportJob

Um einen Datenexportauftrag zu starten, muss der IAM-Aufrufer über die `kms:Decrypt` Berechtigung für den KMS-Schlüssel im Store zum Exportieren aus `kms:GenerateDataKey` und über `kms:Decrypt` Berechtigungen für den Amazon S3 S3-Bucket verfügen, der die Objekte empfängt. Darüber hinaus muss die an den Aufruf übergebene Rolle über `kms:Decrypt` Berechtigungen für den Amazon S3 S3-Bucket verfügen, der die Objekte empfängt. Der IAM-Aufrufer muss außerdem über Berechtigungen verfügen, um die Rolle an den Job weiterzugeben.

StartReadsetActivationJob

Um einen Read-Set-Aktivierungsjob zu starten, muss der IAM-Aufrufer über `kms:GenerateDataKey` Berechtigungen für die `kms:Decrypt` Objekte verfügen.

GetReference, GetReadSet

Um Objekte aus dem Speicher lesen zu können, muss der IAM-Aufrufer über `kms:Decrypt` Berechtigungen für die Objekte verfügen.

Lesen Sie Set S3 GetObject

Um Objekte aus dem Store mithilfe der Amazon S3 `GetObject` S3-API zu lesen, muss der IAM-Aufrufer über die `kms:Decrypt` Berechtigung für die Objekte verfügen. Legen Sie diese Berechtigung sowohl für vom Kunden verwaltete Schlüssel als auch für Konfigurationen fest AWS-eigener Schlüssel .

Wie HealthOmics werden Zuschüsse in AWS KMS verwendet

HealthOmics Analytics erfordert eine Genehmigung zur Nutzung Ihres vom Kunden verwalteten KMS-Schlüssels. Zuschüsse sind nicht erforderlich und werden auch nicht für HealthOmics Workflows verwendet. HealthOmics Storage verwendet den vom Kunden verwalteten Schlüssel direkt vom Service Principal. Verwenden Sie also keine Grants. Wenn Sie einen Analyseshop erstellen, der mit einem vom Kunden verwalteten Schlüssel verschlüsselt ist, erstellt HealthOmics Analytics in Ihrem Namen einen Zuschuss, indem eine CreateGrantAnfrage an AWS KMS gesendet wird. Zuschüsse in AWS KMS werden verwendet, um HealthOmics Zugriff auf einen KMS-Schlüssel in einem Kundenkonto zu gewähren.

Es wird nicht empfohlen, die Zuschüsse, die HealthOmics Analytics in Ihrem Namen gewährt, zu widerrufen oder zurückzuziehen. Wenn Sie die HealthOmics Genehmigung zur Verwendung der AWS-KMS-Schlüssel in Ihrem Konto widerrufen oder zurückziehen, können Sie HealthOmics nicht auf diese Daten zugreifen, neue Ressourcen, die in den Datenspeicher übertragen werden, verschlüsseln oder sie entschlüsseln, wenn sie abgerufen werden.

Wenn Sie einen Zuschuss für widerrufen oder zurückziehen HealthOmics, erfolgt die Änderung sofort. Um die Zugriffsrechte zu widerrufen, empfehlen wir, den Datenspeicher zu löschen, anstatt die Gewährung zu widerrufen. Wenn Sie den Datenspeicher löschen, werden die Zuschüsse HealthOmics in Ihrem Namen zurückgezogen.

Überwachen Sie Ihre Verschlüsselungsschlüssel für AWS HealthOmics

Sie können CloudTrail damit die Anfragen verfolgen, die in Ihrem Namen AWS HealthOmics AWS KMS an gesendet werden, wenn Sie einen vom Kunden verwalteten Schlüssel verwenden. In den Protokolleinträgen im CloudTrail Protokoll wird HealthOmics .amazonaws.com im Feld UserAgent angezeigt, um Anfragen von eindeutig zu unterscheiden. HealthOmics

Bei den folgenden Beispielen handelt es sich um CloudTrail Ereignisse für CreateGrant GenerateDataKey, Decrypt und zur Überwachung von AWS KMS Vorgängen, die aufgerufen werden, DescribeKey um auf Daten zuzugreifen, HealthOmics die mit Ihrem vom Kunden verwalteten Schlüssel verschlüsselt wurden.

Im Folgenden wird auch gezeigt, wie Sie HealthOmics Analytics CreateGrant den Zugriff auf einen vom Kunden bereitgestellten KMS-Schlüssel ermöglichen, sodass HealthOmics dieser KMS-Schlüssel zur Verschlüsselung aller gespeicherten Kundendaten verwendet werden kann.

Sie müssen keine eigenen Zuschüsse erstellen. HealthOmics erstellt in Ihrem Namen einen Zuschuss, indem Sie eine CreateGrant Anfrage an AWS KMS senden. Zuschüsse AWS KMS werden verwendet, um HealthOmics Zugriff auf einen AWS KMS Schlüssel in einem Kundenkonto zu gewähren.


```
{
    "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"
}
```

Das folgende Beispiel zeigt, wie sichergestellt werden kann GenerateDataKey , dass der Benutzer vor dem Speichern über die erforderlichen Berechtigungen zum Verschlüsseln von Daten verfügt.

```
{
    "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"
}
```

Weitere Informationen

Die folgenden Ressourcen bieten weitere Informationen zur Verschlüsselung von Daten im Ruhezustand.

Weitere Informationen zu den Grundkonzepten von AWS Key Management Service finden Sie in der AWS KMS Dokumentation.

Weitere Informationen zu bewährten Sicherheitsmethoden finden Sie in der AWS KMS Dokumentation.

Verschlüsselung während der Übertragung

AWS HealthOmics verwendet TLS 1.2 und höher, um Daten zu verschlüsseln, die über öffentliche Endpunkte und über Backend-Dienste übertragen werden.

Identitäts- und Zugriffsmanagement in HealthOmics

AWS Identity and Access Management (IAM) hilft einem Administrator AWS-Service , den Zugriff auf Ressourcen sicher zu kontrollieren. AWS IAM-Administratoren kontrollieren, wer authentifiziert (angemeldet) und autorisiert werden kann (über Berechtigungen verfügt), um HealthOmics AWS-Ressourcen zu verwenden. IAM ist ein Programm AWS-Service , das Sie ohne zusätzliche Kosten nutzen können.

Themen

Zielgruppe

Wie Sie AWS Identity and Access Management (IAM) verwenden, hängt von Ihrer Rolle ab:

- Servicebenutzer – Fordern Sie von Ihrem Administrator Berechtigungen an, wenn Sie nicht auf Features zugreifen können (siehe Problembehandlung bei AWS HealthOmics Identität und Zugriff).
- Serviceadministrator – Bestimmen Sie den Benutzerzugriff und stellen Sie Berechtigungsanfragen (siehe Wie AWS HealthOmics funktioniert mit IAM).
- IAM-Administrator – Schreiben Sie Richtlinien zur Zugriffsverwaltung (siehe Beispiele für identitätsbasierte Richtlinien für AWS HealthOmics).

Authentifizierung mit Identitäten

Authentifizierung ist die Art und Weise, wie Sie sich AWS mit Ihren Identitätsdaten anmelden. Sie müssen sich als IAM-Benutzer authentifizieren oder eine IAM-Rolle annehmen. Root-Benutzer des AWS-Kontos

Sie können sich als föderierte Identität anmelden, indem Sie Anmeldeinformationen aus einer Identitätsquelle wie AWS IAM Identity Center (IAM Identity Center), Single Sign-On-Authentifizierung oder Anmeldeinformationen verwenden. Google/Facebook Weitere Informationen zum Anmelden finden Sie unter So melden Sie sich bei Ihrem AWS-Konto an im Benutzerhandbuch für AWS-Anmeldung .

AWS Bietet für den programmatischen Zugriff ein SDK und eine CLI zum kryptografischen Signieren von Anfragen. Weitere Informationen finden Sie unter AWS Signature Version 4 for API requests im IAM-Benutzerhandbuch.

AWS-Konto Root-Benutzer

Wenn Sie einen erstellen AWS-Konto, beginnen Sie mit einer Anmeldeidentität, dem sogenannten AWS-Konto Root-Benutzer, der vollständigen Zugriff auf alle AWS-Services Ressourcen hat. Wir raten ausdrücklich davon ab, den Root-Benutzer für Alltagsaufgaben zu verwenden. Eine Liste der Aufgaben, für die Sie sich als Root-Benutzer anmelden müssen, finden Sie unter Tasks that require root user credentials im IAM-Benutzerhandbuch.

Verbundidentität

Es hat sich bewährt, dass menschliche Benutzer für den Zugriff AWS-Services mithilfe temporärer Anmeldeinformationen einen Verbund mit einem Identitätsanbieter verwenden müssen.

Eine föderierte Identität ist ein Benutzer aus Ihrem Unternehmensverzeichnis, Ihrem Directory Service Web-Identitätsanbieter oder der AWS-Services mithilfe von Anmeldeinformationen aus einer Identitätsquelle zugreift. Verbundene Identitäten übernehmen Rollen, die temporäre Anmeldeinformationen bereitstellen.

Für die zentrale Zugriffsverwaltung empfehlen wir AWS IAM Identity Center. Weitere Informationen finden Sie unter Was ist IAM Identity Center? im AWS IAM Identity Center -Benutzerhandbuch.

IAM-Benutzer und -Gruppen

Ein IAM-Benutzer ist eine Identität mit bestimmten Berechtigungen für eine einzelne Person oder Anwendung. Verwenden Sie möglichst temporäre Anmeldeinformationen anstelle von IAM-Benutzern mit langfristigen Anmeldeinformationen. Weitere Informationen finden Sie im IAM-Benutzerhandbuch unter Erfordern, dass menschliche Benutzer den Verbund mit einem Identitätsanbieter verwenden müssen, um AWS mithilfe temporärer Anmeldeinformationen darauf zugreifen zu können.

Eine IAM-Gruppe spezifiziert eine Sammlung von IAM-Benutzern und erleichtert die Verwaltung von Berechtigungen für große Gruppen von Benutzern. Weitere Informationen finden Sie unter Anwendungsfälle für IAM-Benutzer im IAM-Benutzerhandbuch.

IAM-Rollen

Eine IAM-Rolle ist eine Identität mit spezifischen Berechtigungen, die temporäre Anmeldeinformationen bereitstellt. Sie können eine Rolle übernehmen, indem Sie von einer Benutzer- zu einer IAM-Rolle (Konsole) wechseln AWS CLI oder einen AWS API-Vorgang aufrufen. Weitere Informationen finden Sie unter Methoden, um eine Rolle zu übernehmen im IAM-Benutzerhandbuch.

IAM-Rollen sind nützlich für Verbundbenutzerzugriff, temporäre IAM-Benutzerberechtigungen, kontoübergreifenden Zugriff, dienstübergreifenden Zugriff und Anwendungen, die auf Amazon ausgeführt werden. EC2 Weitere Informationen finden Sie unter Kontoübergreifender Ressourcenzugriff in IAM im IAM-Benutzerhandbuch.

Verwalten des Zugriffs mit Richtlinien

Sie kontrollieren den Zugriff, AWS indem Sie Richtlinien erstellen und diese an Identitäten oder Ressourcen anhängen. AWS Eine Richtlinie definiert Berechtigungen, wenn sie mit einer Identität oder Ressource verknüpft sind. AWS bewertet diese Richtlinien, wenn ein Principal eine Anfrage stellt. Die meisten Richtlinien werden AWS als JSON-Dokumente gespeichert. Weitere Informationen zu JSON-Richtliniendokumenten finden Sie unter Übersicht über JSON-Richtlinien im IAM-Benutzerhandbuch.

Mit Hilfe von Richtlinien legen Administratoren fest, wer Zugriff auf was hat, indem sie definieren, welches Prinzipal welche Aktionen auf welchen Ressourcenund unter welchen Bedingungendurchführen darf.

Standardmäßig haben Benutzer, Gruppen und Rollen keine Berechtigungen. Ein IAM-Administrator erstellt IAM-Richtlinien und fügt sie zu Rollen hinzu, die die Benutzer dann übernehmen können. IAM-Richtlinien definieren Berechtigungen unabhängig von der Methode, die zur Ausführung der Operation verwendet wird.

Identitätsbasierte Richtlinien

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität (Benutzer, Gruppe oder Rolle) anfügen können. Diese Richtlinien steuern, welche Aktionen Identitäten für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter Definieren benutzerdefinierter IAM-Berechtigungen mit vom Kunden verwalteten Richtlinien im IAM-Benutzerhandbuch.

Identitätsbasierte Richtlinien können Inline-Richtlinien (direkt in eine einzelne Identität eingebettet) oder verwaltete Richtlinien (eigenständige Richtlinien, die mit mehreren Identitäten verbunden sind) sein. Informationen dazu, wie Sie zwischen verwalteten und Inline-Richtlinien wählen, finden Sie unter Choose between managed policies and inline policies im IAM-Benutzerhandbuch.

Ressourcenbasierte Richtlinien

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anfügen. Beispiele hierfür sind Vertrauensrichtlinien für IAM-Rollen und Amazon S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Sie müssen in einer ressourcenbasierten Richtlinie einen Prinzipal angeben.

Ressourcenbasierte Richtlinien sind Richtlinien innerhalb dieses Diensts. Sie können AWS verwaltete Richtlinien von IAM nicht in einer ressourcenbasierten Richtlinie verwenden.

Weitere Richtlinientypen

AWS unterstützt zusätzliche Richtlinientypen, mit denen die maximalen Berechtigungen festgelegt werden können, die durch gängigere Richtlinientypen gewährt werden:

- Berechtigungsgrenzen – Eine Berechtigungsgrenze legt die maximalen Berechtigungen fest, die eine identitätsbasierte Richtlinie einer IAM-Entität erteilen kann. Weitere Informationen finden Sie unter Berechtigungsgrenzen für IAM-Entitäten im -IAM-Benutzerhandbuch.

- Richtlinien zur Dienstkontrolle (SCPs) — Geben Sie die maximalen Berechtigungen für eine Organisation oder Organisationseinheit in an AWS Organizations. Weitere Informationen finden Sie unter Service-Kontrollrichtlinien im AWS Organizations -Benutzerhandbuch.
- Richtlinien zur Ressourcenkontrolle (RCPs) — Legen Sie die maximal verfügbaren Berechtigungen für Ressourcen in Ihren Konten fest. Weitere Informationen finden Sie im AWS Organizations Benutzerhandbuch unter Richtlinien zur Ressourcenkontrolle (RCPs).
- Sitzungsrichtlinien – Sitzungsrichtlinien sind erweiterte Richtlinien, die als Parameter übergeben werden, wenn Sie eine temporäre Sitzung für eine Rolle oder einen Verbundbenutzer erstellen. Weitere Informationen finden Sie unter Sitzungsrichtlinien im IAM-Benutzerhandbuch.

Mehrere Richtlinientypen

Wenn für eine Anfrage mehrere Arten von Richtlinien gelten, sind die daraus resultierenden Berechtigungen schwieriger zu verstehen. Informationen darüber, wie AWS bestimmt wird, ob eine Anfrage zulässig ist, wenn mehrere Richtlinientypen betroffen sind, finden Sie unter Bewertungslogik für Richtlinien im IAM-Benutzerhandbuch.

Wie AWS HealthOmics funktioniert mit IAM

Bevor Sie IAM zur Verwaltung des Zugriffs auf AWS verwenden, sollten Sie sich darüber informieren HealthOmics, welche IAM-Funktionen für die Nutzung mit AWS verfügbar sind. HealthOmics

IAM-Funktionen, die Sie mit verwenden können AWS HealthOmics

IAM-Feature	HealthOmics Unterstützung
Identitätsbasierte Richtlinien	Ja
Ressourcenbasierte Richtlinien	Nein
Richtlinienaktionen	Ja
Richtlinienressourcen	Ja
Bedingungsschlüssel für die Richtlinie	Nein
ACLs	Nein

IAM-Feature	HealthOmics Unterstützung
ABAC (Tags in Richtlinien)	Ja
Temporäre Anmeldeinformationen	Ja
Prinzipalberechtigungen	Ja
Servicerollen	Ja
Service-verknüpfte Rollen	Nein

Einen allgemeinen Überblick darüber, wie HealthOmics und andere AWS Dienste mit den meisten IAM-Funktionen funktionieren, finden Sie im AWS IAM-Benutzerhandbuch unter Dienste, die mit IAM funktionieren.

Serviceübergreifende Confused-Deputy-Prävention

Das Problem des verwirrten Stellvertreters ist ein Sicherheitsproblem, bei dem eine Entität, die keine Berechtigung zur Durchführung einer Aktion hat, eine privilegiertere Entität zur Durchführung der Aktion zwingen kann. In: AWS Dienststellenübergreifender Identitätswechsel kann zu Problemen mit verwirrten Stellvertretern führen. Ein serviceübergreifender Identitätswechsel kann auftreten, wenn ein Service (der Anruf-Service) einen anderen Service anruft (den aufgerufenen Service). Der Anruf-Service kann so manipuliert werden, dass er seine Berechtigungen verwendet, um auf die Ressourcen eines anderen Kunden zu reagieren, auf die er sonst nicht zugreifen dürfte. Um dies zu verhindern, bietet AWS Tools, mit denen Sie Ihre Daten für alle Services mit Serviceprinzipalen schützen können, die Zugriff auf Ressourcen in Ihrem Konto erhalten haben.

Wir empfehlen, die Kontextschlüssel `aws:SourceArn`und die `aws:SourceAccount`globalen Bedingungsschlüssel in Ressourcenrichtlinien zu verwenden, um die Berechtigungen zu beschränken, die AWS einem anderen Service für die Ressource HealthOmics erteilt.

Um das Problem der verwirrten Stellvertreter in Rollen zu vermeiden HealthOmics, die von übernommen wurden, legen Sie `arn:aws:omics:`*`region`*`:`*`accountNumber`*`:*` in der `aws:SourceArn` Vertrauensrichtlinie der Rolle den Wert auf fest. Der Platzhalter (*) wendet die Bedingung für alle HealthOmics Ressourcen an.

Die folgende Vertrauensstellungsrichtlinie gewährt HealthOmics Zugriff auf Ihre Ressourcen und verwendet die Kontextschlüssel `aws:SourceArn` und die `aws:SourceAccount` globalen

Bedingungsschlüssel, um das Problem des verwirrten Stellvertreters zu verhindern. Verwenden Sie diese Richtlinie, wenn Sie eine Rolle für erstellen 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:*"
        }
      }
    }
  ]
}
```

Identitätsbasierte Richtlinien für HealthOmics

Unterstützt Richtlinien auf Identitätsbasis: Ja

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität anfügen können, wie z. B. IAM-Benutzern, -Benutzergruppen oder -Rollen. Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter Definieren benutzerdefinierter IAM-Berechtigungen mit vom Kunden verwalteten Richtlinien im IAM-Benutzerhandbuch.

Mit identitätsbasierten IAM-Richtlinien können Sie angeben, welche Aktionen und Ressourcen zugelassen oder abgelehnt werden. Darüber hinaus können Sie die Bedingungen festlegen, unter denen Aktionen zugelassen oder abgelehnt werden. Informationen zu sämtlichen Elementen, die Sie in einer JSON-Richtlinie verwenden, finden Sie in der IAM-Referenz für JSON-Richtlinienelemente im IAM-Benutzerhandbuch.

Beispiele für identitätsbasierte Richtlinien für HealthOmics

Beispiele für HealthOmics identitätsbasierte AWS-Richtlinien finden Sie unter. Beispiele für identitätsbasierte Richtlinien für AWS HealthOmics

Ressourcenbasierte Richtlinien innerhalb HealthOmics

Unterstützt ressourcenbasierte Richtlinien: Nein

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anfügen. Beispiele für ressourcenbasierte Richtlinien sind IAM-Rollen-Vertrauensrichtlinien und Amazon-S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Für die Ressource, an welche die Richtlinie angehängt ist, legt die Richtlinie fest, welche Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen in einer ressourcenbasierten Richtlinie einen Prinzipal angeben. Zu den Prinzipalen können Konten, Benutzer, Rollen, Verbundbenutzer oder gehören. AWS-Services

Um kontoübergreifenden Zugriff zu ermöglichen, können Sie ein gesamtes Konto oder IAM-Entitäten in einem anderen Konto als Prinzipal in einer ressourcenbasierten Richtlinie angeben. Weitere Informationen finden Sie unter Kontoübergreifender Ressourcenzugriff in IAM im IAM-Benutzerhandbuch.

Richtlinienaktionen für HealthOmics

Unterstützt Richtlinienaktionen: Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer Zugriff auf was hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

Das Element `Action` einer JSON-Richtlinie beschreibt die Aktionen, mit denen Sie den Zugriff in einer Richtlinie zulassen oder verweigern können. Nehmen Sie Aktionen in eine Richtlinie auf, um Berechtigungen zur Ausführung des zugehörigen Vorgangs zu erteilen.

Eine Liste der HealthOmics Aktionen finden Sie unter Von AWS definierte Aktionen HealthOmics in der Service Authorization Reference.

Bei Richtlinienaktionen wird vor der Aktion das folgende Präfix HealthOmics verwendet:

```
omics
```

Um mehrere Aktionen in einer einzigen Anweisung anzugeben, trennen Sie sie mit Kommata:

```
"Action": [
      "omics:action1",
      "omics:action2"
         ]
```

Beispiele für HealthOmics identitätsbasierte AWS-Richtlinien finden Sie unter. Beispiele für identitätsbasierte Richtlinien für AWS HealthOmics

Richtlinienressourcen für HealthOmics

Unterstützt Richtlinienressourcen: Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer Zugriff auf was hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

Das JSON-Richtlinienelement `Resource` gibt die Objekte an, auf welche die Aktion angewendet wird. Als Best Practice geben Sie eine Ressource mit dem zugehörigen Amazon-Ressourcennamen (ARN) an. Verwenden Sie für Aktionen, die keine Berechtigungen auf Ressourcenebene unterstützen, einen Platzhalter (*), um anzugeben, dass die Anweisung für alle Ressourcen gilt.

```
"Resource": "*"
```

Eine Liste der HealthOmics Ressourcentypen und ihrer ARNs Eigenschaften finden Sie unter Von AWS definierte Ressourcen HealthOmics in der Service Authorization Reference. Informationen zu den Aktionen, mit denen Sie den ARN jeder Ressource angeben können, finden Sie unter Von AWS definierte Aktionen HealthOmics .

Beispiele für HealthOmics identitätsbasierte AWS-Richtlinien finden Sie unter. Beispiele für identitätsbasierte Richtlinien für AWS HealthOmics

Bedingungsschlüssel für Richtlinien für HealthOmics

Bedingungsschlüssel für Richtlinien werden in nicht unterstützt HealthOmics.

Zugriffskontrolllisten (ACLs) in HealthOmics

Unterstützt ACLs: Nein

Zugriffskontrolllisten (ACLs) steuern, welche Principals (Kontomitglieder, Benutzer oder Rollen) über Zugriffsberechtigungen für eine Ressource verfügen. ACLs ähneln ressourcenbasierten Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

Attributbasierte Zugriffskontrolle (ABAC) mit HealthOmics

Unterstützt ABAC (Tags in Richtlinien): Ja

Die attributbasierte Zugriffskontrolle (ABAC) ist eine Autorisierungsstrategie, bei der Berechtigungen basierend auf Attributen, auch als Tags bezeichnet, definiert werden. Sie können Tags an IAM-Entitäten und AWS -Ressourcen anhängen und dann ABAC-Richtlinien entwerfen, die Operationen zulassen, wenn das Tag des Prinzipals mit dem Tag auf der Ressource übereinstimmt.

Um den Zugriff auf der Grundlage von Tags zu steuern, geben Sie im Bedingungselement einer Richtlinie Tag-Informationen an, indem Sie die Schlüssel `aws:ResourceTag/`*`key-name`*, `aws:RequestTag/`*`key-name`*, oder Bedingung `aws:TagKeys` verwenden.

Wenn ein Service alle drei Bedingungsschlüssel für jeden Ressourcentyp unterstützt, lautet der Wert für den Service Ja. Wenn ein Service alle drei Bedingungsschlüssel für nur einige Ressourcentypen unterstützt, lautet der Wert Teilweise.

Weitere Informationen zu ABAC finden Sie unter Definieren von Berechtigungen mit ABAC-Autorisierung im IAM-Benutzerhandbuch. Um ein Tutorial mit Schritten zur Einstellung von ABAC anzuzeigen, siehe Attributbasierte Zugriffskontrolle (ABAC) verwenden im IAM-Benutzerhandbuch.

Weitere Informationen über das Markieren von HealthOmics-Ressourcen mit Tags finden Sie unter Ressourcen taggen in HealthOmics.

Das folgende Beispiel zeigt, wie Sie eine IAM-Richtlinie schreiben können, die den Zugriff auf eine Ressource ohne ein bestimmtes Tag verweigert.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "omics:*"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "Null": {
                  "aws:RequestTag/MyCustomTag": "true"
                }
            }
        }
    ]
}
```

Temporäre Anmeldeinformationen verwenden mit HealthOmics

Unterstützt temporäre Anmeldeinformationen: Ja

Temporäre Anmeldeinformationen ermöglichen kurzfristigen Zugriff auf AWS Ressourcen und werden automatisch erstellt, wenn Sie einen Verbund verwenden oder die Rollen wechseln. AWS empfiehlt, temporäre Anmeldeinformationen dynamisch zu generieren, anstatt langfristige Zugriffsschlüssel zu verwenden. Weitere Informationen finden Sie unter Temporäre Anmeldeinformationen in IAM und AWS-Services , die mit IAM funktionieren im IAM-Benutzerhandbuch.

Serviceübergreifende Prinzipalberechtigungen für HealthOmics

Unterstützt Forward Access Sessions (FAS): Ja

Forward Access Sessions (FAS) verwenden die Berechtigungen des Prinzipals, der einen aufruft AWS-Service, in Kombination mit der Anforderung, Anfragen an nachgelagerte Dienste AWS-Service zu stellen. Einzelheiten zu den Richtlinien für FAS-Anfragen finden Sie unter Zugriffssitzungen weiterleiten.

Servicerollen für HealthOmics

Unterstützt Servicerollen: Ja

Eine Servicerolle ist eine IAM-Rolle, die ein Service annimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM-Administrator kann eine Servicerolle innerhalb von IAM erstellen, ändern und löschen. Weitere Informationen finden Sie unter Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS-Service im IAM-Benutzerhandbuch.

> **Warning**
>
> Durch das Ändern der Berechtigungen für eine Servicerolle kann die HealthOmics Funktionalität beeinträchtigt werden. Bearbeiten Sie Servicerollen nur, HealthOmics wenn Sie dazu eine Anleitung erhalten.

Dienstbezogene Rollen für HealthOmics

Unterstützt serviceverknüpfte Rollen: Ja

Eine dienstbezogene Rolle ist eine Art von Servicerolle, die mit einer verknüpft ist. AWS-Service Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Dienstbezogene Rollen werden in Ihrem Dienst angezeigt AWS-Konto und gehören dem Dienst. Ein IAM-Administrator kann die Berechtigungen für Service-verknüpfte Rollen anzeigen, aber nicht bearbeiten.

Details zum Erstellen oder Verwalten von serviceverknüpften Rollen finden Sie unter AWS -Services, die mit IAM funktionieren. Suchen Sie in der Tabelle nach einem Service mit einem `Yes` in der Spalte Service-linked role (Serviceverknüpfte Rolle). Wählen Sie den Link Yes (Ja) aus, um die Dokumentation für die serviceverknüpfte Rolle für diesen Service anzuzeigen.

Beispiele für identitätsbasierte Richtlinien für AWS HealthOmics

Standardmäßig sind Benutzer und Rollen nicht berechtigt, HealthOmics AWS-Ressourcen zu erstellen oder zu ändern. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern die Berechtigung erteilen, Aktionen für die Ressourcen auszuführen, die sie benötigen.

Informationen dazu, wie Sie unter Verwendung dieser beispielhaften JSON-Richtliniendokumente eine identitätsbasierte IAM-Richtlinie erstellen, finden Sie unter Erstellen von IAM-Richtlinien (Konsole) im IAM-Benutzerhandbuch.

Einzelheiten zu den von AWS definierten Aktionen und Ressourcentypen HealthOmics, einschließlich des Formats ARNs für die einzelnen Ressourcentypen, finden Sie unter Aktionen, Ressourcen und Bedingungsschlüssel für AWS HealthOmics in der Service Authorization Reference.

Themen

Best Practices für Richtlinien

Identitätsbasierte Richtlinien legen fest, ob jemand HealthOmics AWS-Ressourcen in Ihrem Konto erstellen, darauf zugreifen oder diese löschen kann. Dies kann zusätzliche Kosten für Ihr verursachen AWS-Konto. Wenn Sie identitätsbasierte Richtlinien erstellen oder bearbeiten, befolgen Sie diese Richtlinien und Empfehlungen:

- Erste Schritte mit AWS verwalteten Richtlinien und Umstellung auf Berechtigungen mit den geringsten Rechten — Verwenden Sie die AWS verwalteten Richtlinien, die Berechtigungen für viele gängige Anwendungsfälle gewähren, um damit zu beginnen, Ihren Benutzern und Workloads Berechtigungen zu gewähren. Sie sind in Ihrem verfügbar. AWS-Konto Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie vom AWS Kunden verwaltete Richtlinien definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind. Weitere Informationen finden Sie unter Von AWS verwaltete Richtlinien oder Von AWS verwaltete Richtlinien für Auftragsfunktionen im IAM-Benutzerhandbuch.
- Anwendung von Berechtigungen mit den geringsten Rechten – Wenn Sie mit IAM-Richtlinien Berechtigungen festlegen, gewähren Sie nur die Berechtigungen, die für die Durchführung einer Aufgabe erforderlich sind. Sie tun dies, indem Sie die Aktionen definieren, die für bestimmte Ressourcen unter bestimmten Bedingungen durchgeführt werden können, auch bekannt als die geringsten Berechtigungen. Weitere Informationen zur Verwendung von IAM zum Anwenden von Berechtigungen finden Sie unter Richtlinien und Berechtigungen in IAM im IAM-Benutzerhandbuch.
- Verwenden von Bedingungen in IAM-Richtlinien zur weiteren Einschränkung des Zugriffs – Sie können Ihren Richtlinien eine Bedingung hinzufügen, um den Zugriff auf Aktionen und

Ressourcen zu beschränken. Sie können beispielsweise eine Richtlinienbedingung schreiben, um festzulegen, dass alle Anforderungen mithilfe von SSL gesendet werden müssen. Sie können auch Bedingungen verwenden, um Zugriff auf Serviceaktionen zu gewähren, wenn diese für einen bestimmten Zweck verwendet werden AWS-Service, z. CloudFormation B. Weitere Informationen finden Sie unter IAM-JSON-Richtlinienelemente: Bedingung im IAM-Benutzerhandbuch.

- Verwenden von IAM Access Analyzer zur Validierung Ihrer IAM-Richtlinien, um sichere und funktionale Berechtigungen zu gewährleisten – IAM Access Analyzer validiert neue und vorhandene Richtlinien, damit die Richtlinien der IAM-Richtliniensprache (JSON) und den bewährten IAM-Methoden entsprechen. IAM Access Analyzer stellt mehr als 100 Richtlinienprüfungen und umsetzbare Empfehlungen zur Verfügung, damit Sie sichere und funktionale Richtlinien erstellen können. Weitere Informationen finden Sie unter Richtlinienvalidierung mit IAM Access Analyzer im IAM-Benutzerhandbuch.
- Multi-Faktor-Authentifizierung (MFA) erforderlich — Wenn Sie ein Szenario haben, das IAM-Benutzer oder einen Root-Benutzer in Ihrem System erfordert AWS-Konto, aktivieren Sie MFA für zusätzliche Sicherheit. Um MFA beim Aufrufen von API-Vorgängen anzufordern, fügen Sie Ihren Richtlinien MFA-Bedingungen hinzu. Weitere Informationen finden Sie unter Sicherer API-Zugriff mit MFA im IAM-Benutzerhandbuch.

Weitere Informationen zu bewährten Methoden in IAM finden Sie unter Best Practices für die Sicherheit in IAM im IAM-Benutzerhandbuch.

Verwenden der HealthOmics-Konsole

Um auf die HealthOmics AWS-Konsole zugreifen zu können, benötigen Sie ein Mindestmaß an Berechtigungen. Diese Berechtigungen müssen es Ihnen ermöglichen, Details zu den HealthOmics AWS-Ressourcen in Ihrem aufzulisten und anzuzeigen AWS-Konto. Wenn Sie eine identitätsbasierte Richtlinie erstellen, die strenger ist als die mindestens erforderlichen Berechtigungen, funktioniert die Konsole nicht wie vorgesehen für Entitäten (Benutzer oder Rollen) mit dieser Richtlinie.

Sie müssen Benutzern, die nur die API AWS CLI oder die AWS API aufrufen, keine Mindestberechtigungen für die Konsole gewähren. Stattdessen sollten Sie nur Zugriff auf die Aktionen zulassen, die der API-Operation entsprechen, die die Benutzer ausführen möchten.

Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer

In diesem Beispiel wird gezeigt, wie Sie eine Richtlinie erstellen, die IAM-Benutzern die Berechtigung zum Anzeigen der eingebundenen Richtlinien und verwalteten Richtlinien gewährt, die ihrer

Benutzeridentität angefügt sind. Diese Richtlinie umfasst Berechtigungen zum Ausführen dieser Aktion auf der Konsole oder programmgesteuert mithilfe der API AWS CLI oder 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 verwaltete Richtlinien für AWS HealthOmics

Eine AWS verwaltete Richtlinie ist eine eigenständige Richtlinie, die von erstellt und verwaltet wird AWS. AWS Verwaltete Richtlinien sind so konzipiert, dass sie Berechtigungen für viele gängige Anwendungsfälle bereitstellen, sodass Sie damit beginnen können, Benutzern, Gruppen und Rollen Berechtigungen zuzuweisen.

Beachten Sie, dass AWS verwaltete Richtlinien für Ihre speziellen Anwendungsfälle möglicherweise keine Berechtigungen mit den geringsten Rechten gewähren, da sie allen AWS Kunden zur Verfügung stehen. Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie vom Kunden verwaltete Richtlinien definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind.

Sie können die in AWS verwalteten Richtlinien definierten Berechtigungen nicht ändern. Wenn die in einer AWS verwalteten Richtlinie definierten Berechtigungen AWS aktualisiert werden, wirkt sich das Update auf alle Prinzidentitäten (Benutzer, Gruppen und Rollen) aus, denen die Richtlinie zugeordnet ist. AWS aktualisiert eine AWS verwaltete Richtlinie höchstwahrscheinlich, wenn eine neue Richtlinie eingeführt AWS-Service wird oder neue API-Operationen für bestehende Dienste verfügbar werden.

Weitere Informationen finden Sie unter Von AWS verwaltete Richtlinien im IAM-Benutzerhandbuch.

AWS verwaltete Richtlinie: AmazonOmicsFullAccess

Sie können die `AmazonOmicsFullAccess` Richtlinie an Ihre IAM-Identitäten anhängen, um ihnen vollen Zugriff zu gewähren. HealthOmics

Diese Richtlinie gewährt volle Zugriffsberechtigungen für alle HealthOmics Aktionen. Wenn Sie einen Annotations- oder Variantenspeicher erstellen, gewährt Ihnen Omics auch über eine Resource Share Invitation in der Resource Access Manager (RAM) -Konsole Zugriff auf diesen Speicher. Weitere Informationen zu Resource Share-Einladungen über Lake Formation finden Sie unter Kontoübergreifender Datenaustausch in Lake Formation. Für eine Omics-Administratorrichtlinie benötigen Sie außerdem die folgenden Berechtigungen, um auf Ihren Amazon S3 S3-Bucket zuzugreifen.

- 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 verwaltete Richtlinie: AmazonOmicsReadOnlyAccess

Sie können die `AWSOmicsReadOnlyAccess` Richtlinie an Ihre IAM-Identitäten anhängen, wenn Sie die Berechtigungen für diese Identität auf schreibgeschützten Zugriff beschränken möchten.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "omics:Get*",
                "omics:List*"
            ],
            "Resource": "*"
        }
    ]
}
```

HealthOmics Aktualisierungen der verwalteten Richtlinien AWS

Hier finden Sie Informationen zu Aktualisierungen AWS verwalteter Richtlinien HealthOmics seit Beginn der Nachverfolgung dieser Änderungen durch diesen Dienst. Abonnieren Sie den RSS-Feed auf der Seite HealthOmics Dokumentenverlauf, um automatische Benachrichtigungen über Änderungen an dieser Seite zu erhalten.

Änderungen	Beschreibung	Date
AmazonOmicsFullAccess - Neue Richtlinie hinzugefügt	HealthOmics Es wurde eine neue Richtlinie hinzugefügt, um einem Benutzer vollen Zugriff auf alle Aktionen und Ressourcen zu gewähren. Weitere Informationen hierzu finden Sie unter AmazonOmicsFullAccess.	23. Februar 2023
HealthOmics hat begonnen, Änderungen zu verfolgen	HealthOmics hat begonnen, Änderungen für die AWS verwalteten Richtlinien zu verfolgen.	29. November 2022
AmazonOmicsReadOnlyAccess - Neue Richtlinie hinzugefügt	HealthOmics Es wurde eine neue Richtlinie hinzugefügt, die den Zugriff nur auf Lesezugriff beschränkt. Weitere Informationen hierzu finden Sie unter AmazonOmicsReadOnlyAccess.	29. November 2022

Problembehandlung bei AWS HealthOmics Identität und Zugriff

Verwenden Sie die folgenden Informationen, um häufig auftretende Probleme zu diagnostizieren und zu beheben, die bei der Arbeit mit AWS HealthOmics und IAM auftreten können.

Themen

- Ich bin nicht berechtigt, eine Aktion durchzuführen in HealthOmics
- Ich bin nicht berechtigt, iam auszuführen: PassRole
- Ich möchte Personen außerhalb von mir den Zugriff AWS-Konto auf meine HealthOmics Ressourcen ermöglichen

Ich bin nicht berechtigt, eine Aktion durchzuführen in HealthOmics

Wenn Sie eine Fehlermeldung erhalten, dass Sie nicht zur Durchführung einer Aktion berechtigt sind, müssen Ihre Richtlinien aktualisiert werden, damit Sie die Aktion durchführen können.

Der folgende Beispielfehler tritt auf, wenn der IAM-Benutzer `mateojackson` versucht, über die Konsole Details zu einer fiktiven *`my-example-widget`*-Ressource anzuzeigen, jedoch nicht über `omics:`*`GetWidget`*-Berechtigungen verfügt.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 omics:GetWidget on resource: my-example-widget
```

In diesem Fall muss die Richtlinie für den Benutzer `mateojackson` aktualisiert werden, damit er mit der `omics:`*`GetWidget`*-Aktion auf die *`my-example-widget`*-Ressource zugreifen kann.

Wenn Sie Hilfe benötigen, wenden Sie sich an Ihren AWS Administrator. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Ich bin nicht berechtigt, iam auszuführen: PassRole

Wenn Sie eine Fehlermeldung erhalten, dass Sie nicht berechtigt sind, die `iam:PassRole` Aktion auszuführen, müssen Ihre Richtlinien aktualisiert werden, damit Sie eine Rolle an AWS übergeben können HealthOmics.

Einige AWS-Services ermöglichen es Ihnen, eine bestehende Rolle an diesen Service zu übergeben, anstatt eine neue Servicerolle oder eine mit einem Service verknüpfte Rolle zu erstellen. Hierzu benötigen Sie Berechtigungen für die Übergabe der Rolle an den Dienst.

Der folgende Beispielfehler tritt auf, wenn ein IAM-Benutzer mit dem Namen `marymajor` versucht, die Konsole zu verwenden, um eine Aktion in AWS HealthOmics auszuführen. Die Aktion erfordert jedoch, dass der Service über Berechtigungen verfügt, die durch eine Servicerolle gewährt werden. Mary besitzt keine Berechtigungen für die Übergabe der Rolle an den Dienst.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

In diesem Fall müssen die Richtlinien von Mary aktualisiert werden, um die Aktion `iam:PassRole` ausführen zu können.

Wenn Sie Hilfe benötigen, wenden Sie sich an Ihren AWS Administrator. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Ich möchte Personen außerhalb von mir den Zugriff AWS-Konto auf meine HealthOmics Ressourcen ermöglichen

Sie können eine Rolle erstellen, mit der Benutzer in anderen Konten oder Personen außerhalb Ihrer Organisation auf Ihre Ressourcen zugreifen können. Sie können festlegen, wem die Übernahme der Rolle anvertraut wird. Für Dienste, die ressourcenbasierte Richtlinien oder Zugriffskontrolllisten (ACLs) unterstützen, können Sie diese Richtlinien verwenden, um Personen Zugriff auf Ihre Ressourcen zu gewähren.

Weitere Informationen dazu finden Sie hier:

- Informationen darüber, ob AWS diese Funktionen HealthOmics unterstützt, finden Sie unterWie AWS HealthOmics funktioniert mit IAM.
- Informationen dazu, wie Sie Zugriff auf Ihre Ressourcen gewähren können, AWS-Konten die Ihnen gehören, finden Sie im IAM-Benutzerhandbuch unter Bereitstellen von Zugriff für einen IAM-Benutzer in einem anderen AWS-Konto , den Sie besitzen.
- Informationen dazu, wie Sie Dritten Zugriff auf Ihre Ressourcen gewähren können AWS-Konten, finden Sie AWS-Konten im IAM-Benutzerhandbuch unter Gewähren des Zugriffs für Dritte.
- Informationen dazu, wie Sie über einen Identitätsverbund Zugriff gewähren, finden Sie unter Gewähren von Zugriff für extern authentifizierte Benutzer (Identitätsverbund) im IAM-Benutzerhandbuch.
- Informationen zum Unterschied zwischen der Verwendung von Rollen und ressourcenbasierten Richtlinien für den kontoübergreifenden Zugriff finden Sie unter Kontoübergreifender Ressourcenzugriff in IAM im IAM-Benutzerhandbuch.

Konformitätsvalidierung für AWS HealthOmics

Externe Prüfer bewerten die Sicherheit und Einhaltung von Vorschriften im AWS HealthOmics Rahmen mehrerer AWS Compliance-Programme. Dazu gehören HIPAA, FedRAMP und andere. Die folgende Tabelle zeigt die Compliance-Zertifizierungen für den Service. HealthOmics

Zertifizierung	Link
HIPAA	Referenz für HIPAA-fähige Dienste

Zertifizierung	Link
HiTrust-CSF	Gemeinsamer Sicherheitsrahmen der Health Information Trust Alliance
FedRAMP Moderate (Osten/West)	Risiko- und Genehmigungsmanagementprogramm des Bundes
ISO/CSA-STERN	ISO- und CSA STAR-zertifiziert
C5	Katalog zur Kontrolle der Einhaltung von Cloud-Computing-Vorschriften
DoD CC SRG IL2	Leitfaden für Cloud-Computing-Sicherheitsanforderungen des Verteidigungsministeriums
ENS High	Esquema Nacional de Seguridad
FINMA	Eidgenössische Finanzmarktaufsicht
IST MAP	Programm zur Verwaltung und Bewertung der Sicherheit von Informationssystemen
OSPAR	Prüfbericht für ausgelagerte Dienstleister
PCI	Datensicherheitsstandard der Zahlungskartenbranche
Pinakes	Bankenverband CCI — Qualifizierung durch Dritte
PiTuKri	Kriterien für die Bewertung der Informationssicherheit von Cloud-Diensten
SOC 1,2,3	System- und Organisationskontrollen

Eine Liste aller AWS Services, die für bestimmte Compliance-Programme gelten, finden Sie unter AWS-Services in Umfang nach Compliance-Programm . Allgemeine Informationen finden Sie unter AWS -Compliance-Programme.

Sie können Prüfberichte von Drittanbietern herunterladen unter AWS Artifact. Weitere Informationen finden Sie unter Berichte herunterladen unter .

HealthOmics Datenspeicher verwenden die Proben-ID für die interne Benennung von Dateien und für die Kennzeichnung von Ressourcen. Bevor Sie Daten aufnehmen, überprüfen Sie, ob die Proben-ID PHI-Daten enthält. Ist dies der Fall, ändern Sie die Proben-ID, bevor Sie die Daten aufnehmen. Weitere Informationen finden Sie in den Anleitungen auf der Webseite zur AWS HIPAA-Konformität.

Ihre Verantwortung für die Einhaltung der Vorschriften bei der Nutzung AWS HealthOmics hängt von der Vertraulichkeit Ihrer Daten, den Compliance-Zielen Ihres Unternehmens und den geltenden Gesetzen und Vorschriften ab. AWS stellt die folgenden Ressourcen zur Verfügung, die Sie bei der Einhaltung der Vorschriften unterstützen:

- Schnellstartanleitungen für Sicherheit und Compliance – In diesen Bereitstellungsleitfäden werden architektonische Überlegungen erörtert und Schritte für die Bereitstellung von sicherheits- und konformitätsorientierten Basisumgebungen auf AWS angegeben.
- Whitepaper „Architecting for HIPAA Security and Compliance“ — In diesem Whitepaper wird beschrieben, wie Unternehmen HIPAA-konforme Anwendungen erstellen können AWS .
- AWS Compliance-Ressourcen — Diese Sammlung von Arbeitsmappen und Leitfäden kann auf Ihre Branche und Ihren Standort zutreffen.
- Bewertung von Ressourcen anhand von Regeln im AWS Config Entwicklerhandbuch — AWS Config; bewertet, wie gut Ihre Ressourcenkonfigurationen den internen Praktiken, Branchenrichtlinien und Vorschriften entsprechen.
- AWS Security Hub CSPM— Dieser AWS Service bietet einen umfassenden Überblick über Ihren Sicherheitsstatus, sodass Sie überprüfen können AWS , ob Sie die Sicherheitsstandards und Best Practices der Branche einhalten.

Resilienz in HealthOmics

Die AWS globale Infrastruktur basiert auf Availability AWS-Regionen Zones. AWS-Regionen bieten mehrere physisch getrennte und isolierte Availability Zones, die über Netzwerke mit niedriger Latenz, hohem Durchsatz und hoher Redundanz miteinander verbunden sind. Mithilfe von Availability Zones können Sie Anwendungen und Datenbanken erstellen und ausführen, die automatisch Failover zwischen Zonen ausführen, ohne dass es zu Unterbrechungen kommt. Availability Zones sind besser verfügbar, fehlertoleranter und skalierbarer als herkömmliche Infrastrukturen mit einem oder mehreren Rechenzentren.

Weitere Informationen zu Availability Zones AWS-Regionen und Availability Zones finden Sie unter AWS Globale Infrastruktur.

Zusätzlich zur AWS globalen Infrastruktur HealthOmics bietet AWS mehrere Funktionen zur Unterstützung Ihrer Datenausfallsicherheit und Backup-Anforderungen.

AWS HealthOmics und Schnittstellen-VPC-Endpunkte ()AWS PrivateLink

Sie können eine private Verbindung zwischen Ihrer VPC herstellen und AWS HealthOmics einen VPC-Schnittstellen-Endpunkt erstellen. Schnittstellenendpunkte basieren auf einer Technologie AWS PrivateLink, mit der Sie privat auf HealthOmics API-Operationen zugreifen können, ohne dass ein Internet-Gateway, ein NAT-Gerät, eine VPN-Verbindung oder eine AWS Direct Connect-Verbindung erforderlich ist. Instances in Ihrer VPC benötigen keine öffentlichen IP-Adressen, um mit HealthOmics API-Vorgängen zu kommunizieren. Der Datenverkehr zwischen Ihrer VPC und HealthOmics geht nicht außerhalb des Amazon-Netzwerks.

Jeder Schnittstellenendpunkt wird durch eine oder mehrere Elastic-Network-Schnittstellen in Ihren Subnetzen dargestellt.

Weitere Informationen finden Sie unter Interface VPC Endpoints (AWS PrivateLink) im Amazon VPC-Benutzerhandbuch.

VPC-Endpunktrichtlinien werden HealthOmics für alle Regionen außer Israel (Tel Aviv) unterstützt. Standardmäßig HealthOmics ist der vollständige Zugriff auf über den Endpunkt zulässig.

Überlegungen zu HealthOmics VPC-Endpunkten

Bevor Sie einen Schnittstellen-VPC-Endpunkt für einrichten HealthOmics, sollten Sie die Eigenschaften und Einschränkungen der Schnittstellen-Endpunkte im Amazon VPC-Benutzerhandbuch lesen.

HealthOmics unterstützt Aufrufe aller HealthOmics Speicher-API-Aktionen von Ihrer VPC aus.

VPC-Endpunktrichtlinien werden HealthOmics standardmäßig nicht unterstützt, aber Sie können einen VPC-Endpunkt für den vollen HealthOmics Zugriff auf die HealthOmics Speichervorgänge erstellen. Weitere Informationen finden Sie unter Steuerung des Zugriffs auf Services mit VPC-Endpunkten im Amazon-VPC-Benutzerhandbuch.

Erstellen eines Schnittstellen-VPC-Endpunkts für HealthOmics

Sie können einen VPC-Endpunkt für den HealthOmics Service erstellen, indem Sie die Amazon VPC-Konsole oder die AWS Command Line Interface ()AWS CLI verwenden. Weitere Informationen finden Sie unter Erstellung eines Schnittstellenendpunkts im Benutzerhandbuch für Amazon VPC.

Erstellen Sie einen VPC-Endpunkt für, HealthOmics indem Sie die folgenden Dienstnamen verwenden:

- com.amazonaws. *region*.storage-comics
- com.amazonaws. *region*. control-storage-omics
- com.amazonaws. *region*.analytics-Comics
- com.amazonaws. *region*.workflows-Comics
- com.amazonaws. *region*.tags-Comics

Die Regionen USA Ost (Nord-Virginia) und USA West (Oregon) unterstützen FIPS-Endpunkte. AWS PrivateLink Für diese Regionen können Sie auch die folgenden Dienstnamen verwenden:

- 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

Wenn Sie privates DNS für den Endpunkt aktivieren, können Sie API-Anfragen an stellen, HealthOmics indem Sie den Standard-DNS-Namen für die Region verwenden, `omics.us-east-1.amazonaws.com` z. B.

Weitere Informationen finden Sie unter Zugriff auf einen Service über einen Schnittstellenendpunkt im Benutzerhandbuch für Amazon VPC.

Erstellen einer VPC-Endpunktrichtlinie für HealthOmics

Sie können eine Endpunktrichtlinie an Ihren VPC-Endpunkt anhängen, der den Zugriff auf HealthOmics steuert. Die Richtlinie gibt die folgenden Informationen an:

- Der Prinzipal, der die Aktionen ausführen kann
- Aktionen, die ausgeführt werden können
- Ressourcen, für die Aktionen ausgeführt werden können

Weitere Informationen finden Sie unter Steuerung des Zugriffs auf Services mit VPC-Endpunkten im Amazon-VPC-Benutzerhandbuch.

Beispiel: VPC-Endpunktrichtlinie für HealthOmics Aktionen.

Das Folgende ist ein Beispiel für eine Endpunktrichtlinie für HealthOmics. Wenn diese Richtlinie an einen Endpunkt angehängt ist, gewährt sie allen Prinzipalen auf allen Ressourcen Zugriff auf HealthOmics Aktionen.

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\":\"*\"}]}"
```

Besondere Überlegungen für den Zugriff auf Lesesätze mit Amazon S3 URIs

Um über Amazon S3 auf Lesesätze zuzugreifen, URIs wenn Sie eine private Verbindung verwenden, richten Sie die PrivateLink Schnittstellenendpunkte im Sequenzspeicher ein. Nachdem Sie sie eingerichtet haben, haben die Endpunkte die folgenden Formate:

```
com.amazonaws.region.storage-omics
com.amazonaws.region.control-storage-omics
```

Um Gateway-Endpunkte zu verwenden, folgen Sie der Anleitung Gateway-Endpunkte für Amazon S3, um Ihre Gateway-Endpunkte zu konfigurieren. HealthOmics besitzt den Amazon S3 S3-Bucket, sodass Sie die Bucket-Richtlinie nicht erstellen oder anpassen müssen. Gateway-Endpunkte basieren auf der Richtlinie, die dem Benutzer oder der Rolle zugewiesen ist, die auf die Daten zugreift. Sie können jedoch auch Endgeräte mit restriktiveren Richtlinien konfigurieren. Diese Richtlinien können Zugriffsbeschränkungen beinhalten, die auf dem ARN des Amazon S3 Access Point und den Amazon S3 S3-Aktionen basieren.

Überwachung von AWS HealthOmics

Die Überwachung ist ein wichtiger Bestandteil der Aufrechterhaltung der Zuverlässigkeit, Verfügbarkeit und Leistung von AWS HealthOmics und Ihren anderen AWS Lösungen. AWS bietet die folgenden Überwachungstools, um AWS zu beobachten HealthOmics, zu melden, wenn etwas nicht stimmt, und gegebenenfalls automatische Maßnahmen zu ergreifen:

- Amazon CloudWatch überwacht Ihre AWS Ressourcen und die Anwendungen, auf denen Sie laufen, AWS in Echtzeit. Sie können Kennzahlen erfassen und verfolgen, benutzerdefinierte Dashboards erstellen und Alarme festlegen, die Sie benachrichtigen oder Maßnahmen ergreifen, wenn eine bestimmte Metrik einen von Ihnen festgelegten Schwellenwert erreicht. Sie können beispielsweise die CPU-Auslastung oder andere Kennzahlen Ihrer EC2 Amazon-Instances CloudWatch verfolgen und bei Bedarf automatisch neue Instances starten. Weitere Informationen finden Sie im CloudWatch Amazon-Benutzerhandbuch.
- Mit Amazon CloudWatch Logs können Sie Ihre Protokolldateien von EC2 Amazon-Instances und anderen Quellen überwachen CloudTrail, speichern und darauf zugreifen. CloudWatch Logs kann Informationen in den Protokolldateien überwachen und Sie benachrichtigen, wenn bestimmte Schwellenwerte erreicht werden. Sie können Ihre Protokolldaten auch in einem sehr robusten Speicher archivieren. Weitere Informationen finden Sie im Amazon CloudWatch Logs-Benutzerhandbuch.
- AWS CloudTrail erfasst API-Aufrufe und zugehörige Ereignisse, die von oder im Namen Ihres AWS-Konto -Kontos erfolgten, und übermittelt die Protokolldateien an einen von Ihnen angegebenen Amazon-S3-Bucket. Sie können die Benutzer und Konten, die AWS aufgerufen haben, identifizieren, sowie die Quell-IP-Adresse, von der diese Aufrufe stammen, und den Zeitpunkt der Aufrufe ermitteln. Weitere Informationen finden Sie im AWS CloudTrail - Benutzerhandbuch.
- Amazon EventBridge ist ein serverloser Event-Bus-Service, der es einfach macht, Ihre Anwendungen mit Daten aus einer Vielzahl von Quellen zu verbinden. EventBridge liefert einen Stream von Echtzeitdaten aus Ihren eigenen Anwendungen, Software-as-a-Service (SaaS-) Anwendungen und AWS Diensten und leitet diese Daten an Ziele wie Lambda weiter. Auf diese Weise können Sie Ereignisse überwachen, die in Services auftreten, und ereignisgesteuerte Architekturen erstellen. Weitere Informationen finden Sie im EventBridge Amazon-Benutzerhandbuch.

> **Note**
>
> Für Service-Updates konfigurieren und überwachen Sie Ihr Personal Health Dashboard. Weitere Informationen zur Verwaltung des Dashboards finden Sie unter Erste Schritte mit Ihrem AWS Health Dashboard.

Themen

Protokollierung des S3-Zugriffs

Sie können den Amazon S3 S3-API-Zugriff auf HealthOmics Sequenzspeicherdaten mithilfe der im Store erstellten Zugriffsprotokolle überwachen. Sie können sie verwenden CloudWatch , um den S3-Zugriff über HealthOmics API-Operationen zu überwachen. CloudWatch bietet Einblick in den Amazon S3 S3-Zugriff, der von Ihrem eigenen Konto aus erfolgt. Wenn Sie als Dateneigentümer den Zugriff auf ein Drittanbieter-Konto teilen, ist die Zugriffsprotokollierung in nicht verfügbar CloudWatch. Verwenden Sie stattdessen das S3-Zugriffsprotokoll des Shops, das alle S3-Zugriffe auf die Daten im konfigurierten Amazon S3 S3-Bucket protokolliert.

Konfigurieren Sie S3-Zugriffsprotokolle mithilfe der `UpdateSequenceStore` API-Operationen `CreateSequenceStore` oder. Stellen Sie außerdem sicher, dass der HealthOmics Service Principal (`omics.amazonaws.com`) über `s3:PutObject` Berechtigungen für das konfigurierte S3-Präfix verfügt.

> **Note**
>
> Protokolle verwenden die Standardverschlüsselungskonfiguration des Ziel-Buckets. Wenn der Bucket einen vom Kunden verwalteten Schlüssel verwendet, muss der Service Principal Zugriff darauf haben, den Schlüssel zum Schreiben zu verwenden.

Um die Zugriffsprotokollierung zu deaktivieren, verwenden Sie die Zugriffsprotokollkonfiguration `UpdateSequenceStore` und setzen Sie sie auf leer.

Überwachung HealthOmics mit CloudWatch Metriken

Sie können die HealthOmics Nutzung CloudWatch überwachen. Dabei werden Rohdaten gesammelt und zu lesbaren Metriken verarbeitet, die nahezu in Echtzeit verfügbar sind. Diese Statistiken werden 15 Monate gespeichert, damit Sie auf Verlaufsinformationen zugreifen können und einen besseren Überblick darüber erhalten, wie Ihre Webanwendung oder der Service ausgeführt werden. Sie können auch Alarme einrichten, die auf bestimmte Grenzwerte achten und Benachrichtigungen senden oder Aktivitäten auslösen, wenn diese Grenzwerte erreicht werden. Weitere Informationen finden Sie im CloudWatch Amazon-Benutzerhandbuch.

Der AWS HealthOmics Service meldet die folgenden Metriken im `AWS/Omics` Namespace.

Die Metriken zur Anzahl der API-Aufrufe werden für Folgendes AWS HealthOmics APIs gemeldet. Es wird nur die Dimension API-Operation gemeldet.

- Referenz und Referenzspeicher APIs —CreateReferenceStore, DeleteReferenceStore, StartReferenceImportJob
- Sequenzspeicher und Lesesatz APIs — CreateSequenceStore DeleteSequenceStore, StartReadSetImportJob, StartReadSetActivationJob, StartReadSetExportJob
- Variantenspeicher APIs — CreateVariantStore, DeleteVariantStore, StartVariantImportJob, CancelVariantImportJob
- Speicher für Anmerkungen APIs — CreateAnnotationStore, DeleteAnotationStore, StartAnnotationImportJob, CancelAnnotationImportJob
- Arbeitsablauf, Ausführung und Gruppe ausführen APIs — CreateWorkflow DeleteWorkflow StartRun, CancelRun, DeleteRun, CreateRunGroup, DeleteRunGroup

Anzeigen von *AWS HealthOmics*-Metriken

CloudWatch AWS HealthOmics Metriken für können in der CloudWatch Konsole angezeigt werden.

Um Metriken anzuzeigen (CloudWatch Konsole)

1. Melden Sie sich bei der AWS-Managementkonsole an und öffnen Sie die CloudWatch -Konsole.
2. Wählen Sie Metriken, Alle Metriken und dann AWS/Usage aus.

3. Service filtern nach. AWS HealthOmics
4. Wählen Sie die Dimension, den Namen einer Metrik und schließlich Add to graph (Dem Diagramm hinzufügen) aus.
5. Wählen Sie einen Wert für den Datumsbereich aus. Die Anzahl der Kennzahlen für den ausgewählten Zeitraum wird im Diagramm angezeigt.

Einen Alarm erstellen mit CloudWatch

Ein CloudWatch Alarm überwacht eine einzelne Metrik über einen bestimmten Zeitraum und führt eine oder mehrere Aktionen aus: das Senden einer Benachrichtigung an ein Amazon Simple Notification Service (Amazon SNS) -Thema oder an eine Auto Scaling Scaling-Richtlinie. Die Aktion oder Aktionen basieren auf dem Wert der Metrik im Verhältnis zu einem bestimmten Schwellenwert über eine von Ihnen angegebene Anzahl von Zeiträumen. CloudWatch kann Ihnen auch eine Amazon SNS SNS-Nachricht senden, wenn sich der Status des Alarms ändert.

CloudWatch Alarme lösen nur dann Aktionen aus, wenn sich der Status ändert und für den von Ihnen angegebenen Zeitraum andauert.

So zeigen Sie Metriken an (Konsole) CloudWatch

1. Melden Sie sich bei der AWS-Managementkonsole an und öffnen Sie die CloudWatch -Konsole.
2. Wählen Sie Alarms und dann Create Alarm.
3. Wählen Sie AWS/Usage und anschließend mithilfe der Service-Dimension eine AWS HealthOmics Metrik aus.
4. Wählen Sie für Time Range den zu überwachenden Zeitbereich und dann Next.
5. Geben Sie Name (Name) und Description (Beschreibung) ein.
6. Wählen Sie für Whenever >= und geben Sie einen Maximalwert ein.
7. Wenn Sie eine E-Mail senden CloudWatch möchten, wenn der Alarmstatus erreicht ist, wählen Sie im Bereich Aktionen für Wann immer dieser Alarm die Option Status ist ALARM. Wählen Sie unter Benachrichtigung senden an eine Mailingliste aus oder wählen Sie Neue Liste und erstellen Sie eine neue Mailingliste.
8. Nutzen Sie die Alarmvorschau im Bereich Alarm Preview. Wenn Sie mit dem Alarm zufrieden sind, wählen Sie Create Alarm.

Überwachung HealthOmics mit CloudWatch Protokollen

HealthOmics generiert eine Vielzahl von Protokollen, die Ihnen helfen, Ihre Läufe zu verstehen und Fehler zu beheben. Protokolle sind an zwei Orten verfügbar: CloudWatch und in Amazon S3.

Standardmäßig ist bei Läufen die Protokollierung aktiviert. Sie können die Protokollierung für einen Lauf optional deaktivieren, indem Sie dies `LogLevel = OFF` in der startrun Anforderung festlegen.

> **Note**
>
> Für Service-Updates konfigurieren und überwachen Sie Ihr Personal Health Dashboard. Weitere Informationen zur Verwaltung des Dashboards finden Sie unter Erste Schritte mit Ihrem AWS Health Dashboard.

Themen

Protokolltypen für HealthOmics Workflows

HealthOmics bietet die folgenden Arten von Protokollen für Workflows:

- Engine-Logs — Die zugrunde liegenden Workflow-Engines (Nextflow, WDL und CWL) erstellen Engine-Logs für Läufe. Diese Protokolle können Ihnen bei der Behebung von Problemen mit der Workflow-Definition helfen.
- Ausführungsmanifestprotokolle — Diese Protokolle enthalten allgemeine Informationen zu jeder ausgeführten Aufgabe, z. B. Aufgabenstatus, Startzeit, Stoppzeit und Grund für den Fehler (falls die Aufgabe fehlgeschlagen ist).

  Run-Manifest-Protokolle enthalten auch Statistiken zur Ressourcennutzung, die hilfreich sein können, um Möglichkeiten zur Ressourcenoptimierung zu verstehen. Zu diesen Statistiken gehören:

  - CPU-Durchschnitt

  - CPUs maximal
  - CPUs reserviert
  - GPUs reserviert
  - memoryAverageGiB
  - memoryMaximumGiB
  - memoryReservedGiB
  - Laufende Sekunden
- Ausführungsprotokolle — Ausführungsprotokolle geben den allgemeinen Ausführungsstatus und die Uhrzeit an, zu der einzelne Aufgaben gestartet, ausgeführt, gestoppt und abgeschlossen werden. Ausführungsprotokolle geben Ihnen auch Einblick in die Schritte des Dateiimports und -exports.
- Aufgabenprotokolle — Aufgabenprotokolle enthalten detaillierte Protokollierungsinformationen zu einzelnen Aufgaben in Ihrem Lauf. Die Ausgaben in Ihrem Aufgabenprotokoll hängen von der Aufgabendefinition und davon ab, wo Sie Protokollanweisungen in Ihrem Code verwenden. Wenn Ihre Aufgabenprotokolle nicht den erforderlichen Einblick bieten, sollten Sie erwägen, Ihrer Aufgabendefinition zusätzliche Protokollanweisungen hinzuzufügen, um aussagekräftigere Aufgabenprotokolle zu erstellen.
- Cache-Logs ausführen — Run-Cache-Logs geben Aufschluss über den Gesamtstatus der ausgeführten Caches und das Zwischenspeichern von Aufgabenausgaben. Run-Cache-Logs geben Ihnen Einblick in Cache-Treffer und -Fehlschläge bei jeder Ausführung, bei der Caching verwendet wird.
- outputs.json — Für WDL- und CWL-Workflows wird nach Abschluss der Ausführung eine von HealthOmics der Engine generierte Datei mit dem Namen an Ihren Amazon S3 `outputs.json` S3-Bucket gesendet. Diese Datei enthält eine Liste und eine Übersicht aller Ausgaben für den Lauf.

Meldet sich an CloudWatch

CloudWatch generiert Workflow-Protokolle für fehlgeschlagene und erfolgreiche Läufe. Alle Protokolle sind für fehlgeschlagene und erfolgreiche Läufe verfügbar, mit Ausnahme von Engine-Protokollen, die nur für fehlgeschlagene Läufe verfügbar sind.

Sie finden die CloudWatch Workflow-Protokolle in der folgenden Protokollgruppe:`/aws/omics/WorkflowLog`. Außerdem stellt die Ausgabe des API-Vorgangs „Get-Run“ den CloudWatch Protokollstream ARNs für die Modul- und Ausführungsprotokolle bereit.

In der Standardeinstellung werden die CloudWatch Protokolle auf unbestimmte Zeit AWS aufbewahrt. Sie können die Aufbewahrungsrichtlinie für die Protokollgruppe anpassen, um einen Aufbewahrungszeitraum zwischen 10 Jahren und einem Tag festzulegen.

Die folgende Tabelle enthält eine Zusammenfassung der CloudWatch Anmeldungen HealthOmics. Alle Workflow-Protokolle sind für erfolgreiche und fehlgeschlagene Ausführungen verfügbar, mit Ausnahme von Engine-Protokollen, die nur für fehlgeschlagene Läufe verfügbar sind.

Protokollnamen	Verfügbar in CloudWatch Protokollen	Wann ist das Protokoll verfügbar	Format des Protokollstreams
Motorprotokolle	Ja, für fehlgeschlagene Läufe	Nach Abschluss des Laufs	run/ /engine *runID*
Führen Sie Manifestprotokolle aus	Ja	Nach Abschluss der Ausführung	manifest/ run//*runIDrunUUID*
Protokolle ausführen	Ja	In Echtzeit	laufen/ *runID*
Aufgabenprotokolle	Ja	In Echtzeit	run/ /task/ *runID taskID*
Cache-Protokolle ausführen	Ja	In Echtzeit	RunCache/ /*runCacheId runCacheUUID*
outputs.json (WDL und CWL)	Nein	–	–

Loggt sich in Amazon S3 ein

Nur die Engine-Logs und die `outputs.json` Datei werden an Amazon S3 übermittelt.

Nach Abschluss eines Laufs werden die Engine-Protokolle an Ihren S3-Bucket übermittelt und sind unbegrenzt verfügbar, bis Sie sie löschen. Diese Protokolle befinden sich im Protokollverzeichnis der S3-Ausgabe-URI, die Sie für den Workflow angegeben haben.

Der Pfad zum Protokollverzeichnis hat das folgende Format:`s3://{user_provided_path}/logs/`.

Die folgende Tabelle enthält eine Zusammenfassung der in Ihrem Amazon S3 S3-Bucket verfügbaren HealthOmics Protokolle.

Protokollnamen	Verfügbar in Amazon S3	Wann ist das Protokoll verfügbar	Stream-Pfad protokollieren
Engine-Protokolle	Ja	Nach Abschluss des Laufs	s3://*user_provided_path* /logs/engine.log
outputs.json (WDL und CWL)	Ja	Nach Abschluss der Ausführung	s3://*user_provided_path* /*runID*/*runUUID*/logs/outputs.json
Führen Sie Manifestprotokolle, Ausführungsprotokolle und Aufgabenprotokolle aus	Nein	–	–

Interaktive CloudWatch Protokolle in der CLI

Sie können die CloudWatch Protokolle interaktiv mit dem Befehl Live Tail im interaktiven Modus anzeigen. Sie können den Fortschritt des Laufs in Echtzeit verfolgen und bis zu 5 Schlüsselwörter definieren, die in den Protokollen hervorgehoben werden sollen:

```
aws logs start-live-tail  \
  --mode interactive  \
  --log-group-identifiers arn:aws:logs:region:account-ID:log-group:/aws/omics/
WorkflowLog
```

Weitere Informationen finden Sie unter Start live tail in der AWS CLI Befehlsreferenz.

Von der Konsole aus auf CloudWatch Protokolle zugreifen

Um auf die Protokolle für einen Lauf zuzugreifen, können Sie über die Seite mit den Ausführungsdetails in der HealthOmics Konsole direkt auf diese Protokolle zugreifen.

1. Öffnen Sie die HealthOmics -Konsole.
2. Öffnen Sie bei Bedarf den linken Navigationsbereich (±1). Wählen Sie Läufe.
3. Wählen Sie den Lauf aus der Tabelle Runs aus.
4. Auf der Seite mit den Ausführungsdetails können Sie eine der folgenden Aktionen auswählen:

   a. Wählen Sie unter „Ausführungsübersicht“ die Option „Ausführungsprotokolle anzeigen“ aus. Die Konsole öffnet die Ausführungsprotokolle in der CloudWatch Konsole.

   b. Wählen Sie unter Zusammenfassung ausführen die Option Protokolle in Amazon S3 anzeigen aus. Die Konsole öffnet den Logs-Ordner in der Amazon S3 S3-Konsole.

   c. Wählen Sie unter Aufgaben ausführen die Option Protokolle anzeigen, Ausführungsprotokolle anzeigen oder Ausführungsmanifestprotokolle anzeigen für eine Aufgabe aus. Die Konsole öffnet die Protokolle in der CloudWatch Konsole.

Sie können auch von der CloudWatch Konsole aus zu den Protokollen navigieren:

1. Öffnen Sie die CloudWatch Konsole https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im linken Menü Protokollgruppen aus.
3. Wählen Sie die `/aws/omics/WorkflowLog`-Gruppe aus.

   Wenn die Liste der Protokollgruppen lang ist, können Sie Omics in das Suchtextfeld eingeben, um die Liste einzugrenzen.
4. Wenn die Seite mit den Protokollgruppendetails geöffnet wird, wählen Sie den Protokollstream aus, den Sie anzeigen möchten. Die Konsole zeigt die Ereignisse für diesen Protokollstream an.

Protokollieren von AWS HealthOmics API-Aufrufen mit AWS CloudTrail

AWS HealthOmics ist in einen Dienst integriert AWS CloudTrail, der eine Aufzeichnung der Aktionen bereitstellt, die von einem Benutzer, einer Rolle oder einem AWS Dienst in ausgeführt wurden HealthOmics. CloudTrail erfasst alle API-Aufrufe HealthOmics als Ereignisse. Zu den erfassten

Aufrufen gehören Aufrufe von der HealthOmics Konsole und Codeaufrufen für die HealthOmics API-Operationen. Wenn Sie einen Trail erstellen, können Sie die kontinuierliche Bereitstellung von CloudTrail Ereignissen an einen Amazon S3 S3-Bucket aktivieren, einschließlich Ereignissen für HealthOmics. Wenn Sie keinen Trail konfigurieren, können Sie die neuesten Ereignisse trotzdem in der CloudTrail Konsole im Ereignisverlauf anzeigen. Anhand der von gesammelten Informationen können Sie die Anfrage ermitteln CloudTrail, an die die Anfrage gestellt wurde HealthOmics, die IP-Adresse, von der aus die Anfrage gestellt wurde, wer die Anfrage gestellt hat, wann sie gestellt wurde, und weitere Details.

Weitere Informationen CloudTrail dazu finden Sie im AWS CloudTrail Benutzerhandbuch.

HealthOmics Informationen in CloudTrail

CloudTrail ist auf Ihrem aktiviert AWS-Konto , wenn Sie das Konto erstellen. Wenn eine Aktivität in stattfindet HealthOmics, wird diese Aktivität zusammen mit anderen AWS Serviceereignissen in der CloudTrail Ereignishistorie in einem Ereignis aufgezeichnet. Sie können aktuelle Ereignisse in Ihrem anzeigen, suchen und herunterladen AWS-Konto. Weitere Informationen finden Sie unter Ereignisse mit dem CloudTrail Ereignisverlauf anzeigen.

Für eine fortlaufende Aufzeichnung der Ereignisse in Ihrem AWS-Konto, einschließlich der Ereignisse für HealthOmics, erstellen Sie einen Trail. Ein Trail ermöglicht CloudTrail die Übermittlung von Protokolldateien an einen Amazon S3 S3-Bucket. Wenn Sie einen Trail in der Konsole anlegen, gilt dieser für alle AWS-Regionen-Regionen. Der Trail protokolliert Ereignisse aus allen Regionen der AWS Partition und übermittelt die Protokolldateien an den von Ihnen angegebenen Amazon S3 S3-Bucket. Darüber hinaus können Sie andere AWS Dienste konfigurieren, um die in den CloudTrail Protokollen gesammelten Ereignisdaten weiter zu analysieren und darauf zu reagieren. Weitere Informationen finden Sie hier:

- Übersicht zum Erstellen eines Trails
- CloudTrail unterstützte Dienste und Integrationen
- Konfiguration von Amazon SNS SNS-Benachrichtigungen für CloudTrail
- Empfangen von CloudTrail Protokolldateien aus mehreren Regionen und Empfangen von CloudTrail Protokolldateien von mehreren Konten

Alle HealthOmics Aktionen werden von der AWS HealthOmics API-Referenz protokolliert CloudTrail und sind in dieser dokumentiert. Beispielsweise generieren Aufrufe von

`StartVariantImportJob` und `CreateWorkflow` Aktionen Einträge in den CloudTrail Protokolldateien. `CreateReferenceeStore`

Jeder Ereignis- oder Protokolleintrag enthält Informationen zu dem Benutzer, der die Anforderung generiert hat. Die Identitätsinformationen unterstützen Sie bei der Ermittlung der folgenden Punkte:

- Ob die Anfrage mit IAM-Benutzeranmeldedaten gestellt wurde.
- Gibt an, ob die Anforderung mit temporären Sicherheitsanmeldeinformationen für eine Rolle oder einen Verbundbenutzer gesendet wurde.
- Ob die Anfrage von einem anderen AWS Dienst gestellt wurde.

Weitere Informationen finden Sie unter CloudTrail -Element userIdentity.

HealthOmics Logdateieinträge verstehen

Ein Trail ist eine Konfiguration, die die Übertragung von Ereignissen als Protokolldateien an einen von Ihnen angegebenen Amazon S3 S3-Bucket ermöglicht. CloudTrail Protokolldateien enthalten einen oder mehrere Protokolleinträge. Ein Ereignis stellt eine einzelne Anforderung aus einer beliebigen Quelle dar und enthält Informationen über die angeforderte Aktion, Datum und Uhrzeit der Aktion, Anforderungsparameter usw. CloudTrail Protokolldateien sind kein geordneter Stack-Trace der öffentlichen API-Aufrufe, sodass sie nicht in einer bestimmten Reihenfolge angezeigt werden.

Das folgende Beispiel zeigt einen CloudTrail Protokolleintrag, der die CreateWorkflow Aktion demonstriert.

```
{
    "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"
}
```


Verwenden EventBridge mit AWS HealthOmics

HealthOmics sendet Ereignisse an Amazon, EventBridge wenn sich der Status von Ressourcen ändert. Zu den Ressourcen gehören Importaufträge, Exportaufträge, gemeinsam genutzte Ressourcen, Workflows, Aufgaben und Läufe. Für jeden Ressourcentyp gibt es eine Liste von Statusänderungen, die ein Ereignis auslösen.

Ein Eventbus ist ein Router, der Ereignisse empfängt und sie an Ziele weiterleitet. Ihr Konto enthält einen Standard-Event-Bus, der automatisch Ereignisse von AWS Diensten empfängt. Sie können zusätzliche benutzerdefinierte Event-Busse erstellen.

Sie erstellen EventBridge Regeln, um die Aktionen festzulegen, die ausgeführt werden sollen, wenn der Event-Bus Ereignisse empfängt. Sie können beispielsweise eine Regel erstellen, die Sie über Statusänderungen für eine Ressource informiert.

Zu den gängigen Szenarien für die Verwendung von Ereignissen gehören:

- Um zu überwachen, wann ein Benutzer eine Ressource mit Ihnen teilt oder die Freigabe widerruft.
- Um zu überwachen, ob eine Ausführung fehlschlägt oder erfolgreich abgeschlossen wird.

Weitere Informationen zur Verwendung EventBridge finden Sie unter Was ist Amazon EventBridge?

Themen

Eingerichtet EventBridge für HealthOmics

Bevor Sie nach EventBridge Ereignissen suchen können, müssen Sie einen EventBridge Bus erstellen und Regeln für die Ereignisse erstellen, die für Sie von Interesse sind.

Konfigurieren Sie einen EventBridge Bus

Sie können den Standard-Event-Bus für Ihren verwenden AWS-Konto oder einen benutzerdefinierten Event-Bus konfigurieren. Gehen Sie folgendermaßen vor, um einen benutzerdefinierten Event-Bus zu konfigurieren:

1. Öffnen Sie die EventBridge Konsole: https://console.aws.amazon.com/events/.
2. Wählen Sie in der linken Navigationsleiste Event Buses aus.
3. Wählen Sie Create event bus (Ereignisbus erstellen) aus.
4. Geben Sie im Formular Eventbus erstellen einen Namen für den Bus ein.

5. Wählen Sie Create, um den Bus zu erstellen.

Erstellen Sie eine EventBridge Regel

Das folgende Verfahren zeigt, wie Sie eine einfache Regel erstellen. Weitere Informationen zu Regeln finden Sie unter Regeln in EventBridge.

1. Öffnen Sie die EventBridge Konsole: https://console.aws.amazon.com/events/.
2. Klicken Sie im linken Navigationsbereich auf die Option Regeln.
3. Wählen Sie Regel erstellen aus. Die Konsole öffnet das Formular Regel erstellen.
4. Geben Sie im Feld Regeldetails definieren einen Namen für die Regel ein.
   - Geben Sie unter Name einen Namen für den Bus ein.
   - Wählen Sie für Event-Bus den Bus für diese Regel aus.
   - Wählen Sie Weiter aus.
5. Wählen Sie unter Ereignismuster erstellen unter Ereignisquelle AWS-Ereignisse oder EventBridge Partnerereignisse aus.
6. Scrollen Sie nach unten zu Ereignismuster.
   a. Wählen Sie als Ereignisquelle AWS-Services aus.
   b. Geben Sie für AWS-Service omics in den Textfilter ein und wählen Sie AWS HealthOmicsals Service aus.
   c. Wählen Sie als Ereignistyp das Ereignis von Interesse (oder Alle Ereignisse) aus.
   d. Wählen Sie Weiter aus.
7. Wählen Sie unter Ziel (e) auswählen ein Ziel für das Ereignis aus. Wählen Sie beispielsweise den AWS-Service, die gewählte CloudWatch Protokollgruppe und konfigurieren Sie eine Protokollgruppe.

   Für viele Zieltypen benötigt EventBridge die Berechtigung zum Senden von Ereignissen an das Ziel. Die Konsole erstellt diese Berechtigungen für Sie.
8. (Optional) Ordnen Sie unter Tags konfigurieren Tags der Regel zu.
9. Überprüfen Sie unter Überprüfen und aktualisieren die Konfiguration und wählen Sie Regel erstellen aus.

EventBridge Ereignisse in HealthOmics

In der folgenden Tabelle sind die Ereignisse aufgeführt, an die HealthOmics gesendet wird EventBridge, sowie die Liste der möglichen Statuswerte für das Ereignis.

Ereignisname	Mögliche Statuswerte
Änderung des Auftragsstatus „Anmerkung importieren“	Eingereicht, in Bearbeitung, storniert, abgeschlossen, fehlgeschlagen oder mit Fehlern abgeschlossen
Änderung des Status von Annotation Store Teilen	Ausstehend, aktivierend, aktiv, löschend, gelöscht, fehlgeschlagen
Änderung des Status des Annotation Stores	Fehler beim Erstellen, Erstellen, Aktualisieren, Aktualisieren, Löschen, Löschen oder Erstellen
Lesen Sie „Änderung des Status des Aktivierungsauftrags festlegen“	Eingereicht, in Bearbeitung, abgeschlossen, fehlgeschlagen oder mit Fehlern abgeschlossen
Lesen Sie „Änderung des Exportauftragsstatus festlegen“	Eingereicht, in Bearbeitung, abgeschlossen, fehlgeschlagen oder mit Fehlern abgeschlossen
Lesen Sie „Änderung des Importauftrags festlegen“	Eingereicht, in Bearbeitung, abgeschlossen, fehlgeschlagen oder mit Fehlern abgeschlossen
Lesen Sie „Statusänderung festlegen“	Upload wird verarbeitet, Upload ist fehlgeschlagen, aktiv, archiviert, aktiviert oder gelöscht
Änderung des Status des Referenzimport-Jobs	Eingereicht, in Bearbeitung, abgeschlossen, fehlgeschlagen oder mit Fehlern abgeschlossen
Änderung des Referenzstatus	Aktiv oder gelöscht
Änderung des Status des Referenzspeichers	Erstellt, aktualisiert, aktiv oder gelöscht

Ereignisname	Mögliche Statuswerte
Statusänderung ausführen	Ausstehend, wird gestartet, läuft, wird beendet, abgeschlossen, gelöscht, fehlgeschlagen oder storniert
Änderung des Status des Sequenzspeichers	Erstellt, aktualisiert, aktiv oder gelöscht
Änderung des Aufgabenstatus	Ausstehend, wird gestartet, läuft, wird beendet, abgeschlossen, gelöscht, fehlgeschlagen oder storniert
Änderung des Jobstatus beim Variantenimport	Eingereicht, in Bearbeitung, storniert, abgeschlossen, fehlgeschlagen oder mit Fehlern abgeschlossen
Änderung des Status „Variant Store Share“	Ausstehend, aktivierend, aktiv, löschend, gelöscht, fehlgeschlagen
Änderung des Status des Variantenspeichers	Fehler beim Erstellen, Erstellen, Aktualisieren, Aktualisieren, Löschen, Löschen oder Erstellen
Änderung des Workflow-Freigabestatus	Ausstehend, aktivierend, aktiv, löschend, gelöscht, fehlgeschlagen
Änderung des Workflow-Status	Erstellung erfolgreich, Erstellung fehlgesch lagen, Löschvorgang erfolgreich oder Löschfehl er

Struktur von Ereignismeldungen

HealthOmics bietet optimale Zustellung für das Senden von Meldungen zu Zustandsänderungsereignissen an EventBridge. Das Ereignis ist ein Objekt mit JSON-Struktur, das auch Metadatendetails enthält. Sie können die Metadaten als Eingabe verwenden, um entweder das Ereignis neu zu erstellen oder weitere Informationen zu erhalten. Ereignisse umfassen die folgenden Felder:

- `version`— Derzeit 0 (Null) für alle Ereignisse.

- `id`— Eine Version 4-UUID, die für jedes Ereignis generiert wurde.
- `detail-type`— Die Art des Ereignisses, das gesendet wird.
- `account`— Die 12-stellige AWS-Konto ID des Bucket-Besitzers.
- `source`— Identifiziert den Dienst, der das Ereignis generiert hat.
- `time`— Der Zeitpunkt, zu dem das Ereignis eingetreten ist.
- `region`— Identifiziert den AWS-Region des Buckets.
- `resources`— Ein JSON-Array, das den Amazon-Ressourcennamen (ARN) des Buckets enthält.
- `detail`— Ein JSON-Objekt, das Informationen über das Ereignis enthält.

Run-Ereignisse umfassen die folgenden Felder:

- `uuid`— Die allgemein eindeutige Kennung für den Lauf.
- `workflowId`— Workflow-ID des Workflows, der diesem Lauf zugeordnet ist.
- `workflowName`— Name des Workflows, der diesem Lauf zugeordnet ist..
- `runId`— Kennung ausführen.
- `runName`— Name der Ausführung.
- `runOutputUri`— Die URI, für die der Lauf seine Ausgabedaten schreiben wird.

Beispiele für Ereignisnachrichten

Das folgende Beispiel ist ein Ereignis für eine Änderung des Ausführungsstatus, in dem die zusätzlichen Felder angezeigt werden.

```
{
    "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"
    }
}
```

Das folgende Beispiel ist ein Ereignis für eine Änderung des Aufgabenstatus.

```
{
    "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"
    }
}
```

Das Folgende ist ein Beispiel für ein Ereignis bei einer Änderung des Lesesatzstatus.

```
{
  "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"
  }
}
```

Ein ähnliches Ereignis wird für einen Variantenspeicher-Importjob erstellt.

```
{
    "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"
    }
}
```

Das Folgende ist ein Ereignis für eine Änderung des Importauftragsstatus.

```
{
    "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"
    }
}
```

Fehlerbehebung

Die folgenden Themen können Ihnen bei der Behebung von Problemen helfen, die bei der Verwendung von HealthOmics Workflows und Datenspeichern auftreten.

Themen

Problembehandlung bei Workflows

Themen

Wie behebe ich einen fehlgeschlagenen Lauf?

Verwenden Sie den GetRunAPI-Vorgang, um den Grund für den Fehler abzurufen. Weitere Informationen finden Sie unter Gründe für Fehler beim Ausführen.

Wie behebe ich eine fehlgeschlagene Aufgabe?

Sehen Sie sich den Fehlercode in der Fehlermeldung für die Aufgabe an, um den Fehler zu verstehen. Sehen Sie sich die Aufgabenanmeldungen an CloudWatch , um detaillierte Protokollmeldungen für die Aufgabe zu sehen. Wenn Sie keine detaillierten Protokollmeldungen erhalten, können Sie Ihren Workflow überarbeiten, um zusätzliche Protokollanweisungen auszugeben. Weitere Informationen finden Sie unter Überwachung HealthOmics mit CloudWatch Protokollen.

Wo finde ich die Engine-Protokolle für erfolgreich abgeschlossene Läufe?

HealthOmics veröffentlicht nur Protokolle CloudWatch für fehlgeschlagene Läufe. Wenn ein Lauf erfolgreich abgeschlossen wurde, werden die HealthOmics Engine-Protokolle an Ihren Amazon S3 S3-Bucket gesendet. Weitere Informationen finden Sie unter Loggt sich in Amazon S3 ein.

Wie kann ich die Größe der Eingabeparameter für einen Workflow reduzieren?

Sie können bis zu 50 KB an Eingabeparametern für einen Workflow angeben. Sie können Verzeichnisimporte oder Musterblätter verwenden, um diese Größenbeschränkung einzuhalten. Weitere Informationen finden Sie unter Größe der Ausführungsparameter verwalten.

Warum ist mein Lauf nicht abgeschlossen?

Wenn es Probleme mit Ihrem Code gibt und die Prozesse nicht ordnungsgemäß beendet wurden, reagiert Ihr Lauf möglicherweise nicht mehr oder „blockiert“. Weitere Informationen darüber, wie Sie nicht reagierende Läufe verhindern und catch können, finden Sie unterHinweise für nicht reagierende Läufe.

Behebung von Problemen beim Zwischenspeichern von Anrufen

Die folgenden Themen können Ihnen bei der Behebung von Problemen helfen, die beim Zwischenspeichern von Anrufen auftreten.

Themen

- Warum wird mein Lauf nicht im Cache gespeichert?
- Warum verwendet eine Aufgabe den Cache-Eintrag nicht?
- Warum ist das Caching von Aufrufen für eine Aufgabe deaktiviert?

Warum wird mein Lauf nicht im Cache gespeichert?

1. Vergewissern Sie sich, dass der Lauf für die Verwendung eines Caches konfiguriert ist, indem Sie das Feld CacheID in der Antwort auf den GetRun API-Vorgang überprüfen. Führen Sie mit der CLI diesen Befehl aus:`aws omics get-run –id <run_id>`.
2. Wenn die Ausführung erfolgreich war, überprüfen Sie, ob das in der GetRun Antwort zurückgegebene Cache-Verhalten CACHE_ALWAYS lautet. Wenn das Cache-Verhalten auf

CACHE_ON_FAILURE gesetzt ist, werden Läufe nur dann im Cache gespeichert, wenn sie fehlschlagen.

Warum verwendet eine Aufgabe den Cache-Eintrag nicht?

<cache_id><cache_uuid>Öffnen Sie in der `/aws/omics/WorkflowLog` CloudWatch Protokollgruppe den Protokollstream für den Run-Cache: RunCache//.

1. Vergewissern Sie sich, dass bei einem vorherigen Lauf ein Cache-Eintrag für die Aufgabe erstellt wurde, von der Sie erwartet haben, dass sie zwischengespeichert wird. Läufe, die im Cache gespeichert wurden, werden mit der Protokollmeldung CACHE_ENTRY_CREATED aufgezeichnet.
2. Suchen Sie das CACHE_MISS-Protokoll für die Aufgabe und führen Sie es aus, wenn es abgeschlossen ist. Wenn kein Protokolleintrag vorhanden ist, überprüfen Sie, ob der Lauf für die Verwendung des Caches konfiguriert wurde.
3. Wenn ein Cache-Eintrag erstellt wurde, stellen Sie sicher CPUs, dass der Speicher GPUs - und der Container-Digest für beide Aufgaben identisch sind. Der Task-ARN für die Aufgabe, die den Cache-Eintrag erstellt hat, ist in der Protokollnachricht enthalten.
4. Wenn die Rechenanforderungen für beide Aufgaben übereinstimmen, stellen Sie sicher, dass sich die Eingaben zwischen den Aufgaben nicht geändert haben. Öffnen Sie dazu die Engine-Logs. Wenn der Lauf den Status FAILED hat, befinden sich die Protokolle in Cloudwatch Log Group/aws/omics/WorkflowLog. Andernfalls befinden sich die Engine-Logs im Ausgabeverzeichnis des Laufs.

Warum ist das Caching von Aufrufen für eine Aufgabe deaktiviert?

Überprüfen Sie mithilfe der Funktionen der Workflow-Engine, ob die Aufgabe so konfiguriert ist, dass das Caching deaktiviert wird:

- Für WDL-Workflows: Prüfen Sie, ob für die Aufgabe `true` im Metabereich die Option Volatile aktiviert ist
- Für Nextflow-Workflows: Prüfen Sie, ob für die Aufgabe die Cache-Direktive auf gesetzt ist `false`
- Für CWL-Workflows: Prüfen Sie, ob für die Aufgabe EnableReuse für die Funktion auf gesetzt ist `false` WorkReuse

Problembehandlung bei Datenspeichern

Themen

Warum schlägt S3 auf meinem Leseset GetObject fehl?

In den meisten Fällen ist der Fehler auf eine fehlende Erlaubnis zurückzuführen. Die Leseberechtigung für den Sequenzspeicher S3 ist eine bidirektionale Konfiguration, bei der sowohl die S3-Zugriffsrichtlinie für den Sequenzspeicher als auch der IAM-Prinzipal über eine Richtlinie verfügen muss, die den Zugriff ermöglicht. Weitere Informationen zu den Richtlinienanforderungen finden Sie unter. Berechtigungen für den Datenzugriff mit Amazon S3 URIs Vergewissern Sie sich, dass die folgenden Konfigurationen vorhanden sind:

- Die S3-Zugriffsrichtlinie für den Sequenzspeicher hat den Zugriff auf den IAM-Prinzipal oder das Stammverzeichnis des Prinzipalkontos ausdrücklich zugelassen.
- Vergewissern Sie sich, dass der IAM-Prinzipal über eine Richtlinie verfügt, die ausdrücklich die Erlaubnis für die Ressource, auf die zugegriffen wird, vorsieht. Beachten Sie, dass die IAM-Prinzipalrichtlinie bei der Definition von Berechtigungen den Access Point-ARN und nicht den auf dem Access Point-Alias basierenden Pfad verwenden muss und dass der ARN in diesem Zustand ist und nicht zur Angabe einer Ressource verwendet wird.
- Wenn Ihr Shop einen vom Kunden verwalteten Schlüssel (CMK-KMS) verwendet, stellen Sie sicher, dass der IAM-Prinzipal über kms: Entschlüsselungsberechtigungen für den Schlüssel verfügt. Informationen zur Konfiguration der kontoübergreifenden Nutzung finden Sie im Leitfaden für kontoübergreifenden Zugriff auf KMS.

Wenn Sie über eine Richtlinie verfügen, die tagbasierte Zugriffskontrollen verwendet, stellen Sie Folgendes sicher:

- Stellen Sie sicher, dass der Sequenzspeicher die Synchronisierung der Tags abgeschlossen hat. Dazu muss der Status des Speichers lauten active und nichtupdating.
- Stellen Sie sicher, dass der Tag-Schlüssel oder der Schlüsselwert im Lesesatz und in der Richtlinie keine Tippfehler enthalten.

Warum kann ich meinen Annotationsspeicher oder Variantenspeicher in Athena nicht sehen?

Stellen Sie in Lake Formation sicher, dass Sie einen Ressourcenlink erstellen, der auf dem Shop basiert, der mit Ihnen geteilt wurde. Sobald Sie einen Ressourcenlink erstellt haben, auf den Sie zugreifen dürfen, sollte der Shop in Athena sichtbar sein. Weitere Informationen finden Sie unter Konfiguration von Lake Formation für die Verwendung HealthOmics.

Warum kann ich in Athena nicht auf meinen Datenspeicher zugreifen?

Wenn Ihr Annotations- oder Variantenspeicher sichtbar ist, Sie aber eine Fehlermeldung erhalten, dass der Zugriff verweigert wurde, überprüfen Sie, welche Version der Abfrageengine Sie verwenden. Es werden nur Abfragen unterstützt, die mit Engine-Version 3 ausgeführt werden. Weitere Informationen zu den Versionen der Athena-Abfrage-Engine finden Sie in der Amazon Athena Athena-Dokumentation.

Fehlerbehebung mit Amazon Q CLI

Amazon Q CLI kann Ihnen helfen, Ihren Fehlerbehebungsprozess zu optimieren, indem es:

- Analysieren von Workflow-Ausführungen und Debuggen von Aufgabenfehlern
- Erfassung relevanter Protokolle und Fehlermeldungen
- Erstellen von AWS Support-Fällen mit allen erforderlichen Debugging-Protokollen im Anhang
- Löscht personenbezogene Daten (PII) aus den an den Support übermittelten Informationen AWS

Weitere Informationen zur Verwendung von Amazon Q CLI AWS HealthOmics zur Fehlerbehebung und Erstellung von Supportfällen finden Sie im HealthOmics Agentic Generative AI-Tutorial unter. GitHub

> **Warning**
>
> Wenn Sie mit Amazon Q CLI arbeiten, überprüfen Sie alle generierten Inhalte und vorgeschlagenen Maßnahmen, bevor Sie fortfahren. Geben Sie Feedback, um die Antwortqualität zu verbessern und die Anforderungen Ihres Workflows zu erfüllen. Weitere Informationen finden Sie unter Sicherheitsüberlegungen und bewährte Methoden für Amazon Q.

Kontingente für AWS HealthOmics

AWS füllt Ihr Konto mit Standardwerten für die HealthOmics Kontingente auf. Sofern nicht anders angegeben, entspricht jeder Kontingentwert dem Höchstwert pro Region.

> **Important**
>
> Sie können eine Erhöhung der meisten Service- und API-Kontingente beantragen. Weitere Informationen finden Sie in den folgenden Themen.

Themen

HealthOmics Servicekontingenten

In der folgenden Tabelle sind die HealthOmics Servicekontingenten zusammen mit ihren Standardwerten aufgeführt. Um die aktuellen Kontingente für jede Region einzusehen, öffnen Sie die Konsole Service Quotas.

> **Important**
>
> Sie können über die Service Quotas-Konsole eine Erhöhung eines einstellbaren Kontingents beantragen.

Weitere Informationen zu Service Quotas finden Sie unter Beantragung einer Kontingenterhöhung im Servicekontingents-Benutzerhandbuch. Verwenden Sie für ein Kontingent, das in der Service Quotas Quota-Konsole nicht verfügbar ist, das Formular zur Erhöhung des Kontingents.

Name	Standard	Anpas	Description
Analytics — Maximale Anzahl an Speichern von Anmerkungen	Jede unterstützte Region: 10	Yes (Ja)	Die maximale Anzahl von Annotationsspeiche

Name	Standard	Anpas	Description
			rn in der aktuellen AWS Region
Analytics — Maximale Anzahl gleichzeitiger Importjobs für Varianten- oder Annotationsspeicher	Jede unterstützte Region: 5	Yes (Ja)	Die maximale Anzahl gleichzeitiger Importjobs in der aktuellen Region AWS
Analytics — Maximale Anzahl von Dateien pro Variante für den Importjob	Jede unterstützte Region: 1 000	Ja	Die maximale Anzahl von Dateien pro Varianten importjob in der aktuellen AWS Region
Analytik — Maximale Anzahl der Anteile pro Annotationsspeicher	Jede unterstützte Region: 10	Yes (Ja)	Die maximale Anzahl von Shares pro Annotationsspeicher in der aktuellen AWS Region
Analytics — Maximale Anzahl von Shares pro Variantenspeicher	Jede unterstützte Region: 10	Yes (Ja)	Die maximale Anzahl von Shares pro Varianten speicher in der aktuellen AWS Region
Analytics — Maximale Größe jeder Datei in einem Variantenimport-Job	Jede unterstützte Region: 20 Gigabyte	Ja	Die maximale Größe einer Datei in einem Variantenimportjob in der aktuellen AWS Region
Analytik — Maximale Größe jeder Datei in einem Importjob für Anmerkungen	Jede unterstützte Region: 20 Gigabyte	Ja	Die maximale Größe einer Datei in einem Importauftrag für Anmerkungen in der aktuellen AWS Region

Name	Standard	Anpas	Description
Analytik — Maximale Anzahl an Variantenspeichern	Jede unterstützte Region: 10	Yes (Ja)	Die maximale Anzahl von Variantengeschäften in der aktuellen AWS Region
Analytics — Maximale Anzahl an Versionen pro Annotationsspeicher	Jede unterstützte Region: 10	Yes (Ja)	Die maximale Anzahl von Versionen pro Annotationsspeicher in der aktuellen AWS Region
Konfigurationen — Maximale Anzahl an Konfigurationen	Jede unterstützte Region: 10	Yes (Ja)	Die maximale Anzahl von Konfigurationen in der aktuellen AWS Region.
Speicher — Maximale Anzahl gleichzeitiger Read-Set-Aktivierungsjobs	Jede unterstützte Region: 25	Ja	Die maximale Anzahl gleichzeitiger Readset-Aktivierungsjobs in der aktuellen Region AWS
Speicher — Maximale Anzahl gleichzeitiger Sequenz- und Referenzspeicher-Exportaufträge	Jede unterstützte Region: 5	Yes (Ja)	Die maximale Anzahl gleichzeitiger Exportaufträge aus einer Sequenz oder einem Referenzspeicher in der aktuellen Region AWS
Speicher — Maximale Anzahl gleichzeitiger Sequenz- oder Referenzspeicher-Importaufträge	Jede unterstützte Region: 5	Yes (Ja)	Die maximale Anzahl gleichzeitiger Importaufträge für eine Sequenz oder einen Referenzspeicher in der aktuellen Region AWS

Name	Standard	Anpas	Description
Speicher — Maximale Anzahl von Lesesätzen pro Sequenzspeicher	Jede unterstützte Region: 1 000 000	Ja	Die maximale Anzahl von Lesesätzen in einem Sequenzspeicher in der aktuellen AWS Region
Speicher — Maximale Anzahl von Referenzen pro Referenzspeicher	Jede unterstützte Region: 50	Ja	Die maximale Anzahl von Referenzen in einem Referenzspeicher in der aktuellen AWS Region
Speicher — Maximale Anzahl von Sequenzspeichern	Jede unterstützte Region: 20	Ja	Die maximale Anzahl von Sequenzspeichern in der aktuellen AWS Region
Workflows — Höchstzahl aktiv GPUs	Jede unterstützte Region: 12	Ja	Die maximale Anzahl gleichzeitig aktiver Benutzer GPUs in der aktuellen AWS Region. In us-east-1 und us-west-2 werden Quotenerhöhungsanträge für Werte bis zu 500 automatisch genehmigt.
Workflows — Maximale Anzahl gleichzeitiger aktiver Läufe mithilfe von dynamischem Run-Speicher	Jede unterstützte Region: 50	Ja	Die maximale Anzahl aktiver Läufe mit dynamischem Run-Speicher in der aktuellen AWS Region. Anfragen zur Erhöhung des Kontingents für Werte bis zu 200 werden automatisch genehmigt.

Name	Standard	Anpas	Description
Workflows — Maximale Anzahl gleichzeitiger aktiver Läufe unter Verwendung von statischem Run-Speicher	Jede unterstützte Region: 10	Yes (Ja)	Die maximale Anzahl aktiver Läufe mit statischem Run-Speicher in der aktuellen AWS Region. Anfragen zur Erhöhung des Kontingents für Werte bis zu 50 werden automatisch genehmigt.
Workflows — Maximale Anzahl gleichzeitiger Aufgaben pro Lauf	Jede unterstützte Region: 25	Ja	Die maximale Anzahl gleichzeitiger Aufgaben in jeder Ausführung in der aktuellen AWS Region. In us-east-1 und us-west-2 werden Quotenerhöhungsanträge für Werte bis zu 100 automatisch genehmigt.
Workflows — Maximale Ausführungsdauer	Jede unterstützte Region: 604.800 Sekunden	Ja	Die maximale Workflow-Ausführungsdauer in der aktuellen AWS Region.
Workflows — Maximale Anzahl von Durchläufen (aktiv oder inaktiv)	Jede unterstützte Region: 100 000	Ja	Die maximale Anzahl von Läufen (aktiv oder inaktiv) in der aktuellen AWS Region.
Workflows — Maximale Anzahl von Anteilen pro Workflow	Jede unterstützte Region: 100	Yes (Ja)	Die maximale Anzahl von Shares pro Workflow in der aktuellen AWS Region

Name	Standard	Anpas	Description
Workflows — Maximale statische Laufspeicherkapazität pro Lauf	Jede unterstützte Region: 9.600	Ja	Die maximale statische Laufspeicherkapazität in Gibibyte (GiB) für jeden Lauf in der aktuellen AWS Region. In us-east-1 und us-west-2 werden Quotenerhöhungsanträge für Werte bis zu 50.000 automatisch genehmigt.
Workflows — Maximale Anzahl an Arbeitsabläufen	Jede unterstützte Region: 1 000	Ja	Die maximale Anzahl von Workflows in der aktuellen AWS Region.
Workflows — Transaktionen pro Sekunde (TPS) für den Vorgang StartRun	Jede unterstützte Region: 1	Ja	Die maximalen Transaktionen pro Sekunde (TPS) für den StartRun Vorgang in der aktuellen AWS Region.

HealthOmics Kontingente mit fester Größe

HealthOmics Enthält zusätzlich zu den HealthOmics Servicekontingenten Kontingente mit festen Größen. Für diese Werte können Sie keine Erhöhung beantragen.

Sofern nicht anders angegeben, ist in jedem Kontingent der Höchstwert pro Region aufgeführt.

Themen

- HealthOmics Analytics-Kontingente mit fester Größe
- HealthOmics Speicherplatz mit fester Größe, Kontingente
- HealthOmics Workflow-Kontingente mit fester Größe
- HealthOmics Feste Größenkontingente für den Ready2Run-Workflow

HealthOmics Analytics-Kontingente mit fester Größe

Die folgende Tabelle zeigt die maximal unterstützten Werte für Analytics-Kontingente. Diese Werte sind nicht anpassbar.

Name	Description	Maximum	Einstellbar Ja/Nein
Analytik — Maximale Anzahl von Dateien pro Importauftrag für den Annotatio nsspeicher	Die maximale Anzahl von Dateien pro Importauftrag für Anmerkungen.	1	Nein

HealthOmics Speicherplatz mit fester Größe, Kontingente

Die folgende Tabelle zeigt die maximal unterstützten Werte für Speicherdateien. Diese Werte sind nicht anpassbar.

Name	Description	Maximum	Einstellbar Ja/Nein
Speicher — Maximale Größe der S3-Zugrif fsressourcenrichtlinie	Die maximale Größe der S3-Zugriffsressour cenrichtlinie	15 KB	Nein
Speicher — Höchstzahl der weitergegebenen Set-Level-Tags	Die maximale Anzahl von Tag-Schlüsseln auf Set-Ebene pro Speicher, die sich auf das S3-Objekt übertragen	5	Nein
Speicher — Maximale Anzahl von Lesesätze n pro Aktivieru ngsauftrag	Die maximale Anzahl von Lesesätzen pro Aktivierungsauftrag.	20	Nein

Name	Description	Maximum	Einstellbar Ja/Nein
Speicher — Maximale Anzahl von Lesesätzen pro Exportauftrag	Die maximale Anzahl von Lesesätzen pro Exportauftrag.	100	Nein
Speicher — Maximale Anzahl von Lesesätzen pro Importauftrag	Die maximale Anzahl von Lesesätzen pro Importauftrag.	100	Nein
Speicher — Maximale Anzahl an Referenzspeichern	Die maximale Anzahl von Referenzspeichern.	1	Nein
Speicher — Maximale Bauteilgröße für einen direkten Upload	Die maximale Bauteilgröße für den direkten Upload in einen Sequenzspeicher.	100 MB	Nein
Speicher — Maximale Anzahl an Teilen in einer Datei für den direkten Upload	Die maximale Anzahl von Teilen in einer Datei für den direkten Upload in einen Sequenzspeicher.	10.000	Nein
Speicher — Maximale Referenzgröße	Die maximale Größe einer Referenzdatei, die in einen Referenzspeicher importiert werden kann.	15 GB	Nein

Name	Description	Maximum	Einstellbar Ja/Nein
Speicher — Maximale Größe der Lesesatzquelle	Die maximale Größe einer einzelnen Quelldatei in einem Lesesatz, die in einen Sequenzspeicher importiert werden kann.	976 GB	Nein

HealthOmics Workflow-Kontingente mit fester Größe

Die folgende Tabelle zeigt die unterstützten Höchstwerte für Workflow-Kontingente. Diese Werte sind nicht anpassbar.

Name	Description	Maximale Größe	Einstellbar Ja/Nein
Workflows — Maximale Anzahl ausgeführter Gruppen	Die maximale Anzahl von Ausführungsgruppen.	1000	Nein
Workflows — Maximale Anzahl ausgeführter Caches	Die maximale Anzahl von Run-Caches, die Sie für ein Konto erstellen können. Ein oder mehrere Läufe können sich denselben Run-Cache teilen. Es gibt kein Kontingent für die Anzahl der Läufe, die pro Konto zwischengespeichert werden HealthOmics können.	1000	Nein

Name	Description	Maximale Größe	Einstellbar Ja/Nein
Workflows — Maximale Anzahl an Workflow-Versionen	Die maximale Anzahl von Workflow-Versionen pro Workflow.	1000	Nein
Workflows — Größe des Containers der CPU-Instanz	Die maximale Container-Image-Größe für eine CPU-Instanz.	45 GiB	Nein
Workflows — Größe des Containers der GPU-Instanz	Die maximale Container-Image-Größe für eine GPU-Instanz.	95 GiB	Nein
GPU-Instanz /dev/shm gemeinsam genutzter Speicher	Die maximale Menge an gemeinsam genutztem Speicher pro GPU-Instanz.	8 GB pro GPU	Nein
Workflows — Führen Sie die Parameterdatei aus	Die maximale Größe einer Ausführungsparameterdatei.	50.000 Byte	Nein
Workflows — Vorlagendatei mit Workflow-Parametern	Die maximale Anzahl von Einträgen und die maximale Dateigröße für eine Workflow-Parameter-Vorlagendatei. Dieses Kontingent gilt für Workflows, die Sie mit der Konsole oder API erstellen.	1.000 Einträge, 400 KB	Nein

Name	Description	Maximale Größe	Einstellbar Ja/Nein
Workflows — Größe der Workflow-Definitionsdatei — API	Die maximale Größe der Workflow-Definitionsdatei, wenn Sie den Workflow mithilfe der API-Operation oder eines AWS SDK erstellen.	100 MB	Nein
Workflows — Größe der Workflow-Definitionsdatei — Konsole (direkter Upload)	Die maximale Größe der Workflow-Definitionsdatei, die Sie als direkten Upload bereitstellen können, wenn Sie den Workflow mithilfe der Konsole erstellen.	4,4 MB	Nein
Workflows — Größe der Workflow-Definitionsdatei — Konsole (Upload von Amazon S3)	Die maximale Größe der Workflow-Definitionsdatei, die Sie als Upload von Amazon S3 bereitstellen können, wenn Sie den Workflow mit der Konsole erstellen.	100 MB	Nein
Workflows — Größe des Repositorys	Die maximale Größe eines externen Code-Repositorys.	1 GiB	Nein
Workflows — Individuelle Dateigröße des Repositorys	Die maximale Größe einer einzelnen Datei aus einem externen Code-Repository.	100 MiB	Nein

Name	Description	Maximale Größe	Einstellbar Ja/Nein
Workflows — Größe der README-Datei	Die maximale Größe einer README-Datei.	500 KiB	Nein

Vorschläge, wie Sie die Größe Ihrer Ausführungsparameterdatei reduzieren können, finden Sie unterGröße der Ausführungsparameter verwalten.

HealthOmics Feste Größenkontingente für den Ready2Run-Workflow

Jeder Ready2Run-Workflow hat eine maximale Größe der Eingabedatei. In der folgenden Tabelle sind die Dateigrößeneinheiten in Gibibytes (GiB) aufgeführt. Diese maximalen Dateigrößen sind nicht anpassbar.

Name des Ready2Run-Workflows	Maximale Größe der Eingabedatei (GiB)	Einstellbar (Ja/Nein)
AlphaFold für 601-1200 Reste	1	Nein
AlphaFold für bis zu 600 Rückstände	1	Nein
Bases2Fastq für 2x150	1000	Nein
Bases2Fastq für 2x300	1000	Nein
Bases2Fastq für 2x75	500	Nein
ESMFold für bis zu 800 Reste	1	Nein
GATK-BP fq2bam	64	Nein
GATK-BP-Keimbahn bam2vcf für das 30-fache Genom	39	Nein
GATK-BP-Keimbahn fq2vcf für 30-faches Genom	64	Nein

Name des Ready2Run-Workflows	Maximale Größe der Eingabedatei (GiB)	Einstellbar (Ja/Nein)
GATK-BP Somatisches WES bam2vcf	86	Nein
NVIDIA Parabricks BAM2 FQ2 BAM WGS für bis zu 30X	80	Nein
NVIDIA BAM2 FQ2 Parabricks BAM WGS für bis zu 50X	120	Nein
NVIDIA BAM2 FQ2 Parabricks BAM WGS für bis zu 5X	20	Nein
NVIDIA FQ2 Parabricks BAM WGS für bis zu 30-fach	71	Nein
NVIDIA FQ2 Parabricks BAM WGS für bis zu 50X	137	Nein
NVIDIA FQ2 Parabricks BAM WGS für bis zu 5X	13	Nein
NVIDIA Parabricks Germline WGS für bis zu 30-fach DeepVariant	71	Nein
NVIDIA Parabricks Germline WGS für bis zu 50X DeepVariant	137	Nein
NVIDIA Parabricks Germline WGS für bis zu 5X DeepVaria nt	12	Nein
NVIDIA Parabricks Germline WGS für bis zu 30-fach HaplotypeCaller	71	Nein

Name des Ready2Run-Workflows	Maximale Größe der Eingabedatei (GiB)	Einstellbar (Ja/Nein)
NVIDIA Parabricks Germline WGS für bis zu 50X Haplotype Caller	137	Nein
NVIDIA Parabricks Germline WGS für bis zu 5X Haplotype Caller	13	Nein
NVIDIA Parabricks Somatic Mutect2 WGS für bis zu 50X	196	Nein
sc RNAseq mit Kallisto BUStools	119	Nein
Sc RNAseq mit in Alevinen gebratenem Lachs	119	Nein
sc RNAseq mit STARsolo	119	Nein
Sentieon Germline BAM WES für bis zu 300x	9	Nein
Sentieon Germline BAM WGS für bis zu 32x	18	Nein
Sentieon Germline FASTQ WES für bis zu 100x	5	Nein
Sentieon Germline FASTQ WES für bis zu 300x	26	Nein
Sentieon Germline FASTQ WGS für bis zu 32x	51	Nein
Sentieon LongRead für ONT	25	Nein

Name des Ready2Run-Workflows	Maximale Größe der Eingabedatei (GiB)	Einstellbar (Ja/Nein)
Sentieon für LongRead PacBio HiFi	58	Nein
Sentieon Somatic WES	50	Nein
Sentieon Somatic WGS	113	Nein
Ultima Genomics für bis zu 40x DeepVariant	91	Nein

HealthOmics API-Kontingente

HealthOmics hat die folgenden Kontingente für API-Operationen. Wo angegeben, ist das Kontingent anpassbar. Verwenden Sie das Formular zur Erhöhung der Quote, um eine Erhöhung der Quote zu beantragen.

Für jeden aufgeführten API-Vorgang entspricht das Kontingent der maximalen Anzahl an Transaktionen pro Sekunde (TPS) für diesen API-Vorgang in jeder Region.

Themen

- Allgemeine API-Kontingente
- Speicher-API-Kontingente
- Workflow-API-Kontingente
- Analytics-API-Kontingente

Allgemeine API-Kontingente

In der folgenden Tabelle sind allgemeine API-Operationen aufgeführt, die für mehr als eine Kategorie gelten (Speicher, Workflows und Analysen).

API-Operation	Standardmäßige maximale TPS-Anzahl	Einstellbar (Ja/Nein)
AcceptShare, CreateSha re, DeleteShare, GetShare, ListShares	1 TPS	Ja

Speicher-API-Kontingente

In der folgenden Tabelle sind die Speicher-API-Operationen aufgeführt.

Speicher-API-Betrieb	Standardmäßige maximale TPS-Anzahl	Einstellbar (Ja/Nein)
CreateSequenceStore, UpdateSequenceStore, DeleteSequenceStore, CreateReferenceStore, DeleteReferenceStore	1 TPS	Ja
BatchDeleteReadSet, DeleteReference	1 TPS	Ja
CreateMultipartReadSetUploa d, CompleteMultipartR eadSetUpload, AbortMult ipartReadSetUpload	1 TPS	Nein
Ruft 3 abAccessPolicy, gibt 3 ab, löscht S3 AccessPolicy AccessPolicy	1 TPS	Ja
GetReference	10 TPS	Ja
UploadReadSetPart	10 TPS	Ja
GetReadSet	30 TPS	Ja

Speicher-API-Betrieb	Standardmäßige maximale TPS-Anzahl	Einstellbar (Ja/Nein)
GetSequenceStore, ListSequenceStores	5 TPS	Ja
GetReadSetMetadata, ListReadSets	5 TPS	Ja
StartReadSetImportJob, GetReadSetImportJob, ListReadSetImportJobs	5 TPS	Ja
StartReadSetExportJob, GetReadSetExportJob, ListReadSetExportJobs	5 TPS	Ja
ListReferenceStores	5 TPS	Ja
StartReferencetImportJob, GetReferenceImportJob, ListReferenceImportJobs	5 TPS	Ja
ListReferences, GetReferenceMetadata	5 TPS	Ja
StartReadsetActivationJob	5 TPS	Ja
ListReadsetActivationJobs, GetReadSetActivationJob	5 TPS	Ja
ListMultipartReadSetUploads, ListReadSetUploadParts	5 TPS	Ja
TagResource, UntagResource, ListTagsForResource	5 TPS	Ja

Workflow-API-Kontingente

In der folgenden Tabelle sind die Workflow-API-Operationen aufgeführt.

Workflow-API-Betrieb	Standardmäßige maximale TPS-Anzahl	Einstellbar (Ja/Nein)
StartRun	1 TPS	Ja
CreateWorkflow	5 TPS	Ja
CancelRun, DeleteRun , GetRun, GetRunTask, ListRunTasks, ListRuns	10 TPS	Ja
CreateRunGroup, DeleteRun Group, GetRunGroup, ListRunGroups, UpdateRun Group	10 TPS	Ja
CreateRunCache, UpdateRun Cache, DeleteRunCache, GetRunCache, ListRunCaches	10 TPS	Ja
DeleteWorkflow, GetWorkfl ow, ListWorkflows, UpdateWor kflow	10 TPS	Ja

Analytics-API-Kontingente

In der folgenden Tabelle sind die Analytics-API-Operationen aufgeführt.

Betrieb der Analytics-API	Standardmäßige maximale TPS-Anzahl	Einstellbar (Ja/Nein)
CreateVariantStore, DeleteVar iantStore, GetVariantStore,	1 TPS	Nein

Betrieb der Analytics-API	Standardmäßige maximale TPS-Anzahl	Einstellbar (Ja/Nein)
ListVariantStores, UpdateVariantStore		
StartVariantImportJob, CancelVariantImportJob, GetVariantImportJob, ListVariantImportJobs	1 TPS	Nein
CreateAnnotationStore, DeleteAnnotationStore, GetAnnotationStore, ListAnnotationStores, UpdateAnnotationStore	1 TPS	Nein
StartAnnotationImportJob, ListAnnotationImportJobs, GetAnnotationImportJob, CancelAnnotationImportJob	1 TPS	Nein

Dokumentenverlauf für das HealthOmics Benutzerhandbuch

In der folgenden Tabelle werden die Dokumentationsversionen für beschrieben HealthOmics.

Änderung	Beschreibung	Datum
AWS HealthOmics Varianten speicher und Annotatio nsspeicher stehen Neukunden nicht mehr zur Verfügung.	AWS HealthOmics Varianten speicher und Annotatio nsspeicher stehen Neukunden nicht mehr zur Verfügung . Weitere Informationen finden Sie unter Änderung der Verfügbarkeit von AWS HealthOmics Varianten speichern und Annotatio nsspeichern.	7. November 2025
AWS HealthOmics Variant Stores und Annotation Stores werden ab dem 7. November 2025 nicht mehr für Neukunden geöffnet sein.	AWS HealthOmics Varianten geschäfte und Annotatio nsspeicher werden ab dem 7. November 2025 nicht mehr für Neukunden geöffnet sein. Wenn Sie Variantenspeicher oder Annotationsspeicher nutzen möchten, melden Sie sich vor diesem Datum an. Vorhandene Kunden können den Service weiterhin wie gewohnt verwenden. Weitere Informationen finden Sie unter Änderung der Verfügbar keit von AWS HealthOmi cs Variantenspeichern und Annotationsspeichern.	7. Oktober 2025
Neue Funktionen	HealthOmics Unterstützung für Workflows hinzugefügt, um ein	28. August 2025

	privates Amazon ECR-Repos itory mit einer Upstream-Registry zu synchronisieren. Weitere Informationen finden Sie unter Container-Images für private Workflows in HealthOmics.	
Neue README- und Repository-Integrationsfunk tionen	Unterstützung für die Erstellun g von Workflows aus externen Code-Repositorys und README-Dateien hinzugefü gt.	24. Juli 2025
Neue Funktionen	HealthOmics Unterstüt zung für die automatische Nextflow-Parameterinterpola tion hinzugefügt. Weitere Informationen finden Sie unter Parameter-Vorlagendateien für Workflows. HealthOmics	27. Juni 2025
Neue Funktionen	HealthOmics Unterstützung für Workflows hinzugefügt, um die Ausführungsparamet er aus einer WDL-Workflow-Definitionsdatei zu interpoli eren. Weitere Informationen finden Sie unter Parameter-Vorlagendateien für Workflows . HealthOmics	30. Mai 2025

Neue Funktionen	HealthOmics Unterstützung für Workflow-Versionierung hinzugefügt. Weitere Informationen finden Sie unter Workflow-Versionierung in. HealthOmics	18. April 2025
Neue Funktionen	HealthOmics elastischer Durchsatz für dynamischen Run-Speicher hinzugefügt. Weitere Informationen finden Sie unter Speichertypen ausführen in HealthOmics.	16. April 2025
Neue Funktionen	HealthOmics Es wurden attributbasierte Zugriffskontrollen für Sequence Store S3-Standorte und die Möglichkeit hinzugefügt, bis zu fünf Read-Set-Tags mit einem Sequence Store S3-Objekt zu synchronisieren. Weitere Informationen finden Sie unter Einen HealthOmics Sequenzspeicher erstellen.	22. November 2024
Neue Funktionen	HealthOmics Unterstützung für das Zwischenspeichern von Anrufen, auch bekannt als Resume, für private Workflows hinzugefügt. Weitere Informationen finden Sie unter Anruf-Caching.	20. November 2024

Neue Funktionen	HealthOmics Es wurden neue API-Felder hinzugefügt, mit denen Sie Eingabejobs für Sequenzspeicher und Lesesätzen zuordnen können.	29. August 2024
Neue Funktionen	HealthOmics Unterstützung für die Verwaltung von Nextflow-Versionen hinzugefügt. Weitere Informationen finden Sie unter Nextflow-Versionen.	14. August 2024
Neue Funktionen	HealthOmics Unterstützung für gemeinsam genutzte Workflows und dynamischen Run-Speicher hinzugefügt.	30. April 2024
Neue Funktionen	HealthOmics Unterstützung für Amazon S3 S3-Zugriff auf Referenz- und Sequenzspeicher sowie Unterstützung für hinzugefügt SHA256 ETags.	15. April 2024
Neue Funktionen	HealthOmics Entitäts-Tags (ETags) für Sequenzspeicher hinzugefügt.	06. Oktober 2023
Neue Funktionen	HealthOmics Versionsverwaltung für Annotationsspeicher und gemeinsame Nutzung von Analysespeichern hinzugefügt.	15. August 2023
Neue Funktionen	HealthOmics Common Workflow Language (CWL) als unterstützte Sprache für HealthOmics Workflows hinzugefügt.	30. Juni 2023

Neue Funktionen	HealthOmics neue Ready2Run-Workflows, GPU-Unterstützung für Workflows, Datenanalyse für Annotationsspeicher, direktes Hochladen in den HealthOmics Speicher und Integration mit hinzugefügt. EventBridge	15. Mai 2023
Neue verwaltete Richtlinie	HealthOmics Es wurde eine neue verwaltete Richtlinie hinzugefügt, die vollen Zugriff bietet. Weitere Informationen finden Sie unter Von AWS verwaltete Richtlinien.	23. Februar 2023
Neue verwaltete Richtlinie	HealthOmics Es wurde eine neue verwaltete Richtlinie hinzugefügt, die den Zugriff nur auf Lesezugriff beschränkt. Weitere Informationen finden Sie unter Von AWS verwaltete Richtlinien.	29. November 2022
Erstversion	Erste Version des HealthOmics Benutzerhandbuchs	29. November 2022

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.