aws

Panduan Pengguna

AWS HealthOmics

Versi latest

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

AWS HealthOmics: Panduan Pengguna

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

Merek dagang dan tampilan dagang Amazon tidak boleh digunakan sehubungan dengan produk atau layanan apa pun yang bukan milik Amazon, dengan cara apa pun yang dapat menyebabkan kebingungan di antara pelanggan, atau dengan cara apa pun yang merendahkan atau mendiskreditkan Amazon. Semua merek dagang lain yang tidak dimiliki oleh Amazon merupakan hak milik masing-masing pemiliknya, yang mungkin atau mungkin tidak terafiliasi, terkait dengan, atau disponsori oleh Amazon.

Table of Contents

Apa itu AWS HealthOmics?

AWS HealthOmics adalah layanan yang memenuhi syarat HIPAA yang mempercepat pengujian diagnostik klinis, penemuan obat, dan penelitian pertanian dengan sepenuhnya mengelola infrastruktur kompleks di balik alur kerja bioinformatika Anda. HealthOmics mendukung bahasa alur kerja standar industri (WDL, Nextflow, CWL) dan menskalakan infrastruktur bioinformatika dengan mulus untuk mendukung data dari puluhan ribu pengujian per hari—semuanya dengan biaya per sampel yang dapat diprediksi. HealthOmics menangani kompleksitas teknis seperti mengelola sumber daya komputasi dan memelihara mesin alur kerja sehingga Anda dapat fokus sepenuhnya pada terobosan ilmiah.

Topik

Pemberitahuan penting

HealthOmics dimaksudkan hanya untuk mentransfer, menyimpan, memformat, atau menampilkan data, dan untuk penyediaan infrastruktur dan dukungan konfigurasi untuk mengelola alur kerja. HealthOmics bukan pengganti saran medis profesional, diagnosis, atau perawatan, dan tidak dimaksudkan untuk menyembuhkan, mengobati, mengurangi, mencegah, atau mendiagnosis penyakit atau kondisi kesehatan apa pun. Anda bertanggung jawab untuk melembagakan tinjauan manusia sebagai bagian dari penggunaan apa pun AWS HealthOmics, termasuk terkait dengan produk pihak ketiga yang dimaksudkan untuk menginformasikan pengambilan keputusan klinis.

HealthOmics fitur

Kasus penggunaan primer untuk HealthOmics:

- Diagnostik klinis — Membangun dan menskalakan alur kerja pengujian diagnostik dengan biaya yang dapat diprediksi dan infrastruktur yang dikelola sepenuhnya yang tumbuh seiring dengan volume pengujian Anda.
- Penemuan obat — Mempercepat penelitian terapeutik dengan mengatur model fondasi biologis dalam skala besar, memungkinkan iterasi cepat di jutaan kandidat potensial.
- Penelitian pertanian — Meningkatkan sifat tanaman seperti toleransi kekeringan dan ketahanan hama melalui alur kerja bertenaga AI yang meningkatkan ketahanan pangan dan produktivitas pertanian.

Manfaat utama dari HealthOmics:

- Skalabilitas - Skala alur kerja di 100.000+ v bersamaan CPUs untuk mendukung puluhan ribu pengujian setiap hari dengan manajemen infrastruktur nol dan biaya per sampel yang dapat diprediksi.
- Fokus pada sains, bukan infrastruktur — Gunakan bahasa alur kerja yang sudah dikenal dan APIs sementara AWS secara otomatis menangani orkestrasi infrastruktur dan manajemen data di belakang layar.
- Menjaga kepatuhan — Jejak audit yang komprehensif, pelacakan sumber data, dan infrastruktur yang memenuhi syarat HIPAA yang dirancang untuk alur kerja klinis—semuanya out-of-the-box — mendukung pengembangan solusi yang memenuhi persyaratan peraturan.

HealthOmics terdiri dari tiga komponen utama:

- HealthOmics alur kerja — Jalankan perhitungan bioinformatika pada infrastruktur yang disediakan dan diskalakan secara otomatis.
- HealthOmics penyimpanan — Simpan dan bagikan petabyte data genomik secara efisien dengan biaya rendah per gigabase.
- HealthOmics analitik — Siapkan data genomik untuk analisis multiomik dan multimodal.

Gunakan komponen ini secara independen atau gabungkan mereka untuk end-to-end solusi.

HealthOmics konsep

Topik ini mencakup definisi untuk konsep dan istilah kunci yang khusus untuk HealthOmics, untuk membantu Anda memahami terminologi yang HealthOmics digunakan panduan ini.

Topik

Alur kerja

Dengan HealthOmics Alur Kerja, Anda dapat memproses dan menganalisis data genomik Anda.

- Alur kerja — Definisi keseluruhan dari proses ujung ke ujung termasuk parameter dan referensi ke alat. Definisi alur kerja dapat dinyatakan sebagai WDL, Nextflow, atau CWL. Setiap alur kerja yang dibuat memiliki pengenal unik.
- Jalankan - Pemanggilan tunggal alur kerja. Run individual menggunakan data input yang Anda tentukan dan menghasilkan output. Setiap run yang dibuat memiliki pengenal unik.
- Tugas — Proses individu dalam proses. HealthOmics Alur kerja menggunakan spesifikasi komputasi yang ditentukan ini untuk menjalankan tugas Anda. Setiap tugas memiliki pengenal unik.
- Jalankan grup - Sekelompok run yang dapat Anda atur vCPU maks, durasi maks, atau run bersamaan maks untuk membantu membatasi sumber daya komputasi yang digunakan per proses. Anda dapat menentukan dan mengonfigurasi prioritas untuk menjalankan dalam grup run. Misalnya, Anda dapat menentukan bahwa menjalankan prioritas tinggi akan dilakukan sebelum prioritas yang lebih rendah, menciptakan antrian prioritas. Ini opsional untuk menggunakan Run Group, dan setiap Run Group memiliki pengenal unik.

Penyimpanan

Penyimpanan data dipisahkan menjadi penyimpanan urutan, untuk urutan genomik Anda dan informasi terkait, dan toko referensi, untuk semua genom referensi Anda. Istilah-istilah berikut menjelaskan implementasi yang khusus untuk HealthOmics.

- Sequence store — Penyimpanan data untuk penyimpanan file genomik. Anda dapat memiliki satu atau lebih toko urutan di dalamnya HealthOmics. Izin akses dan AWS KMS enkripsi dapat diatur pada penyimpanan urutan untuk mengontrol siapa yang memiliki akses ke data.
- Set baca — Set baca adalah abstraksi pembacaan genomik, yang disimpan dalam format FASTQ, BAM, atau CRAM. Set baca dapat diimpor ke toko urutan dan dianotasi dengan metadata. Anda dapat menerapkan izin untuk membaca set menggunakan kontrol akses berbasis atribut (ABAC).

- Referensi — Referensi genom digunakan dengan pembacaan untuk mengidentifikasi di mana dalam genom pembacaan tertentu, atau kelompok pembacaan, dipetakan. Ini dalam format FASTA dan disimpan di toko referensi.
- Toko referensi — Penyimpanan data untuk penyimpanan genom referensi. Anda dapat memiliki satu toko referensi di setiap akun dan wilayah.

Analitik

Anda dapat mengubah dan menganalisis data genomik Anda dengan HealthOmics Analytics. Buat toko varian atau toko anotasi untuk menyertakan informasi tambahan untuk kueri Anda.

- Toko varian — penyimpanan data yang menyimpan data varian pada skala populasi. Toko varian mendukung input Genomic Variant Call Format (GvCF) dan VCF.
- Penyimpanan anotasi — Penyimpanan data yang mewakili database anotasi, seperti dari file TSV/CSV, VCF, atau General Feature Format (). GFF3 Toko Anotasi dipetakan ke sistem koordinat yang sama dengan toko varian selama impor.

Layanan terkait

Layanan berikut bekerja dengan HealthOmics.

- Amazon Elastic Container Registry - Setiap alur kerja pribadi menggunakan image Amazon ECR (dalam repositori Amazon ECR pribadi) untuk memuat semua executable, library, dan skrip yang diperlukan untuk menjalankan alur kerja.
- Amazon Simple Storage Service - Amazon S3 menyediakan penyimpanan file untuk data Store dan Workflow.
- AWS Lake Formation — Lake Formation mengelola akses data ke penyimpanan data Analytics Anda.
- Amazon Athena — Gunakan Athena untuk melakukan kueri di toko Varian Anda.
- Amazon SageMaker AI — Gunakan SageMaker AI untuk menjalankan HealthOmics tugas menggunakan notebook Jupyter.
- GitHub connections— Gunakan koneksi untuk menghubungkan respoitori kode eksternal Anda ke alur kerja Anda. HealthOmics

Cara mengakses HealthOmics

Anda dapat mengakses AWS HealthOmics fitur menggunakan konsol manajemen, CLI, SDKs atau API.

- AWS Management Console — Menyediakan antarmuka web yang dapat Anda gunakan untuk mengakses HealthOmics.
- AWS Command Line Interface (AWS CLI) - Menyediakan perintah untuk serangkaian AWS layanan yang luas, termasuk AWS HealthOmics, dan didukung di Windows, macOS, dan Linux. Untuk informasi lebih lanjut tentang menginstal AWS CLI, lihat AWS Command Line Interface.
- AWS SDKs — AWS menyediakan SDKs (Kit Pengembangan Perangkat Lunak) yang terdiri dari pustaka dan kode sampel untuk berbagai bahasa dan platform pemrograman (termasuk Java, Python, Ruby, .NET, iOS, dan Android). SDKs Menyediakan cara yang nyaman untuk digunakan secara HealthOmics terprogram. Untuk informasi selengkapnya, lihat Pusat Pengembang AWS SDK.
- AWS API — Anda dapat menggunakan operasi API untuk mengakses dan mengelola secara HealthOmics terprogram. Untuk informasi lebih lanjut, lihat Referensi API HealthOmics .

Wilayah dan titik akhir untuk AWS HealthOmics

Untuk daftar lengkap wilayah dan titik akhir, lihat Referensi AWS Umum.

Selain AWS wilayah yang aktif secara default, ada juga Wilayah Opt-in yang perlu diaktifkan. Untuk mempelajari lebih lanjut tentang cara mengaktifkan atau menonaktifkan Wilayah, lihat Menentukan AWS Wilayah mana yang dapat digunakan akun Anda dalam panduan Manajemen AWS Akun.

Pelajari selengkapnya

Pelajari lebih lanjut HealthOmics dari lokakarya dan tutorial ini:

- HealthOmics lokakarya — lokakarya HealthOmics ujung ke ujung
- AWS sumber daya genomik - Repositori ECR Amazon publik yang terkait dengan genomik
- Tutorial Python — Tutorial notebook Jupyter tentang GitHub, yang mencakup HealthOmics penyimpanan, analitik, dan alur kerja

Menjadi akrab dengan HealthOmics alat tambahan yang AWS menyediakan:

- WDL linter - HealthOmics linter untuk WDL
- Linter Nextflow - HealthOmics linter untuk Nextflow
- HealthOmics Alat pembantu Amazon ECR - Alat pembantu Amazon ECR untuk HealthOmics
- HealthOmics alat aktif GitHub — Alat untuk bekerja dengan HealthOmics (Manajer transfer, pengurai URI, dijalankan ulang Omics, Run analyzer).

AWS HealthOmics toko varian dan perubahan ketersediaan toko anotasi

Setelah mempertimbangkan dengan cermat, kami memutuskan untuk menutup toko AWS HealthOmics varian dan toko anotasi kepada pelanggan baru mulai 7 November 2025. Pelanggan yang sudah ada dapat terus menggunakan layanan ini seperti biasa.

Bagian berikut menjelaskan opsi migrasi untuk membantu Anda memindahkan toko varian dan toko analitik ke solusi baru. Untuk pertanyaan atau masalah apa pun, buat kasus dukungan di support.console.aws.amazon.com.

Topik

Ikhtisar opsi migrasi

Opsi migrasi berikut memberikan alternatif untuk menggunakan toko varian dan toko anotasi:

1. Gunakan implementasi referensi HealthOmics yang disediakan dari logika ETL.

   Gunakan bucket tabel S3 untuk penyimpanan dan terus gunakan layanan AWS analitik yang ada.
2. Buat solusi menggunakan kombinasi AWS layanan yang ada.

   Untuk ETL, Anda dapat menulis pekerjaan Glue ETL khusus, atau menggunakan kode HAIL atau GLOW sumber terbuka di EMR, untuk mengubah data varian.

   Gunakan bucket tabel S3 untuk penyimpanan dan terus gunakan layanan analitik yang ada AWS
3. Pilih AWS mitra yang menawarkan alternatif toko varian dan anotasi.

Opsi migrasi untuk logika ETL

Pertimbangkan opsi migrasi berikut untuk logika ETL:

1. HealthOmics menyediakan logika ETL penyimpanan varian saat ini sebagai HealthOmics alur kerja referensi. Anda dapat menggunakan mesin alur kerja ini untuk memberi daya pada proses ETL data varian yang persis sama dengan penyimpanan varian, tetapi dengan kontrol penuh atas logika ETL.

   Alur kerja referensi ini tersedia berdasarkan permintaan. Untuk meminta akses, buat kasus dukungan di support.console.aws.amazon.com.
2. Untuk mengubah data varian, Anda dapat menulis pekerjaan ETL Glue kustom, atau menggunakan kode HAIL atau GLOW sumber terbuka di EMR.

Opsi migrasi untuk penyimpanan

Sebagai pengganti penyimpanan data yang dihosting layanan, Anda dapat menggunakan bucket tabel Amazon S3 untuk menentukan skema tabel kustom.Untuk informasi selengkapnya tentang bucket tabel, lihat Bucket tabel di Panduan Pengguna Amazon S3.

Anda dapat menggunakan bucket tabel untuk tabel Iceberg yang dikelola sepenuhnya di Amazon S3.

Anda dapat memunculkan kasus dukungan untuk meminta HealthOmics tim memigrasikan data dari varian atau penyimpanan anotasi ke bucket tabel Amazon S3 yang dikonfigurasi.

Setelah data diisi di keranjang tabel Amazon S3, Anda dapat menghapus toko varian dan toko anotasi. Untuk informasi selengkapnya, lihat Menghapus toko HealthOmics analitik.

Analitik

Untuk analitik data, terus gunakan layanan AWS analitik, seperti Amazon Athena, Amazon EMR, Amazon Redshift, atau Amazon Quick Suite.

AWS Mitra

Anda dapat bekerja dengan AWS mitra yang menyediakan ETL yang dapat disesuaikan, skema tabel, alat kueri dan analisis bawaan, dan antarmuka pengguna untuk berinteraksi dengan data.

Contoh

Contoh berikut menunjukkan cara membuat tabel yang cocok untuk menyimpan data VCF dan GVCF.

Athena DDL

Anda dapat menggunakan contoh DDL berikut di Athena untuk membuat tabel yang cocok untuk menyimpan data VCF dan GVCF dalam satu tabel. Contoh ini tidak sama persis dengan struktur penyimpanan varian, tetapi berfungsi dengan baik untuk kasus penggunaan umum.

Buat nilai Anda sendiri untuk DATABASE_NAME dan TABLE_NAME saat Anda membuat tabel.

```
 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'
);
```

Buat tabel menggunakan Python (tanpa Athena)

Contoh kode Python berikut menunjukkan cara membuat tabel tanpa menggunakan Athena.

```
 import boto3
from pyiceberg.catalog import Catalog, load_catalog
from pyiceberg.schema import Schema
from pyiceberg.table import Table
```

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


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

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

    return load_catalog("s3tables", **catalog_config)


def create_namespace(catalog: Catalog, namespace: str) -> None:
    try:
        catalog.create_namespace(namespace)
        print(f"Created namespace: {namespace}")
    except Exception as e:
        if "already exists" in str(e):
            print(f"Namespace {namespace} already exists.")
        else:
            raise e


def create_table(catalog: Catalog, namespace: str, table_name: str, schema: Schema,
                 partition_spec: PartitionSpec = None, sort_order: SortOrder = None) ->
 Table:
```

```
    if catalog.table_exists(f"{namespace}.{table_name}"):
        print(f"Table {namespace}.{table_name} already exists.")
        return catalog.load_table(f"{namespace}.{table_name}")

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

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

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


def main(bucket_arn: str, namespace: str, table_name: str):
    # Schema definition
    genomic_variants_schema = Schema(
        NestedField(1, "sample_name", StringType(), required=True),
        NestedField(2, "variant_name", StringType(), required=True),
        NestedField(3, "chrom", StringType(), required=True),
        NestedField(4, "pos", LongType(), required=True),
        NestedField(5, "ref", StringType(), required=True),
        NestedField(6, "alt", ListType(element_id=1000, element_type=StringType(),
 element_required=True), required=True),
        NestedField(7, "qual", DoubleType()),
        NestedField(8, "filter", StringType()),
        NestedField(9, "genotype", StringType()),
        NestedField(10, "info", MapType(key_type=StringType(), key_id=1001,
 value_type=StringType(), value_id=1002)),
        NestedField(11, "attributes", MapType(key_type=StringType(), key_id=2001,
 value_type=StringType(), value_id=2002)),
        NestedField(12, "is_reference_block", BooleanType()),
        identifier_field_ids=[1, 2, 3, 4]
    )

    # Partition and sort specifications
    partition_spec = PartitionSpec(
```

```
        PartitionField(source_id=1, field_id=1001, transform=BucketTransform(128),
 name="sample_bucket"),
        PartitionField(source_id=3, field_id=1002, transform=IdentityTransform(),
 name="chrom")
    )

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

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

    return table


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

    main(bucket_arn, namespace, table_name)
```

Menyiapkan HealthOmics

Untuk mengatur AWS HealthOmics, daftar, buat pengguna administratif Akun AWS, dan kelola akses untuk pengguna tambahan dengan aman.

Topik

Mendaftar untuk Akun AWS

Jika Anda tidak memiliki Akun AWS, selesaikan langkah-langkah berikut untuk membuatnya.

Untuk mendaftar untuk Akun AWS

1. Buka https://portal.aws.amazon.com/billing/pendaftaran.
2. Ikuti petunjuk online.

   Bagian dari prosedur pendaftaran melibatkan menerima panggilan telepon atau pesan teks dan memasukkan kode verifikasi pada keypad telepon.

   Saat Anda mendaftar untuk sebuah Akun AWS, sebuah Pengguna root akun AWSdibuat. Pengguna root memiliki akses ke semua Layanan AWS dan sumber daya di akun. Sebagai praktik keamanan terbaik, tetapkan akses administratif ke pengguna, dan gunakan hanya pengguna root untuk melakukan tugas yang memerlukan akses pengguna root.

AWS mengirimi Anda email konfirmasi setelah proses pendaftaran selesai. Kapan saja, Anda dapat melihat aktivitas akun Anda saat ini dan mengelola akun Anda dengan masuk https://aws.amazon.comke/ dan memilih Akun Saya.

Buat pengguna dengan akses administratif

Setelah Anda mendaftar Akun AWS, amankan Pengguna root akun AWS, aktifkan AWS IAM Identity Center, dan buat pengguna administratif sehingga Anda tidak menggunakan pengguna root untuk tugas sehari-hari.

Amankan Anda Pengguna root akun AWS

1. Masuk ke Konsol Manajemen AWSsebagai pemilik akun dengan memilih pengguna Root dan memasukkan alamat Akun AWS email Anda. Di laman berikutnya, masukkan kata sandi.

   Untuk bantuan masuk dengan menggunakan pengguna root, lihat Masuk sebagai pengguna root di AWS Sign-In Panduan Pengguna.
2. Mengaktifkan autentikasi multi-faktor (MFA) untuk pengguna root Anda.

   Untuk petunjuk, lihat Mengaktifkan perangkat MFA virtual untuk pengguna Akun AWS root (konsol) Anda di Panduan Pengguna IAM.

Buat pengguna dengan akses administratif

1. Aktifkan Pusat Identitas IAM.

   Untuk mendapatkan petunjuk, silakan lihat Mengaktifkan AWS IAM Identity Center di Panduan Pengguna AWS IAM Identity Center .
2. Di Pusat Identitas IAM, berikan akses administratif ke pengguna.

   Untuk tutorial tentang menggunakan Direktori Pusat Identitas IAM sebagai sumber identitas Anda, lihat Mengkonfigurasi akses pengguna dengan default Direktori Pusat Identitas IAM di Panduan AWS IAM Identity Center Pengguna.

Masuk sebagai pengguna dengan akses administratif

- Untuk masuk dengan pengguna Pusat Identitas IAM, gunakan URL masuk yang dikirim ke alamat email saat Anda membuat pengguna Pusat Identitas IAM.

  Untuk bantuan masuk menggunakan pengguna Pusat Identitas IAM, lihat Masuk ke portal AWS akses di Panduan AWS Sign-In Pengguna.

Tetapkan akses ke pengguna tambahan

1. Di Pusat Identitas IAM, buat set izin yang mengikuti praktik terbaik menerapkan izin hak istimewa paling sedikit.

   Untuk petunjuknya, lihat Membuat set izin di Panduan AWS IAM Identity Center Pengguna.
2. Tetapkan pengguna ke grup, lalu tetapkan akses masuk tunggal ke grup.

   Untuk petunjuk, lihat Menambahkan grup di Panduan AWS IAM Identity Center Pengguna.

Buat izin IAM untuk HealthOmics

Untuk menggunakan HealthOmics, konfigurasikan izin IAM berikut:

- Kebijakan berbasis identitas IAM untuk diakses oleh pengguna di akun Anda. HealthOmics
- Peran layanan IAM HealthOmics untuk mengakses sumber daya atas nama Anda.
- Izin di layanan lain (seperti Lake Formation dan Amazon ECR) untuk pengguna Anda dan HealthOmics layanan untuk mengakses sumber daya.

Untuk informasi selengkapnya tentang mengonfigurasi izin IAM, lihat. HealthOmics Izin IAM untuk HealthOmics

Connect dengan repositori kode eksternal

Dengan AWS HealthOmics, Anda dapat mengelola alur kerja Anda menggunakan repositori berbasis Git melalui. AWS CodeConnections HealthOmics menggunakan koneksi ini untuk mengakses repositori kode sumber Anda.

Sebelum bekerja dengan repositori kode eksternal, ikuti panduan Menyiapkan koneksi untuk mulai bekerja dengan. AWS CodeConnections Verifikasi bahwa Anda telah membuat kebijakan dan izin IAM yang tepat untuk akun Anda AWS . Untuk daftar penyedia Git yang didukung dan informasi selengkapnya, lihat Penyedia pihak ketiga untuk apa saya dapat membuat koneksi? .

Buat koneksi

Untuk membuat koneksi dengan penyedia repositori pilihan Anda, ikuti tutorial Buat koneksi.

Menggunakan Amazon Q CLI dengan HealthOmics

Amazon Q CLI menyediakan interaksi bahasa alami dengan AWS HealthOmics, memungkinkan Anda untuk melakukan alur kerja genomik yang kompleks dan tugas analisis menggunakan perintah percakapan. Untuk menggunakan Amazon Q CLI, pastikan untuk mengonfigurasi izin IAM untuk HealthOmics dan layanan lainnya (seperti, CloudWatch Amazon ECR, atau Amazon S3) untuk Amazon Q untuk mengakses sumber daya mereka.

Tutorial AI generatif HealthOmics Agentic memberikan step-by-step panduan untuk mengonfigurasi file konteks dan memungkinkan Amazon Q CLI untuk membuat, menjalankan, dan mengoptimalkan alur kerja Anda. AWS HealthOmics

Memulai dengan HealthOmics

Untuk memulai HealthOmics, pastikan Anda telah mengatur izin dan peran IAM dengan benar. HealthOmics

Menggunakan alur kerja Ready2Run di konsol HealthOmics

Latihan berikut menunjukkan cara menggunakan alur kerja Ready2Run. Alur kerja Ready2Run telah dikonfigurasi sebelumnya dengan parameter dan referensi alat yang Anda butuhkan untuk menjalankan alur kerja. Penerbit alur kerja menyediakan data sampel, sehingga Anda tidak perlu membuat data sendiri.

1. Buka konsol HealthOmics .
2. Pilih panel navigasi (≡) di kiri atas, dan pilih alur kerja Ready2Run.
3. Pada halaman alur kerja Ready2Run, pilih alur kerja. ESMFold for up to 800 residues Konsol membuka halaman detail untuk alur kerja itu.
4. Tab detail memberikan informasi tentang alur kerja. Untuk mencoba alur kerja, di kanan atas halaman pilih Mulai jalankan.
5. Di halaman Tentukan rincian jalankan, masukkan nama jalankan.
6. Masukkan atau pilih lokasi Amazon S3 untuk output run.
7. Untuk Jalankan mode retensi metadata, pilih apakah akan menyimpan atau menghapus data runmeta.
8. Di panel peran Layanan, pilih Buat dan gunakan peran layanan baru.
9. Pilih Berikutnya.
10. Pada halaman Tambahkan nilai parameter, pilih Jalankan alur kerja dengan data uji Ready2Run.
11. Pilih Berikutnya.
12. Tinjau input Anda, lalu pilih Mulai jalankan.

Contoh permintaan untuk Amazon Q CLI

Amazon Q CLI dapat menjalankan alur kerja genom dan tugas analisis dalam AWS HealthOmics menggunakan perintah bahasa alami. Contoh prompt berikut memungkinkan Anda membuat alur kerja, mengelola proses, dan menganalisis data genom. Untuk informasi selengkapnya dan contoh petunjuk HealthOmics, lihat tutorial AI generatif HealthOmics Agentic di. GitHub

- “Buat file alur kerja WDL 1.1 karena `main.wdl` itu akan berjalan. HealthOmics Alur kerja akan mengambil genom referensi sebagai input dan pasangan file fastq. Ini akan mengindeks genom referensi menggunakan BWA dan kemudian memetakan setiap pasangan file fastq ke referensi. Akhirnya gabungkan setiap BAM yang dipetakan ke satu file BAM dan keluarkan file ini dan itu adalah indeks bai.”
- “Package alur kerja dan buat di HealthOmics”
- “Perbarui file input.json untuk menggunakan file nyata dari bucket Amazon S3 sayaomics-my-bucket-with-genome-data" (Berikan lokasi bucket Amazon S3 tertentu, atau biarkan Amazon Q menjelajah)
- “Temukan wadah yang sesuai di repositori Amazon ECR saya dan perbarui inputs.json untuk menggunakan ini”
- “Temukan atau buat peran IAM yang sesuai untuk digunakan saat menjalankan alur kerja”
- “Buat cache run untuk alur kerja saya”
- “Jalankan alur kerja di HealthOmics”
- “Periksa status lari”

> **Warning**
>
> Saat bekerja dengan Amazon Q CLI, tinjau semua konten yang dihasilkan dan tindakan yang diusulkan sebelum melanjutkan. Berikan umpan balik untuk meningkatkan kualitas respons dan agar sesuai dengan kebutuhan alur kerja Anda. Untuk informasi selengkapnya, lihat Pertimbangan keamanan dan praktik terbaik untuk Amazon Q.

Alur kerja pribadi di HealthOmics

Gunakan alur kerja Private saat Anda ingin membuat definisi alur kerja Anda sendiri. Definisi alur kerja menentukan informasi tentang alur kerja dan mendefinisikan tugas alur kerja. Run adalah pemanggilan tunggal alur kerja, dan tugas adalah proses tunggal dalam proses.

HealthOmics mendukung definisi alur kerja yang Anda buat di Workflow Description Language (WDL), Common Workflow Language (CWL), atau Nextflow.

HealthOmics alur kerja menyediakan fitur opsional berikut:

- Run groups— Anda dapat menambahkan alur kerja pribadi ke grup run untuk mengontrol penggunaan komputasi. Grup lari adalah kumpulan alur kerja yang berbagi serangkaian batas sumber daya, seperti proses bersamaan maksimum dan durasi lari maksimum. Anda menetapkan batas ini untuk mengontrol sumber daya komputasi yang digunakan grup run.
- Call caching— Anda dapat menggunakan cache panggilan untuk menyimpan dan menggunakan kembali output tugas, yang menghasilkan durasi berjalan lebih pendek dan menghitung penghematan biaya.
- Sharing workflows— Anda dapat berbagi alur kerja pribadi Anda dengan orang lain Akun AWS di Wilayah yang sama.
- Workflow versions— Anda dapat membuat versi alur kerja pribadi. Pembuatan versi alur kerja memberikan kemampuan bagi pengguna untuk memilih kapan harus mulai menggunakan fungsionalitas yang diperbarui. Versi alur kerja tidak dapat diubah dan memberikan tingkat asal data yang sama dengan alur kerja.

Untuk informasi tentang mengonfigurasi izin IAM untuk alur kerja, lihat. Izin IAM untuk HealthOmics

Untuk contoh lengkap tentang cara menggunakan alur kerja HealthOmics pribadi, lihat Tutorial HealthOmics Github atau tutorial ujung ke ujung AWS workshop untuk. HealthOmics

Topik

- Membuat alur kerja pribadi di HealthOmics
- Pembuatan versi alur kerja di HealthOmics
- Menggunakan HealthOmics run
- Menggunakan grup HealthOmics lari
- Panggilan caching untuk menjalankan HealthOmics

- Berbagi HealthOmics alur kerja

Membuat alur kerja pribadi di HealthOmics

Alur kerja pribadi bergantung pada berbagai sumber daya yang Anda buat dan konfigurasikan sebelum membuat alur kerja:

- Workflow definition file:File definisi alur kerja yang ditulis dalamWDL,Nextflow, atauCWL. Definisi alur kerja menentukan input dan output untuk menjalankan yang menggunakan alur kerja. Ini juga mencakup spesifikasi untuk menjalankan dan menjalankan tugas untuk alur kerja Anda, termasuk persyaratan komputasi dan memori. File definisi alur kerja harus dalam `.zip` format. Untuk informasi selengkapnya, lihat File definisi alur kerja.
  - Anda dapat menggunakan Amazon Q CLI untuk membangun dan memvalidasi file definisi alur kerja Anda di WDL, Nextflow, dan CWL. Untuk informasi selengkapnya, lihat Contoh prompt untuk Amazon Q CLI dan tutorial AI HealthOmics generatif Agentic. GitHub
- (Optional) Parameter template file:File template parameter yang ditulis dalam fileJSON. Buat file untuk menentukan parameter run, atau HealthOmics buat template parameter untuk Anda. Untuk informasi selengkapnya, lihat File templat parameter untuk HealthOmics alur kerja.
- Amazon ECR container images:Buat repositori Amazon ECR pribadi untuk alur kerja. Buat gambar kontainer di repositori pribadi, atau sinkronkan konten registri upstream yang didukung dengan repositori pribadi Amazon ECR Anda.
- (Optional) Sentieon licenses:Minta Sentieon lisensi untuk menggunakan Sentieon perangkat lunak dalam alur kerja pribadi.

Secara opsional, Anda dapat menjalankan linter pada definisi alur kerja sebelum atau setelah Anda membuat alur kerja. linterTopik ini menjelaskan linter yang tersedia di HealthOmics.

Topik

- HealthOmics integrasi alur kerja dengan repositori berbasis Git
- File definisi alur kerja di HealthOmics
- File template parameter untuk HealthOmics alur kerja
- Gambar kontainer untuk alur kerja pribadi
- HealthOmics Alur kerja file README
- Meminta lisensi Sentieon untuk alur kerja pribadi

- Linter alur kerja di HealthOmics
- HealthOmics operasi alur kerja

HealthOmics integrasi alur kerja dengan repositori berbasis Git

Saat membuat alur kerja (atau versi alur kerja), Anda memberikan definisi alur kerja untuk menentukan informasi tentang alur kerja, proses, dan tugas. HealthOmics dapat mengambil definisi alur kerja sebagai arsip.zip (disimpan secara lokal atau dalam bucket Amazon S3), atau dari repositori berbasis Git yang didukung.

HealthOmics Integrasi dengan repositori berbasis Git memungkinkan kemampuan berikut:

- Pembuatan alur kerja langsung dari instans publik, pribadi, dan yang dikelola sendiri.
- Integrasi file README alur kerja dan templat parameter dari repositori.
- Support untuk GitHub, GitLab, dan repositori Bitbucket.

Dengan menggunakan repositori berbasis Git, Anda menghindari langkah-langkah manual mengunduh file definisi alur kerja dan memasukkan file template parameter, membuat arsip.zip, dan kemudian mementaskan arsip ke S3. Ini menyederhanakan pembuatan alur kerja untuk skenario seperti contoh berikut:

1. Anda ingin memulai dengan cepat menggunakan alur kerja open source umum, seperti nf-core. HealthOmicssecara otomatis mengambil semua definisi alur kerja dan memasukkan file template parameter dari repositori nf-core GitHub dan menggunakan file-file ini untuk membuat alur kerja baru Anda.
2. Anda menggunakan alur kerja publik dari GitHub, dan beberapa pembaruan baru tersedia. Anda dapat dengan mudah membuat versi HealthOmics alur kerja baru menggunakan definisi alur kerja yang diperbarui GitHub sebagai sumbernya. Pengguna alur kerja Anda dapat memilih antara alur kerja asli atau versi alur kerja baru yang Anda buat.
3. Tim Anda sedang membangun pipa berpemilik yang tidak bersifat publik. Anda menyimpan kode Anda di repositori git pribadi dan menggunakan definisi alur kerja ini untuk alur kerja Anda. HealthOmics Tim sering memperbarui definisi alur kerja sebagai bagian dari siklus hidup pengembangan alur kerja berulang. Anda dapat dengan mudah membuat versi alur kerja baru seperti yang diperlukan dari repositori pribadi Anda.

Topik

Repositori berbasis Git yang didukung

HealthOmics mendukung repositori publik dan pribadi untuk penyedia berbasis Git berikut:

- GitHub
- GitLab
- Bitbucket

HealthOmics mendukung repositori yang dikelola sendiri untuk penyedia berbasis Git berikut:

- GitHubEnterpriseServer
- GitLabSelfManaged

HealthOmics mendukung penggunaan koneksi lintas akun untuk GitHub, GitLab, dan Bitbucket. Siapkan izin bersama melalui AWS Resource Access Manager. Sebagai contoh, lihat Koneksi bersama dalam panduan CodePipeline pengguna.

Konfigurasikan koneksi ke repositori kode eksternal

Hubungkan alur kerja Anda ke repositori berbasis Git menggunakan AWS. CodeConnection HealthOmics menggunakan koneksi ini untuk mengakses repositori kode sumber Anda.

> **Note**
>
> CodeConnections Layanan AWS tidak tersedia di wilayah il-central-1. Untuk wilayah ini, konfigurasikan layanan us-east-1 untuk membuat alur kerja atau versi alur kerja dari repositori.

Buat koneksi

Sebelum Anda dapat membuat koneksi, ikuti petunjuk dalam Menyiapkan koneksi di Panduan Pengguna Alat Konsol Pengembang.

Untuk membuat sambungan, ikuti petunjuk di Buat sambungan di Panduan Pengguna Alat Konsol Pengembang.

Konfigurasikan otorisasi untuk koneksi

Anda harus mengotorisasi koneksi menggunakan OAuth alur penyedia. Pastikan status koneksi `AVAILABLE` sebelum Anda menggunakannya.

Sebagai contoh, lihat posting blog Cara Membuat AWS HealthOmics Alur Kerja dari Konten di Git.

Mengakses repositori yang dikelola sendiri

Untuk mengatur koneksi ke repositori yang GitLab dikelola sendiri, gunakan Token Akses Pribadi admin saat membuat host. Pembuatan koneksi berikutnya mengakses Oauth dengan akun pelanggan.

Contoh berikut menyiapkan koneksi ke repositori yang GitLab dikelola sendiri:

1. Siapkan akses ke Token Akses Pribadi pengguna admin.

   Untuk menyiapkan PAT di repositori yang dikelola GitLab sendiri, lihat Token akses pribadi di GitLab Dokumen.
2. Membuat host

   a. Arahkan ke> CodePipelinePengaturan> Koneksi.
   b. Pilih tab Hosts dan kemudian pilih Create Host.
   c. Konfigurasi bidang berikut:

      - Masukkan nama host
      - Untuk jenis penyedia, pilih GitLab Self Managed
      - Masukkan URL Host
      - Masukkan informasi VPC jika host didefinisikan dalam VPC

   d. Pilih Buat Host, yang membuat host dalam status PENDING.
   e. Untuk menyelesaikan pengaturan, pilih Siapkan Host.
   f. Masukkan Personal Access Token (PAT) dari pengguna Admin, lalu pilih Lanjutkan.

3. Buat koneksi

    a. Pilih Buat Koneksi pada tab Koneksi.
    b. Untuk jenis penyedia, pilih GitLab dikelola sendiri.
    c. Di bawah Pengaturan Koneksi> Masukkan Nama Koneksi, masukkan URL Host yang sebelumnya Anda buat.
    d. Jika instans yang GitLab dikelola sendiri hanya dapat diakses melalui VPC, konfigurasikan detail VPC.
    e. Pilih Perbarui Koneksi Tertunda. Jendela modal mengarahkan Anda kembali ke halaman GitLab login.
    f. Masukkan nama pengguna dan kata sandi untuk akun pelanggan dan selesaikan proses otorisasi.
    g. Untuk penyiapan pertama kali, pilih Authorize AWS Connector for Gitlab Self Managed.

Kuota yang terkait dengan repositori kode eksternal

Untuk HealthOmics integrasi dengan repositori kode eksternal, ada ukuran maksimum untuk repositori, setiap file repositori, dan setiap file README. Lihat perinciannya di HealthOmics alur kerja kuota ukuran tetap.

Izin IAM yang diperlukan

Tambahkan tindakan berikut ke kebijakan IAM berbasis identitas Anda:

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

File definisi alur kerja di HealthOmics

Anda menggunakan definisi alur kerja untuk menentukan informasi tentang alur kerja, proses, dan tugas dalam proses. Anda membuat definisi alur kerja dalam satu atau beberapa file menggunakan bahasa definisi alur kerja. HealthOmics mendukung definisi alur kerja yang ditulis dalam WDL, Nextflow, atau CWL.

HealthOmics mendukung pilihan berikut untuk definisi alur kerja WDL:

- WDL - Menyediakan mesin WDL yang sesuai spesifikasi.
- WDL lunak - Dirancang untuk menangani alur kerja yang dimigrasikan dari Cromwell. Ini mendukung arahan pelanggan Cromwell dan beberapa logika non-kesesuaian. Lihat perinciannya di Konversi tipe implisit di WDL yang lunak.

Untuk informasi tentang masing-masing bahasa alur kerja, lihat bagian detail khusus bahasa di bawah ini.

Anda menentukan jenis informasi berikut dalam definisi alur kerja:

- Language version— Bahasa dan versi definisi alur kerja.
- Compute and memory— Persyaratan komputasi dan memori untuk tugas-tugas dalam alur kerja.
- Inputs— Lokasi input ke tugas alur kerja. Untuk informasi selengkapnya, lihat HealthOmics jalankan masukan.
- Outputs— Lokasi untuk menyimpan output yang dihasilkan tugas.
- Task resources— Persyaratan komputasi dan memori untuk setiap tugas.
- Accelerators— sumber daya lain yang dibutuhkan tugas, seperti akselerator.

Topik

HealthOmics persyaratan definisi alur kerja

File definisi HealthOmics alur kerja harus memenuhi persyaratan berikut:

- Tugas harus menentukan input/output parameter, repositori kontainer Amazon ECR, dan spesifikasi runtime seperti memori atau alokasi CPU.
- Verifikasi bahwa peran IAM Anda memiliki izin yang diperlukan.
  - Alur kerja Anda memiliki akses ke data input dari AWS sumber daya, seperti Amazon S3.
  - Alur kerja Anda memiliki akses ke layanan repositori eksternal bila diperlukan.
- Deklarasikan file output dalam definisi alur kerja. Untuk menyalin file run perantara ke lokasi output, deklarasikan sebagai output alur kerja.
- Lokasi input dan output harus berada di Wilayah yang sama dengan alur kerja.
- HealthOmics input alur kerja penyimpanan harus dalam `ACTIVE` status. HealthOmics tidak akan mengimpor input dengan `ARCHIVED` status, menyebabkan alur kerja gagal. Untuk informasi tentang input objek Amazon S3, lihat. HealthOmics jalankan masukan
- mainLokasi alur kerja bersifat opsional jika arsip ZIP Anda berisi definisi alur kerja tunggal atau file bernama 'utama'.
  - Contoh jalur: `workflow-definition/main-file.wdl`
- Sebelum Anda membuat alur kerja dari Amazon S3 atau drive lokal Anda, buat arsip zip file definisi alur kerja dan dependensi apa pun, seperti subalur kerja.
- Kami menyarankan Anda mendeklarasikan container Amazon ECR dalam alur kerja sebagai parameter input untuk validasi izin Amazon ECR.

Pertimbangan Nextflow tambahan:

- /bin

  Definisi alur kerja Nextflow dapat mencakup folder/bin dengan skrip yang dapat dieksekusi. Jalur ini memiliki akses read-only plus executable ke tugas. Tugas yang mengandalkan skrip ini harus menggunakan wadah yang dibangun dengan interpreter skrip yang sesuai. Praktik terbaik adalah memanggil penerjemah secara langsung. Contoh:

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

- includeConfig

  Definisi alur kerja berbasis NextFlow dapat menyertakan file nextflow.config yang membantu mengabstraksi definisi parameter atau memproses profil sumber daya. Untuk mendukung pengembangan dan eksekusi pipeline Nextflow di beberapa lingkungan, gunakan konfigurasi HealthOmics -spesifik yang Anda tambahkan ke konfigurasi global menggunakan direktif includeConfig. Untuk mempertahankan portabilitas, konfigurasikan alur kerja untuk menyertakan file hanya saat dijalankan HealthOmics dengan menggunakan kode berikut:

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

- Reports

  HealthOmics tidak mendukung laporan dag, trace, dan eksekusi yang dihasilkan mesin. Anda dapat menghasilkan alternatif untuk laporan penelusuran dan eksekusi menggunakan kombinasi panggilan GetRunTask API GetRun dan.

Pertimbangan CWL tambahan:

- Container image uri interpolation

  HealthOmics memungkinkan properti DockerPull DockerRequirement untuk menjadi ekspresi javascript inline. Contoh:

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

  Ini memungkinkan Anda untuk menentukan gambar kontainer URIs sebagai parameter input ke alur kerja.

- Javascript expressions

  Ekspresi Javascript harus `strict mode` sesuai.

- Operation process

  HealthOmics tidak mendukung proses Operasi CWL.

Dukungan versi untuk bahasa definisi HealthOmics alur kerja

HealthOmics mendukung file definisi alur kerja yang ditulis dalam Nextflow, WDL, atau CWL. Bagian berikut memberikan informasi tentang dukungan HealthOmics versi untuk bahasa-bahasa ini.

Topik

Dukungan versi WDL

HealthOmics mendukung versi 1.0, 1.1, dan versi pengembangan spesifikasi WDL.

Setiap dokumen WDL harus menyertakan pernyataan versi untuk menentukan versi mana (mayor dan minor) dari spesifikasi yang dipatuhi. Untuk informasi selengkapnya tentang versi, lihat pembuatan versi WDL

Versi 1.0 dan 1.1 dari spesifikasi WDL tidak mendukung jenis. `Directory` Untuk menggunakan `Directory` tipe untuk input atau output, atur versi ke development baris pertama file:

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

Dukungan versi CWL

HealthOmics mendukung versi 1.0, 1.1, dan 1.2 dari bahasa CWL.

Anda dapat menentukan versi bahasa dalam file definisi alur kerja CWL. Untuk informasi selengkapnya tentang CWL, lihat panduan pengguna CWL

Dukungan versi Nextflow

HealthOmics mendukung tiga versi stabil Nextflow. Nextflow biasanya merilis versi stabil setiap enam bulan. HealthOmics tidak mendukung rilis “edge” bulanan.

HealthOmics mendukung fitur yang dirilis di setiap versi, tetapi tidak fitur pratinjau.

Versi yang didukung

HealthOmics mendukung versi Nextflow berikut:

- Nextflow v22.04.01 DSL 1 dan DSL 2
- Nextflow v23.10.0 DSL 2 (default)
- Nextflow v24.10.8 DSL 2

Untuk memigrasikan alur kerja Anda ke versi terbaru yang didukung (v24.10.8), ikuti panduan pemutakhiran Nextflow.

Ada beberapa perubahan yang mengganggu saat bermigrasi dari Nextflow v23 ke v24, seperti yang dijelaskan di bagian berikut dari panduan migrasi Nextflow:

- Melanggar perubahan di 24.04
- Melanggar perubahan di 24.10

Mendeteksi dan memproses versi Nextflow

HealthOmics mendeteksi versi DSL dan versi Nextflow yang Anda tentukan. Ini secara otomatis menentukan versi Nextflow terbaik untuk dijalankan berdasarkan input ini.

Versi DSL

HealthOmics mendeteksi versi DSL yang diminta dalam file definisi alur kerja Anda. Misalnya, Anda dapat menentukan:`nextflow.enable.dsl=2`.

HealthOmics mendukung DSL 2 secara default. Ini menyediakan kompatibilitas mundur dengan DSL 1, jika ditentukan dalam file definisi alur kerja Anda.

- Jika Anda menentukan DSL 2, HealthOmics jalankan Nextflow v23.10.0, kecuali Anda menentukan Nextflow v22.04.0 atau v24.10.8.
- Jika Anda menentukan DSL 1, HealthOmics jalankan Nextflow v22.04 DSL1 (satu-satunya versi yang didukung yang menjalankan DSL 1).
- Jika Anda tidak menentukan versi DSL, atau jika tidak HealthOmics dapat mengurai informasi DSL karena alasan apa pun (seperti kesalahan sintaks dalam file definisi alur kerja Anda), HealthOmics default ke DSL 2 dan menjalankan Nextflow v23.10.0.
- Untuk meningkatkan alur kerja Anda dari DSL 1 ke DSL 2 untuk memanfaatkan versi dan fitur perangkat lunak Nextflow terbaru, lihat Migrasi dari DSL 1.

Versi Nextflow

HealthOmics mendeteksi versi Nextflow yang diminta dalam file konfigurasi Nextflow (nextflow.config), jika Anda menyediakan file ini. Kami menyarankan Anda menambahkan `nextflowVersion` klausa di akhir file untuk menghindari penggantian tak terduga dari konfigurasi yang disertakan. Untuk informasi selengkapnya, lihat konfigurasi Nextflow.

Anda dapat menentukan versi Nextflow atau rentang versi menggunakan sintaks berikut:

```
// 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 memproses informasi versi Nextflow sebagai berikut:

- Jika Anda menggunakan = untuk menentukan versi persis yang HealthOmics mendukung, HealthOmics gunakan versi itu.
- Jika Anda menggunakan ! untuk menentukan versi yang tepat atau rentang versi yang tidak didukung, HealthOmics memunculkan pengecualian dan gagal dijalankan. Pertimbangkan untuk menggunakan opsi ini jika Anda ingin ketat dengan permintaan versi dan gagal dengan cepat jika permintaan menyertakan versi yang tidak didukung.
- Jika Anda menentukan rentang versi, HealthOmics gunakan versi terbaru yang didukung dalam rentang tersebut, kecuali rentang tersebut menyertakan v24.10.8. Dalam hal ini, HealthOmics berikan preferensi ke versi sebelumnya. Misalnya, jika rentang mencakup v23.10.0 dan v24.10.8, pilih v23.10.0. HealthOmics
- Jika tidak ada versi yang diminta, atau jika versi yang diminta tidak valid atau tidak dapat diuraikan karena alasan apa pun:

- Jika Anda menentukan DSL 1, HealthOmics jalankan Nextflow v22.04.
- Jika tidak, HealthOmics jalankan Nextflow v23.10.0.

Anda dapat mengambil informasi berikut tentang versi Nextflow yang HealthOmics digunakan untuk setiap proses:

- Log run berisi informasi tentang versi Nextflow aktual yang HealthOmics digunakan untuk menjalankan.
- HealthOmics menambahkan peringatan di log run jika tidak ada kecocokan langsung dengan versi yang Anda minta atau jika perlu menggunakan versi yang berbeda dari yang Anda tentukan.
- Respons terhadap operasi GetRun API menyertakan field (`engineVersion`) dengan versi Nextflow aktual yang HealthOmics digunakan untuk menjalankan. Misalnya:

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

Persyaratan komputasi dan memori untuk tugas HealthOmics

HealthOmics menjalankan tugas alur kerja pribadi Anda dalam instance omics. HealthOmics menyediakan berbagai jenis instance untuk mengakomodasi berbagai jenis tugas. Setiap jenis instans memiliki memori tetap dan konfigurasi vCPU (dan konfigurasi GPU tetap untuk jenis instans komputasi yang dipercepat). Biaya penggunaan instance omics bervariasi tergantung pada jenis instans. Untuk detailnya, lihat halaman HealthOmics Harga.

Untuk tugas dalam alur kerja, Anda menentukan memori yang diperlukan dan v CPUs dalam file definisi alur kerja. Saat tugas alur kerja berjalan, HealthOmics alokasikan instance omics terkecil yang mengakomodasi memori yang diminta dan v. CPUs Misalnya, jika tugas membutuhkan 64 GiB memori dan 8 vCPUs, HealthOmics pilih. `omics.r.2xlarge`

Kami menyarankan Anda meninjau jenis instans dan mengatur v CPUs dan ukuran memori yang Anda minta agar sesuai dengan instans yang paling sesuai dengan kebutuhan Anda. Wadah tugas menggunakan jumlah v CPUs dan ukuran memori yang Anda tentukan dalam file definisi alur kerja Anda, bahkan jika jenis instance memiliki v CPUs dan memori tambahan.

Daftar berikut berisi informasi tambahan tentang vCPU dan alokasi memori:

- Alokasi sumber daya kontainer adalah batas sulit. Jika tugas kehabisan memori atau mencoba menggunakan v tambahanCPUs , tugas menghasilkan log kesalahan dan keluar.

- Jika Anda tidak menentukan persyaratan komputasi atau memori, HealthOmics pilih omics.c.large dan default ke konfigurasi dengan 1 vCPU dan 1 GiB memori.
- Konfigurasi minimum yang dapat Anda minta adalah 1 vCPU dan 1 GiB memori.
- Jika Anda menentukan vCPUs, memori, atau GPUs yang melebihi jenis instans yang didukung, HealthOmics melempar pesan kesalahan dan alur kerja gagal validasi
- Jika Anda menentukan satuan pecahan, HealthOmics bulatkan ke bilangan bulat terdekat.
- HealthOmics menyimpan sejumlah kecil memori (5%) untuk agen manajemen dan logging, sehingga alokasi memori penuh mungkin tidak selalu tersedia untuk aplikasi dalam tugas.
- HealthOmics mencocokkan jenis instance agar sesuai dengan persyaratan komputasi dan memori yang Anda tentukan, dan dapat menggunakan campuran generasi perangkat keras. Untuk alasan ini, mungkin ada beberapa variasi kecil dalam waktu menjalankan tugas untuk tugas yang sama.

Topik ini memberikan detail tentang jenis instance yang HealthOmics mendukung.

Topik

> **Note**
>
> Untuk instans standar, komputasi, dan memori yang dioptimalkan, tingkatkan ukuran bandwidth instans jika instance memerlukan throughput yang lebih tinggi. Instans Amazon EC2 dengan kurang dari 16 vCPU (ukuran 4xl dan lebih kecil) dapat mengalami ledakan throughput. Untuk informasi selengkapnya tentang throughput instans Amazon EC2, lihat Bandwidth instans yang tersedia Amazon EC2.

Jenis instans standar

Untuk tipe instans standar, konfigurasi bertujuan untuk keseimbangan daya komputasi dan memori.

HealthOmics mendukung instance 32xlarge dan 48xlarge di wilayah ini: US West (Oregon) dan US East (Virginia N.).

Instans	Jumlah v CPUs	Memori
omics.m.large	2	8 GiB
omics.m.xlarge	4	16 GiB
omics.m.2xlarge	8	32 GiB
omics.m.4xlarge	16	64 GiB
omics.m.8xlarge	32	128 GiB
omics.m.12xlarge	48	192 GiB
omics.m.16xlarge	64	256 GiB
omics.m.24xlarge	96	384 GiB
omics.m.32xlarge	128	512 GiB
omics.m.48xlarge	192	768 GiB

Instans yang dioptimalkan untuk komputasi

Untuk tipe instans yang dioptimalkan komputasi, konfigurasi memiliki daya komputasi yang lebih besar dan memori yang lebih sedikit.

HealthOmics mendukung instance 32xlarge dan 48xlarge di wilayah ini: US West (Oregon) dan US East (Virginia N.).

Instans	Jumlah v CPUs	Memori
omics.c.large	2	4 GiB
omics.c.xlarge	4	8 GiB
omics.c.2xlarge	8	16 GiB
omics.c.4xlarge	16	32 GiB

Instans	Jumlah v CPUs	Memori
omics.c.8xlarge	32	64 GiB
omics.c.12xlarge	48	96 GiB
omics.c.16xlarge	64	128 GiB
omics.c.24xlarge	96	192 GiB
omics.c.32xlarge	128	256 GiB
omics.c.48xlarge	192	384 GiB

Instans yang dioptimalkan untuk memori

Untuk jenis instans yang dioptimalkan memori, konfigurasi memiliki daya komputasi yang lebih sedikit dan lebih banyak memori.

HealthOmics mendukung instance 32xlarge dan 48xlarge di wilayah ini: US West (Oregon) dan US East (Virginia N.).

Instans	Jumlah v CPUs	Memori
omics.r.large	2	16 GiB
omics.r.xlarge	4	32 GiB
omics.r.2xlarge	8	64 GiB
omics.r.4xlarge	16	128 GiB
omics.r.8xlarge	32	256 GiB
omics.r.12xlarge	48	384 GiB
omics.r.16xlarge	64	512 GiB
omics.r.24xlarge	96	768 GiB

Instans	Jumlah v CPUs	Memori
omics.r.32xlarge	128	1024 GiB
omics.r.48xlarge	192	1536 GiB

Instans komputasi yang dipercepat

Anda dapat secara opsional menentukan sumber daya GPU untuk setiap tugas dalam alur kerja, sehingga HealthOmics mengalokasikan instance komputasi yang dipercepat untuk tugas tersebut. Untuk informasi tentang cara menentukan informasi GPU dalam file definisi alur kerja, lihat. Akselerator tugas dalam definisi HealthOmics alur kerja

Jika Anda menentukan GPU yang mendukung beberapa jenis instans, HealthOmics pilih jenis instans berdasarkan ketersediaan. Jika kedua jenis instans tersedia, HealthOmics berikan preferensi ke instance biaya yang lebih rendah.

Instans G4 tidak didukung di Wilayah Israel (Tel Aviv). Instans G5 tidak didukung di Wilayah Asia Pasifik (Singapura).

Topik

- Jenis instans G6 dan G6e
- Instans G4 dan G5

Jenis instans G6 dan G6e

HealthOmics mendukung konfigurasi instans komputasi akselerasi G6 berikut. Semua instans omics.g6 menggunakan Nvidia L4 atau Nvidia L4 A10G. GPUs

HealthOmics mendukung contoh G6 dan G6e di wilayah ini: US West (Oregon) dan US East (Virginia N.).

Instans	Jumlah v CPUs	Memori	Jumlah GPUs	Memori GPU	
omics.g6.xlarge	4	16 GiB	1	24 GiB	

Instans	Jumlah v CPUs	Memori	Jumlah GPUs	Memori GPU	
omics.g6.2xlarge	8	32 GiB	1	24 GiB	
omics.g6.4xlarge	16	64 GiB	1	24 GiB	
omics.g6.8xlarge	32	128 GiB	1	24 GiB	
omics.g6.12xlarge	48	192 GiB	4	96 GiB	
omics.g6.16xlarge	64	256 GiB	1	24 GiB	
omics.g6.24xlarge	96	384 GiB	4	96 GiB	

Semua instans omics.g6e menggunakan Nvidia L40. GPUs

Instans	Jumlah v CPUs	Memori	Jumlah GPUs	Memori GPU	
omics.g6e.xlarge	4	32 GiB	1	48 GiB	
omics.g6e.2xlarge	8	64 GiB	1	48 GiB	
omics.g6e.4xlarge	16	128 GiB	1	48 GiB	
omics.g6e.8xlarge	32	256 GiB	1	48 GiB	

Instans	Jumlah v CPUs	Memori	Jumlah GPUs	Memori GPU	
omics.g6e .12xlarge	48	384 GiB	4	192 GiB	
omics.g6e .16xlarge	64	512 GiB	1	48 GiB	
omics.g6e .24xlarge	96	768 GiB	4	192 GiB	

Instans G4 dan G5

HealthOmics mendukung konfigurasi instans komputasi akselerasi G4 dan G5 berikut.

Semua instans omics.g5 menggunakan Nvidia L4 A10G, Nvidia Tesla A10G, atau Nvidia Tesla T4 A10G. GPUs

Instans	Jumlah v CPUs	Memori	Jumlah GPUs	Memori GPU	
omics.g5. xlarge	4	16 GiB	1	24 GiB	
omics.g5. 2xlarge	8	32 GiB	1	24 GiB	
omics.g5. 4xlarge	16	64 GiB	1	24 GiB	
omics.g5. 8xlarge	32	128 GiB	1	24 GiB	
omics.g5. 12xlarge	48	192 GiB	4	96 GiB	
omics.g5. 16xlarge	64	256 GiB	1	24 GiB	

Instans	Jumlah v CPUs	Memori	Jumlah GPUs	Memori GPU	
omics.g5.24xlarge	96	384 GiB	4	96 GiB	

Semua instans omics.g4dn menggunakan Nvidia Tesla T4 atau Nvidia Tesla T4 A10G. GPUs

Instans	Jumlah v CPUs	Memori	Jumlah GPUs	Memori GPU	
omics.g4dn.xlarge	4	16 GiB	1	16 GiB	
omics.g4dn.2xlarge	8	32 GiB	1	16 GiB	
omics.g4dn.4xlarge	16	64 GiB	1	16 GiB	
omics.g4dn.8xlarge	32	128 GiB	1	16 GiB	
omics.g4dn.12xlarge	48	192 GiB	4	64 GiB	
omics.g4dn.16xlarge	64	256 GiB	1	24 GiB	

Output tugas dalam definisi HealthOmics alur kerja

Anda menentukan output tugas dalam definisi alur kerja. Secara default, HealthOmics buang semua file tugas perantara saat alur kerja selesai. Untuk mengekspor file perantara, Anda mendefinisikannya sebagai output.

Jika Anda menggunakan caching panggilan, HealthOmics menyimpan output tugas ke cache, termasuk file perantara yang Anda tentukan sebagai output.

Topik berikut mencakup contoh definisi tugas untuk setiap bahasa definisi alur kerja.

Topik

Output tugas untuk WDL

Untuk definisi alur kerja yang ditulis dalam WDL, tentukan output Anda di bagian alur kerja tingkat atas. outputs

HealthOmics

Topik

Output tugas untuk STDOUT

Contoh ini membuat tugas bernama `SayHello` yang menggemakan konten STDOUT ke file output tugas. stdoutFungsi WDL menangkap konten STDOUT (dalam contoh ini, string input Hello World!) dalam filestdout_file.

Karena HealthOmics membuat log untuk semua konten STDOUT, output juga muncul di CloudWatch Log, bersama dengan informasi logging STDERR lainnya untuk tugas tersebut.

```
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()
    }
}
```

Output tugas untuk STDERR

Contoh ini membuat tugas bernama `SayHello` yang menggemakan konten STDERR ke file output tugas. stderrFungsi WDL menangkap konten STDERR (dalam contoh ini, string input Hello World!) dalam filestderr_file.

Karena HealthOmics membuat log untuk semua konten STDERR, output akan muncul di CloudWatch Log, bersama dengan informasi logging STDERR lainnya untuk tugas tersebut.

```
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()
    }
}
```

Output tugas ke file

Dalam contoh ini, SayHello tugas membuat dua file (message.txt dan info.txt) dan secara eksplisit menyatakan file-file ini sebagai output bernama (message_file dan 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"
    }
}
```

Output tugas ke array file

Dalam contoh ini, `GenerateGreetings` tugas menghasilkan array file sebagai output tugas. Tugas secara dinamis menghasilkan satu file ucapan untuk setiap anggota array input. `names` Karena nama file tidak diketahui sampai runtime, definisi output menggunakan fungsi WDL glob () untuk menampilkan semua file yang cocok dengan pola. `*_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")
    }
}
```

Output tugas untuk Nextflow

Untuk definisi alur kerja yang ditulis dalam Alur Berikutnya, tentukan direktif PublishDir untuk mengekspor konten tugas ke bucket Amazon S3 keluaran Anda. Setel nilai publishDir ke. `/mnt/workflow/pubdir`

HealthOmics Untuk mengekspor file ke Amazon S3, file harus ada di direktori ini.

Jika tugas menghasilkan sekelompok file output untuk digunakan sebagai input untuk tugas berikutnya, kami sarankan Anda mengelompokkan file-file ini dalam direktori dan memancarkan direktori sebagai output tugas. Menghitung setiap file individu dapat mengakibatkan kemacetan I/O di sistem file yang mendasarinya. Misalnya:

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

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

Output tugas untuk CWL

Untuk definisi alur kerja yang ditulis dalam CWL, Anda dapat menentukan output tugas menggunakan tugas. `CommandLineTool` Bagian berikut menunjukkan contoh `CommandLineTool` tugas yang menentukan berbagai jenis output.

Topik

- Output tugas untuk STDOUT
- Output tugas untuk STDERR
- Output tugas ke file
- Output tugas ke array file

Output tugas untuk STDOUT

Contoh ini membuat `CommandLineTool` tugas yang menggemakan konten STDOUT ke file keluaran teks bernama. output.txt Misalnya, jika Anda memberikan masukan berikut, output tugas yang dihasilkan adalah Hello World! dalam output.txt file.

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

`outputs`Direktif menentukan bahwa nama output adalah example_out dan jenisnya adalah. `stdout` Untuk tugas hilir untuk mengkonsumsi output dari tugas ini, itu akan merujuk ke output sebagai`example_out`.

Karena HealthOmics membuat log untuk semua konten STDERR dan STDOUT, output juga muncul di CloudWatch Log, bersama dengan informasi logging STDERR lainnya untuk tugas tersebut.

```
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
```

Output tugas untuk STDERR

Contoh ini membuat `CommandLineTool` tugas yang menggemakan konten STDERR ke file keluaran teks bernama. stderr.txt Tugas memodifikasi `baseCommand` sehingga `echo` menulis ke STDERR (bukan STDOUT).

`outputs`Direktif menentukan bahwa nama output adalah stderr_out dan jenisnya adalah. `stderr`

Karena HealthOmics membuat log untuk semua konten STDERR dan STDOUT, output akan muncul di CloudWatch Log, bersama dengan informasi logging STDERR lainnya untuk tugas tersebut.

```
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
```

Output tugas ke file

Contoh ini membuat `CommandLineTool` tugas yang membuat arsip tar terkompresi dari file input. Anda memberikan nama arsip sebagai parameter input (archive_name).

outputsDirektif menentukan bahwa jenis `archive_file` output adalah`File`, dan menggunakan referensi ke parameter input untuk mengikat `archive_name` ke file output.

```
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
```

Output tugas ke array file

Dalam contoh ini, `CommandLineTool` tugas membuat array file menggunakan `touch` perintah. Perintah menggunakan string dalam parameter `files-to-create` input untuk memberi nama file.

Perintah mengeluarkan array file. Array mencakup file apa pun di direktori kerja yang cocok dengan `glob` pola. Contoh ini menggunakan pola wildcard (“*”) yang cocok dengan semua file.

```
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
```

Sumber daya tugas dalam definisi HealthOmics alur kerja

Dalam definisi alur kerja, tentukan hal berikut untuk setiap tugas:

- Gambar kontainer untuk tugas. Untuk informasi selengkapnya, lihat Gambar kontainer untuk alur kerja pribadi.
- Jumlah CPUs dan memori yang dibutuhkan untuk tugas tersebut. Untuk informasi selengkapnya, lihat Persyaratan komputasi dan memori untuk tugas HealthOmics.

HealthOmics mengabaikan spesifikasi penyimpanan per tugas apa pun. HealthOmics menyediakan penyimpanan run yang dapat diakses oleh semua tugas dalam proses. Untuk informasi selengkapnya, lihat Jalankan jenis penyimpanan dalam HealthOmics alur kerja.

WDL

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

Untuk alur kerja WDL, HealthOmics mencoba hingga dua percobaan ulang untuk tugas yang gagal karena kesalahan layanan (permintaan API mengembalikan kode status HTTP 5XX). Untuk informasi selengkapnya tentang percobaan ulang tugas, lihatTugas Mencoba Ulang.

Anda dapat memilih keluar dari perilaku coba lagi dengan menentukan konfigurasi berikut untuk tugas dalam file definisi WDL:

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

NextFlow

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

CWL

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

```
        ramMax: 4000 # specified in mebibytes
```

Akselerator tugas dalam definisi HealthOmics alur kerja

Dalam definisi alur kerja, Anda dapat menentukan spesifikasi akselerator GPU untuk tugas secara opsional. HealthOmics mendukung nilai spesifikasi akselerator berikut, bersama dengan jenis instance yang didukung:

Spesifikasi akselerator	Jenis contoh Healthomics				
nvidia-tesla-t4	G4				
nvidia-tesla-t4-a10g	G4 dan G5				
nvidia-tesla-a10g	G5				
nvidia-l4-a10g	G5 dan G6				
nvidia-l4	G6				
nvidia-l40-an	G6e				

Jika Anda menentukan jenis akselerator yang mendukung beberapa jenis instans, HealthOmics pilih jenis instans berdasarkan kapasitas yang tersedia. Jika kedua jenis instans tersedia, HealthOmics berikan preferensi ke instance biaya yang lebih rendah.

Untuk detail tentang jenis instance, lihatInstans komputasi yang dipercepat.

Dalam contoh berikut, definisi alur kerja menentukan `nvidia-l4` sebagai akselarator:

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"
```

Spesifikasi definisi alur kerja WDL

Topik berikut memberikan rincian tentang jenis dan arahan yang tersedia untuk definisi alur kerja WDL di. HealthOmics

Topik

- Metadata tugas di WDL
- Contoh definisi alur kerja WDL

Konversi tipe implisit di WDL yang lunak

HealthOmics mendukung konversi tipe implisit dalam file input.json dan definisi alur kerja. Untuk menggunakan casting tipe implisit, tentukan mesin alur kerja sebagai WDL lunak saat Anda membuat alur kerja. WDL lunak dirancang untuk menangani alur kerja yang dimigrasikan dari Cromwell. Ini mendukung arahan pelanggan Cromwell dan beberapa logika non-kesesuaian.

WDL lunak mendukung konversi tipe untuk item berikut dalam daftar pengecualian terbatas WDL:

- Mengapung ke Int, di mana paksaan tidak menghasilkan kehilangan presisi (seperti 1.0 peta ke 1).
- String ke Int/Float, di mana paksaan tidak menghasilkan kehilangan presisi.
- Petakan [W, X] ke Array [Pair [Y, Z]], dalam kasus di mana W dapat dipaksakan ke Y dan X dapat dipaksakan ke Z.
- Array [Pasangkan [W, X]] ke Peta [Y, Z], dalam kasus di mana W dapat dipaksakan ke Y dan X dapat dipaksakan ke Z (seperti 1,0 peta ke 1).

Untuk menggunakan casting tipe implisit, tentukan mesin alur kerja sebagai WDL_LENIENT saat Anda membuat alur kerja atau versi alur kerja.

Di konsol, parameter mesin alur kerja diberi nama Bahasa. Dalam API, parameter workflow engine diberi nama engine. Untuk informasi selengkapnya, lihat Buat alur kerja pribadi atau Buat versi alur kerja.

Definisi namespace di input.json

HealthOmics mendukung variabel yang sepenuhnya memenuhi syarat di input.json. Misalnya, jika Anda mendeklarasikan dua variabel input bernama number1 dan number2 dalam alur kerja: SumWorkflow

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

Anda dapat menggunakannya sebagai variabel yang sepenuhnya memenuhi syarat di input.json:

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

Tipe primitif di WDL

Tabel berikut menunjukkan bagaimana input dalam peta WDL ke tipe primitif yang cocok. HealthOmics menyediakan dukungan terbatas untuk pemaksaan tipe, jadi sebaiknya Anda menyetel tipe eksplisit.

Jenis primitif

Jenis WDL	Jenis JSON	Contoh WDL	Contoh kunci dan nilai JSON	Catatan
`Boolean`	`boolean`	`Boolean b`	`"b": true`	Nilainya harus huruf kecil dan tidak dikutip.
`Int`	`integer`	`Int i`	`"i": 7`	Harus tidak dikutip.
`Float`	`number`	`Float f`	`"f": 42.2`	Harus tidak dikutip.
`String`	`string`	`String s`	`"s": "characters"`	String JSON yang merupakan URI harus dipetakan ke file WDL untuk diimpor.
`File`	`string`	`File f`	`"f": "s3://amzn-s3-demo-bucket1/`	Amazon S3 dan HealthOmics penyimpanan URIs diimpor selama peran

Jenis WDL	Jenis JSON	Contoh WDL	Contoh kunci dan nilai JSON	Catatan
			`path/to/file"`	IAM yang disediakan untuk alur kerja memiliki akses baca ke objek ini. Tidak ada skema URI lain yang didukung (seperti`file://`,`https:` dan`ftp://`). URI harus menentukan objek. Itu tidak bisa menjadi direktori yang berarti tidak dapat diakhiri dengan`/`.

Jenis WDL	Jenis JSON	Contoh WDL	Contoh kunci dan nilai JSON	Catatan
`Directory`	`string`	`Directory d`	`"d": "s3://bucket/path/"`	`Directory` Jenis ini tidak termasuk dalam WDL 1.0 atau 1.1, jadi Anda harus menambahkan `version development` ke header file WDL. URI harus berupa URI Amazon S3 dan dengan awalan yang diakhiri dengan '/'. Semua isi direktori akan disalin secara rekursif ke alur kerja sebagai unduhan tunggal. `Directory` Seharusnya hanya berisi file yang terkait dengan alur kerja.

Jenis kompleks di WDL

Tabel berikut menunjukkan bagaimana input dalam peta WDL ke jenis JSON kompleks yang cocok. Tipe kompleks dalam WDL adalah struktur data yang terdiri dari tipe primitif. Struktur data seperti daftar akan dikonversi ke array.

Jenis kompleks

Jenis WDL	Jenis JSON	Contoh WDL	Contoh kunci dan nilai JSON	Catatan
`Array`	`array`	`Array[Int] nums`	`“nums": [1, 2, 3]`	Anggota array harus mengikuti format tipe array WDL.
`Pair`	`object`	`Pair[String, Int] str_to_i`	`“str_to_i": {"left": "0", "right": 1}`	Setiap nilai pasangan harus menggunakan format JSON dari jenis WDL yang cocok.
`Map`	`object`	`Map[Int, String] int_to_string`	`"int_to_string": { 2: "hello", 1: "goodbye" }`	Setiap entri di peta harus menggunakan format JSON dari jenis WDL yang cocok.
`Struct`	`object`	`struct SampleBamAndIndex { String sample_name File bam File bam_index`	`"b_and_i": { "sample_name": "NA12878", "bam": "s3://amzn-s3-demo-bucket1/`	Nama-nama anggota struct harus sama persis dengan nama-nama kunci objek JSON. Setiap nilai harus menggunakan

Jenis WDL	Jenis JSON	Contoh WDL	Contoh kunci dan nilai JSON	Catatan
		`} SampleBam AndIndex b_and_i`	`NA12878.b am", "bam_inde x": "s3:// amzn- s3-demo- bucket1/ NA12878.b am.bai" }`	format JSON dari jenis WDL yang cocok.
`Object`	N/A	N/A	N/A	`Object`Jenis WDL sudah usang dan harus diganti dengan `Struct` dalam semua kasus.

Arahan di WDL

HealthOmics mendukung arahan berikut di semua versi WDL yang mendukung. HealthOmics

Konfigurasikan sumber daya GPU

HealthOmics mendukung atribut runtime acceleratorType dan acceleratorCount dengan semua instance GPU yang didukung. HealthOmics juga mendukung alias bernama gpuType dangpuCount, yang memiliki fungsi yang sama dengan rekan-rekan akselerator mereka. Jika definisi WDL berisi kedua arahan, HealthOmics gunakan nilai akselerator.

Contoh berikut menunjukkan cara menggunakan arahan ini:

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

Konfigurasikan coba lagi tugas untuk kesalahan layanan

HealthOmics mendukung hingga dua percobaan ulang untuk tugas yang gagal karena kesalahan layanan (kode status HTTP 5XX). Anda dapat mengonfigurasi jumlah maksimum percobaan ulang (1 atau 2) dan Anda dapat memilih keluar dari percobaan ulang untuk kesalahan layanan. Secara default, HealthOmics mencoba maksimal dua percobaan ulang.

Contoh berikut ditetapkan `preemptible` untuk memilih keluar dari percobaan ulang untuk kesalahan layanan:

```
{
  preemptible: 0
}
```

Untuk informasi selengkapnya tentang percobaan ulang tugas HealthOmics, lihatTugas Mencoba Ulang.

Konfigurasikan tugas coba lagi untuk kehabisan memori

HealthOmics mendukung percobaan ulang untuk tugas yang gagal karena kehabisan memori (kode keluar wadah 137, kode status HTTP 4XX). HealthOmics menggandakan jumlah memori untuk setiap upaya coba lagi.

Secara default, HealthOmics tidak mencoba lagi untuk jenis kegagalan ini. Gunakan `maxRetries` arahan untuk menentukan jumlah maksimum percobaan ulang.

Contoh berikut ditetapkan `maxRetries` ke 3, sehingga HealthOmics upaya maksimal empat upaya untuk menyelesaikan tugas (upaya awal ditambah tiga percobaan ulang):

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

> **Note**
>
> Coba lagi tugas untuk kehabisan memori membutuhkan GNU findutils 4.2.3+. Wadah HealthOmics gambar default menyertakan paket ini. Jika Anda menentukan gambar kustom dalam definisi WDL Anda, pastikan bahwa gambar tersebut menyertakan GNU findutils 4.2.3+.

Konfigurasikan kode pengembalian

Atribut ReturnCodes menyediakan mekanisme untuk menentukan kode pengembalian, atau satu set kode pengembalian, yang menunjukkan keberhasilan pelaksanaan tugas. Mesin WDL menghormati kode pengembalian yang Anda tentukan di bagian runtime definisi WDL, dan menetapkan status tugas yang sesuai.

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

HealthOmics juga mendukung alias bernama continueOnReturnCode, yang memiliki kemampuan yang sama dengan ReturnCodes. Jika Anda menentukan kedua atribut, HealthOmics menggunakan nilai ReturnCodes.

Metadata tugas di WDL

HealthOmics mendukung opsi metadata berikut untuk tugas WDL.

Nonaktifkan caching tingkat tugas dengan atribut volatile

Atribut volatile memungkinkan Anda untuk menonaktifkan caching panggilan untuk tugas-tugas tertentu dalam alur kerja WDL Anda. Ketika tugas ditandai sebagai volatile, itu akan selalu mengeksekusi dan tidak pernah menggunakan hasil cache, bahkan ketika caching diaktifkan untuk dijalankan.

Tambahkan atribut volatile ke bagian meta definisi tugas Anda:

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

    input {
        String input_file
    }

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

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

Contoh definisi alur kerja WDL

Contoh berikut menunjukkan definisi alur kerja pribadi untuk mengkonversi dari `CRAM` ke `BAM` dalam WDL. `BAM`Alur kerja `CRAM` to mendefinisikan dua tugas dan menggunakan alat dari `genomes-in-the-cloud` wadah, yang ditampilkan dalam contoh dan tersedia untuk umum.

Contoh berikut menunjukkan cara menyertakan wadah Amazon ECR sebagai parameter. Ini memungkinkan HealthOmics untuk memverifikasi izin akses ke penampung Anda sebelum memulai menjalankan proses.

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

Contoh berikut menunjukkan cara menentukan file mana yang akan digunakan dalam proses Anda, saat file berada di bucket Amazon S3.

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

Jika Anda ingin menentukan file dari toko urutan, tunjukkan bahwa seperti yang ditunjukkan dalam contoh berikut, menggunakan URI untuk penyimpanan urutan.

```
{
    "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"
}
```

Anda kemudian dapat menentukan alur kerja Anda di WDL seperti yang ditunjukkan pada contoh berikut.

```
 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}"
    }
}
```

Spesifikasi definisi alur kerja alur berikutnya

HealthOmics mendukung DSL1 Nextflow dan. DSL2 Lihat perinciannya di Dukungan versi Nextflow.

Nextflow DSL2 didasarkan pada bahasa pemrograman Groovy, sehingga parameternya dinamis dan pemaksaan tipe dimungkinkan menggunakan aturan yang sama seperti Groovy. Parameter dan nilai yang disediakan oleh input JSON tersedia di parameter (`params`) peta alur kerja.

Topik

- Gunakan plugin nf-schema dan nf-validation
- Tentukan penyimpanan URIs
- Arahan Nextflow
- Ekspor konten tugas

Gunakan plugin nf-schema dan nf-validation

> **Note**
>
> Ringkasan HealthOmics dukungan untuk plugin:
>
> - v22.04 - tidak ada dukungan untuk plugin
> - v23.10 — mendukung dan `nf-schema` `nf-validation`
> - v24.10 - mendukung `nf-schema`

HealthOmics menyediakan dukungan berikut untuk plugin Nextflow:

- Untuk Nextflow v23.10, HealthOmics pra-instal plugin nf-validation @1 .1.1.
- Untuk Nextflow v23.10 dan yang lebih baru, HealthOmics pra-instal plugin nf-schema @2 .3.0.
- Anda tidak dapat mengambil plugin tambahan selama menjalankan alur kerja. HealthOmics mengabaikan versi plugin lain yang Anda tentukan dalam `nextflow.config` file.
- Untuk Nextflow v24 dan yang lebih tinggi, `nf-schema` adalah versi baru dari plugin yang tidak digunakan lagi. `nf-validation` Untuk informasi selengkapnya, lihat skema nf di repositori Nextflow. GitHub

Tentukan penyimpanan URIs

Ketika Amazon S3 atau HealthOmics URI digunakan untuk membuat file atau objek jalur Nextflow, itu membuat objek yang cocok tersedia untuk alur kerja, selama akses baca diberikan. Penggunaan awalan atau direktori diizinkan untuk Amazon S3. URIs Sebagai contoh, lihat Format parameter masukan Amazon S3.

HealthOmics sebagian mendukung penggunaan pola glob di Amazon URIs S3 atau Storage HealthOmics . URIs Gunakan pola Glob dalam definisi alur kerja untuk pembuatan `path` atau `file` saluran. Untuk perilaku yang diharapkan dan kasus yang tepat, lihatNextflow Penanganan pola Glob di input Amazon S3.

Arahan Nextflow

Anda mengonfigurasi arahan Nextflow dalam file konfigurasi Nextflow atau definisi alur kerja. Daftar berikut menunjukkan urutan prioritas yang HealthOmics digunakan untuk menerapkan pengaturan konfigurasi, dari prioritas terendah hingga tertinggi:

1. Konfigurasi global dalam file konfigurasi.
2. Bagian tugas dari definisi alur kerja.
3. Selektor khusus tugas dalam file konfigurasi.

Topik

- Strategi coba lagi tugas menggunakan errorStrategy
- Upaya coba lagi tugas menggunakan maxRetries
- Menyisih dari tugas coba lagi menggunakan omicsRetryOn5xx
- Durasi tugas menggunakan time direktif

Strategi coba lagi tugas menggunakan **`errorStrategy`**

Gunakan `errorStrategy` direktif untuk menentukan strategi kesalahan tugas. Secara default, ketika tugas kembali dengan indikasi kesalahan (status keluar bukan nol), tugas berhenti dan HealthOmics mengakhiri seluruh proses. Jika Anda menyetel `errorStrategy` ke`retry`, HealthOmics coba satu coba lagi tugas yang gagal. Untuk menambah jumlah percobaan ulang, lihatUpaya coba lagi tugas menggunakan `maxRetries`.

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

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

Untuk informasi tentang cara HealthOmics menangani percobaan ulang tugas selama proses, lihatTugas Mencoba Ulang.

Upaya coba lagi tugas menggunakan **`maxRetries`**

Secara default, HealthOmics tidak mencoba mencoba ulang tugas yang gagal, atau mencoba satu percobaan lagi jika Anda mengonfigurasi. `errorStrategy` Untuk meningkatkan jumlah percobaan ulang maksimum, atur `errorStrategy` ke `retry` dan konfigurasikan jumlah maksimum percobaan ulang menggunakan arahan`maxRetries`.

Contoh berikut menetapkan jumlah maksimum percobaan ulang ke 3 dalam konfigurasi global.

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

Contoh berikut menunjukkan cara mengatur `maxRetries` di bagian tugas definisi alur kerja.

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

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

Contoh berikut menunjukkan cara menentukan konfigurasi khusus tugas dalam file konfigurasi Nextflow, berdasarkan pemilih nama atau label.

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

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

Menyisih dari tugas coba lagi menggunakan **`omicsRetryOn5xx`**

Untuk Nextflow v23 dan v24, HealthOmics mendukung percobaan ulang tugas jika tugas gagal karena kesalahan layanan (kode status HTTP 5XX). Secara default, HealthOmics mencoba hingga dua percobaan ulang dari tugas yang gagal.

Anda dapat mengonfigurasi `omicsRetryOn5xx` untuk memilih keluar dari percobaan ulang tugas untuk kesalahan layanan. Untuk informasi selengkapnya tentang coba lagi tugas HealthOmics, lihatTugas Mencoba Ulang.

Contoh berikut mengkonfigurasi `omicsRetryOn5xx` dalam konfigurasi global untuk memilih keluar dari tugas coba lagi.

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

Contoh berikut menunjukkan cara mengkonfigurasi `omicsRetryOn5xx` di bagian tugas definisi alur kerja.

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

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

Contoh berikut menunjukkan cara menyetel `omicsRetryOn5xx` konfigurasi khusus tugas dalam file konfigurasi Nextflow, berdasarkan pemilih nama atau label.

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

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

Durasi tugas menggunakan **`time`** direktif

HealthOmics menyediakan kuota yang dapat disesuaikan (lihatHealthOmics kuota layanan) untuk menentukan durasi maksimum untuk menjalankan. Untuk alur kerja Nextflow v23 dan v24, Anda juga dapat menentukan durasi tugas maksimum menggunakan direktif Nextflow. `time`

Selama pengembangan alur kerja baru, menyetel durasi tugas maksimum membantu Anda menangkap tugas runaway dan tugas yang berjalan lama.

Untuk informasi selengkapnya tentang direktif waktu Nextflow, lihat direktif waktu di referensi Nextflow.

HealthOmics memberikan dukungan berikut untuk arahan waktu Nextflow:

1. HealthOmics mendukung granularitas 1 menit untuk arahan waktu. Anda dapat menentukan nilai antara 60 detik dan nilai durasi lari maksimum.
2. Jika Anda memasukkan nilai kurang dari 60, HealthOmics bulatkan hingga 60 detik. Untuk nilai di atas 60, HealthOmics bulatkan ke menit terdekat.

3. Jika alur kerja mendukung percobaan ulang untuk tugas, HealthOmics coba ulang tugas jika waktu habis.
4. Jika tugas habis waktu (atau waktu coba lagi terakhir habis), HealthOmics batalkan tugas. Operasi ini dapat memiliki durasi satu hingga dua menit.
5. Pada batas waktu tugas, HealthOmics menetapkan status run dan task menjadi gagal, dan membatalkan tugas lain dalam proses (untuk tugas dalam status Mulai, Pending, atau Running). HealthOmics mengekspor output dari tugas yang diselesaikan sebelum batas waktu ke lokasi output S3 yang Anda tentukan.
6. Waktu yang dihabiskan tugas dalam status tertunda tidak dihitung terhadap durasi tugas.
7. Jika run adalah bagian dari grup run dan grup run habis lebih cepat dari pengatur waktu tugas, proses dan tugas akan beralih ke status gagal.

Tentukan durasi batas waktu menggunakan satu atau beberapa unit berikut:`ms`,,`s`, `mh`, ataud.

Contoh berikut menunjukkan cara menentukan konfigurasi global dalam file konfigurasi Nextflow. Ini menetapkan batas waktu global 1 jam dan 30 menit.

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

Contoh berikut menunjukkan cara menentukan direktif waktu di bagian tugas definisi alur kerja. Contoh ini menetapkan batas waktu 3 hari, 5 jam, dan 4 menit. Nilai ini lebih diutamakan daripada nilai global dalam file konfigurasi, tetapi tidak diutamakan daripada arahan waktu khusus tugas untuk dalam file konfigurasi. `my_label`

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

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

Contoh berikut menunjukkan cara menentukan arahan waktu khusus tugas dalam file konfigurasi Nextflow, berdasarkan pemilih nama atau label. Contoh ini menetapkan nilai batas waktu tugas global

30 menit. Ini menetapkan nilai 2 jam untuk tugas `myTask` dan menetapkan nilai 3 jam untuk tugas dengan label`my_label`. Untuk tugas yang cocok dengan pemilih, nilai ini lebih diutamakan daripada nilai global dan nilai dalam definisi alur kerja.

```
process {
    time = '30m'

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

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

Ekspor konten tugas

Untuk alur kerja yang ditulis dalam Nextflow, tentukan direktif PublishDir untuk mengekspor konten tugas ke bucket Amazon S3 keluaran Anda. Seperti yang ditunjukkan pada contoh berikut, atur nilai publishDir ke. `/mnt/workflow/pubdir` Untuk mengekspor file ke Amazon S3, file harus ada di direktori ini.

```
 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
  """
}
```

Spesifikasi definisi alur kerja CWL

Alur kerja yang ditulis dalam Bahasa Alur Kerja Umum, atau CWL, menawarkan fungsionalitas serupa dengan alur kerja yang ditulis dalam WDL dan Alur Berikutnya. Anda dapat menggunakan Amazon S3 atau HealthOmics penyimpanan URIs sebagai parameter input.

Jika Anda menentukan input dalam SecondaryFile dalam sub alur kerja, tambahkan definisi yang sama dalam alur kerja utama.

HealthOmics alur kerja tidak mendukung proses operasi. Untuk mempelajari lebih lanjut tentang proses operasi dalam alur kerja CWL, lihat dokumentasi CWL.

Praktik terbaik adalah menentukan alur kerja CWL terpisah untuk setiap wadah yang Anda gunakan. Kami menyarankan Anda untuk tidak melakukan hardcode entri DockerPull dengan URI ECR Amazon tetap.

Topik

- Mengkonversi alur kerja CWL untuk digunakan HealthOmics
- Menyisih dari tugas coba lagi menggunakan omicsRetryOn5xx
- Lingkarkan langkah alur kerja
- Coba lagi tugas dengan peningkatan memori
- Contoh

Mengkonversi alur kerja CWL untuk digunakan HealthOmics

Untuk mengonversi definisi alur kerja CWL yang ada untuk digunakan HealthOmics, buat perubahan berikut:

- Ganti semua wadah Docker URIs dengan Amazon URIs ECR.
- Pastikan bahwa semua file alur kerja dideklarasikan dalam alur kerja utama sebagai input, dan semua variabel didefinisikan secara eksplisit.
- Pastikan bahwa semua JavaScript kode adalah keluhan mode ketat.

Menyisih dari tugas coba lagi menggunakan **`omicsRetryOn5xx`**

HealthOmics mendukung percobaan ulang tugas jika tugas gagal karena kesalahan layanan (kode status HTTP 5XX). Secara default, HealthOmics mencoba hingga dua percobaan ulang dari tugas yang gagal. Untuk informasi selengkapnya tentang coba lagi tugas HealthOmics, lihatTugas Mencoba Ulang.

Untuk memilih keluar dari tugas coba lagi untuk kesalahan layanan, konfigurasikan `omicsRetryOn5xx` arahan dalam definisi alur kerja. Anda dapat menentukan arahan ini di bawah

persyaratan atau petunjuk. Kami merekomendasikan menambahkan arahan sebagai petunjuk untuk portabilitas.

```
requirements:
  ResourceRequirement:
    omicsRetryOn5xx: false

hints:
  ResourceRequirement:
    omicsRetryOn5xx: false
```

Persyaratan mengesampingkan petunjuk. Jika implementasi tugas menyediakan persyaratan sumber daya dalam petunjuk yang juga disediakan oleh persyaratan dalam alur kerja terlampir, persyaratan terlampir akan diutamakan.

Jika persyaratan tugas yang sama muncul pada tingkat alur kerja yang berbeda, HealthOmics gunakan entri paling spesifik dari `requirements` (atau`hints`, jika tidak ada entri di`requirements`). Daftar berikut menunjukkan urutan prioritas yang HealthOmics digunakan untuk menerapkan pengaturan konfigurasi, dari prioritas terendah hingga tertinggi:

- Tingkat alur kerja
- Tingkat langkah
- Bagian tugas dari definisi alur kerja

Contoh berikut menunjukkan cara mengkonfigurasi `omicsRetryOn5xx` direktif di berbagai tingkat alur kerja. Dalam contoh ini, persyaratan tingkat alur kerja mengesampingkan petunjuk tingkat alur kerja. Konfigurasi persyaratan pada tingkat tugas dan langkah mengesampingkan konfigurasi petunjuk.

```
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
```

Lingkarkan langkah alur kerja

HealthOmics mendukung perulangan langkah alur kerja. Anda dapat menggunakan loop untuk menjalankan langkah alur kerja berulang kali hingga kondisi tertentu terpenuhi. Ini berguna untuk proses berulang di mana Anda perlu mengulangi tugas beberapa kali atau sampai hasil tertentu tercapai.

Catatan: Fungsionalitas loop membutuhkan CWL versi 1.2 atau yang lebih baru. Alur kerja yang menggunakan versi CWL lebih awal dari 1.2 tidak mendukung operasi loop.

Untuk menggunakan loop dalam alur kerja CWL Anda, tentukan persyaratan Loop. Contoh berikut menunjukkan konfigurasi persyaratan loop:

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

`loopWhen`Bidang mengontrol saat loop berakhir. Dalam contoh ini, loop berlanjut selama penghitung kurang dari nilai maksimum. `loop`Bidang mendefinisikan bagaimana parameter input diperbarui antara iterasi. `loopSource`Menentukan output dari umpan iterasi sebelumnya ke iterasi berikutnya. `outputMethod`Bidang diatur untuk `last` mengembalikan hanya output iterasi akhir ini.

Coba lagi tugas dengan peningkatan memori

HealthOmics mendukung percobaan ulang otomatis kegagalan out-of-memory tugas. Ketika tugas keluar dengan kode 137 (out-of-memory), HealthOmics membuat tugas baru dengan peningkatan alokasi memori berdasarkan pengganda yang ditentukan.

> **Note**
>
> HealthOmics mencoba kembali out-of-memory kegagalan hingga 3 kali atau sampai alokasi memori mencapai 1536 GiB, batas mana pun yang tercapai terlebih dahulu.

Contoh berikut menunjukkan cara mengkonfigurasi out-of-memory coba lagi:

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

Ketika tugas gagal karena out-of-memory, HealthOmics menghitung alokasi memori coba lagi menggunakan rumus:. `previous_run_memory × memoryRetryMultiplier` Pada contoh di atas, jika tugas dengan memori 4096 MB gagal, upaya coba lagi menggunakan memori 4096 × 2,5 = 10.240 MB.

`memoryRetryMultiplier`Parameter mengontrol berapa banyak memori tambahan yang dialokasikan untuk upaya coba lagi:

- Nilai default: Jika Anda tidak menentukan nilai, nilai defaultnya 2 (menggandakan memori)
- Rentang yang valid: Harus angka positif lebih besar dari1. Nilai yang tidak valid menghasilkan kesalahan validasi 4XX
- Nilai efektif minimum: Nilai antara 1 dan `1.5` secara otomatis ditingkatkan `1.5` untuk memastikan peningkatan memori yang berarti dan mencegah upaya coba lagi yang berlebihan

Contoh

Berikut ini adalah contoh alur kerja yang ditulis dalam CWL.

```
cwlVersion: v1.2
class: Workflow

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

out_filename: string
docker_image: string


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

steps:
copy_step:
in:
  in_file: in_file
  out_filename: out_filename
  docker_image: docker_image
out: [copied_file]
run: copy.cwl
```

File berikut mendefinisikan `copy.cwl` tugas.

```
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)"
```

Berikut ini adalah contoh alur kerja yang ditulis dalam CWL dengan persyaratan GPU.

```
cwlVersion: v1.2
class: CommandLineTool
baseCommand: ["/bin/bash", "docm_haplotypeCaller.sh"]
$namespaces:
cwltool: http://commonwl.org/cwltool#
requirements:
cwltool:CUDARequirement:
cudaDeviceCountMin: 1
cudaComputeCapability: "nvidia-tesla-t4"
cudaVersionMin: "1.0"
InlineJavascriptRequirement: {}
InitialWorkDirRequirement:
listing:
- entryname: 'docm_haplotypeCaller.sh'
  entry: |
          nvidia-smi --query-gpu=gpu_name,gpu_bus_id,vbios_version --format=csv

inputs: []
outputs: []
```

Contoh definisi alur kerja

Contoh berikut menunjukkan definisi alur kerja yang sama di WDL, Nextflow, dan 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
```

File template parameter untuk HealthOmics alur kerja

Template parameter menentukan parameter input untuk alur kerja. Anda dapat menentukan parameter input untuk membuat alur kerja Anda lebih fleksibel dan serbaguna. Misalnya, Anda dapat menentukan parameter untuk lokasi Amazon S3 dari file genom referensi. Template parameter dapat disediakan melalui layanan repositori berbasis Git atau drive lokal Anda. Pengguna kemudian dapat menjalankan alur kerja menggunakan berbagai set data.

Anda dapat membuat template parameter untuk alur kerja Anda, atau HealthOmics dapat menghasilkan template parameter untuk Anda.

Template parameter adalah file JSON. Dalam file, setiap parameter input adalah objek bernama yang harus cocok dengan nama input alur kerja. Saat Anda memulai proses, jika Anda tidak memberikan nilai untuk semua parameter yang diperlukan, proses gagal.

Objek parameter masukan mencakup atribut berikut:

- description— Atribut wajib ini adalah string yang ditampilkan konsol di halaman Mulai jalankan. Deskripsi ini juga dipertahankan sebagai run metadata.
- optional— Atribut opsional ini menunjukkan apakah parameter input adalah opsional. Jika Anda tidak menentukan optional bidang, parameter input diperlukan.

Contoh template parameter berikut menunjukkan bagaimana menentukan parameter masukan.

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

Menghasilkan templat parameter

HealthOmics menghasilkan template parameter dengan mengurai definisi alur kerja untuk mendeteksi parameter input. Jika Anda menyediakan file template parameter untuk alur kerja, parameter dalam file Anda akan mengganti parameter yang terdeteksi dalam definisi alur kerja.

Ada sedikit perbedaan antara logika parsing mesin CWL, WDL, dan Nextflow, seperti yang dijelaskan di bagian berikut.

Topik

Deteksi parameter untuk CWL

Di mesin alur kerja CWL, logika parsing membuat asumsi berikut:

- Tipe yang didukung nullable ditandai sebagai parameter input opsional.
- Tipe yang didukung non-null ditandai sebagai parameter input yang diperlukan.
- Setiap parameter dengan nilai default ditandai sebagai parameter input opsional.
- Deskripsi diekstraksi dari `label` bagian dari definisi `main` alur kerja. Jika tidak `label` ditentukan, deskripsi akan kosong (string kosong).

Tabel berikut menunjukkan contoh interpolasi CWL. Untuk setiap contoh, nama parameternya adalah`x`. Jika parameter diperlukan, Anda harus memberikan nilai untuk parameter. Jika parameternya opsional, Anda tidak perlu memberikan nilai.

Tabel ini menunjukkan contoh interpolasi CWL untuk tipe primitif.

Input	Contoh input/output	Wajib
`x:` `  type: int`	1 atau 2 atau...	Ya

Input	Contoh input/output	Wajib

`x:`
`  type: int`
`  default: 2`
	Nilai default adalah 2. Masukan yang valid adalah 1 atau 2 atau...	Tidak

`x:` `  type: int?`	Masukan yang valid adalah None atau 1 atau 2 atau...	Tidak

`x:`
`  type: int?`
`  default: 2`
	Nilai default adalah 2. Masukan yang valid adalah None atau 1 atau 2 atau...	Tidak

Tabel berikut menunjukkan contoh interpolasi CWL untuk tipe kompleks. Tipe kompleks adalah kumpulan tipe primitif.

Input	Contoh input/output	Wajib

`x:`
`  type: array`
`  items: int`
	[] atau [1,2,3]	Ya

`x:`
`  type: array?`
`  items: int`
	Tidak ada atau [] atau [1,2,3]	Tidak

`x:`
`  type: array`
`  items: int?`
	[] atau [Tidak ada, 3, tidak ada]	Ya

`x:`
`  type: array?`
`  items: int?`
	[Tidak] atau Tidak ada atau [1,2,3] atau [Tidak ada, 3] tetapi tidak []	Tidak

Deteksi parameter untuk WDL

Di mesin alur kerja WDL, logika parsing membuat asumsi berikut:

- Tipe yang didukung nullable ditandai sebagai parameter input opsional.
- Untuk tipe yang didukung yang tidak dapat dibatalkan:
  - Setiap variabel input dengan penetapan literal atau ekspresi ditandai sebagai parameter opsional. Misalnya:

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

  - Jika tidak ada nilai atau ekspresi yang ditetapkan ke parameter input, mereka akan ditandai sebagai parameter yang diperlukan.
- Deskripsi diekstraksi dari `parameter_meta` dalam definisi `main` alur kerja. Jika tidak `parameter_meta` ditentukan, deskripsi akan kosong (string kosong). Untuk informasi lebih lanjut, lihat spesifikasi WDL untuk metadata Parameter.

Tabel berikut menunjukkan contoh interpolasi WDL. Untuk setiap contoh, nama parameternya adalah`x`. Jika parameter diperlukan, Anda harus memberikan nilai untuk parameter. Jika parameternya opsional, Anda tidak perlu memberikan nilai.

Tabel ini menunjukkan contoh interpolasi WDL untuk tipe primitif.

Input	Contoh input/output	Wajib
Int x	1 atau 2 atau...	Ya
Int x = 2	2	Tidak
Int x = 1+2	3	Tidak
Int x = y+z	y+z	Tidak
Int? x	Tidak ada atau 1 atau 2 atau...	Ya
Int? x = 2	Tidak ada atau 2	Tidak
Int? x = 1+2	Tidak ada atau 3	Tidak

Input	Contoh input/output	Wajib
Int? x = y+z	Tidak ada atau y+z	Tidak

Tabel berikut menunjukkan contoh interpolasi WDL untuk tipe kompleks. Tipe kompleks adalah kumpulan tipe primitif.

Input	Contoh input/output	Wajib
Array [Int] x	[1,2,3] atau []	Ya
Array [Int] + x	[1], tetapi tidak []	Ya
Array [Int]? x	Tidak ada atau [] atau [1,2,3]	Tidak
Array [Int?] x	[] atau [Tidak ada, 3, tidak ada]	Ya
Array [Int?] =? x	[Tidak] atau Tidak ada atau [1,2,3] atau [Tidak ada, 3] tetapi tidak []	Tidak
Contoh struktur {String a, Int y} nanti di input: Contoh MySample	`String a = mySample.a Int y = mySample.y`	Ya
Contoh struktur {String a, Int y}	`if (defined(mySample)) { String a = mySample.a`	Tidak

Input	Contoh input/output	Wajib		
nanti di input: Sampel? MySample	`Int y = mySample.y }`			

Deteksi parameter untuk Nextflow

Untuk Nextflow, HealthOmics buat template parameter dengan mengurai file. `nextflow_schema.json` Jika definisi alur kerja tidak menyertakan file skema, HealthOmics parsing file definisi alur kerja utama.

Topik

- Mengurai file skema
- Mengurai file utama
- Parameter bersarang
- Contoh interpolasi Nextflow

Mengurai file skema

Agar parsing berfungsi dengan benar, pastikan file skema memenuhi persyaratan berikut:

- File skema diberi nama `nextflow_schema.json` dan terletak di direktori yang sama dengan file alur kerja utama.
- File skema adalah JSON valid seperti yang didefinisikan dalam salah satu skema berikut:
  - skema json. org/draft/2020-12/schema.
  - skema json. org/draft-07/schema.

HealthOmics `nextflow_schema.json`mem-parsing file untuk menghasilkan template parameter:

- Ekstrak semua properties yang didefinisikan dalam skema.
- Termasuk properti description jika tersedia untuk properti.
- Mengidentifikasi apakah setiap parameter opsional atau wajib, berdasarkan required bidang properti.

Contoh berikut menunjukkan file definisi dan file parameter yang dihasilkan.

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

Template parameter yang dihasilkan:

```
{
    "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
    }
}
```

Mengurai file utama

Jika definisi alur kerja tidak menyertakan `nextflow_schema.json` file, HealthOmics parsing file definisi alur kerja utama.

HealthOmics menganalisis `params` ekspresi yang ditemukan di file definisi alur kerja utama dan dalam file. `nextflow.config` Semua `params` dengan nilai default ditandai sebagai opsional.

Agar parsing berfungsi dengan benar, perhatikan persyaratan berikut:

- HealthOmics hanya mem-parsing file definisi alur kerja utama. Untuk memastikan semua parameter ditangkap, kami sarankan Anda menghubungkan semua params submodul dan alur kerja yang diimpor.
- File konfigurasi adalah opsional. Jika Anda mendefinisikannya, beri nama `nextflow.config` dan letakkan di direktori yang sama dengan file definisi alur kerja utama.

Contoh berikut menunjukkan file definisi dan template parameter yang dihasilkan.

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

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

Template parameter yang dihasilkan:

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

Untuk nilai default yang didefinisikan dalam nextflow.config, HealthOmics mengumpulkan `params` tugas dan parameter yang dideklarasikan dalam`params {}`, seperti yang ditunjukkan pada contoh berikut. Dalam pernyataan penugasan, `params` harus muncul di sisi kiri pernyataan.

```
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}"
```

Template parameter yang dihasilkan:

```
{
    // 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
    }
}
```

Parameter bersarang

Keduanya `nextflow_schema.json` dan `nextflow.config` memungkinkan parameter bersarang. Namun, template HealthOmics parameter hanya membutuhkan parameter tingkat atas. Jika alur kerja Anda menggunakan parameter bersarang, Anda harus menyediakan objek JSON sebagai input untuk parameter tersebut.

Parameter bersarang dalam file skema

HealthOmics melewatkan bersarang params saat mengurai file. `nextflow_schema.json` Misalnya, jika Anda mendefinisikan `nextflow_schema.json` file berikut:

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

HealthOmics mengabaikan `input_file` dan `input_num` ketika menghasilkan template parameter:

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

Saat Anda menjalankan alur kerja ini, HealthOmics mengharapkan `input.json` file yang mirip dengan berikut ini:

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

Parameter bersarang dalam file konfigurasi

HealthOmics tidak mengumpulkan bersarang params dalam `nextflow.config` file, dan melewatkannya selama penguraian. Misalnya, jika Anda mendefinisikan `nextflow.config` file berikut:

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

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

HealthOmics mengabaikan `params.nested.beta` dan `params.group.delta` ketika menghasilkan template parameter:

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

Contoh interpolasi Nextflow

Tabel berikut menunjukkan contoh interpolasi Nextflow untuk params di file utama.

Parameter	Wajib
params.input_file	Ya
params.input_file = "s3://bucket/data.json”	Tidak
params.nested.input_file	N/A

Parameter	Wajib
params.nested.input_file = "s3://bucket/data.json”	N/A

Tabel berikut menunjukkan contoh interpolasi Nextflow untuk params dalam file. `nextflow.config`

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

Gambar kontainer untuk alur kerja pribadi

HealthOmics mendukung gambar kontainer yang dihosting di repositori pribadi Amazon ECR. Anda dapat membuat gambar kontainer dan mengunggahnya ke repositori pribadi. Anda juga dapat menggunakan registri pribadi Amazon ECR Anda sebagai cache tarik untuk menyinkronkan konten pendaftar hulu.

Repositori Amazon ECR Anda harus berada di AWS Wilayah yang sama dengan akun yang memanggil layanan. Yang berbeda Akun AWS dapat memiliki gambar kontainer, selama repositori

gambar sumber memberikan izin yang sesuai. Untuk informasi selengkapnya, lihat Kebijakan untuk akses ECR Amazon lintas akun.

Sebaiknya tentukan image container Amazon ECR URIs sebagai parameter dalam alur kerja sehingga akses dapat diverifikasi sebelum proses dimulai. Ini juga mempermudah menjalankan alur kerja di Wilayah baru dengan mengubah parameter Region.

> **Note**
>
> HealthOmics tidak mendukung kontainer ARM dan tidak mendukung akses ke repositori publik.

Untuk informasi tentang mengonfigurasi izin IAM untuk HealthOmics mengakses Amazon ECR, lihat. HealthOmics Izin sumber daya

Topik

Sinkronisasi dengan pendaftar kontainer pihak ketiga

Anda dapat menggunakan aturan cache pull through Amazon ECR untuk menyinkronkan repositori di registri upstream yang didukung dengan repositori pribadi Amazon ECR Anda. Untuk informasi selengkapnya, lihat Menyinkronkan registri upstream di Panduan Pengguna Amazon ECR.

Pull through cache secara otomatis membuat repositori gambar di registri pribadi Anda saat Anda membuat cache, dan secara otomatis menyinkronkan dengan gambar yang di-cache ketika ada perubahan pada gambar upstream.

HealthOmics mendukung pull through cache untuk registri hulu berikut:

- Amazon ECR Public
- Registri gambar kontainer Kubernetes
- dermaga

- Hub Docker
- Registri Kontainer Microsoft Azure
- GitHub Registri Kontainer
- GitLab Registri Kontainer

HealthOmics tidak mendukung penarikan cache untuk repositori pribadi Amazon ECR hulu.

Manfaat menggunakan Amazon ECR pull through cache meliputi:

1. Anda menghindari keharusan memigrasikan gambar kontainer secara manual ke Amazon ECR atau untuk menyinkronkan pembaruan dari repositori pihak ketiga.
2. Alur kerja mengakses gambar kontainer yang disinkronkan di repositori pribadi Anda, yang lebih andal daripada mengunduh konten saat dijalankan dari registri publik.
3. Karena Amazon ECR menarik cache menggunakan struktur URI yang dapat diprediksi, HealthOmics layanan dapat secara otomatis memetakan URI pribadi Amazon ECR dengan URI registri hulu. Anda tidak perlu memperbarui dan mengganti nilai URI dalam definisi alur kerja.

Topik

- Mengkonfigurasi pull through cache
- Pemetaan registri
- Pemetaan gambar

Mengkonfigurasi pull through cache

Amazon ECR menyediakan registri untuk Anda Akun AWS di setiap Wilayah. Pastikan Anda membuat konfigurasi Amazon ECR di wilayah yang sama tempat Anda berencana menjalankan alur kerja.

Bagian berikut menjelaskan tugas konfigurasi untuk pull through cache.

Tugas konfigurasi

- Buat aturan cache tarik
- Izin registri untuk registri hulu
- Templat pembuatan repositori
- Membuat alur kerja

Buat aturan cache tarik

Buat aturan cache pull through Amazon ECR untuk setiap registri upstream yang memiliki gambar yang ingin Anda cache. Aturan menentukan pemetaan antara registri upstream dan repositori pribadi Amazon ECR.

Untuk registri upstream yang memerlukan autentikasi, Anda memberikan kredensialnya menggunakan AWS Secrets Manager.

> **Note**
>
> Jangan mengubah aturan pull through cache saat run aktif menggunakan repositori pribadi. Proses berjalan bisa gagal atau, yang lebih kritis, menghasilkan pipeline Anda menggunakan gambar yang tidak terduga.

Untuk informasi selengkapnya, lihat Membuat aturan cache tarik melalui di Panduan Pengguna Amazon Elastic Container Registry.

Buat aturan cache tarik menggunakan konsol

Untuk mengonfigurasi cache tarik melalui, ikuti langkah-langkah ini menggunakan konsol Amazon ECR:

1. Buka konsol Amazon ECR: https://console.aws.amazon.com /ecr
2. Dari menu kiri, di bawah Registri pribadi, perluas Fitur & Pengaturan. Lalu pilih Tarik melalui cache.
3. Dari halaman Tarik melalui cache, pilih Tambahkan aturan.
4. Di panel registri Upstream, pilih registri upstream untuk disinkronkan dengan registri pribadi Anda, lalu pilih Berikutnya.
5. Jika registri upstream memerlukan otentikasi, konsol akan membuka halaman baru tempat Anda menentukan rahasia SageMaker AI yang berisi kredensyal Anda. Pilih Berikutnya.
6. Di bawah Tentukan ruang nama, di panel namespace Cache, pilih apakah akan membuat repositori pribadi menggunakan awalan repositori tertentu atau tanpa awalan. Jika Anda memilih untuk menggunakan awalan, tentukan nama awalan di awalan repositori Cache.
7. Di panel namespace Upstream, pilih apakah akan menarik dari repositori upstream menggunakan awalan repositori tertentu atau tanpa awalan. Jika Anda memilih untuk menggunakan awalan, tentukan nama awalan di awalan repositori Upstream.

Panel contoh Namespace menunjukkan contoh permintaan tarik, URL hulu, dan URL repositori cache yang dibuat.

8. Pilih Berikutnya.
9. Tinjau konfigurasi dan pilih Buat untuk membuat aturan.

Untuk informasi selengkapnya, lihat Membuat aturan cache tarik (AWS Management Console).

Buat aturan cache tarik melalui menggunakan CLI

Gunakan create-pull-through-cache-rule perintah Amazon ECR untuk membuat aturan cache pull through. Untuk registrasi upstream yang memerlukan otentikasi, simpan kredensialnya dalam rahasia Secrets Manager.

Bagian berikut memberikan contoh untuk setiap registri upstream yang didukung.

Untuk Amazon ECR Publik

Contoh berikut membuat aturan cache pull through untuk registri Publik Amazon ECR. Ini menentukan awalan repositori`ecr-public`, yang menghasilkan setiap repositori yang dibuat menggunakan aturan pull through cache untuk memiliki skema penamaan. `ecr-public/`*`upstream-repository-name`*

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

Untuk Registri Kontainer Kubernetes

Contoh berikut membuat aturan pull through cache untuk registri publik Kubernetes. Ini menentukan awalan repositori`kubernetes`, yang menghasilkan setiap repositori yang dibuat menggunakan aturan pull through cache untuk memiliki skema penamaan. `kubernetes/`*`upstream-repository-name`*

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

Untuk Quay

Contoh berikut membuat aturan pull through cache untuk registri publik Quay. Ini menentukan awalan repositoriquay, yang menghasilkan setiap repositori yang dibuat menggunakan aturan pull through cache untuk memiliki skema penamaan. quay/*upstream-repository-name*

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

Untuk Docker Hub

Contoh berikut membuat aturan pull through cache untuk registri Docker Hub. Ini menentukan awalan repositoridocker-hub, yang menghasilkan setiap repositori yang dibuat menggunakan aturan pull through cache untuk memiliki skema penamaan. docker-hub/*upstream-repository-name* Anda harus menentukan Nama Sumber Daya Amazon (ARN) lengkap dari rahasia yang berisi kredenal Docker Hub Anda.

```
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
```

Untuk Registri GitHub Kontainer

Contoh berikut membuat aturan pull through cache untuk GitHub Container Registry. Ini menentukan awalan repositorigithub, yang menghasilkan setiap repositori yang dibuat menggunakan aturan pull through cache untuk memiliki skema penamaan. github/*upstream-repository-name* Anda harus menentukan Nama Sumber Daya Amazon (ARN) lengkap dari rahasia yang berisi kredensyal Registri GitHub Penampung Anda.

```
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
```

Untuk Microsoft Azure Container Registry

Contoh berikut membuat aturan pull through cache untuk Microsoft Azure Container Registry. Ini menentukan awalan repositori`azure`, yang menghasilkan setiap repositori yang dibuat menggunakan aturan pull through cache untuk memiliki skema penamaan. `azure/`*`upstream-repository-name`* Anda harus menentukan Nama Sumber Daya Amazon (ARN) lengkap dari rahasia yang berisi kredensyal Microsoft Azure Container Registry Anda.

```
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
```

Untuk Registri GitLab Kontainer

Contoh berikut membuat aturan pull through cache untuk GitLab Container Registry. Ini menentukan awalan repositori`gitlab`, yang menghasilkan setiap repositori yang dibuat menggunakan aturan pull through cache untuk memiliki skema penamaan. `gitlab/`*`upstream-repository-name`* Anda harus menentukan Nama Sumber Daya Amazon (ARN) lengkap dari rahasia yang berisi kredensyal Registri GitLab Penampung Anda.

```
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
```

Untuk informasi selengkapnya, lihat Membuat aturan cache tarik (CLI) di Panduan Pengguna Amazon ECR.

Anda dapat menggunakan perintah get-run-task CLI untuk mengambil informasi tentang gambar kontainer yang digunakan untuk tugas tertentu:

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

Outputnya mencakup informasi berikut tentang gambar kontainer:

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

Izin registri untuk registri hulu

Gunakan izin registri HealthOmics untuk memungkinkan penggunaan cache pull through dan untuk menarik gambar kontainer ke dalam registri pribadi Amazon ECR. Tambahkan kebijakan Amazon ECR Registry ke registri yang menyediakan kontainer yang digunakan dalam proses.

Kebijakan berikut memberikan izin kepada HealthOmics layanan untuk membuat repositori dengan awalan pull through cache yang ditentukan dan untuk memulai penarikan upstream ke dalam repositori ini.

1. Dari konsol Amazon ECR, buka menu kiri, di bawah Registri pribadi, perluas izin Registry. Lalu pilih Hasilkan pernyataan.
2. Di sisi kanan atas, pilih JSON. Masukkan kebijakan yang mirip dengan berikut ini:

   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/*"
            ]
        }
```

```
    ]
}
```

Templat pembuatan repositori

Untuk menggunakan pull through caching in HealthOmics, repositori Amazon ECR harus memiliki template pembuatan repositori. Template mendefinisikan pengaturan konfigurasi saat Anda atau Amazon ECR membuat repositori pribadi untuk registri upstream.

Setiap template berisi awalan namespace repositori, yang digunakan Amazon ECR untuk mencocokkan repositori baru dengan templat tertentu. Template menentukan konfigurasi untuk semua setelan repositori termasuk kebijakan akses berbasis sumber daya, kekekalan tag, enkripsi, dan kebijakan siklus hidup.

Untuk informasi selengkapnya, lihat Templat pembuatan repositori di Panduan Pengguna Amazon Elastic Container Registry.

Cara membuat template pembuatan repositori:

1. Dari konsol Amazon ECR, buka menu kiri, di bawah Registri pribadi, perluas Fitur dan pengaturan. Lalu pilih Template pembuatan repositori.
2. Pilih Buat templat.
3. Dalam rincian Template, pilih Tarik melalui cache.
4. Pilih apakah akan menerapkan template ini ke awalan tertentu atau ke semua repositori yang tidak cocok dengan template lain.

   Jika Anda memilih awalan tertentu, masukkan nilai awalan namespace di Awalan. Anda menentukan awalan ini ketika Anda membuat aturan PTC.
5. Pilih Berikutnya.
6. Di halaman konfigurasi pembuatan repositori, masukkan izin Repositori. Gunakan salah satu contoh pernyataan kebijakan, atau masukkan yang mirip dengan contoh berikut:

   JSON

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

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

7. Secara opsional, Anda dapat menambahkan pengaturan repositori seperti kebijakan siklus hidup dan tag. Amazon ECR menerapkan aturan ini untuk semua gambar kontainer yang dibuat untuk menarik cache yang menggunakan awalan yang ditentukan.
8. Pilih Berikutnya.
9. Tinjau konfigurasi dan pilih Berikutnya.

Membuat alur kerja

Saat Anda membuat alur kerja atau alur kerja baru, tinjau pemetaan registri dan perbarui jika diperlukan. Lihat perinciannya di Buat alur kerja pribadi.

Pemetaan registri

Anda menentukan pemetaan registri untuk memetakan antara awalan di registri ECR Amazon pribadi Anda dan nama registri hulu.

Untuk informasi selengkapnya tentang pemetaan registri Amazon ECR, lihat Membuat aturan cache tarik melalui di Amazon ECR.

Contoh berikut menunjukkan pemetaan registri ke Docker Hub, Quay, dan Amazon ECR Public.

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

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

Pemetaan gambar

Anda menentukan pemetaan gambar untuk memetakan antara nama gambar seperti yang didefinisikan dalam alur kerja Amazon ECR pribadi Anda dan nama gambar di registri hulu.

Anda dapat menggunakan pemetaan gambar dengan registri yang mendukung penarikan cache. Anda juga dapat menggunakan pemetaan gambar dengan registri hulu yang HealthOmics tidak mendukung penarikan cache. Anda perlu menyinkronkan registri hulu secara manual dengan repositori pribadi Anda.

Untuk informasi selengkapnya tentang pemetaan gambar Amazon ECR, lihat Membuat aturan cache tarik melalui di Amazon ECR.

Contoh berikut menunjukkan pemetaan dari gambar ECR Amazon pribadi ke gambar genomik publik dan gambar Ubuntu terbaru.

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

Pertimbangan umum untuk gambar kontainer Amazon ECR

- Arsitektur

  HealthOmics mendukung wadah x86_64. Jika mesin lokal Anda berbasis ARM, seperti Apple Mac, gunakan perintah seperti berikut ini untuk membuat image container x86_64:

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

- Entrypoint dan shell

  HealthOmics mesin alur kerja menyuntikkan skrip bash sebagai penggantian perintah ke gambar kontainer yang digunakan oleh tugas alur kerja. Dengan demikian, gambar kontainer harus dibangun tanpa ENTRYPOINT yang ditentukan sehingga shell bash adalah default.

- Jalur yang dipasang

  Sistem file bersama dipasang ke tugas kontainer di /tmp. Setiap data atau perkakas yang dibangun ke dalam gambar kontainer di lokasi ini akan diganti.

  Definisi alur kerja tersedia untuk tugas melalui mount hanya-baca di /mnt/workflow.

- Ukuran gambar

  Lihat HealthOmics alur kerja kuota ukuran tetap untuk ukuran gambar kontainer maksimum.

Variabel lingkungan untuk HealthOmics alur kerja

HealthOmics menyediakan variabel lingkungan yang memiliki informasi tentang alur kerja yang berjalan di wadah. Anda dapat menggunakan nilai variabel-variabel ini dalam logika tugas alur kerja Anda.

Semua variabel HealthOmics alur kerja dimulai dengan `AWS_WORKFLOW_` awalan. Awalan ini adalah awalan variabel lingkungan yang dilindungi. Jangan gunakan awalan ini untuk variabel Anda sendiri dalam wadah alur kerja.

HealthOmics menyediakan variabel lingkungan alur kerja berikut:

AWS_REGION

Variabel ini adalah wilayah tempat penampung berjalan.

AWS_WORKFLOW_JALANKAN

Variabel ini adalah nama run saat ini.

AWS_WORKFLOW_RUN_ID

Variabel ini adalah pengidentifikasi run dari run saat ini.

AWS_WORKFLOW_RUN_UUID

Variabel ini adalah UUID run dari run saat ini.

AWS_WORKFLOW_TUGAS

Variabel ini adalah nama tugas saat ini.

AWS_WORKFLOW_TASK_ID

Variabel ini adalah pengidentifikasi tugas dari tugas saat ini.

AWS_WORKFLOW_TUGAS_UUID

Variabel ini adalah tugas UUID dari tugas saat ini.

Contoh berikut menunjukkan nilai-nilai khas untuk setiap variabel lingkungan:

```
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
```

Menggunakan Java di gambar kontainer Amazon ECR

Jika tugas alur kerja menggunakan aplikasi Java seperti GATK, pertimbangkan persyaratan memori berikut untuk wadah:

- Aplikasi Java menggunakan memori tumpukan dan memori heap. Secara default, memori heap maksimum adalah persentase dari total memori yang tersedia dalam wadah. Default ini tergantung pada distribusi JVM tertentu dan versi JVM, jadi konsultasikan dokumentasi yang relevan untuk JVM Anda atau secara eksplisit atur memori heap maksimum menggunakan opsi baris perintah Java (seperti `-Xmx`).

- Jangan mengatur memori heap maksimum menjadi 100% dari alokasi memori kontainer, karena tumpukan JVM juga membutuhkan memori. Memori juga diperlukan untuk pengumpul sampah JVM dan proses sistem operasi lainnya yang berjalan di wadah.
- Beberapa aplikasi Java, seperti GATK, dapat menggunakan pemanggilan metode asli atau pengoptimalan lain seperti file pemetaan memori. Teknik-teknik ini memerlukan alokasi memori yang dilakukan “off heap”, yang tidak dikendalikan oleh parameter heap maksimum JVM.

  Jika Anda tahu (atau mencurigai) bahwa aplikasi Java Anda mengalokasikan memori off-heap, pastikan alokasi memori tugas Anda mencakup persyaratan memori off-heap.

  Jika alokasi off-heap ini menyebabkan wadah kehabisan memori, Anda biasanya tidak akan melihat OutOfMemory kesalahan Java, karena JVM tidak mengontrol memori ini.

Tambahkan input tugas ke gambar wadah Amazon ECR

Tambahkan semua executable, library, dan skrip yang diperlukan untuk menjalankan tugas alur kerja ke dalam image Amazon ECR yang digunakan untuk menjalankan tugas.

Ini adalah praktik terbaik untuk menghindari penggunaan skrip, binari, dan pustaka yang berada di luar gambar wadah tugas. Hal ini sangat penting ketika menggunakan `nf-core` alur kerja yang menggunakan `bin` direktori sebagai bagian dari paket alur kerja. Sementara direktori ini akan tersedia untuk tugas alur kerja, itu dipasang sebagai direktori read-only. Sumber daya yang diperlukan dalam direktori ini harus disalin ke dalam gambar tugas dan tersedia saat runtime atau saat membuat gambar kontainer yang digunakan untuk tugas tersebut.

Lihat HealthOmics alur kerja kuota ukuran tetap untuk ukuran maksimum gambar kontainer yang HealthOmics mendukung.

HealthOmics Alur kerja file README

Anda dapat mengunggah file README.md yang berisi instruksi, diagram, dan informasi penting untuk alur kerja Anda. Setiap versi alur kerja mendukung satu file README, yang dapat Anda perbarui kapan saja.

Persyaratan README meliputi:

- File README harus dalam format markdown (.md)
- Ukuran file maksimal: 500 KiB

Topik

Gunakan README yang ada

READMEs diekspor dari repositori Git berisi tautan relatif yang biasanya tidak berfungsi di luar repositori. HealthOmics Integrasi Git secara otomatis mengonversinya menjadi tautan absolut untuk rendering yang tepat di konsol, menghilangkan kebutuhan akan pembaruan URL manual.

Untuk READMEs diimpor dari Amazon S3 atau drive lokal, gambar dan tautan harus menggunakan publik URLs atau jalur relatifnya diperbarui untuk rendering yang tepat.

> **Note**
>
> Gambar harus di-host secara publik untuk ditampilkan di HealthOmics konsol. Gambar yang disimpan GitHub Enterprise Server atau GitLab Self-Managed repositori tidak dapat dirender.

Kondisi rendering

HealthOmics Konsol menginterpolasi gambar dan tautan yang dapat diakses publik menggunakan jalur absolut. Untuk merender URLs dari repositori pribadi, pengguna harus memiliki akses ke repositori. Untuk GitHub Enterprise Server atau GitLab Self-Managed repositori, yang menggunakan domain khusus, HealthOmics tidak dapat menyelesaikan tautan relatif atau merender gambar yang disimpan di repositori pribadi ini.

Tabel berikut menunjukkan elemen penurunan harga yang didukung oleh tampilan README AWS konsol.

Elemen	AWS konsol
Pemberitahuan	Ya, tetapi tanpa ikon
Lencana	Ya
Pemformatan teks dasar	Ya

Elemen	AWS konsol
Blok kode	Ya, tetapi tidak memiliki sorotan sintaks dan fungsi tombol salin
Bagian yang dapat dilipat	Ya
Judul	Ya
Format gambar	Ya
Gambar (dapat diklik)	Ya
Jeda baris	Ya
Diagram putri duyung	Hanya dapat membuka grafik, memindahkan posisi grafik, dan menyalin kode
Kutipan	Ya
Subskrip dan superskrip	Ya
Tabel	Ya, tetapi tidak mendukung perataan teks
Penjajaran teks	Ya

Menggunakan gambar dan tautan URLs

Bergantung pada penyedia sumber Anda, struktur basis Anda URLs untuk halaman dan gambar dalam format berikut.

- `{username}`: Nama pengguna tempat repositori di-host.
- `{repo}`: Nama repositori.
- `{ref}`: Referensi sumber (cabang, tag, dan ID komit).
- `{path}`: Jalur file ke halaman atau gambar di repositori.

Penyedia sumber	URL Halaman	URL gambar

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

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

GitHub,GitLab, dan Bitbucket mendukung halaman dan gambar URLs yang terhubung ke repositori publik. Tabel berikut menunjukkan dukungan masing-masing penyedia sumber untuk merender gambar dan tautan URLs untuk repositori pribadi.

Dukungan repositori pribadi		
Penyedia sumber	URL Halaman	URL gambar
GitHub	Hanya dengan akses ke repositori	Tidak
GitLab	Hanya dengan akses ke repositori	Tidak
Bitbucket	Hanya dengan akses ke repositori	Tidak

Meminta lisensi Sentieon untuk alur kerja pribadi

Jika alur kerja pribadi Anda menggunakan perangkat lunak Sentieon, Anda memerlukan lisensi Senieon. Ikuti langkah-langkah berikut untuk meminta dan menyiapkan lisensi untuk perangkat lunak Sentieon:

- Minta lisensi Sentieon
  - Kirim email ke grup dukungan Sentieon (support@sentieon.com) untuk meminta lisensi perangkat lunak.
    - Berikan ID Pengguna AWS Canonical Anda di email.
    - Temukan ID Pengguna AWS Canonical Anda dengan mengikuti petunjuk ini.
- Perbarui peran HealthOmics layanan Anda untuk memberinya akses ke proxy server lisensi Sentieon dan bucket Sentieon Omics di Wilayah Anda. Contoh berikut memberikan akses masuk`us-east-1`. Jika diperlukan, ganti teks ini dengan Wilayah Anda.

  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/*"
            ]
        }
    ]
 }
```

- Hasilkan kasus AWS dukungan untuk mendapatkan akses ke proxy server lisensi Sentieon.
  - Untuk membuat kasus dukungan, navigasikan ke support.console.aws.amazon.com.
  - Berikan Anda Akun AWS dan Wilayah dalam kasus dukungan. Akun Anda ditambahkan ke daftar yang diizinkan untuk proxy server lisensi.
- Bangun alur kerja pribadi Anda menggunakan wadah Sentieon dan skrip lisensi Sentieon.

  - Untuk petunjuk tambahan tentang penggunaan alat Sentieon di dalam alur kerja pribadi, lihat Sentieon-Amazon-Omics di. GitHub
- Versi perangkat lunak Sentieon 202112.07 dan yang lebih tinggi mendukung proxy server lisensi. HealthOmics Untuk menggunakan versi perangkat lunak Sentieon lebih awal dari 202112.07, hubungi dukungan Sentieon.

Linter alur kerja di HealthOmics

Setelah membuat alur kerja, sebaiknya jalankan linter pada alur kerja sebelum memulai proses pertama. Linter mendeteksi kesalahan yang dapat menyebabkan run gagal.

Untuk WDL, HealthOmics secara otomatis menjalankan linter saat Anda membuat alur kerja. Output linter tersedia di `statusMessage` bidang get-workflow respons. Gunakan perintah CLI berikut untuk mengambil output status (gunakan ID alur kerja dari alur kerja WDL yang Anda buat):

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

HealthOmics menyediakan linter yang dapat Anda jalankan pada pembelahan alur kerja sebelum Anda membuat alur kerja. Jalankan linter ini pada pipeline yang ada tempat Anda bermigrasi. HealthOmics

- WDL— Gambar ECR Amazon publik untuk menjalankan linter WDL.

- Nextflow— Gambar ECR Amazon publik untuk menjalankan aturan Linter untuk Nextflow. Anda dapat mengakses kode sumber untuk linter ini dari GitHub.
- CWL— tidak tersedia

HealthOmics operasi alur kerja

Untuk membuat alur kerja pribadi, Anda perlu:

- Workflow definition file:File definisi alur kerja yang ditulis dalamWDL,Nextflow, atauCWL. Definisi alur kerja menentukan input dan output untuk menjalankan yang menggunakan alur kerja. Ini juga mencakup spesifikasi untuk menjalankan dan menjalankan tugas untuk alur kerja Anda,

termasuk persyaratan komputasi dan memori. File definisi alur kerja harus dalam `.zip` format. Untuk informasi selengkapnya, lihat File definisi alur kerja di HealthOmics.

  - Anda dapat menggunakan Amazon Q CLI untuk membangun dan memvalidasi file definisi alur kerja Anda di WDL, Nextflow, dan CWL. Untuk informasi selengkapnya, lihat Contoh prompt untuk Amazon Q CLI dan tutorial AI HealthOmics generatif Agentic. GitHub

- (Optional) Parameter template file:File template parameter yang ditulis dalam fileJSON. Buat file untuk menentukan parameter run, atau HealthOmics buat template parameter untuk Anda. Untuk informasi selengkapnya, lihat File templat parameter untuk HealthOmics alur kerja.
- Amazon ECR container images:Buat repositori Amazon ECR pribadi untuk setiap kontainer yang digunakan dalam alur kerja. Buat gambar kontainer untuk alur kerja dan simpan di repositori pribadi, atau sinkronkan konten registri upstream yang didukung dengan repositori pribadi ECR Anda.
- (Optional) Sentieon licenses:Minta Sentieon lisensi untuk menggunakan Sentieon perangkat lunak dalam alur kerja pribadi.

Untuk file definisi alur kerja yang lebih besar dari 4 MiB (zip), pilih salah satu opsi ini selama pembuatan alur kerja:

- Unggah ke folder Amazon Simple Storage Service dan tentukan lokasinya.
- Unggah ke repositori eksternal (ukuran maksimal 1 GiB) dan tentukan detail repositori.

Setelah membuat alur kerja, Anda dapat memperbarui informasi alur kerja berikut dengan operasi: `UpdateWorkflow`

- Nama
- Deskripsi
- Jenis penyimpanan default
- Kapasitas penyimpanan default (dengan ID alur kerja)
- Berkas README.md

Untuk mengubah informasi lain dalam alur kerja, buat alur kerja atau versi alur kerja baru.

Gunakan pembuatan versi alur kerja untuk mengatur dan menyusun alur kerja Anda. Versi juga membantu Anda mengelola pengenalan pembaruan alur kerja berulang. Untuk informasi selengkapnya tentang versi, lihatBuat versi alur kerja.

Topik

Buat alur kerja pribadi

Buat alur kerja menggunakan HealthOmics konsol, perintah AWS CLI, atau salah satu. AWS SDKs

> **Note**
>
> Jangan sertakan informasi identitas pribadi (PII) apa pun dalam nama alur kerja. Nama-nama ini terlihat di CloudWatch log.

Saat Anda membuat alur kerja, HealthOmics tetapkan pengenal unik universal (UUID) ke alur kerja. UUID alur kerja adalah Pengenal Unik Global (panduan) yang unik di seluruh alur kerja dan versi alur kerja. Untuk tujuan asal data, kami menyarankan Anda menggunakan UUID alur kerja untuk mengidentifikasi alur kerja secara unik.

Jika tugas alur kerja Anda menggunakan alat eksternal (executable, pustaka, atau skrip), Anda membuat alat ini menjadi gambar kontainer. Anda memiliki opsi berikut untuk menghosting gambar kontainer:

- Host gambar kontainer di registri pribadi ECR. Prasyarat untuk opsi ini:
  - Buat repositori pribadi ECR, atau pilih repositori yang ada.
  - Konfigurasikan kebijakan sumber daya ECR seperti yang dijelaskan dalamIzin Amazon ECR.
  - Unggah gambar kontainer Anda ke repositori pribadi.
- Sinkronkan gambar kontainer dengan konten registri pihak ketiga yang didukung. Prasyarat untuk opsi ini:
  - Dalam registri pribadi ECR, konfigurasikan aturan pull through cache untuk setiap registri upstream. Untuk informasi selengkapnya, lihat Pemetaan gambar.
  - Konfigurasikan kebijakan sumber daya ECR seperti yang dijelaskan dalamIzin Amazon ECR.

- Buat template pembuatan repositori. Template menentukan pengaturan saat Amazon ECR membuat repositori pribadi untuk registri upstream.
- Buat pemetaan awalan untuk memetakan ulang referensi gambar kontainer dalam definisi alur kerja ke ruang nama cache ECR.

Saat membuat alur kerja, Anda memberikan definisi alur kerja yang berisi informasi tentang alur kerja, proses, dan tugas. HealthOmics dapat mengambil definisi alur kerja sebagai arsip.zip yang disimpan secara lokal atau di bucket Amazon S3, atau dari repositori berbasis Git yang didukung.

Topik

Membuat alur kerja menggunakan konsol

Langkah-langkah untuk membuat alur kerja

1. Buka konsol HealthOmics .
2. Jika diperlukan, buka panel navigasi kiri (≡). Pilih Alur kerja pribadi.
3. Pada halaman alur kerja pribadi, pilih Buat alur kerja.
4. Pada halaman Tentukan alur kerja, berikan informasi berikut:
    1. Nama alur kerja: Nama khas untuk alur kerja ini. Sebaiknya setel nama alur kerja untuk mengatur proses Anda di AWS HealthOmics konsol dan CloudWatch log.
    2. Deskripsi (opsional): Deskripsi alur kerja ini.
5. Di panel definisi Alur Kerja, berikan informasi berikut:
    1. Bahasa alur kerja (opsional): Pilih bahasa spesifikasi alur kerja. Jika tidak, HealthOmics tentukan bahasa dari definisi alur kerja.
    2. Untuk sumber definisi Alur Kerja, pilih untuk mengimpor folder definisi dari repositori berbasis Git, lokasi Amazon S3, atau dari drive lokal.
        a. Untuk Impor dari layanan repositori:

> **Note**
>
> HealthOmics mendukung repositori publik dan pribadi untukGitHub,,, GitLabBitbucket,GitHub self-managed. GitLab self-managed

i. Pilih Koneksi untuk menghubungkan AWS sumber daya Anda ke repositori eksternal. Untuk membuat koneksi, lihatConnect dengan repositori kode eksternal.

> **Note**
>
> Pelanggan di TLV wilayah tersebut perlu membuat koneksi di wilayah IAD (us-east-1) untuk membuat alur kerja.

ii. Dalam ID repositori lengkap, masukkan ID repositori Anda sebagai nama pengguna/repo-nama. Verifikasi bahwa Anda memiliki akses ke file di repositori ini.

iii. Dalam referensi Sumber (opsional), masukkan referensi sumber repositori (cabang, tag, atau ID komit). HealthOmics menggunakan cabang default jika tidak ada referensi sumber yang ditentukan.

iv. Di Kecualikan pola file, masukkan pola file untuk mengecualikan folder, file, atau ekstensi tertentu. Ini membantu mengelola ukuran data saat mengimpor file repositori. Ada maksimal 50 pola, dan pola harus mengikuti sintaks pola glob. Contoh:

A. `tests/`

B. `*.jpeg`

C. `large_data.zip`

b. Untuk Pilih folder definisi dari S3:

i. Masukkan lokasi Amazon S3 yang berisi folder definisi alur kerja zip. Bucket Amazon S3 harus berada di wilayah yang sama dengan alur kerja.

ii. Jika akun Anda tidak memiliki bucket Amazon S3, masukkan ID akun pemilik bucket di ID AWS akun pemilik bucket S3. Informasi ini diperlukan agar HealthOmics dapat memverifikasi kepemilikan bucket.

c. Untuk Pilih folder definisi dari sumber lokal:

i. Masukkan lokasi drive lokal dari folder definisi alur kerja zip.

3. Jalur file definisi alur kerja utama (opsional): Masukkan jalur file dari folder definisi alur kerja zip atau repositori ke file. `main` Parameter ini tidak diperlukan jika hanya ada satu file di folder definisi alur kerja, atau jika file utama diberi nama “main”.

6. Di panel file README (opsional), pilih Sumber file README dan berikan informasi berikut:

   - Untuk Impor dari layanan repositori, di jalur file README, masukkan path ke file README dalam repositori.
   - Untuk Pilih file dari S3, dalam file README di S3, masukkan URI Amazon S3 untuk file README.
   - Untuk Pilih file dari sumber lokal: di README - opsional, pilih Pilih file untuk memilih file penurunan harga (.md) yang akan diunggah.

7. Di panel konfigurasi penyimpanan run default, berikan tipe penyimpanan run default dan kapasitas untuk menjalankan yang menggunakan alur kerja ini:

   1. Jalankan jenis penyimpanan: Pilih apakah akan menggunakan penyimpanan statis atau dinamis sebagai default untuk penyimpanan berjalan sementara. Defaultnya adalah penyimpanan statis.
   2. Jalankan kapasitas penyimpanan (opsional): Untuk jenis penyimpanan run statis, Anda dapat memasukkan jumlah default penyimpanan run yang diperlukan untuk alur kerja ini. Nilai default untuk parameter ini adalah 1200 GiB. Anda dapat mengganti nilai default ini saat memulai proses.

8. Tag (opsional): Anda dapat mengaitkan hingga 50 tag dengan alur kerja ini.

9. Pilih Berikutnya.

10. Pada halaman Tambahkan parameter alur kerja (opsional), pilih sumber Parameter:

    1. Untuk Parse dari file definisi alur kerja, secara otomatis HealthOmics akan mengurai parameter alur kerja dari file definisi alur kerja.
    2. Untuk menyediakan template parameter dari repositori Git, gunakan path ke file template parameter dari repositori Anda.
    3. Untuk Pilih file JSON dari sumber lokal, unggah JSON file dari sumber lokal yang menentukan parameter.
    4. Untuk Masukkan parameter alur kerja secara manual, masukkan nama dan deskripsi parameter secara manual.

11. Di panel pratinjau Parameter, Anda dapat meninjau atau mengubah parameter untuk versi alur kerja ini. Jika Anda mengembalikan JSON file, Anda kehilangan perubahan lokal yang Anda buat.
12. Pilih Berikutnya.
13. Pada halaman pemetaan ulang URI Container, di panel aturan Pemetaan, Anda dapat menentukan aturan pemetaan URI untuk alur kerja Anda.

    Untuk Sumber file pemetaan, pilih salah satu opsi berikut:

    - Tidak ada - Tidak ada aturan pemetaan yang diperlukan.
    - Pilih file JSON dari S3 - Tentukan lokasi S3 untuk file pemetaan.
    - Pilih file JSON dari sumber lokal - Tentukan lokasi file pemetaan di perangkat lokal Anda.
    - Masukkan pemetaan secara manual - masukkan pemetaan registri dan pemetaan gambar di panel Pemetaan.
14. Konsol menampilkan panel Pemetaan. Jika Anda memilih file sumber pemetaan, konsol menampilkan nilai dari file.

    a. Dalam pemetaan Registry, Anda dapat mengedit pemetaan atau menambahkan pemetaan (maksimum 20 pemetaan registri).

       Setiap pemetaan registri berisi bidang-bidang berikut:

       - URL registri hulu — URI dari registri hulu.
       - Awalan repositori ECR - Awalan repositori untuk digunakan dalam repositori pribadi Amazon ECR.
       - (Opsional) Awalan repositori hulu - Awalan repositori di registri hulu.
       - (Opsional) ID akun ECR — ID akun akun yang memiliki gambar kontainer hulu.

    b. Dalam pemetaan Gambar, Anda dapat mengedit pemetaan gambar atau menambahkan pemetaan (maksimum 100 pemetaan gambar).

       Setiap pemetaan gambar berisi bidang-bidang berikut:

       - Gambar sumber - Menentukan URI gambar sumber di registri hulu.
       - Gambar tujuan - Menentukan URI gambar yang sesuai di registri ECR Amazon pribadi.
15. Pilih Berikutnya.
16. Tinjau konfigurasi alur kerja, lalu pilih Buat alur kerja.

Membuat alur kerja menggunakan CLI

Jika file alur kerja Anda dan file template parameter ada di mesin lokal Anda, Anda dapat membuat alur kerja menggunakan perintah CLI berikut.

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

`create-workflow`Operasi mengembalikan respon berikut:

```
{
  "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"
}
```

Parameter opsional untuk digunakan saat membuat alur kerja

Anda dapat menentukan salah satu parameter opsional saat membuat alur kerja. Untuk detail sintaks, lihat CreateWorkflowdi AWS HealthOmics API Referensi.

Topik

- Tentukan definisi alur kerja lokasi Amazon S3
- Menggunakan definisi alur kerja dari repositori berbasis Git
- Tentukan file Readme
- Tentukan file main definisi
- Tentukan jenis penyimpanan run
- Tentukan konfigurasi GPU
- Konfigurasikan parameter pemetaan cache pull through

Tentukan definisi alur kerja lokasi Amazon S3

Jika file definisi alur kerja Anda terletak di folder Amazon S3, tentukan lokasi menggunakan parameter, seperti `definition-uri` yang ditunjukkan pada contoh berikut. Jika akun Anda tidak memiliki bucket Amazon S3, berikan ID pemiliknya. Akun AWS

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

Menggunakan definisi alur kerja dari repositori berbasis Git

Untuk menggunakan definisi alur kerja dari repositori berbasis Git yang didukung, gunakan parameter dalam `definition-repository` permintaan Anda. Jangan berikan `definition` parameter lain, karena permintaan gagal jika menyertakan lebih dari satu sumber input.

`definition-respository`Parameter berisi bidang-bidang berikut:

- connectionArn— ARN dari Koneksi Kode yang menghubungkan sumber daya AWS Anda ke repositori eksternal.
- fullRepositoryId— Masukkan ID repositori sebagai. `owner-name/repo-name` Verifikasi bahwa Anda memiliki akses ke file di repositori ini.
- sourceReference(Opsional) — Masukkan tipe referensi repositori (BRANCH, TAG, atau COMMIT) dan nilai.

  HealthOmics menggunakan komit terbaru di cabang default jika Anda tidak menentukan referensi sumber.
- excludeFilePatterns(Opsional) - Masukkan pola file untuk mengecualikan folder, file, atau ekstensi tertentu. Ini membantu mengelola ukuran data saat mengimpor file repositori. Berikan maksimal 50 pola. pola harus mengikuti sintaks pola glob. Contoh:
  - `tests/`
  - `*.jpeg`
  - `large_data.zip`

Saat Anda menentukan definisi alur kerja dari repositori berbasis Git, gunakan `parameter-template-path` untuk menentukan file template parameter. Jika Anda tidak memberikan parameter ini, HealthOmics buat alur kerja tanpa templat parameter.

Contoh berikut menunjukkan parameter yang terkait dengan konten dari repositori pribadi berbasis Git:

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

Untuk contoh lainnya, lihat posting blog Cara Membuat HealthOmics Alur Kerja AWS dari Konten di Git.

Tentukan file Readme

Anda dapat menentukan lokasi file README menggunakan salah satu parameter berikut:

- readme-markdown— Input string atau file di mesin lokal Anda.
- readme-uri— URI file yang disimpan di S3.
- readme-path — Jalur ke file README di repositori.

Gunakan readme-path hanya dalam hubungannya dengan definition-respositori. Jika Anda tidak menentukan parameter README apa pun, HealthOmics impor file README.md tingkat root di repositori (jika ada).

Contoh berikut menunjukkan bagaimana menentukan lokasi file README menggunakan readme-path dan readme-uri.

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

# Using README from S3
aws omics create-workflow \
    --name "s3-readme-workflow" \
    --definition-repository '...' \
    --readme-uri "s3://my-bucket/workflow-docs/readme.md"
```

Untuk informasi selengkapnya, lihat HealthOmics Alur kerja file README.

Tentukan file main definisi

Jika Anda menyertakan beberapa file definisi alur kerja, gunakan `main` parameter untuk menentukan file definisi utama untuk alur kerja Anda.

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

Tentukan jenis penyimpanan run

Anda dapat menentukan jenis penyimpanan run default (DYNAMIC atau STATIC) dan menjalankan kapasitas penyimpanan (diperlukan untuk penyimpanan statis). Untuk informasi selengkapnya tentang menjalankan jenis penyimpanan, lihatJalankan jenis penyimpanan dalam HealthOmics alur kerja.

```
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  \
```

Tentukan konfigurasi GPU

Gunakan parameter akselerator untuk membuat alur kerja yang berjalan pada instance komputasi yang dipercepat. Contoh berikut menunjukkan bagaimana menggunakan `accelerators` parameter. Anda menentukan konfigurasi GPU dalam definisi alur kerja. Lihat Instans komputasi yang dipercepat.

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

Konfigurasikan parameter pemetaan cache pull through

Jika Anda menggunakan fitur pemetaan cache Amazon ECR pull through, Anda dapat mengganti pemetaan default. Untuk informasi selengkapnya tentang parameter penyiapan kontainer, lihatGambar kontainer untuk alur kerja pribadi.

Dalam contoh berikut, file `mappings.json` berisi konten ini:

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

Tentukan parameter pemetaan dalam perintah create-workflow:

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

Anda juga dapat menentukan lokasi S3 dari file parameter pemetaan:

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

Membuat alur kerja menggunakan SDK

Anda dapat membuat alur kerja menggunakan salah satu. SDKs Contoh berikut menunjukkan cara membuat alur kerja menggunakan Python SDK

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

Memperbarui alur kerja pribadi

Anda dapat memperbarui alur kerja menggunakan HealthOmics konsol, perintah AWS CLI, atau salah satu. AWS SDKs

> **Note**
>
> Jangan sertakan informasi identitas pribadi (PII) apa pun dalam nama alur kerja. Nama-nama ini terlihat di CloudWatch log.

Topik

- Memperbarui alur kerja menggunakan konsol
- Memperbarui alur kerja menggunakan CLI
- Memperbarui alur kerja menggunakan SDK

Memperbarui alur kerja menggunakan konsol

Langkah-langkah untuk memperbarui alur kerja

1. Buka konsol HealthOmics .
2. Jika diperlukan, buka panel navigasi kiri (≡). Pilih Alur kerja pribadi.
3. Pada halaman Alur kerja pribadi, pilih alur kerja yang akan diperbarui.
4. Pada halaman Workflow:
   - Jika alur kerja memiliki versi, pastikan Anda memilih versi Default.
   - Pilih Edit yang dipilih dari daftar Tindakan.
5. Pada halaman Edit alur kerja, Anda dapat mengubah salah satu nilai berikut:
   - Nama alur kerja.
   - Deskripsi alur kerja.
   - Jenis penyimpanan Jalankan default untuk alur kerja.
   - Kapasitas penyimpanan Jalankan default (jika jenis penyimpanan run adalah penyimpanan statis). Untuk informasi selengkapnya tentang konfigurasi penyimpanan run default, lihatMembuat alur kerja menggunakan konsol.
6. Pilih Simpan perubahan untuk menerapkan perubahan.

Memperbarui alur kerja menggunakan CLI

Seperti yang ditunjukkan pada contoh berikut, Anda dapat memperbarui nama alur kerja dan deskripsi. Anda juga dapat mengubah jenis penyimpanan run default (STATIC atau DYNAMIC) dan menjalankan kapasitas penyimpanan (untuk tipe penyimpanan statis). Untuk informasi selengkapnya tentang menjalankan jenis penyimpanan, lihatJalankan jenis penyimpanan dalam HealthOmics alur kerja.

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

Anda tidak menerima tanggapan atas `update-workflow` permintaan tersebut.

Memperbarui alur kerja menggunakan SDK

Anda dapat memperbarui alur kerja menggunakan salah satu file. SDKs

Contoh berikut menunjukkan cara memperbarui alur kerja menggunakan Python SDK

```
import boto3

omics = boto3.client('omics')

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

Menghapus alur kerja pribadi

Ketika Anda tidak lagi membutuhkan alur kerja, Anda dapat menghapusnya menggunakan HealthOmics konsol, perintah AWS CLI, atau salah satu. AWS SDKs Anda dapat menghapus alur kerja yang memenuhi kriteria berikut:

- Statusnya AKTIF atau GAGAL.
- Tidak memiliki saham aktif.

- Anda telah menghapus semua versi alur kerja.

Menghapus alur kerja tidak memengaruhi proses yang sedang berlangsung yang menggunakan alur kerja.

Topik

- Menghapus alur kerja menggunakan konsol
- Menghapus alur kerja menggunakan CLI
- Menghapus alur kerja menggunakan SDK

Menghapus alur kerja menggunakan konsol

Untuk menghapus alur kerja

1. Buka konsol HealthOmics .
2. Jika diperlukan, buka panel navigasi kiri (≡). Pilih Alur kerja pribadi.
3. Pada halaman alur kerja pribadi, pilih alur kerja yang akan dihapus.
4. Pada halaman Alur Kerja, pilih Hapus yang dipilih dari daftar Tindakan.
5. Dalam modal Hapus alur kerja, masukkan “konfirmasi” untuk mengonfirmasi penghapusan.
6. Pilih Hapus.

Menghapus alur kerja menggunakan CLI

Contoh berikut menunjukkan bagaimana Anda dapat menggunakan AWS CLI perintah untuk menghapus alur kerja. Untuk menjalankan contoh, ganti *`workflow id`* dengan ID alur kerja yang ingin Anda hapus.

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

HealthOmics tidak mengirimkan tanggapan atas `delete-workflow` permintaan tersebut.

Menghapus alur kerja menggunakan SDK

Anda dapat menghapus alur kerja menggunakan salah satu file. SDKs

Contoh berikut menunjukkan cara menghapus alur kerja menggunakan Python SDK.

```
import boto3

omics = boto3.client('omics')

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

Verifikasi status alur kerja

Setelah membuat alur kerja, Anda dapat memverifikasi status dan melihat detail alur kerja lainnya menggunakan get-workflow, seperti yang ditunjukkan.

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

Respons mencakup detail alur kerja, termasuk status, seperti yang ditunjukkan.

```
{
    "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"
  }
```

Anda dapat memulai proses menggunakan alur kerja ini setelah transisi status ke. `ACTIVE`

Mereferensikan file genom dari definisi alur kerja

Objek penyimpanan HealthOmics referensi dapat dirujuk dengan URI seperti berikut ini. Gunakan milik Anda sendiri*account ID*,*reference store ID*, dan *reference ID* di mana ditunjukkan.

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

Beberapa alur kerja akan memerlukan `INDEX` file `SOURCE` dan file untuk genom referensi. URI sebelumnya adalah bentuk pendek default dan akan default ke file SOURCE. Untuk menentukan salah satu file, Anda dapat menggunakan formulir URI panjang, sebagai berikut.

```
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
```

Menggunakan set baca urutan akan memiliki pola yang sama, seperti yang ditunjukkan.

```
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
```

Beberapa set baca, seperti yang didasarkan pada FASTQ, dapat berisi pembacaan berpasangan. Dalam contoh berikut, mereka disebut sebagai SOURCE1 dan SOURCE2. Format seperti BAM dan CRAM hanya akan memiliki SOURCE1 file. Beberapa set baca akan berisi file INDEX seperti `bai` atau `crai` file. URI sebelumnya adalah bentuk pendek default dan akan default ke file. SOURCE1 Untuk menentukan file atau indeks yang tepat, Anda dapat menggunakan formulir URI panjang, sebagai berikut.

```
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
```

Berikut ini adalah contoh file JSON input yang menggunakan dua Omics Storage. URIs

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

Referensikan file JSON input di AWS CLI dengan menambahkan permintaan `--inputs file://<input_file.json>` start-run Anda.

Pembuatan versi alur kerja di HealthOmics

Jika Anda perlu membuat perubahan pada alur kerja, Anda dapat membuat alur kerja baru atau versi alur kerja baru. Versi tidak dapat diubah, kecuali untuk perubahan konfigurasi yang diizinkan yang tidak memengaruhi logika eksekusi.

Versi alur kerja memberikan manfaat berikut:

- Versi membentuk kelompok logis alur kerja yang terkait. Anda dapat menambahkan nama yang ditentukan pengguna ke setiap versi alur kerja untuk mengelolanya dengan lebih mudah (terutama untuk alur kerja dengan sejumlah besar versi).
- Anda dapat menjalankan beberapa versi alur kerja secara bersamaan.
- Semua versi alur kerja berbagi ID alur kerja dan ARN dasar yang sama, yang dapat menyederhanakan pengelolaan pipeline setelah Anda memodifikasi alur kerja.
- Versi alur kerja menyediakan tingkat asal data yang sama dengan alur kerja. Versi tidak dapat diubah, dan HealthOmics membuat ARN unik untuk setiap versi alur kerja. Versi ARN menyertakan ID alur kerja dan nama versi, seperti yang ditunjukkan pada contoh berikut:

  `arn:aws:omics:us-west-2:123456789012:workflow/1234567/version/myUniqueVersionName`
- Jika Anda memiliki alur kerja bersama, Anda dapat memperbarui alur kerja tanpa mengganggu pelanggan (yang dapat terus menggunakan versi sebelumnya). Pelanggan dapat mengakses semua versi alur kerja. Jika Anda membuat versi baru, Anda tidak perlu membagikan ulang alur kerja.
- Saat memulai alur kerja, Anda dapat menentukan versi alur kerja.
  - Pengguna dapat memilih untuk tetap pada versi stabil untuk produksi berjalan, dan mencoba versi terbaru untuk uji coba.
  - Pengguna dapat kembali ke alur kerja versi sebelumnya, jika mereka mengalami masalah dengan versi baru.
  - Pelanggan alur kerja bersama dapat memilih versi mana yang akan digunakan.

Topik

- Buat versi alur kerja
- Memperbarui versi alur kerja
- Menghapus versi alur kerja

Versi alur kerja default

Setelah Anda membuat satu atau beberapa versi alur kerja, HealthOmics memperlakukan alur kerja asli sebagai versi default. Saat memulai proses, Anda dapat menentukan versi alur kerja untuk menjalankan secara opsional. Jika Anda tidak menentukan versi saat memulai proses, HealthOmics gunakan versi default.

Di konsol, HealthOmics menunjukkan alur kerja asli dengan label versi Default. Konsol menggunakan label ini hanya setelah Anda membuat satu atau beberapa versi alur kerja. Alur kerja asli selalu tetap menjadi versi default. Anda tidak dapat menetapkan versi lain untuk menjadi default.

Anda tidak dapat menghapus versi default alur kerja jika ada versi lain yang terkait dengan alur kerja. Lihat informasi yang lebih lengkap di Menghapus alur kerja pribadi.

Buat versi alur kerja

Saat membuat alur kerja versi baru, Anda perlu menentukan nilai konfigurasi untuk versi baru. Itu tidak mewarisi nilai konfigurasi apa pun dari alur kerja.

Saat Anda membuat versi, berikan nama versi yang unik untuk alur kerja ini. Anda tidak dapat mengubah nama setelah HealthOmics membuat versi.

Nama versi harus dimulai dengan huruf atau angka dan dapat mencakup huruf besar dan huruf kecil, angka, tanda hubung, titik dan garis bawah. Panjang maksimum adalah 64 karakter. Misalnya, Anda dapat menggunakan skema penamaan sederhana, seperti version1, version2, version3. Anda juga dapat mencocokkan versi alur kerja Anda dengan konvensi versi internal Anda sendiri, seperti 2.7.0, 2.7.1, 2.7.2.

Secara opsional, gunakan kolom deskripsi versi untuk menambahkan catatan tentang versi ini. Sebagai contoh: Fix for syntax error in workflow definition.

> **Note**
>
> Jangan sertakan informasi identitas pribadi (PII) apa pun dalam nama versi. Nama versi muncul di alur kerja versi ARN.

HealthOmics menetapkan ARN unik ke versi alur kerja. ARN unik berdasarkan kombinasi ID alur kerja dan nama versi.

> **Warning**
>
> Setelah menghapus versi alur kerja, Anda dapat HealthOmics menggunakan kembali nama versi untuk versi alur kerja yang berbeda. Praktik terbaik adalah tidak menggunakan kembali nama versi. Jika Anda menggunakan kembali nama, alur kerja dan setiap versi memiliki UUID unik yang dapat Anda gunakan sebagai sumber.

Topik

Buat versi alur kerja menggunakan konsol

Langkah-langkah untuk membuat versi alur kerja

1. Buka konsol HealthOmics .
2. Jika diperlukan, buka panel navigasi kiri (≡). Pilih Alur kerja pribadi.
3. Pada halaman alur kerja pribadi, pilih alur kerja untuk versi baru.
4. Pada halaman Rincian alur kerja, pilih Buat versi baru.
5. Pada halaman Buat versi, berikan informasi berikut:
    1. Nama versi: Masukkan nama untuk versi alur kerja yang unik di seluruh alur kerja.
    2. Deskripsi versi (opsional): Anda dapat menggunakan bidang deskripsi untuk menambahkan catatan tentang versi ini.
6. Di panel definisi Alur Kerja, berikan informasi berikut:
    1. Bahasa alur kerja (opsional): Pilih bahasa spesifikasi untuk versi alur kerja. Jika tidak, HealthOmics tentukan bahasa dari definisi alur kerja.
    2. Untuk sumber definisi Alur Kerja, pilih untuk mengimpor folder definisi dari repositori berbasis Git, lokasi Amazon S3, atau dari drive lokal.

a. Untuk Impor dari layanan repositori:

> **Note**
>
> HealthOmics mendukung repositori publik dan pribadi untukGitHub,,, GitLabBitbucket,GitHub self-managed. GitLab self-managed

i. Pilih Koneksi untuk menghubungkan AWS sumber daya Anda ke repositori eksternal. Untuk membuat koneksi, lihatConnect dengan repositori kode eksternal.

> **Note**
>
> Pelanggan di TLV wilayah tersebut perlu membuat koneksi di wilayah IAD (us-east-1) untuk membuat alur kerja.

ii. Dalam ID repositori lengkap, masukkan ID repositori Anda sebagai nama pengguna/repo-nama. Pastikan Anda memiliki akses ke file di repositori ini.

iii. Dalam referensi Sumber (opsional), masukkan referensi sumber repositori (cabang, tag, atau ID komit). HealthOmics menggunakan cabang default jika tidak ada referensi sumber yang ditentukan.

iv. Di Kecualikan pola file, masukkan pola file untuk mengecualikan folder, file, atau ekstensi tertentu. Ini membantu mengelola ukuran data saat mengimpor file repositori. Ada maksimal 50 pola, dan pola harus mengikuti sintaks pola glob. Contoh:

A. `tests/`

B. `*.jpeg`

C. `large_data.zip`

b. Untuk Pilih folder definisi dari S3:

i. Masukkan lokasi Amazon S3 yang berisi folder definisi alur kerja zip. Bucket Amazon S3 harus berada di wilayah yang sama dengan alur kerja.

ii. Jika akun Anda tidak memiliki bucket Amazon S3, masukkan ID akun pemilik bucket di ID AWS akun pemilik bucket S3. Informasi ini diperlukan agar HealthOmics dapat memverifikasi kepemilikan bucket.

c. Untuk Pilih folder definisi dari sumber lokal:

i. Masukkan lokasi drive lokal dari folder definisi alur kerja zip.

3. Jalur file definisi alur kerja utama (opsional): Masukkan jalur file dari folder definisi alur kerja zip atau repositori ke file. `main` Parameter ini tidak diperlukan jika hanya ada satu file di folder definisi alur kerja, atau jika file utama diberi nama “main”.

7. Di panel file README (opsional), pilih Sumber file README dan berikan informasi berikut:

   - Untuk Impor dari layanan repositori, di jalur file README, masukkan path ke file README dalam repositori.
   - Untuk Pilih file dari S3, dalam file README di S3, masukkan URI Amazon S3 untuk file README.
   - Untuk Pilih file dari sumber lokal: di README - opsional, pilih Pilih file untuk memilih file penurunan harga (.md) yang akan diunggah.

8. Di panel konfigurasi penyimpanan run default, berikan tipe penyimpanan run default dan kapasitas untuk menjalankan yang menggunakan alur kerja ini:

   1. Jalankan jenis penyimpanan: Pilih apakah akan menggunakan penyimpanan statis atau dinamis sebagai default untuk penyimpanan berjalan sementara. Defaultnya adalah penyimpanan statis.
   2. Jalankan kapasitas penyimpanan (opsional): Untuk jenis penyimpanan run statis, Anda dapat memasukkan jumlah default penyimpanan run yang diperlukan untuk alur kerja ini. Nilai default untuk parameter ini adalah 1200 GiB. Anda dapat mengganti nilai default ini saat memulai proses.

9. Tag (opsional): Anda dapat mengaitkan hingga 50 tag dengan versi alur kerja ini.

10. Pilih Berikutnya.

11. Pada halaman Tambahkan parameter alur kerja (opsional), pilih sumber Parameter:

    1. Untuk Parse dari file definisi alur kerja, secara otomatis HealthOmics akan mengurai parameter alur kerja dari file definisi alur kerja.
    2. Untuk menyediakan template parameter dari repositori Git, gunakan path ke file template parameter dari repositori Anda.
    3. Untuk Pilih file JSON dari sumber lokal, unggah JSON file dari sumber lokal yang menentukan parameter.
    4. Untuk Masukkan parameter alur kerja secara manual, masukkan nama dan deskripsi parameter secara manual.

12. Di panel pratinjau Parameter, Anda dapat meninjau atau mengubah parameter untuk versi alur kerja ini. Jika Anda mengembalikan JSON file, Anda kehilangan perubahan lokal yang Anda buat.
13. Pada halaman pemetaan ulang URI Container, di panel aturan Pemetaan, Anda dapat menentukan aturan pemetaan URI untuk alur kerja Anda.

    Untuk Sumber file pemetaan, pilih salah satu opsi berikut:

    - Tidak ada - Tidak ada aturan pemetaan yang diperlukan.
    - Pilih file JSON dari S3 - Tentukan lokasi S3 untuk file pemetaan.
    - Pilih file JSON dari sumber lokal - Tentukan lokasi file pemetaan di perangkat lokal Anda.
    - Masukkan pemetaan secara manual - masukkan pemetaan registri dan pemetaan gambar di panel Pemetaan.
14. Konsol menampilkan panel Pemetaan. Jika Anda memilih file sumber pemetaan, konsol menampilkan nilai dari file.

    a. Dalam pemetaan Registry, Anda dapat mengedit pemetaan atau menambahkan pemetaan (maksimum 20 pemetaan registri).

       Setiap pemetaan registri berisi bidang-bidang berikut:

       - URL registri hulu — URI dari registri hulu.
       - Awalan repositori ECR - Awalan repositori untuk digunakan dalam repositori pribadi Amazon ECR.
       - (Opsional) Awalan repositori hulu - Awalan repositori di registri hulu.
       - (Opsional) ID akun ECR — ID akun akun yang memiliki gambar kontainer hulu.

    b. Dalam pemetaan Gambar, Anda dapat mengedit pemetaan gambar atau menambahkan pemetaan (maksimum 100 pemetaan gambar).

       Setiap pemetaan gambar berisi bidang-bidang berikut:

       - Gambar sumber - Menentukan URI gambar sumber di registri hulu.
       - Gambar tujuan - Menentukan URI gambar yang sesuai di registri ECR Amazon pribadi.
15. Pilih Berikutnya.
16. Tinjau konfigurasi versi, lalu pilih Buat versi.

Saat versi dibuat, konsol kembali ke halaman detail alur kerja dan menampilkan versi baru di tabel Alur Kerja dan versi.

Buat versi alur kerja menggunakan CLI

Anda dapat membuat versi alur kerja menggunakan operasi `CreateWorkflowVersion` API. Untuk parameter opsional, HealthOmics gunakan default berikut:

Parameter	Default
Engine	Ditentukan dari definisi alur kerja
Tipe penyimpanan	Statis
Kapasitas penyimpanan (untuk penyimpanan statis)	1200 GiB
Utama	Ditentukan berdasarkan isi folder definisi alur kerja. Lihat perinciannya di HealthOmics persyaratan definisi alur kerja.
Akselerator	none
Tanda	none

Contoh CLI berikut membuat versi alur kerja dengan penyimpanan statis sebagai penyimpanan run default:

```
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
```

Jika file definisi alur kerja Anda terletak di folder Amazon S3, masukkan lokasi menggunakan `definition-uri` parameter, bukan. `definition-zip` Untuk informasi selengkapnya, lihat CreateWorkflowVersiondi AWS HealthOmics API Referensi.

Anda menerima tanggapan berikut `create-workflow-version` atas permintaan tersebut.

```
{
  "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"
}
```

Membuat versi alur kerja menggunakan SDK

Anda dapat membuat alur kerja menggunakan salah satu. SDKs

Contoh berikut menunjukkan cara membuat versi alur kerja menggunakan Python SDK

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

Verifikasi status versi alur kerja

Setelah membuat versi alur kerja, Anda dapat memverifikasi status dan melihat detail alur kerja lainnya menggunakan get-workflow-version, seperti yang ditunjukkan.

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

Respons memberi Anda detail alur kerja Anda, termasuk status, seperti yang ditunjukkan.

```
{
  "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"
}
```

Sebelum Anda dapat memulai proses dengan versi alur kerja ini, status harus beralih ke`ACTIVE`.

Memperbarui versi alur kerja

Anda dapat memperbarui deskripsi dan konfigurasi penyimpanan run default untuk versi alur kerja pribadi. Untuk mengubah informasi lain dalam versi alur kerja, buat versi baru.

Topik

- Memperbarui versi alur kerja menggunakan konsol
- Perbarui versi alur kerja menggunakan CLI
- Memperbarui versi alur kerja menggunakan SDK

Memperbarui versi alur kerja menggunakan konsol

Untuk memperbarui versi alur kerja

1. Buka konsol HealthOmics .
2. Jika diperlukan, buka panel navigasi kiri (≡). Pilih Alur kerja pribadi.
3. Pada halaman alur kerja pribadi, pilih alur kerja.

4. Pada halaman Alur Kerja, pilih versi alur kerja yang akan diperbarui dan pilih Edit dipilih dari daftar Tindakan.

   - Jika Anda memilih versi default, konsol akan membuka halaman Edit alur kerja. Untuk informasi selengkapnya, lihat Memperbarui alur kerja pribadi.
   - Jika Anda memilih versi yang ditentukan pengguna, konsol akan membuka halaman Edit versi.
5. Pada halaman Edit versi, berikan informasi berikut

   - Deskripsi versi (opsional) - Deskripsi versi ini.
6. Di panel konfigurasi penyimpanan jalankan default, berikan nilai default berikut untuk proses yang menggunakan versi alur kerja ini. Anda dapat mengganti nilai default saat memulai proses:

   - Untuk jenis penyimpanan Jalankan, pilih Statis atau Dinamis.
   - Untuk penyimpanan run statis, pilih jumlah default kapasitas penyimpanan Jalankan untuk proses yang menggunakan versi alur kerja ini. Nilai default untuk parameter ini adalah 1200 GiB.
7. Pilih Simpan perubahan.

Konsol kembali ke halaman detail alur kerja dan menampilkan spanduk halaman dengan versi alur kerja yang diperbarui.

Perbarui versi alur kerja menggunakan CLI

Anda dapat memperbarui parameter untuk versi alur kerja menggunakan perintah CLI berikut. Kombinasi ID alur kerja dan nama versi secara unik mengidentifikasi versi.

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

Anda tidak menerima tanggapan `update-workflow-version` atas permintaan tersebut.

Memperbarui versi alur kerja menggunakan SDK

Anda dapat memperbarui versi alur kerja menggunakan salah satu versi. SDKs Contoh SDK python berikut menunjukkan cara memperbarui jenis penyimpanan dan deskripsi untuk versi alur kerja.

```
import boto3

omics = boto3.client('omics')

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

Menghapus versi alur kerja

Anda dapat menghapus versi alur kerja yang ditentukan pengguna menggunakan konsol, CLI, atau salah satu versi. SDKs Menghapus versi alur kerja tidak memengaruhi proses yang sedang berlangsung yang menggunakan versi alur kerja.

Anda tidak dapat menghapusVersi alur kerja default. Anda menghapus semua versi yang ditentukan pengguna, lalu menghapus alur kerja.

Topik

- Hapus versi alur kerja menggunakan konsol
- Hapus versi alur kerja menggunakan CLI
- Menghapus versi alur kerja menggunakan SDK

Hapus versi alur kerja menggunakan konsol

Untuk menghapus versi alur kerja

1. Buka konsol HealthOmics .
2. Jika diperlukan, buka panel navigasi kiri (≡). Pilih Alur kerja pribadi.
3. Pada halaman alur kerja pribadi, pilih alur kerja.
4. Pada halaman Alur Kerja, pilih versi alur kerja yang akan dihapus dan pilih Hapus yang dipilih dari daftar Tindakan.
5. Dalam modal versi Hapus alur kerja, masukkan “konfirmasi” untuk mengonfirmasi penghapusan.
6. Pilih Hapus.

Konsol menampilkan spanduk halaman dengan versi alur kerja yang dihapus.

Hapus versi alur kerja menggunakan CLI

Anda dapat menghapus versi alur kerja yang ditentukan pengguna menggunakan perintah CLI berikut. Kombinasi ID alur kerja dan nama versi secara unik mengidentifikasi versi.

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

Anda tidak menerima tanggapan `delete-workflow-version` atas permintaan tersebut.

Menghapus versi alur kerja menggunakan SDK

Anda dapat menghapus alur kerja menggunakan salah satu file. SDKs

Contoh berikut menunjukkan cara menghapus alur kerja menggunakan Python SDK.

```
import boto3

omics = boto3.client('omics')

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

Menggunakan HealthOmics run

Setelah Anda membuat alur kerja, Anda dapat mulai berjalan menggunakan alur kerja.

Saat Anda memulai proses, HealthOmics mengalokasikan penyimpanan run sementara untuk mesin alur kerja yang akan digunakan selama proses. Untuk memastikan isolasi dan keamanan data HealthOmics , berikan penyimpanan pada awal setiap proses, dan hentikan penyediaannya di akhir proses.

HealthOmics menyediakan beberapa kuota yang terkait dengan alur kerja dan tugas. Nilai default sengaja konservatif, untuk membantu Anda menghindari pembengkakan biaya yang tidak terduga.

Anda dapat meminta peningkatan kuota ini. Untuk informasi selengkapnya, lihat HealthOmics kuota layanan.

Saat Anda memulai proses, HealthOmics tetapkan ID run dan run uuid ke run. Berjalan di akun memiliki proses yang unik IDs. Namun, HealthOmics menggunakan kembali run yang dihapus IDs, sehingga run dan run yang dihapus dapat memiliki ID run yang sama. Selain itu, sangat jarang tetapi mungkin alur kerja bersama memiliki ID run yang sama dengan menjalankan di akun Anda.

run uuidIni adalah Pengenal Unik Global (guid) yang dapat Anda gunakan untuk mengidentifikasi berjalan di seluruh akun atau untuk membedakan antara dua proses di akun Anda yang memiliki ID run yang sama.

> **Note**
>
> Untuk tujuan asal data, kami menyarankan Anda menggunakan run uuid untuk mengidentifikasi proses secara unik. run uuidIni juga merupakan pengidentifikasi terbaik untuk ditautkan ke sistem manajemen informasi lab internal Anda (LIMs) atau sistem pelacakan sampel.

Anda dapat menggunakan Amazon Q CLI untuk mengoptimalkan proses dan menganalisis kinerja lari. Untuk informasi selengkapnya, lihat Contoh prompt untuk Amazon Q CLI dan tutorial AI HealthOmics generatif Agentic. GitHub

Topik

- Jalankan jenis penyimpanan dalam HealthOmics alur kerja
- Jalankan mode retensi untuk HealthOmics berjalan
- HealthOmics jalankan masukan
- Jalankan siklus hidup dalam alur kerja HealthOmics
- HealthOmics jalankan output
- Jalankan alasan kegagalan
- Siklus hidup tugas dalam proses HealthOmics
- Jalankan pengoptimalan untuk HealthOmics alur kerja pribadi
- Jalankan operasi di HealthOmics

Jalankan jenis penyimpanan dalam HealthOmics alur kerja

Saat Anda memulai proses, HealthOmics mengalokasikan penyimpanan run sementara untuk mesin alur kerja yang akan digunakan selama proses. HealthOmicsmenyediakan penyimpanan run sementara sebagai sistem file.

Untuk alur kerja atau alur kerja tertentu, Anda dapat memilih penyimpanan run dinamis atau statis. Secara default, HealthOmics menyediakan penyimpanan Dynamic run.

> **Note**
>
> Jalankan penggunaan penyimpanan menimbulkan biaya ke akun Anda. Untuk informasi harga tentang penyimpanan run statis dan dinamis, lihat HealthOmicsharga.

Bagian berikut memberikan informasi untuk dipertimbangkan saat memutuskan jenis penyimpanan run mana yang akan digunakan.

Penyimpanan berjalan dinamis

Sebaiknya gunakan penyimpanan run dinamis untuk sebagian besar proses, termasuk proses yang memerlukan waktu mulai lebih cepat, berjalan di mana Anda tidak mengetahui kebutuhan penyimpanan sebelumnya, dan untuk siklus pengujian pengembangan berulang.

Anda tidak perlu memperkirakan penyimpanan atau throughput yang diperlukan untuk menjalankan. HealthOmics secara dinamis menskalakan ukuran penyimpanan ke atas atau ke bawah, berdasarkan pemanfaatan sistem file selama dijalankan. HealthOmics juga secara dinamis menskalakan throughput berdasarkan kebutuhan alur kerja. Jalankan tidak pernah gagal karena kehabisan penyimpanan untuk kesalahan sistem file.

Dynamic run storage memberikan provisioning/deprovisioning waktu yang lebih cepat daripada penyimpanan run statis. Pengaturan yang lebih cepat merupakan keuntungan bagi sebagian besar alur kerja dan juga merupakan keuntungan selama development/test siklus.

Setelah proses selesai (jalur sukses atau jalur gagal), operasi GetRun API mengembalikan penyimpanan maksimum yang digunakan oleh proses di bidang StorageCapacity. Anda juga dapat menemukan informasi ini di log manifes run yang terletak di grup omics log. Untuk proses penyimpanan dinamis yang selesai dalam 2 jam, nilai penyimpanan maksimum mungkin tidak tersedia.

Untuk penyimpanan run dinamis, run menyediakan sistem file yang menggunakan protokol NFS. NFS memperlakukan CREATE, DELETE, dan RENAME operasi file sebagai non-idempoten, yang kadang-kadang dapat menyebabkan kondisi balapan untuk operasi ini yang perlu ditangani kode Anda dengan anggun. Misalnya, kode Anda tidak boleh gagal jika mencoba menghapus file yang tidak ada. Sebelum mengadopsi penyimpanan run dinamis, sebaiknya sesuaikan kode alur kerja Anda agar tahan terhadap operasi file non-idempoten. Lihat Contoh kode untuk penanganan operasi non-idempoten yang aman.

Contoh kode untuk penanganan operasi non-idempoten yang aman

Contoh python berikut menunjukkan cara menghapus file tanpa gagal jika file tidak ada.

```
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")
```

Contoh berikut menggunakan shell Bash. Untuk menghapus file dengan aman meskipun tidak ada, gunakan:

```
rm -f my_file
```

Untuk memindahkan (mengganti nama) file dengan aman, jalankan perintah move hanya jika file `old_name` ada di direktori saat ini.

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

Untuk membuat direktori, gunakan perintah berikut:

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

Penyimpanan berjalan statis

Untuk penyimpanan run statis, run menyediakan sistem file yang menggunakan protokol Lustre. Protokol ini tahan terhadap operasi file non-idempoten secara default. Anda tidak perlu menyesuaikan kode alur kerja Anda untuk menangani operasi file non-idempoten.

HealthOmics mengalokasikan jumlah penyimpanan run yang tetap. Anda menentukan nilai ini ketika Anda memulai proses. Penyimpanan run default adalah 1200 GiB, jika Anda tidak menentukan nilai. Saat Anda menentukan nilai untuk ukuran penyimpanan dalam permintaan StartRun API, sistem akan membulatkan nilai ke kelipatan terdekat 1200 GiB. Jika ukuran penyimpanan itu tidak tersedia, itu membulatkan ke kelipatan terdekat 2400 GiB.

Untuk penyimpanan run statis, HealthOmics berikan nilai throughput berikut:

- Throughput dasar 200 per MB/s TiB dari kapasitas penyimpanan yang disediakan.
- Throughput burst hingga 1300 per MB/s TiB dari kapasitas penyimpanan yang disediakan.

Jika ukuran penyimpanan yang ditentukan terlalu rendah, proses gagal dengan kesalahan Kehabisan penyimpanan untuk sistem file. Penyimpanan yang dijalankan statis sangat cocok untuk alur kerja yang dapat diprediksi dengan persyaratan penyimpanan yang diketahui.

Static run storage cocok untuk beban kerja yang besar dan meledak dengan konkurensi tugas tinggi (misalnya, sejumlah besar RNASeq sampel yang diproses secara paralel). Ini memberikan throughput sistem file yang lebih tinggi per GiB dan biaya per GiB yang lebih rendah daripada penyimpanan run dinamis.

Menghitung penyimpanan run statis yang diperlukan

Alur kerja membutuhkan kapasitas tambahan ketika menggunakan penyimpanan run statis (dibandingkan dengan penyimpanan run dinamis) karena instalasi sistem file dasar menggunakan 7% dari kapasitas sistem file statis.

Jika Anda menjalankan alur kerja penyimpanan dinamis untuk mengukur penyimpanan maksimum yang digunakan oleh proses, gunakan perhitungan berikut untuk menentukan jumlah minimum penyimpanan statis yang diperlukan:

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

Contoh:

```
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.
```

Oleh karena itu, 1200GiB (kapasitas minimum untuk penyimpanan run statis) sudah cukup untuk menjalankan ini.

Jalankan mode retensi untuk HealthOmics berjalan

Setelah proses selesai, HealthOmics arsipkan metadata run ke. CloudWatch Secara default, CloudWatch simpan data yang dijalankan tanpa batas waktu, kecuali Anda mengubah kebijakan CloudWatch penyimpanan. Output yang dijalankan juga disimpan di Amazon S3 hingga Anda menghapusnya.

Salah satu yang dapat disesuaikan HealthOmics kuota layanan adalah maximum number of runs (active and inactive) di suatu wilayah. HealthOmics mempertahankan run metatdata hingga jumlah run ini untuk digunakan oleh operasi konsol dan API (ListRuns dan). GetRun Saat memulai proses, Anda dapat menyetel parameter mode retensi jalankan untuk menunjukkan perilaku retensi untuk proses tersebut. Parameter mendukung nilai REMOVE dan RESTAIN.

Untuk menjalankan baru dengan mode retensi disetel ke REMOVE, jika HealthOmics mencoba menambahkan run setelah menyimpan jumlah maksimum run, secara otomatis menghapus metadata untuk proses tertua yang telah mengatur mode REMOVE. Penghapusan ini tidak memengaruhi data yang disimpan di CloudWatch atau Amazon S3.

RETAIN adalah nilai default untuk menjalankan mode retensi. Untuk berjalan dalam mode ini, sistem tidak menghapus metadata run. Jika HealthOmics mencapai jumlah maksimum run, semua diatur ke RESTAIN, Anda tidak akan dapat membuat run tambahan sampai Anda menghapus beberapa run.

Jika Anda berencana untuk menjalankan batch lebih dari jumlah maksimum run pada saat yang sama, pastikan untuk mengatur mode retensi run ke REMOVE. Jika tidak, batch gagal saat HealthOmics mencoba memulai proses berikutnya setelah maksimum.

Pertimbangan tambahan untuk menggunakan mode retensi HAPUS:

- Saat pertama kali mulai menggunakan REMOVE sebagai mode retensi, pertimbangkan untuk menghapus satu atau lebih proses yang menggunakan mode RETAIN, untuk membebaskan slot.

Saat Anda memulai proses REMOVE tambahan, penghapusan otomatis mengambil alih, sehingga cukup slot tersedia untuk proses baru.

- Jika Anda ingin menjalankan ulang run yang diarsipkan (atau satu set run), gunakan alat CLI HealthOmics runun ulang. Untuk informasi selengkapnya dan contoh cara menggunakan alat ini, lihat Omics menjalankan ulang di repositori HealthOmics alat GitHub .
- Kami menyarankan Anda mengonfigurasi nama unik untuk setiap proses. Setelah HealthOmics menghapus proses, Anda tidak dapat menggunakan konsol atau API untuk menemukan nama jalankan atau menjalankan ID. Namun, Anda dapat menggunakan CloudWatch untuk mencari nama run, jadi gunakan nama unik untuk mendapatkan hasil pencarian terbaik.
- Anda dapat menggunakan CloudWatch start-query perintah untuk mendapatkan informasi tentang proses yang diarsipkan. Jika nama run tidak unik, kueri dapat mengembalikan beberapa manifes. Parameter waktu mulai dan akhir waktu menentukan rentang waktu untuk pencarian.

```
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>
```

start-queryPerintah mengembalikan ID query. Melewati ID kueri ke get-query-results perintah mengembalikan hasil kueri.

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

HealthOmics jalankan masukan

Jika definisi alur kerja menentukan file input untuk alur kerja atau tugas alur kerja, HealthOmics tahapkan file ke volume awal yang didedikasikan untuk menjalankan alur kerja. File input ini hanya-baca, yang mencegah tugas memodifikasi input potensial ke tugas lain dalam alur kerja. Untuk impor direktori, direktori juga hanya-baca.

Banyak aplikasi genomik berasumsi bahwa file indeks ditempatkan bersama dengan file urutan (seperti file pendamping untuk `bai` file). `bam` Untuk menyertakan file indeks, tentukan sebagai input tugas dalam definisi alur kerja.

Topik

- Mengelola ukuran parameter run

- Format parameter masukan Amazon S3
- Status arsip masukan Amazon S3

Mengelola ukuran parameter run

Saat Anda memulai proses, Anda menentukan input run di objek atau file JSON parameter run. Anda dapat menentukan hingga 50 KB parameter run untuk alur kerja. Anda dapat menggunakan teknik berikut untuk tetap berada dalam batasan ukuran ini:

- Gunakan impor direktori

  Untuk menentukan sejumlah besar file input, tentukan satu parameter sebagai lokasi Amazon S3 yang berisi semua file, daripada menentukan parameter untuk setiap lokasi file. Untuk informasi selengkapnya, lihat topik berikutnya (format parameter input Amazon S3).

- Gunakan lembar sampel

  Lembar sampel adalah file CSV atau TSV dengan satu kolom untuk alamat fastq.gz (atau dua untuk pembacaan berpasangan) dan kolom tambahan untuk metadata seperti nama sampel. Anda menentukan lembar sampel sebagai parameter input run, bukan parameter untuk setiap file input.

  Alur kerja Anda menentukan bagaimana lembar sampel Anda memetakan ke struktur data dalam alur kerja. Meskipun Anda dapat menulis kode untuk lembar sampel di WDL dan CWL, mereka lebih umum di. NextFlow Sebagai contoh, lihat lembar sampel di situs nf-core GitHub .

Format parameter masukan Amazon S3

Untuk parameter input yang menerima lokasi Amazon S3, parameter dapat menentukan lokasi satu file atau seluruh direktori file. Menggunakan direktori memiliki keuntungan sebagai berikut:

- Kenyamanan - Anda menentukan nama direktori sebagai parameter. Anda tidak mencantumkan setiap nama file.
- Kekompakan - Parameter input ukuran file maksimum adalah 50 KB. Jika Anda memberikan daftar panjang nama file input, Anda dapat melebihi maksimum ini.

Amazon S3 adalah sistem penyimpanan objek datar, sehingga tidak mendukung direktori. Anda mengelompokkan file ke dalam “direktori” dengan memberikan setiap file key prefix objek yang

sama. Untuk informasi selengkapnya tentang awalan kunci objek Amazon S3, lihat Mengatur objek menggunakan awalan.

HealthOmics menafsirkan nilai parameter masukan sebagai berikut:

- Jika lokasi Amazon S3 tidak diakhiri dengan garis miring ke depan atau menggunakan pola glob, HealthOmics mengharapkan nilai parameter menjadi kunci untuk satu objek Amazon S3.

  Misalnya, Anda menentukan untuk memasukkan `s3://myfiles/runs/inputs/a/file1.fastq` file1.fastq
- Jika lokasi Amazon S3 diakhiri dengan garis miring ke depan, HealthOmics interpretasikan nilai parameter sebagai awalan Amazon S3. Ini memuat semua objek Amazon S3 dengan awalan itu.

  Misalnya, Anda dapat menentukan `s3://myfiles/runs/inputs/a/` untuk memuat semua objek yang kuncinya dimulai dengan awalan ini.
- Untuk Nextflow, mendukung HealthOmics sebagian pola glob untuk Amazon S3 dalam parameter input. URIs

  Misalnya, Anda dapat menentukan "`s3://myfiles/runs/inputs/a/*.gz`" untuk memasukkan semua file.gz yang kuncinya dimulai dengan awalan ini.

Nextflow Penanganan pola Glob di input Amazon S3

Pola Glob	HealthOmics Perilaku Pertandingan	Catatan
s3://bucket/directory/*.txt	Cocokkan semua `.txt` objek pada kedalaman apa pun di bawah awalan s3://buck et/directory/. For example, matches s3://bucket/directory/ abc.txt or s3://bucket/directory/ subDir/123.txt dll.	
s3://bucket/directory/**/*.txt	Cocokkan semua `.txt` objek pada kedalaman apa pun di bawah awalan s3://buck et/directory/. For example,	Di S3, ** setara dengan. *

Pola Glob	HealthOmics Perilaku Pertandingan	Catatan
	matches s3://bucket/directory/abc.txt or s3://bucket/directory/subDir/123.txt dll.	
s3://bucket/directory/{a,b}.txt	s3://bucket/directory/a.txt, s3://bucket/directory/b.txt	
s3://bucket/directory/?. txt	Mencocokkan objek di root awalan yang nama filenya adalah satu karakter diikuti oleh. `.txt` Misalnya, cocok dengan s3://bucket/directory/a.txt but not s3://bucket/directory/someDir/a.txt or s3://bucket/directory/someDir/subDir/a.txt	
s3://bucket/directory/[0-9].txt	s3:///9.txt bucket/directory/0.txt, s3://bucket/directory/1.txt, ... ,s3://bucket/directory	
s3://bucket/directory/[0-9].txt	s3:///3.txt bucket/directory/1.txt, s3://bucket/directory/2.txt, s3://bucket/directory	
s3://bucket/directory/[0-9].txt	s3://bucket/directory/b.txt, s3://bucket/directory/c.txt, ... ,s3://bucket/directory/Y.txt	

Penanganan garis miring ganda khusus bahasa di input Amazon S3

HealthOmics mempertahankan perilaku mesin asli untuk setiap mesin alur kerja saat menangani garis miring ganda di Amazon S3 URIs, sehingga Anda tidak perlu membuat perubahan apa pun pada alur kerja saat memigrasikannya. HealthOmics Bagian berikut menjelaskan bagaimana setiap mesin menangani berbagai skenario.

WDL

Jika parameter input menyertakan garis miring ganda di tengah atau di ujung URI, mesin WDL mempertahankan garis miring ganda.

Parameter masukan	Lokasi yang diharapkan	
s3://myfiles/runs/inputs//file1.fastq	s3://myfiles/runs/inputs//file1.fastq	
s3:////myfiles/runs/inputs	s3:////myfiles/runs/inputs	

Alur berikutnya

Jika parameter input menyertakan garis miring ganda di tengah URI, mesin Nextflow mempertahankan garis miring ganda. Untuk garis miring ganda di akhir URI, mesin Nextflow menyelesaikannya menjadi satu garis miring.

Parameter masukan	Lokasi yang diharapkan	
s3://myfiles/runs/inputs//file1.fastq	s3://myfiles/runs/inputs//file1.fastq	
s3://myfiles//runs/inputs//*.gz	s3://myfiles//runs/inputs//*.gz	
s3://myfiles//runs/inputs//	s3://myfiles//runs/inputs/	

CWL

Jika parameter input menyertakan garis miring ganda di tengah atau di ujung URI, mesin CWL mempertahankan garis miring ganda.

Parameter masukan	Lokasi yang diharapkan	
s3://myfiles// runs/inputs//file1.fastq	s3://myfiles// runs/inputs//file1.fastq	
s3://myfiles//runs/inputs//	s3://myfiles//runs/inputs//	

Status arsip masukan Amazon S3

HealthOmics dapat mengambil objek Amazon S3 yang dikirim S3 secara real time. Untuk objek yang berada dalam status penyimpanan yang diarsipkan restore berikut, objek untuk membuatnya tersedia untuk HealthOmics:

- Pengambilan Fleksibel atau kelas penyimpanan Deep Archive di Amazon S3 Glacier.
- Akses yang Diarsipkan atau Tingkat Akses Arsip Dalam dalam tingkatan Cerdas.

Untuk informasi tentang memulihkan objek, lihat Memulihkan objek yang diarsipkan di Panduan Pengguna Amazon S3.

Jalankan siklus hidup dalam alur kerja HealthOmics

Anda dapat melacak kemajuan proses dengan memantau status run. HealthOmics memperbarui status run saat proses berjalan melalui siklus hidupnya.

Anda dapat mengambil status run menggunakan salah satu metode berikut:

- HealthOmics Konsol menampilkan status setiap proses pada Runs halaman.
- Operasi GetRun API mengembalikan status run saat ini.
- Anda dapat memantau status lari menggunakan EventBridge acara. Untuk informasi selengkapnya, lihat Menggunakan EventBridge dengan AWS HealthOmics.

Topik

- Jalankan nilai status
- Tugas Mencoba Ulang
- Implikasi harga dari status run

Jalankan nilai status

Saat Anda memulai proses, HealthOmics atur status run kePending. Saat proses berjalan melalui siklus hidupnya, HealthOmics memperbarui nilai status untuk mencerminkan kemajuannya saat ini.

> **Note**
>
> Anda tidak dikenakan biaya selama status run selain Running. Untuk informasi detail, lihat bagian berikutnya.

HealthOmics mendukung nilai status run berikut:

Tertunda

Jalankan dalam antrian, menunggu untuk memulai. Runs biasanya tetap di Pending untuk waktu yang singkat sebelum mereka mulai.

- Runs dapat tetap di Pending untuk waktu yang lebih lama jika Anda mengirimkan banyak pekerjaan pada saat yang bersamaan.
- Runs tetap di Pending setelah akun Anda mencapai jumlah maksimum proses bersamaan.
- Run tetap di Pending jika run adalah bagian dari grup run yang telah mencapai nilai maksimum sumber dayanya.
- Anda dapat menyesuaikan prioritas lari sehingga proses antrian tertentu dimulai sebelum yang lain. Untuk informasi selengkapnya tentang menjalankan prioritas, lihatJalankan prioritas.

Starting

HealthOmics membuat run dan menyediakan sumber daya yang diperlukan untuk menjalankan (seperti penyimpanan run sementara dan node mesin).

- HealthOmics menyediakan penyimpanan run sementara pada awal proses, dan membatalkan ketentuan penyimpanan run saat proses Berhenti.

Berjalan

Run tetap dalam status Running selama proses impor, pemrosesan setiap tugas, dan proses ekspor.

- HealthOmics mengimpor file input ke sistem file penyimpanan sementara. File input hanya baca, untuk mencegah tugas memodifikasi input ke tugas lain dalam alur kerja.
- Selama ekspor file, HealthOmics ekspor file output dari sistem file penyimpanan run ke lokasi S3.
- HealthOmics mengirimkan log run dan log tugas CloudWatch secara real time saat status run sedang Berjalan. Untuk informasi selengkapnya, lihat Log masuk CloudWatch.

Stopping

Setelah menyelesaikan proses ekspor, proses transisi ke status Berhenti.

- HealthOmics membatalkan semua sumber daya (termasuk sistem file penyimpanan yang dijalankan dan simpul mesin).

Selesai

Run bertransisi ke Selesai setelah HealthOmics menyelesaikan deprovisioning sumber daya.

- HealthOmics telah menyelesaikan semua tugas yang dijalankan dan mengekspor data keluaran tanpa kesalahan.
- Output run tersedia di lokasi output Amazon S3 URI yang ditentukan. Untuk WDL dan CWL, HealthOmics menghasilkan file ringkasan keluaran run, yang memberikan informasi tentang file. HealthOmics jalankan output
- Log manifes proses akhir dan log mesin (jika ada) tersedia di CloudWatch.
- Untuk menjalankan yang mendukung percobaan ulang tugas, menjalankan dengan status Selesai dapat menyertakan satu atau beberapa tugas yang gagal. Selama percobaan ulang tugas berhasil untuk setiap tugas yang gagal, HealthOmics transisi run ke Selesai. HealthOmics menetapkan ID tugas baru untuk setiap percobaan ulang, sehingga proses menyertakan tugas IDs untuk upaya yang gagal dan upaya yang diselesaikan.

Failed

HealthOmics mengalami satu atau lebih kesalahan dan gagal menyelesaikan semua tugas yang dijalankan.

- Proses yang gagal mentransisikan melalui status Berhenti saat HealthOmics membatalkan ketentuan sumber daya.

Dibatalkan

Pengguna memulai permintaan untuk membatalkan proses.

- HealthOmics menghentikan tugas yang sedang berjalan dan membatalkan semua sumber daya.
- HealthOmics tidak mengekspor data keluaran run apa pun saat pengguna membatalkan proses. Anda tidak memiliki akses ke file perantara apa pun untuk proses yang dibatalkan.
- Akun Anda dikenakan biaya untuk tugas dan sumber daya yang digunakan proses selama status Running sebelum pembatalan.
- Tidak ada biaya jika Anda membatalkan proses dalam status Tertunda atau Mulai.

Tugas Mencoba Ulang

HealthOmics mendukung percobaan ulang tugas untuk tugas yang gagal karena kesalahan layanan (kode status HTTP 5XX).

Jika setiap tugas dalam proses akhirnya selesai, bahkan jika mereka memerlukan percobaan ulang, HealthOmics transisi run ke Selesai. HealthOmics menetapkan ID tugas baru untuk setiap percobaan ulang, sehingga proses menyertakan tugas IDs untuk upaya yang gagal dan upaya yang diselesaikan.

Perilaku coba ulang default bergantung pada bahasa definisi yang digunakan alur kerja. Default untuk Nextflow adalah tidak ada percobaan ulang. Untuk WDL dan CWL, mencoba hingga dua HealthOmics percobaan ulang tugas yang gagal, tetapi Anda dapat memilih keluar dari percobaan ulang tugas untuk tugas tertentu atau untuk semua tugas dalam alur kerja. Coba lagi tugas berguna untuk mengatasi kesalahan layanan intermiten. Namun, Anda mungkin mempertimbangkan untuk memilih keluar dari tugas yang idempoten.

Untuk informasi spesifik tentang setiap bahasa definisi alur kerja, lihat topik berikut:

- WDL - Konfigurasikan perilaku coba lagi tugas dalam definisi alur kerja. Lihat Mengonfigurasi perilaku coba ulang tugas WDL.
- Nextflow — Konfigurasikan perilaku coba ulang tugas di file konfigurasi Nextflow atau definisi alur kerja. Lihat Mengonfigurasi perilaku coba lagi tugas Nextflow.
- CWL - Konfigurasikan perilaku coba lagi tugas dalam definisi alur kerja. Lihat Mengonfigurasi perilaku coba ulang tugas CWL.

Implikasi harga dari status run

Akun Anda dapat dikenakan biaya saat status run sedang berjalan. Anda tidak dikenakan biaya selama status run lainnya. Misalnya, tidak ada biaya untuk sumber daya saat proses dimulai atau berhenti.

Run dengan status Running memiliki implikasi penagihan berikut:

- Akun Anda dikenakan biaya untuk penggunaan sistem file penyimpanan yang dijalankan saat status run sedang berjalan. Untuk informasi tentang jenis penyimpanan yang dijalankan, LihatJalankan jenis penyimpanan dalam HealthOmics alur kerja.
- Akun Anda dikenakan biaya untuk menjalankan tugas, berdasarkan sumber daya komputasi dan memori yang Anda tentukan untuk setiap tugas dalam definisi alur kerja, dan berdasarkan

durasi tugas. Untuk informasi selengkapnya, lihat Persyaratan komputasi dan memori untuk tugas HealthOmics.

- Setiap tugas memiliki ambang batas penagihan minimum satu menit. Jika Anda menjalankan tugas kurang dari satu menit, Anda dikenakan biaya untuk penggunaan minimum satu menit. Jika memungkinkan, kelompokkan tugas-tugas kecil bersama untuk mengoptimalkan biaya. Tugas pengelompokan juga mengurangi waktu berjalan dengan menghindari spin-up beberapa tugas berurutan.

Untuk informasi tambahan tentang HealthOmics harga, lihat HealthOmics Harga.

HealthOmics jalankan output

Ketika WDL atau CWL run selesai, output menyertakan file ringkasan output (dalam format JSON) yang mencantumkan semua output yang dihasilkan oleh run. Anda dapat menggunakan file ringkasan keluaran untuk tujuan ini:

- Secara terprogram menentukan file output yang dihasilkan run.
- Validasi bahwa run menghasilkan semua output yang diharapkan.

Topik

- Jalankan ringkasan keluaran untuk WDL
- Jalankan ringkasan keluaran untuk CWL

Jalankan ringkasan keluaran untuk WDL

Ketika WDL run selesai, HealthOmics membuat file ringkasan output bernama. output.json

Untuk setiap output alur kerja, ada key/value pasangan yang sesuai dalam file. Kunci berisi nama alur kerja dan nama output dalam format berikut:`WorkflowName.output_name`. Untuk output file, nilainya adalah URI S3 yang menunjuk ke lokasi output di S3 tempat file disimpan. Untuk output Array [File], nilainya adalah array S3 URIs.

Contoh berikut menunjukkan output.json file untuk alur kerja bernamaBWAMappingWorkflow.

```
{
  "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"
  ]
}
```

Jika alur kerja menghasilkan output dengan tipe non-file (seperti String, Int, Float, atau Bool), nilai bidang adalah primitif JSON. Misalnya:

```
{
  "MyWorkflow.my_int_ouput": 1,
  "MyWorkflow.my_bool_output": false,
    ...
}
```

Jalankan ringkasan keluaran untuk CWL

Saat proses CWL selesai, HealthOmics buat file ringkasan keluaran bernama outputs.json di lokasi berikut:

```
{my-S3outputpath}/{runId}/{run-uuid}/logs/outputs.json
```

File ringkasan output mencakup daftar output. Setiap output adalah key/value pasangan, di mana kuncinya adalah nama output. Nilai adalah objek yang mencakup properti berikut:

- lokasi - Jalur yang sepenuhnya memenuhi syarat ke file output
- basename — Bagian nama file dari jalur
- class — Jenis output, yang biasanya File
- ukuran — Ukuran file dalam byte

Dalam contoh berikut, file output.json memiliki daftar dua file output.

```
{
  "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
  }
}
```

Jalankan alasan kegagalan

Jika proses gagal, gunakan operasi GetRunAPI untuk mengambil alasan kegagalan.

Tinjau alasan kegagalan untuk membantu Anda memecahkan masalah mengapa proses gagal. Tabel berikut mencantumkan setiap alasan kegagalan bersama dengan deskripsi kesalahan.

Alasan kegagalan	Deskripsi kesalahan
ASSUME_ROLE_FAILED	HealthOmics tidak memiliki izin untuk mengambil peran. Tentukan HealthOmics kepala sekolah dalam hubungan kepercayaan untuk peran tersebut.
CANNOT_START_CONTAINER_ERROR	Tidak dapat memulai tugas alur kerja:*name*, id: *ID* wadah menggunakan gambar:*image name*. Pastikan gambar tersebut valid dan coba lagi.

Alasan kegagalan	Deskripsi kesalahan
CANNOT_START_CONTAINER_SIZE_ERROR	Tidak dapat memulai tugas alur kerja:*name*, id: *ID* wadah menggunakan gambar:*image name*. Pastikan ukuran gambar kurang dari 45 GiB (95 GiB untuk instance GPU) dan coba lagi.
ECR_PERMISSION_ERROR	HealthOmics tidak memiliki izin untuk mengakses URI gambar. Konfirmasikan bahwa repositori pribadi Amazon ECR ada dan telah memberikan akses ke kepala layanan. HealthOmics
EXPORT_FAILED	Ekspor gagal. Periksa apakah bucket keluaran ada dan peran run memiliki izin tulis ke bucket.
FILE_SYSTEM_OUT_OF_SPACE	Sistem file tidak memiliki cukup ruang. Tingkatkan ukuran sistem file dan jalankan lagi.
IMAGE_VERIFICATION_FAILURE	Tidak dapat memverifikasi gambar*image name*. Untuk memperbaiki masalah, coba tarik gambar dan kemudian dorong ke repositori ECR Anda lagi.
IMPORT_FAILED	Impor gagal. Periksa apakah file input ada dan peran run dapat mengakses input.
INACTIVE_OMICS_STORAGE_RESOURCE	URI HealthOmics penyimpanan tidak dalam status AKTIF. Aktifkan set baca dan coba lagi. Untuk mempelajari lebih lanjut tentang mengaktifkan set baca, lihatMengaktifkan set baca di HealthOmics.
INPUT_URI_NOT_FOUND	URI yang disediakan tidak ada:*uri*. Periksa apakah jalur URI ada dan konfirmasikan bahwa peran dapat mengakses objek.
INSTANCE_RESERVATION_FAILED	Tidak ada kapasitas instance yang cukup untuk menyelesaikan alur kerja. Tunggu dan coba jalankan alur kerja lagi.
TIDAK VALID_ECR_IMAGE_URI	Struktur URI gambar Amazon ECR tidak valid. Berikan URI yang valid dan coba lagi.

Alasan kegagalan	Deskripsi kesalahan
INVALID_TASK_RESOURCE_VALUE	GPU, CPU, atau memori yang diminta terlalu tinggi untuk kapasitas komputasi yang tersedia, atau kurang dari nilai minimum 1 untuk tugas. *ID*
INPUT URI_TIDAK VALID	Struktur URI tidak valid*uri*. Periksa struktur URI dan coba lagi.
MODIFIED_INPUT_RESOURCE	URI yang disediakan *uri* telah dimodifikasi setelah proses dimulai. Coba lagi lari.
OUT_OF_MEMORY_ERROR	Tugas alur kerja *ID* kehabisan memori. Tingkatkan nilai memori dalam definisi alur kerja dan coba jalankan lagi.
RUN_TASK_FAILED	Jalankan gagal karena tugas gagal. Untuk men-debug kegagalan tugas, gunakan operasi GetRunTaskAPI dan aliran Amazon CloudWatch Logs.
RUN_TIMED_OUT	Jalankan batas waktu setelah *number* menit.
SERVICE_ERROR	Ada kesalahan sementara dalam layanan. Coba jalankan alur kerja lagi.
TUGAS_TIMED_OUT	Tugas *id* habis setelah *number* detik.
TIDAK DISUPPORTED_INPUT_SIZE	Ukuran input total terlalu tinggi. Kurangi ukuran input dan coba lagi.
WORKFLOW_RUN_FAILED	Alur kerja berjalan gagal. Tinjau aliran CloudWatch log mesin Log: *ID* untuk men-debug kegagalan.
WORKFLOW_VER_VALIDATION_FAILED	HealthOmics tidak mendukung versi Nextflow yang diminta: *version* --. Versi terbaru yang didukung adalah*version*. Ubah versi Nextflow Anda ke versi yang didukung dan coba lagi.
TIDAK DISUPPORTED_GPU_INSTANCE_TYPE	Jenis instance yang diminta tidak didukung di*Region*. Coba lagi jalankan dengan jenis instans GPU yang didukung di Wilayah ini. Jenis instance yang tersedia adalah*GPU instance types*.

Panduan untuk proses yang tidak responsif

Saat mengembangkan alur kerja baru, proses atau tugas tertentu bisa menjadi “macet” atau “hang” jika ada masalah dengan kode Anda, dan tugas gagal keluar dari proses dengan benar. Ini bisa menjadi tantangan untuk memecahkan masalah dan catch, karena itu normal untuk tugas berjalan untuk waktu yang lama. Untuk mencegah dan mengidentifikasi proses yang tidak responsif, ikuti praktik terbaik yang disarankan di bagian berikut.

Praktik terbaik untuk mencegah proses yang tidak responsif

- Pastikan Anda menutup semua file yang dibuka dalam kode tugas Anda. Membuka terlalu banyak file kadang-kadang dapat menyebabkan masalah threading dalam mesin alur kerja.
- Proses latar belakang yang dibuat oleh tugas alur kerja harus keluar saat tugas keluar. Namun, jika proses latar belakang tidak keluar dengan bersih, Anda harus secara eksplisit mematikan proses itu dalam kode tugas Anda.
- Pastikan proses Anda tidak berputar tanpa keluar. Hal ini dapat menyebabkan proses yang tidak responsif, dan memerlukan perubahan pada kode definisi alur kerja Anda untuk menyelesaikannya.
- Berikan memori dan alokasi CPU yang sesuai untuk tugas Anda. Analisis CloudWatch log atau gunakan alur kerja yang Jalankan Analyzer berhasil diselesaikan untuk memverifikasi bahwa Anda memiliki alokasi komputasi yang optimal. Gunakan `headroom` parameter Run Analyzer untuk menyertakan headroom tambahan, memastikan proses memiliki sumber daya yang cukup untuk diselesaikan. Sertakan setidaknya 5% headroom dalam memori dan CPU yang dialokasikan, untuk memperhitungkan proses sistem operasi latar belakang.
  - Selain itu, tingkatkan ukuran bandwidth instance jika instance membutuhkan throughput yang lebih tinggi. EC2 Instans Amazon dengan kurang dari 16 v CPUs (ukuran 4xl dan lebih kecil) dapat mengalami ledakan throughput. Untuk informasi selengkapnya tentang throughput EC2 instans Amazon, lihat Bandwidth instans EC2 yang tersedia Amazon.
- Pastikan Anda menggunakan ukuran sistem file yang benar untuk menjalankan Anda. Untuk proses tidak responsif yang menggunakan penyimpanan run statis, pertimbangkan untuk meningkatkan alokasi penyimpanan run statis untuk mengaktifkan throughput IO dan kapasitas penyimpanan yang lebih tinggi pada sistem file. Analisis manifes run untuk melihat penyimpanan sistem file maksimum, gunakan Run Analyzer untuk menentukan apakah alokasi sistem file perlu ditingkatkan.

Praktik terbaik untuk menangkap proses yang tidak responsif

- Saat mengembangkan alur kerja baru, gunakan grup run dengan batas waktu run maks yang ditetapkan untuk menangkap kode runaway. Misalnya, jika lari membutuhkan waktu 1 jam untuk menyelesaikannya, letakkan di grup lari yang habis waktu setelah 2 atau 3 jam (atau periode waktu yang berbeda berdasarkan kasus penggunaan Anda) untuk menangkap pekerjaan run-away. Juga, terapkan buffer untuk memperhitungkan varians dalam waktu pemrosesan.
- Siapkan serangkaian grup lari dengan batas waktu proses maksimum yang berbeda. Misalnya, Anda dapat menetapkan jangka pendek ke grup run yang menghentikan proses setelah beberapa jam, dan grup jangka panjang yang mengakhiri proses setelah beberapa hari, berdasarkan durasi alur kerja yang diharapkan.
- HealthOmics memiliki batas layanan durasi lari maksimum default 604.800 Detik, atau 7 hari, yang dapat disesuaikan melalui permintaan di alat kuota. Hanya minta peningkatan batas layanan kuota ini jika Anda telah menjalankan pendekatan tersebut dalam durasi seminggu. Jika Anda memiliki campuran jangka pendek dan panjang dan tidak menggunakan grup lari, pertimbangkan untuk menempatkan run jangka panjang di akun terpisah dengan batas layanan durasi lari maksimum yang lebih tinggi.
- Periksa CloudWatch log untuk tugas yang Anda curigai mungkin tidak responsif. Jika tugas biasanya mengeluarkan pernyataan log reguler dan belum melakukannya untuk waktu yang lama, tugas tersebut kemungkinan macet atau beku.

Apa yang harus dilakukan jika Anda mengalami proses yang tidak responsif

- Batalkan proses untuk menghindari biaya tambahan.
- Periksa log tugas untuk memeriksa apakah ada proses yang gagal keluar dengan benar.
- Periksa log mesin untuk mengidentifikasi perilaku mesin yang tidak normal.
- Bandingkan log tugas dan mesin dari proses yang tidak responsif dengan proses yang identik dan berhasil diselesaikan. Ini dapat membantu mengidentifikasi perbedaan yang mungkin menyebabkan perilaku tidak responsif.
- Jika Anda tidak dapat menentukan akar penyebabnya, angkat kasus dukungan dan sertakan yang berikut ini:
  - ARN dari stuck run dan ARN dari run identik yang berhasil diselesaikan.
  - Log mesin (tersedia setelah proses dibatalkan atau gagal)
  - Log tugas untuk tugas yang tidak responsif. Kami tidak memerlukan log tugas untuk semua tugas dalam alur kerja untuk memecahkan masalah.

Siklus hidup tugas dalam proses HealthOmics

Tugas adalah proses tunggal dalam menjalankan. HealthOmics memetakan setiap tugas dalam alur kerja Anda ke jenis instans komputasi omics yang paling sesuai dengan sumber daya tugas yang diperlukan. Anda menentukan sumber daya yang diperlukan dalam definisi alur kerja. Untuk informasi lebih lanjut, LihatPersyaratan komputasi dan memori untuk tugas HealthOmics.

HealthOmics menyediakan penyimpanan berjalan sementara untuk tugas yang akan digunakan. HealthOmics menyalin file input tugas ke penyimpanan run sementara sebagai file hanya-baca. HealthOmics menyediakan tautan simbolis sehingga tugas dapat mengakses file input dari direktori kerja. Tugas hanya memiliki akses ke file yang Anda deklarasikan dalam file definisi alur kerja.

Nilai status tugas

Anda dapat melacak kemajuan tugas dengan memantau status tugas. Saat Anda memulai proses, HealthOmics tetapkan status tugas Pending untuk setiap tugas yang sedang dijalankan. Saat tugas dimulai dan berlanjut melalui siklus hidupnya, HealthOmics perbarui nilai status untuk mencerminkan kemajuannya saat ini.

Anda dapat mengambil status tugas menggunakan salah satu metode berikut:

- HealthOmics Konsol menampilkan status setiap tugas dalam proses pada Run details halaman.
- Operasi GetRunTask API mengembalikan status tugas.
- Anda dapat memantau status tugas menggunakan EventBridge acara. Untuk informasi selengkapnya, lihat Menggunakan EventBridge dengan AWS HealthOmics.

Anda dapat mengambil status tugas saat ini menggunakan operasi GetRunTask API. HealthOmics Konsol menampilkan status untuk setiap tugas dalam proses pada Run details halaman.

HealthOmics mendukung nilai status tugas berikut:

Tertunda

Tugas Anda dalam antrian, menunggu untuk memulai. Tugas tetap tertunda untuk waktu yang singkat sebelum mereka mulai.

- Tugas tetap tertunda setelah akun Anda mencapai jumlah maksimum tugas bersamaan.
- Tugas tetap tertunda jika run adalah bagian dari grup run yang telah mencapai nilai maksimum sumber dayanya.

- Anda dapat menyesuaikan prioritas lari sehingga proses antrian tertentu dan tugasnya dimulai sebelum proses antrian lainnya. Untuk informasi selengkapnya tentang menjalankan prioritas, lihat Jalankan prioritas

Starting

HealthOmics membuat tugas dan menyediakan sumber daya yang diperlukan untuk tugas tersebut, seperti simpul tugas alur kerja.

Berjalan

Status tugas berjalan saat HealthOmics sedang memproses tugas.

Stopping

Setelah menyelesaikan pemrosesan tugas dan mengekspor data keluaran, tugas beralih ke Berhenti.

- HealthOmics deprovision node tugas alur kerja.

Selesai

HealthOmics telah selesai memproses tugas dan telah mentransfer data output ke sistem file penyimpanan run.

Failed

HealthOmics mengalami kesalahan saat memproses tugas dan tidak menyelesaikannya.

- Tugas beralih ke status Berhenti (HealthOmics membatalkan ketentuan sumber daya) dan kemudian ke status Gagal.
- Jika kesalahan adalah kesalahan layanan (kode status HTTP 5XX), dan alur kerja mendukung percobaan ulang untuk tugas ini, HealthOmics mencoba untuk memproses tugas lagi. HealthOmics menetapkan ID tugas baru untuk mencoba lagi.

Dibatalkan

HealthOmics menghentikan tugas setelah permintaan yang dimulai pengguna untuk membatalkan proses.

- Tugas beralih ke status Berhenti (HealthOmics membatalkan ketentuan sumber daya) dan kemudian ke status Dibatalkan.

Memecahkan masalah tugas alur kerja

Berikut ini adalah praktik dan pertimbangan terbaik untuk memecahkan masalah tugas Anda.

- Log tugas bergantung pada STDOUT dan STDERR diproduksi oleh tugas. Jika aplikasi yang digunakan dalam tugas tidak menghasilkan salah satu dari ini, maka tidak akan ada log tugas. Untuk membantu debugging, gunakan aplikasi dalam `verbose` mode.
- Untuk melihat perintah yang dijalankan dalam tugas bersama dengan nilai interpolasinya, gunakan perintah `set -x` Bash. Ini dapat membantu menentukan apakah tugas menggunakan input yang benar dan mengidentifikasi di mana kesalahan mungkin membuat tugas tidak berjalan sebagaimana dimaksud.
- Gunakan `echo` perintah untuk menampilkan nilai-nilai variabel ke STDOUT atauSTDERR. Ini membantu Anda mengonfirmasi bahwa mereka sedang diatur seperti yang diharapkan.
- Gunakan perintah seperti `ls -l <name_of_input_file>` untuk mengonfirmasi bahwa input ada dan memiliki ukuran yang diharapkan. Jika tidak, ini mungkin mengungkapkan masalah dengan tugas sebelumnya yang menghasilkan output kosong karena bug.
- Gunakan perintah `df -Ph . | awk 'NR==2 {print $4}'` dalam skrip tugas untuk menentukan ruang yang saat ini tersedia untuk tugas dan membantu mengidentifikasi situasi di mana Anda mungkin perlu menjalankan alur kerja dengan alokasi penyimpanan tambahan.

Menyertakan salah satu perintah sebelumnya dalam skrip tugas mengasumsikan bahwa wadah tugas juga menyertakan perintah ini dan bahwa mereka berada di lingkungan `path` wadah.

Jalankan pengoptimalan untuk HealthOmics alur kerja pribadi

Anda dapat mengoptimalkan proses untuk total biaya, total waktu berjalan, atau kombinasi keduanya. HealthOmics menyediakan data dan alat untuk membantu Anda menjalankan keputusan pengoptimalan. Run optimization tidak berlaku untuk alur kerja Ready2Run, karena Anda tidak memiliki kontrol apa pun atas cara layanan mengelola penyediaan sumber daya untuk alur kerja ini.

Langkah pertama adalah memahami penggunaan sumber daya tugas saat ini dan biaya untuk tugas-tugas yang sedang dijalankan, dan kemudian menerapkan metode untuk mengoptimalkan biaya operasional dan kinerja.

Topik

- Metode untuk mengoptimalkan konkurensi sumber daya

Jalankan Analyzer

HealthOmics menyediakan alat open source bernama Run Analyzer. Alat ini mengekstrak informasi penggunaan sumber daya tingkat tugas untuk dijalankan dan menyarankan peluang pengoptimalan untuk kinerja biaya dan lari.

> **Note**
>
> Run analyzer memperkirakan biaya tugas dan potensi penghematan biaya berdasarkan AWS daftar harga pada saat Anda menjalankan alat. Nilai rekomendasi pengoptimalan dan terapkan rekomendasi yang masuk akal untuk kasus penggunaan Anda. Uji pengoptimalan yang Anda adopsi untuk memastikan bahwa mereka bekerja untuk lari Anda.

Jalankan Analyzer melakukan tugas-tugas berikut:

- Mengevaluasi memori dan menghitung kemacetan.
- Mengidentifikasi tugas yang disediakan secara berlebihan untuk memori atau CPU, dan merekomendasikan ukuran instans baru yang dapat mengurangi biaya.
- Menghitung perkiraan biaya untuk tugas individu dan menghitung potensi penghematan biaya jika Anda menerapkan rekomendasi.
- Memberi Anda tampilan timeline tugas sehingga Anda dapat memverifikasi dependensi tugas dan urutan pemrosesan. Garis waktu juga membantu Anda mengidentifikasi tugas yang berjalan lama.
- Memberikan rekomendasi tentang ukuran sistem file untuk penyimpanan run.
- Menunjukkan waktu penyediaan tugas sehingga Anda dapat mengidentifikasi area di mana beban kontainer besar mungkin memperlambat waktu penyediaan.
- Alat ini mencakup parameter input (headroom) yang dapat Anda gunakan untuk mengontrol agresivitas rekomendasi pengoptimalan.

Bagian berikut mencakup saran khusus untuk menggunakan Run Analyzer untuk mengoptimalkan proses.

Tentukan biaya operasional

Anda dapat menggunakan metode dan pedoman berikut untuk menentukan biaya operasional:

- Untuk melihat total biaya operasional untuk periode penagihan, ikuti langkah-langkah berikut:

  1. Buka konsol Billing and Cost Management dan pilih Bills.
  2. Dalam Biaya berdasarkan layanan, perluas Omics.
  3. Perluas wilayah, lalu lihat biaya semua proses Anda yang dirinci berdasarkan jenis instance omics, menjalankan tipe penyimpanan, dan alur kerja Ready2Run.

- Untuk menghasilkan laporan biaya yang mencakup informasi untuk setiap proses, ikuti langkah-langkah berikut:

  1. Buka konsol Billing and Cost Management dan pilih Data Exports.
  2. Pilih Buat untuk membuat ekspor data baru.
  3. Masukkan nama Ekspor untuk ekspor data. Simpan bidang lain pada nilai defaultnya untuk membuat laporan CUR (biaya dan penggunaan).
  4. Untuk perincian Waktu, pilih per jam atau harian.
  5. Di bawah Pengaturan penyimpanan ekspor data, lakukan langkah-langkah konfigurasi berikut:

     a. Konfigurasikan bucket Amazon S3 untuk ekspor data.
     b. Untuk pembuatan versi File, pilih apakah akan menimpa file ekspor yang ada atau membuat file baru setiap kali.

     Sistem menghasilkan laporan pertama dalam 24 jam ke depan dan menghasilkan laporan berikutnya sekali sehari.
  6. Untuk informasi selengkapnya tentang cara membuat ekspor data, lihat Membuat ekspor data di Panduan Pengguna Ekspor AWS Data.

- Anda dapat menandai proses Anda untuk memantau dan mengoptimalkan biaya berdasarkan kategori, seperti menurut tim atau proyek. Jika Anda menggunakan tag, ikuti langkah-langkah berikut untuk melihat biaya yang dijalankan berdasarkan kategori tag:

  1. Buka konsol Billing and Cost Management dan pilih Cost Explorer.
  2. Di Laporkan parameter > Kelompokkan menurut, pilih Tag sebagai dimensi. dan pilih nama Tag yang diinginkan.

- Untuk melihat penggunaan sumber daya untuk tugas, lihat log manifes jalankan CloudWatch. Untuk informasi selengkapnya, lihat Pemantauan HealthOmics dengan CloudWatch Log.
- Gunakan Jalankan Analyzer alat untuk mengekstrak informasi penggunaan sumber daya tugas untuk dijalankan.

Tentukan penggunaan waktu berjalan

Anda dapat menggunakan metode berikut untuk membantu Anda menyelidiki penggunaan waktu berjalan:

- Dari halaman Runs konsol, Anda dapat melihat total waktu berjalan untuk menjalankan.
- Dari halaman Run details, Anda dapat melihat item berikut:
  - Lihat total waktu lari untuk lari.
  - Lihat waktu berjalan untuk setiap tugas dalam proses.
  - Pilih salah satu tautan untuk melihat log di Amazon S3, atau untuk melihat log run atau menjalankan log manifes. CloudWatch
- Dari daftar Jalankan tugas, pilih tautan Lihat log untuk tugas untuk melihat log tugas CloudWatch.
- Respons terhadap operasi `listRuns` API mencakup waktu mulai berjalan dan waktu berhenti, sehingga Anda dapat menghitung total waktu berjalan.
- Jalankan AnalyzerAlat ini menunjukkan durasi tugas pada tampilan timeline. Alat ini memberikan representasi visual dari urutan pemrosesan tugas, yang dapat Anda cocokkan dengan urutan yang diharapkan.

Metode untuk mengoptimalkan proses

HealthOmics secara otomatis menyediakan, mengelola, dan mengoptimalkan sumber daya yang melakukan pementasan data (seperti impor data dan ekspor data). HealthOmics juga memulai dan menjalankan mesin alur kerja untuk alur kerja Anda. Namun, Anda dapat memengaruhi waktu mulai berjalan, waktu mulai tugas, dan waktu menjalankan tugas secara keseluruhan dengan menyetel berbagai konfigurasi run. Pendekatan keseluruhan Anda terhadap definisi dan desain alur kerja juga memengaruhi waktu menjalankan tugas. Daftar berikut menjelaskan faktor-faktor yang dapat mempengaruhi kinerja run dan task:

Jalankan jenis penyimpanan

> Jenis penyimpanan run berdampak pada kinerja run dan menjalankan waktu penyediaan. Dynamic run storage menyediakan lebih cepat dan tidak pernah kehabisan memori, karena diskalakan secara dinamis dengan kebutuhan penyimpanan run Anda. Dynamic run storage juga cocok untuk alur kerja dalam pengembangan, di mana Anda mungkin sering memulai dan menghentikan alur kerja untuk memecahkan masalah.

Penyimpanan berjalan statis membutuhkan waktu penyediaan sistem file yang lebih lama, tetapi dapat menyelesaikan beberapa proses lebih cepat, biasanya jika proses memiliki konkurensi tugas yang tinggi atau membutuhkan kapasitas sistem file yang lebih besar dari 9,6 TiB. Penyimpanan berjalan statis sangat cocok untuk alur kerja yang berjalan lama dengan I/O persyaratan tinggi.

Untuk membantu Anda mengevaluasi biaya vs. kinerja setiap jenis penyimpanan run untuk proses tertentu, Anda dapat mencoba pengujian A/B untuk melihat jenis penyimpanan run mana yang memberikan kinerja yang lebih baik. Juga, pertimbangkan untuk menggunakan penyimpanan run dinamis untuk siklus pengembangan Anda, lalu gunakan penyimpanan run statis untuk produksi berjalan dalam skala besar.

Untuk informasi selengkapnya tentang menjalankan jenis penyimpanan Jalankan jenis penyimpanan dalam HealthOmics alur kerja

Penyediaan berlebih menjalankan penyimpanan statis

Jika perhitungan tugas alur kerja Anda dibatasi oleh 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 kemacetan, meningkatkan ukuran sistem file untuk mengurangi waktu menjalankan tugas dapat mengurangi biaya keseluruhan.

Kurangi ukuran gambar kontainer

Saat setiap tugas dimulai, HealthOmics memuat wadah yang Anda tentukan untuk tugas tersebut. Kontainer yang lebih besar membutuhkan waktu lebih lama untuk dimuat. Optimalkan wadah Anda sekecil mungkin untuk meningkatkan efisiensi peluncuran tugas baru. Jika Anda menambahkan kumpulan data besar ke kontainer, pertimbangkan untuk menyimpan kumpulan data di S3 dan meminta alur kerja Anda mengimpor data dari S3. Untuk ukuran wadah maksimum yang HealthOmics mendukung, lihatHealthOmics alur kerja kuota ukuran tetap.

Ukuran tugas

Anda dapat menggabungkan tugas kecil dan berurutan menjadi satu tugas untuk menghemat waktu penyediaan tugas. Selain itu, HealthOmics memiliki biaya durasi tugas minimum satu menit, jadi menggabungkan tugas dapat mengurangi biaya. Dalam tugas gabungan, Anda mungkin dapat menggunakan pipa Unix untuk menghindari I/O biaya serialisasi dan deserialisasi file.

Kompresi file

Hindari terlalu mengompresi file perantara alur kerja. Sebagian besar format genomik menggunakan kompresi “gzip” atau “block gzip”. Dekompresi file input tugas dan kompresi

ulang file output tugas dapat menghabiskan sebagian besar dari keseluruhan penggunaan CPU tugas. Beberapa aplikasi genomik memungkinkan Anda untuk mengatur tingkat kompresi saat membuat serial output. Dengan mengurangi tingkat kompresi, Anda dapat mengurangi waktu CPU, meskipun file yang lebih besar meningkatkan waktu yang dihabiskan untuk menulis ke disk. Bergantung pada tugas dan aplikasinya, Anda dapat menemukan tingkat kompresi optimal untuk file perantara yang menghasilkan waktu berjalan terpendek. Kami menyarankan Anda memulai dengan menargetkan tugas dengan file output terbesar. Tingkat kompresi 2 bekerja dengan baik untuk beberapa skenario. Anda dapat mulai dengan level ini untuk kasus penggunaan Anda, dan membandingkan hasilnya dengan mencoba tingkat kompresi lainnya.

Jumlah utas

Jika Anda menentukan utas dalam definisi tugas Anda, atur jumlah utas ke nilai yang sama dengan jumlah v yang dimintaCPUs.

Tentukan komputasi dan memori

Jika Anda tidak menentukan memori atau sumber daya komputasi dalam tugas Anda, HealthOmics tetapkan instance type (`omics.c.large`) terkecil sebagai default. Deklarasikan persyaratan memori dan komputasi Anda secara eksplisit jika Anda ingin menetapkan jenis instans yang HealthOmics lebih besar.

HealthOmics mengalokasikan jumlah vCPUs, memori, dan sumber daya GPU yang Anda minta. Misalnya, jika Anda meminta 15v CPUs dan 33GiB, HealthOmics mengalokasikan instance omics.m.4xl (16v, 64GB) untuk tugas AndaCPUs, tetapi tugas Anda hanya dapat menggunakan 15 v dan 33GiB. CPUs Oleh karena itu, kami menyarankan Anda meminta v CPUs dan sumber daya memori yang cocok dengan instance omics.

Batch beberapa sampel menjadi satu run

Karena penyediaan sistem file membutuhkan waktu pada awal proses, Anda dapat menghemat waktu penyediaan dengan mengelompokkan beberapa sampel ke dalam proses yang sama. Pertimbangkan faktor-faktor berikut sebelum memutuskan pendekatan ini:

- Satu sampel yang buruk dapat menyebabkan alur kerja gagal, sehingga pengelompokan sampel dapat meningkatkan jumlah alur kerja yang gagal. Jika Anda tidak yakin bahwa alur kerja Anda akan berhasil sebagian besar waktu, satu run per sampel bisa menjadi pendekatan yang lebih baik.
- HealthOmics mengalokasikan satu sistem file penyimpanan run untuk seluruh alur kerja. Untuk sekumpulan sampel, pastikan untuk menentukan jumlah penyimpanan run yang cukup besar untuk memproses semua sampel.

- Ada jumlah maksimum penyimpanan run per alur kerja, sehingga dapat membatasi jumlah sampel yang dapat Anda tambahkan ke batch.
- Ukuran penyimpanan run minimum adalah 1,2 TiB, jadi batching dapat mengurangi biaya jika alur kerja menggunakan penyimpanan yang jauh lebih sedikit daripada minimum untuk setiap sampel.
- Jalankan penyimpanan dapat menangani beberapa koneksi simultan, jadi memiliki banyak tugas menggunakan penyimpanan run yang sama seharusnya tidak menyebabkan I/O kemacetan.
- Setiap run memiliki set tag sendiri. Jika Anda menandai alur kerja dengan informasi untuk penganggaran atau pelacakan, mungkin lebih baik menggunakan proses terpisah.
- Peran IAM berlaku untuk seluruh proses. Setiap pengguna memiliki akses ke semua data untuk batch sampel. Memiliki alur kerja yang terpisah memberi Anda kemampuan untuk menggunakan izin yang lebih halus.
- HealthOmics menetapkan kuota tingkat akun untuk jumlah maksimum alur kerja bersamaan dan jumlah maksimum tugas bersamaan dalam alur kerja. Untuk informasi tentang cara meminta kenaikan kuota ini, lihatHealthOmics kuota layanan.

Gunakan parameter untuk gambar kontainer

Parameterisasi gambar kontainer Anda daripada menyematkannya URIs dalam alur kerja. Jika mereka menjalankan parameter, HealthOmics memvalidasi bahwa run memiliki akses ke kontainer Anda sebelum proses dimulai. Jika tidak, tugas gagal selama menjalankan, ketika Anda telah mengeluarkan biaya untuk tugas yang diselesaikan. Juga, karena ini adalah input berparameter, HealthOmics menghasilkan checksum dalam manifes run, yang meningkatkan asal run.

Gunakan linter

Gunakan linter untuk menemukan kesalahan alur kerja umum sebelum Anda menjalankan alur kerja baru. Untuk informasi selengkapnya, lihat Linter alur kerja di HealthOmics.

Gunakan EventBridge untuk menandai masalah

Gunakan peringatan EventBridge khusus untuk menangkap anomali yang spesifik untuk logika bisnis Anda.

Gunakan toko urutan

Pertimbangkan untuk menggunakan penyimpanan urutan untuk data sumber Anda untuk menghemat biaya penyimpanan. Untuk informasi selengkapnya, lihat data Store omics hemat biaya pada skala apa pun dengan HealthOmics posting blog.

Dampak varians ukuran file antar proses

Pengguna sering merancang dan menguji berjalan menggunakan satu set kecil data pengujian, kemudian menemukan berbagai macam data dengan varians ukuran file yang signifikan dalam proses produksi. Pastikan Anda memperhitungkan varians ini saat Anda mengoptimalkan proses.

Daftar berikut menjelaskan rekomendasi untuk pengoptimalan di mana ada varians yang signifikan dalam ukuran file:

Variasikan ukuran file dalam data pengujian Anda

Cobalah untuk menggunakan data pengujian selama pengembangan yang memiliki jumlah varians yang representatif.

Gunakan Run Analyzer

Gunakan alat Run Analyzer di berbagai sampel untuk memperhitungkan varians dalam ukuran data.

Anda dapat menggunakan run analyzer untuk memahami varians antara proses dalam sampel data produksi Anda. Gunakan `--batch` mode di Run Analyzer untuk menghasilkan statistik untuk sejumlah proses dan menganalisis sumber daya komputasi maksimum yang diperlukan untuk menangani outlier dalam kumpulan data Anda.

Misalnya, Anda dapat memberikan run analyzer sel aliran penuh data dalam mode batch untuk memahami vCPU puncak dan pemanfaatan memori untuk sel aliran penuh.

Kurangi varians ukuran kumpulan data masukan

Jika Anda melihat varians tinggi dalam ukuran sampel, Anda dapat membagi dua sampel di hulu HealthOmics dan memilih ukuran sistem file yang berbeda untuk setiap batch untuk menghemat biaya penyimpanan yang dijalankan.

Di WDL, gunakan `size` fungsi untuk membagi alokasi sumber daya untuk tugas individu untuk sampel besar versus kecil. Terapkan strategi ini ke tugas Anda yang paling mahal untuk memiliki dampak paling besar.

Di Nextflow, gunakan sumber daya bersyarat untuk mengatur alokasi sumber daya berdasarkan ukuran file atau nama file. Untuk informasi selengkapnya, lihat Sumber daya proses bersyarat di situs Nextflow GitHub .

Jangan mengoptimalkan terlalu cepat

Selesaikan kode dan logika alur kerja Anda sebelum berinvestasi dalam upaya penyetelan kinerja yang signifikan. Mengubah kode Anda dapat berdampak signifikan pada sumber daya yang diperlukan. Jika Anda mengoptimalkan proses terlalu cepat dalam proses pengembangan, Anda mungkin terlalu mengoptimalkan atau Anda mungkin perlu mengoptimalkan lagi jika definisi alur kerja berubah nanti.

Jalankan kembali alat Run Analyzer secara berkala

Jika Anda membuat perubahan pada definisi alur kerja dari waktu ke waktu atau jika varians sampel Anda berubah, jalankan alat Run Analyzer secara berkala untuk membantu Anda membuat pengoptimalan tambahan.

Metode untuk mengoptimalkan konkurensi sumber daya

HealthOmics menyediakan kemampuan berikut untuk membantu Anda mengontrol dan mengelola biaya saat pemrosesan berjalan dalam skala besar:

- Gunakan grup run untuk mengontrol biaya dan penggunaan sumber daya Anda. Anda dapat menetapkan nilai maksimum dalam grup run untuk jumlah run bersamaan, v CPUs GPUs, dan total waktu berjalan per tugas. Jika tim atau grup terpisah menggunakan akun yang sama, Anda dapat membuat grup lari terpisah untuk setiap tim. Anda dapat mengontrol penggunaan sumber daya dan biaya per tim dan dengan mengonfigurasi nilai maksimum grup run. Untuk informasi selengkapnya, lihat Menggunakan grup HealthOmics lari.
- Selama pengembangan, Anda dapat mengonfigurasi grup run terpisah dengan nilai maksimum yang lebih rendah untuk menangkap tugas runaway.
- Service Quotas juga membantu melindungi akun Anda dari permintaan sumber daya yang berlebihan. Untuk informasi tentang Service Quotas, termasuk cara meminta kenaikan nilai kuota, lihat HealthOmics kuota layanan

Jalankan operasi di HealthOmics

Anda dapat memulai, menjalankan ulang, mengkloning, membatalkan, atau menghapus proses:

- Start— HealthOmics membuat proses baru menggunakan pengaturan konfigurasi yang Anda tentukan dan kemudian mulai menjalankan.

- Rerun— HealthOmics membuat run baru yang merupakan duplikat dari run yang Anda tentukan. Anda dapat menjalankan kembali proses yang dihapus menggunakan HealthOmics rerun alat ini.
- Clone— Anda dapat mengkloning proses yang ada menggunakan konsol. Konsol membuka halaman Clone run dan mengisi ulang bidang konfigurasi menggunakan nilai dari proses yang ada. Anda dapat memodifikasi nilai sesuai kebutuhan dan memulai proses kloning.
- Cancel— Anda dapat membatalkan proses yang belum selesai. Saat Anda membatalkan proses, HealthOmics tidak menyimpan output run apa pun.
- Delete— Anda dapat menghapus proses yang telah selesai secara manual, atau mengatur mode retensi jalankan HealthOmics untuk menghapus proses tertua secara otomatis. Untuk informasi selengkapnya tentang mode retensi, lihatthe section called “Jalankan mode retensi”.

Topik

Mulai lari HealthOmics

Saat memulai proses, Anda menentukan sumber daya yang HealthOmics dialokasikan untuk digunakan selama proses dijalankan.

Tentukan jenis penyimpanan run dan jumlah penyimpanan (untuk penyimpanan statis). Untuk memastikan isolasi dan keamanan data HealthOmics , berikan penyimpanan pada awal setiap proses, dan hentikan penyediaannya di akhir proses. Untuk informasi tambahan, lihat Jalankan jenis penyimpanan dalam HealthOmics alur kerja.

Tentukan lokasi Amazon S3 untuk file output. Jika Anda menjalankan alur kerja volume tinggi secara bersamaan, gunakan URIs output Amazon S3 terpisah untuk setiap alur kerja untuk menghindari pembatasan bucket. Untuk informasi selengkapnya, lihat Mengatur objek menggunakan awalan di Panduan Pengguna Amazon S3 dan Skalakan Koneksi Penyimpanan Secara Horizontal di whitepaper Mengoptimalkan Kinerja Amazon S3.

Anda juga dapat menentukan prioritas lari. Bagaimana prioritas memengaruhi proses tergantung pada apakah run dikaitkan dengan grup run. Untuk informasi tambahan, lihat Jalankan prioritas.

Jika alur kerja memiliki satu atau beberapa versi, Anda dapat menentukan versi saat memulai proses. Jika Anda tidak menentukan versi, HealthOmics mulai versi alur kerja default.

Saat menggunakan HealthOmics API, Anda dapat memberikan ID permintaan unik untuk setiap proses. ID permintaan adalah token idempotensi yang HealthOmics digunakan untuk mengidentifikasi permintaan duplikat. dan memulai proses hanya sekali.

> **Note**
>
> Anda menentukan peran layanan IAM saat memulai proses. Secara opsional, konsol dapat membuat peran layanan untuk Anda. Untuk informasi selengkapnya, lihat Peran layanan untuk AWS HealthOmics.

Topik

HealthOmics menjalankan parameter

Saat Anda memulai proses, Anda menentukan input run di file JSON parameter run atau Anda dapat memasukkan nilai parameter sebaris. Untuk informasi tentang mengelola ukuran file JSON parameter run, lihatMengelola ukuran parameter run.

HealthOmics mendukung jenis JSON berikut untuk nilai parameter.

Jenis JSON	Contoh kunci dan nilai	Catatan
boolean	“b” :benar	Nilai tidak dalam tanda kutip, dan semua huruf kecil.
integer	“Aku” :7	Nilai tidak dalam tanda kutip.
number	“f” :42.3	Nilai tidak dalam tanda kutip.

Jenis JSON	Contoh kunci dan nilai	Catatan
string	“s” :"karakter”	Nilai ada dalam tanda kutip. Gunakan tipe string untuk nilai teks dan URIs. Target URI harus jenis input yang diharapkan.
array	“a”: [1,2,3]	Nilai tidak dalam tanda kutip. Anggota array masing-ma sing harus memiliki tipe yang ditentukan oleh parameter input.
object	“o”: {"left” :"a”, “right” :1}	Di WDL, objek peta ke WDL Pair, Map, atau Struct

Memulai lari menggunakan konsol

Untuk memulai lari

1. Buka konsol HealthOmics .
2. Jika diperlukan, buka panel navigasi kiri (≡). Pilih Runs.
3. Pada halaman Runs, pilih Start run.
4. Di panel Run details, berikan informasi berikut

   - Sumber alur kerja - Pilih alur kerja yang dimiliki atau alur kerja bersama.
   - ID alur kerja - ID alur kerja yang terkait dengan proses ini.
   - Versi alur kerja (Opsional) - Pilih versi alur kerja yang akan digunakan untuk menjalankan ini. Jika Anda tidak memilih versi, proses akan menggunakan versi default alur kerja.
   - Jalankan nama - Nama khas untuk lari ini.
   - Jalankan prioritas (Opsional) - Prioritas proses ini. Angka yang lebih tinggi menentukan prioritas yang lebih tinggi, dan tugas prioritas tertinggi dijalankan terlebih dahulu.
   - Jalankan tipe penyimpanan - Tentukan jenis penyimpanan di sini untuk mengganti jenis penyimpanan run default yang ditentukan untuk alur kerja. Penyimpanan statis

mengalokasikan jumlah penyimpanan yang tetap untuk dijalankan. Skala penyimpanan dinamis naik dan turun sesuai kebutuhan untuk setiap tugas yang dijalankan.

- Jalankan kapasitas penyimpanan - Untuk penyimpanan run statis, tentukan jumlah penyimpanan yang diperlukan untuk menjalankan. Entri ini mengganti jumlah penyimpanan run default yang ditentukan untuk alur kerja.
- Pilih tujuan output S3 - Lokasi S3 tempat output run akan disimpan.
- ID akun pemilik bucket keluaran (Opsional) - Jika akun Anda tidak memiliki bucket keluaran, masukkan Akun AWS ID pemilik bucket. Informasi ini diperlukan agar HealthOmics dapat memverifikasi kepemilikan bucket.
- Jalankan mode retensi metadata - Pilih apakah akan mempertahankan metadata untuk semua proses atau minta sistem menghapus metadata run tertua saat akun Anda mencapai jumlah maksimum proses. Untuk informasi selengkapnya, lihat Jalankan mode retensi untuk HealthOmics berjalan.

5. Di bawah Peran layanan, Anda dapat menggunakan peran layanan yang ada atau membuat yang baru.
6. (Opsional) Untuk Tag, Anda dapat menetapkan hingga 50 tag untuk dijalankan.
7. Pilih Berikutnya.
8. Pada halaman Tambah nilai parameter, berikan parameter run. Anda dapat mengunggah file JSON yang menentukan parameter atau memasukkan nilai secara manual.
9. Pilih Berikutnya.
10. Di panel Run group, Anda dapat secara opsional menentukan grup run untuk menjalankan ini. Untuk informasi selengkapnya, lihat Menggunakan grup HealthOmics lari.
11. Di panel Run cache, Anda dapat secara opsional menentukan cache run untuk proses ini. Untuk informasi selengkapnya, lihat Mengkonfigurasi run dengan run cache menggunakan konsol.
12. Pilih Tinjau dan mulai jalankan.
13. Setelah Anda meninjau konfigurasi run, pilih Mulai jalankan.

Memulai menjalankan menggunakan API

Gunakan operasi API start-run untuk membuat dan memulai proses.

Contoh berikut menentukan ID alur kerja dan peran layanan. Contoh ini menyetel mode retensi ke`REMOVE`. Untuk informasi selengkapnya tentang mode retensi, lihatJalankan mode retensi untuk HealthOmics berjalan.

```
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
```

Sebagai tanggapan, Anda mendapatkan output berikut. `uuid`Ini unik untuk dijalankan, dan bersama dengan `outputUri` dapat digunakan untuk melacak di mana data output ditulis.

```
{
    "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"
}
```

Sertakan file parameter

Jika templat parameter untuk alur kerja mendeklarasikan parameter yang diperlukan, Anda dapat memberikan file JSON lokal dari input saat memulai alur kerja. File JSON berisi nama yang tepat dari setiap parameter input dan nilai untuk parameter.

Referensikan file JSON masukan di AWS CLI dengan `--parameters file://<input_file.json>` menambahkan `start-run` permintaan Anda. Untuk informasi selengkapnya tentang parameter run, lihatHealthOmics jalankan masukan.

Berikan ID permintaan

Anda dapat memberikan yang unik `requestId` untuk setiap lari. ID permintaan adalah token idempotensi yang HealthOmics digunakan untuk menangkap permintaan duplikat. Itu tidak akan memulai proses jika ID permintaan adalah duplikat dari proses sebelumnya.

Jika Anda menggunakan infrastruktur (seperti fungsi Lambda atau fungsi langkah) untuk mengatur proses mulai, praktik terbaik adalah memberikan ID permintaan unik untuk setiap permintaan. StartRun Ini memastikan bahwa jika infrastruktur Anda secara tidak sengaja memulai proses yang sudah dimulai, tidak HealthOmics akan memulai proses duplikat. Misalnya, jika infrastruktur mencoba untuk memulihkan dari kesalahan upstream, itu mungkin menjalankan kembali skrip yang mencoba memulai proses yang merupakan permintaan duplikat.

Pilih versi alur kerja

Anda dapat menentukan versi alur kerja untuk dijalankan. Jika Anda tidak menentukan versi, HealthOmics mulai proses dengan versi alur kerja default.

```
aws omics start-run
     --workflow-id workflow id \
      ...
     --workflow-version-name '1.2.1'
```

Ganti jenis penyimpanan run

Anda dapat mengganti jenis penyimpanan run default yang disetel dalam alur kerja.

```
aws omics start-run
       --workflow-id workflow id \
        ...
       --storage-type STATIC
       --storage-capacity 2400
```

Jalankan alur kerja GPU

Anda juga dapat menentukan ID alur kerja GPU, seperti yang ditunjukkan pada contoh berikut:

```
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
```

Dapatkan informasi tentang lari

Anda dapat menggunakan ID dalam respons dengan API get-run untuk memeriksa status run, seperti yang ditunjukkan.

```
aws omics get-run --id run id
```

Respons dari operasi API ini memberi tahu Anda status alur kerja yang dijalankan. Status yang mungkin adalah`PENDING`,, `STARTING`RUNNING, dan`COMPLETED`. Saat dijalankan`COMPLETED`, Anda

dapat menemukan file keluaran yang dipanggil `outfile.txt` di bucket Amazon S3 keluaran Anda, di folder yang dinamai sesuai dengan ID run.

Operasi API get-run juga menampilkan detail lainnya, seperti apakah alur kerja itu `Ready2Run` atau, mesin alur kerja`PRIVATE`, dan detail akselerator. Contoh berikut menunjukkan respons untuk menjalankan alur kerja pribadi, dijelaskan dalam WDL dengan akselerator GPU dan tidak ada tag yang ditetapkan untuk menjalankan.

```
{
    "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": {}
}
```

Anda dapat melihat status semua proses dengan operasi API list-run, seperti yang ditunjukkan.

```
aws omics list-runs
```

Untuk melihat semua tugas diselesaikan untuk menjalankan tertentu, gunakan list-run-tasksAPI.

```
aws omics list-run-tasks --id task ID
```

Untuk mendapatkan detail tugas tertentu, gunakan get-run-task API.

```
aws omics get-run-task --id <run_id> --task-id task ID
```

Setelah proses selesai, metadata dikirim ke CloudWatch bawah aliran. **manifest/run/<run ID>/<run UUID>**

Berikut ini adalah contoh manifes.


```
{
    "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"
  },
  ...
```


Jalankan metadata tidak dihapus jika tidak ada di log. CloudWatch

Jalankan kembali lari HealthOmics

Untuk proses yang belum Anda hapus, gunakan konsol atau API untuk menjalankan kembali proses. Untuk menjalankan yang Anda hapus, gunakan alat ini. HealthOmics rerun

Topik

- Jalankan kembali proses menggunakan konsol
- Jalankan kembali proses menggunakan API
- Menggunakan alat Rerun

Jalankan kembali proses menggunakan konsol

Dari konsol, ikuti langkah-langkah ini untuk menjalankan kembali proses:

1. Buka konsol HealthOmics .
2. Jika diperlukan, buka panel navigasi kiri (≡). Pilih Runs.

3. Pada halaman Runs, pilih run untuk dijalankan kembali.
4. Dari menu tindakan di atas tabel, pilih Re-run.

Jalankan kembali proses menggunakan API

Gunakan operasi StartRun API untuk menjalankan kembali proses yang sudah ada. Berikan masukan yang diperlukan berikut:

- Peran layanan ARN ()`roleArn`.
- ID dari run ke duplikat (`runId`).
- Lokasi Amazon S3 tempat proses menyimpan output run (). `outputUri`

```
aws omics start-run
     --run-id run id \
     --role-arn arn:aws:iam::1234567892012:role/service-role/
OmicsWorkflow-20221004T164236 \
     --output-uri s3://workflow-output-b6f2fce1
```

Menggunakan alat Rerun

Untuk proses yang dihapus, Anda dapat mengunduh dan menggunakan HealthOmics rerun alat untuk menjalankan kembali proses. Alat ini mengambil informasi yang dijalankan dari manifes CloudWatch Log. Unduh rerun alat dari GitHub repositori HealthOmics Alat.

Contoh berikut menunjukkan cara menggunakan rerun alat ini.

```
aws-healthomics-rerun 9876543
```

Jika run ada di CloudWatch, Anda menerima respons yang mirip dengan contoh output berikut. Jika alur kerja tidak ada lagi, Anda menerima pesan galat.

```
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": {}
}
```

Kloning lari di HealthOmics

Anda dapat mengkloning proses yang ada menggunakan HealthOmics konsol. Kloning membuat proses baru menggunakan nilai konfigurasi run yang dikloning. Anda dapat memodifikasi nilai default ini dan menambahkan input opsional lainnya.

1. Buka konsol HealthOmics .
2. Jika diperlukan, buka panel navigasi kiri (≡). Pilih Runs.
3. Pada halaman Runs, pilih run to clone.
4. Dari menu tindakan di atas tabel, pilih Clone run. Konsol membuka formulir Clone run. Formulir identik dengan Start run, kecuali konsol mengisi formulir dengan semua nilai yang relevan dari proses kloning.

   Konsol membuat ID run baru untuk klon run, dan menambahkan ID run ini sebagai akhiran ke nama run.

Saat Anda melanjutkan melalui halaman formulir, Anda dapat menyesuaikan nilai konfigurasi sesuai kebutuhan.

5. Setelah Anda meninjau konfigurasi run, pilih Mulai jalankan.

Batalkan proses di HealthOmics

Anda dapat membatalkan proses jika statusnya adalah`PENDING`,`STARTING`,`RUNNING`, atau`STOPPING`.

> **Note**
>
> Saat Anda membatalkan proses, HealthOmics tidak menyimpan output run apa pun.

Dari konsol, ikuti langkah-langkah berikut untuk membatalkan proses:

1. Buka konsol HealthOmics .
2. Jika diperlukan, buka panel navigasi kiri (≡). Pilih Runs.
3. Pada halaman Runs, pilih run untuk membatalkan.
4. Konsol membuka halaman Run details. Dari spanduk status di bagian atas halaman, pilih Stop run.
5. Masukkan konfirmasi untuk menghentikan proses.

Untuk membatalkan proses menggunakan API, gunakan operasi CancelRun API.

Contoh berikut menunjukkan cara membatalkan run menggunakan file AWS CLI . Untuk menjalankan contoh, ganti *run id* dengan ID run yang ingin Anda batalkan. Jika berhasil, tidak ada respon.

```
aws omics cancel-run --id run id
```

Hapus run in HealthOmics

Ketika Anda tidak lagi membutuhkan run, Anda dapat menghapusnya menggunakan AWS CLI, API, atau konsol. Anda dapat menghapus run ketika statusnya adalah `COMPLETED` atau`CANCELED`.

Dari konsol, ikuti langkah-langkah berikut untuk menghapus proses:

1. Buka konsol HealthOmics .
2. Jika diperlukan, buka panel navigasi kiri (≡). Pilih Runs.
3. Pada halaman Runs, pilih satu atau beberapa run untuk dihapus.
4. Dari menu tindakan di atas tabel, pilih Hapus.
5. Dalam bentuk modal, ketik konfirmasi untuk mengonfirmasi penghapusan.

AWS CLI Perintah berikut menghapus run. Untuk menjalankan contoh, ganti *run id* dengan ID run yang ingin Anda hapus. Tidak ada respons jika proses berhasil dihapus.

```
aws omics delete-run --id run id
```

Menggunakan grup HealthOmics lari

Anda dapat membuat grup run secara opsional untuk membatasi sumber daya komputasi untuk proses yang Anda tambahkan ke grup. Jalankan grup dapat membantu Anda:

- Antrikan lari Anda sehingga Anda tidak melebihi batas layanan.
- Tangkap tugas run-away dengan menetapkan durasi lari maksimum.
- Kelola prioritas setiap lari sehingga proses yang paling penting selesai terlebih dahulu.

Jika Anda menyetel vCPU, GPU, atau run bersamaan maksimum, menjalankan tugas akan mengantri saat maksimum tercapai. Jika Anda menetapkan durasi lari maksimum, proses gagal jika melebihi durasi maksimum.

Gunakan pengaturan run priority untuk menetapkan prioritas dalam grup run.

Batas layanan lebih diutamakan daripada batas grup run. Misalnya, jika Anda menetapkan maksimum grup run ke nilai yang lebih tinggi daripada maksimum layanan Anda di suatu wilayah, HealthOmics menerapkan maksimum layanan.

Topik

- Jalankan prioritas
- Buat grup run menggunakan konsol
- Buat grup run menggunakan CLI

- Hapus grup run menggunakan konsol
- Hapus grup run menggunakan CLI

Jalankan prioritas

Anda dapat menggunakan run priority untuk menetapkan prioritas run dalam grup run.

Jika beberapa run memiliki prioritas yang sama, run yang dimulai terlebih dahulu memiliki prioritas yang lebih tinggi.

Anda juga dapat menetapkan prioritas untuk menjalankan yang tidak ada dalam grup run. Prioritas dibandingkan dengan prioritas semua proses lain yang tidak ada dalam grup lari

Anda menetapkan prioritas lari saat Anda memulai lari. Untuk informasi selengkapnya, lihat Mulai lari HealthOmics.

Buat grup run menggunakan konsol

Untuk membuat grup run

1. Buka konsol HealthOmics .
2. Jika diperlukan, buka panel navigasi kiri (≡). Pilih Jalankan grup.
3. Pada halaman Jalankan grup, pilih Buat grup lari.
4. Pada halaman Create run group details, berikan informasi berikut
   - Jalankan nama grup - Nama unik untuk grup run ini.
   - Max vCPU untuk menjalankan bersamaan - Jumlah maksimum v CPUs yang dapat berjalan secara bersamaan di semua proses aktif dalam grup run.
   - Max GPUs - Jumlah maksimum GPUs yang dapat berjalan secara bersamaan di semua run aktif dalam grup run.
   - Max run time (mins) per run - Waktu maksimum untuk setiap run (dalam menit). Jika run melebihi waktu berjalan maksimum, run gagal secara otomatis.
   - Max berjalan bersamaan - Jumlah maksimum run yang dapat berjalan pada saat yang sama.
5. (opsional) Anda dapat menambahkan hingga 50 tag ke grup run.

6. Pilih Buat grup run.

Buat grup run menggunakan CLI

Untuk membuat grup run, gunakan operasi create-run-groupAPI untuk membuat grup run bernama`TestRunGroup`. Contoh berikut menetapkan maksimum 20 CPUs, 10 GPUs, 5 run, dan durasi lari maksimum 600 menit.

```
aws omics create-run-group --name TestRunGroup \
--max-cpus 20 \
--max-gpus 10 \
--max-duration 600 \
--max-runs 5
```

Respons dari operasi API ini mencakup ID yang baru dibuat`RunGroup`.

```
{
    "arn": "arn:aws:omics:us-west-2:12345678901:runGroup/2839621",
    "id": "2839621",
    "tags": {}
}
```

Untuk mendapatkan informasi tambahan tentang grup run, gunakan ID ini dengan operasi get-run-groupAPI, seperti yang ditunjukkan pada contoh berikut.

```
aws omics get-run-group --id run group id
```

Responsnya mencakup pengaturan batas untuk grup run dan tag yang ditetapkan.

```
{
    "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
}
```

Anda juga dapat menggunakan operasi list-run-groupAPI untuk melihat semua grup run yang dibuat.

```
aws omics list-run-groups
```

Hapus grup run menggunakan konsol

Anda dapat menghapus grup run jika tidak ada run yang terkait dengan grup run tersebut dengan statusPENDING,STARTING,RUNNING, atauSTOPPING.

Untuk menghapus grup run, ikuti langkah-langkah berikut.

1. Buka konsol HealthOmics .
2. Jika diperlukan, buka panel navigasi kiri (≡). Pilih Jalankan grup.
3. Pada halaman Jalankan grup, pilih grup jalankan yang akan dihapus dan pilih Hapus di xx.

Hapus grup run menggunakan CLI

Anda dapat menghapus grup run jika tidak ada run yang terkait dengan grup run tersebut dengan statusPENDING,STARTING,RUNNING, atauSTOPPING.

Contoh berikut menunjukkan bagaimana Anda dapat menggunakan AWS CLI untuk menghapus grup run. Anda tidak akan menerima tanggapan. Untuk menjalankan contoh, ganti *run group id* dengan ID grup run yang ingin Anda hapus.

```
aws omics delete-run-group --id run group id
```

Panggilan caching untuk menjalankan HealthOmics

AWS HealthOmics mendukung caching panggilan, juga dikenal sebagai resume, untuk alur kerja pribadi. Caching panggilan menyimpan output dari tugas alur kerja yang diselesaikan setelah proses selesai. Proses selanjutnya dapat menggunakan output tugas dari cache, daripada menghitung output tugas lagi. Caching panggilan mengurangi penggunaan sumber daya komputasi, yang menghasilkan durasi berjalan lebih pendek dan penghematan biaya komputasi.

Anda dapat mengakses file output tugas yang di-cache setelah proses selesai. Untuk melakukan debugging tugas lanjutan dan pemecahan masalah, Anda dapat menyimpan file tugas perantara dengan menentukan file ini sebagai output tugas dalam definisi alur kerja.

Anda dapat menggunakan caching panggilan untuk menyimpan hasil tugas yang telah selesai dari proses yang gagal. Jalankan berikutnya dimulai dari tugas terakhir yang berhasil diselesaikan, daripada menghitung tugas yang diselesaikan lagi.

Jika HealthOmics tidak menemukan entri cache yang cocok untuk tugas, proses tidak akan gagal. HealthOmics menghitung ulang tugas dan tugas-tugas dependennya.

Untuk informasi tentang pemecahan masalah caching panggilan, lihat. Memecahkan masalah caching panggilan

Topik

Cara kerja caching panggilan

Untuk menggunakan caching panggilan, Anda membuat cache run dan mengonfigurasinya agar memiliki lokasi Amazon S3 terkait untuk data yang di-cache. Saat Anda memulai proses, Anda menentukan cache run. Cache run tidak didedikasikan untuk satu alur kerja. Berjalan dari beberapa alur kerja dapat menggunakan cache yang sama.

Selama fase ekspor dijalankan, sistem mengekspor output tugas yang diselesaikan ke lokasi Amazon S3. Untuk mengekspor file tugas perantara, deklarasikan file-file ini sebagai output tugas dalam definisi alur kerja. Caching panggilan juga menyimpan metadata secara internal dan membuat hash unik untuk setiap entri cache.

Untuk setiap tugas dalam proses, mesin alur kerja mendeteksi apakah ada entri cache yang cocok untuk tugas ini. Jika tidak ada entri cache yang cocok, HealthOmics hitung tugas. Jika ada entri cache yang cocok, mesin mengambil hasil cache.

Untuk mencocokkan entri cache, HealthOmics gunakan mekanisme hashing yang disertakan dalam mesin alur kerja asli. HealthOmicsmemperluas implementasi hash yang ada ini untuk memperhitungkan HealthOmics variabel, seperti S3 ETag dan intisari wadah ECR.

HealthOmics mendukung caching panggilan untuk versi bahasa alur kerja ini:

- WDL versi 1.0, 1.1, dan versi pengembangan
- Nextflow versi 23.10 dan 24.10
- Semua versi CWL

> **Note**
>
> HealthOmics tidak mendukung caching panggilan untuk alur kerja Ready2Run.

Topik

Model tanggung jawab bersama

Ada tanggung jawab bersama antara pengguna dan AWS untuk menentukan apakah tugas dan proses adalah kandidat yang baik untuk caching panggilan. Call caching mencapai hasil terbaik ketika semua tugas idempoten (eksekusi berulang tugas menggunakan input yang sama menghasilkan hasil yang sama).

Namun, jika tugas mencakup elemen non-deterministik (seperti generasi bilangan acak atau waktu sistem), eksekusi tugas yang berulang menggunakan input yang sama dapat menghasilkan output yang berbeda. Hal ini dapat mempengaruhi efektivitas caching panggilan dengan cara-cara berikut:

- Jika HealthOmics menggunakan entri cache (dibuat oleh proses sebelumnya) yang tidak identik dengan output yang akan dihasilkan oleh eksekusi tugas untuk proses saat ini, proses dapat menghasilkan hasil yang berbeda dari proses yang sama tanpa caching.
- HealthOmics mungkin tidak menemukan entri cache yang cocok untuk tugas yang seharusnya cocok, karena output tugas non-deterministik. Jika tidak menemukan entri cache yang valid, proses menghitung ulang tugas secara tidak perlu, yang mengurangi manfaat penghematan biaya menggunakan caching panggilan.

Berikut ini adalah perilaku tugas yang diketahui yang dapat menyebabkan hasil non-deterministik yang memengaruhi hasil caching panggilan:

- Menggunakan generator angka acak.
- Ketergantungan pada waktu sistem.
- Menggunakan konkurensi (kondisi ras dapat menyebabkan varians keluaran).
- Mengambil file lokal atau jarak jauh di luar apa yang ditentukan dalam parameter input tugas.

Untuk skenario lain yang dapat menyebabkan perilaku non-deterministik, lihat Input proses non-deterministik di situs dokumentasi Nextflow.

Jika Anda menduga bahwa tugas menghasilkan output yang non-deterministik, pertimbangkan untuk menggunakan fitur mesin alur kerja untuk menghindari caching tugas tertentu yang non-deterministik. Untuk petunjuk tentang cara memilih keluar dari caching untuk tugas individual dalam setiap bahasa alur kerja yang didukung, lihat. Fitur caching khusus mesin

Kami menyarankan Anda meninjau alur kerja dan persyaratan tugas spesifik Anda secara menyeluruh sebelum mengaktifkan caching panggilan di lingkungan mana pun di mana caching panggilan yang tidak efektif atau output yang berbeda dari yang diharapkan dapat menimbulkan risiko. Misalnya, potensi keterbatasan caching panggilan harus dipertimbangkan dengan cermat dalam menentukan apakah caching panggilan sesuai untuk kasus penggunaan klinis.

Persyaratan caching untuk tugas

HealthOmics cache output tugas untuk tugas yang memenuhi persyaratan berikut:

- Tugas harus mendefinisikan wadah. HealthOmics tidak akan menyimpan output untuk tugas tanpa wadah.
- Tugas harus menghasilkan satu atau lebih output. Anda menentukan output tugas dalam definisi alur kerja.
- Definisi alur kerja tidak boleh menggunakan nilai dinamis. Misalnya, jika Anda meneruskan parameter ke tugas dengan nilai yang bertambah dengan setiap proses, HealthOmics tidak akan men-cache output tugas.

> **Note**
>
> Jika beberapa tugas dalam proses menggunakan gambar kontainer yang sama, HealthOmics berikan versi gambar yang sama untuk semua tugas ini. Setelah HealthOmics menarik gambar, ia mengabaikan pembaruan apa pun pada gambar kontainer selama durasi proses. Pendekatan ini memberikan pengalaman yang dapat diprediksi dan konsisten serta mencegah potensi masalah yang dapat muncul dari pembaruan pada gambar kontainer yang diterapkan di pertengahan proses.

Jalankan kinerja cache

Saat Anda mengaktifkan caching panggilan untuk dijalankan, Anda mungkin melihat dampak berikut pada kinerja run:

- Selama proses pertama, HealthOmics menyimpan data cache untuk tugas-tugas dalam proses. Anda mungkin mengalami waktu ekspor yang lebih lama untuk menjalankan ini, karena caching panggilan meningkatkan jumlah data ekspor.
- Dalam proses berikutnya, saat melanjutkan proses dari cache, ini dapat mempersingkat jumlah langkah pemrosesan dan mengurangi waktu berjalan Anda.
- Jika Anda juga memilih untuk mendeklarasikan file perantara sebagai output, maka waktu ekspor Anda mungkin lebih lama karena data ini bisa lebih bertele-tele.

Penyimpanan data cache dan peristiwa pembatalan

Tujuan utama dari cache run adalah untuk mengoptimalkan perhitungan tugas dalam proses. Jika ada entri cache pencocokan yang valid untuk tugas, HealthOmics gunakan entri cache alih-alih menghitung ulang tugas. Jika tidak, HealthOmics kembali ke perilaku layanan default, yaitu menghitung ulang tugas dan tugas dependennya. Dengan menggunakan pendekatan ini, cache meleset tidak menyebabkan run gagal.

Kami menyarankan Anda mengelola ukuran cache run. Seiring waktu, entri cache mungkin tidak lagi valid karena pembaruan mesin atau HealthOmics layanan alur kerja atau karena perubahan yang Anda buat dalam menjalankan atau menjalankan tugas. Bagian berikut memberikan detail tambahan.

Topik

- Pembaruan versi manifes dan kesegaran data

- Jalankan perilaku cache
- Kontrol ukuran cache jalankan

Pembaruan versi manifes dan kesegaran data

Secara berkala, HealthOmics layanan dapat memperkenalkan fitur baru atau pembaruan mesin alur kerja yang membatalkan beberapa atau semua entri cache yang dijalankan. Dalam situasi ini, proses Anda dapat mengalami kehilangan cache satu kali.

HealthOmics membuat file manifes JSON untuk setiap entri cache. Untuk proses yang dimulai setelah 12 Februari 2025, file manifes menyertakan parameter versi. Jika pembaruan layanan membatalkan entri cache apa pun, HealthOmics tambahkan nomor versi sehingga Anda dapat mengidentifikasi entri cache lama untuk dihapus.

Contoh berikut menunjukkan file manifes dengan versi diatur ke 2:

```
{
     "arn": "arn:aws:omics:us-west-2:12345678901:runCache/0123456/
cacheEntry/1234567-195f-3921-a1fa-ffffcef0a6a4",
     "s3uri": "s3://example/1234567-d0d1-e230-
d599-10f1539f4a32/1348677/4795326/7e8c69b1-145f-3991-a1fa-ffffcef0a6a4",
     "taskArn": "arn:aws:omics:us-west-2:12345678901:task/4567891",
     "workDir": "/mnt/workflow/1234567-d0d1-e230-d599-10f1539f4a32/workdir/call-
TxtFileCopyTask/5w6tn5feyga7noasjuecdeoqpkltrfo3/wxz2fuddlo6hc4uh5s2lreaayczduxdm",
     "files": [
         {
             "name": "output_txt_file",
             "path": "out/output_txt_file/outfile.txt",
             "etag": "ajdhyg9736b9654673b9fbb486753bc8"
         }
     ],
     "nextflowContext": {},
     "otherOutputs": {},
     "version": 2,
  }
```

Untuk menjalankan dengan entri cache yang tidak lagi valid, buat kembali cache untuk membuat entri baru yang valid. Lakukan langkah-langkah berikut untuk setiap lari:

1. Mulai proses sekali dengan retensi cache disetel ke CACHE ALWAYS. Jalankan ini membuat entri cache baru.

2. Untuk proses selanjutnya, atur retensi cache ke pengaturan sebelumnya (CACHE ALWAYS atau CACHE ON FAILURE).

Untuk membersihkan entri cache yang tidak lagi valid, Anda dapat menghapus entri cache ini dari cache Amazon S3 bucket. HealthOmics jangan pernah menggunakan kembali entri cache ini. Jika Anda memilih untuk mempertahankan entri yang tidak valid, tidak ada dampak pada proses Anda.

> **Note**
>
> Caching panggilan menyimpan data output tugas di lokasi Amazon S3 yang ditentukan untuk cache, yang menimbulkan biaya ke Anda. Akun AWS

Jalankan perilaku cache

Anda dapat mengatur perilaku run cache untuk menyimpan output tugas untuk menjalankan yang gagal (cache pada kegagalan) atau untuk semua proses (cache selalu). Saat Anda membuat cache run, Anda mengatur perilaku cache default untuk semua proses yang menggunakan cache ini. Anda dapat mengganti perilaku default saat memulai proses.

Cache on failureberguna jika Anda men-debug alur kerja yang gagal setelah beberapa tugas berhasil diselesaikan. Proses selanjutnya dilanjutkan dari tugas terakhir yang berhasil diselesaikan jika semua variabel unik yang dipertimbangkan oleh hash identik dengan proses sebelumnya.

Cache alwaysberguna jika Anda memperbarui tugas dalam alur kerja yang berhasil diselesaikan. Kami menyarankan Anda mengikuti langkah-langkah ini:

1. Buat run baru. Atur perilaku Cache ke Cache selalu, dan mulai jalankan.
2. Setelah proses selesai, perbarui tugas dalam alur kerja dan mulai proses baru dengan set perilaku Cache selalu. Proses ini memproses tugas yang diperbarui dan tugas selanjutnya yang memiliki ketergantungan pada tugas yang diperbarui. Semua tugas lain menggunakan hasil cache.
3. Ulangi langkah 2 sesuai kebutuhan, hingga pengembangan selesai untuk tugas yang diperbarui.
4. Gunakan tugas yang diperbarui sesuai kebutuhan di future run. Ingatlah untuk mengalihkan proses berikutnya ke Cache saat gagal jika Anda berencana untuk menggunakan input baru atau berbeda untuk proses ini.

> **Note**
>
> Kami merekomendasikan Cache selalu mode saat menggunakan set data pengujian yang sama, tetapi tidak untuk batch run. Jika Anda mengatur mode ini untuk sejumlah besar proses, sistem dapat mengekspor sejumlah besar data ke Amazon S3, sehingga meningkatkan waktu ekspor dan biaya penyimpanan.

Kontrol ukuran cache jalankan

HealthOmics tidak menghapus atau mengarsipkan secara otomatis data cache yang dijalankan atau menerapkan aturan pembersihan Amazon S3 untuk mengelola data cache. Kami menyarankan Anda melakukan pembersihan cache secara teratur untuk menghemat biaya penyimpanan Amazon S3 dan menjaga ukuran cache run Anda tetap dapat dikelola. Anda dapat menghapus file secara langsung atau menyetel retention/replication kebijakan data pada bucket cache run.

Misalnya, Anda dapat mengonfigurasi kebijakan siklus hidup Amazon S3 agar objek kedaluwarsa setelah 90 hari, atau Anda dapat membersihkan data cache secara manual di akhir setiap proyek pengembangan.

Informasi berikut dapat membantu Anda mengelola ukuran data cache:

- Anda dapat melihat berapa banyak data yang ada di cache dengan memeriksa Amazon S3. HealthOmics tidak memantau atau melaporkan ukuran cache.
- Jika Anda menghapus entri cache yang valid, proses berikutnya tidak akan gagal. HealthOmics menghitung ulang tugas dan tugas-tugas dependennya.
- Jika Anda memodifikasi nama cache atau struktur direktori sedemikian rupa sehingga tidak HealthOmics dapat menemukan entri yang cocok untuk tugas, HealthOmics hitung ulang tugas tersebut.

Jika Anda perlu memeriksa apakah entri cache masih valid, periksa nomor versi manifes cache. Untuk informasi selengkapnya, lihat Pembaruan versi manifes dan kesegaran data.

Membuat cache run

Saat membuat cache run, Anda menentukan lokasi Amazon S3 untuk data cache. Data ini harus segera dapat diakses. Caching panggilan tidak mengambil objek yang diarsipkan di Glacier (seperti kelas penyimpanan GFR dan GDA).

Jika bucket Amazon S3 untuk data cache dimiliki oleh orang lain Akun AWS, berikan ID akun tersebut saat Anda membuat cache run.

Membuat cache run menggunakan konsol

Dari konsol, ikuti langkah-langkah ini untuk membuat cache run.

1. Buka konsol HealthOmics .
2. Jika diperlukan, buka panel navigasi kiri (≡). Pilih Jalankan cache.
3. Dari halaman Jalankan cache, pilih Buat cache run.
4. Di panel Run cache details pada halaman Create run cache, konfigurasikan bidang-bidang ini:
    a. Masukkan nama untuk cache run.
    b. (Opsional) Masukkan deskripsi.
    c. Masukkan lokasi S3 untuk output cache. Pilih bucket di Wilayah yang sama dengan alur kerja Anda.
    d. (Opsional) Masukkan Akun AWS pemilik bucket untuk memverifikasi kepemilikan bucket. Jika Anda tidak memasukkan nilai, nilai default adalah ID akun Anda.
    e. Di bawah perilaku Cache, konfigurasikan perilaku default (apakah akan men-cache output untuk proses yang gagal atau untuk semua proses). Saat memulai proses, Anda dapat mengganti perilaku default secara opsional.
5. (Opsional) Kaitkan satu atau beberapa tag dengan cache run.
6. Pilih Buat cache run. Konsol menampilkan cache run baru di tabel Jalankan cache.

Membuat cache run menggunakan CLI

Gunakan perintah create-run-cacheCLI untuk membuat cache run. Perilaku cache default adalah`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"
```

Jika pembuatan berhasil, Anda menerima respons dengan bidang berikut.

```
{
  "arn": "string",
  "id": "string",
  "status": "ACTIVE"
  "tags": {}
  }
```

Memperbarui cache run

Anda dapat mengubah nama cache, deskripsi, tag, atau perilaku cache, tetapi bukan lokasi S3 untuk cache.

Memperbarui cache run menggunakan konsol

Dari konsol, ikuti langkah-langkah ini untuk memperbarui cache run.

1. Buka konsol HealthOmics .
2. Jika diperlukan, buka panel navigasi kiri (≡). Pilih Jalankan cache.
3. Dari tabel Jalankan cache, pilih cache jalankan yang akan diperbarui, lalu pilih Edit.
4. Di panel Run cache details, Anda dapat memperbarui kolom run cache name, description, dan cache behavior.
5. (Opsional) Kaitkan satu atau beberapa tag baru dengan cache run, atau hapus tag yang ada.
6. Pilih Simpan cache run.

Memperbarui cache run menggunakan CLI

Gunakan perintah update-run-cacheCLI untuk memperbarui cache run.

```
aws omics update-run-cache \
      --name "workflow 123 run cache" \
      --id "workflow id" \
      --description "my run cache" \
      --cache-behavior "CACHE_ALWAYS"
```

Jika pembaruan berhasil, Anda menerima respons tanpa bidang data.

Menghapus cache run

Anda dapat menghapus cache run jika tidak ada run aktif yang menggunakannya. Jika ada proses yang menggunakan cache run, tunggu hingga proses selesai atau Anda dapat membatalkan proses.

Menghapus cache run akan menghapus sumber daya dan metadatanya, tetapi tidak menghapus data di Amazon S3. Setelah Anda menghapus cache, Anda tidak dapat melampirkannya kembali atau menggunakannya untuk menjalankan selanjutnya.

Data yang di-cache tetap ada di Amazon S3 untuk pemeriksaan Anda. Anda dapat menghapus data cache lama menggunakan Delete operasi S3 standar. Atau, buat kebijakan siklus hidup Amazon S3 untuk menghapus data cache yang tidak lagi digunakan.

Menghapus cache run menggunakan konsol

Dari konsol, ikuti langkah-langkah ini untuk menghapus cache run.

1. Buka konsol HealthOmics .
2. Jika diperlukan, buka panel navigasi kiri (≡). Pilih Jalankan cache.
3. Dari tabel Jalankan cache, pilih cache jalankan yang akan dihapus.
4. Dari menu tabel Jalankan cache, pilih Hapus.
5. Dari dialog modal, simpan tautan data cache Amazon S3 untuk referensi future, lalu konfirmasikan bahwa Anda ingin menghapus cache run.

   Anda dapat menggunakan tautan Amazon S3 untuk memeriksa data yang di-cache, tetapi Anda tidak dapat menautkan kembali data ke cache run lainnya. Hapus data cache saat Anda selesai inspeksi.

Menghapus cache run menggunakan CLI

Gunakan perintah delete-run-cacheCLI untuk menghapus cache run.

```
aws omics delete-run-cache \
      --id "my cache id"
```

Jika penghapusan berhasil, Anda menerima respons tanpa bidang data.

Isi cache run

HealthOmics mengatur cache run Anda dengan struktur berikut di bucket S3 Anda:

```
s3://{cache.S3location}/{cache.uuid}/runID/taskID/{cacheentry.uuid}/
```

Cache.uuid adalah id unik global untuk cache. Cacheentry.uuid adalah uuid unik secara global untuk tugas yang di-cache. HealthOmics menetapkan uuids ke cache dan tugas.

Untuk semua mesin alur kerja, cache berisi file-file berikut:

- {cacheentryuuid}.jsonFile - HealthOmics membuat file manifes ini, yang berisi informasi tentang cache, termasuk daftar semua item dalam cache, dan versi cache.
- File output tugas - Setiap output tugas terdiri dari satu atau lebih file, seperti yang didefinisikan oleh tugas.

Untuk alur kerja yang menggunakan Nextflow, mesin Nextflow membuat file tambahan ini dalam cache:

- command.outFile - File ini berisi isi stdout eksekusi tugas.
- .exitcodeFile - File ini berisi kode keluar tugas (integer).

> **Note**
>
> Jika Anda ingin mengakses file tugas perantara di cache run untuk pemecahan masalah lanjutan, deklarasikan file ini sebagai output tugas dalam definisi alur kerja.

Fitur caching khusus mesin

HealthOmics mencoba memberikan implementasi caching panggilan yang konsisten di seluruh mesin alur kerja. Ada beberapa perbedaan berdasarkan bagaimana setiap mesin alur kerja menangani kasus tertentu:

- Alur berikutnya

  - Caching di berbagai versi Nextflow tidak dijamin. Misalnya, jika Anda menjalankan tugas di v23.10.0 dan kemudian menjalankan tugas yang sama di v24.10.8, HealthOmics mungkin menganggap proses kedua sebagai cache yang hilang.
  - Anda dapat mematikan caching untuk tugas individual dengan menggunakan false direktif cache. Untuk informasi tentang arahan ini, lihat Proses dalam spesifikasi Nextflow.
  - HealthOmics menggunakan mode lunak Nextflow, tetapi tidak mendukung mode cache dalam.
  - Caching mengevaluasi setiap objek S3 individu jika Anda menggunakan pola glob di jalur S3 ke input untuk tugas. Jika Anda menambahkan objek baru, HealthOmics hitung ulang hanya tugas yang menggunakan objek baru.
  - HealthOmics tidak men-cache percobaan ulang tugas. Perilaku ini konsisten dengan perilaku default Nextflow.
- WDL
  - HealthOmics mendukung jenis “direktori” baru untuk input saat Anda menggunakan versi pengembangan alur kerja WDL. Untuk caching panggilan, jika ada objek di direktori yang berubah, HealthOmics hitung ulang semua tugas yang memasukkan direktori.
  - HealthOmics mendukung caching tingkat tugas, tetapi bukan caching tingkat alur kerja.
  - Anda dapat menonaktifkan caching untuk tugas individual dengan menggunakan atribut volatile. Untuk informasi selengkapnya, lihat Nonaktifkan caching tingkat tugas dengan atribut volatile.
- CWL
  - Output konstan dari tugas tidak terlihat secara eksplisit dari manifes. HealthOmics cache output konstan sebagai file perantara.
  - Anda dapat mengontrol caching untuk tugas individual dengan menggunakan WorkReusefitur ini.

Menggunakan cache run

Secara default, run tidak menggunakan cache run. Untuk menggunakan cache untuk menjalankan, Anda menentukan cache run dan perilaku run cache saat Anda memulai proses.

Setelah proses selesai, Anda dapat menggunakan konsol, CloudWatch Log, atau operasi API untuk melacak klik cache atau memecahkan masalah cache. Untuk detailnya, lihat Melacak informasi caching panggilan dan Memecahkan masalah caching panggilan.

Jika satu atau beberapa tugas dalam proses menghasilkan output non-deterministik, kami sangat menyarankan agar Anda tidak menggunakan caching panggilan untuk menjalankan, atau Anda

memilih keluar dari tugas khusus ini dari caching. Untuk informasi selengkapnya, lihat Model tanggung jawab bersama.

> **Note**
>
> Anda memberikan peran layanan IAM saat memulai proses. Untuk menggunakan caching panggilan, peran layanan memerlukan izin untuk mengakses lokasi Amazon S3 run cache. Untuk informasi selengkapnya, lihat Peran layanan untuk AWS HealthOmics.

Anda dapat menggunakan Amazon Q CLI untuk menganalisis dan mengelola data cache run Anda. Untuk informasi selengkapnya, lihat Contoh prompt untuk Amazon Q CLI dan tutorial AI HealthOmics generatif Agentic. GitHub

Topik

- Mengkonfigurasi run dengan run cache menggunakan konsol
- Mengkonfigurasi run dengan run cache menggunakan CLI
- Kasus kesalahan untuk menjalankan cache
- Melacak informasi caching panggilan

Mengkonfigurasi run dengan run cache menggunakan konsol

Dari konsol, Anda mengonfigurasi cache run untuk dijalankan saat Anda memulai proses.

1. Buka konsol HealthOmics .
2. Jika diperlukan, buka panel navigasi kiri (≡). Pilih Runs.
3. Pada halaman Runs, pilih run untuk memulai.
4. Pilih Mulai jalankan dan selesaikan langkah 1 dan 2 dari Mulai jalankan seperti yang dijelaskan dalamMemulai lari menggunakan konsol.
5. Pada langkah 3 dari Mulai jalankan, pilih Pilih cache run yang ada.
6. Pilih cache dari daftar drop-down Jalankan ID cache.
7. Untuk mengganti perilaku cache run default, pilih perilaku Cache untuk dijalankan. Untuk informasi selengkapnya, lihat Jalankan perilaku cache.
8. Lanjutkan ke langkah 4 dari Mulai jalankan.

Mengkonfigurasi run dengan run cache menggunakan CLI

Untuk memulai proses yang menggunakan cache run, tambahkan parameter cache-id ke perintah CLI start-run. Secara opsional, gunakan `cache-behavior` parameter untuk mengganti perilaku default yang Anda konfigurasikan untuk menjalankan cache. Contoh berikut hanya menunjukkan bidang cache untuk perintah:

```
aws omics start-run \
        ...
      --cache-id "xxxxxx"    \
      --cache-behavior  CACHE_ALWAYS
```

Jika operasi berhasil, Anda menerima respons tanpa bidang data.

Kasus kesalahan untuk menjalankan cache

Untuk skenario berikut, HealthOmics mungkin tidak cache output tugas, bahkan untuk menjalankan dengan perilaku cache diatur ke Cache selalu.

- Jika proses mengalami kesalahan sebelum tugas pertama berhasil diselesaikan, tidak ada output cache untuk diekspor.
- Jika proses ekspor gagal, HealthOmics tidak menyimpan output tugas ke lokasi cache Amazon S3.
- Jika proses gagal karena filesystem out of space kesalahan, caching panggilan tidak menyimpan output tugas apa pun.
- Jika Anda membatalkan proses, caching panggilan tidak menyimpan output tugas apa pun.
- Jika proses mengalami batas waktu berjalan, caching panggilan tidak menyimpan output tugas apa pun, bahkan jika Anda mengonfigurasi proses untuk menggunakan cache saat gagal.

Melacak informasi caching panggilan

Anda dapat melacak peristiwa caching panggilan (seperti menjalankan cache hits) menggunakan konsol, CLI, CloudWatch atau Log.

Topik

- Lacak klik cache menggunakan konsol
- Lacak caching panggilan menggunakan CLI

- Lacak caching panggilan menggunakan CloudWatch Log

Lacak klik cache menggunakan konsol

Di halaman run details untuk menjalankan, tabel Jalankan tugas menampilkan informasi klik Cache untuk setiap tugas. Tabel ini juga menyertakan tautan ke entri cache terkait. Gunakan prosedur berikut untuk melihat informasi cache hit untuk menjalankan.

1. Buka konsol HealthOmics .
2. Jika diperlukan, buka panel navigasi kiri (≡). Pilih Runs.
3. Pada halaman Runs, pilih run untuk diperiksa.
4. Pada halaman run details, pilih tab Jalankan tugas untuk menampilkan tabel tugas.
5. Jika tugas memiliki cache hit, kolom Cache hit berisi link ke lokasi entri cache run di Amazon S3.
6. Pilih tautan untuk memeriksa entri run cache.

Lacak caching panggilan menggunakan CLI

Gunakan perintah CLI get-run untuk mengonfirmasi apakah run menggunakan cache panggilan.

```
aws omics get-run --id 1234567
```

Sebagai tanggapan, jika `cacheId` bidang disetel, run menggunakan cache itu.

Gunakan perintah list-run-tasksCLI untuk mengambil lokasi data cache untuk setiap tugas yang di-cache dalam proses.

```
aws omics list-run-tasks --id 1234567
```

Sebagai tanggapan, jika bidang cacheHit untuk tugas benar, bidang caches3URI menyediakan lokasi data cache untuk tugas itu.

Anda juga dapat menggunakan perintah get-run-taskCLI untuk mengambil lokasi data cache untuk tugas tertentu:

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

Lacak caching panggilan menggunakan CloudWatch Log

HealthOmics membuat log aktivitas cache di grup `/aws/omics/WorkflowLog` CloudWatch log. <cache_id><cache_uuid>Ada aliran log untuk setiap cache run: Runcache//.

Untuk menjalankan yang menggunakan caching panggilan, buat HealthOmics entri CloudWatch Log untuk peristiwa ini:

- membuat entri cache (CACHE_ENTRY_CREATED)
- mencocokkan entri cache (CACHE_HIT)
- gagal mencocokkan entri cache (CACHE_MISS)

Untuk informasi selengkapnya tentang log ini, lihatLog masuk CloudWatch.

Gunakan kueri CloudWatch Insights berikut pada grup `/aws/omics/WorkflowLog` log untuk menampilkan jumlah klik cache per run untuk cache ini:

```
filter @logStream like 'runCache/<CACHE_ID>/'
 fields @timestamp, @message
 filter logMessage like 'CACHE_HIT'
 parse "run: *," as run
 stats count(*) as cacheHits by run
```

Gunakan kueri berikut untuk mengembalikan jumlah entri cache yang dibuat oleh setiap proses:

```
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
```

Berbagi HealthOmics alur kerja

Sebagai pemilik alur kerja pribadi, Anda dapat berbagi alur kerja dengan Akun AWS di wilayah yang sama. Untuk berbagi alur kerja dengan lebih dari satu Akun AWS, Anda membuat beberapa pembagian dari alur kerja yang sama.

Sebagai pemilik, Anda dapat mencabut akses ke alur kerja bersama dengan menghapus pembagian.

> **Note**
>
> HealthOmics secara otomatis memungkinkan alur kerja bersama untuk mengakses repositori Amazon ECR saat alur kerja berjalan di akun pelanggan. Anda tidak perlu memberikan akses repositori tambahan untuk alur kerja bersama.

Saat Anda berbagi alur kerja, pelanggan dapat menggunakan salah satu versi alur kerja. Jika Anda memerlukan kontrol akses tingkat versi untuk alur kerja bersama, sebaiknya buat alur kerja terpisah daripada menggunakan versi alur kerja.

Topik

Berlangganan alur kerja bersama

Untuk berlangganan alur kerja bersama, Anda mengikuti langkah-langkah keseluruhan ini untuk menerima dan menggunakan alur kerja:

1. Gunakan konsol atau API untuk menerima pembagian. Setel wilayah Anda saat ini ke wilayah yang sama dengan permintaan berbagi.
   - Untuk menemukan permintaan berbagi di konsol, navigasikan ke halaman Semua Sumber Daya berbagi, lalu pilih tab Dibagikan dengan saya.
2. Gunakan konsol atau API untuk membuat proses untuk alur kerja bersama.
   - Untuk menemukan halaman detail alur kerja di konsol, navigasikan ke Dibagikan dengan saya (lihat langkah 1), lalu pilih tautan Sumber daya untuk alur kerja bersama.
3. Anda memberikan data masukan Anda sendiri untuk alur kerja.
4. Alur kerja bersama berjalan di Anda Akun AWS.

Sebagai pelanggan alur kerja bersama, sistem memblokir Anda dari melakukan tindakan alur kerja berikut:

- Mengekspor alur kerja bersama
- Menjalankan kembali alur kerja bersama
  - Anda membuat proses baru untuk alur kerja bersama.
- Berbagi kembali alur kerja.
- Menetapkan tag ke alur kerja.
- Menghapus alur kerja.
  - Saat Anda tidak lagi membutuhkan alur kerja, Anda menghapus pembagian alur kerja.

Lihat Berbagi sumber daya lintas akun di AWS HealthOmics untuk informasi tambahan tentang berbagi sumber daya.

Memantau status pembagian alur kerja

HealthOmics mengirimkan acara EventBridge untuk setiap perubahan status dari pembagian alur kerja. Jika Anda ingin menerima pemberitahuan tentang perubahan status tertentu, siapkan EventBridge aturan untuk memantau Pembagian alur kerja peristiwa Perubahan Status. Contoh:

- Anda ingin pemberitahuan setiap kali Anda menerima permintaan berbagi alur kerja, dan setiap kali pengguna mencabut pembagian alur kerja.
- Setelah memulai permintaan berbagi alur kerja, Anda ingin menerima pemberitahuan saat pengguna menerima atau menolak permintaan tersebut.

Untuk detail tentang menggunakan acara, lihatMenggunakan EventBridge dengan AWS HealthOmics.

Berbagi alur kerja pribadi menggunakan konsol

Dari konsol, Anda dapat berbagi alur kerja pribadi dengan Akun AWS di wilayah yang sama dengan alur kerja.

Untuk berbagi alur kerja pribadi

1. Buka konsol HealthOmics .
2. Jika diperlukan, buka panel navigasi kiri (≡). Pilih Alur kerja pribadi.

3. Dari tabel Alur kerja pada halaman Alur kerja pribadi, pilih alur kerja untuk dibagikan, dan pilih Bagikan.
4. Di panel Bagikan rincian halaman alur kerja Bagikan, masukkan nama deskriptif untuk berbagi dan masukkan Akun AWS pelanggan.
5. Pilih Bagikan sumber daya. Konsol menampilkan pembagian sumber daya di halaman Semua berbagi sumber daya.

Keadaan awal saham tertunda. Setelah pelanggan menerima bagian, negara berubah menjadi aktif.

Berbagi alur kerja pribadi menggunakan CLI

Gunakan operasi create-share API untuk membuat pembagian alur kerja. Pelanggan utama adalah pengguna Akun AWS yang akan mendapatkan akses ke alur kerja.

```
aws omics create-share \
  --resource-arn "arn:aws:omics:us-west-2:555555555555:workflow/123456" \
  --principal-subscriber "123456789012" \
  --name "my_Share-123"
```

Jika pembuatan berhasil, Anda menerima respons dengan ID dan status berbagi.

```
{
"shareId": "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a",
"name": "my_Share-123",
"status": "PENDING"
}
```

Bagian tetap dalam status tertunda sampai pelanggan menerimanya menggunakan operasi `accept-share` API.

Lihat Berbagi sumber daya lintas akun di AWS HealthOmics contoh penggunaan API lainnya.

Menerima alur kerja bersama menggunakan konsol

Anda dapat menggunakan konsol untuk menerima pembagian alur kerja yang ditawarkan. Pastikan untuk mengatur konsol ke Wilayah yang sama dengan alur kerja.

1. Buka konsol HealthOmics .

2. Jika diperlukan, buka panel navigasi kiri (≡). Pilih All Resource share, lalu pilih tab Shared with me.
3. Dari tabel Sumber daya yang dibagikan dengan saya, pilih pembagian alur kerja, lalu pilih Terima.

Setelah Anda menerima alur kerja, pilih tautan Sumber daya untuk alur kerja bersama untuk melihat detailnya.

Menjalankan alur kerja bersama menggunakan konsol

Setelah Anda menerima pembagian alur kerja, Anda dapat memulai menjalankan alur kerja.

1. Buka konsol HealthOmics .
2. Jika diperlukan, buka panel navigasi kiri (≡). Pilih All Resource share, lalu pilih tab Shared with me.
3. Dari tabel Sumber daya yang dibagikan dengan saya, pilih tautan Sumber daya untuk alur kerja bersama.
4. Di halaman Rincian alur kerja, pilih Buat jalankan.

   Konsol membuka halaman Buat jalankan, dengan tipe alur kerja (bersama) dan ID Alur Kerja yang telah diisi sebelumnya.
5. Konfigurasikan bidang yang tersisa di formulir Create run. Untuk informasi tambahan, lihat Memulai lari menggunakan konsol.

Menjalankan alur kerja bersama menggunakan API

Gunakan get-workflow untuk mengambil ARN dari alur kerja bersama.

```
aws omics get-workflow --id 1234567 \
--workflow-owner-id 55555555555
```

Saat Anda menjalankan alur kerja, berikan Akun AWS ID pemilik alur kerja dan ARN alur kerja bersama.

```
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
```

Alur kerja Ready2Run di HealthOmics

Alur kerja Ready2Run adalah alur kerja yang telah dikonfigurasi sebelumnya yang diterbitkan oleh penerbit pihak ketiga. Beberapa penerbit, seperti Sentieon Inc, menawarkan alur kerja berbasis langganan. Alur kerja Ready2Run lainnya tidak memerlukan langganan, dan beberapa alur kerja bersifat open source, seperti alur kerja NF-Core.

Alur kerja Ready2Run sangat cocok untuk skenario berikut:

- Anda ingin fokus pada analisis keluaran pipa dan menghasilkan hasil, tanpa perlu menyiapkan infrastruktur yang mendasarinya.
- Anda ingin mereplikasi hasil Anda menggunakan alur kerja yang telah ditetapkan.
- Sebagai pengembang perangkat lunak, Anda ingin mengintegrasikan aplikasi Anda secara langsung dengan HealthOmics SDK.

HealthOmics mendukung pembuatan versi untuk alur kerja Ready2Run. Untuk alur kerja Ready2Run yang menawarkan versi, Anda dapat menentukan nama versi saat memulai proses.

Semua alur kerja Ready2Run menyediakan log, termasuk CloudWatch log, yang dapat Anda gunakan untuk pemecahan masalah.

> **Note**
>
> Alur kerja Sentieon Ready2Run berbasis langganan. Saat Anda menjalankan alur kerja Sentieon Ready2Run untuk pertama kalinya di akun, Sentieon secara otomatis membuat lisensi evaluasi dua minggu untuk akun Anda. Akun AWS Lisensi ini berlaku untuk semua alur kerja Sentieon Ready2Run. Setelah periode evaluasi berakhir, Anda dapat meminta lisensi permanen atau meminta perpanjangan lisensi evaluasi. Lihat Subscribing to Sentieon Ready2Run workflows untuk detail.

Topik

Tersedia alur kerja Ready2Run di HealthOmics

Tabel berikut mencantumkan alur kerja Ready2Run yang tersedia di. HealthOmics

Anda dapat masuk ke HealthOmicskonsol untuk melihat informasi terperinci tentang alur kerja ini, termasuk parameter input dan diagram alur kerja. Untuk informasi harga tentang alur kerja Ready2Run, lihat Harga. HealthOmics

> **Note**
>
> Setiap alur kerja Ready2Run memiliki ukuran file input maksimum. Ukuran file maksimum ini tidak dapat disesuaikan.

Nama alur kerja	Penerbit	Diperlukan langganan?	Ukuran file masukan maksimum (GiB)	Perkiraan waktu berjalan (HH:MM)
AlphaFold untuk 601-1200 residu	Google DeepMind	Tidak	1	11:15
AlphaFold hingga 600 residu	Google DeepMind	Tidak	1	7:30
Basis2Fastq untuk 2x150	Elemen Biosains	Tidak	1000	1:45
Basis2Fastq untuk 2x300	Elemen Biosains	Tidak	1000	1:30
Basis2Fastq untuk 2x75	Elemen Biosains	Tidak	500	0:45
ESMFold hingga 800 residu	Penelitian Meta	Tidak	1	0:15
GATK-BP fq2bam	Institut Luas	Tidak	64	10:10

Nama alur kerja	Penerbit	Diperlukan langganan?	Ukuran file masukan maksimum (GiB)	Perkiraan waktu berjalan (HH:MM)
GATK-BP Germline bam2vcf untuk genom 30x	Institut Luas	Tidak	39	2:45
GATK-BP Germline fq2vcf untuk genom 30x	Institut Luas	Tidak	64	12:30
GATK-BP Somatik WES bam2vcf	Institut Luas	Tidak	86	1:30
NVIDIA Parabricks BAM2 FQ2 BAM WGS hingga 30X	Perusahaan NVIDIA	Tidak	80	1:39
NVIDIA Parabricks BAM2 FQ2 BAM WGS hingga 50X	Perusahaan NVIDIA	Tidak	120	2:45
NVIDIA Parabricks BAM2 FQ2 BAM WGS hingga 5X	Perusahaan NVIDIA	Tidak	20	0:18
NVIDIA Parabricks FQ2 BAM WGS hingga 30X	Perusahaan NVIDIA	Tidak	71	1:00

Nama alur kerja	Penerbit	Diperlukan langganan?	Ukuran file masukan maksimum (GiB)	Perkiraan waktu berjalan (HH:MM)
NVIDIA Parabricks FQ2 BAM WGS hingga 50X	Perusahaan NVIDIA	Tidak	137	1:45
NVIDIA Parabricks FQ2 BAM WGS hingga 5X	Perusahaan NVIDIA	Tidak	13	0:15
NVIDIA Parabricks Germline DeepVariant WGS hingga 30X	Perusahaan NVIDIA	Tidak	71	2:00
NVIDIA Parabricks Germline DeepVariant WGS hingga 50X	Perusahaan NVIDIA	Tidak	137	3:30
NVIDIA Parabricks Germline DeepVariant WGS hingga 5X	Perusahaan NVIDIA	Tidak	12	0:30

Nama alur kerja	Penerbit	Diperlukan langganan?	Ukuran file masukan maksimum (GiB)	Perkiraan waktu berjalan (HH:MM)
NVIDIA Parabricks Germline HaplotypeCaller WGS hingga 30X	Perusahaan NVIDIA	Tidak	71	1:15
NVIDIA Parabricks Germline HaplotypeCaller WGS hingga 50X	Perusahaan NVIDIA	Tidak	137	2:00
NVIDIA Parabricks Germline HaplotypeCaller WGS hingga 5X	Perusahaan NVIDIA	Tidak	13	0:15
NVIDIA Parabricks Somatic Mutect2 WGS hingga 50X	Perusahaan NVIDIA	Tidak	196	0:45
sc RNAseq dengan Kallisto BUStools	NF-inti	Tidak	119	1:30
sc RNAseq dengan Salmon Alevin-goreng	NF-inti	Tidak	119	2:30

Nama alur kerja	Penerbit	Diperlukan langganan?	Ukuran file masukan maksimum (GiB)	Perkiraan waktu berjalan (HH:MM)
sc RNAseq dengan STARsolo	NF-inti	Tidak	119	2:30
Sentieon Germline BAM WES hingga 300x	Sentieon, Inc.	Ya	9	1:00
Sentieon Germline BAM WGS hingga 32x	Sentieon, Inc.	Ya	18	1:30
Sentieon Germline FASTQ WES hingga 100x	Sentieon, Inc.	Ya	5	0:45
Sentieon Germline FASTQ WES hingga 300x	Sentieon, Inc.	Ya	26	2:00
Sentieon Germline FASTQ WGS hingga 32x	Sentieon, Inc.	Ya	51	3:30
Sentieon LongRead untuk ONT	Sentieon, Inc.	Ya	25	1:30
Sentieon LongRead untuk PacBio HiFi	Sentieon, Inc.	Ya	58	4:00

Nama alur kerja	Penerbit	Diperlukan langganan?	Ukuran file masukan maksimum (GiB)	Perkiraan waktu berjalan (HH:MM)
Sentieon Somatik WES	Sentieon, Inc.	Ya	50	2:30
Sentieon Somatik WGS	Sentieon, Inc.	Ya	113	4:30
Ultima Genomics hingga DeepVariant 40x	Genomik Ultima	Tidak	91	1:55

Saat Anda menggunakan alur kerja Ready2Run, alur kerja Anda sudah dikonfigurasi sebelumnya dan tidak dapat diedit. Berbeda dengan alur kerja pribadi, alur kerja Ready2Run tidak mendukung hal berikut:

- Meningkatkan ukuran file input maksimum
- Mengubah sumber daya komputasi atau menjalankan penyimpanan
- Mengubah definisi alur kerja atau kontainer
- Menambahkan run ke grup run
- Berbagi alur kerja

Jika penerbit telah membagikan alur kerja Ready2Run GitHub, Anda dapat membuat alur kerja pribadi Anda sendiri berdasarkan alur kerja Ready2Run. Tabel berikut menyediakan link ke GitHub alur kerja untuk setiap penerbit.

Penerbit	Alur kerja pada GitHub
Google DeepMind, Penelitian Meta	Alur kerja pelipatan protein
Elemen Biosains	Untuk informasi, hubungi Element Biosciences
Institut Luas	Alur kerja GATK

Penerbit	Alur kerja pada GitHub
Perusahaan NVIDIA	Alur kerja parabricks
nf-inti	Alur kerja NF-Core
Sentimeon	Alur kerja Sentieon
Genomik Ultima	Alur kerja Ultima Genomics

Berlangganan alur kerja Sentieon Ready2Run

Alur kerja Sentieon Ready2Run berbasis langganan. Saat Anda menjalankan alur kerja Sentieon Ready2Run untuk pertama kalinya di akun, Sentieon secara otomatis membuat lisensi evaluasi dua minggu untuk akun Anda. Akun AWS Lisensi ini berlaku untuk semua alur kerja Sentieon Ready2Run. Setelah periode evaluasi berakhir, Anda dapat meminta lisensi permanen atau meminta perpanjangan lisensi evaluasi.

Ikuti langkah-langkah ini untuk berlangganan alur kerja Sentieon Ready2Run:

- Temukan ID Pengguna AWS Canonical Anda dengan mengikuti petunjuk ini.
- Kirim email ke grup dukungan Sentieon (support@sentieon.com) untuk meminta lisensi perangkat lunak. Berikan ID Pengguna AWS Canonical Anda di email.

Memulai HealthOmics alur kerja Ready2Run menggunakan konsol

Menggunakan alur kerja Ready2Run di konsol mirip dengan menggunakan alur kerja pribadi. Salah satu perbedaan utama adalah bahwa penerbit alur kerja menyediakan data sampel, sehingga Anda dapat mencoba alur kerja tanpa membuat data Anda sendiri.

Untuk menggunakan alur kerja Ready2Run di konsol

1. Buka konsol HealthOmics .
2. Jika diperlukan, buka panel navigasi kiri (≡). Pilih alur kerja Ready2Run.
3. Pada halaman alur kerja Ready2Run, pilih alur kerja yang ingin Anda gunakan. Konsol membuka halaman detail untuk alur kerja itu.

4. Tab detail mencantumkan informasi seperti nama, harga daftar per proses, deskripsi, jenis bahasa alur kerja, kapasitas penyimpanan jalankan, status, tanggal pembuatan, dan parameter dengan deskripsi. Tab detail juga memberi tahu Anda apakah alur kerja memerlukan langganan.
5. Untuk menggunakan alur kerja, pilih Buat jalankan
6. Di halaman Tentukan rincian jalankan, masukkan nama jalankan. Secara opsional, Anda dapat menentukan versi alur kerja. Anda juga dapat menambahkan prioritas run ke run.
7. Masukkan atau pilih lokasi Amazon S3 untuk output run.
8. Untuk mode penyimpanan metadata Jalankan, pilih apakah akan mempertahankan atau menghapus metadata run.
9. Di panel peran Layanan, pilih apakah akan menggunakan peran layanan yang ada atau membuat yang baru.
10. (Opsional) Tambahkan tag untuk membantu mengidentifikasi dan mengelola proses Anda.
11. Pilih Berikutnya.
12. Dari halaman Tambahkan parameter, pilih salah satu opsi untuk menambahkan nilai parameter run:

    - Pilih file parameter (dalam format JSON) dari lokasi Amazon S3.
    - Pilih file parameter (dalam format JSON) dari drive lokal Anda.
    - Masukkan nilai parameter secara manual.
    - Jalankan alur kerja dengan data sampel Ready2Run yang disediakan oleh penerbit alur kerja.
13. Jika Anda mengunggah file JSON, konsol mem-parsing file dan melakukan validasi sebaris. Anda kemudian dapat memperbarui nilai parameter Anda secara manual sesuai kebutuhan.
14. Pilih Berikutnya.
15. Tinjau input Anda, lalu pilih Mulai jalankan.

Memulai HealthOmics alur kerja Ready2Run menggunakan API

Sebagian besar operasi API berperilaku serupa untuk alur kerja Ready2Run dan alur kerja pribadi.

Untuk mengembalikan daftar alur kerja Ready2Run yang tersedia, gunakan daftar-alur kerja dengan parameter yang disetel ke RUN. `type` READY2

```
aws omics list-workflows --type READY2RUN
```

Setelah mengidentifikasi alur kerja yang akan dijalankan dari respons list-alur kerja, Anda dapat menggunakan get-workflow dengan parameter untuk mendapatkan detail selengkapnya. `--id`

```
aws omics get-workflow --type READY2RUN --id workflow id
```

Untuk menjalankan alur kerja Ready2Run, Anda dapat menggunakan operasi API start-run dengan parameter tipe alur kerja yang disetel ke, seperti yang ditunjukkan pada contoh berikut `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
```

Untuk menentukan versi alur kerja, gunakan parameter versi alur kerja, seperti yang ditunjukkan dalam contoh ini.

```
aws-omics start-run \
  --workflow-type READY2RUN \
  ...
  --version-name '3.0.0'
```

Untuk memantau proses Anda, Anda dapat menggunakan operasi get-run API, seperti yang ditunjukkan.

```
aws-omics get-run \
  --id run id
```

HealthOmics penyimpanan

Gunakan HealthOmics penyimpanan untuk menyimpan, mengambil, mengatur, dan berbagi data genomik secara efisien dan dengan biaya rendah. HealthOmics penyimpanan memahami hubungan antara objek data yang berbeda, sehingga Anda dapat menentukan set baca mana yang berasal dari data sumber yang sama. Ini memberi Anda asal data.

Data yang disimpan dalam `ACTIVE` status dapat diambil segera. Data yang belum diakses selama 30 hari atau lebih disimpan dalam `ARCHIVE` status. Untuk mengakses data yang diarsipkan, Anda dapat mengaktifkannya kembali melalui operasi API atau konsol.

HealthOmics toko urutan dirancang untuk menjaga integritas konten file. Namun, kesetaraan bitwise dari file data yang diimpor dan file yang diekspor tidak dipertahankan karena kompresi selama tiering aktif dan arsip.

Selama konsumsi, buat HealthOmics tag entitas, atau HealthOmics ETag, untuk memungkinkan memvalidasi integritas konten file data Anda. Bagian sekuensing diidentifikasi dan ditangkap sebagai ETag pada tingkat sumber dari set baca. ETag Perhitungan tidak mengubah file aktual atau data genom. Setelah set baca dibuat, ETag seharusnya tidak berubah sepanjang siklus hidup sumber set baca. Ini berarti bahwa mengimpor ulang file yang sama menghasilkan ETag nilai yang sama yang dihitung.

Topik

HealthOmics ETags dan asal-usul data

A HealthOmics ETag (tag entitas) adalah hash dari konten yang dicerna di toko urutan. Ini menyederhanakan pengambilan dan pemrosesan data sambil mempertahankan integritas konten dari file data yang dicerna. Ini ETag mencerminkan perubahan pada konten semantik objek, bukan metadata-nya. Jenis dan algoritma set baca yang ditentukan menentukan bagaimana ETag dihitung. ETag Perhitungan tidak mengubah file aktual atau data genom. Ketika skema jenis file dari set baca mengizinkannya, urutan menyimpan memperbarui bidang yang ditautkan ke sumber data.

File memiliki identitas bitwise dan identitas semantik. Identitas bitwise berarti bahwa bit dari sebuah file identik, dan identitas semantik berarti bahwa isi dari sebuah file identik. Identitas semantik tahan terhadap perubahan metadata dan perubahan kompresi karena menangkap integritas konten file.

Set baca di penyimpanan HealthOmics urutan menjalani compression/decompression siklus dan pelacakan asal data di seluruh siklus hidup objek. Selama pemrosesan ini, identitas bitwise dari file yang tertelan dapat berubah dan diharapkan berubah setiap kali file diaktifkan; namun, identitas semantik file dipertahankan. Identitas semantik ditangkap sebagai tag HealthOmics entitas, atau ETag yang dihitung selama penyerapan penyimpanan urutan dan tersedia sebagai metadata set baca.

Ketika skema tipe file dari set baca mengizinkannya, lapisan pembaruan penyimpanan urutan ditautkan ke sumber data. Untuk file UBam, BAM, dan CRAM, `Comment` tag baru `@CO` atau ditambahkan ke header. Komentar berisi ID penyimpanan urutan dan stempel waktu konsumsi.

Amazon S3 ETags

Saat mengakses file menggunakan URI Amazon S3, operasi API Amazon S3 juga dapat mengembalikan nilai Amazon S3 dan checksum. ETag Nilai Amazon S3 ETag dan checksum berbeda dari nilai HealthOmics ETags karena mereka mewakili identitas bitwise file. Untuk mempelajari metadata deskriptif dan Objek selengkapnya, lihat dokumentasi Amazon S3 Object API. ETag Nilai Amazon S3 dapat berubah dengan setiap siklus aktivasi set baca dan Anda dapat menggunakannya untuk memvalidasi pembacaan file. Namun, jangan cache ETag nilai Amazon S3 yang akan digunakan untuk validasi identitas file selama siklus hidup file karena tidak tetap konsisten. Sebaliknya, HealthOmics ETag tetap konsisten sepanjang siklus hidup set baca.

Bagaimana HealthOmics menghitung ETags

ETag Itu dihasilkan dari hash dari konten file yang dicerna. Keluarga ETag algoritme diatur ke secara MD5up default, tetapi dapat dikonfigurasi secara berbeda selama pembuatan penyimpanan urutan.

Ketika ETag dihitung, algoritma dan hash yang dihitung ditambahkan ke set baca. MD5 Algoritma yang didukung untuk jenis file adalah sebagai berikut.

- FASTQ_ MD5up - Menghitung MD5 hash dari sumber set baca FASTQ lengkap yang tidak terkompresi.
- BAM_ MD5up — Menghitung MD5 hash dari bagian penyelarasan dari sumber kumpulan baca BAM atau UBAM yang tidak terkompresi seperti yang direpresentasikan dalam SAM, berdasarkan referensi tertaut, jika tersedia.
- CRAM_ MD5up — Menghitung MD5 hash dari bagian penyelarasan dari sumber set baca CRAM yang tidak terkompresi seperti yang direpresentasikan dalam SAM, berdasarkan referensi tertaut.

> **Note**
>
> MD5 hashing dikenal rentan terhadap tabrakan. Karena itu, dua file yang berbeda mungkin memiliki yang sama ETag jika mereka diproduksi untuk mengeksploitasi tabrakan yang diketahui.

Algoritma berikut didukung untuk SHA256 keluarga. Algoritma dihitung sebagai berikut:

- FASTQ_ SHA256up - Menghitung hash SHA-256 dari sumber set baca FASTQ lengkap yang tidak terkompresi.
- BAM_ SHA256up — Menghitung hash SHA-256 dari bagian penyelarasan dari sumber kumpulan baca BAM atau UBAM yang tidak terkompresi seperti yang direpresentasikan dalam SAM, berdasarkan referensi tertaut, jika tersedia.
- CRAM_ SHA256up — Menghitung hash SHA-256 dari bagian penyelarasan dari sumber set baca CRAM yang tidak terkompresi seperti yang direpresentasikan dalam SAM, berdasarkan referensi tertaut.

Algoritma berikut didukung untuk SHA512 keluarga. Algoritma dihitung sebagai berikut:

- FASTQ_ SHA512up - Menghitung hash SHA-512 dari sumber set baca FASTQ lengkap yang tidak terkompresi.
- BAM_ SHA512up — Menghitung hash SHA-512 dari bagian penyelarasan dari sumber kumpulan baca BAM atau UBAM yang tidak terkompresi seperti yang direpresentasikan dalam SAM, berdasarkan referensi tertaut, jika tersedia.

- CRAM_ SHA512up — Menghitung hash SHA-512 dari bagian penyelarasan dari sumber set baca CRAM yang tidak terkompresi seperti yang direpresentasikan dalam SAM, berdasarkan referensi tertaut.

Membuat toko HealthOmics referensi

Penyimpanan referensi di HealthOmics adalah penyimpanan data untuk penyimpanan genom referensi. Anda dapat memiliki satu toko referensi di masing-masing Akun AWS dan Wilayah. Anda dapat membuat toko referensi menggunakan konsol atau CLI.

Topik

Membuat toko referensi menggunakan konsol

Untuk membuat toko referensi

1. Buka konsol HealthOmics .
2. Jika diperlukan, buka panel navigasi kiri (≡). Pilih Toko Referensi.
3. Pilih Genom referensi dari opsi penyimpanan data Genomics.
4. Anda dapat memilih genom referensi yang diimpor sebelumnya atau mengimpor yang baru. Jika Anda belum mengimpor genom referensi, pilih Impor genom referensi di kanan atas.
5. Pada halaman pekerjaan Buat referensi genom impor, pilih opsi Buat cepat atau Buat manual untuk membuat toko referensi, lalu berikan informasi berikut.
   - Nama genom referensi - Nama unik untuk toko ini.
   - Deskripsi (opsional) - Deskripsi toko referensi ini.
   - Peran IAM - Pilih peran dengan akses ke genom referensi Anda.
   - Referensi dari Amazon S3 - Pilih file urutan referensi Anda di bucket Amazon S3.
   - Tag (opsional) - Berikan hingga 50 tag untuk toko referensi ini.

Membuat toko referensi menggunakan CLI

Contoh berikut menunjukkan cara membuat toko referensi dengan menggunakan AWS CLI. Anda dapat memiliki satu toko referensi per AWS Wilayah.

Toko referensi mendukung penyimpanan file FASTA dengan ekstensi`.fasta`,`.fa`,`.fas`,`.fsa`,`.faa`,`.fna`,`.ffn`,`.frn`, `.mpfa`,`.seq`,`.txt`. `bgzip`Versi ekstensi ini juga didukung.

Dalam contoh berikut, ganti *`reference store name`* dengan nama yang Anda pilih untuk toko referensi Anda.

```
aws omics create-reference-store --name "reference store name"
```

Anda menerima respons JSON dengan ID dan nama toko referensi, ARN, dan stempel waktu kapan toko referensi Anda dibuat.

```
{
    "id": "3242349265",
    "arn": "arn:aws:omics:us-west-2:555555555555:referenceStore/3242349265",
    "name": "MyReferenceStore",
    "creationTime": "2022-07-01T20:58:42.878Z"
}
```

Anda dapat menggunakan ID toko referensi dalam AWS CLI perintah tambahan. Anda dapat mengambil daftar toko referensi yang IDs ditautkan ke akun Anda dengan menggunakan list-reference-storesperintah, seperti yang ditunjukkan pada contoh berikut.

```
aws omics list-reference-stores
```

Sebagai tanggapan, Anda menerima nama toko referensi yang baru Anda buat.

```
{
    "referenceStores": [
        {
              "id": "3242349265",
              "arn": "arn:aws:omics:us-west-2:555555555555:referenceStore/3242349265",
              "name": "MyReferenceStore",
             "creationTime": "2022-07-01T20:58:42.878Z"
         }
```

```
    ]
}
```

Setelah Anda membuat toko referensi, Anda dapat membuat pekerjaan impor untuk memuat file referensi genom ke dalamnya. Untuk melakukannya, Anda harus menggunakan atau membuat peran IAM untuk mengakses data. Berikut ini adalah contoh kebijakan .

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/*"
            ]
        }
    ]
}
```

Anda juga harus memiliki kebijakan kepercayaan yang mirip dengan contoh berikut.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": [
                   "omics.amazonaws.com"
                ]
```

```
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

Anda sekarang dapat mengimpor genom referensi. Contoh ini menggunakan Genome Reference Consortium Human Build 38 (hg38), yang merupakan akses terbuka dan tersedia dari Registry of Open Data on. AWS Bucket yang menampung data ini berbasis di US East (Ohio). Untuk menggunakan bucket di AWS Wilayah lain, Anda dapat menyalin data ke bucket Amazon S3 yang dihosting di Wilayah Anda. Gunakan AWS CLI perintah berikut untuk menyalin genom ke bucket Amazon S3 Anda.

```
aws s3 cp s3://broad-references/hg38/v0/Homo_sapiens_assembly38.fasta s3://amzn-s3-
demo-bucket
```

Anda kemudian dapat memulai pekerjaan impor Anda. Ganti*reference store ID*,*role ARN*, dan *source file path* dengan masukan Anda sendiri.

```
aws omics start-reference-import-job --reference-store-id reference store ID --role-
arn role ARN --sources source file path
```

Setelah data diimpor, Anda menerima respons berikut di JSON.

```
{
        "id": "7252016478",
        "referenceStoreId": "3242349265",
        "roleArn": "arn:aws:iam::111122223333:role/OmicsReferenceImport",
        "status": "CREATED",
        "creationTime": "2022-07-01T21:15:13.727Z"
}
```

Anda dapat memantau status pekerjaan dengan menggunakan perintah berikut. Dalam contoh berikut, ganti *reference store ID* dan *job ID* dengan ID toko referensi Anda dan ID pekerjaan yang ingin Anda pelajari lebih lanjut.

```
aws omics get-reference-import-job --reference-store-id reference store ID --id job ID
```

Sebagai tanggapan, Anda menerima tanggapan dengan detail untuk toko referensi itu dan statusnya.

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

Anda juga dapat menemukan referensi yang diimpor dengan mencantumkan referensi Anda dan memfilternya berdasarkan nama referensi. Ganti *reference store ID* dengan ID toko referensi Anda, dan tambahkan filter opsional untuk mempersempit daftar.

```
aws omics list-references --reference-store-id reference store ID --filter
 name=MyReference
```

Sebagai tanggapan, Anda menerima informasi berikut.

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

Untuk mempelajari lebih lanjut tentang metadata referensi, gunakan operasi get-reference-metadataAPI. Dalam contoh berikut, ganti *reference store ID* dengan ID toko referensi Anda dan *reference ID* dengan ID referensi yang ingin Anda pelajari lebih lanjut.

```
aws omics get-reference-metadata --reference-store-id reference store ID --id reference
 ID
```

Anda menerima informasi berikut sebagai tanggapan.

```
{
    "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
        }
    }
}
```

Anda juga dapat mengunduh bagian dari file referensi dengan menggunakan get-reference. Dalam contoh berikut, ganti *reference store ID* dengan ID toko referensi Anda dan *reference ID* dengan ID referensi yang ingin Anda unduh.

```
aws omics get-reference --reference-store-id reference store ID --id reference ID --
part-number 1 outfile.fa
```

Membuat toko HealthOmics urutan

HealthOmics penyimpanan urutan mendukung penyimpanan file genom dalam format tidak selaras `FASTQ` (hanya gzip) dan. `uBAM` Ini juga mendukung format selaras `BAM` dan`CRAM`.

File yang diimpor disimpan sebagai set baca. Anda dapat menambahkan tag ke set baca dan menggunakan kebijakan IAM untuk mengontrol akses ke set baca. Set baca yang selaras memerlukan genom referensi untuk menyelaraskan urutan genom, tetapi ini opsional untuk set baca yang tidak selaras.

Untuk menyimpan set baca, pertama-tama Anda membuat toko urutan. Saat membuat penyimpanan urutan, Anda dapat menentukan bucket Amazon S3 opsional sebagai lokasi fallback dan lokasi penyimpanan log akses S3. Lokasi fallback digunakan untuk menyimpan file apa pun yang gagal membuat set baca selama unggahan langsung. Lokasi fallback tersedia untuk toko urutan yang dibuat setelah 15 Mei 2023. Anda menentukan lokasi fallback saat Anda membuat toko urutan.

Anda dapat menentukan hingga lima tombol tag set baca. Saat Anda membuat atau memperbarui set baca dengan kunci tag yang cocok dengan salah satu kunci ini, tag set baca disebarkan ke objek Amazon S3 yang sesuai. Tag sistem yang dibuat oleh HealthOmics disebarkan secara default.

Topik

Membuat toko urutan menggunakan konsol

Untuk membuat toko urutan

1. Buka konsol HealthOmics .
2. Jika diperlukan, buka panel navigasi kiri (≡). Pilih toko Sequence.
3. Pada halaman Create sequence store, berikan informasi berikut

    - Nama toko urutan - Nama unik untuk toko ini.

- Deskripsi (opsional) - Deskripsi toko urutan ini.

4. Untuk lokasi Fallback di S3, tentukan lokasi Amazon S3. HealthOmics menggunakan lokasi fallback untuk menyimpan file apa pun yang gagal membuat set baca selama unggahan langsung. Anda harus memberikan akses tulis HealthOmics layanan ke lokasi fallback Amazon S3. Untuk contoh kebijakan, lihat Konfigurasikan lokasi fallback.

   Lokasi fallback tidak tersedia untuk toko urutan yang dibuat sebelum 16 Mei 2023.

5. (Opsional) Untuk tombol tag set Baca untuk propagasi S3, Anda dapat memasukkan hingga lima tombol set baca untuk disebarkan dari set baca ke Objek S3 yang mendasarinya. Dengan menyebarkan tag dari set baca ke objek S3, Anda dapat memberikan izin akses S3 berdasarkan pengguna and/or akhir tag untuk melihat tag yang disebarkan melalui operasi Amazon S3 API. getObjectTagging

   a. Masukkan satu nilai kunci di kotak teks. Konsol membuat kotak teks baru untuk menambahkan kunci berikutnya.

   b. (Opsional) Pilih Hapus untuk menghapus semua tombol.

6. Di bawah Enkripsi Data, pilih apakah Anda ingin enkripsi data dimiliki dan dikelola oleh AWS atau menggunakan CMK yang dikelola pelanggan.

7. (Opsional) Di bawah Akses data S3, pilih apakah akan membuat peran dan kebijakan baru untuk mengakses penyimpanan urutan melalui Amazon S3.

8. (Opsional) Untuk pencatatan akses S3, pilih `Enabled` apakah Anda ingin Amazon S3 mengumpulkan catatan log akses.

   Untuk lokasi logging Access di S3, tentukan lokasi Amazon S3 untuk menyimpan log. Bidang ini hanya terlihat jika Anda mengaktifkan pencatatan akses S3.

9. Tag (opsional) - Berikan hingga 50 tag untuk toko urutan ini. Tag ini terpisah dari tag set baca yang disetel selama import/tag pembaruan set baca

Setelah Anda membuat toko, itu siap untukMengimpor file genom.

Membuat toko urutan menggunakan CLI

Dalam contoh berikut, ganti *`sequence store name`* dengan nama yang Anda pilih untuk toko urutan Anda.

```
aws omics create-sequence-store --name sequence store name --fallback-location "s3://
amzn-s3-demo-bucket"
```

Anda menerima respons berikut di JSON, yang menyertakan nomor ID untuk penyimpanan urutan yang baru dibuat.

```
{
    "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"
}
```

Anda juga dapat melihat semua toko urutan yang terkait dengan akun Anda dengan menggunakan list-sequence-storesperintah, seperti yang ditunjukkan pada berikut ini.

```
aws omics list-sequence-stores
```

Anda menerima tanggapan berikut.

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

Anda dapat menggunakan get-sequence-storeuntuk mempelajari lebih lanjut tentang penyimpanan urutan dengan menggunakan ID-nya, seperti yang ditunjukkan pada contoh berikut:

```
aws omics get-sequence-store --id sequence store ID
```

Anda menerima tanggapan berikut:


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


Setelah pembuatan, beberapa parameter toko juga dapat diperbarui. Ini dapat dilakukan melalui Console atau `updateSequenceStore` operasi API.

Memperbarui toko urutan

Untuk memperbarui toko urutan, ikuti langkah-langkah ini:

1. Buka konsol HealthOmics .
2. Jika diperlukan, buka panel navigasi kiri (≡). Pilih toko Sequence.
3. Pilih toko urutan untuk diperbarui.
4. Di panel Detail, pilih Edit.
5. Pada halaman Edit detail, Anda dapat memperbarui bidang berikut:

    - Nama toko urutan - Nama unik untuk toko ini.

- Deskripsi - Deskripsi toko urutan ini.
- Lokasi mundur di S3, tentukan lokasi Amazon S3. HealthOmics menggunakan lokasi fallback untuk menyimpan file apa pun yang gagal membuat set baca selama unggahan langsung.
- Baca tombol tag set untuk propagasi S3 Anda dapat memasukkan hingga lima tombol set baca untuk disebarkan ke Amazon S3.
- (Opsional) Untuk pencatatan akses S3, pilih `Enabled` apakah Anda ingin Amazon S3 mengumpulkan catatan log akses.

  Untuk lokasi logging Access di S3, tentukan lokasi Amazon S3 untuk menyimpan log. Bidang ini hanya terlihat jika Anda mengaktifkan pencatatan akses S3.
- Tag (opsional) - Berikan hingga 50 tag untuk toko urutan ini.

Memperbarui tag set baca untuk penyimpanan urutan

Untuk memperbarui tag set baca atau bidang lain untuk penyimpanan urutan, ikuti langkah-langkah berikut:

1. Buka konsol HealthOmics .
2. Jika diperlukan, buka panel navigasi kiri (≡). Pilih toko Sequence.
3. Pilih toko urutan yang ingin Anda perbarui.
4. Pilih tab Detail.
5. Pilih Edit.
6. Tambahkan tag set baca baru atau hapus tag yang ada, sesuai kebutuhan.
7. Perbarui nama, deskripsi, lokasi fallback, atau akses data S3, sesuai kebutuhan.
8. Pilih Simpan perubahan.

Mengimpor file genom

Untuk mengimpor file genom ke penyimpanan urutan, ikuti langkah-langkah ini:

Untuk mengimpor file genomik

1. Buka konsol HealthOmics .
2. Jika diperlukan, buka panel navigasi kiri (≡). Pilih pilih toko Urutan.

3. Pada halaman Sequence store, pilih toko urutan yang ingin Anda impor file Anda.
4. Pada halaman penyimpanan urutan individual, pilih Impor file genom.
5. Pada halaman Tentukan detail impor, berikan informasi berikut

   - Peran IAM - Peran IAM yang dapat mengakses file genom di Amazon S3.
   - Genom referensi - Genom referensi untuk data genomik ini.
6. Pada halaman Tentukan manifes impor, tentukan informasi berikut File manifes. File manifes adalah file JSON atau YAMG yang menjelaskan informasi penting dari data genomik Anda. Untuk informasi tentang file manifes, lihatMengimpor set baca ke toko HealthOmics urutan.
7. Klik Buat pekerjaan impor.

Menghapus HealthOmics referensi dan toko urutan

Penyimpanan referensi dan urutan dapat dihapus. Toko urutan hanya dapat dihapus jika tidak berisi set baca, dan toko referensi hanya dapat dihapus jika tidak berisi referensi. Menghapus urutan atau toko referensi juga menghapus tag apa pun yang terkait dengan toko itu.

Contoh berikut menunjukkan cara menghapus toko referensi dengan menggunakan AWS CLI. Jika tindakan berhasil, Anda tidak akan menerima tanggapan. Dalam contoh berikut, ganti *reference store ID* dengan ID toko referensi Anda.

```
aws omics delete-reference-store --id reference store ID
```

Contoh berikut menunjukkan cara menghapus toko urutan. Anda tidak menerima tanggapan jika tindakan berhasil. Dalam contoh berikut, ganti *sequence store ID* dengan ID toko urutan Anda.

```
aws omics delete-sequence-store --id sequence store ID
```

Anda juga dapat menghapus referensi di toko referensi seperti yang ditunjukkan pada contoh berikut. Referensi hanya dapat dihapus jika tidak digunakan dalam set baca, toko varian, atau penyimpanan anotasi. Dalam contoh berikut, ganti *reference store ID* dengan ID toko referensi Anda, dan ganti *reference ID* dengan ID untuk referensi yang ingin Anda hapus.

```
aws omics delete-reference  --id reference ID --reference-store-id reference store ID
```

Mengimpor set baca ke toko HealthOmics urutan

Setelah Anda membuat toko urutan, buat pekerjaan impor untuk mengunggah set baca ke penyimpanan data. Anda dapat mengunggah file dari bucket Amazon S3, atau Anda dapat mengunggah langsung menggunakan operasi API sinkron. Bucket Amazon S3 Anda harus berada di Wilayah yang sama dengan toko urutan Anda.

Anda dapat mengunggah kombinasi set baca yang selaras dan tidak selaras ke dalam penyimpanan urutan Anda, namun, jika ada set baca dalam impor Anda yang selaras, Anda harus menyertakan genom referensi.

Anda dapat menggunakan kembali kebijakan akses IAM yang Anda gunakan untuk membuat toko Referensi.

Topik berikut menjelaskan langkah-langkah utama yang Anda ikuti untuk mengimpor set baca ke dalam penyimpanan urutan Anda dan kemudian mendapatkan informasi tentang data yang diimpor.

Topik

Unggah file ke Amazon S3

Contoh berikut menunjukkan cara memindahkan file ke bucket Amazon S3 Anda.

```
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
```

Sampel BAM dan CRAM digunakan dalam contoh ini memerlukan referensi genom yang berbeda, Hg19 danHg38. Untuk mempelajari lebih lanjut atau mengakses referensi ini, lihat Referensi Genom Luas di Registri Data Terbuka di AWS.

Membuat file manifes

Anda juga harus membuat file manifes di JSON untuk memodelkan pekerjaan impor `import.json` (lihat contoh berikut). Jika Anda membuat penyimpanan urutan di konsol, Anda tidak perlu menentukan `sequenceStoreId` ataurole`ARN`, jadi file manifes Anda dimulai dengan `sources` input.

API manifest

Contoh berikut mengimpor tiga set baca dengan menggunakan API: satu`FASTQ`, satu`BAM`, dan satu`CRAM`.

```
{
  "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

Kode contoh ini digunakan untuk mengimpor satu set baca dengan menggunakan konsol.

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

Atau, Anda dapat mengunggah file manifes dalam format YAMAL.

Memulai pekerjaan impor

Untuk memulai pekerjaan impor, gunakan AWS CLI perintah berikut.

```
aws omics start-read-set-import-job --cli-input-json file://import.json
```

Anda menerima tanggapan berikut, yang menunjukkan penciptaan lapangan kerja yang sukses.

```
{
  "id": "3660451514",
  "sequenceStoreId": "3936421177",
  "roleArn": "arn:aws:iam::111122223333:role/OmicsImport",
  "status": "CREATED",
  "creationTime": "2022-07-13T22:14:59.309Z"
}
```

Pantau pekerjaan impor

Setelah pekerjaan impor dimulai, Anda dapat memantau kemajuannya dengan perintah berikut. Dalam contoh berikut, ganti *`sequence store id`* dengan ID penyimpanan urutan Anda, dan ganti *`job import ID`* dengan ID impor.

```
aws omics get-read-set-import-job --sequence-store-id sequence store id --id job import
 ID
```

Berikut ini menunjukkan status untuk semua pekerjaan impor yang terkait dengan ID penyimpanan urutan yang ditentukan.

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

Temukan file urutan yang diimpor

Setelah pekerjaan selesai, Anda dapat menggunakan operasi list-read-setsAPI untuk menemukan file urutan yang diimpor. Dalam contoh berikut, ganti *sequence store id* dengan ID toko urutan Anda.

```
aws omics list-read-sets --sequence-store-id sequence store id
```

Anda menerima tanggapan berikut.


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

Dapatkan detail tentang set baca

Untuk melihat detail selengkapnya tentang set baca, gunakan operasi GetReadSetMetadataAPI. Dalam contoh berikut, ganti *`sequence store id`* dengan ID penyimpanan urutan Anda, dan ganti *`read set id`* dengan ID set baca Anda.

```
aws omics get-read-set-metadata --sequence-store-id sequence store id --id read set id
```

Anda menerima tanggapan berikut.

```
{
"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"
}
```

Unduh file data set baca

Anda dapat mengakses objek untuk set baca aktif menggunakan operasi Amazon S3 GetObject API. URI untuk objek dikembalikan dalam respons GetReadSetMetadataAPI. Untuk informasi selengkapnya, lihat Mengakses set HealthOmics baca dengan Amazon S3 URIs.

Atau, gunakan operasi HealthOmics GetReadSet API. Anda dapat menggunakan GetReadSet untuk mengunduh secara paralel dengan mengunduh bagian-bagian individual. Bagian-bagian ini mirip dengan bagian Amazon S3. Berikut ini adalah contoh cara mengunduh bagian 1 dari set baca. Dalam contoh berikut, ganti *sequence store id* dengan ID penyimpanan urutan Anda, dan ganti *read set id* dengan ID set baca Anda.

```
aws omics get-read-set --sequence-store-id sequence store id --id read set id  --part-
number 1 outfile.bam
```

Anda juga dapat menggunakan HealthOmics Transfer Manager untuk mengunduh file untuk HealthOmics referensi atau set baca. Anda dapat mengunduh HealthOmics Transfer Manager di sini. Untuk informasi selengkapnya tentang menggunakan dan menyiapkan Manajer Transfer, lihat GitHubRepositori ini.

Unggah langsung ke toko HealthOmics urutan

Kami menyarankan Anda menggunakan Manajer HealthOmics Transfer untuk menambahkan file ke toko urutan Anda. Untuk informasi selengkapnya tentang menggunakan Manajer Transfer, lihat GitHubRepositori ini. Anda juga dapat mengunggah set baca langsung ke toko urutan melalui operasi API unggahan langsung.

Set baca unggahan langsung ada terlebih dahulu dalam `PROCESSING_UPLOAD` status. Ini berarti bahwa bagian file saat ini sedang diunggah, dan Anda dapat mengakses metadata set baca. Setelah bagian diunggah dan checksum divalidasi, set baca menjadi `ACTIVE` dan berperilaku sama seperti set baca yang diimpor.

Jika unggahan langsung gagal, status set baca ditampilkan sebagai`UPLOAD_FAILED`. Anda dapat mengonfigurasi bucket Amazon S3 sebagai lokasi fallback untuk file yang gagal diunggah. Lokasi fallback tersedia untuk toko urutan yang dibuat setelah 15 Mei 2023.

Topik

- Unggah langsung ke toko urutan menggunakan AWS CLI

- Konfigurasikan lokasi fallback

Unggah langsung ke toko urutan menggunakan AWS CLI

Untuk memulai, mulai upload multipart. Anda dapat melakukan ini dengan menggunakan AWS CLI, seperti yang ditunjukkan pada contoh berikut.

Untuk mengunggah langsung menggunakan AWS CLI perintah

1. Buat bagian dengan memisahkan data Anda, seperti yang ditunjukkan pada contoh berikut.

```
split -b 100MiB SRR233106_1.filt.fastq.gz source1_part_
```

2. Setelah file sumber Anda berada di beberapa bagian, buat unggahan set baca multibagian, seperti yang ditunjukkan pada contoh berikut. Ganti *sequence store ID* dan parameter lainnya dengan ID penyimpanan urutan Anda dan nilai lainnya.

```
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"
```

Anda mendapatkan metadata `uploadID` dan lainnya dalam respons. Gunakan `uploadID` untuk langkah selanjutnya dari proses pengunggahan.

```
{
"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. Tambahkan set baca Anda ke unggahan. Jika file Anda cukup kecil, Anda hanya perlu melakukan langkah ini sekali. Untuk file yang lebih besar, Anda melakukan langkah ini untuk setiap bagian file Anda. Jika Anda mengunggah bagian baru dengan menggunakan nomor bagian yang sebelumnya digunakan, bagian tersebut akan menimpa bagian yang diunggah sebelumnya.

   Dalam contoh berikut, ganti*sequence store ID*,*upload ID*, dan parameter lainnya dengan nilai-nilai Anda.

```
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
```

   Responsnya adalah ID yang dapat Anda gunakan untuk memverifikasi bahwa file yang diunggah cocok dengan file yang Anda inginkan.

```
{
"checksum": "984979b9928ae8d8622286c4a9cd8e99d964a22d59ed0f5722e1733eb280e635"
}
```

4. Lanjutkan mengunggah bagian-bagian file Anda, jika perlu. Untuk memverifikasi bahwa set baca Anda telah diunggah, gunakan operasi API list-read-set-upload-parts, seperti yang ditunjukkan pada berikut ini. Dalam contoh berikut, ganti*sequence store ID* ,*upload ID*, dan *part source* dengan masukan Anda sendiri.

```
aws omics list-read-set-upload-parts \
 --sequence-store-id sequence store ID \
 --upload-id upload ID \
 --part-source SOURCE1
```

   Respons mengembalikan jumlah set baca, ukuran, dan stempel waktu saat terbaru diperbarui.

```
{
"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. Untuk melihat semua unggahan set baca multibagian aktif, gunakan list-multipart-read-set-upload, seperti yang ditunjukkan pada berikut ini. Ganti *sequence store ID* dengan ID untuk toko urutan Anda sendiri.

```
aws omics list-multipart-read-set-uploads --sequence-store-id
            sequence store ID
```

API ini hanya mengembalikan unggahan set baca multibagian yang sedang berlangsung. Setelah set baca yang dicernaACTIVE, atau jika unggahan gagal, unggahan tidak akan dikembalikan sebagai respons terhadap list-multipart-read-set-uploads API. Untuk melihat set baca aktif, gunakan list-read-setsAPI. Contoh respon untuk list-multipart-read-set-upload ditampilkan di berikut ini.

```
{
"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. Setelah Anda mengunggah semua bagian file Anda, gunakan complete-multipart-read-set-upload untuk mengakhiri proses upload, seperti yang ditunjukkan pada contoh berikut. Ganti *sequence store IDupload ID*,, dan parameter untuk bagian dengan nilai Anda sendiri.

```
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"}]'
```

Respons untuk complete-multipart-read-set-upload adalah set baca IDs untuk set baca yang Anda impor.

```
{
"readSetId": "0000000001"
}
```

7. Untuk menghentikan unggahan, gunakan abort-multipart-read-set-upload dengan ID upload untuk mengakhiri proses upload. Ganti *sequence store ID* dan *upload ID* dengan nilai parameter Anda sendiri.

```
aws omics abort-multipart-read-set-upload \
--sequence-store-id sequence store ID \
--upload-id upload ID
```

8. Setelah unggahan selesai, ambil data Anda dari set baca dengan menggunakan get-read-set, seperti yang ditunjukkan pada berikut ini. Jika unggahan masih diproses, get-read-setmengembalikan metadata terbatas, dan file indeks yang dihasilkan tidak tersedia. Ganti *sequence store ID* dan parameter lainnya dengan input Anda sendiri.

```
aws omics get-read-set
 --sequence-store-id sequence store ID \
 --id read set ID \
 --file SOURCE1 \
 --part-number 1 myfile.fastq.gz
```

9. Untuk memeriksa metadata, termasuk status unggahan Anda, gunakan operasi get-read-set-metadataAPI.

```
aws omics get-read-set-metadata --sequence-store-id sequence store ID --id read set
 ID
```

Responsnya mencakup detail metadata seperti jenis file, ARN referensi, jumlah file, dan panjang urutan. Ini juga termasuk status. Status yang mungkin adalah`PROCESSING_UPLOAD`,`ACTIVE`, dan`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"
}
```

Konfigurasikan lokasi fallback

Saat membuat atau memperbarui penyimpanan urutan, Anda dapat mengonfigurasi bucket Amazon S3 sebagai lokasi fallback untuk file yang gagal diunggah. Bagian file untuk set baca tersebut ditransfer ke lokasi fallback. Lokasi fallback tersedia untuk toko urutan yang dibuat setelah 15 Mei 2023.

Buat kebijakan bucket Amazon S3 untuk memberikan akses HealthOmics tulis ke lokasi fallback Amazon S3, seperti yang ditunjukkan pada contoh berikut:

```
{
    "Effect": "Allow",
```

```
    "Principal": {
        "Service": "omics.amazonaws.com"
    },
    "Action": "s3:PutObject",
    "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
}
```

Jika bucket Amazon S3 untuk fallback atau log akses menggunakan kunci terkelola pelanggan, tambahkan izin berikut ke kebijakan kunci:

```
 {
    "Sid": "Allow use of key",
    "Effect": "Allow",
    "Principal": {
        "Service": "omics.amazonaws.com"
    },
    "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey*"
    ],
    "Resource": "*"
}
```

Mengekspor set HealthOmics baca ke bucket Amazon S3

Anda dapat mengekspor set baca sebagai pekerjaan ekspor batch ke bucket Amazon S3. Untuk melakukannya, pertama-tama buat kebijakan IAM yang memiliki akses tulis ke bucket, mirip dengan contoh kebijakan IAM berikut.

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"
  }
]
}
```

Setelah kebijakan IAM diberlakukan, mulailah pekerjaan ekspor set baca Anda. Contoh berikut menunjukkan cara melakukannya dengan menggunakan operasi start-read-set-export-job API. Dalam contoh berikut, ganti semua parameter, seperti*sequence store ID*,,*destination*, dan *role ARNsources*, dengan masukan Anda.

```
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
```

Anda menerima respons berikut dengan informasi tentang toko urutan asal dan bucket Amazon S3 tujuan.

```
{
```

```
"id": <job-id>,
"sequenceStoreId": <sequence-store-id>,
"destination": <destination-s3-uri>,
"status": "SUBMITTED",
"creationTime": "2022-10-22T01:33:38.079000+00:00"
}
```

Setelah pekerjaan dimulai, Anda dapat menentukan statusnya dengan menggunakan operasi get-read-set-export-job API, seperti yang ditunjukkan pada berikut ini. Ganti *sequence store ID* dan *job ID* dengan ID toko urutan dan ID pekerjaan Anda, masing-masing.

```
aws omics get-read-set-export-job --id job-id --sequence-store-id sequence store ID
```

Anda dapat melihat semua pekerjaan ekspor yang diinisialisasi untuk penyimpanan urutan dengan menggunakan operasi API list-read-set-export-jobs, seperti yang ditunjukkan di bawah ini. Ganti *sequence store ID* dengan ID toko urutan Anda.

```
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"
  }
]
}
```

Selain mengekspor set baca Anda, Anda juga dapat membagikannya dengan menggunakan akses Amazon URIs S3. Untuk mempelajari informasi lebih lanjut, lihat Mengakses set HealthOmics baca dengan Amazon S3 URIs.

Mengakses set HealthOmics baca dengan Amazon S3 URIs

Anda dapat menggunakan jalur URI Amazon S3 untuk mengakses set baca penyimpanan urutan aktif Anda.

Dengan jalur URI Amazon S3, Anda dapat menggunakan operasi Amazon S3 untuk membuat daftar, berbagi, dan mengunduh set baca Anda. Akses melalui S3 APIs mempercepat kolaborasi dan integrasi alat mengingat banyak alat industri sudah dibangun untuk dibaca dari S3. Selain itu, Anda dapat berbagi akses ke S3 APIs dengan akun lain dan memberikan akses baca lintas wilayah ke data.

HealthOmics tidak mendukung akses URI Amazon S3 ke set baca yang diarsipkan. Ketika Anda mengaktifkan set baca, itu dikembalikan ke jalur URI yang sama setiap kali.

Dengan data yang dimuat ke HealthOmics toko, karena URI Amazon S3 didasarkan pada titik akses Amazon S3, Anda dapat langsung berintegrasi dengan alat standar industri yang membaca Amazon S3, seperti berikut URIs ini:

- Aplikasi analisis visual seperti Integrative Genomics Viewer (IGV) atau UCSC Genome Browser.
- Alur kerja umum dengan ekstensi Amazon S3 seperti CWL, WDL, dan Nextflow.
- Alat apa pun yang dapat mengautentikasi dan membaca dari titik akses Amazon URIs S3 atau membaca Amazon S3 yang telah ditetapkan sebelumnya. URIs
- Utilitas Amazon S3 seperti Mountpoint atau. CloudFront

Amazon S3 Mountpoint memungkinkan Anda menggunakan bucket Amazon S3 sebagai sistem file lokal. Untuk mempelajari lebih lanjut tentang Mountpoint dan menginstalnya untuk digunakan, lihat Mountpoint untuk Amazon S3.

Amazon CloudFront adalah layanan jaringan pengiriman konten (CDN) yang dibuat untuk kinerja tinggi, keamanan, dan kenyamanan pengembang. Untuk mempelajari lebih lanjut tentang menggunakan Amazon CloudFront, lihat CloudFront dokumentasi Amazon. Untuk mengatur CloudFront dengan toko urutan, hubungi AWS HealthOmics tim.

Akun root pemilik data diaktifkan untuk tindakan S3:GetObject, S3:GetObjectTagging, dan S3:List Bucket pada awalan penyimpanan urutan. Agar pengguna di akun dapat mengakses data, Anda membuat kebijakan IAM dan melampirkannya ke pengguna atau peran. Untuk contoh kebijakan, lihat Izin untuk akses data menggunakan Amazon S3 URIs.

Anda dapat menggunakan operasi Amazon S3 API berikut pada set baca aktif untuk mencantumkan dan mengambil data Anda. Anda dapat mengakses set baca yang diarsipkan melalui Amazon URIs S3 setelah diaktifkan.

- GetObject— Mengambil objek dari Amazon S3.

- HeadObject— Operasi HEAD mengambil metadata dari objek tanpa mengembalikan objek itu sendiri. Operasi ini berguna jika Anda hanya menginginkan metadata objek.
- ListObjects dan ListObject v2 - Mengembalikan beberapa atau semua (hingga 1.000) objek dalam ember.
- CopyObject— Membuat salinan objek yang sudah disimpan di Amazon S3. HealthOmicsmendukung penyalinan ke jalur akses Amazon S3, tetapi tidak menulis ke titik akses.

HealthOmics toko urutan mempertahankan identitas semantik file melalui. ETags Sepanjang siklus hidup file, Amazon ETag S3, yang didasarkan pada identitas bitwise, dapat berubah, HealthOmics ETag namun tetap sama. Untuk mempelajari selengkapnya, lihat HealthOmics ETags dan asal-usul data.

Topik

- Struktur URI Amazon S3 dalam penyimpanan HealthOmics
- Menggunakan IGV yang Dihosting atau Lokal untuk mengakses set baca
- Menggunakan Samtools atau HTSlib di HealthOmics
- Menggunakan Mountpoint HealthOmics
- Menggunakan CloudFront dengan HealthOmics

Struktur URI Amazon S3 dalam penyimpanan HealthOmics

Semua file dengan Amazon S3 URIs memiliki `omics:subjectId` dan tag `omics:sampleId` sumber daya. Anda dapat menggunakan tag ini untuk berbagi akses dengan menggunakan kebijakan IAM melalui pola seperti`"s3:ExistingObjectTag/omics:subjectId": "pattern desired"`.

Struktur file adalah sebagai berikut:

`.../`*`account_id`*`/sequenceStore/`*`seq_store_id`*`/readSet/`*`read_set_id`*`/`*`files`*`.`

Untuk file yang diimpor ke toko urutan dari Amazon S3, toko urutan mencoba mempertahankan nama sumber asli. Ketika nama bertentangan, sistem menambahkan informasi set baca untuk memastikan bahwa nama file unik. Misalnya, untuk set baca fastq, jika kedua nama file sama, untuk membuat nama unik, dimasukkan sebelum .fastq.gz `sourceX` atau.fq.gz. Untuk upload langsung, nama file mengikuti pola berikut:

- Untuk FASTQ— *read_set_name* _ .fastq.gz *sourcex*
- Untuk uBAM/BAM/CRAM —*read_set_name*. *file extension*dengan ekstensi `.bam` atau `.cram`. Contohnya adalah `NA193948.bam`.

Untuk set baca yang BAM atau CRAM, file indeks secara otomatis dihasilkan selama proses konsumsi. Untuk file indeks yang dihasilkan, ekstensi indeks yang tepat di akhir nama file diterapkan. Ini memiliki pola *<name of the Source the index is on>.<file index extension>.* Ekstensi indeks adalah `.bai` atau `.crai`.

Menggunakan IGV yang Dihosting atau Lokal untuk mengakses set baca

IGV adalah browser genom yang digunakan untuk menganalisis file BAM dan CRAM. Ini membutuhkan file dan indeks karena hanya menampilkan sebagian genom pada satu waktu. IGV dapat diunduh dan digunakan secara lokal, dan ada panduan untuk membuat IGV yang dihosting AWS. Versi web publik tidak didukung karena membutuhkan CORS.

IGV lokal bergantung pada AWS konfigurasi lokal untuk mengakses file. Pastikan peran yang digunakan dalam konfigurasi tersebut memiliki kebijakan yang dilampirkan yang memungkinkan GetObject izin kms: Dekripsi dan s3: ke URI s3 dari kumpulan baca yang diakses. Setelah itu, di IGV, Anda dapat menggunakan “File> load from URL” dan paste di URI untuk sumber dan indeks. Atau, presigned URLs dapat dibuat dan digunakan dengan cara yang sama, yang akan melewati konfigurasi AWS. Perhatikan bahwa CORS tidak didukung dengan akses Amazon S3 URI, jadi permintaan yang mengandalkan CORS tidak didukung.

Contoh AWS Hosted IGV bergantung pada AWS Cognito untuk membuat konfigurasi dan izin yang benar di dalam lingkungan. Pastikan kebijakan dibuat dengan izin enableSKMS:Decrypt dan s3: GetObject ke URI Amazon S3 dari set baca yang sedang diakses, dan tambahkan kebijakan ini ke peran yang ditetapkan ke kumpulan pengguna Cognito. Setelah itu, di IGV, Anda dapat menggunakan “File> load from URL” dan masukkan URI untuk sumber dan indeks. Atau, presigned URLs dapat dibuat dan digunakan dengan cara yang sama, yang melewati konfigurasi AWS.

Perhatikan bahwa toko urutan tidak akan muncul di bawah tab “Amazon” karena itu hanya menampilkan bucket yang Anda miliki di Wilayah tempat AWS profil dikonfigurasi.

Menggunakan Samtools atau HTSlib di HealthOmics

HTSlib adalah pustaka inti yang dibagikan oleh beberapa alat seperti Samtools, RSAMTools PySam, dan lainnya. Gunakan HTSlib versi 1.20 atau yang lebih baru untuk mendapatkan dukungan tanpa

batas untuk Poin Akses Amazon S3. Untuk versi HTSlib pustaka yang lebih lama, Anda dapat menggunakan solusi berikut:

- Tetapkan variabel lingkungan untuk host HTS Amazon S3 dengan:. `export HTS_S3_HOST="s3.region.amazonaws.com"`
- Hasilkan URL presigned untuk file yang ingin Anda gunakan. Jika BAM atau CRAM sedang digunakan, pastikan bahwa URL presigned dihasilkan untuk file dan indeks. Setelah itu, kedua file dapat digunakan dengan perpustakaan.
- Gunakan Mountpoint untuk memasang urutan penyimpanan atau membaca awalan set di lingkungan yang sama tempat Anda menggunakan pustaka. HTSlib Dari sini, file dapat diakses dengan menggunakan jalur file lokal.

Menggunakan Mountpoint HealthOmics

Mountpoint untuk Amazon S3 adalah klien file throughput tinggi yang sederhana untuk memasang bucket Amazon S3 sebagai sistem file lokal. Dengan Mountpoint untuk Amazon S3, aplikasi Anda dapat mengakses objek yang disimpan di Amazon S3 melalui operasi file seperti buka dan baca. Mountpoint untuk Amazon S3 secara otomatis menerjemahkan operasi ini ke dalam panggilan API objek Amazon S3, memberikan aplikasi Anda akses ke penyimpanan elastis dan throughput Amazon S3 melalui antarmuka file.

Mountpoint dapat diinstal dengan menggunakan petunjuk instalasi Mountpoint. Mountpoint menggunakan Profil AWS yang bersifat lokal untuk penginstalan dan berfungsi pada tingkat awalan Amazon S3. Pastikan profil yang digunakan memiliki kebijakan yang memungkinkan izin s3:GetObject, s3:ListBucket, dan kms: Dekripsi ke awalan URI Amazon S3 dari kumpulan baca atau penyimpanan urutan yang diakses. Setelah itu, ember dapat dipasang dengan menggunakan jalur berikut:

```
mount-s3 access point arn local path to mount --prefix prefix to sequence store or read
 set --region region
```

Menggunakan CloudFront dengan HealthOmics

Amazon CloudFront adalah layanan jaringan pengiriman konten (CDN) yang dibuat untuk kinerja tinggi, keamanan, dan kenyamanan pengembang. Pelanggan yang ingin menggunakan CloudFront harus bekerja dengan tim Layanan untuk mengaktifkan CloudFront distribusi. Bekerja dengan tim akun Anda untuk melibatkan tim HealthOmics layanan.

Mengaktifkan set baca di HealthOmics

Anda dapat mengaktifkan set baca yang diarsipkan dengan operasi start-read-set-activation-job API, atau melalui AWS CLI, seperti yang ditunjukkan pada contoh berikut. Ganti *sequence store ID* dan *read set id* dengan ID toko urutan Anda dan set baca 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
```

Anda menerima tanggapan yang berisi informasi pekerjaan aktivasi, seperti yang ditunjukkan di bawah ini.

```
{
    "id": "12345678",
    "sequenceStoreId": "1234567890",
    "status": "SUBMITTED",
    "creationTime": "2022-10-22T00:50:54.670000+00:00"
}
```

Setelah pekerjaan aktivasi dimulai, Anda dapat memantau kemajuannya dengan operasi API get-read-set-activation-job. Berikut ini adalah contoh cara menggunakan AWS CLI untuk memeriksa status pekerjaan aktivasi Anda. Ganti *job ID* dan *sequence store ID* dengan ID toko urutan dan pekerjaan Anda IDs, masing-masing.

```
aws omics get-read-set-activation-job --id job ID --sequence-store-id sequence store ID
```

Respons merangkum pekerjaan aktivasi, seperti yang ditunjukkan pada berikut ini.

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

Anda dapat memeriksa status pekerjaan aktivasi dengan operasi get-read-set-metadataAPI. Status yang mungkin adalah`ACTIVE`,`ACTIVATING`, dan`ARCHIVED`. Dalam contoh berikut, ganti *`sequence store ID`* dengan ID penyimpanan urutan Anda, dan ganti *`read set ID`* dengan ID set baca Anda.

```
aws omics get-read-set-metadata --sequence-store-id sequence store ID --id read set ID
```

Tanggapan berikut menunjukkan kepada Anda bahwa set baca aktif.

```
{
    "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"
    }
}
```

Anda dapat melihat semua pekerjaan aktivasi set baca dengan menggunakan list-read-set-activation-jobs, seperti yang ditunjukkan pada contoh berikut. Dalam contoh berikut, ganti *sequence store ID* dengan ID toko urutan Anda.

```
aws omics list-read-set-activation-jobs --sequence-store-id sequence store ID
```

Anda menerima tanggapan berikut.

```
{
    "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 analitik

> **Important**
>
> AWS HealthOmics toko varian dan toko anotasi tidak lagi terbuka untuk pelanggan baru. Pelanggan yang sudah ada dapat terus menggunakan layanan ini seperti biasa. Untuk informasi selengkapnya, lihat AWS HealthOmics toko varian dan perubahan ketersediaan toko anotasi.

HealthOmics analitik mendukung penyimpanan dan analisis varian dan anotasi genom. Analytics menyediakan dua jenis sumber daya penyimpanan - Toko varian dan toko Anotasi. Anda menggunakan sumber daya ini untuk menyimpan, mengubah, dan menanyakan data varian genom dan data anotasi. Setelah mengimpor data ke dalam datastore, Anda dapat menggunakan Athena untuk membentuk analisis lanjutan pada data.

Anda dapat menggunakan HealthOmics konsol atau API untuk membuat dan mengelola toko, mengimpor data, dan berbagi data penyimpanan analitik dengan kolaborator.

Varian menyimpan data dukungan dalam format VCF, dan anotasi menyimpan dukungan TSV/CSV dan format. GFF3 Koordinat genom direpresentasikan sebagai interval setengah terbuka berbasis nol, setengah tertutup. Ketika data Anda berada di penyimpanan data HealthOmics analitik, akses ke file VCF dikelola melalui. AWS Lake Formation Anda kemudian dapat menanyakan file VCF dengan menggunakan Amazon Athena. Kueri harus menggunakan mesin kueri Athena versi 3. Untuk membaca lebih lanjut tentang versi mesin kueri Athena, lihat dokumentasi Amazon Athena.

Topik

- Berbagi toko HealthOmics analitik

Membuat toko HealthOmics varian

> **Important**
>
> AWS HealthOmics toko varian dan toko anotasi tidak lagi terbuka untuk pelanggan baru. Pelanggan yang sudah ada dapat terus menggunakan layanan ini seperti biasa. Untuk informasi selengkapnya, lihat AWS HealthOmics toko varian dan perubahan ketersediaan toko anotasi.

Topik berikut menjelaskan cara membuat toko HealthOmics varian menggunakan konsol dan API.

Topik

- Membuat toko varian menggunakan konsol
- Membuat toko varian menggunakan API

Membuat toko varian menggunakan konsol

Anda dapat membuat toko varian menggunakan HealthOmics konsol.

1. Buka konsol HealthOmics .
2. Jika diperlukan, buka panel navigasi kiri (≡). Pilih toko Varian.
3. Pada halaman Create variant store, berikan informasi berikut
   - Nama toko varian - Nama unik untuk toko ini.
   - Deskripsi (opsional) - Deskripsi toko varian ini.
   - Genom referensi - Genom referensi untuk toko varian ini.
   - Enkripsi Data - Pilih apakah Anda ingin enkripsi data dimiliki dan dikelola oleh AWS atau oleh Anda sendiri.
   - Tag (opsional) - Berikan hingga 50 tag untuk toko varian ini.
4. Pilih Buat toko varian.

Membuat toko varian menggunakan API

Gunakan operasi HealthOmics `CreateVariantStore` API untuk membuat toko varian. Anda juga dapat melakukan operasi ini dengan AWS CLI.

Untuk membuat toko varian, Anda memberikan nama untuk toko dan ARN toko referensi. Toko varian siap untuk menelan data ketika statusnya berubah menjadi READY.

Contoh berikut menggunakan AWS CLI untuk membuat toko varian.

```
aws omics create-variant-store --name myvariantstore \
    --reference referenceArn="arn:aws:omics:us-
west-2:555555555555:referenceStore/123456789/reference/5987565360"
```

Untuk mengonfirmasi pembuatan toko varian Anda, Anda menerima respons berikut.

```
{
    "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"
}
```

Untuk mempelajari lebih lanjut tentang toko varian, gunakan get-variant-storeAPI.

```
aws omics get-variant-store --name myvariantstore
```

Anda menerima tanggapan berikut.

```
{
    "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
}
```

Untuk melihat semua toko varian yang terkait dengan akun, gunakan list-variant-storesAPI.

```
aws omics list-variant-stores
```

Anda menerima respons yang mencantumkan semua toko varian, beserta statusnya IDs, dan detail lainnya, seperti yang ditunjukkan pada contoh respons berikut.

```
{
    "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
        }
    ]
}
```

Anda juga dapat memfilter respons untuk list-variant-storesAPI berdasarkan status atau kriteria lainnya.

File VCF yang diimpor ke toko analitik yang dibuat pada atau setelah 15 Mei 2023 telah menentukan skema untuk anotasi Variant Effect Predictor (VEP). Ini membuatnya lebih mudah untuk menanyakan dan mengurai data VCF yang diimpor. Perubahan tidak memengaruhi penyimpanan yang dibuat sebelum 15 Mei 2023, kecuali jika `annotation fields` parameter tersebut disertakan dalam

panggilan API atau CLI. Untuk toko-toko ini, menggunakan `annotation fields` parameter akan menyebabkan permintaan gagal.

Membuat pekerjaan impor toko HealthOmics varian

> **Important**
>
> AWS HealthOmics toko varian dan toko anotasi tidak lagi terbuka untuk pelanggan baru. Pelanggan yang sudah ada dapat terus menggunakan layanan ini seperti biasa. Untuk informasi selengkapnya, lihat AWS HealthOmics toko varian dan perubahan ketersediaan toko anotasi.

Contoh berikut menunjukkan cara menggunakan AWS CLI untuk membuat pekerjaan impor untuk toko varian.

```
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"}
    ]
}
```

Untuk toko yang dibuat setelah 15 Mei 2023, contoh berikut menunjukkan cara menambahkan `--annotation-fields` parameter. Bidang anotasi didefinisikan dengan impor.

```
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"
}
```

Gunakan get-variant-import-jobuntuk memeriksa status.

```
aws omics get-variant-import-job --job-id 08279950-a9e3-4cc3-9a3c-a574f9c9e229
```

Anda akan menerima respons JSON yang menunjukkan status pekerjaan impor Anda. Anotasi VEP di VCF diuraikan untuk informasi yang disimpan di kolom INFO sebagai pasangan. ID/Value ID default untuk kolom INFO anotasi Efek Variant Ensembl adalah CSQ, tetapi Anda dapat menggunakan `--annotation-fields` parameter untuk menunjukkan nilai kustom yang digunakan di kolom INFO. Parsing saat ini didukung untuk anotasi VEP.

Untuk toko yang dibuat sebelum 15 Mei 2023 atau untuk file VCF yang tidak menyertakan anotasi VEP, responsnya tidak menyertakan bidang anotasi apa pun.

```
{
    "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",
}
```

Anotasi VEP yang merupakan bagian dari file VCF disimpan sebagai skema yang telah ditentukan dengan struktur berikut. Bidang ekstra dapat digunakan untuk menyimpan bidang VEP tambahan yang tidak disertakan dalam skema default.

```
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>
    >>
>
```

Parsing dilakukan dengan pendekatan upaya terbaik. Jika entri VEP tidak mengikuti spesifikasi standar VEP, itu tidak akan diurai dan baris dalam array akan kosong.

Untuk toko varian baru, respons untuk get-variant-import-jobakan menyertakan bidang anotasi, seperti yang ditunjukkan.

```
aws omics get-variant-import-job --job-id 08279950-a9e3-4cc3-9a3c-a574f9c9e229
```

Anda menerima respons JSON yang menunjukkan status pekerjaan impor Anda.

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

Anda dapat menggunakan list-variant-import-jobsuntuk melihat semua pekerjaan impor dan statusnya.

```
aws omics list-variant-import-jobs --ids 7a1c67e3-b7f9-434d-817b-9c571fd63bea
```

Tanggapan tersebut berisi informasi sebagai berikut.

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

Jika perlu, Anda dapat membatalkan pekerjaan impor dengan perintah berikut.

```
aws omics cancel-variant-import-job
     --job-id edd7b8ce-xmpl-47e2-bc99-258cac95a508
```

Membuat toko HealthOmics anotasi

> **Important**
>
> AWS HealthOmics toko varian dan toko anotasi tidak lagi terbuka untuk pelanggan baru. Pelanggan yang sudah ada dapat terus menggunakan layanan ini seperti biasa. Untuk informasi selengkapnya, lihat AWS HealthOmics toko varian dan perubahan ketersediaan toko anotasi.

Penyimpanan anotasi adalah penyimpanan data yang mewakili database anotasi, seperti dari file TSV, VCF, atau GFF. Jika genom referensi yang sama ditentukan, penyimpanan anotasi dipetakan ke sistem koordinat yang sama dengan penyimpanan varian selama impor. Topik berikut menunjukkan cara menggunakan HealthOmics konsol dan AWS CLI untuk membuat dan mengelola toko anotasi.

Topik

Membuat toko anotasi menggunakan konsol

Gunakan prosedur berikut untuk membuat toko anotasi dengan HealthOmics konsol.

Untuk membuat toko anotasi

1. Buka konsol HealthOmics .
2. Jika diperlukan, buka panel navigasi kiri (≡). Pilih toko anotasi.
3. Pada halaman Toko anotasi, pilih Buat toko anotasi.
4. Pada halaman Create annotation store, berikan informasi berikut

   - Nama toko anotasi - Nama unik untuk toko ini.
   - Deskripsi (opsional) - Deskripsi genom referensi ini.

- Format data dan detail skema - Pilih format file data dan unggah definisi skema untuk toko ini.
- Genom referensi - Genom referensi untuk anotasi ini.
- Enkripsi Data - Pilih apakah Anda ingin enkripsi data dimiliki dan dikelola oleh AWS atau oleh Anda sendiri.
- Tag (opsional) - Berikan hingga 50 tag untuk toko anotasi ini.

5. Pilih Buat toko anotasi.

Membuat toko anotasi menggunakan API

Contoh berikut menunjukkan cara membuat toko anotasi menggunakan. AWS CLI Untuk semua operasi AWS CLI dan API, Anda harus menentukan format data Anda.

```
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
```

Anda menerima tanggapan berikut untuk mengonfirmasi pembuatan toko anotasi Anda.

```
{
           "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"
       }
```

Untuk mempelajari lebih lanjut tentang penyimpanan anotasi, gunakan get-annotation-storeAPI.

```
aws omics get-annotation-store --name my_annotation_store
```

Anda menerima tanggapan berikut.

```
{
```

```
          "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
      }
```

Untuk melihat semua penyimpanan anotasi yang terkait dengan akun, gunakan operasi list-annotation-storesAPI.

```
aws omics list-annotation-stores
```

Anda menerima respons yang mencantumkan semua toko anotasi, beserta statusnya IDs, dan detail lainnya, seperti yang ditunjukkan dalam contoh respons berikut.

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

Anda juga dapat memfilter tanggapan berdasarkan status atau kriteria lainnya.

Membuat pekerjaan impor untuk toko HealthOmics anotasi

> **Important**
>
> AWS HealthOmics toko varian dan toko anotasi tidak lagi terbuka untuk pelanggan baru. Pelanggan yang sudah ada dapat terus menggunakan layanan ini seperti biasa. Untuk informasi selengkapnya, lihat AWS HealthOmics toko varian dan perubahan ketersediaan toko anotasi.

Topik

Membuat pekerjaan impor anotasi menggunakan API

Contoh berikut menunjukkan cara menggunakan AWS CLI untuk memulai pekerjaan impor anotasi.

```
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"}'
```

Penyimpanan anotasi yang dibuat sebelum 15 Mei 2023 mengembalikan pesan kesalahan jika bidang anotasi disertakan. Mereka tidak mengembalikan output untuk operasi API apa pun yang terlibat dengan pekerjaan impor toko anotasi.

Anda kemudian dapat menggunakan operasi get-annotation-import-jobAPI dan `job ID` parameter untuk mempelajari detail selengkapnya tentang pekerjaan impor anotasi.

```
aws omics get-annotation-import-job --job-id 9e4198fb-fa85-446c-9301-9b823a1a8ba8
```

Anda menerima respons berikut, termasuk bidang anotasi.

```
{
          "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"}
       }
```

Untuk melihat semua pekerjaan impor toko anotasi, gunakan list-annotation-import-jobs.

```
aws omics list-annotation-import-jobs --ids 9e4198fb-fa85-446c-9301-9b823a1a8ba8
```

Tanggapan tersebut mencakup detail dan status pekerjaan impor toko anotasi Anda.

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

```
    }
```

Parameter tambahan untuk format TSV dan VCF

Untuk format TSV dan VCF, ada parameter tambahan yang menginformasikan API tentang cara mengurai input Anda.

> **Important**
>
> Data anotasi CSV yang diekspor dengan mesin kueri secara langsung mengembalikan informasi dari impor dataset. Jika data yang diimpor berisi rumus atau perintah, file tersebut mungkin tunduk pada injeksi CSV. Oleh karena itu, file yang diekspor dengan mesin kueri dapat meminta peringatan keamanan. Untuk menghindari aktivitas berbahaya, matikan tautan dan makro saat membaca file ekspor.

Pengurai TSV juga melakukan operasi bioinformatika dasar, seperti normalisasi kiri dan standardisasi koordinat genomik, yang tercantum dalam tabel berikut.

Jenis format	Deskripsi
Generik	File teks generik. Tidak ada informasi genom.
`CHR_POS`	Posisi awal - 1, Tambahkan posisi akhir, yang sama dengan`POS`.
`CHR_POS_REF_ALT`	Berisi informasi contig, posisi 1-basis, ref dan alt alel.
`CHR_START_END_REF_ALT_ONE_BASE`	Berisi informasi alel contig, start, end, ref dan alt. Koordinat berbasis 1.
`CHR_START_END_ZERO_BASE`	Berisi posisi contig, start, dan end. Koordinat berbasis 0.
`CHR_START_END_ONE_BASE`	Berisi posisi contig, start, dan end. Koordinat berbasis 1.

Jenis format	Deskripsi
CHR_START_END_REF_ALT_ZERO_BASE	Berisi informasi alel contig, start, end, ref dan alt. Koordinat berbasis 0.

Permintaan penyimpanan anotasi impor TSV terlihat seperti contoh berikut.

```
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"
        }
    }
}'
```

Membuat toko anotasi berformat TSV

Contoh berikut membuat toko anotasi menggunakan file terbatas tab yang berisi header, baris, dan komentar. KoordinatnyaCHR_START_END_ONE_BASED, dan berisi peta HG19 gen dari Sinopsis Peta Gen Manusia 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}]}'
```

Anda dapat mengimpor file dengan atau tanpa header. Untuk menunjukkan ini dalam permintaan CLI, gunakan`header=false`, seperti yang ditunjukkan dalam contoh pekerjaan impor berikut.

```
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="#"}}'
```

Contoh berikut membuat toko anotasi untuk file tempat tidur. File tempat tidur adalah file tab yang dibatasi sederhana. Dalam contoh ini, kolomnya adalah kromosom, awal, akhir, dan nama wilayah. Koordinat berbasis nol, dan data tidak memiliki 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}]}'
```

Anda kemudian dapat mengimpor file tempat tidur ke toko anotasi dengan menggunakan perintah CLI berikut.

```
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="#"}}'
```

Contoh berikut membuat penyimpanan anotasi untuk file tab dibatasi yang berisi beberapa kolom pertama dari file VCF, diikuti oleh kolom dengan informasi anotasi. Ini berisi posisi genom dengan informasi tentang kromosom, awal, referensi dan alel alternatif, dan berisi tajuk.

```
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}]}'
```

Anda kemudian akan mengimpor file ke toko anotasi menggunakan perintah CLI berikut.

```
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="#"}}'
```

Contoh berikut menunjukkan bagaimana pelanggan dapat membuat penyimpanan anotasi untuk file mim2gene. File MIM2gene menyediakan hubungan antara gen di OMIM dan pengidentifikasi gen lainnya. Ini tab dibatasi dan berisi komentar.

```
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}]}'
```

Anda kemudian dapat mengimpor data ke toko Anda sebagai berikut.

```
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="#"}}'
```

Memulai pekerjaan impor berformat VCF

Untuk file VCF, ada dua input tambahan, `ignoreQualField` dan`ignoreFilterField`, yang mengabaikan atau menyertakan parameter tersebut seperti yang ditunjukkan.

```
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
    }
   }'
```

Anda juga dapat membatalkan impor toko anotasi, seperti yang ditunjukkan. Jika pembatalan berhasil, Anda tidak menerima tanggapan atas panggilan ini AWS CLI . Namun, jika ID pekerjaan impor tidak ditemukan atau pekerjaan impor selesai, Anda menerima pesan galat.

```
aws omics cancel-annotation-import-job --job-id edd7b8ce-xmpl-47e2-bc99-258cac95a508
```

> **Note**
>
> Riwayat pekerjaan impor metadata Anda untuk get-annotation-import-job,, get-variant-import-joblist-annotation-import-jobs, dan dihapus list-variant-import-jobssecara otomatis setelah dua tahun. Varian dan anotasi data yang diimpor tidak otomatis dihapus dan tetap berada di penyimpanan data Anda.

Membuat versi toko HealthOmics anotasi

> **Important**
>
> AWS HealthOmics toko varian dan toko anotasi tidak lagi terbuka untuk pelanggan baru. Pelanggan yang sudah ada dapat terus menggunakan layanan ini seperti biasa. Untuk informasi selengkapnya, lihat AWS HealthOmics toko varian dan perubahan ketersediaan toko anotasi.

Anda dapat membuat versi baru dari toko anotasi untuk mengumpulkan versi yang berbeda dari database anotasi Anda. Ini membantu Anda mengatur data anotasi Anda, yang diperbarui secara berkala.

Untuk membuat versi baru dari penyimpanan anotasi yang ada, gunakan create-annotation-store-versionAPI seperti yang ditunjukkan pada contoh berikut.

```
aws omics create-annotation-store-version \
     --name my_annotation_store \
     --version-name my_version
```

Anda akan mendapatkan respons berikut dengan ID versi penyimpanan anotasi, mengonfirmasi bahwa versi baru anotasi Anda telah dibuat.

```
{
     "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"
}
```

Untuk memperbarui deskripsi versi penyimpanan anotasi, Anda dapat menggunakan update-annotation-store-versionuntuk menambahkan pembaruan ke versi penyimpanan anotasi.

```
aws omics update-annotation-store-version \
    --name my_annotation_store \
    --version-name my_version \
    --description "New Description"
```

Anda akan menerima tanggapan berikut, mengonfirmasi bahwa versi toko anotasi telah diperbarui.

```
{
    "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"
}
```

Untuk melihat detail versi toko anotasi, gunakan get-annotation-store-version.

```
aws omics get-annotation-store-version --name my_annotation_store --version-name
 my_version
```

Anda akan menerima respons dengan nama versi, status, dan detail lainnya.

```
{
    "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
    }
```

Untuk melihat semua versi toko anotasi, Anda dapat menggunakan list-annotation-store-versions, seperti yang ditunjukkan pada contoh berikut.

```
aws omics list-annotation-store-versions --name my_annotation_store
```

Anda akan menerima tanggapan dengan informasi berikut

```
{
  "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
    }
}
```

Jika Anda tidak lagi memerlukan versi penyimpanan anotasi, Anda dapat menggunakan delete-annotation-store-versionsuntuk menghapus versi penyimpanan anotasi, seperti yang ditunjukkan pada contoh berikut.

```
aws omics delete-annotation-store-versions --name my_annotation_store --versions
 my_version
```

Jika versi toko dihapus tanpa kesalahan, Anda akan menerima respons berikut.

```
{
  "errors": []
}
```

Jika ada kesalahan, Anda akan menerima respons dengan detail kesalahan, seperti yang ditunjukkan.

```
{
  "errors": [
    {
      "versionName": "my_version",
      "message": "Version with versionName: my_version was not found."
    }
  ]
}
```

Jika Anda mencoba menghapus versi penyimpanan anotasi yang memiliki pekerjaan impor aktif, Anda akan menerima respons dengan kesalahan, seperti yang ditunjukkan.

```
{
  "errors": [
    {
      "versionName": "my_version",
      "message": "version has an inflight import running"
    }
  ]
}
```

Dalam hal ini, Anda dapat memaksa penghapusan versi penyimpanan anotasi, seperti yang ditunjukkan pada contoh berikut.

```
aws omics delete-annotation-store-versions --name my_annotation_store --versions
 my_version --force
```

Menghapus toko HealthOmics analitik

> **Important**
>
> AWS HealthOmics toko varian dan toko anotasi tidak lagi terbuka untuk pelanggan baru. Pelanggan yang sudah ada dapat terus menggunakan layanan ini seperti biasa. Untuk informasi selengkapnya, lihat AWS HealthOmics toko varian dan perubahan ketersediaan toko anotasi.

Saat Anda menghapus penyimpanan varian atau anotasi, sistem juga menghapus semua data yang diimpor di toko tersebut dan tag yang terkait.

Contoh berikut menunjukkan cara menghapus toko varian menggunakan AWS CLI. Jika tindakan berhasil, status penyimpanan varian bertransisi ke`DELETING`.

```
aws omics delete-variant-store --id <variant-store-id>
```

Contoh berikut menunjukkan cara menghapus toko anotasi. Jika tindakan berhasil, anotasi menyimpan status transisi ke. `DELETING` Toko anotasi tidak dapat dihapus jika ada lebih dari satu versi.

```
aws omics delete-annotation-store --id <annotation-store-id>
```

Menanyakan HealthOmics data analitik

> **Important**
>
> AWS HealthOmics toko varian dan toko anotasi tidak lagi terbuka untuk pelanggan baru. Pelanggan yang sudah ada dapat terus menggunakan layanan ini seperti biasa. Untuk informasi selengkapnya, lihat AWS HealthOmics toko varian dan perubahan ketersediaan toko anotasi.

Anda dapat melakukan kueri di toko varian Anda menggunakan AWS Lake Formation Amazon Athena atau Amazon EMR. Sebelum Anda menjalankan kueri apa pun, selesaikan prosedur penyiapan (dijelaskan di bagian berikut) untuk Lake Formation dan Amazon Athena.

Untuk informasi tentang Amazon EMR, lihat Tutorial: Memulai Amazon EMR

Untuk toko varian yang dibuat setelah 26 September 2024, HealthOmics mempartisi toko dengan ID sampel. Partisi ini berarti HealthOmics menggunakan ID sampel untuk mengoptimalkan penyimpanan informasi varian. Kueri yang menggunakan informasi sampel sebagai filter akan mengembalikan hasil lebih cepat, karena kueri memindai lebih sedikit data.

HealthOmics menggunakan sampel IDs sebagai nama file partisi. Sebelum Anda menelan data, periksa apakah ID sampel berisi data PHI. Jika ya, ubah ID sampel sebelum Anda menelan data. Untuk informasi selengkapnya tentang konten apa yang harus disertakan dan tidak disertakan dalam sampel IDs, lihat panduan di halaman web kepatuhan AWS HIPAA.

Topik

- Mengkonfigurasi Lake Formation untuk digunakan HealthOmics
- Mengkonfigurasi Athena untuk kueri
- Menjalankan kueri di toko HealthOmics varian

Mengkonfigurasi Lake Formation untuk digunakan HealthOmics

> **Important**
>
> AWS HealthOmics toko varian dan toko anotasi tidak lagi terbuka untuk pelanggan baru. Pelanggan yang sudah ada dapat terus menggunakan layanan ini seperti biasa. Untuk informasi selengkapnya, lihat AWS HealthOmics toko varian dan perubahan ketersediaan toko anotasi.

Sebelum Anda menggunakan Lake Formation untuk mengelola penyimpanan HealthOmics data, lakukan prosedur konfigurasi Lake Formation berikut.

Topik

- Membuat atau memverifikasi administrator Lake Formation
- Membuat tautan sumber daya menggunakan konsol Lake Formation

- Mengkonfigurasi izin untuk berbagi sumber daya AWS RAM

Membuat atau memverifikasi administrator Lake Formation

Sebelum Anda dapat membuat data lake di Lake Formation, Anda menentukan satu atau beberapa administrator.

Administrator adalah pengguna dan peran dengan izin untuk membuat tautan sumber daya. Anda mengatur administrator data lake per akun per wilayah.

Buat pengguna admin di konsol Lake Formation

1. Buka konsol AWS Lake Formation: Lake Formation console
2. Jika konsol menampilkan panel Welcome to Lake Formation, pilih Memulai.

   Lake Formation menambahkan Anda ke tabel administrator danau Data.
3. Jika tidak, dari menu kiri, pilih Peran dan tugas administrasi.
4. Tambahkan administrator tambahan sesuai kebutuhan.

Membuat tautan sumber daya menggunakan konsol Lake Formation

Untuk membuat sumber daya bersama yang dapat ditanyakan pengguna, kontrol akses default harus dinonaktifkan. Untuk mempelajari selengkapnya tentang menonaktifkan kontrol akses default, lihat Mengubah setelan keamanan default untuk data lake Anda di dokumentasi Lake Formation. Anda dapat membuat tautan sumber daya secara individual atau sebagai grup, sehingga Anda dapat mengakses data di Amazon Athena atau AWS layanan lain (seperti Amazon EMR).

Membuat tautan sumber daya di konsol AWS Lake Formation dan membagikannya dengan pengguna HealthOmics Analytics

1. Buka konsol AWS Lake Formation: Lake Formation console
2. Di bilah navigasi utama, pilih Database.
3. Dalam tabel Databases, pilih database yang diinginkan.
4. Dari menu Buat, pilih Tautan sumber daya.
5. Masukkan nama tautan Sumber Daya. Jika Anda berencana untuk mengakses database dari Athena, masukkan nama hanya menggunakan huruf kecil (hingga 256 karakter).
6. Pilih Buat.

7. Tautan sumber daya baru sekarang terdaftar di bawah Database.

Berikan akses ke sumber daya bersama menggunakan konsol Lake Formation

Administrator database Lake Formation dapat memberikan akses ke sumber daya bersama menggunakan prosedur berikut.

1. Buka konsol AWS Lake Formation: https://console.aws.amazon.com/lakeformation/
2. Di bilah navigasi utama, pilih Database.
3. Pada halaman Database, pilih tautan sumber daya yang sebelumnya Anda buat.
4. Dari menu Tindakan, pilih Hibah sesuai target.
5. Pada halaman Berikan izin data di bawah Prinsipal, pilih pengguna atau peran IAM.
6. Dari menu tarik-turun pengguna atau peran IAM, temukan pengguna yang ingin Anda berikan aksesnya.
7. Selanjutnya, di bawah LF-tag atau kartu sumber daya katalog, pilih opsi Sumber daya katalog data bernama.
8. Dari menu tarik-turun Tabel-opsional, pilih Semua Tabel atau tabel yang sebelumnya Anda buat.
9. Di kartu izin Tabel, di bawah Izin tabel pilih Jelaskan dan Pilih.
10. Selanjutnya, pilih Grant.

Untuk melihat izin Lake Formation, pilih Izin data lake dari panel navigasi utama. Tabel menunjukkan database yang tersedia dan link sumber daya.

Mengkonfigurasi izin untuk berbagi sumber daya AWS RAM

Di konsol AWS Lake Formation, lihat izin dengan memilih Izin data lake di bilah navigasi utama. Pada halaman Izin data, Anda dapat melihat tabel yang menampilkan tipe Sumber Daya, Database, dan **ARN** yang terkait dengan sumber daya bersama di bawah RAM Resource Share. Jika Anda perlu menerima pembagian sumber daya AWS Resource Access Manager (AWS RAM), AWS Lake Formation akan memberi tahu Anda di konsol.

HealthOmics dapat secara implisit menerima pembagian AWS RAM sumber daya selama pembuatan toko. Untuk menerima pembagian AWS RAM sumber daya, pengguna IAM atau peran yang memanggil operasi `CreateVariantStore` atau `CreateAnnotationStore` API harus mengizinkan tindakan berikut:

- `ram:GetResourceShareInvitations`- Tindakan ini memungkinkan HealthOmics untuk menemukan undangan.
- `ram:AcceptResourceShareInvitation`- Tindakan ini memungkinkan HealthOmics untuk menerima undangan dengan menggunakan token FAS.

Tanpa izin ini, Anda melihat kesalahan otorisasi selama pembuatan toko.

Berikut adalah contoh kebijakan yang mencakup tindakan ini. Tambahkan kebijakan ini ke pengguna IAM atau peran yang menerima pembagian AWS RAM sumber daya.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "omics:*",
        "ram:AcceptResourceShareInvitation",
        "ram:GetResourceShareInvitations"
      ],
      "Resource": "*"
    }
  ]
}
```

Mengkonfigurasi Athena untuk kueri

> **Important**
>
> AWS HealthOmics toko varian dan toko anotasi tidak lagi terbuka untuk pelanggan baru. Pelanggan yang sudah ada dapat terus menggunakan layanan ini seperti biasa. Untuk informasi selengkapnya, lihat AWS HealthOmics toko varian dan perubahan ketersediaan toko anotasi.

Anda dapat menggunakan Athena untuk menanyakan varian dan anotasi. Sebelum Anda menjalankan kueri apa pun, lakukan tugas penyiapan berikut:

Topik

Konfigurasikan lokasi hasil kueri menggunakan konsol Athena

Untuk mengonfigurasi lokasi hasil kueri, ikuti langkah-langkah ini.

1. Buka konsol Athena: konsol Athena
2. Di bilah navigasi utama, pilih Editor kueri.
3. Di editor kueri, pilih tab Pengaturan, lalu pilih Kelola.
4. Masukkan awalan S3 lokasi untuk menyimpan hasil kueri.

Konfigurasikan workgroup dengan Athena engine v3

Untuk mengonfigurasi grup kerja, ikuti langkah-langkah ini.

1. Buka konsol Athena: konsol Athena
2. Di bilah navigasi utama, pilih Workgroups, lalu Buat workgroup.
3. Masukkan nama untuk workgroup.
4. Pilih Athena SQL sebagai jenis mesin.
5. Di bawah Upgrade mesin kueri, pilih Manual.
6. Di bawah Mesin versi Query, pilih Athena versi 3.
7. Pilih Buat grup kerja.

Menjalankan kueri di toko HealthOmics varian

Important

AWS HealthOmics toko varian dan toko anotasi tidak lagi terbuka untuk pelanggan baru. Pelanggan yang sudah ada dapat terus menggunakan layanan ini seperti biasa. Untuk

informasi selengkapnya, lihat AWS HealthOmics toko varian dan perubahan ketersediaan toko anotasi.

Anda dapat melakukan kueri di toko varian Anda menggunakan Amazon Athena. Perhatikan bahwa koordinat genom dalam penyimpanan varian dan anotasi direpresentasikan sebagai interval setengah terbuka setengah tertutup berbasis nol.

Jalankan kueri sederhana menggunakan konsol Athena

Contoh berikut menunjukkan bagaimana menjalankan query sederhana.

1. Buka editor Kueri Athena: Athena Query editor
2. Di bawah Workgroup, pilih workgroup yang Anda buat selama penyiapan.
3. Verifikasi bahwa sumber data adalah AwsDataCatalog.
4. Untuk Database, pilih tautan sumber daya database yang Anda buat selama penyiapan Lake Formation.
5. Salin kueri berikut ke Editor Kueri di bawah tab Query 1:

```
SELECT * from omicsvariants limit 10
```

6. Pilih Jalankan untuk menjalankan kueri. Konsol mengisi tabel hasil dengan 10 baris pertama omicsvariants tabel.

Jalankan kueri kompleks menggunakan konsol Athena

Contoh berikut menunjukkan bagaimana menjalankan query kompleks. Untuk menjalankan kueri ini, impor `ClinVar` ke toko anotasi.

Jalankan kueri yang kompleks

1. Buka editor Kueri Athena: Athena Query editor
2. Di bawah Workgroup, pilih workgroup yang Anda buat selama penyiapan.
3. Verifikasi bahwa sumber data adalah AwsDataCatalog.
4. Untuk Database, pilih tautan sumber daya database yang Anda buat selama penyiapan Lake Formation.

5. Pilih + di kanan atas untuk membuat tab kueri baru bernama Query 2.
6. Salin kueri berikut ke Editor Kueri di bawah tab Query 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. Pilih Jalankan untuk mulai menjalankan kueri.

Berbagi toko HealthOmics analitik

> **Important**
>
> AWS HealthOmics toko varian dan toko anotasi tidak lagi terbuka untuk pelanggan baru. Pelanggan yang sudah ada dapat terus menggunakan layanan ini seperti biasa. Untuk informasi selengkapnya, lihat AWS HealthOmics toko varian dan perubahan ketersediaan toko anotasi.

Sebagai pemilik toko varian atau toko anotasi, Anda dapat berbagi toko dengan akun AWS lainnya. Pemilik dapat mencabut akses ke sumber daya bersama dengan menghapus pembagian.

Sebagai pelanggan toko bersama, Anda terlebih dahulu menerima bagian tersebut. Anda kemudian dapat menentukan alur kerja yang menggunakan toko bersama. Data muncul sebagai tabel di keduanya AWS Glue dan Lake Formation.

Ketika Anda tidak lagi memerlukan akses ke toko, Anda menghapus bagian.

Lihat Berbagi sumber daya lintas akun di AWS HealthOmics untuk informasi tambahan tentang berbagi sumber daya.

Membuat berbagi toko

Untuk membuat store share, gunakan operasi create-share API. Pelanggan utama adalah pengguna Akun AWS yang akan berlangganan saham. Contoh berikut membuat share untuk toko varian. Untuk berbagi toko dengan lebih dari satu akun, Anda membuat beberapa saham dari toko yang sama.

```
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"
```

Jika pembuatan berhasil, Anda menerima respons dengan ID dan status berbagi.

```
{
       "shareId": "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a",
       "name": "my_Share-123",
       "status": "PENDING"
  }
```

Bagian tetap dalam status tertunda hingga pelanggan menerimanya menggunakan operasi API terima-berbagi.

Berbagi sumber daya lintas akun di AWS HealthOmics

Gunakan berbagi lintas akun untuk berbagi sumber daya dengan kolaborator tanpa membuat salinan atau memodifikasi kebijakan sumber daya IAM. Sumber daya berikut mendukung berbagi lintas akun:

- HealthOmics toko varian
- HealthOmics toko anotasi
- Alur kerja pribadi

Berbagi sumber daya mencakup langkah-langkah berikut:

1. Pemilik sumber daya membuat bagian, dan menentukan ARN sumber daya dan pelanggan Akun AWS yang dituju. Pembagian sumber daya tetap dalam keadaan tertunda sampai pelanggan menerima bagian tersebut.
2. Pelanggan menerima pembagian sumber daya untuk mendapatkan akses ke sumber daya. Transisi berbagi sumber daya ke status pengaktifan.
3. HealthOmics Layanan ini menyediakan akun pelanggan dengan akses ke sumber daya.
4. Pemilik sumber daya dapat menghapus bagian, atau pelanggan dapat mencabut akses mereka ke bagian tersebut. Pelanggan tidak dapat menghapus berbagi atau sumber daya terkait.

Topik

Membuat berbagi

Anda dapat menggunakan operasi create-share API untuk membuat share. Pelanggan utama adalah pengguna Akun AWS yang akan berlangganan sumber daya bersama. Contoh berikut membuat share untuk toko varian.

```
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"
```

Jika pembuatan berhasil, Anda menerima respons dengan ID dan status berbagi.

```
{
"shareId": "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a",
     "name": "my_Share-123",
     "status": "PENDING"
}
```

Bagian tetap dalam status tertunda sampai pelanggan menerimanya menggunakan operasi accept-share API.

```
 aws omics accept-share \
    --share-id "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a"
```

Setelah pelanggan menerima bagian, saham beralih ke keadaan aktif.

```
{
"status": "ACTIVATING"
}
```

Mengambil informasi tentang berbagi

Gunakan operasi get-share API untuk mengambil informasi tentang share.

```
aws omics get-share --share-id "495c21bedc889d07d0ab69d710a6841e-
dd75ab7a1a9c384fa848b5bd8e5a7e0a"
```

Respons API mencakup informasi metadata tentang pembagian.

```
{
  "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"
    }
}
```

Lihat saham yang Anda miliki

Gunakan API list-shares untuk mengambil informasi tentang masing-masing saham yang Anda miliki.

```
aws omics list-shares  --resource-owner SELF
```

Respons API menyertakan metadata untuk setiap share yang Anda miliki.

Lihat saham yang diterima dari akun lain

Gunakan API list-shares untuk melihat semua saham yang Anda terima dari akun lain.

```
aws omics list-shares  --resource-owner OTHER
```

Respons API menyertakan metadata untuk setiap share yang Anda terima.

Hapus berbagi

Gunakan delete-share API untuk menghapus share setelah Anda tidak lagi membutuhkannya.

```
aws omics delete-share \
    --share-id "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a"
```

Menandai sumber daya di HealthOmics

Topik

Pemberitahuan penting

HealthOmics melindungi data pelanggan berdasarkan kebijakan AWS Shared Responsibility Model. Ini berarti bahwa semua data pelanggan dienkripsi baik dalam transisi maupun saat istirahat. Namun, tidak semua nama yang dimasukkan pelanggan untuk sumber daya seperti penyimpanan data atau operasi berbasis pekerjaan dienkripsi. Mereka tidak boleh mengandung Informasi Identifikasi Pribadi atau Informasi Kesehatan yang Dilindungi. Untuk informasi selengkapnya, lihat Keamanan di AWS HealthOmics.

Sumber daya penandaan HealthOmics

Anda dapat menetapkan metadata ke sumber daya AWS menggunakan tag. Setiap tag adalah label yang terdiri dari kunci dan nilai yang ditentukan pengguna. Tag membantu Anda mengelola, mengidentifikasi, mengatur, dan memfilter sumber daya.

Topik ini menjelaskan kategori dan strategi penandaan yang umum digunakan untuk membantu Anda menerapkan strategi penandaan yang konsisten dan efektif. Bagian berikut mengasumsikan pengetahuan dasar tentang sumber daya AWS, penandaan, penagihan terperinci, dan. AWS Identity and Access Management

Setiap tag memiliki dua bagian:

- Kunci tag (misalnya, CostCenter, Lingkungan, atau Proyek). Kunci tag peka huruf besar dan kecil.
- Nilai tag (misalnya, 111122223333 atau Produksi). Seperti kunci tag, nilai tag peka huruf besar dan kecil.

Anda dapat menggunakan tag untuk mengategorikan sumber daya berdasarkan tujuan, pemilik, lingkungan, atau kriteria lainnya. Untuk informasi selengkapnya, lihat Strategi Penandaan AWS.

Anda dapat menambahkan, mengubah, atau menghapus tag untuk sumber daya dari konsol layanan sumber daya, API layanan, atau AWS CLI.

Untuk mengaktifkan penandaan, pastikan TagResources diotorisasi. Anda dapat mengotorisasi TagResources dengan melampirkan kebijakan IAM seperti contoh berikut.

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": "*"
        }
    ]
}
```

Praktik terbaik

Saat Anda membuat strategi penandaan untuk sumber daya AWS, ikuti praktik terbaik:

- Jangan menyimpan Informasi Identifikasi Pribadi (PII), Informasi Kesehatan yang Dilindungi (PHI) atau informasi sensitif lainnya dalam tag.
- Gunakan format tag terstandardisasi yang peka huruf besar dan kecil serta terapkan secara konsisten di semua jenis sumber daya.
- Pertimbangkan pedoman tag yang mendukung berbagai tujuan, seperti mengelola kontrol akses sumber daya, pelacakan biaya, otomatisasi, dan organisasi.
- Gunakan alat otomatis untuk membantu mengelola tag sumber daya. AWS Resource Groups dan Resource Groups Tagging API memungkinkan kontrol terprogram tag, sehingga memungkinkan untuk secara otomatis mengelola, mencari, dan memfilter tag dan sumber daya.
- Penandaan lebih efektif ketika Anda menggunakan lebih banyak tag.
- Tag dapat diedit atau dimodifikasi sesuai kebutuhan pengguna. Namun untuk memperbarui tag kontrol akses, Anda juga harus memperbarui kebijakan yang mereferensikan tag tersebut untuk mengontrol akses ke sumber daya Anda.

Persyaratan penandaan

Tag memiliki persyaratan sebagai berikut:

- Kunci tidak dapat diawali dengan aws:.
- Kunci harus unik per set tag.
- Kunci harus antara 1 dan 128 karakter yang diizinkan.
- Nilai harus antara 0 dan 256 karakter yang diizinkan.
- Nilai tidak perlu unik per set tag.
- Karakter yang diizinkan untuk kunci dan nilai adalah huruf Unicode, digit, spasi putih, dan salah satu simbol berikut: _.:/= + - @.
- Kunci dan nilai peka huruf besar dan kecil.

Tag set baca toko urutan

Untuk penyimpanan urutan, tag yang dibuat pada set baca berada di tingkat sumber daya set baca. Set baca juga berisi objek di bawahnya yang dapat diakses, dicari, dan dibatasi menggunakan APIs

S3. Secara default, ID sampel (Omics:sampleID) dan ID subjek (Omics:subjectID) ditambahkan ke objek.

Selain itu, hingga lima tag dapat disinkronkan antara set baca dan objek di bawahnya. Konfigurasi tag yang akan disinkronkan adalah konfigurasi tingkat toko yang ditetapkan selama pembuatan toko atau pembaruan menggunakan `propogatedSetLevelTags` parameter.

Jika sudah ada data di toko, memperbarui kunci mungkin memakan waktu. Selama pembaruan ini, HealthOmics ubah status toko menjadiUpdating. Setelah selesai, HealthOmics atur status toko keActive. Saat tag menyebar, izin yang mengandalkan tag mungkin tidak diberlakukan. Izin akan diberlakukan setelah propagasi tag selesai.

Ketika tag disetel atau diperbarui pada set baca, sistem memutuskan apakah akan memperbarui objek untuk set baca tersebut, berdasarkan konfigurasi toko.

Menambahkan tag ke sumber HealthOmics daya

Menambahkan tag ke sumber daya dapat membantu Anda mengidentifikasi dan mengatur sumber daya AWS serta mengelola akses ke sumber daya AWS. Pertama, Anda menambahkan satu atau lebih tag (pasangan nilai kunci) ke sumber daya. Anda dapat menggunakan hingga 50 tag per sumber daya. Ada juga batasan pada karakter yang dapat Anda gunakan di bidang kunci dan nilai.

Setelah menambahkan tag, Anda dapat membuat kebijakan IAM untuk mengelola akses ke AWS sumber daya berdasarkan tag ini. Anda dapat menggunakan HealthOmics konsol atau AWS CLI untuk menambahkan tag ke sumber daya. Menambahkan tag ke repositori dapat memengaruhi akses ke repositori tersebut. Sebelum menambahkan tag ke penyimpanan data, tinjau kebijakan IAM apa pun yang mungkin menggunakan tag untuk mengontrol akses ke sumber daya seperti penyimpanan data.

Tag layanan dibuat secara otomatis untuk subjek dan id sampel untuk penyimpanan urutan.

Ikuti langkah-langkah ini untuk menggunakan AWS CLI untuk menambahkan tag ke HealthOmics sumber daya. Misalnya, untuk menambahkan tag ke toko urutan saat sedang dibuat, Anda akan menggunakan perintah berikut di AWS CLI. Nama toko urutan adalah MySequenceStore, dan dua tag yang ditambahkan dengan kunci adalah key1 dan key2 dengan nilai masing-masing sebagai value1 dan value2 :

```
aws omics create-sequence-store --name "MySequenceStore" --tags key1=value1,key2=value2
```

Output tidak mencantumkan tag. Ini mengembalikan respons berikut.

```
{
    "id": "6860403586",
    "referenceStoreId": "4889894479",
    "roleArn": "arn:aws:iam::555555555555:role/ImportTest",
    "status": "CREATED",
    "creationTime": "2022-07-21T01:19:07.194Z"
}
```

Untuk menambahkan tag ke sumber daya yang ada, Anda akan menjalankan perintah contoh berikut.

```
aws omics tag-resource --resource-arn arn:aws:omics:us-
west-2:555555555555:sequenceStore/2275234794 --tags key1=value1,key2=value2
```

Jika berhasil, perintah ini tidak mengembalikan respon.

Listing tag untuk sumber daya

Ikuti langkah-langkah ini untuk menggunakan AWS CLI untuk melihat daftar AWS tag untuk HealthOmics sumber daya. Jika tidak ada tanda yang telah ditambahkan, daftar yang ditampilkan kosong.

Di terminal atau baris perintah, jalankan list-tags-for-resource perintah seperti yang ditunjukkan pada contoh berikut.

```
aws omics list-tags-for-resource --resource-arn arn:aws:omics:us-
west-2:555555555555:sequenceStore/2275234794
```

Anda akan menerima daftar tag sebagai tanggapan, dalam format JSON.

```
 {
    "tags": {
        "key1": "value1",
        "key2": "value2"
    }
}
```

Menghapus tag dari penyimpanan data

Anda dapat menghapus satu atau beberapa tag yang terkait dengan sumber daya. Menghapus tag tidak menghapus tag dari sumber daya AWS lain yang terkait dengan tag tersebut.

Di terminal atau baris perintah, jalankan perintah untag-resource, tentukan Amazon Resource Name (ARN) dari sumber daya tempat Anda ingin menghapus tag dan kunci tag tag yang ingin Anda hapus.

```
aws omics untag-resource --resource-arn arn:aws:omics:us-
west-2:555555555555:sequenceStore/2275234794 --tag-keys key1,key2
```

Jika berhasil, perintah ini tidak mengembalikan respons. Untuk memverifikasi tanda yang terkait dengan sumber daya, jalankan perintah list-tags-for-resource.

Izin IAM untuk HealthOmics

Anda dapat menggunakan AWS Identity and Access Management (IAM) untuk mengelola akses ke HealthOmics API dan sumber daya seperti toko dan alur kerja. Untuk pengguna dan aplikasi di akun yang digunakan HealthOmics, Anda mengelola izin dalam kebijakan izin yang dapat diterapkan pada pengguna, grup, atau peran IAM.

Untuk mengelola izin bagi pengguna dan aplikasi di akun Anda, gunakan kebijakan yang HealthOmics menyediakan, atau menulis sendiri. HealthOmics Konsol menggunakan beberapa layanan untuk mendapatkan informasi tentang konfigurasi dan pemicu fungsi Anda. Anda dapat menggunakan kebijakan yang disediakan apa adanya, atau sebagai titik awal untuk kebijakan yang lebih ketat.

HealthOmics menggunakan peran layanan IAM untuk mengakses layanan lain atas nama Anda. Misalnya, Anda akan membuat atau memilih peran layanan saat menjalankan alur kerja yang membaca data dari Amazon S3. Untuk beberapa fitur, Anda juga perlu mengonfigurasi izin pada sumber daya di layanan lain. Tinjau persyaratan ini sebelum Anda mulai bekerja dengan HealthOmics

Untuk informasi selengkapnya tentang IAM role, lihat Apa itu IAM? dalam Panduan Pengguna IAM.

Topik

Kebijakan IAM berbasis identitas untuk HealthOmics

Untuk memberikan akses kepada pengguna di akun Anda HealthOmics, Anda menggunakan kebijakan berbasis identitas di AWS Identity and Access Management (IAM). Kebijakan berbasis identitas dapat diterapkan langsung ke pengguna IAM, atau ke grup dan peran IAM yang terkait dengan pengguna. Anda juga dapat memberikan izin kepada pengguna di akun lain untuk berperan dalam akun Anda dan mengakses HealthOmics sumber daya Anda.

Untuk memberikan izin bagi pengguna untuk melakukan tindakan pada versi alur kerja, Anda harus menambahkan alur kerja dan versi alur kerja tertentu ke daftar sumber daya.

Kebijakan IAM berikut memungkinkan pengguna untuk mengakses semua tindakan HealthOmics API, dan meneruskan peran layanan ke HealthOmics.

Example Kebijakan pengguna

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "omics:*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "iam:PassRole"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": "omics.amazonaws.com"
        }
      }
    }
  ]
}
```

Saat Anda menggunakan HealthOmics, Anda juga berinteraksi dengan AWS layanan lain. Untuk mengakses layanan ini, gunakan kebijakan terkelola yang disediakan oleh setiap layanan. Untuk membatasi akses ke subset sumber daya, Anda dapat menggunakan kebijakan terkelola sebagai titik awal untuk membuat kebijakan Anda sendiri yang lebih ketat.

- AmazonS3 FullAccess — Akses ke ember Amazon S3 dan objek yang digunakan oleh pekerjaan.
- Amazon EC2 ContainerRegistryFullAccess - Akses ke registri Amazon ECR dan repositori untuk gambar wadah alur kerja.
- AWSLakeFormationDataAdmin— Akses ke database dan tabel Lake Formation yang dibuat oleh toko analitik.
- ResourceGroupsandTagEditorFullAccess— Tag HealthOmics sumber daya dengan HealthOmics menandai operasi API.

Kebijakan sebelumnya tidak mengizinkan pengguna untuk membuat peran IAM. Untuk pengguna dengan izin ini untuk menjalankan pekerjaan, administrator harus membuat peran layanan yang memberikan HealthOmics izin untuk mengakses sumber data. Untuk informasi selengkapnya, lihat Peran layanan untuk AWS HealthOmics.

Tentukan izin IAM kustom untuk menjalankan

Anda dapat menyertakan alur kerja, menjalankan, atau menjalankan grup apa pun yang direferensikan oleh `StartRun` permintaan dalam permintaan otorisasi. Untuk melakukannya, daftar kombinasi alur kerja, menjalankan, atau menjalankan grup yang diinginkan dalam kebijakan IAM. Misalnya, Anda dapat membatasi penggunaan alur kerja ke grup run atau run tertentu. Anda juga dapat menentukan bahwa alur kerja hanya digunakan dengan grup run.

Berikut ini adalah contoh kebijakan IAM yang memungkinkan alur kerja tunggal dengan grup run tunggal.

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/*"
            ]
        }
    ]
}
```

Peran layanan untuk AWS HealthOmics

Peran layanan adalah peran AWS Identity and Access Management (IAM) yang memberikan izin bagi AWS layanan untuk mengakses sumber daya di akun Anda. Anda memberikan peran layanan AWS HealthOmics saat memulai pekerjaan impor atau memulai proses.

HealthOmics Konsol dapat membuat peran yang diperlukan untuk Anda. Jika Anda menggunakan HealthOmics API untuk mengelola sumber daya, buat peran layanan menggunakan konsol IAM. Untuk informasi selengkapnya, lihat Membuat peran untuk mendelegasikan izin ke. Layanan AWS

Peran layanan harus memiliki kebijakan kepercayaan berikut.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "omics.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Kebijakan kepercayaan memungkinkan HealthOmics layanan untuk mengambil peran.

Topik

- Contoh kebijakan layanan IAM
- Contoh CloudFormation template

Contoh kebijakan layanan IAM

Dalam contoh ini, nama sumber daya dan akun IDs adalah placeholder untuk Anda ganti dengan nilai aktual.

Contoh berikut menunjukkan kebijakan untuk peran layanan yang dapat Anda gunakan untuk memulai proses. Kebijakan ini memberikan izin untuk mengakses lokasi keluaran Amazon S3, grup log alur kerja, dan container Amazon ECR untuk dijalankan.

> **Note**
>
> Jika Anda menggunakan caching panggilan untuk menjalankan, tambahkan cache jalankan lokasi Amazon S3 sebagai sumber daya di izin s3.

Example Kebijakan peran layanan untuk memulai proses

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/*"
            ]
        }
    ]
}
```

Contoh berikut menunjukkan kebijakan untuk peran layanan yang dapat Anda gunakan untuk pekerjaan impor toko. Kebijakan ini memberikan izin untuk mengakses lokasi input Amazon S3.

Example Peran layanan untuk pekerjaan toko Referensi

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"
            ]
        }

    ]
}
```

Contoh CloudFormation template

Contoh CloudFormation template berikut membuat peran layanan yang memberikan HealthOmics izin untuk mengakses bucket Amazon S3 yang memiliki nama yang diawali dengan`omics-`, dan untuk mengunggah log alur kerja.

Example Izin toko referensi, Amazon S3, dan CloudWatch Log

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

Izin Amazon ECR

Sebelum HealthOmics layanan dapat menjalankan alur kerja dalam wadah dari repositori Amazon ECR pribadi Anda, Anda membuat kebijakan sumber daya untuk repositori. Kebijakan memberikan izin kepada HealthOmics layanan untuk menggunakan kontainer. Anda menambahkan kebijakan sumber daya ini ke setiap repositori pribadi yang direferensikan oleh alur kerja.

> **Note**
>
> Repositori pribadi dan alur kerja harus berada di wilayah yang sama.

Jika AWS akun yang berbeda memiliki alur kerja dan repositori, Anda perlu mengonfigurasi izin lintas akun.

Anda tidak perlu memberikan akses repositori tambahan untuk alur kerja bersama. Namun, Anda dapat membuat kebijakan yang mengizinkan atau menolak akses alur kerja tertentu ke gambar kontainer.

Untuk menggunakan fitur cache Amazon ECR pull through, Anda perlu membuat kebijakan izin registri.

Bagian berikut menjelaskan cara mengonfigurasi izin sumber daya Amazon ECR untuk skenario ini. Untuk informasi selengkapnya tentang izin di Amazon ECR, lihat Izin registri pribadi di Amazon ECR.

Topik

Membuat kebijakan sumber daya untuk repositori Amazon ECR

Buat kebijakan sumber daya untuk memungkinkan HealthOmics layanan menjalankan alur kerja menggunakan wadah di repositori. Kebijakan ini memberikan izin kepada kepala HealthOmics layanan untuk mengakses tindakan ECR Amazon yang diperlukan.

Ikuti langkah-langkah ini untuk membuat kebijakan:

1. Buka halaman repositori pribadi di konsol Amazon ECR dan pilih repositori yang Anda berikan akses.
2. Dari navigasi bilah samping, pilih Izin.
3. Pilih Edit.
4. Pilih Edit kebijakan JSON.
5. Tambahkan pernyataan kebijakan berikut, lalu pilih Simpan.

   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": "*"
        }
    ]
}
```

Menjalankan alur kerja dengan kontainer lintas akun

Jika AWS akun yang berbeda memiliki alur kerja dan penampung, Anda perlu mengonfigurasi izin lintas akun berikut:

1. Perbarui kebijakan Amazon ECR untuk repositori untuk secara eksplisit memberikan izin ke akun yang memiliki alur kerja.
2. Perbarui peran layanan untuk akun yang memiliki alur kerja untuk memberinya akses ke image kontainer.

Contoh berikut menunjukkan kebijakan sumber daya Amazon ECR yang memberikan akses ke akun yang memiliki alur kerja.

Dalam contoh ini:

- ID akun alur kerja: 111122223333
- ID akun repositori kontainer: 444455556666
- Nama kontainer: 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": "*"
        }
    ]
}
```

Untuk menyelesaikan penyiapan, tambahkan pernyataan kebijakan berikut ke peran layanan akun yang memiliki alur kerja. Kebijakan memberikan izin untuk peran layanan untuk mengakses image kontainer “samtools”. Pastikan untuk mengganti nomor akun, nama kontainer, dan wilayah dengan nilai Anda sendiri.

```
{
    "Sid": "CrossAccountEcrRepoPolicy",
    "Effect": "Allow",
    "Action": ["ecr:BatchCheckLayerAvailability", "ecr:BatchGetImage",
 "ecr:GetDownloadUrlForLayer"],
    "Resource": "arn:aws:ecr:us-west-2:444455556666:repository/samtools"
}
```

Kebijakan Amazon ECR untuk alur kerja bersama

> **Note**
>
> HealthOmics secara otomatis memungkinkan alur kerja bersama untuk mengakses repositori Amazon ECR di akun pemilik alur kerja, sementara alur kerja berjalan di akun pelanggan. Anda tidak perlu memberikan akses repositori tambahan untuk alur kerja bersama. Untuk informasi selengkapnya, lihat Berbagi HealthOmics alur kerja.

Secara default, pelanggan tidak memiliki akses ke repositori Amazon ECR untuk menggunakan container yang mendasarinya. Secara opsional, Anda dapat menyesuaikan akses ke repositori Amazon ECR dengan menambahkan kunci kondisi ke kebijakan sumber daya repositori. Bagian berikut memberikan contoh kebijakan.

Batasi akses ke alur kerja tertentu

Anda dapat mencantumkan alur kerja individual dalam pernyataan kondisi, sehingga hanya alur kerja ini yang dapat menggunakan kontainer dalam repositori. Kunci SourceArnkondisi menentukan ARN dari alur kerja bersama. Contoh berikut memberikan izin untuk alur kerja yang ditentukan untuk menggunakan repositori ini.

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"
          }
       }
     }
  ]
}
```

Batasi akses ke akun tertentu

Anda dapat mencantumkan akun pelanggan dalam pernyataan kondisi, sehingga hanya akun ini yang memiliki izin untuk menggunakan kontainer di repositori. Kunci SourceAccountkondisi menentukan Akun AWS pelanggan. Contoh berikut memberikan izin untuk akun yang ditentukan untuk menggunakan repositori ini.

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"
        }
      }
    }
  ]
}
```

Anda juga dapat menolak izin Amazon ECR untuk pelanggan tertentu, seperti yang ditunjukkan dalam contoh kebijakan berikut.

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"
                }
            }
         }
```

```
        ]
    }
```

Kebijakan untuk Amazon ECR menarik cache

Untuk menggunakan cache pull through Amazon ECR, Anda membuat kebijakan izin registri. Anda juga membuat template pembuatan repositori, yang mendefinisikan izin untuk repositori yang dibuat oleh Amazon ECR pull through cache.

Bagian berikut mencakup contoh kebijakan ini. Untuk informasi selengkapnya tentang pull through cache, lihat Menyinkronkan registri upstream dengan registri pribadi Amazon ECR di Panduan Pengguna Amazon Elastic Container Registry.

Kebijakan izin registri

Untuk menggunakan Amazon ECR pull through cache, buat kebijakan izin registri. Kebijakan izin registri memberikan kontrol atas replikasi dan menarik izin cache.

Untuk replikasi lintas akun, Anda harus secara eksplisit mengizinkan masing-masing Akun AWS yang dapat mereplikasi repositori ke registri Anda.

Secara default, saat Anda membuat aturan cache pull through, prinsipal IAM apa pun yang memiliki izin untuk menarik gambar dari registri pribadi juga dapat menggunakan aturan pull through cache. Anda dapat menggunakan izin registri untuk memperluas cakupan izin ini ke repositori tertentu.

Tambahkan kebijakan izin registri ke akun yang memiliki gambar kontainer.

Dalam contoh berikut, kebijakan memungkinkan HealthOmics layanan untuk membuat repositori untuk setiap registri upstream dan untuk memulai permintaan tarik upstream dari repositori yang dibuat.

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/*"
            ]
        }
    ]
}
```

Template pembuatan repositori

Untuk menggunakan cache pull through HealthOmics, repositori Amazon ECR harus memiliki template pembuatan repositori. Template mendefinisikan pengaturan konfigurasi untuk repositori pribadi yang dibuat untuk registri upstream.

Setiap template berisi awalan namespace repositori, yang digunakan Amazon ECR untuk mencocokkan repositori baru dengan templat tertentu. Template dapat menentukan konfigurasi untuk semua pengaturan repositori termasuk kebijakan akses berbasis sumber daya, kekekalan tag, enkripsi, dan kebijakan siklus hidup. Untuk informasi selengkapnya, lihat Templat pembuatan repositori di Panduan Pengguna Amazon Elastic Container Registry.

Dalam contoh berikut, kebijakan memungkinkan HealthOmics layanan untuk memulai permintaan tarik upstream dari repositori upstream.

JSON

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

```
                "ecr:BatchGetImage",
                "ecr:GetDownloadUrlForLayer"
            ],
            "Resource": "*"
        }
    ]
}
```

Kebijakan untuk akses ECR Amazon lintas akun

Untuk akses lintas akun, pemilik repositori pribadi memperbarui kebijakan izin registri dan templat pembuatan repositori untuk mengizinkan akses ke akun lain dan peran jalankan akun tersebut.

Dalam kebijakan izin registri, tambahkan pernyataan kebijakan untuk mengizinkan peran jalankan akun lain mengakses tindakan ECR Amazon:

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/*"
        }
    ]
}
```

Dalam template pembuatan repositori, tambahkan pernyataan kebijakan untuk mengizinkan peran jalankan akun lain mengakses gambar kontainer baru. Secara opsional, Anda dapat menambahkan pernyataan kondisi untuk membatasi akses ke alur kerja tertentu:

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"
                }
            }
        }
    ]
}
```

Tambahkan izin untuk dua tindakan tambahan (CreateRepository dan BatchImportUpstreamImage) dalam peran run dan tentukan sumber daya yang dapat diakses oleh peran run.

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 Izin sumber daya

AWS HealthOmics membuat dan mengakses sumber daya di layanan lain atas nama Anda saat Anda menjalankan pekerjaan atau membuat toko. Dalam beberapa kasus, Anda perlu mengonfigurasi izin di layanan lain untuk mengakses sumber daya atau mengizinkan HealthOmics untuk mengaksesnya.

Untuk izin sumber daya yang terkait dengan Amazon ECR, lihat. Izin Amazon ECR

Izin Lake Formation

Sebelum Anda menggunakan fitur analitik HealthOmics, konfigurasikan pengaturan basis data default di Lake Formation.

Untuk mengonfigurasi izin sumber daya di Lake Formation

1. Buka halaman pengaturan katalog Data di konsol Lake Formation.
2. Hapus centang persyaratan kontrol akses IAM untuk database dan tabel di bawah Izin default untuk database dan tabel yang baru dibuat.
3. Pilih Simpan.

HealthOmics Analytics auto menerima data jika kebijakan layanan Anda memiliki izin RAM yang benar, seperti contoh berikut.

JSON

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

```
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "omics:*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ram:AcceptResourceShareInvitation",
        "ram:GetResourceShareInvitations"
      ],
      "Resource": "*"
    }
  ]
}
```

Izin untuk akses data menggunakan Amazon S3 URIs

Anda dapat mengakses data penyimpanan urutan menggunakan operasi HealthOmics API atau operasi Amazon S3 API.

Untuk akses HealthOmics API, HealthOmics izin dikelola melalui kebijakan IAM. Namun, S3 Access memerlukan dua tingkat konfigurasi: izin eksplisit dalam Kebijakan Akses S3 Store dan kebijakan IAM. Untuk mempelajari selengkapnya tentang menggunakan kebijakan IAM dengan HealthOmics, lihat Peran layanan untuk HealthOmics.

Ada tiga cara untuk berbagi kemampuan membaca objek menggunakan Amazon APIs S3:

1. Pembagian berbasis kebijakan - Berbagi ini mengharuskan pengaktifan prinsipal IAM baik dalam kebijakan Akses S3 dan menulis kebijakan IAM dan melampirkannya ke kepala IAM. Lihat topik berikutnya untuk lebih jelasnya.
2. Presigned URLs — Anda juga dapat membuat URL pra-ditandatangani yang dapat dibagikan untuk file di toko urutan. Untuk mempelajari selengkapnya tentang membuat presigned URLs menggunakan Amazon S3, lihat Menggunakan URLs presigned dalam dokumentasi Amazon S3. Kebijakan akses S3 penyimpanan urutan mendukung pernyataan untuk membatasi kemampuan URL yang telah ditetapkan sebelumnya.

3. Peran yang diasumsikan — Buat peran dalam akun pemilik data yang memiliki kebijakan akses yang memungkinkan pengguna untuk mengambil peran tersebut.

Topik

Berbagi berdasarkan kebijakan

Jika Anda mengakses data penyimpanan urutan menggunakan URI S3 langsung, HealthOmics berikan langkah-langkah keamanan yang disempurnakan untuk kebijakan akses bucket S3 terkait.

Aturan berikut berlaku untuk kebijakan akses S3 baru. Untuk kebijakan yang ada, aturan berlaku saat Anda memperbarui kebijakan berikutnya:

- Kebijakan akses S3 mendukung elemen kebijakan berikut
  - Versi, Id, Pernyataan, Sid, Efek, Prinsip, Tindakan, Sumber Daya, Kondisi
- Kebijakan akses S3 mendukung kunci kondisi berikut:
  - s3:ExistingObjectTag/<key>, s3: awalan, s3: signatureversion, s3: TlsVersion
  - Kebijakan juga mendukung aws: PrincipalArn dengan operator kondisi berikut: ArnEquals dan ArnLike

Jika Anda mencoba menambahkan atau memperbarui kebijakan untuk menyertakan elemen atau kondisi yang tidak didukung, sistem akan menolak permintaan tersebut.

Topik

Kebijakan akses S3 default

Saat Anda membuat penyimpanan urutan, HealthOmics buat kebijakan akses S3 default yang memberikan akun root pemilik penyimpanan data izin berikut untuk semua objek yang dapat diakses

di penyimpanan urutan: S3:, S3GetObject, dan S3GetObjectTagging:. ListBucket Kebijakan default yang dibuat adalah:

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/*"
        }
    ]
}
```

Menyesuaikan kebijakan akses

Jika kebijakan akses S3 kosong, akses S3 tidak diperbolehkan. Jika ada kebijakan yang ada dan Anda perlu menghapus akses s3, gunakan `deleteS3AccessPolicy` untuk menghapus semua akses.

Untuk menambahkan batasan pada berbagi atau memberikan akses ke akun lain, Anda dapat memperbarui kebijakan menggunakan `PutS3AccessPolicy` API. Pembaruan kebijakan tidak dapat melampaui awalan untuk penyimpanan urutan atau tindakan yang ditentukan.

Kebijakan IAM

Untuk mengizinkan pengguna atau akses utama IAM menggunakan Amazon APIs S3, selain izin dalam kebijakan akses S3, kebijakan IAM perlu dibuat dan dilampirkan ke prinsipal untuk memberikan akses. Kebijakan yang mengizinkan akses API Amazon S3 dapat diterapkan di tingkat penyimpanan urutan atau pada tingkat set baca. Pada tingkat set baca, izin dapat dibatasi baik melalui awalan atau menggunakan filter tag sumber daya untuk pola ID sampel atau subjek.

Jika toko urutan menggunakan kunci yang dikelola pelanggan (CMK), prinsipal juga harus memiliki hak untuk menggunakan kunci KMS untuk dekripsi. Untuk informasi selengkapnya, lihat Akses KMS lintas akun di Panduan AWS Key Management Service Pengembang.

Contoh berikut memberikan akses pengguna ke toko urutan. Anda dapat menyempurnakan akses dengan kondisi tambahan atau filter berbasis sumber daya.

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/*"
        }
      }
    }
  ]
}
```

Kontrol akses berbasis tanda

Untuk menggunakan kontrol akses berbasis tag, penyimpanan urutan harus diperbarui terlebih dahulu untuk menyebarkan kunci tag yang akan digunakan. Konfigurasi ini diatur selama pembuatan atau pembaruan penyimpanan urutan. Setelah tag disebarkan, kondisi tag dapat digunakan untuk menambahkan batasan lebih lanjut. Pembatasan dapat ditempatkan dalam kebijakan Akses S3 atau pada kebijakan IAM. Berikut ini adalah contoh kebijakan akses S3 berbasis tab yang akan ditetapkan:

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

Contoh Pembatasan

Skenario: Membuat share di mana pemilik data dapat membatasi kemampuan pengguna untuk mengunduh data “ditarik”.

Dalam skenario ini, pemilik data (akun #111111111111) mengelola penyimpanan data. Pemilik data ini membagikan data dengan berbagai pengguna pihak ketiga, termasuk peneliti (akun #999999999999). Sebagai bagian dari pengelolaan data, pemilik data secara berkala mendapatkan permintaan untuk menarik data peserta. Untuk mengelola penarikan ini, pemilik data pertama-tama membatasi akses unduhan langsung saat menerima permintaan dan akhirnya menghapus data sesuai kebutuhan mereka.

Untuk memenuhi kebutuhan ini, pemilik data menyiapkan penyimpanan urutan dan setiap set baca menerima tag untuk “status” yang akan disetel ke “ditarik” jika permintaan penarikan datang. Untuk data dengan tag yang disetel ke nilai ini, mereka ingin memastikan tidak ada pengguna yang dapat menjalankan “getObject” pada file ini. Untuk melakukan pengaturan ini, pemilik data perlu memastikan dua langkah diambil.

Langkah 1. Untuk penyimpanan urutan, pastikan bahwa tag status diperbarui untuk disebarkan. Ini dilakukan dengan menambahkan tombol “status” ke `propogatedSetLevelTags` saat memanggil `createSequenceStore` atau `updateSequenceStore.`

Langkah 2. Perbarui Kebijakan Akses s3 toko untuk membatasi getObject pada objek dengan tag status disetel ke ditarik. Ini dilakukan dengan memperbarui kebijakan akses toko menggunakan

`PutS3AccesPolicy` API. Kebijakan berikut akan memungkinkan pelanggan untuk tetap melihat file yang ditarik saat mencantumkan objek tetapi mencegah mereka mengaksesnya:

- Pernyataan 1 (restrictedGetWithdrawal): Akun 999999999999 tidak dapat mengambil objek yang ditarik.
- Pernyataan 2 (ownerGetAll): Akun 111111111111, pemilik data, dapat mengambil semua objek, termasuk objek yang ditarik.
- Pernyataan 3 (everyoneListAll): Semua akun bersama, 111111111111 dan 999999999999, dapat menjalankan operasi pada seluruh awalan. 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/*"
                }
            }
        }
    ]
```

```
}
```

Keamanan di AWS HealthOmics

Keamanan cloud di AWS adalah prioritas tertinggi. Sebagai AWS pelanggan, Anda mendapat manfaat dari pusat data dan arsitektur jaringan yang dibangun untuk memenuhi persyaratan organisasi yang paling sensitif terhadap keamanan.

Keamanan adalah tanggung jawab bersama antara Anda AWS dan Anda. Model tanggung jawab bersama menjelaskan hal ini sebagai keamanan dari cloud dan keamanan dalam cloud:

- Keamanan cloud — AWS bertanggung jawab untuk melindungi infrastruktur yang menjalankan AWS layanan di AWS Cloud. AWS juga memberi Anda layanan yang dapat Anda gunakan dengan aman. Auditor pihak ketiga secara teratur menguji dan memverifikasi efektivitas keamanan kami sebagai bagian dari Program AWS Kepatuhan Program AWS Kepatuhan . Untuk mempelajari tentang program kepatuhan yang berlaku untuk AWS HealthOmics, lihat AWS Layanan dalam Lingkup oleh AWS Layanan Program Kepatuhan .
- Keamanan di cloud — Tanggung jawab Anda ditentukan oleh AWS layanan yang Anda gunakan. Anda juga bertanggung jawab atas faktor lain, yang mencakup sensitivitas data Anda, persyaratan perusahaan Anda, serta undang-undang dan peraturan yang berlaku.

Dokumentasi ini membantu Anda memahami cara menerapkan model tanggung jawab bersama saat menggunakan AWS HealthOmics. Topik berikut menunjukkan cara mengonfigurasi AWS HealthOmics untuk memenuhi tujuan keamanan dan kepatuhan Anda. Anda juga mempelajari cara menggunakan AWS layanan lain yang membantu Anda memantau dan mengamankan HealthOmics sumber daya AWS Anda.

Topik

Perlindungan data di AWS HealthOmics

Model tanggung jawab AWS bersama model tanggung berlaku untuk perlindungan data di AWS HealthOmics. Seperti yang dijelaskan dalam model AWS ini, bertanggung jawab untuk melindungi infrastruktur global yang menjalankan semua AWS Cloud. Anda bertanggung jawab untuk mempertahankan kendali atas konten yang di-host pada infrastruktur ini. Anda juga bertanggung jawab atas tugas-tugas konfigurasi dan manajemen keamanan untuk Layanan AWS yang Anda gunakan. Lihat informasi yang lebih lengkap tentang privasi data dalam Pertanyaan Umum Privasi Data. Lihat informasi tentang perlindungan data di Eropa di pos blog Model Tanggung Jawab Bersama dan GDPR AWS di Blog Keamanan AWS .

Untuk tujuan perlindungan data, kami menyarankan Anda melindungi Akun AWS kredensyal dan mengatur pengguna individu dengan AWS IAM Identity Center atau AWS Identity and Access Management (IAM). Dengan cara itu, setiap pengguna hanya diberi izin yang diperlukan untuk memenuhi tanggung jawab tugasnya. Kami juga menyarankan supaya Anda mengamankan data dengan cara-cara berikut:

- Gunakan autentikasi multi-faktor (MFA) pada setiap akun.
- Gunakan SSL/TLS untuk berkomunikasi dengan AWS sumber daya. Kami mensyaratkan TLS 1.2 dan menganjurkan TLS 1.3.
- Siapkan API dan pencatatan aktivitas pengguna dengan AWS CloudTrail. Untuk informasi tentang penggunaan CloudTrail jejak untuk menangkap AWS aktivitas, lihat Bekerja dengan CloudTrail jejak di AWS CloudTrail Panduan Pengguna.
- Gunakan solusi AWS enkripsi, bersama dengan semua kontrol keamanan default di dalamnya Layanan AWS.
- Gunakan layanan keamanan terkelola tingkat lanjut seperti Amazon Macie, yang membantu menemukan dan mengamankan data sensitif yang disimpan di Amazon S3.
- Jika Anda memerlukan modul kriptografi tervalidasi FIPS 140-3 saat mengakses AWS melalui antarmuka baris perintah atau API, gunakan titik akhir FIPS. Lihat informasi selengkapnya tentang titik akhir FIPS yang tersedia di Standar Pemrosesan Informasi Federal (FIPS) 140-3.

Kami sangat merekomendasikan agar Anda tidak pernah memasukkan informasi identifikasi yang sensitif, seperti nomor rekening pelanggan Anda, ke dalam tanda atau bidang isian bebas seperti bidang Nama. Ini termasuk saat Anda bekerja dengan AWS HealthOmics atau lainnya Layanan AWS menggunakan konsol, API AWS CLI, atau AWS SDKs. Data apa pun yang Anda masukkan ke dalam tanda atau bidang isian bebas yang digunakan untuk nama dapat digunakan untuk log penagihan

atau log diagnostik. Saat Anda memberikan URL ke server eksternal, kami sangat menganjurkan supaya Anda tidak menyertakan informasi kredensial di dalam URL untuk memvalidasi permintaan Anda ke server itu.

Enkripsi saat diam

Topik

Untuk melindungi data pelanggan yang sensitif saat istirahat, AWS HealthOmics berikan enkripsi secara default menggunakan kunci AWS Key Management Service (AWS KMS) milik layanan. Kunci yang dikelola pelanggan juga didukung. Untuk mempelajari selengkapnya tentang kunci yang dikelola pelanggan, lihat Amazon Key Management Service.

Semua penyimpanan HealthOmics data (Storage and Analytics) mendukung penggunaan kunci yang dikelola pelanggan. Konfigurasi enkripsi tidak dapat diubah setelah penyimpanan data dibuat. Jika penyimpanan data menggunakan Kunci milik AWS, itu akan dilambangkan sebagai AWS_OWNED_KMS_KEY dan Anda tidak akan melihat kunci spesifik yang digunakan untuk enkripsi saat istirahat.

Untuk HealthOmics Alur Kerja, kunci yang dikelola pelanggan tidak didukung oleh sistem file sementara; namun, semua data dienkripsi saat istirahat secara otomatis menggunakan algoritme enkripsi sandi blok XTS-AES-256 untuk mengenkripsi sistem file. Pengguna dan peran IAM yang digunakan untuk memulai alur kerja juga harus memiliki akses ke AWS KMS kunci yang digunakan untuk bucket input dan output alur kerja. Alur kerja tidak menggunakan hibah, dan AWS KMS enkripsi terbatas pada bucket Amazon S3 input dan output. Peran IAM yang digunakan baik untuk alur kerja juga APIs harus memiliki akses ke AWS KMS kunci yang digunakan serta bucket input dan output Amazon S3. Anda dapat menggunakan peran dan izin IAM untuk mengontrol akses atau AWS KMS kebijakan. Untuk mempelajari lebih lanjut, lihat Otentikasi dan kontrol akses untuk AWS KMS.

Saat Anda menggunakan AWS Lake Formation HealthOmics Analytics, izin dekripsi apa pun yang terkait dengan Lake Formation juga diberikan ke bucket Amazon S3 input dan output. Informasi lebih

lanjut tentang cara AWS Lake Formation mengelola izin dapat ditemukan di AWS Lake Formation dokumentasi.

HealthOmics Analytics memberikan izin kms: Dekripsi Lake Formation untuk membaca data terenkripsi di bucket Amazon S3. Selama Anda memiliki izin untuk menanyakan data melalui Lake Formation, Anda akan dapat membaca data terenkripsi. Akses ke data dikendalikan melalui kontrol akses data di Lake Formation, bukan melalui kebijakan kunci KMS. Untuk mempelajari lebih lanjut, lihat permintaan layanan AWS AWS Terpadu dalam dokumentasi Lake Formation.

Kunci milik AWS

Secara default, HealthOmics digunakan Kunci milik AWS untuk mengenkripsi data secara otomatis saat istirahat, karena data ini dapat berisi informasi sensitif seperti informasi identitas pribadi (PII) atau Informasi Kesehatan yang Dilindungi (PHI). Kunci milik AWS tidak disimpan di akun Anda. Mereka adalah bagian dari kumpulan kunci KMS yang dimiliki dan dikelola AWS untuk digunakan di beberapa akun AWS.

Layanan AWS dapat digunakan Kunci milik AWS untuk melindungi data Anda. Anda tidak dapat melihat, mengelola, atau mengakses Kunci milik AWS, atau mengaudit penggunaannya. Namun, Anda tidak perlu melakukan pekerjaan apa pun atau mengubah program apa pun untuk melindungi kunci yang mengenkripsi data Anda.

Anda tidak dikenakan biaya bulanan atau biaya penggunaan untuk penggunaan Kunci milik AWS, dan biaya tersebut tidak dihitung terhadap kuota AWS KMS untuk akun Anda. Untuk informasi selengkapnya, lihat Kunci yang dikelola AWS.

Kunci yang dikelola pelanggan

HealthOmics mendukung penggunaan kunci terkelola pelanggan simetris yang Anda buat, miliki, dan kelola untuk menambahkan enkripsi lapisan kedua melalui enkripsi milik AWS yang ada. Karena Anda memiliki kontrol penuh atas lapisan enkripsi ini, Anda dapat melakukan tugas-tugas seperti:

- Menetapkan dan memelihara kebijakan utama, kebijakan IAM, dan hibah
- Memutar bahan kriptografi kunci
- Mengaktifkan dan menonaktifkan kebijakan utama
- Menambahkan tanda
- Membuat alias kunci
- Kunci penjadwalan untuk penghapusan

Anda juga dapat menggunakan CloudTrail untuk melacak permintaan yang HealthOmics dikirim AWS KMS atas nama Anda. AWS KMS Biaya tambahan berlaku. Untuk informasi selengkapnya, lihat kunci terkelola pelanggan.

Membuat kunci yang dikelola pelanggan

Anda dapat membuat kunci terkelola pelanggan simetris dengan menggunakan AWS Management Console, atau. AWS KMS APIs

Ikuti langkah-langkah untuk Membuat kunci terkelola pelanggan simetris di Panduan Pengembang AWS Key Management Service.

Kebijakan utama mengontrol akses ke kunci yang dikelola pelanggan Anda. Setiap kunci yang dikelola pelanggan harus memiliki persis satu kebijakan utama, yang berisi pernyataan yang menentukan siapa yang dapat menggunakan kunci dan bagaimana mereka dapat menggunakannya. Saat membuat kunci terkelola pelanggan, Anda dapat menentukan kebijakan kunci. Untuk informasi selengkapnya, lihat Mengelola akses ke kunci terkelola pelanggan di Panduan Pengembang AWS Key Management Service.

Untuk menggunakan kunci yang dikelola pelanggan dengan sumber daya HealthOmics Analytics Anda, prinsipal panggilan memerlukan kms: CreateGrant operasi dalam kebijakan utama. Ini memungkinkan sistem untuk menggunakan Token FAS untuk membuat hibah ke kunci yang dikelola pelanggan yang mengontrol akses ke kunci KMS tertentu. Kunci ini memberi pengguna akses ke operasi kms:grant yang membutuhkan. HealthOmics Lihat Menggunakan hibah untuk informasi lebih lanjut.

Untuk HealthOmics analitik, operasi API berikut harus diizinkan untuk prinsipal panggilan:

- kms: CreateGrant menambahkan hibah ke kunci terkelola pelanggan tertentu, yang memungkinkan akses ke operasi hibah di HealthOmics Analytics.
- kms: DescribeKey memberikan detail kunci yang dikelola pelanggan yang diperlukan untuk memvalidasi kunci. Ini diperlukan untuk semua operasi.
- kms: GenerateDataKey menyediakan akses untuk mengenkripsi sumber daya saat istirahat untuk semua operasi penulisan. Selain itu, tindakan ini memberikan detail kunci yang dikelola pelanggan yang dapat digunakan layanan untuk memvalidasi bahwa penelepon memiliki akses untuk menggunakan kunci.
- KMS: Decrypt menyediakan akses untuk membaca atau mencari operasi untuk sumber daya terenkripsi.

Untuk menggunakan kunci yang dikelola pelanggan dengan sumber daya HealthOmics penyimpanan Anda, prinsipal HealthOmics layanan dan prinsipal panggilan harus diizinkan dalam kebijakan kunci. Hal ini memungkinkan layanan untuk memvalidasi bahwa penelepon memiliki akses ke kunci dan menggunakan prinsip layanan untuk menjalankan manajemen toko menggunakan kunci yang dikelola pelanggan. Untuk HealthOmics penyimpanan, kebijakan utama untuk prinsipal layanan harus mengizinkan operasi API berikut:

- kms: DescribeKey memberikan detail kunci yang dikelola pelanggan yang diperlukan untuk memvalidasi kunci. Ini diperlukan untuk semua operasi.
- kms: GenerateDataKey menyediakan akses untuk mengenkripsi sumber daya saat istirahat untuk semua operasi penulisan. Selain itu, tindakan ini memberikan detail kunci yang dikelola pelanggan yang dapat digunakan layanan untuk memvalidasi bahwa penelepon memiliki akses untuk menggunakan kunci.
- KMS: Decrypt menyediakan akses untuk membaca atau mencari operasi untuk sumber daya terenkripsi.

Contoh berikut menunjukkan pernyataan kebijakan yang memungkinkan prinsipal layanan untuk membuat dan berinteraksi dengan toko HealthOmics urutan atau referensi yang dienkripsi menggunakan kunci yang dikelola pelanggan:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
{
    "Effect": "Allow",
    "Principal": {
        "Service": "omics.amazonaws.com"
    },
    "Action": [
        "kms:Decrypt",
        "kms:DescribeKey",
        "kms:Encrypt",
        "kms:GenerateDataKey*"
    ],
    "Resource": "*"
        }
    ]
```

```
}
```

Contoh berikut menunjukkan kebijakan yang membuat izin untuk penyimpanan data untuk mendekripsi data dari bucket Amazon S3.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "omics:GetReference",
                "omics:GetReferenceMetadata"
            ],
            "Resource": [
                "arn:aws:omics:us-east-1:123456789012:referenceStore/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::[[s3path]]/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:123456789012:key/key_id"
            ],
            "Condition": {
              "StringEquals": {
                  "kms:ViaService": [
                    "s3.us-east-1.amazonaws.com"
```

```
                ]
            }
        }
    ]
}
```

Izin IAM yang diperlukan untuk menggunakan kunci terkelola pelanggan

Saat membuat sumber daya seperti penyimpanan data dengan AWS KMS enkripsi menggunakan kunci yang dikelola pelanggan, ada izin yang diperlukan untuk kebijakan kunci dan kebijakan IAM untuk pengguna atau peran IAM.

Anda dapat menggunakan kms: ViaService condition key untuk membatasi penggunaan kunci KMS hanya untuk permintaan yang berasal dari. HealthOmics

Untuk informasi selengkapnya tentang kebijakan utama, lihat Mengaktifkan kebijakan IAM di Panduan Pengembang AWS Key Management Service.

Topik

- Izin API Analytics
- Izin API penyimpanan
- Cara HealthOmics menggunakan hibah di AWS KMS
- Memantau kunci enkripsi Anda untuk AWS HealthOmics

Izin API Analytics

Untuk HealthOmics analitik, pengguna atau peran IAM yang membuat toko harus memiliki, kms: Dekripsi kms: CreateGrant kms:GenerateDataKey, dan kms: DescribeKey izin ditambah izin yang diperlukan. HealthOmics

Izin API penyimpanan

Untuk HealthOmics penyimpanan APIs, pengguna IAM atau peran yang memanggil operasi API berikut memerlukan izin yang tercantum:

CreateReferenceStore, CreateSequenceStore

Untuk membuat toko, pemanggil IAM harus memiliki `kms:DescribeKey` izin ditambah izin yang diperlukan. HealthOmics Prinsipal HealthOmics layanan memanggil `kms:GenerateDataKeyWithoutPlaintext` untuk melakukan pemeriksaan validasi akses untuk pemuatan dan akses data.

StartReadSetImportJob, StartReferenceImportJob

Untuk memulai pekerjaan impor data, pemanggil IAM harus memiliki `kms:Decrypt` dan `kms:GenerateDataKey` izin untuk kunci KMS di toko untuk impor, dan `kms:Decrypt` izin di bucket Amazon S3 yang berisi objek yang akan diimpor. Selain itu, peran yang diteruskan ke panggilan harus memiliki `kms:Decrypt` izin di bucket Amazon S3 yang berisi objek yang akan diimpor. Penelepon IAM juga harus memiliki izin untuk meneruskan peran ke pekerjaan.

CreateMultipartReadSetUpload, UploadReadSetPart, CompleteMultipartReadSetUpload

Untuk menyelesaikan unggahan multi-bagian, penelepon IAM harus memiliki `kms:Decrypt` dan `kms:GenerateDataKey` membuat, mengunggah, dan menyelesaikan unggahan multi-bagian.

StartReadSetExportJob

Untuk memulai pekerjaan ekspor data, pemanggil IAM harus memiliki `kms:Decrypt` izin untuk kunci KMS di toko untuk mengekspor dari dan `kms:GenerateDataKey` dan `kms:Decrypt` izin pada bucket Amazon S3 yang menerima objek. Selain itu, peran yang diteruskan ke panggilan harus memiliki `kms:Decrypt` izin di bucket Amazon S3 yang menerima objek. Penelepon IAM juga harus memiliki izin untuk meneruskan peran ke pekerjaan.

StartReadsetActivationJob

Untuk memulai pekerjaan aktivasi set baca, pemanggil IAM harus memiliki `kms:Decrypt` dan `kms:GenerateDataKey` izin untuk objek.

GetReference, GetReadSet

Untuk membaca objek dari toko, pemanggil IAM harus memiliki `kms:Decrypt` izin untuk objek.

Baca Set S3 GetObject

Untuk membaca objek dari toko menggunakan Amazon S3 `GetObject` API, pemanggil IAM harus memiliki `kms:Decrypt` izin untuk objek. Tetapkan izin ini untuk kunci dan Kunci milik AWS konfigurasi yang dikelola pelanggan.

Cara HealthOmics menggunakan hibah di AWS KMS

HealthOmics Analytics memerlukan hibah untuk menggunakan kunci KMS yang dikelola pelanggan Anda. Hibah tidak diperlukan atau digunakan untuk HealthOmics Alur Kerja. HealthOmics Penyimpanan menggunakan kunci yang dikelola pelanggan langsung dari kepala layanan, jadi jangan gunakan hibah. Saat Anda membuat toko analitik yang dienkripsi dengan kunci yang dikelola pelanggan, HealthOmics analitik akan membuat hibah atas nama Anda dengan mengirimkan CreateGrantpermintaan ke AWS KMS. Hibah di AWS KMS digunakan untuk HealthOmics memberikan akses ke kunci KMS di akun pelanggan.

Tidak disarankan untuk mencabut atau menghentikan hibah yang dibuat HealthOmics analitik atas nama Anda. Jika Anda mencabut atau menghentikan hibah yang memberikan HealthOmics izin untuk menggunakan kunci AWS KMS di akun Anda, HealthOmics tidak dapat mengakses data ini, mengenkripsi sumber daya baru yang didorong ke penyimpanan data, atau mendekripsi ketika ditarik.

Ketika Anda mencabut atau pensiun hibah untuk HealthOmics, perubahan terjadi segera. Untuk mencabut hak akses, kami sarankan Anda menghapus penyimpanan data daripada mencabut hibah. Saat Anda menghapus penyimpanan data, HealthOmics pensiun hibah atas nama Anda.

Memantau kunci enkripsi Anda untuk AWS HealthOmics

Anda dapat menggunakan CloudTrail untuk melacak permintaan yang AWS HealthOmics dikirim atas nama Anda saat menggunakan kunci yang dikelola pelanggan. AWS KMS Entri log di CloudTrail log HealthOmics menampilkan.amazonaws.com di bidang UserAgent untuk membedakan dengan jelas permintaan yang dibuat oleh. HealthOmics

Contoh berikut adalah CloudTrail peristiwa untuk CreateGrant,, Dekripsi GenerateDataKey, dan DescribeKey untuk memantau AWS KMS operasi yang dipanggil oleh HealthOmics untuk mengakses data yang dienkripsi oleh kunci yang dikelola pelanggan Anda.

Berikut ini juga menunjukkan cara menggunakan CreateGrant untuk memungkinkan HealthOmics analitik mengakses kunci KMS yang disediakan pelanggan, memungkinkan HealthOmics untuk menggunakan kunci KMS itu untuk mengenkripsi semua data pelanggan saat istirahat.

Anda tidak diharuskan untuk membuat hibah Anda sendiri. HealthOmics membuat hibah atas nama Anda dengan mengirimkan CreateGrant permintaan ke AWS KMS. Hibah AWS KMS digunakan untuk memberikan HealthOmics akses ke AWS KMS kunci di akun pelanggan.

```
{
    "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"
}
```

Contoh berikut menunjukkan cara menggunakan GenerateDataKey untuk memastikan pengguna memiliki izin yang diperlukan untuk mengenkripsi data sebelum menyimpannya.

```
{
    "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"
}
```


Pelajari selengkapnya

Sumber daya berikut memberikan informasi lebih lanjut tentang enkripsi data saat istirahat.

Untuk informasi selengkapnya tentang konsep dasar AWS Key Management Service, lihat AWS KMS dokumentasinya.

Untuk informasi selengkapnya tentang praktik terbaik Keamanan dalam AWS KMS dokumentasi.

Enkripsi saat bergerak

AWS HealthOmics menggunakan TLS 1.2+ untuk mengenkripsi data dalam perjalanan melalui titik akhir publik dan melalui layanan backend.

Manajemen identitas dan akses di HealthOmics

AWS Identity and Access Management (IAM) adalah Layanan AWS yang membantu administrator mengontrol akses ke AWS sumber daya dengan aman. Administrator IAM mengontrol siapa yang dapat diautentikasi (masuk) dan diberi wewenang (memiliki izin) untuk menggunakan sumber daya AWS. HealthOmics IAM adalah Layanan AWS yang dapat Anda gunakan tanpa biaya tambahan.

Topik

Audiens

Cara Anda menggunakan AWS Identity and Access Management (IAM) berbeda berdasarkan peran Anda:

- Pengguna layanan - minta izin dari administrator Anda jika Anda tidak dapat mengakses fitur (lihat Memecahkan masalah AWS HealthOmics identitas dan akses)
- Administrator layanan - tentukan akses pengguna dan mengirimkan permintaan izin (lihat Bagaimana AWS HealthOmics bekerja dengan IAM)
- Administrator IAM - tulis kebijakan untuk mengelola akses (lihat Contoh kebijakan berbasis identitas untuk AWS HealthOmics)

Mengautentikasi dengan identitas

Otentikasi adalah cara Anda masuk AWS menggunakan kredensi identitas Anda. Anda harus diautentikasi sebagai Pengguna root akun AWS, pengguna IAM, atau dengan mengasumsikan peran IAM.

Anda dapat masuk sebagai identitas federasi menggunakan kredensyal dari sumber identitas seperti AWS IAM Identity Center (Pusat Identitas IAM), autentikasi masuk tunggal, atau kredensyal. Google/ Facebook Untuk informasi selengkapnya tentang cara masuk, lihat Cara masuk ke Akun AWS Anda dalam Panduan Pengguna AWS Sign-In .

Untuk akses terprogram, AWS sediakan SDK dan CLI untuk menandatangani permintaan secara kriptografis. Untuk informasi selengkapnya, lihat AWS Signature Version 4 untuk permintaan API dalam Panduan Pengguna IAM.

Akun AWS pengguna root

Saat Anda membuat Akun AWS, Anda mulai dengan satu identitas masuk yang disebut pengguna Akun AWS root yang memiliki akses lengkap ke semua Layanan AWS dan sumber daya. Kami sangat menyarankan agar Anda tidak menggunakan pengguna root untuk tugas sehari-hari. Untuk tugas yang memerlukan kredensial pengguna root, lihat Tugas yang memerlukan kredensial pengguna root dalam Panduan Pengguna IAM.

Identitas terfederasi

Sebagai praktik terbaik, mewajibkan pengguna manusia untuk menggunakan federasi dengan penyedia identitas untuk mengakses Layanan AWS menggunakan kredensyal sementara.

Identitas federasi adalah pengguna dari direktori perusahaan Anda, penyedia identitas web, atau Directory Service yang mengakses Layanan AWS menggunakan kredensyal dari sumber identitas. Identitas terfederasi mengambil peran yang memberikan kredensial sementara.

Untuk manajemen akses terpusat, kami menyarankan AWS IAM Identity Center. Untuk informasi selengkapnya, lihat Apa itu Pusat Identitas IAM? dalam Panduan Pengguna AWS IAM Identity Center .

Pengguna dan grup IAM

Pengguna IAM adalah identitas dengan izin khusus untuk satu orang atau aplikasi. Sebaiknya gunakan kredensial sementara alih-alih pengguna IAM dengan kredensial jangka panjang. Untuk informasi selengkapnya, lihat Mewajibkan pengguna manusia untuk menggunakan federasi dengan penyedia identitas untuk mengakses AWS menggunakan kredensyal sementara di Panduan Pengguna IAM.

Grup IAM menentukan kumpulan pengguna IAM dan mempermudah pengelolaan izin untuk pengguna dalam jumlah besar. Untuk mempelajari selengkapnya, lihat Kasus penggunaan untuk pengguna IAM dalam Panduan Pengguna IAM.

Peran IAM

Peran IAM adalah identitas dengan izin khusus yang menyediakan kredensial sementara. Anda dapat mengambil peran dengan beralih dari pengguna ke peran IAM (konsol) atau dengan memanggil operasi AWS CLI atau AWS API. Untuk informasi selengkapnya, lihat Metode untuk mengambil peran dalam Panduan Pengguna IAM.

Peran IAM berguna untuk akses pengguna gabungan, izin pengguna IAM sementara, akses lintas akun, akses lintas layanan, dan aplikasi yang berjalan di Amazon. EC2 Untuk informasi selengkapnya, lihat Akses sumber daya lintas akun di IAM dalam Panduan Pengguna IAM.

Mengelola akses menggunakan kebijakan

Anda mengontrol akses AWS dengan membuat kebijakan dan melampirkannya ke AWS identitas atau sumber daya. Kebijakan menentukan izin saat dikaitkan dengan identitas atau sumber daya. AWS mengevaluasi kebijakan ini ketika kepala sekolah membuat permintaan. Sebagian besar kebijakan disimpan AWS sebagai dokumen JSON. Untuk informasi selengkapnya tentang dokumen kebijakan JSON, lihat Gambaran umum kebijakan JSON dalam Panduan Pengguna IAM.

Menggunakan kebijakan, administrator menentukan siapa yang memiliki akses ke apa dengan mendefinisikan principal mana yang dapat melakukan tindakan pada sumber daya apa, dan dalam kondisi apa.

Secara default, pengguna dan peran tidak memiliki izin. Administrator IAM membuat kebijakan IAM dan menambahkannya ke peran, yang kemudian dapat diambil oleh pengguna. Kebijakan IAM mendefinisikan izin terlepas dari metode yang Anda gunakan untuk melakukan operasinya.

Kebijakan berbasis identitas

Kebijakan berbasis identitas adalah dokumen kebijakan izin JSON yang Anda lampirkan ke identitas (pengguna, grup, atau peran). Kebijakan ini mengontrol tindakan apa yang bisa dilakukan oleh identitas tersebut, terhadap sumber daya yang mana, dan dalam kondisi apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat Tentukan izin IAM kustom dengan kebijakan yang dikelola pelanggan dalam Panduan Pengguna IAM.

Kebijakan berbasis identitas dapat berupa kebijakan inline (disematkan langsung ke dalam satu identitas) atau kebijakan terkelola (kebijakan mandiri yang dilampirkan pada banyak identitas). Untuk mempelajari cara memilih antara kebijakan terkelola dan kebijakan inline, lihat Pilih antara kebijakan terkelola dan kebijakan inline dalam Panduan Pengguna IAM.

Kebijakan berbasis sumber daya

Kebijakan berbasis sumber daya adalah dokumen kebijakan JSON yang Anda lampirkan ke sumber daya. Contohnya termasuk kebijakan kepercayaan peran IAM dan kebijakan bucket Amazon S3. Dalam layanan yang mendukung kebijakan berbasis sumber daya, administrator layanan dapat menggunakannya untuk mengontrol akses ke sumber daya tertentu. Anda harus menentukan principal dalam kebijakan berbasis sumber daya.

Kebijakan berbasis sumber daya merupakan kebijakan inline yang terletak di layanan tersebut. Anda tidak dapat menggunakan kebijakan AWS terkelola dari IAM dalam kebijakan berbasis sumber daya.

Jenis-jenis kebijakan lain

AWS mendukung jenis kebijakan tambahan yang dapat menetapkan izin maksimum yang diberikan oleh jenis kebijakan yang lebih umum:

- Batasan izin – Menetapkan izin maksimum yang dapat diberikan oleh kebijakan berbasis identitas kepada entitas IAM. Untuk informasi selengkapnya, lihat Batasan izin untuk entitas IAM dalam Panduan Pengguna IAM.
- Kebijakan kontrol layanan (SCPs) — Tentukan izin maksimum untuk organisasi atau unit organisasi di AWS Organizations. Untuk informasi selengkapnya, lihat Kebijakan kontrol layanan dalam Panduan Pengguna AWS Organizations .
- Kebijakan kontrol sumber daya (RCPs) — Tetapkan izin maksimum yang tersedia untuk sumber daya di akun Anda. Untuk informasi selengkapnya, lihat Kebijakan kontrol sumber daya (RCPs) di Panduan AWS Organizations Pengguna.
- Kebijakan sesi – Kebijakan lanjutan yang diteruskan sebagai parameter saat membuat sesi sementara untuk peran atau pengguna terfederasi. Untuk informasi selengkapnya, lihat Kebijakan sesi dalam Panduan Pengguna IAM.

Berbagai jenis kebijakan

Ketika beberapa jenis kebijakan berlaku pada suatu permintaan, izin yang dihasilkan lebih rumit untuk dipahami. Untuk mempelajari cara AWS menentukan apakah akan mengizinkan permintaan saat beberapa jenis kebijakan terlibat, lihat Logika evaluasi kebijakan di Panduan Pengguna IAM.

Bagaimana AWS HealthOmics bekerja dengan IAM

Sebelum Anda menggunakan IAM untuk mengelola akses ke AWS HealthOmics, pelajari fitur IAM apa yang tersedia untuk digunakan dengan AWS. HealthOmics

Fitur IAM yang dapat Anda gunakan AWS HealthOmics

Fitur IAM	HealthOmics dukungan
Kebijakan berbasis identitas	Ya
Kebijakan berbasis sumber daya	Tidak
Tindakan kebijakan	Ya
Sumber daya kebijakan	Ya
Kunci kondisi kebijakan	Tidak
ACLs	Tidak
ABAC (tag dalam kebijakan)	Ya
Kredensial sementara	Ya
Izin principal	Ya
Peran layanan	Ya
Peran terkait layanan	Tidak

Untuk mendapatkan tampilan tingkat tinggi tentang cara HealthOmics dan AWS layanan lain bekerja dengan sebagian besar fitur IAM, lihat AWS layanan yang bekerja dengan IAM di Panduan Pengguna IAM.

Pencegahan "confused deputy" lintas layanan

Masalah "confused deputy" adalah masalah keamanan di mana entitas yang tidak memiliki izin untuk melakukan tindakan dapat memengaruhi entitas yang memiliki hak akses lebih tinggi untuk

melakukan tindakan. Pada tahun AWS, peniruan lintas layanan dapat mengakibatkan masalah wakil yang membingungkan. Peniruan identitas lintas layanan dapat terjadi ketika satu layanan (layanan yang dipanggil) memanggil layanan lain (layanan yang dipanggil). Layanan panggilan dapat dimanipulasi untuk menggunakan izinnya untuk bertindak atas sumber daya pelanggan lain dengan cara yang seharusnya tidak memiliki izin untuk mengakses. Untuk mencegah hal ini, AWS menyediakan alat yang membantu Anda melindungi data untuk semua layanan dengan principal layanan yang telah diberi akses ke sumber daya di akun Anda.

Sebaiknya gunakan kunci konteks kondisi `aws:SourceAccount`global `aws:SourceArn`dan global dalam kebijakan sumber daya untuk membatasi izin yang HealthOmics diberikan AWS layanan lain ke sumber daya.

Untuk mencegah masalah wakil yang membingungkan dalam peran yang diasumsikan oleh HealthOmics, tetapkan nilai `aws:SourceArn` ke `arn:aws:omics:`*`region`*`:`*`accountNumber`*`:*` dalam kebijakan kepercayaan peran. Wildcard (`*`) menerapkan kondisi untuk semua HealthOmics sumber daya.

Kebijakan hubungan kepercayaan berikut memberikan HealthOmics akses ke sumber daya Anda dan menggunakan kunci konteks kondisi `aws:SourceAccount` global `aws:SourceArn` dan global untuk mencegah masalah wakil yang membingungkan. Gunakan kebijakan ini saat Anda membuat peran 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:*"
        }
      }
    }
  ]
}
```

Kebijakan berbasis identitas untuk HealthOmics

Mendukung kebijakan berbasis identitas: Ya

Kebijakan berbasis identitas adalah dokumen kebijakan izin JSON yang dapat Anda lampirkan ke sebuah identitas, seperti pengguna IAM, grup pengguna IAM, atau peran IAM. Kebijakan ini mengontrol jenis tindakan yang dapat dilakukan oleh pengguna dan peran, di sumber daya mana, dan berdasarkan kondisi seperti apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat Tentukan izin IAM kustom dengan kebijakan terkelola pelanggan dalam Panduan Pengguna IAM.

Dengan kebijakan berbasis identitas IAM, Anda dapat menentukan secara spesifik apakah tindakan dan sumber daya diizinkan atau ditolak, serta kondisi yang menjadi dasar dikabulkan atau ditolaknya tindakan tersebut. Untuk mempelajari semua elemen yang dapat Anda gunakan dalam kebijakan JSON, lihat Referensi elemen kebijakan JSON IAM dalam Panduan Pengguna IAM.

Contoh kebijakan berbasis identitas untuk HealthOmics

Untuk melihat contoh kebijakan HealthOmics berbasis identitas AWS, lihat. Contoh kebijakan berbasis identitas untuk AWS HealthOmics

Kebijakan berbasis sumber daya dalam HealthOmics

Mendukung kebijakan berbasis sumber daya: Tidak

Kebijakan berbasis sumber daya adalah dokumen kebijakan JSON yang Anda lampirkan ke sumber daya. Contoh kebijakan berbasis sumber daya adalah kebijakan kepercayaan peran IAM dan kebijakan bucket Amazon S3. Dalam layanan yang mendukung kebijakan berbasis sumber daya, administrator layanan dapat menggunakannya untuk mengontrol akses ke sumber daya tertentu. Untuk sumber daya tempat kebijakan dilampirkan, kebijakan menentukan tindakan apa yang dapat

dilakukan oleh principal tertentu pada sumber daya tersebut dan dalam kondisi apa. Anda harus menentukan principal dalam kebijakan berbasis sumber daya. Prinsipal dapat mencakup akun, pengguna, peran, pengguna federasi, atau. Layanan AWS

Untuk mengaktifkan akses lintas akun, Anda dapat menentukan secara spesifik seluruh akun atau entitas IAM di akun lain sebagai principal dalam kebijakan berbasis sumber daya. Untuk informasi selengkapnya, lihat Akses sumber daya lintas akun di IAM dalam Panduan Pengguna IAM.

Tindakan kebijakan untuk HealthOmics

Mendukung tindakan kebijakan: Ya

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Yaitu, di mana utama dapat melakukan tindakan pada sumber daya, dan dalam kondisi apa.

Elemen `Action` dari kebijakan JSON menjelaskan tindakan yang dapat Anda gunakan untuk mengizinkan atau menolak akses dalam sebuah kebijakan. Sertakan tindakan dalam kebijakan untuk memberikan izin untuk melakukan operasi terkait.

Untuk melihat daftar HealthOmics tindakan, lihat Tindakan yang Ditentukan oleh AWS HealthOmics di Referensi Otorisasi Layanan.

Tindakan kebijakan HealthOmics menggunakan awalan berikut sebelum tindakan:

```
omics
```

Untuk menetapkan secara spesifik beberapa tindakan dalam satu pernyataan, pisahkan tindakan tersebut dengan koma.

```
"Action": [
      "omics:action1",
      "omics:action2"
         ]
```

Untuk melihat contoh kebijakan HealthOmics berbasis identitas AWS, lihat. Contoh kebijakan berbasis identitas untuk AWS HealthOmics

Sumber daya kebijakan untuk HealthOmics

Mendukung sumber daya kebijakan: Ya

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Yaitu, di mana utama dapat melakukan tindakan pada sumber daya, dan dalam kondisi apa.

Elemen kebijakan JSON `Resource` menentukan objek yang menjadi target penerapan tindakan. Praktik terbaiknya, tentukan sumber daya menggunakan Amazon Resource Name (ARN). Untuk tindakan yang tidak mendukung izin di tingkat sumber daya, gunakan wildcard (*) untuk menunjukkan bahwa pernyataan tersebut berlaku untuk semua sumber daya.

```
"Resource": "*"
```

Untuk melihat daftar jenis HealthOmics sumber daya dan jenisnya ARNs, lihat Sumber Daya yang Ditentukan oleh AWS HealthOmics di Referensi Otorisasi Layanan. Untuk mempelajari tindakan mana yang dapat Anda tentukan ARN dari setiap sumber daya, lihat Tindakan yang Ditentukan oleh AWS. HealthOmics

Untuk melihat contoh kebijakan HealthOmics berbasis identitas AWS, lihat. Contoh kebijakan berbasis identitas untuk AWS HealthOmics

Kunci kondisi kebijakan untuk HealthOmics

Kunci kondisi kebijakan tidak didukung di HealthOmics.

Daftar kontrol akses (ACLs) di HealthOmics

Mendukung ACLs: Tidak

Access control lists (ACLs) mengontrol prinsipal mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACLs mirip dengan kebijakan berbasis sumber daya, meskipun mereka tidak menggunakan format dokumen kebijakan JSON.

Kontrol akses berbasis atribut (ABAC) dengan HealthOmics

Mendukung ABAC (tanda dalam kebijakan): Ya

Kontrol akses berbasis atribut (ABAC) adalah strategi otorisasi yang menentukan izin berdasarkan atribut tanda. Anda dapat melampirkan tag ke entitas dan AWS sumber daya IAM, lalu merancang kebijakan ABAC untuk mengizinkan operasi saat tag prinsipal cocok dengan tag pada sumber daya.

Untuk mengendalikan akses berdasarkan tanda, berikan informasi tentang tanda di elemen kondisi dari kebijakan menggunakan kunci kondisi `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, atau `aws:TagKeys`.

Jika sebuah layanan mendukung ketiga kunci kondisi untuk setiap jenis sumber daya, nilainya adalah Ya untuk layanan tersebut. Jika suatu layanan mendukung ketiga kunci kondisi untuk hanya beberapa jenis sumber daya, nilainya adalah Parsial.

Untuk informasi selengkapnya tentang ABAC, lihat Tentukan izin dengan otorisasi ABAC dalam Panduan Pengguna IAM. Untuk melihat tutorial yang menguraikan langkah-langkah pengaturan ABAC, lihat Menggunakan kontrol akses berbasis atribut (ABAC) dalam Panduan Pengguna IAM.

Untuk informasi selengkapnya tentang penandaan sumber daya HealthOmics, lihat Menandai sumber daya di HealthOmics.

Contoh berikut menunjukkan bagaimana Anda dapat menulis kebijakan IAM yang menolak akses ke sumber daya tanpa tag tertentu.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "omics:*"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "Null": {
                  "aws:RequestTag/MyCustomTag": "true"
                }
            }
        }
```

```
    ]
}
```

Menggunakan kredensi sementara dengan HealthOmics

Mendukung kredensial sementara: Ya

Kredensyal sementara menyediakan akses jangka pendek ke AWS sumber daya dan secara otomatis dibuat saat Anda menggunakan federasi atau beralih peran. AWS merekomendasikan agar Anda secara dinamis menghasilkan kredensyal sementara alih-alih menggunakan kunci akses jangka panjang. Untuk informasi selengkapnya, lihat Kredensial keamanan sementara di IAM dan Layanan AWS yang berfungsi dengan IAM dalam Panduan Pengguna IAM.

Izin utama lintas layanan untuk HealthOmics

Mendukung sesi akses terusan (FAS): Ya

Sesi akses teruskan (FAS) menggunakan izin dari pemanggilan utama Layanan AWS, dikombinasikan dengan permintaan Layanan AWS untuk membuat permintaan ke layanan hilir. Untuk detail kebijakan ketika mengajukan permintaan FAS, lihat Sesi akses terusan.

Peran layanan untuk HealthOmics

Mendukung peran layanan: Ya

Peran layanan adalah peran IAM yang diambil oleh sebuah layanan untuk melakukan tindakan atas nama Anda. Administrator IAM dapat membuat, mengubah, dan menghapus peran layanan dari dalam IAM. Untuk informasi selengkapnya, lihat Buat sebuah peran untuk mendelegasikan izin ke Layanan AWS dalam Panduan pengguna IAM.

> **Warning**
>
> Mengubah izin untuk peran layanan dapat merusak HealthOmics fungsionalitas. Edit peran layanan hanya jika HealthOmics memberikan panduan untuk melakukannya.

Peran terkait layanan untuk HealthOmics

Mendukung peran terkait layanan: Tidak

Peran terkait layanan adalah jenis peran layanan yang ditautkan ke. Layanan AWS Layanan tersebut dapat menjalankan peran untuk melakukan tindakan atas nama Anda. Peran terkait layanan muncul di Anda Akun AWS dan dimiliki oleh layanan. Administrator IAM dapat melihat, tetapi tidak dapat mengedit izin untuk peran terkait layanan.

Untuk detail tentang pembuatan atau manajemen peran terkait layanan, lihat Layanan AWS yang berfungsi dengan IAM. Cari layanan dalam tabel yang memiliki `Yes` di kolom Peran terkait layanan. Pilih tautan Ya untuk melihat dokumentasi peran terkait layanan untuk layanan tersebut.

Contoh kebijakan berbasis identitas untuk AWS HealthOmics

Secara default, pengguna dan peran tidak memiliki izin untuk membuat atau memodifikasi HealthOmics sumber daya AWS. Untuk memberikan izin kepada pengguna untuk melakukan tindakan di sumber daya yang mereka perlukan, administrator IAM dapat membuat kebijakan IAM.

Untuk mempelajari cara membuat kebijakan berbasis identitas IAM dengan menggunakan contoh dokumen kebijakan JSON ini, lihat Membuat kebijakan IAM (konsol) di Panduan Pengguna IAM.

Untuk detail tentang tindakan dan jenis sumber daya yang ditentukan oleh AWS HealthOmics, termasuk format ARNs untuk setiap jenis sumber daya, lihat Tindakan, Sumber Daya, dan Kunci Kondisi untuk AWS HealthOmics di Referensi Otorisasi Layanan.

Topik

- Praktik terbaik kebijakan
- Menggunakan konsol HealthOmics
- Mengizinkan pengguna melihat izin mereka sendiri

Praktik terbaik kebijakan

Kebijakan berbasis identitas menentukan apakah seseorang dapat membuat, mengakses, atau menghapus HealthOmics sumber daya AWS di akun Anda. Tindakan ini membuat Akun AWS Anda dikenai biaya. Ketika Anda membuat atau mengedit kebijakan berbasis identitas, ikuti panduan dan rekomendasi ini:

- Mulailah dengan kebijakan AWS terkelola dan beralih ke izin hak istimewa paling sedikit — Untuk mulai memberikan izin kepada pengguna dan beban kerja Anda, gunakan kebijakan AWS terkelola yang memberikan izin untuk banyak kasus penggunaan umum. Mereka tersedia di Anda Akun AWS. Kami menyarankan Anda mengurangi izin lebih lanjut dengan menentukan kebijakan

yang dikelola AWS pelanggan yang khusus untuk kasus penggunaan Anda. Untuk informasi selengkapnya, lihat Kebijakan yang dikelola AWS atau Kebijakan yang dikelola AWS untuk fungsi tugas dalam Panduan Pengguna IAM.

- Menerapkan izin dengan hak akses paling rendah – Ketika Anda menetapkan izin dengan kebijakan IAM, hanya berikan izin yang diperlukan untuk melakukan tugas. Anda melakukannya dengan mendefinisikan tindakan yang dapat diambil pada sumber daya tertentu dalam kondisi tertentu, yang juga dikenal sebagai izin dengan hak akses paling rendah. Untuk informasi selengkapnya tentang cara menggunakan IAM untuk mengajukan izin, lihat Kebijakan dan izin dalam IAM dalam Panduan Pengguna IAM.
- Gunakan kondisi dalam kebijakan IAM untuk membatasi akses lebih lanjut – Anda dapat menambahkan suatu kondisi ke kebijakan Anda untuk membatasi akses ke tindakan dan sumber daya. Sebagai contoh, Anda dapat menulis kondisi kebijakan untuk menentukan bahwa semua permintaan harus dikirim menggunakan SSL. Anda juga dapat menggunakan ketentuan untuk memberikan akses ke tindakan layanan jika digunakan melalui yang spesifik Layanan AWS, seperti CloudFormation. Untuk informasi selengkapnya, lihat Elemen kebijakan JSON IAM: Kondisi dalam Panduan Pengguna IAM.
- Gunakan IAM Access Analyzer untuk memvalidasi kebijakan IAM Anda untuk memastikan izin yang aman dan fungsional – IAM Access Analyzer memvalidasi kebijakan baru dan yang sudah ada sehingga kebijakan tersebut mematuhi bahasa kebijakan IAM (JSON) dan praktik terbaik IAM. IAM Access Analyzer menyediakan lebih dari 100 pemeriksaan kebijakan dan rekomendasi yang dapat ditindaklanjuti untuk membantu Anda membuat kebijakan yang aman dan fungsional. Untuk informasi selengkapnya, lihat Validasi kebijakan dengan IAM Access Analyzer dalam Panduan Pengguna IAM.
- Memerlukan otentikasi multi-faktor (MFA) - Jika Anda memiliki skenario yang mengharuskan pengguna IAM atau pengguna root di Anda, Akun AWS aktifkan MFA untuk keamanan tambahan. Untuk meminta MFA ketika operasi API dipanggil, tambahkan kondisi MFA pada kebijakan Anda. Untuk informasi selengkapnya, lihat Amankan akses API dengan MFA dalam Panduan Pengguna IAM.

Untuk informasi selengkapnya tentang praktik terbaik dalam IAM, lihat Praktik terbaik keamanan di IAM dalam Panduan Pengguna IAM.

Menggunakan konsol HealthOmics

Untuk mengakses HealthOmics konsol AWS, Anda harus memiliki set izin minimum. Izin ini harus memungkinkan Anda untuk membuat daftar dan melihat detail tentang HealthOmics sumber daya

AWS di Anda Akun AWS. Jika Anda membuat kebijakan berbasis identitas yang lebih ketat daripada izin minimum yang diperlukan, konsol tidak akan berfungsi sebagaimana mestinya untuk entitas (pengguna atau peran) dengan kebijakan tersebut.

Anda tidak perlu mengizinkan izin konsol minimum untuk pengguna yang melakukan panggilan hanya ke AWS CLI atau AWS API. Sebagai gantinya, izinkan akses hanya ke tindakan yang sesuai dengan operasi API yang coba mereka lakukan.

Mengizinkan pengguna melihat izin mereka sendiri

Contoh ini menunjukkan cara membuat kebijakan yang mengizinkan pengguna IAM melihat kebijakan inline dan terkelola yang dilampirkan ke identitas pengguna mereka. Kebijakan ini mencakup izin untuk menyelesaikan tindakan ini di konsol atau menggunakan API atau secara terprogram. AWS CLI 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 kebijakan terkelola untuk AWS HealthOmics

Kebijakan AWS terkelola adalah kebijakan mandiri yang dibuat dan dikelola oleh AWS. AWS Kebijakan terkelola dirancang untuk memberikan izin bagi banyak kasus penggunaan umum sehingga Anda dapat mulai menetapkan izin kepada pengguna, grup, dan peran.

Perlu diingat bahwa kebijakan AWS terkelola mungkin tidak memberikan izin hak istimewa paling sedikit untuk kasus penggunaan spesifik Anda karena tersedia untuk digunakan semua pelanggan. AWS Kami menyarankan Anda untuk mengurangi izin lebih lanjut dengan menentukan kebijakan yang dikelola pelanggan yang khusus untuk kasus penggunaan Anda.

Anda tidak dapat mengubah izin yang ditentukan dalam kebijakan AWS terkelola. Jika AWS memperbarui izin yang ditentukan dalam kebijakan AWS terkelola, pembaruan akan memengaruhi semua identitas utama (pengguna, grup, dan peran) yang dilampirkan kebijakan tersebut. AWS kemungkinan besar akan memperbarui kebijakan AWS terkelola saat baru Layanan AWS diluncurkan atau operasi API baru tersedia untuk layanan yang ada.

Untuk informasi selengkapnya, lihat Kebijakan terkelola AWS dalam Panduan Pengguna IAM.

AWS kebijakan terkelola: AmazonOmicsFullAccess

Anda dapat melampirkan `AmazonOmicsFullAccess` kebijakan ke identitas IAM Anda untuk memberi mereka akses penuh. HealthOmics

Kebijakan ini memberikan izin akses penuh ke semua HealthOmics tindakan. Saat Anda membuat anotasi atau toko varian, Omics juga akan memberi Anda akses ke penyimpanan itu melalui Undangan Berbagi Sumber Daya di konsol Resource Access Manager (RAM). Untuk informasi selengkapnya tentang undangan Berbagi Sumber Daya melalui Lake Formation, lihat berbagi data lintas akun di Lake Formation. Untuk kebijakan admin Omics, Anda juga memerlukan izin berikut untuk mengakses bucket Amazon S3 Anda.

- 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 kebijakan terkelola: AmazonOmicsReadOnlyAccess

Anda dapat melampirkan `AWSOmicsReadOnlyAccess` kebijakan ke identitas IAM Anda ketika Anda ingin membatasi izin untuk identitas tersebut untuk akses hanya-baca.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "omics:Get*",
                "omics:List*"
            ],
            "Resource": "*"
        }
    ]
}
```

HealthOmics pembaruan kebijakan AWS terkelola

Lihat detail tentang pembaruan kebijakan AWS terkelola HealthOmics sejak layanan ini mulai melacak perubahan ini. Untuk peringatan otomatis tentang perubahan pada halaman ini, berlangganan umpan RSS di halaman Riwayat HealthOmics dokumen.

Ubah	Deskripsi	Date
AmazonOmicsFullAccess - Kebijakan baru ditambahkan	HealthOmics menambahkan kebijakan baru untuk memberikan pengguna akses penuh ke semua tindakan dan sumber daya. Untuk mempelajari selengkapnya, lihat AmazonOmicsFullAccess.	23 Februari 2023
HealthOmics mulai melacak perubahan	HealthOmics mulai melacak perubahan untuk kebijakan AWS terkelolanya.	29 November 2022
AmazonOmicsReadOnlyAccess - Kebijakan baru ditambahkan	HealthOmics menambahkan kebijakan baru yang membatasi akses hanya untuk membaca. Untuk mempelajari lebih lanjut, AmazonOmicsReadOnlyAccess.	29 November 2022

Memecahkan masalah AWS HealthOmics identitas dan akses

Gunakan informasi berikut untuk membantu Anda mendiagnosis dan memperbaiki masalah umum yang mungkin Anda temui saat bekerja dengan AWS HealthOmics dan IAM.

Topik

- Saya tidak berwenang untuk melakukan tindakan di HealthOmics
- Saya tidak berwenang untuk melakukan iam: PassRole
- Saya ingin mengizinkan orang di luar saya Akun AWS untuk mengakses HealthOmics sumber daya saya

Saya tidak berwenang untuk melakukan tindakan di HealthOmics

Jika Anda menerima pesan kesalahan bahwa Anda tidak memiliki otorisasi untuk melakukan tindakan, kebijakan Anda harus diperbarui agar Anda dapat melakukan tindakan tersebut.

Contoh kesalahan berikut terjadi ketika pengguna IAM `mateojackson` mencoba menggunakan konsol untuk melihat detail tentang suatu sumber daya *`my-example-widget`* rekaan, tetapi tidak memiliki izin `omics:`*`GetWidget`* rekaan.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 omics:GetWidget on resource: my-example-widget
```

Dalam hal ini, kebijakan untuk pengguna `mateojackson` harus diperbarui untuk mengizinkan akses ke sumber daya *`my-example-widget`* dengan menggunakan tindakan `omics:`*`GetWidget`*.

Jika Anda memerlukan bantuan, hubungi AWS administrator Anda. Administrator Anda adalah orang yang memberi Anda kredensial masuk.

Saya tidak berwenang untuk melakukan iam: PassRole

Jika Anda menerima kesalahan bahwa Anda tidak diizinkan untuk melakukan `iam:PassRole` tindakan, kebijakan Anda harus diperbarui agar Anda dapat meneruskan peran ke AWS HealthOmics.

Beberapa Layanan AWS memungkinkan Anda untuk meneruskan peran yang ada ke layanan tersebut alih-alih membuat peran layanan baru atau peran terkait layanan. Untuk melakukannya, Anda harus memiliki izin untuk meneruskan peran ke layanan.

Contoh kesalahan berikut terjadi ketika pengguna IAM bernama `marymajor` mencoba menggunakan konsol untuk melakukan tindakan di AWS HealthOmics. Namun, tindakan tersebut memerlukan layanan untuk mendapatkan izin yang diberikan oleh peran layanan. Mary tidak memiliki izin untuk meneruskan peran tersebut pada layanan.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

Dalam kasus ini, kebijakan Mary harus diperbarui agar dia mendapatkan izin untuk melakukan tindakan `iam:PassRole` tersebut.

Jika Anda memerlukan bantuan, hubungi AWS administrator Anda. Administrator Anda adalah orang yang memberi Anda kredensial masuk.

Saya ingin mengizinkan orang di luar saya Akun AWS untuk mengakses HealthOmics sumber daya saya

Anda dapat membuat peran yang dapat digunakan pengguna di akun lain atau orang-orang di luar organisasi Anda untuk mengakses sumber daya Anda. Anda dapat menentukan siapa saja yang dipercaya untuk mengambil peran tersebut. Untuk layanan yang mendukung kebijakan berbasis sumber daya atau daftar kontrol akses (ACLs), Anda dapat menggunakan kebijakan tersebut untuk memberi orang akses ke sumber daya Anda.

Untuk mempelajari selengkapnya, periksa referensi berikut:

- Untuk mengetahui apakah AWS HealthOmics mendukung fitur-fitur ini, lihatBagaimana AWS HealthOmics bekerja dengan IAM.
- Untuk mempelajari cara menyediakan akses ke sumber daya Anda di seluruh sumber daya Akun AWS yang Anda miliki, lihat Menyediakan akses ke pengguna IAM di pengguna lain Akun AWS yang Anda miliki di Panduan Pengguna IAM.
- Untuk mempelajari cara menyediakan akses ke sumber daya Anda kepada pihak ketiga Akun AWS, lihat Menyediakan akses yang Akun AWS dimiliki oleh pihak ketiga dalam Panduan Pengguna IAM.
- Untuk mempelajari cara memberikan akses melalui federasi identitas, lihat Menyediakan akses ke pengguna terautentikasi eksternal (federasi identitas) dalam Panduan Pengguna IAM.
- Untuk mempelajari perbedaan antara menggunakan peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat Akses sumber daya lintas akun di IAM di Panduan Pengguna IAM.

Validasi kepatuhan untuk AWS HealthOmics

Auditor pihak ketiga menilai keamanan dan kepatuhan AWS HealthOmics sebagai bagian dari beberapa program AWS kepatuhan. Ini termasuk HIPAA, FedRAMP, dan lainnya. Tabel berikut menunjukkan sertifikasi kepatuhan untuk HealthOmics layanan.

Sertifikasi	Tautan
HIPAA	Referensi Layanan yang Memenuhi Syarat HIPAA

Sertifikasi	Tautan
HiTrust-CSF	Kerangka Keamanan Umum Aliansi Kepercayaan Informasi Kesehatan
FedRAMP Moderat (Timur/Barat)	Program Manajemen Risiko dan Otorisasi Federal
BINTANG ISO/CSA	Bersertifikat ISO dan CSA STAR
C5	Katalog Kontrol Kepatuhan Cloud Computing
DoD CC SRG IL2	Panduan Persyaratan Keamanan Komputasi Awan Departemen Pertahanan
ENS High	Esquema Nacional de Seguridad
FINMA	Otoritas Pengawas Pasar Keuangan Swiss
ISMAP	Program Pengelolaan dan Penilaian Keamanan Sistem Informasi
OSPAR	Laporan Audit Penyedia Layanan Outsourcing
PCI	Standar Keamanan Data Industri Kartu Pembayaran
Pinakes	Asosiasi perbankan CCI - Kualifikasi Pihak Ketiga
PiTuKri	Kriteria untuk Menilai Keamanan Informasi Layanan Cloud
SOC 1,2,3	Kontrol Sistem dan Organisasi

Untuk daftar semua AWS layanan dalam cakupan program kepatuhan tertentu, lihat AWS Services in Scope by Compliance Program . Untuk informasi umum, lihat Program Kepatuhan AWS .

Anda dapat mengunduh laporan audit pihak ketiga menggunakan AWS Artifact. Untuk informasi selengkapnya, lihat Mengunduh Laporan di AWS Artifact .

HealthOmics penyimpanan data menggunakan ID sampel untuk penamaan file internal dan untuk menandai sumber daya. Sebelum Anda menelan data, periksa apakah ID sampel berisi data PHI. Jika ya, ubah ID sampel sebelum Anda menelan data. Untuk informasi selengkapnya, lihat panduan tentang halaman web kepatuhan AWS HIPAA.

Tanggung jawab kepatuhan Anda saat menggunakan AWS HealthOmics ditentukan oleh sensitivitas data Anda, tujuan kepatuhan perusahaan Anda, dan hukum dan peraturan yang berlaku. AWS menyediakan sumber daya berikut untuk membantu kepatuhan:

- Panduan Quick Start Keamanan dan Kepatuhan – Panduan deployment ini membahas pertimbangan arsitektur dan menyediakan langkah–langkah untuk melakukan deployment terhadap lingkungan dasar di AWS yang menjadi fokus keamanan dan kepatuhan.
- Arsitektur untuk Whitepaper Keamanan dan Kepatuhan HIPAA — Whitepaper ini menjelaskan bagaimana perusahaan dapat menggunakan untuk membuat aplikasi yang sesuai dengan HIPAA. AWS
- AWS Sumber Daya AWS — Kumpulan buku kerja dan panduan ini mungkin berlaku untuk industri dan lokasi Anda.
- Mengevaluasi Sumber Daya dengan Aturan dalam Panduan AWS Config Pengembang — AWS Config; menilai seberapa baik konfigurasi sumber daya Anda mematuhi praktik internal, pedoman industri, dan peraturan.
- AWS Security Hub CSPM— AWS Layanan ini memberikan pandangan komprehensif tentang keadaan keamanan Anda di dalamnya AWS yang membantu Anda memeriksa kepatuhan Anda terhadap standar industri keamanan dan praktik terbaik.

Ketahanan di HealthOmics

Infrastruktur AWS global dibangun di sekitar Wilayah AWS dan Availability Zones. Wilayah AWS menyediakan beberapa Availability Zone yang terpisah secara fisik dan terisolasi, yang terhubung dengan latensi rendah, throughput tinggi, dan jaringan yang sangat redundan. Dengan Zona Ketersediaan, Anda dapat merancang serta mengoperasikan aplikasi dan basis data yang secara otomatis melakukan fail over di antara zona tanpa gangguan. Zona Ketersediaan memiliki ketersediaan dan toleransi kesalahan yang lebih baik, dan dapat diskalakan dibandingkan infrastruktur pusat data tunggal atau multi tradisional.

Untuk informasi selengkapnya tentang Wilayah AWS dan Availability Zone, lihat Infrastruktur AWS Global.

Selain infrastruktur AWS global, AWS HealthOmics menawarkan beberapa fitur untuk membantu mendukung ketahanan data dan kebutuhan pencadangan Anda.

AWS HealthOmics dan antarmuka titik akhir VPC ()AWS PrivateLink

Anda dapat membuat koneksi pribadi antara VPC Anda dan AWS HealthOmics dengan membuat antarmuka VPC endpoint. Endpoint antarmuka didukung oleh AWS PrivateLink, teknologi yang dapat Anda gunakan untuk mengakses operasi HealthOmics API secara pribadi tanpa gateway internet, perangkat NAT, koneksi VPN, atau koneksi AWS Direct Connect. Instans di VPC Anda tidak memerlukan alamat IP publik untuk berkomunikasi HealthOmics dengan operasi API. Lalu lintas antara VPC Anda dan HealthOmics tidak keluar dari jaringan Amazon.

Setiap titik akhir antarmuka diwakili oleh satu atau beberapa Antarmuka Jaringan Elastis di subnet Anda.

Untuk informasi selengkapnya, lihat Titik akhir VPC Antarmuka (AWS PrivateLink) di Panduan Pengguna Amazon VPC.

Kebijakan titik akhir VPC didukung HealthOmics untuk semua Wilayah kecuali Israel (Tel Aviv). Secara default, akses penuh ke HealthOmics diizinkan melalui titik akhir.

Pertimbangan untuk titik akhir HealthOmics VPC

Sebelum menyiapkan titik akhir VPC antarmuka HealthOmics, pastikan Anda meninjau properti dan batasan titik akhir Antarmuka di Panduan Pengguna Amazon VPC.

HealthOmics mendukung panggilan ke semua tindakan HealthOmics Storage API dari VPC Anda.

Kebijakan titik akhir VPC tidak didukung secara HealthOmics default, tetapi Anda dapat membuat titik akhir VPC untuk akses penuh HealthOmics untuk operasi Penyimpanan. HealthOmics Untuk informasi selengkapnya, lihat Mengontrol Akses ke Layanan dengan titik akhir VPC dalam Panduan Pengguna Amazon VPC.

Buat VPC endpoint antarmuka untuk HealthOmics

Anda dapat membuat titik akhir VPC untuk HealthOmics layanan menggunakan konsol VPC Amazon atau (). AWS Command Line Interface AWS CLI Untuk informasi selengkapnya, lihat Membuat titik akhir antarmuka dalam Panduan Pengguna Amazon VPC.

Buat titik akhir VPC HealthOmics dengan menggunakan nama layanan berikut:

- com.amazonaws. *region*.penyimpanan-omics
- com.amazonaws. *region*. control-storage-omics
- com.amazonaws. *region*.analytics-omics
- com.amazonaws. *region*.workflows-omics
- com.amazonaws. *region*.tags-omics

Wilayah AS Timur (Virginia N.) dan AS Barat (Oregon) mendukung titik akhir AWS PrivateLink FIPS. Untuk wilayah ini, Anda juga dapat menggunakan nama layanan berikut:

- 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

Jika Anda mengaktifkan DNS pribadi untuk titik akhir, Anda dapat membuat permintaan API HealthOmics dengan menggunakan nama DNS default untuk Wilayah, misalnya,. `omics.us-east-1.amazonaws.com`

Untuk informasi lebih lanjut, lihat Mengakses layanan melalui titik akhir antarmuka dalam Panduan Pengguna Amazon VPC.

Membuat kebijakan VPC endpoint untuk HealthOmics

Anda dapat melampirkan kebijakan titik akhir ke VPC endpoint yang mengendalikan akses ke HealthOmics. Kebijakan menentukan informasi berikut ini:

- Prinsip-prinsip yang dapat melakukan tindakan.
- Tindakan yang dapat dilakukan
- Sumber daya yang dapat digunakan untuk mengambil tindakan

Untuk informasi selengkapnya, lihat Mengendalikan akses ke layanan dengan VPC endpoint di Panduan Pengguna Amazon VPC.

Contoh: Kebijakan titik akhir VPC untuk tindakan. HealthOmics

Berikut ini adalah contoh kebijakan endpoint untuk HealthOmics. Saat dilampirkan ke titik akhir, kebijakan ini memberikan akses ke HealthOmics tindakan untuk semua prinsipal di semua sumber daya.

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\":\"*\"}]}"
```

Pertimbangan khusus untuk mengakses set baca menggunakan Amazon S3 URIs

Untuk mengakses set baca melalui Amazon S3 URIs saat Anda menggunakan koneksi pribadi, siapkan titik akhir PrivateLink antarmuka di penyimpanan urutan. Setelah Anda mengaturnya, titik akhir memiliki format berikut:

```
com.amazonaws.region.storage-omics
com.amazonaws.region.control-storage-omics
```

Untuk menggunakan titik akhir Gateway, ikuti panduan Titik akhir Gateway untuk Amazon S3 untuk mengonfigurasi titik akhir gateway Anda. HealthOmics memiliki bucket Amazon S3, jadi Anda tidak perlu membuat atau menyesuaikan kebijakan bucket. Titik akhir Gateway bergantung pada kebijakan yang dilampirkan pada pengguna atau peran yang mengakses data, tetapi Anda juga dapat mengonfigurasi titik akhir dengan kebijakan yang lebih ketat. Kebijakan ini dapat mencakup pembatasan akses berdasarkan tindakan Amazon S3 Access Point ARN dan Amazon S3.

Pemantauan AWS HealthOmics

Pemantauan adalah bagian penting dalam menjaga keandalan, ketersediaan, dan kinerja AWS HealthOmics dan AWS solusi Anda yang lain. AWS menyediakan alat pemantauan berikut untuk menonton AWS HealthOmics, melaporkan ketika ada sesuatu yang salah, dan mengambil tindakan otomatis bila perlu:

- Amazon CloudWatch memantau AWS sumber daya Anda dan aplikasi yang Anda jalankan AWS secara real time. Anda dapat mengumpulkan dan melacak metrik, membuat dasbor yang disesuaikan, dan mengatur alarm yang memberi tahu Anda atau mengambil tindakan saat metrik tertentu mencapai ambang batas yang ditentukan. Misalnya, Anda dapat CloudWatch melacak penggunaan CPU atau metrik lain dari EC2 instans Amazon Anda dan secara otomatis meluncurkan instans baru bila diperlukan. Untuk informasi selengkapnya, lihat Panduan CloudWatch Pengguna Amazon.
- Amazon CloudWatch Logs memungkinkan Anda memantau, menyimpan, dan mengakses file log Anda dari EC2 instans Amazon CloudTrail, dan sumber lainnya. CloudWatch Log dapat memantau informasi dalam file log dan memberi tahu Anda ketika ambang batas tertentu terpenuhi. Anda juga dapat mengarsipkan data log dalam penyimpanan yang sangat durabel. Untuk informasi selengkapnya, lihat Panduan Pengguna Amazon CloudWatch Logs.
- AWS CloudTrail merekam panggilan API dan kejadian terkait yang dilakukan oleh atau atas Akun AWS Anda dan mengirimkan berkas log ke bucket Amazon S3 yang Anda tentukan. Anda dapat mengidentifikasi pengguna dan akun yang memanggil AWS, alamat IP asal panggilan dilakukan, dan waktu panggilan terjadi. Untuk informasi selengkapnya, lihat Panduan Pengguna AWS CloudTrail.
- Amazon EventBridge adalah layanan bus acara tanpa server yang memudahkan untuk menghubungkan aplikasi Anda dengan data dari berbagai sumber. EventBridge mengirimkan aliran data real-time dari aplikasi Anda sendiri, aplikasi Software-as-a-Service (SaaS), AWS dan layanan dan rute data tersebut ke target seperti Lambda. Hal ini memungkinkan Anda memantau kejadian yang terjadi dalam layanan, dan membangun arsitektur yang didorong kejadian. Untuk informasi selengkapnya, lihat Panduan EventBridge Pengguna Amazon.

> **Note**
>
> Untuk pembaruan layanan, konfigurasikan dan pantau Dashboard Personal Health Anda. Untuk informasi selengkapnya tentang cara mengelola dasbor, lihat Memulai dengan Dasbor AWS Health Anda.

Topik

Pencatatan akses S3

Anda dapat memantau akses API Amazon S3 ke data penyimpanan HealthOmics urutan menggunakan log akses yang dibuat di toko. Anda dapat menggunakan CloudWatch untuk memantau akses S3 dari operasi HealthOmics API. CloudWatch memberikan visibilitas ke akses Amazon S3 yang berasal dari akun Anda sendiri. Jika Anda, sebagai pemilik data, berbagi akses ke akun pihak ketiga, akses masuk tidak tersedia CloudWatch. Sebagai gantinya, gunakan S3 Access Log. yang mencatat semua akses S3 ke data di bucket Amazon S3 yang dikonfigurasi.

Konfigurasikan Log Akses S3 menggunakan operasi `CreateSequenceStore` atau `UpdateSequenceStore` API. Juga, pastikan bahwa HealthOmics service principal (`omics.amazonaws.com`) memiliki `s3:PutObject` izin untuk awalan S3 yang dikonfigurasi.

> **Note**
>
> Log menggunakan konfigurasi enkripsi default bucket tujuan. Jika bucket menggunakan kunci yang dikelola pelanggan, prinsipal layanan harus memiliki akses untuk menggunakan kunci tersebut untuk menulis.

Untuk menonaktifkan pencatatan akses, gunakan `UpdateSequenceStore` dan atur konfigurasi log akses ke kosong.

Pemantauan HealthOmics dengan CloudWatch metrik

Anda dapat memantau HealthOmics penggunaan CloudWatch, yang mengumpulkan data mentah dan memprosesnya menjadi metrik yang dapat dibaca, mendekati waktu nyata. Statistik ini disimpan untuk jangka waktu 15 bulan, sehingga Anda dapat mengakses informasi historis dan mendapatkan perspektif yang lebih baik tentang performa aplikasi atau layanan web Anda. Anda juga dapat mengatur alarm yang memperhatikan ambang batas tertentu dan mengirim notifikasi atau mengambil tindakan saat ambang batas tersebut terpenuhi. Untuk informasi selengkapnya, lihat Panduan CloudWatch Pengguna Amazon.

AWS HealthOmics Layanan melaporkan metrik berikut di `AWS/Omics` namespace.

Metrik Hitungan Panggilan API dilaporkan sebagai berikut AWS HealthOmics APIs. Hanya dimensi Operasi API yang dilaporkan.

- Toko referensi dan referensi APIs —CreateReferenceStore, DeleteReferenceStore, StartReferenceImportJob
- Menyimpan urutan dan set baca APIs —CreateSequenceStore, DeleteSequenceStore, StartReadSetImportJob, StartReadSetActivationJob, StartReadSetExportJob
- Toko varian APIs — CreateVariantStore, DeleteVariantStore, StartVariantImportJob, CancelVariantImportJob
- Toko anotasi APIs — CreateAnnotationStore,, DeleteAnotationStore, StartAnnotationImportJob CancelAnnotationImportJob
- Alur kerja, jalankan, dan jalankan grup APIs — CreateWorkflow, DeleteWorkflow, StartRun, CancelRun, DeleteRun, CreateRunGroup DeleteRunGroup

Melihat metrik *AWS HealthOmics*

CloudWatch metrik untuk AWS HealthOmics dapat dilihat di konsol. CloudWatch

Untuk melihat metrik (CloudWatch konsol)

1. Masuk ke AWS Management Console dan buka CloudWatch konsol.
2. Pilih Metrik, pilih Semua Metrik, lalu pilih AWS/Penggunaan.
3. Layanan Filter untuk AWS HealthOmics.
4. Pilih dimensi, pilih nama metrik, lalu pilih Tambahkan ke grafik.

5. Pilih nilai untuk rentang tanggal. Hitungan metrik untuk rentang tanggal yang dipilih akan ditampilkan dalam grafik.

Membuat alarm menggunakan CloudWatch

CloudWatch Alarm mengawasi satu metrik selama periode waktu tertentu, dan melakukan satu atau beberapa tindakan: mengirim pemberitahuan ke topik Simple Notification Service Amazon (Amazon SNS) atau kebijakan Auto Scaling. Tindakan atau tindakan didasarkan pada nilai metrik relatif terhadap ambang batas tertentu selama sejumlah periode waktu yang Anda tentukan. CloudWatch juga dapat mengirimi Anda pesan Amazon SNS saat alarm berubah status.

CloudWatch alarm memanggil tindakan hanya ketika status berubah dan telah bertahan selama periode yang Anda tentukan.

Untuk melihat metrik (CloudWatch konsol)

1. Masuk ke AWS Management Console dan buka CloudWatch konsol.
2. Pilih Alarm, lalu pilih Buat Alarm.
3. Pilih AWS/Usage, lalu pilih AWS HealthOmics metrik menggunakan dimensi Service.
4. Untuk Rentang Waktu, pilih rentang waktu untuk dipantau, lalu pilih Selanjutnya.
5. Masukkan Nama dan Deskripsi.
6. Untuk Kapan pun, pilih >=, dan ketik nilai maksimum.
7. Jika Anda CloudWatch ingin mengirim email saat status alarm tercapai, di bagian Tindakan, untuk Setiap kali alarm ini, pilih Status adalah ALARM. Untuk Kirim pemberitahuan ke, pilih milis atau pilih Daftar baru dan buat milis baru.
8. Pratinjau alarm di bagian Pratinjau Alarm. Jika Anda puas dengan alarm, pilih Buat Alarm.

Pemantauan HealthOmics dengan CloudWatch Log

HealthOmics menghasilkan berbagai log untuk membantu Anda memahami dan memecahkan masalah proses Anda. Log tersedia di dua tempat: CloudWatch dan Amazon S3.

Secara default, proses telah mengaktifkan logging. Anda secara opsional dapat menonaktifkan logging untuk menjalankan dengan menyetel `LogLevel = OFF` startrun permintaan.

> **Note**
>
> Untuk pembaruan layanan, konfigurasikan dan pantau Dashboard Personal Health Anda. Untuk informasi selengkapnya tentang cara mengelola dasbor, lihat Memulai dengan Dasbor AWS Health Anda.

Topik

Jenis log untuk HealthOmics alur kerja

HealthOmics menyediakan jenis log berikut untuk alur kerja:

- Log mesin — Mesin alur kerja yang mendasarinya (Nextflow, WDL, dan CWL) menghasilkan log mesin untuk dijalankan. Log ini dapat membantu Anda memecahkan masalah definisi alur kerja.
- Jalankan log manifes — Log ini memberikan informasi tingkat tinggi tentang setiap tugas yang dijalankan, seperti status tugas, waktu mulai, waktu berhenti, dan alasan gagal (jika tugas gagal).

  Jalankan log manifes juga melaporkan statistik pemanfaatan sumber daya yang dapat membantu untuk memahami peluang pengoptimalan sumber daya. Statistik ini meliputi:

  - CPUSAverage
  - CPUSMaksimum
  - CPUSreserved
  - GPUSreserved
  - memoryAverageGiB
  - memoryMaximumGiB
  - memoryReservedGiB
  - RunningSeconds

- Run logs — Run logs menyediakan status run keseluruhan dan waktu ketika tugas individu dimulai, berjalan, berhenti, dan selesai. Jalankan log juga memberi Anda visibilitas ke langkah-langkah impor dan ekspor file.
- Log tugas — Log tugas memberikan informasi pencatatan terperinci tentang tugas individual dalam proses Anda. Output dalam log tugas Anda bergantung pada definisi tugas dan di mana Anda menggunakan pernyataan log dalam kode Anda. Jika log tugas Anda tidak memberikan tingkat wawasan yang Anda butuhkan, pertimbangkan untuk menambahkan pernyataan log tambahan ke definisi tugas Anda untuk menghasilkan log tugas yang lebih berwawasan.
- Jalankan log cache - Jalankan log cache memberikan status keseluruhan cache run dan caching output tugas. Jalankan log cache memberi Anda visibilitas ke dalam klik dan kesalahan cache untuk setiap proses yang menggunakan caching.
- Outputs.json — Untuk alur kerja WDL dan CWL, HealthOmics mengirimkan file yang dibuat mesin, bernama, ke bucket Amazon S3 Anda setelah proses selesai. `outputs.json` File ini mencakup daftar dan peta semua output untuk dijalankan.

Log masuk CloudWatch

CloudWatch menghasilkan log alur kerja untuk menjalankan yang gagal dan berjalan dengan sukses. Semua log tersedia untuk proses yang gagal dan berjalan dengan sukses, kecuali log mesin hanya tersedia untuk proses yang gagal.

Anda dapat menemukan log CloudWatch alur kerja di grup log berikut:`/aws/omics/WorkflowLog`. Selain itu, output dari operasi get-run API menyediakan aliran CloudWatch log ARNs untuk log mesin dan menjalankan log.

Secara default, AWS menyimpan CloudWatch Log tanpa batas. Anda dapat menyesuaikan kebijakan penyimpanan grup log untuk menetapkan periode retensi antara 10 tahun dan satu hari.

Tabel berikut memberikan ringkasan dari CloudWatch Log in HealthOmics. Semua log alur kerja tersedia untuk proses yang berhasil dan gagal berjalan, kecuali log mesin hanya tersedia untuk proses yang gagal.

Nama log	Tersedia dalam CloudWatch Log	Kapan log tersedia	Format aliran log
Log mesin	Ya, untuk proses yang gagal	Setelah lari selesai	jalankan//mesin *runID*

Nama log	Tersedia dalam CloudWatch Log	Kapan log tersedia	Format aliran log
Jalankan log manifes	Ya	Setelah lari selesai	manifest/ lari//*runIDrunUUID*
Jalankan log	Ya	Secara real time	lari/ *runID*
Log tugas	Ya	Secara real time	jalankan// tugas/*runIDtaskID*
Jalankan log cache	Ya	Secara real time	Runcache/ /*runCacheI d runCacheUUID*
Outputs.json (WDL dan CWL)	Tidak	tidak berlaku	T/A

Log di Amazon S3

Hanya log mesin dan `outputs.json` file yang dikirim ke Amazon S3.

Setelah proses selesai, log mesin dikirim ke bucket S3 Anda dan tersedia tanpa batas hingga Anda menghapusnya. Log ini terletak di direktori log dari URI keluaran S3 yang Anda tentukan untuk alur kerja.

Jalur ke direktori log memiliki format berikut:`s3://{user_provided_path}/logs/`.

Tabel berikut memberikan ringkasan HealthOmics log yang tersedia di bucket Amazon S3 Anda.

Nama log	Tersedia di Amazon S3	Kapan log tersedia	Jalur aliran log
Log mesin	Ya	Setelah lari selesai	s3:///logs/engine. log *user_prov ided_path*
Outputs.json (WDL dan CWL)	Ya	Setelah lari selesai	s3:///*user_prov ided_path*

Nama log	Tersedia di Amazon S3	Kapan log tersedia	Jalur aliran log
			/*runID*/logs/outputs.json *runUUID*
Jalankan log manifes, jalankan log, dan log tugas	Tidak	tidak berlaku	T/A

CloudWatch Log Interaktif di CLI

Anda dapat melihat CloudWatch Log secara interaktif menggunakan perintah Live Tail dalam mode interaktif. Anda dapat melacak kemajuan run secara real time dan menentukan hingga 5 kata kunci untuk disorot di log:

```
aws logs start-live-tail  \
  --mode interactive  \
  --log-group-identifiers arn:aws:logs:region:account-ID:log-group:/aws/omics/WorkflowLog
```

Untuk informasi selengkapnya, lihat Memulai ekor langsung di Referensi AWS CLI Perintah.

Mengakses CloudWatch Log dari konsol

Untuk mengakses log untuk dijalankan, Anda dapat menautkan langsung ke log ini dari halaman Run details di HealthOmics konsol.

1. Buka konsol HealthOmics .
2. Jika diperlukan, buka panel navigasi kiri (≡). Pilih Runs.
3. Pilih run dari tabel Runs.
4. Di halaman run details, Anda dapat memilih salah satu tindakan ini:
    a. Dari Run summary, pilih View run logs. Konsol membuka log run di CloudWatch konsol.
    b. Dari Run summary, pilih Lihat log di Amazon S3. Konsol membuka folder log di konsol Amazon S3.
    c. Dari Jalankan tugas, pilih Lihat log, Lihat log jalankan, atau Lihat menjalankan log manifes untuk tugas. Konsol membuka log di CloudWatch konsol.

Anda juga dapat menavigasi ke log dari CloudWatch konsol:

1. Buka CloudWatch konsol https://console.aws.amazon.com/cloudwatch/.
2. Dari menu sebelah kiri, pilih Grup log.
3. Pilih grup `/aws/omics/WorkflowLog`.

   Jika daftar grup log panjang, Anda dapat memasukkan omics di kotak teks pencarian untuk mempersempit daftar.
4. Saat halaman detail grup Log terbuka, pilih aliran log yang ingin Anda lihat. Konsol menampilkan peristiwa untuk aliran log ini.

Logging panggilan AWS HealthOmics API menggunakan AWS CloudTrail

AWS HealthOmics terintegrasi dengan AWS CloudTrail, layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau AWS layanan di HealthOmics. CloudTrail menangkap semua panggilan API untuk HealthOmics sebagai peristiwa. Panggilan yang diambil termasuk panggilan dari HealthOmics konsol dan panggilan kode ke operasi HealthOmics API. Jika Anda membuat jejak, Anda dapat mengaktifkan pengiriman CloudTrail acara secara berkelanjutan ke bucket Amazon S3, termasuk acara untuk. HealthOmics Jika Anda tidak mengonfigurasi jejak, Anda masih dapat melihat peristiwa terbaru di CloudTrail konsol dalam Riwayat acara. Dengan menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan yang dibuat HealthOmics, alamat IP dari mana permintaan dibuat, siapa yang membuat permintaan, kapan dibuat, dan detail tambahan.

Untuk mempelajari selengkapnya CloudTrail, lihat Panduan AWS CloudTrail Pengguna.

HealthOmics informasi di CloudTrail

CloudTrail diaktifkan pada Akun AWS saat Anda membuat akun. Ketika aktivitas terjadi di HealthOmics, aktivitas tersebut dicatat dalam suatu CloudTrail peristiwa bersama dengan peristiwa AWS layanan lainnya dalam riwayat Acara. Anda dapat melihat, mencari, dan mengunduh acara terbaru di situs Anda Akun AWS. Untuk informasi selengkapnya, lihat Melihat peristiwa dengan Riwayat CloudTrail acara.

Untuk catatan acara yang sedang berlangsung di Anda Akun AWS, termasuk acara untuk HealthOmics, buat jejak. Jejak memungkinkan CloudTrail untuk mengirimkan file log ke bucket

Amazon S3. Secara default, saat Anda membuat jejak di konsol, jejak tersebut berlaku untuk semua Wilayah AWS. Jejak mencatat peristiwa dari semua Wilayah di AWS partisi dan mengirimkan file log ke bucket Amazon S3 yang Anda tentukan. Selain itu, Anda dapat mengonfigurasi AWS layanan lain untuk menganalisis lebih lanjut dan menindaklanjuti data peristiwa yang dikumpulkan dalam CloudTrail log. Untuk informasi selengkapnya, lihat berikut:

- Gambaran umum untuk membuat jejak
- CloudTrail layanan dan integrasi yang didukung
- Mengonfigurasi notifikasi Amazon SNS untuk CloudTrail
- Menerima file CloudTrail log dari beberapa wilayah dan Menerima file CloudTrail log dari beberapa akun

Semua HealthOmics tindakan dicatat oleh CloudTrail dan didokumentasikan dalam Referensi AWS HealthOmics API. Misalnya, panggilan ke`CreateReferenceeStore`, `StartVariantImportJob` dan `CreateWorkflow` tindakan menghasilkan entri dalam file CloudTrail log.

Setiap entri peristiwa atau log berisi informasi tentang entitas yang membuat permintaan tersebut. Informasi identitas membantu Anda menentukan hal berikut ini:

- Apakah permintaan dibuat dengan kredenal pengguna IAM.
- Apakah permintaan tersebut dibuat dengan kredensial keamanan sementara untuk satu peran atau pengguna gabungan.
- Apakah permintaan itu dibuat oleh AWS layanan lain.

Untuk informasi selengkapnya, lihat Elemen userIdentity CloudTrail .

Memahami entri file HealthOmics log

Trail adalah konfigurasi yang memungkinkan pengiriman peristiwa sebagai file log ke bucket Amazon S3 yang Anda tentukan. CloudTrail file log berisi satu atau lebih entri log. Peristiwa mewakili permintaan tunggal dari sumber manapun dan mencakup informasi tentang tindakan yang diminta, tanggal dan waktu tindakan, parameter permintaan, dan sebagainya. CloudTrail file log bukanlah jejak tumpukan yang diurutkan dari panggilan API publik, jadi file tersebut tidak muncul dalam urutan tertentu.

Contoh berikut menunjukkan entri CloudTrail log yang menunjukkan CreateWorkflow tindakan.

```
{
    "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"
}
```


Menggunakan EventBridge dengan AWS HealthOmics

HealthOmics mengirimkan peristiwa ke Amazon EventBridge saat sumber daya mengubah status. Sumber daya termasuk pekerjaan impor, pekerjaan ekspor, pembagian sumber daya, alur kerja, tugas, dan proses. Untuk setiap jenis sumber daya, ada daftar perubahan status yang menghasilkan suatu peristiwa.

Bus acara adalah router yang menerima acara dan mengirimkannya ke tujuan. Akun Anda menyertakan bus acara default yang secara otomatis menerima acara dari AWS layanan. Anda dapat membuat bus acara khusus tambahan.

Anda membuat EventBridge aturan untuk menentukan tindakan yang harus diambil ketika bus acara menerima acara. Misalnya, Anda dapat membuat aturan yang memberi tahu Anda tentang perubahan status untuk sumber daya.

Skenario umum untuk menggunakan acara meliputi:

- Untuk memantau kapan pengguna berbagi sumber daya dengan Anda atau mencabut pembagian.
- Untuk memantau apakah proses gagal atau berhasil diselesaikan.

Untuk informasi selengkapnya tentang penggunaan EventBridge, lihat Apa itu Amazon EventBridge?

Topik

Siapkan EventBridge untuk HealthOmics

Sebelum Anda dapat memantau EventBridge acara, buat EventBridge bus dan buat aturan untuk acara yang menarik.

Konfigurasikan EventBridge bus

Anda dapat menggunakan bus acara default untuk Anda Akun AWS atau mengkonfigurasi bus acara khusus. Untuk mengonfigurasi bus acara khusus, ikuti langkah-langkah berikut:

1. Buka EventBridge konsol: https://console.aws.amazon.com/events/.
2. Di navigasi kiri, pilih Bus acara.
3. Pilih Buat bus peristiwa.
4. Dalam formulir Buat bus acara, masukkan nama untuk bus.
5. Pilih Buat untuk membuat bus.

Buat EventBridge aturan

Prosedur berikut menunjukkan cara membuat aturan sederhana. Untuk informasi selengkapnya tentang aturan, lihat Aturan di EventBridge.

1. Buka EventBridge konsol: https://console.aws.amazon.com/events/.
2. Di navigasi kiri, pilih Aturan.
3. Pilih Buat aturan. Konsol membuka formulir Create rule.
4. Di Tentukan detail aturan, berikan nama untuk aturan tersebut.
   - Untuk Nama, masukkan nama untuk bus.
   - Untuk bus acara, pilih bus untuk aturan ini.
   - Pilih Berikutnya.
5. Dalam pola acara Build, di bawah Sumber acara pilih acara AWS atau acara EventBridge mitra.
6. Gulir ke bawah ke pola Acara.
   a. Untuk sumber Acara, pilih layanan AWS.
   b. Untuk layanan AWS, masukkan omics di filter teks dan pilih AWS HealthOmicssebagai layanan.
   c. Untuk jenis Acara pilih acara yang diminati (atau Semua acara).

d. Pilih Berikutnya.

7. Di Pilih target, pilih target untuk acara tersebut. Misalnya, pilih layanan AWS, grup CloudWatch log yang dipilih, dan konfigurasikan grup log.

   Untuk banyak jenis target, EventBridge perlu izin untuk mengirim acara ke target. Konsol membuat izin ini untuk Anda.

8. (Opsional) Di Konfigurasikan tag, kaitkan tag dengan aturan.
9. Di Tinjau dan perbarui, tinjau konfigurasi dan pilih Buat aturan.

EventBridge peristiwa di HealthOmics

Tabel berikut mencantumkan peristiwa yang HealthOmics dikirim ke EventBridge, dan daftar nilai status yang mungkin untuk acara tersebut.

Nama peristiwa	Nilai status yang mungkin
Anotasi Impor Perubahan Status Job	Dikirim, sedang berlangsung, dibatalkan, diselesaikan, gagal, atau diselesaikan dengan kegagalan
Perubahan Status Bagikan Toko Anotasi	Menunggu, mengaktifkan, aktif, menghapus, dihapus, gagal
Perubahan Status Toko Anotasi	Membuat, membuat, memperbarui, memperbarui, menghapus, menghapus, atau membuat gagal
Baca Mengatur Perubahan Status Pekerjaan Aktivasi	Dikirim, dalam proses, selesai, gagal, atau diselesaikan dengan kegagalan
Baca Mengatur Perubahan Status Pekerjaan Ekspor	Dikirim, dalam proses, selesai, gagal, atau diselesaikan dengan kegagalan
Baca Mengatur Impor Perubahan Status Job	Dikirim, dalam proses, selesai, gagal, atau diselesaikan dengan kegagalan

Nama peristiwa	Nilai status yang mungkin
Baca Mengatur Perubahan Status	Pemrosesan unggahan, pengunggahan gagal, aktif, diarsipkan, mengaktifkan, atau dihapus
Referensi Impor Perubahan Status Job	Dikirim, dalam proses, selesai, gagal, atau diselesaikan dengan kegagalan
Perubahan Status Referensi	Aktif atau dihapus
Perubahan Status Toko Referensi	Dibuat, diperbarui, aktif, atau dihapus
Jalankan Perubahan Status	Menunggu, memulai, menjalankan, menghentikan, menyelesaikan, dihapus, gagal, atau dibatalkan
Perubahan Status Toko Urutan	Dibuat, diperbarui, aktif, atau dihapus
Perubahan Status Tugas	Menunggu, memulai, menjalankan, menghentikan, menyelesaikan, dihapus, gagal, atau dibatalkan
Perubahan Status Pekerjaan Impor Varian	Dikirim, sedang berlangsung, dibatalkan, diselesaikan, gagal, atau diselesaikan dengan kegagalan
Perubahan Status Berbagi Toko Varian	Menunggu, mengaktifkan, aktif, menghapus, dihapus, gagal
Perubahan Status Toko Varian	Membuat, membuat, memperbarui, memperbarui, menghapus, menghapus, atau membuat gagal
Perubahan Status Berbagi Alur Kerja	Menunggu, mengaktifkan, aktif, menghapus, dihapus, gagal
Perubahan Status Alur Kerja	Keberhasilan penciptaan, kegagalan penciptaan, keberhasilan penghapusan, atau kegagalan penghapusan

Struktur pesan peristiwa

HealthOmics menyediakan pengiriman upaya terbaik untuk mengirim pesan peristiwa perubahan status ke EventBridge. Acara ini adalah objek dengan struktur JSON yang juga berisi detail metadata. Anda dapat menggunakan metadata sebagai masukan untuk membuat ulang acara atau untuk mempelajari informasi lebih lanjut. Acara meliputi bidang-bidang berikut:

- `version`— Saat ini 0 (nol) untuk semua acara.
- `id`— Versi 4 UUID dihasilkan untuk setiap acara.
- `detail-type`— Jenis acara yang sedang dikirim.
- `account`— 12 digit Akun AWS ID pemilik bucket.
- `source`— Mengidentifikasi layanan yang menghasilkan acara.
- `time`- Waktu peristiwa itu terjadi.
- `region`— Mengidentifikasi ember. Wilayah AWS
- `resources`— Sebuah array JSON yang berisi Amazon Resource Name (ARN) dari bucket.
- `detail`— Objek JSON yang berisi informasi tentang acara tersebut.

Jalankan acara meliputi bidang-bidang berikut:

- `uuid`— Pengidentifikasi unik secara universal untuk menjalankan.
- `workflowId`— Pengidentifikasi alur kerja dari alur kerja yang terkait dengan proses ini.
- `workflowName`— Nama alur kerja yang terkait dengan proses ini..
- `runId`— Jalankan pengenal.
- `runName`— Jalankan nama.
- `runOutputUri`— URI untuk tempat run akan menulis data outputnya.

Contoh pesan acara

Contoh berikut adalah peristiwa untuk perubahan status run, menunjukkan bidang tambahan.

```
{
    "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"
    }
}
```

Contoh berikut adalah peristiwa untuk perubahan status tugas.

```
{
    "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"
    }
}
```

Berikut ini adalah contoh peristiwa untuk perubahan status set baca.

```
{
  "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"
  }
}
```

Peristiwa serupa dibuat untuk pekerjaan impor toko varian.

```
{
    "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"
    }
}
```

Berikut ini adalah peristiwa untuk perubahan status pekerjaan impor.

```
{
    "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"
    }
}
```

Pemecahan Masalah

Topik berikut dapat membantu Anda memecahkan masalah yang Anda temui saat menggunakan HealthOmics alur kerja dan penyimpanan data.

Topik

Memecahkan masalah alur kerja

Topik

Bagaimana cara memecahkan masalah yang gagal?

Gunakan operasi GetRunAPI untuk mengambil alasan kegagalan. Untuk informasi selengkapnya, lihat Jalankan alasan kegagalan.

Bagaimana cara memecahkan masalah tugas yang gagal?

Tinjau kode kesalahan dari pesan kegagalan tugas untuk memahami kegagalan. Tinjau log tugas CloudWatch untuk melihat pesan pencatatan terperinci untuk tugas tersebut. Jika Anda tidak mendapatkan pesan log terperinci, Anda dapat merevisi alur kerja Anda untuk mengeluarkan pernyataan log tambahan. Untuk informasi selengkapnya, lihat Pemantauan HealthOmics dengan CloudWatch Log.

Di mana saya menemukan log mesin untuk proses yang berhasil diselesaikan?

HealthOmics menerbitkan log ke CloudWatch untuk gagal berjalan saja. Jika proses berhasil diselesaikan, HealthOmics kirimkan log mesin ke bucket Amazon S3 Anda. Untuk informasi selengkapnya, lihat Log di Amazon S3.

Bagaimana saya bisa mengurangi ukuran parameter input untuk alur kerja?

Anda dapat menentukan hingga 50 KB parameter input untuk alur kerja. Anda dapat menggunakan impor direktori atau lembar sampel untuk tetap berada dalam batasan ukuran ini. Untuk informasi selengkapnya, lihat Mengelola ukuran parameter run.

Mengapa lari saya tidak selesai?

Jika ada masalah dengan kode Anda dan prosesnya belum keluar dengan benar, proses Anda bisa menjadi tidak responsif atau “macet”. Untuk informasi selengkapnya tentang cara mencegah dan menangkap proses yang tidak responsif, lihatPanduan untuk proses yang tidak responsif.

Memecahkan masalah caching panggilan

Topik berikut dapat membantu Anda memecahkan masalah yang Anda temui dengan caching panggilan.

Topik

Mengapa run saya tidak disimpan ke cache?

1. Verifikasi bahwa proses dikonfigurasi untuk menggunakan cache dengan memeriksa bidang CacheID dalam respons operasi GetRun API. Menggunakan CLI, jalankan perintah ini:. `aws omics get-run –id <run_id>`
2. Jika proses berhasil, verifikasi perilaku cache yang dikembalikan dalam GetRun respons adalah CACHE_ALWAYS. Jika perilaku cache diatur ke CACHE_ON_FAILURE, run hanya akan menyimpan ke cache ketika gagal.

Mengapa tugas tidak menggunakan entri cache?

<cache_id><cache_uuid>Dalam grup `/aws/omics/WorkflowLog` CloudWatch log, buka aliran log untuk menjalankan cache: Runcache//.

1. Verifikasi bahwa proses sebelumnya membuat entri cache untuk tugas yang Anda harapkan akan di-cache. Jalankan yang telah disimpan ke cache akan direkam dengan pesan log CACHE_ENTRY_CREATED.
2. Temukan log CACHE_MISS untuk tugas dan jalankan yang selesai. Jika tidak ada entri log, periksa apakah run telah dikonfigurasi untuk menggunakan cache.
3. Jika entri cache dibuat, verifikasi bahwa intisari CPUs, memori, GPUs dan wadah identik untuk kedua tugas. Tugas ARN untuk tugas yang membuat entri cache ada di pesan log.
4. Jika persyaratan komputasi untuk kedua tugas cocok, verifikasi bahwa input tidak berubah di antara tugas. Untuk melakukan ini, buka log mesin. Jika run memiliki status GAGAL, log akan berada di Cloudwatch Log Group/. aws/omics/WorkflowLog Jika tidak, log mesin dapat ditemukan di direktori output run.

Mengapa caching panggilan untuk tugas dinonaktifkan?

Periksa apakah tugas dikonfigurasi untuk memilih keluar dari caching menggunakan fitur mesin alur kerja:

- Untuk alur kerja WDL: Periksa apakah tugas memiliki set volatile di bagian meta `true`
- Untuk alur kerja Nextflow: Periksa apakah tugas memiliki direktif cache yang disetel ke `false`
- Untuk alur kerja CWL: Periksa apakah tugas memiliki EnableReuse disetel ke fitur `false` WorkReuse

Memecahkan masalah penyimpanan data

Topik

- Mengapa S3 GetObject gagal pada set baca saya?
- Mengapa saya tidak dapat melihat toko anotasi atau toko varian saya di Athena?
- Mengapa saya tidak dapat mengakses penyimpanan data saya di Athena?

Mengapa S3 GetObject gagal pada set baca saya?

Paling umum, kegagalan adalah karena izin yang hilang. Izin membaca Sequence store S3 adalah konfigurasi dua arah yang memerlukan kebijakan akses S3 penyimpanan urutan untuk mengizinkan akses dan prinsipal IAM memiliki kebijakan yang dilampirkan yang memungkinkan akses. Untuk detail lebih lanjut tentang persyaratan kebijakan, lihatIzin untuk akses data menggunakan Amazon S3 URIs. Periksa apakah konfigurasi berikut sudah ada:

- Kebijakan akses S3 penyimpanan urutan telah secara eksplisit mengizinkan akses ke prinsipal IAM atau akar akun kepala sekolah.
- Periksa apakah kepala sekolah IAM memiliki kebijakan yang secara eksplisit memberikan izin ke sumber daya yang diakses. Perhatikan bahwa kebijakan utama IAM harus menggunakan Access Point ARN dan bukan jalur berbasis Alias Access point saat menentukan izin dan ARN dalam kondisi dan tidak digunakan untuk menentukan sumber daya.
- Jika toko Anda menggunakan kunci yang dikelola pelanggan (CMK-KMS), pastikan bahwa prinsipal IAM memiliki izin kms: dekripsi pada kunci tersebut. Lihat panduan akses lintas akun KMS untuk mengonfigurasi penggunaan di seluruh akun.

Jika Anda memiliki kebijakan yang menggunakan kontrol akses berbasis tag, pastikan hal berikut:

- Pastikan bahwa toko urutan telah selesai menyinkronkan tag. Untuk ini, status toko harus active dan tidakupdating.
- Pastikan tidak ada kesalahan ketik pada kunci tag atau nilai kunci pada set baca dan kebijakan.

Mengapa saya tidak dapat melihat toko anotasi atau toko varian saya di Athena?

Di Lake Formation, pastikan untuk membuat tautan sumber daya berdasarkan toko yang dibagikan kepada Anda. Setelah Anda membuat tautan sumber daya yang memiliki izin untuk diakses, toko akan terlihat di Athena. Untuk informasi selengkapnya, lihat Mengkonfigurasi Lake Formation untuk digunakan HealthOmics.

Mengapa saya tidak dapat mengakses penyimpanan data saya di Athena?

Jika anotasi atau toko varian Anda terlihat tetapi Anda menerima pesan kesalahan yang mengatakan bahwa akses ditolak, periksa versi mesin kueri yang Anda gunakan. Hanya kueri yang dijalankan

menggunakan engine versi 3 yang didukung. Untuk membaca lebih lanjut tentang versi mesin kueri Athena, lihat dokumentasi Amazon Athena.

Pemecahan masalah dengan Amazon Q CLI

Amazon Q CLI dapat membantu merampingkan proses pemecahan masalah Anda dengan:

- Menganalisis alur kerja berjalan dan men-debug kegagalan tugas
- Mengumpulkan log dan pesan kesalahan yang relevan
- Membuat kasus AWS Support dengan semua log debugging yang diperlukan terpasang
- Menyunting informasi identitas pribadi (PII) dari informasi yang dikirimkan ke Support AWS

Untuk informasi selengkapnya tentang menggunakan Amazon Q CLI AWS HealthOmics untuk pemecahan masalah dan membuat kasus dukungan, lihat tutorial AI generatif HealthOmics Agentic di. GitHub

> **Warning**
>
> Saat bekerja dengan Amazon Q CLI, tinjau semua konten yang dihasilkan dan tindakan yang diusulkan sebelum melanjutkan. Berikan umpan balik untuk meningkatkan kualitas respons dan untuk mencocokkan persyaratan alur kerja Anda. Untuk informasi selengkapnya, lihat Pertimbangan keamanan dan praktik terbaik untuk Amazon Q.

Kuota untuk AWS HealthOmics

AWS mengisi akun Anda dengan nilai default untuk HealthOmics kuota. Kecuali dinyatakan lain, setiap nilai kuota adalah nilai maksimum per wilayah.

> **Important**
>
> Anda dapat meminta kenaikan ke sebagian besar kuota layanan dan kuota API. Lihat topik berikut untuk detailnya.

Topik

HealthOmics kuota layanan

Tabel di bawah ini mencantumkan kuota HealthOmics layanan, bersama dengan nilai defaultnya. Untuk melihat kuota saat ini untuk setiap Wilayah, buka konsol Service Quotas.

> **Important**
>
> Anda dapat meminta peningkatan kuota yang dapat disesuaikan menggunakan konsol Service Quotas.

Untuk informasi selengkapnya tentang kuota layanan, lihat Meminta peningkatan kuota dalam Panduan Pengguna Service Quotas. Untuk kuota yang tidak tersedia di konsol Service Quotas, gunakan formulir peningkatan kuota.

Nama	Default	Dapat disesuaikan	Deskripsi
Analytics - Toko anotasi maksimum	Setiap Wilayah yang didukung: 10	Ya	Jumlah maksimum toko anotasi di wilayah saat ini AWS
Analytics - Pekerjaan impor penyimpanan varian atau anotasi bersamaan maksimum	Setiap Wilayah yang didukung: 5	Ya	Jumlah maksimum pekerjaan impor bersamaan di wilayah saat ini AWS
Analytics - File maksimum per pekerjaan impor toko varian	Setiap Wilayah yang didukung: 1.000	Ya	Jumlah maksimum file per varian pekerjaan impor di AWS wilayah saat ini
Analytics - Saham maksimum per toko anotasi	Setiap Wilayah yang didukung: 10	Ya	Jumlah maksimum saham per toko anotasi di wilayah saat ini AWS
Analytics - Saham maksimum per toko varian	Setiap Wilayah yang didukung: 10	Ya	Jumlah maksimum saham per toko varian di AWS wilayah saat ini
Analytics - Ukuran maksimum setiap file dalam pekerjaan impor varian	Setiap Wilayah yang didukung: 20 Gigabytes	Ya	Ukuran maksimum satu file dalam pekerjaan impor varian di AWS wilayah saat ini
Analytics - Ukuran maksimum setiap file dalam pekerjaan impor anotasi	Setiap Wilayah yang didukung: 20 Gigabytes	Ya	Ukuran maksimum satu file dalam pekerjaan impor anotasi di wilayah saat ini AWS

Nama	Default	Dapat disesuaikan	Deskripsi
Analytics - Toko varian maksimum	Setiap Wilayah yang didukung: 10	Ya	Jumlah maksimum toko varian di AWS wilayah saat ini
Analytics - Versi maksimum per toko anotasi	Setiap Wilayah yang didukung: 10	Ya	Jumlah maksimum versi per penyimpanan anotasi di wilayah saat ini AWS
Konfigurasi - Konfigurasi maksimum	Setiap Wilayah yang didukung: 10	Ya	Jumlah maksimum konfigurasi di AWS wilayah saat ini.
Penyimpanan - Pekerjaan aktivasi set baca bersamaan maksimum	Setiap Wilayah yang didukung: 25	Ya	Jumlah maksimum pekerjaan aktivasi set baca bersamaan di wilayah saat ini AWS
Penyimpanan - Urutan bersamaan maksimum dan pekerjaan ekspor toko referensi	Setiap Wilayah yang didukung: 5	Ya	Jumlah maksimum pekerjaan ekspor bersamaan dari urutan atau toko referensi di wilayah saat ini AWS
Penyimpanan - Urutan bersamaan maksimum atau pekerjaan impor toko referensi	Setiap Wilayah yang didukung: 5	Ya	Jumlah maksimum pekerjaan impor bersamaan untuk urutan atau penyimpanan referensi di wilayah saat ini AWS
Penyimpanan - Set baca maksimum per toko urutan	Setiap Wilayah yang didukung: 1.000.000	Ya	Jumlah maksimum set baca di penyimpanan urutan di AWS wilayah saat ini

Nama	Default	Dapat disesuaikan	Deskripsi
Penyimpanan - Referensi maksimum per toko referensi	Setiap Wilayah yang didukung: 50	Ya	Jumlah maksimum referensi di toko referensi di AWS wilayah saat ini
Penyimpanan - Toko urutan maksimum	Setiap Wilayah yang didukung: 20	Ya	Jumlah maksimum penyimpanan urutan di AWS wilayah saat ini
Alur kerja - Aktif maksimum GPUs	Setiap Wilayah yang didukung: 12	Ya	Jumlah maksimum aktif bersamaan GPUs di AWS wilayah saat ini. Di us-east-1 dan us-west-2, permintaan peningkatan kuota untuk nilai hingga 500 secara otomatis disetujui.
Alur kerja - Proses aktif bersamaan maksimum menggunakan penyimpanan run dinamis	Setiap Wilayah yang didukung: 50	Ya	Jumlah maksimum proses aktif menggunakan penyimpanan run dinamis di AWS wilayah saat ini. Permintaan peningkatan kuota untuk nilai hingga 200 secara otomatis disetujui.

Nama	Default	Dapat disesuaikan	Deskripsi
Alur kerja - Proses aktif bersamaan maksimum menggunakan penyimpanan run statis	Setiap Wilayah yang didukung: 10	Ya	Jumlah maksimum proses aktif menggunakan penyimpanan run statis di AWS wilayah saat ini. Permintaan peningkatan kuota untuk nilai hingga 50 secara otomatis disetujui.
Alur kerja - Tugas bersamaan maksimum per proses	Setiap Wilayah yang didukung: 25	Ya	Jumlah maksimum tugas bersamaan di setiap proses di AWS wilayah saat ini. Di us-east-1 dan us-west-2, permintaan peningkatan kuota untuk nilai hingga 100 secara otomatis disetujui.
Alur kerja - Durasi lari maksimum	Setiap Wilayah yang didukung: 604,800 Detik	Ya	Durasi lari alur kerja maksimum di AWS wilayah saat ini.
Alur Kerja - Lari maksimum (aktif atau tidak aktif)	Setiap Wilayah yang didukung: 100.000	Ya	Jumlah maksimum run (aktif atau tidak aktif) di AWS wilayah saat ini.
Alur Kerja - Saham maksimum per alur kerja	Setiap Wilayah yang didukung: 100	Ya	Jumlah maksimum saham per alur kerja di wilayah saat ini AWS

Nama	Default	Dapat disesuaikan	Deskripsi
Alur kerja - Kapasitas penyimpanan lari statis maksimum per proses	Setiap Wilayah yang didukung: 9.600	Ya	Kapasitas penyimpanan run statis maksimum dalam gibibytes (GiB) untuk setiap proses di wilayah saat ini. AWS Di us-east-1 dan us-west-2, permintaan peningkatan kuota untuk nilai hingga 50.000 secara otomatis disetujui.
Alur kerja - Alur kerja maksimum	Setiap Wilayah yang didukung: 1.000	Ya	Jumlah maksimum alur kerja di AWS wilayah saat ini.
Alur Kerja - Transaksi per detik (TPS) untuk operasi StartRun	Setiap Wilayah yang didukung: 1	Ya	Transaksi maksimum per detik (TPS) untuk StartRun operasi di AWS wilayah saat ini.

HealthOmics kuota ukuran tetap

Selain ituHealthOmics kuota layanan, HealthOmics termasuk kuota yang memiliki ukuran tetap. Anda tidak dapat meminta peningkatan untuk nilai-nilai ini.

Kecuali dinyatakan lain, setiap kuota mencantumkan nilai maksimum per wilayah.

Topik

- HealthOmics kuota ukuran tetap analitik
- HealthOmics penyimpanan kuota ukuran tetap
- HealthOmics alur kerja kuota ukuran tetap
- HealthOmics Alur kerja Ready2Run kuota ukuran tetap

HealthOmics kuota ukuran tetap analitik

Tabel berikut menunjukkan nilai maksimum yang didukung untuk kuota analitik. Nilai-nilai ini tidak dapat disesuaikan.

Nama	Deskripsi	Maksimum	Disesuaikan Ya/Tidak
Analytics - File maksimum per pekerjaan impor toko anotasi	Jumlah maksimum file per pekerjaan impor anotasi.	1	Tidak

HealthOmics penyimpanan kuota ukuran tetap

Tabel berikut menunjukkan nilai maksimum yang didukung untuk file penyimpanan. Nilai-nilai ini tidak dapat disesuaikan.

Nama	Deskripsi	Maksimum	Disesuaikan Ya/Tidak
Penyimpanan - Ukuran kebijakan sumber daya akses S3 maksimum	Ukuran maksimum kebijakan sumber daya akses S3	15 KB	Tidak
Penyimpanan - Tag level set maksimum yang disebarkan	Jumlah maksimum kunci tag tingkat set, per toko, yang dipropogasi ke objek S3	5	Tidak
Penyimpanan - Set baca maksimum per pekerjaan aktivasi	Jumlah maksimum set baca per pekerjaan aktivasi.	20	Tidak
Penyimpanan - Set baca maksimum per pekerjaan ekspor	Jumlah maksimum set baca per pekerjaan ekspor.	100	Tidak

Nama	Deskripsi	Maksimum	Disesuaikan Ya/Tidak
Penyimpanan - Set baca maksimum per pekerjaan impor	Jumlah maksimum set baca per pekerjaan impor.	100	Tidak
Penyimpanan - Toko referensi maksimum	Jumlah maksimum toko referensi.	1	Tidak
Penyimpanan - Ukuran bagian maksimum untuk upload langsung	Ukuran bagian maksimum untuk upload langsung ke toko urutan.	100 MB	Tidak
Penyimpanan - Bagian maksimum dalam file untuk diunggah langsung	Jumlah maksimum bagian dalam file untuk diunggah langsung ke toko urutan.	10.000	Tidak
Penyimpanan - Ukuran referensi maksimum	Ukuran maksimum file referensi yang dapat diimpor ke toko referensi.	15 GB	Tidak
Penyimpanan - Ukuran sumber set baca maksimum	Ukuran maksimum file sumber tunggal dalam set baca yang dapat diimpor ke toko urutan.	976 GB	Tidak

HealthOmics alur kerja kuota ukuran tetap

Tabel berikut menunjukkan nilai maksimum yang didukung untuk kuota alur kerja. Nilai-nilai ini tidak dapat disesuaikan.

Nama	Deskripsi	Ukuran maksimum	Disesuaikan Ya/Tidak
Alur kerja - Grup lari maksimum	Jumlah maksimum grup lari.	1000	Tidak
Alur Kerja - Cache run maksimum	Jumlah maksimum cache run yang dapat Anda buat untuk satu akun. Satu atau lebih run dapat berbagi cache run yang sama. Tidak ada kuota untuk jumlah run yang HealthOmics dapat di-cache per akun.	1000	Tidak
Alur kerja - Versi alur kerja maksimum	Jumlah maksimum versi alur kerja per alur kerja.	1000	Tidak
Alur kerja - ukuran wadah instance CPU	Ukuran gambar kontainer maksimum untuk instance CPU.	45 GiB	Tidak
Alur kerja - ukuran wadah instans GPU	Ukuran gambar kontainer maksimum untuk instance GPU.	95 GiB	Tidak
Instans GPU /dev/shm memori bersama	Jumlah maksimum memori bersama per instans GPU.	8 GB per GPU	Tidak
Alur kerja - Jalankan file parameter	Ukuran maksimum file parameter run.	50.000 byte	Tidak

Nama	Deskripsi	Ukuran maksimum	Disesuaikan Ya/Tidak
Alur Kerja - File templat parameter alur kerja	Jumlah maksimum entri dan ukuran file maksimum untuk file template parameter alur kerja. Kuota ini berlaku untuk alur kerja yang Anda buat menggunakan konsol atau API.	1.000 entri, 400 KB	Tidak
Alur Kerja - Ukuran file definisi alur kerja - API	Ukuran maksimum file definisi alur kerja saat Anda membuat alur kerja menggunakan operasi API atau SDK. AWS	100 MB	Tidak
Alur kerja - Ukuran file definisi alur kerja - Konsol (unggah langsung)	Ukuran maksimum file definisi alur kerja yang dapat Anda berikan sebagai unggahan langsung, saat Anda membuat alur kerja menggunakan konsol.	4,4 MB	Tidak
Alur kerja - Ukuran file definisi alur kerja - Konsol (unggah dari Amazon S3)	Ukuran maksimum file definisi alur kerja yang dapat Anda berikan sebagai unggahan dari Amazon S3, saat Anda membuat alur kerja menggunakan konsol.	100 MB	Tidak

Nama	Deskripsi	Ukuran maksimum	Disesuaikan Ya/Tidak
Alur kerja - Ukuran repositori	Ukuran maksimum repositori kode eksternal.	1 GiB	Tidak
Alur Kerja - Repositori ukuran file individual	Ukuran maksimum file individual dari repositori kode eksternal.	100 MiB	Tidak
Alur kerja - ukuran file README	Ukuran maksimum file README.	500 KiB	Tidak

Untuk saran tentang cara mengurangi ukuran file parameter run Anda, lihatMengelola ukuran parameter run.

HealthOmics Alur kerja Ready2Run kuota ukuran tetap

Setiap alur kerja Ready2Run memiliki ukuran file input maksimum. Dalam tabel berikut, satuan ukuran file tercantum dalam Gibibytes (GiB). Ukuran file maksimum ini tidak dapat disesuaikan.

Nama alur kerja Ready2Run	Ukuran file masukan maksimum (GiB)	Dapat disesuaikan (Ya/Tidak)
AlphaFold untuk 601-1200 residu	1	Tidak
AlphaFold hingga 600 residu	1	Tidak
Basis2Fastq untuk 2x150	1000	Tidak
Basis2Fastq untuk 2x300	1000	Tidak
Basis2Fastq untuk 2x75	500	Tidak
ESMFold hingga 800 residu	1	Tidak
GATK-BP fq2bam	64	Tidak

Nama alur kerja Ready2Run	Ukuran file masukan maksimum (GiB)	Dapat disesuaikan (Ya/Tidak)
GATK-BP Germline bam2vcf untuk genom 30x	39	Tidak
GATK-BP Germline fq2vcf untuk genom 30x	64	Tidak
GATK-BP Somatik WES bam2vcf	86	Tidak
NVIDIA Parabricks BAM2 FQ2 BAM WGS hingga 30X	80	Tidak
NVIDIA Parabricks BAM2 FQ2 BAM WGS hingga 50X	120	Tidak
NVIDIA Parabricks BAM2 FQ2 BAM WGS hingga 5X	20	Tidak
NVIDIA Parabricks FQ2 BAM WGS hingga 30X	71	Tidak
NVIDIA Parabricks FQ2 BAM WGS hingga 50X	137	Tidak
NVIDIA Parabricks FQ2 BAM WGS hingga 5X	13	Tidak
NVIDIA Parabricks Germline DeepVariant WGS hingga 30X	71	Tidak
NVIDIA Parabricks Germline DeepVariant WGS hingga 50X	137	Tidak
NVIDIA Parabricks Germline DeepVariant WGS hingga 5X	12	Tidak

Nama alur kerja Ready2Run	Ukuran file masukan maksimum (GiB)	Dapat disesuaikan (Ya/Tidak)
NVIDIA Parabricks Germline HaplotypeCaller WGS hingga 30X	71	Tidak
NVIDIA Parabricks Germline HaplotypeCaller WGS hingga 50X	137	Tidak
NVIDIA Parabricks Germline HaplotypeCaller WGS hingga 5X	13	Tidak
NVIDIA Parabricks Somatic Mutect2 WGS hingga 50X	196	Tidak
sc RNAseq dengan Kallisto BUStools	119	Tidak
sc RNAseq dengan Salmon Alevin-goreng	119	Tidak
sc RNAseq dengan STARsolo	119	Tidak
Sentieon Germline BAM WES hingga 300x	9	Tidak
Sentieon Germline BAM WGS hingga 32x	18	Tidak
Sentieon Germline FASTQ WES hingga 100x	5	Tidak
Sentieon Germline FASTQ WES hingga 300x	26	Tidak

Nama alur kerja Ready2Run	Ukuran file masukan maksimum (GiB)	Dapat disesuaikan (Ya/Tidak)
Sentieon Germline FASTQ WGS hingga 32x	51	Tidak
Sentieon LongRead untuk ONT	25	Tidak
Sentieon LongRead untuk PacBio HiFi	58	Tidak
Sentieon Somatik WES	50	Tidak
Sentieon Somatik WGS	113	Tidak
Ultima Genomics hingga DeepVariant 40x	91	Tidak

HealthOmics Kuota API

HealthOmics memiliki kuota berikut yang terkait dengan operasi API. Jika ditunjukkan, kuota dapat disesuaikan. Untuk meminta kenaikan, gunakan formulir peningkatan kuota.

Untuk setiap operasi API yang tercantum, kuota adalah transaksi maksimum per detik (TPS) untuk operasi API tersebut di setiap Wilayah.

Topik

Kuota API umum

Tabel berikut mencantumkan operasi API umum yang berlaku untuk lebih dari satu kategori (penyimpanan, alur kerja, dan analitik).

Operasi API	TPS maksimum default	Dapat disesuaikan (Ya/Tidak)
AcceptShare, CreateSha re, DeleteShare, GetShare, ListShares	1 TPS	Ya

Kuota API penyimpanan

Tabel berikut mencantumkan operasi API penyimpanan.

Operasi API penyimpanan	TPS maksimum default	Dapat disesuaikan (Ya/Tidak)
CreateSequenceStore, UpdateSequenceStore, DeleteSequenceStore, CreateReferenceStore, DeleteReferenceStore	1 TPS	Ya
BatchDeleteReadSet, DeleteReference	1 TPS	Ya
CreateMultipartReadSetUploa d, CompleteMultipartR eadSetUpload, AbortMult ipartReadSetUpload	1 TPS	Tidak
GetS3, putS3AccessPolicy, AccessPolicy Hapus3 AccessPolicy	1 TPS	Ya
GetReference	10 TPS	Ya
UploadReadSetPart	10 TPS	Ya
GetReadSet	30 TPS	Ya
GetSequenceStore, ListSeque nceStores	5 TPS	Ya

Operasi API penyimpanan	TPS maksimum default	Dapat disesuaikan (Ya/Tidak)
GetReadSetMetadata, ListReadSets	5 TPS	Ya
StartReadSetImportJob, GetReadSetImportJob, ListReadSetImportJobs	5 TPS	Ya
StartReadSetExportJob, GetReadSetExportJob, ListReadSetExportJobs	5 TPS	Ya
ListReferenceStores	5 TPS	Ya
StartReferencetImportJob, GetReferenceImportJob, ListReferenceImportJobs	5 TPS	Ya
ListReferences, GetReferenceMetadata	5 TPS	Ya
StartReadsetActivationJob	5 TPS	Ya
ListReadsetActivationJobs, GetReadSetActivationJob	5 TPS	Ya
ListMultipartReadSetUploads, ListReadSetUploadParts	5 TPS	Ya
TagResource, UntagResource, ListTagsForResource	5 TPS	Ya

Kuota API alur kerja

Tabel berikut mencantumkan operasi API alur kerja.

Operasi API alur kerja	TPS maksimum default	Dapat disesuaikan (Ya/Tidak)
StartRun	1 TPS	Ya
CreateWorkflow	5 TPS	Ya
CancelRun, DeleteRun , GetRun, GetRunTask, ListRunTasks, ListRuns	10 TPS	Ya
CreateRunGroup, DeleteRun Group, GetRunGroup, ListRunGroups, UpdateRun Group	10 TPS	Ya
CreateRunCache, UpdateRun Cache, DeleteRunCache, GetRunCache, ListRunCaches	10 TPS	Ya
DeleteWorkflow, GetWorkfl ow, ListWorkflows, UpdateWor kflow	10 TPS	Ya

Kuota API Analytics

Tabel berikut mencantumkan operasi API analitik.

Operasi API Analytics	TPS maksimum default	Dapat disesuaikan (Ya/Tidak)
CreateVariantStore, DeleteVar iantStore, GetVariantStore, ListVariantStores, UpdateVar iantStore	1 TPS	Tidak
StartVariantImportJob, CancelVariantImportJob,	1 TPS	Tidak

Operasi API Analytics	TPS maksimum default	Dapat disesuaikan (Ya/Tidak)
GetVariantImportJob, ListVariantImportJobs		
CreateAnnotationStore, DeleteAnnotationStore, GetAnnotationStore, ListAnnotationStores, UpdateAnnotationStore	1 TPS	Tidak
StartAnnotationImportJob, ListAnnotationImportJobs, GetAnnotationImportJob, CancelAnnotationImportJob	1 TPS	Tidak

Riwayat dokumen untuk Panduan HealthOmics Pengguna

Tabel berikut menjelaskan rilis dokumentasi untuk HealthOmics.

Perubahan	Deskripsi	Tanggal
AWS HealthOmics toko varian dan toko anotasi tidak lagi terbuka untuk pelanggan baru.	AWS HealthOmics toko varian dan toko anotasi tidak lagi terbuka untuk pelanggan baru. Untuk informasi selengkapnya, lihat perubahan ketersediaan toko AWS HealthOmics varian dan anotasi toko.	November 7, 2025
AWS HealthOmics toko varian dan toko anotasi tidak akan lagi dibuka untuk pelanggan baru mulai 7 November 2025.	AWS HealthOmics toko varian dan toko anotasi tidak akan lagi dibuka untuk pelanggan baru mulai 7 November 2025. Jika Anda ingin menggunak an toko varian atau toko anotasi, daftar sebelum tanggal tersebut. Pelanggan yang sudah ada dapat terus menggunakan layanan ini seperti biasa. Untuk informasi selengkapnya, lihat perubahan ketersediaan toko AWS HealthOmics varian dan anotasi toko.	Oktober 7, 2025
Fitur Baru	HealthOmics menambahk an dukungan untuk alur kerja untuk menyinkronkan repositor i Amazon ECR pribadi dengan registri hulu. Untuk mempelaja ri selengkapnya, lihat Gambar	Agustus 28, 2025

	kontainer untuk alur kerja pribadi di HealthOmics.	
Fitur integrasi README dan repositori baru	Menambahkan dukungan untuk membuat alur kerja dari repositori kode eksternal dan file README.	Juli 24, 2025
Fitur Baru	HealthOmics menambahkan dukungan untuk interpolasi parameter otomatis Nextflow. Untuk mempelajari selengkapnya, lihat File templat parameter untuk HealthOmics alur kerja.	Juni 27, 2025
Fitur Baru	HealthOmics menambahkan dukungan untuk alur kerja untuk menginterpolasi parameter run dari file definisi alur kerja WDL. Untuk mempelajari selengkapnya, lihat File templat parameter untuk HealthOmics alur kerja.	30 Mei 2025
Fitur Baru	HealthOmics menambahkan dukungan untuk pembuatan versi alur kerja. Untuk mempelajari selengkapnya, lihat Pembuatan versi alur kerja. HealthOmics	April 18, 2025
Fitur Baru	HealthOmics menambahkan throughput elastis untuk penyimpanan run dinamis. Untuk mempelajari selengkapnya, lihat Menjalankan jenis penyimpanan di HealthOmics.	April 16, 2025

Fitur Baru	HealthOmics menambahkan kontrol akses berbasis atribut untuk lokasi Sequence Store S3, dan kemampuan untuk menyinkronkan hingga lima tag read-set ke objek Sequence Store S3. Untuk mempelajari selengkapnya, lihat Membuat toko HealthOmics urutan.	November 22, 2024
Fitur Baru	HealthOmics menambahkan dukungan untuk caching panggilan, juga dikenal sebagai resume, untuk alur kerja pribadi. Untuk mempelajari lebih lanjut, lihat Caching panggilan.	November 20, 2024
Fitur Baru	HealthOmics menambahkan bidang API baru untuk membantu Anda memetakan antara pekerjaan input penyimpanan urutan dan set baca.	Agustus 29, 2024
Fitur Baru	HealthOmics menambahkan dukungan untuk mengelola versi Nextflow. Untuk mempelajari lebih lanjut, lihat versi Nextflow.	Agustus 14, 2024
Fitur Baru	HealthOmics menambahkan dukungan untuk alur kerja bersama dan penyimpanan run dinamis.	April 30, 2024

Fitur Baru	HealthOmics menambahkan dukungan untuk akses Amazon S3 ke penyimpanan referensi dan urutan, dan dukungan untuk. SHA256 ETags	April 15, 2024
Fitur Baru	HealthOmics menambahkan tag entitas (ETags) untuk toko urutan.	Oktober 6, 2023
Fitur Baru	HealthOmics menambahkan versi toko anotasi dan berbagi toko analitik.	15 Agustus 2023
Fitur Baru	HealthOmics menambahkan Common Workflow Language (CWL) sebagai bahasa yang didukung untuk alur kerja. HealthOmics	Juni 30, 2023
Fitur Baru	HealthOmics menambahkan alur kerja Ready2Run baru, dukungan GPU untuk alur kerja, penguraian data untuk penyimpanan anotasi, unggah langsung ke penyimpanan, dan integrasi dengan. HealthOmics EventBridge	15 Mei 2023
Kebijakan terkelola baru	HealthOmics menambahkan kebijakan terkelola baru yang menyediakan akses penuh. Untuk mempelajari selengkapnya, lihat kebijakan terkelola AWS.	23 Februari 2023

Kebijakan terkelola baru	HealthOmics menambahkan kebijakan terkelola baru yang membatasi akses hanya untuk dibaca. Untuk mempelajari selengkapnya, lihat kebijakan terkelola AWS.	29 November 2022
Rilis awal	Rilis awal Panduan HealthOmics Pengguna	29 November 2022

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.