aws

Guide de l’utilisateur

# AWS HealthOmics

Version latest

# AWS HealthOmics: Guide de l’utilisateur

# Table of Contents

# Qu'est-ce que c'est AWS HealthOmics ?

AWS HealthOmics est un service éligible à la loi HIPAA qui accélère les tests de diagnostic clinique, la découverte de médicaments et la recherche agricole en gérant entièrement l'infrastructure complexe qui sous-tend vos flux de travail bioinformatiques. HealthOmics prend en charge les langages de flux de travail standard (WDL, Nextflow, CWL) et adapte facilement l'infrastructure bioinformatique pour prendre en charge les données de dizaines de milliers de tests par jour, le tout avec un coût par échantillon prévisible. HealthOmics gère les complexités techniques telles que la gestion des ressources informatiques et la maintenance des moteurs de flux de travail afin que vous puissiez vous concentrer entièrement sur les avancées scientifiques.

Rubriques



## Avis important

HealthOmics est uniquement destiné au transfert, au stockage, au formatage ou à l'affichage de données, ainsi qu'à la fourniture d'une infrastructure et d'un support de configuration pour la gestion des flux de travail. HealthOmics ne remplace pas un avis médical, un diagnostic ou un traitement professionnel et n'est pas destiné à guérir, traiter, atténuer, prévenir ou diagnostiquer une maladie ou un problème de santé. Il vous incombe de procéder à un examen humain dans le cadre de toute utilisation de tout produit tiers destiné à éclairer la prise de décisions cliniques AWS HealthOmics, y compris en association avec celui-ci.

## HealthOmics features

Principaux cas d'utilisation pour HealthOmics :

- Diagnostic clinique — Créez et adaptez des flux de travail de tests diagnostiques avec des coûts prévisibles et une infrastructure entièrement gérée qui évolue avec le volume de vos tests.
- Découverte de médicaments — Accélérez la recherche thérapeutique en orchestrant des modèles de base biologiques à grande échelle, permettant ainsi une itération rapide sur des millions de candidats potentiels.
- Recherche agricole — Améliorez les caractéristiques des cultures telles que la tolérance à la sécheresse et la résistance aux ravageurs grâce à des flux de travail basés sur l'IA qui améliorent la sécurité alimentaire et la productivité agricole.

Principaux avantages de HealthOmics :

- Évolutivité — Faites évoluer les flux de travail sur plus de 100 000 machines virtuelles simultanées CPUs pour prendre en charge des dizaines de milliers de tests par jour, sans aucune gestion d'infrastructure et avec un coût par échantillon prévisible.
- Concentrez-vous sur la science et non sur l'infrastructure : utilisez des langages de flux de travail familiers APIs tout en gérant AWS automatiquement l'orchestration de l'infrastructure et la gestion des données en arrière-plan.
- Maintien de la conformité — Des pistes d'audit complètes, le suivi de la provenance des données et une infrastructure conforme à la loi HIPAA conçue pour les flux de travail cliniques out-of-the-box soutiennent le développement de solutions conformes aux exigences réglementaires.

HealthOmics se compose de trois éléments principaux :

- HealthOmics flux de travail — Exécutez des calculs bioinformatiques sur une infrastructure automatiquement provisionnée et dimensionnée.
- HealthOmics stockage — Stockez et partagez des pétaoctets de données génomiques de manière efficace à un faible coût par gigabase.
- HealthOmics analyse — Préparer les données génomiques pour les analyses multiomiques et multimodales.

Utilisez ces composants indépendamment ou combinez-les pour obtenir une end-to-end solution.

# HealthOmics concepts

Cette rubrique décrit les définitions des principaux concepts et termes spécifiques à HealthOmics, afin de vous aider à comprendre la terminologie HealthOmics utilisée dans ce guide.

Rubriques



## Flux de travail

Avec HealthOmics Workflows, vous pouvez traiter et analyser vos données génomiques.

- Flux de travail : définition globale d'un processus de bout en bout, y compris les paramètres et les références aux outils. Les définitions de flux de travail peuvent être exprimées sous la forme WDL, Nextflow ou CWL. Chaque flux de travail créé possède un identifiant unique.
- Exécuter : appel unique d'un flux de travail. Une exécution individuelle utilise les données d'entrée que vous avez définies et produit une sortie. Chaque course créée possède un identifiant unique.
- Tâche : processus individuels au cours d'une exécution. HealthOmics Les flux de travail utilisent ces spécifications de calcul définies pour exécuter votre tâche. Chaque tâche possède un identifiant unique.
- Groupe d'exécutions : groupe d'exécutions pour lequel vous pouvez définir le nombre maximal de vCPU, la durée maximale ou le nombre maximum d'exécutions simultanées afin de limiter les ressources de calcul utilisées par exécution. Vous pouvez spécifier et configurer les priorités de vos courses au sein d'un groupe d'exécutions. Par exemple, vous pouvez spécifier qu'une exécution de priorité élevée sera effectuée avant une exécution de priorité inférieure, créant ainsi une file d'attente prioritaire. L'utilisation d'un groupe d'exécution est facultative, et chaque groupe d'exécution possède un identifiant unique.

## Stockage

Le stockage des données est séparé en magasins de séquences, pour vos séquences génomiques et les informations connexes, et en un magasin de référence, pour tous vos génomes de référence. Les termes suivants décrivent les implémentations spécifiques à HealthOmics.

- Magasin de séquences : magasin de données pour le stockage de fichiers génomiques. Vous pouvez y inclure un ou plusieurs magasins de séquences HealthOmics. Les autorisations d'accès et le AWS KMS chiffrement peuvent être définis sur un magasin de séquences pour contrôler qui a accès aux données.
- Ensemble de lectures — Un jeu de lectures est une abstraction des lectures génomiques, qui sont stockées aux formats FASTQ, BAM ou CRAM. Les ensembles de lectures peuvent être importés dans des magasins de séquences et annotés avec des métadonnées. Vous pouvez appliquer des autorisations aux ensembles de lecture à l'aide du contrôle d'accès basé sur les attributs (ABAC).
- Référence — Une référence génomique est utilisée avec les lectures pour identifier à quel endroit d'un génome une lecture spécifique, ou un groupe de lectures, est mappé. Ils sont au format FASTA et stockés dans le magasin de référence.
- Magasin de référence — Un magasin de données pour le stockage des génomes de référence. Vous pouvez avoir un seul magasin de référence pour chaque compte et chaque région.

### Analyse

Vous pouvez transformer et analyser vos données génomiques avec HealthOmics Analytics. Créez un magasin de variantes ou un magasin d'annotations pour inclure des informations supplémentaires pour vos requêtes.

- Magasin de variantes : magasin de données qui stocke les données des variantes à l'échelle de la population. Les magasins de variants prennent en charge à la fois le format d'appel de variants génomique (GvCF) et les entrées VCF.
- Magasin d'annotations : magasin de données représentant une base de données d'annotations, telle qu'une base de données provenant d'un fichier TSV/CSV, VCF ou General Feature Format ()GFF3. Les magasins d'annotations sont mappés sur le même système de coordonnées que les magasins de variantes lors d'une importation.

## Services connexes

Les services suivants fonctionnent avec HealthOmics.

- Amazon Elastic Container Registry : chaque flux de travail privé utilise une image Amazon ECR (dans un référentiel Amazon ECR privé) pour contenir tous les exécutables, bibliothèques et scripts nécessaires à l'exécution du flux de travail.

- Amazon Simple Storage Service — Amazon S3 fournit un stockage de fichiers pour les données du magasin et du flux de travail.
- AWS Lake Formation — Lake Formation gère l'accès aux données de vos magasins de données Analytics.
- Amazon Athena — Utilisez Athena pour effectuer des requêtes sur vos boutiques Variant.
- Amazon SageMaker AI — Utilisez l' SageMaker IA pour exécuter des HealthOmics tâches à l'aide des blocs-notes Jupyter.
- GitHub connections— Utilisez des connexions pour connecter vos référentiels de code externes à vos HealthOmics flux de travail.

## Comment accéder HealthOmics

Vous pouvez accéder aux AWS HealthOmics fonctionnalités à l'aide de la console de gestion, de la CLI SDKs ou de l'API.

- AWS Console de gestion : fournit une interface Web à laquelle vous pouvez accéder HealthOmics.
- AWS Command Line Interface (AWS CLI) — Fournit des commandes pour un large éventail de AWS services, notamment AWS HealthOmics, et est compatible avec Windows, macOS et Linux. Pour plus d'informations sur l'installation du AWS CLI, consultez AWS Command Line Interface.
- AWS SDKs — AWS fournit SDKs (kits de développement logiciel) composés de bibliothèques et d'exemples de code pour différents langages de programmation et plateformes (notamment Java, Python, Ruby, .NET, iOS et Android). Ils SDKs fournissent un moyen pratique d'utiliser HealthOmics par programmation. Pour plus d'informations, consultez le AWS SDK Developer Center.
- AWS API — Vous pouvez utiliser les opérations d'API pour accéder et gérer HealthOmics par programmation. Pour plus d'informations, consultez la page Référence de l'API HealthOmics .

## Régions et points de terminaison pour AWS HealthOmics

Pour une liste complète des régions et des points de terminaison, consultez la référence AWS générale.

Outre les AWS régions actives par défaut, il existe également des régions optionnelles qui doivent être activées. Pour en savoir plus sur l'activation ou la désactivation d'une région, voir Spécifier AWS les régions que votre compte peut utiliser dans le guide de gestion de AWS compte.

# En savoir plus

Apprenez-en davantage HealthOmics grâce à ces ateliers et tutoriels :

- HealthOmics atelier — atelier de HealthOmics bout en bout
- AWS ressources génomiques — Référentiels publics Amazon ECR relatifs à la génomique
- Tutoriels Python — Tutoriels Jupyter Notebook sur le HealthOmics stockage GitHub, l'analyse et les flux de travail

Familiarisez-vous avec HealthOmics les outils supplémentaires qui AWS fournissent :

- Linter WDL — HealthOmics Linter pour WDL
- Nextflow linter — HealthOmics Linter pour Nextflow
- HealthOmics Outil d'assistance Amazon ECR — Outil d'assistance Amazon ECR pour HealthOmics
- HealthOmics tools on GitHub — Outils pour travailler avec HealthOmics (gestionnaire de transfert, analyseur d'URI, réexécution d'Omics, analyseur d'exécution).

# AWS HealthOmics modification de la disponibilité du magasin de variantes et du magasin d'annotations

Après mûre réflexion, nous avons décidé de fermer les magasins de AWS HealthOmics variantes et les magasins d'annotations aux nouveaux clients à compter du 7 novembre 2025. Les clients existants peuvent continuer à utiliser le service normalement.

La section suivante décrit les options de migration qui vous aideront à déplacer vos magasins de variantes et vos magasins d'analyse vers de nouvelles solutions. Pour toute question ou préoccupation, créez un dossier d'assistance sur support.console.aws.amazon.com.

Rubriques



## Vue d'ensemble des options de migration

Les options de migration suivantes constituent une alternative à l'utilisation de magasins de variantes et de magasins d'annotations :

1. Utilisez l'implémentation HealthOmics de référence de la logique ETL fournie.

   Utilisez des compartiments de table S3 pour le stockage et continuez à utiliser les services AWS d'analyse existants.
2. Créez une solution à l'aide d'une combinaison de AWS services existants.

   Pour l'ETL, vous pouvez écrire des tâches Glue ETL personnalisées ou utiliser le code open source HAIL ou GLOW sur EMR pour transformer les données des variantes.

   Utilisez des compartiments de table S3 pour le stockage et continuez à utiliser les services d' AWS analyse existants

3. Sélectionnez un AWS partenaire proposant une alternative aux variantes et au magasin d'annotations.

## Options de migration pour la logique ETL

Envisagez les options de migration suivantes pour la logique ETL :

1. HealthOmics fournit la logique ETL du magasin de variantes actuel en tant que HealthOmics flux de travail de référence. Vous pouvez utiliser le moteur de ce flux de travail pour exécuter exactement le même processus ETL de données de variantes que le magasin de variantes, mais en contrôlant totalement la logique ETL.

   Ce flux de travail de référence est disponible sur demande. Pour demander l'accès, créez un dossier d'assistance sur support.console.aws.amazon.com.
2. Pour transformer les données des variantes, vous pouvez écrire des tâches Glue ETL personnalisées ou utiliser du code open source HAIL ou GLOW sur EMR.

## Options de migration pour le stockage

En remplacement du magasin de données hébergé par un service, vous pouvez utiliser des compartiments de table Amazon S3 pour définir un schéma de table personnalisé. Pour plus d'informations sur les compartiments de table, consultez les compartiments de table dans le guide de l'utilisateur d'Amazon S3.

Vous pouvez utiliser des compartiments de table pour les tables Iceberg entièrement gérées dans Amazon S3.

Vous pouvez présenter un dossier d'assistance pour demander à l' HealthOmics équipe de migrer les données de votre magasin de variantes ou d'annotations vers le compartiment de table Amazon S3 que vous avez configuré.

Une fois que vos données sont renseignées dans le compartiment de table Amazon S3, vous pouvez supprimer vos magasins de variantes et vos magasins d'annotations. Pour plus d'informations, consultez la section Suppression de magasins HealthOmics d'analyses.

# Analyse

Pour l'analyse des données, continuez à utiliser des services AWS d'analyse tels qu'Amazon Athena, Amazon EMR, AmazonRedshift ou Amazon Quick Suite.

# AWS Les partenaires

Vous pouvez travailler avec un AWS partenaire qui fournit un ETL personnalisable, des schémas de table, des outils de requête et d'analyse intégrés, ainsi que des interfaces utilisateur pour interagir avec les données.

# Exemples

Les exemples suivants montrent comment créer des tables adaptées au stockage de données VCF et GVCF.

## Athéna DDL

Vous pouvez utiliser l'exemple DDL suivant dans Athena pour créer une table adaptée au stockage des données VCF et GVCF dans une seule table. Cet exemple n'est pas l'équivalent exact de la structure de magasin variant, mais il fonctionne bien pour un cas d'utilisation générique.

Créez vos propres valeurs pour DATABASE_NAME et TABLE_NAME lorsque vous créez la table.

```
 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'
);
```

## Création de tables en Python (sans Athena)

L'exemple de code Python suivant montre comment créer les tables sans utiliser Athena.

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


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

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

    return load_catalog("s3tables", **catalog_config)


def create_namespace(catalog: Catalog, namespace: str) -> None:
```

```
    try:
        catalog.create_namespace(namespace)
        print(f"Created namespace: {namespace}")
    except Exception as e:
        if "already exists" in str(e):
            print(f"Namespace {namespace} already exists.")
        else:
            raise e


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

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

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

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


def main(bucket_arn: str, namespace: str, table_name: str):
    # Schema definition
    genomic_variants_schema = Schema(
        NestedField(1, "sample_name", StringType(), required=True),
        NestedField(2, "variant_name", StringType(), required=True),
        NestedField(3, "chrom", StringType(), required=True),
        NestedField(4, "pos", LongType(), required=True),
        NestedField(5, "ref", StringType(), required=True),
        NestedField(6, "alt", ListType(element_id=1000, element_type=StringType(),
 element_required=True), required=True),
        NestedField(7, "qual", DoubleType()),
```

```
        NestedField(8, "filter", StringType()),
        NestedField(9, "genotype", StringType()),
        NestedField(10, "info", MapType(key_type=StringType(), key_id=1001,
 value_type=StringType(), value_id=1002)),
        NestedField(11, "attributes", MapType(key_type=StringType(), key_id=2001,
 value_type=StringType(), value_id=2002)),
        NestedField(12, "is_reference_block", BooleanType()),
        identifier_field_ids=[1, 2, 3, 4]
    )

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

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

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

    return table


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

    main(bucket_arn, namespace, table_name)
```

# Con HealthOmicsfiguration

Pour le configurer AWS HealthOmics, inscrivez-vous à un Compte AWS, créez un utilisateur administratif et gérez en toute sécurité l'accès pour d'autres utilisateurs.

Rubriques



## Inscrivez-vous pour un Compte AWS

Si vous n'en avez pas Compte AWS, procédez comme suit pour en créer un.

Pour vous inscrire à un Compte AWS

1. Ouvrez l'https://portal.aws.amazon.com/billing/inscription.
2. Suivez les instructions en ligne.

   Dans le cadre de la procédure d'inscription, vous recevrez un appel téléphonique ou un SMS et vous saisirez un code de vérification en utilisant le clavier numérique du téléphone.

   Lorsque vous vous inscrivez à un Compte AWS, un Utilisateur racine d'un compte AWSest créé. Par défaut, seul l'utilisateur racine a accès à l'ensemble des Services AWS et des ressources de ce compte. La meilleure pratique de sécurité consiste à attribuer un accès administratif à un utilisateur, et à utiliser uniquement l'utilisateur racine pour effectuer les tâches nécessitant un accès utilisateur racine.

AWS vous envoie un e-mail de confirmation une fois le processus d'inscription terminé. À tout moment, vous pouvez consulter l'activité actuelle de votre compte et gérer votre compte en accédant à https://aws.amazon.com/et en choisissant Mon compte.

# Création d'un utilisateur doté d'un accès administratif

Une fois que vous vous êtes inscrit à un utilisateur administratif Compte AWS, que vous Utilisateur racine d'un compte AWS l'avez sécurisé AWS IAM Identity Center, que vous l'avez activé et que vous en avez créé un, afin de ne pas utiliser l'utilisateur root pour les tâches quotidiennes.

Sécurisez votre Utilisateur racine d'un compte AWS

1. Connectez-vous en AWS Management Consoletant que propriétaire du compte en choisissant Utilisateur root et en saisissant votre adresse Compte AWS e-mail. Sur la page suivante, saisissez votre mot de passe.

   Pour obtenir de l'aide pour vous connecter en utilisant l'utilisateur racine, consultez Connexion en tant qu'utilisateur racine dans le Guide de l'utilisateur Connexion à AWS .
2. Activez l'authentification multifactorielle (MFA) pour votre utilisateur racine.

   Pour obtenir des instructions, consultez la section Activer un périphérique MFA virtuel pour votre utilisateur Compte AWS root (console) dans le guide de l'utilisateur IAM.

Création d'un utilisateur doté d'un accès administratif

1. Activez IAM Identity Center.

   Pour obtenir des instructions, consultez Activation d' AWS IAM Identity Center dans le Guide de l'utilisateur AWS IAM Identity Center .
2. Dans IAM Identity Center, octroyez un accès administratif à un utilisateur.

   Pour un didacticiel sur l'utilisation du Répertoire IAM Identity Center comme source d'identité, voir Configurer l'accès utilisateur par défaut Répertoire IAM Identity Center dans le Guide de AWS IAM Identity Center l'utilisateur.

Connexion en tant qu'utilisateur doté d'un accès administratif

- Pour vous connecter avec votre utilisateur IAM Identity Center, utilisez l'URL de connexion qui a été envoyée à votre adresse e-mail lorsque vous avez créé l'utilisateur IAM Identity Center.

Pour obtenir de l'aide pour vous connecter en utilisant un utilisateur d'IAM Identity Center, consultez la section Connexion au portail AWS d'accès dans le guide de l'Connexion à AWS utilisateur.

Attribution d'un accès à d'autres utilisateurs

1. Dans IAM Identity Center, créez un ensemble d'autorisations qui respecte la bonne pratique consistant à appliquer les autorisations de moindre privilège.

   Pour obtenir des instructions, consultez Création d'un ensemble d'autorisations dans le Guide de l'utilisateur AWS IAM Identity Center .
2. Attribuez des utilisateurs à un groupe, puis attribuez un accès par authentification unique au groupe.

   Pour obtenir des instructions, consultez Ajout de groupes dans le Guide de l'utilisateur AWS IAM Identity Center .

## Créez des autorisations IAM pour HealthOmics

Pour les utiliser HealthOmics, configurez les autorisations IAM suivantes :

- Politiques basées sur l'identité IAM auxquelles les utilisateurs de votre compte peuvent accéder. HealthOmics
- Rôle de service IAM permettant HealthOmics d'accéder aux ressources en votre nom.
- Autorisations accordées à vos utilisateurs et au HealthOmics service pour accéder aux ressources dans d'autres services (tels que Lake Formation et Amazon ECR).

Pour plus d'informations sur la configuration des autorisations IAM pour HealthOmics, consultezAutorisations IAM pour HealthOmics.

## Connectez-vous à des référentiels de code externes

Avec AWS HealthOmics, vous pouvez gérer vos flux de travail à l'aide de référentiels basés sur Git via. AWS CodeConnections HealthOmics utilise cette connexion pour accéder à vos référentiels de code source.

Avant d'utiliser des référentiels de code externes, suivez le guide de configuration des connexions pour commencer à travailler avec AWS CodeConnections. Vérifiez que vous avez créé les politiques et autorisations IAM appropriées pour votre AWS compte. Pour obtenir la liste des fournisseurs Git pris en charge et plus d'informations, voir Pour quels fournisseurs tiers puis-je créer des connexions ? .

Création d'une connexion

Pour créer une connexion avec votre fournisseur de dépôt préféré, suivez le didacticiel Créer une connexion.

# Utilisation de la CLI Amazon Q avec HealthOmics

Amazon Q CLI fournit des interactions en langage naturel avec AWS HealthOmics, ce qui vous permet d'effectuer des flux de travail génomiques complexes et des tâches d'analyse à l'aide de commandes conversationnelles. Pour utiliser l'interface de ligne de commande Amazon Q, veillez à configurer les autorisations IAM HealthOmics et les autres services (tels qu' CloudWatchAmazon ECR ou Amazon S3) permettant à Amazon Q d'accéder à ses ressources.

Le didacticiel HealthOmics Agentic Generative AI fournit des step-by-step conseils pour configurer les fichiers de contexte et permettre à Amazon Q CLI de créer, d'exécuter et d'optimiser vos AWS HealthOmics flux de travail.

# Commencer avec HealthOmics

Pour commencer HealthOmics, assurez-vous d'avoir correctement configuré vos autorisations et rôles IAM pour HealthOmics.

## Utilisation d'un flux de travail Ready2Run dans la console HealthOmics

L'exercice suivant montre comment utiliser un flux de travail Ready2Run. Un flux de travail Ready2Run est préconfiguré avec les paramètres et les références d'outils dont vous avez besoin pour exécuter le flux de travail. L'éditeur de flux de travail fournit des exemples de données, vous n'avez donc pas besoin de créer vos propres données.

1. Ouvrez la HealthOmics console.
2. Sélectionnez le volet de navigation (≡) en haut à gauche, puis sélectionnez les flux de travail Ready2Run.
3. Sur la page des flux de travail Ready2Run, choisissez le ESMFold for up to 800 residues flux de travail. La console ouvre la page de détails de ce flux de travail.
4. L'onglet Détails fournit des informations sur le flux de travail. Pour tester le flux de travail, en haut à droite de la page, sélectionnez Démarrer l'exécution.
5. Dans la page Spécifier les détails de l'exécution, entrez un nom d'exécution.
6. Entrez ou sélectionnez un emplacement Amazon S3 pour la sortie d'exécution.
7. Pour le mode Exécuter la conservation des métadonnées, choisissez de conserver ou de supprimer les données runmeta.
8. Dans le panneau Rôle de service, choisissez Créer et utiliser un nouveau rôle de service.
9. Choisissez Suivant.
10. Sur la page Ajouter des valeurs de paramètres, choisissez Exécuter le flux de travail avec les données de test Ready2Run.
11. Choisissez Suivant.
12. Passez en revue vos entrées, puis choisissez Démarrer l'exécution.

# Exemples d'instructions pour Amazon Q CLI

La CLI Amazon Q peut exécuter des flux de travail génomiques et des tâches d'analyse à AWS HealthOmics l'aide de commandes en langage naturel. Les exemples d'invite suivants vous permettent de créer des flux de travail, de gérer des séries et d'analyser des données génomiques. Pour plus d'informations et des exemples d'instructions HealthOmics, consultez le didacticiel HealthOmics Agentic Generative AI sur. GitHub

- « Créez un fichier de flux de travail WDL 1.1 tel `main.wdl` qu'il sera exécuté. HealthOmics Le flux de travail prendra un génome de référence en entrée et des paires de fichiers fastq. Il indexera le génome de référence à l'aide de BWA, puis mappera chaque paire de fichiers fastq à la référence. Enfin, fusionnez chaque BAM mappé en un seul fichier BAM et produisez ce fichier et son index bai. »
- « Package du flux de travail et créez-le dedans HealthOmics »
- « Mettez à jour le fichier inputs.json pour utiliser les fichiers réels de mon compartiment Amazon S3 `omics-my-bucket-with-genome-data` » (indiquez un emplacement de compartiment Amazon S3 spécifique ou laissez Amazon Q explorer)
- « Trouvez des conteneurs appropriés dans mes référentiels Amazon ECR et mettez à jour inputs.json pour les utiliser »
- « Trouvez ou créez un rôle IAM approprié à utiliser lors de l'exécution du flux de travail »
- « Créer un cache d'exécution pour mon flux de travail »
- « Exécuter le flux de travail dans HealthOmics »
- « Vérifiez l'état de la course »

> **Warning**
>
> Lorsque vous travaillez avec Amazon Q CLI, passez en revue tout le contenu généré et les actions proposées avant de continuer. Fournissez des commentaires pour améliorer la qualité des réponses et répondre aux exigences de votre flux de travail. Pour plus d'informations, consultez Considérations relatives à la sécurité et bonnes pratiques pour Amazon Q.

# Workflows privés dans HealthOmics

Utilisez des flux de travail privés lorsque vous souhaitez créer votre propre définition de flux de travail. La définition du flux de travail fournit des informations sur le flux de travail et définit les tâches du flux de travail. Une exécution est une invocation unique d'un flux de travail, et une tâche est un processus unique au sein de l'exécution.

HealthOmics prend en charge les définitions de flux de travail que vous créez dans le langage WDL (Workflow Description Language), le langage CWL (Common Workflow Language) ou Nextflow.

HealthOmics les flux de travail fournissent les fonctionnalités optionnelles suivantes :

- Run groups— Vous pouvez ajouter des flux de travail privés à un groupe d'exécution pour contrôler l'utilisation du calcul. Un groupe d'exécution est un ensemble d'exécutions de flux de travail partageant un ensemble de limites de ressources, telles que le nombre maximum d'exécutions simultanées et la durée d'exécution maximale. Vous définissez ces limites pour contrôler les ressources de calcul consommées par le groupe d'exécution.
- Call caching— Vous pouvez utiliser un cache d'appels pour enregistrer et réutiliser les résultats des tâches, ce qui permet de réduire les durées d'exécution et de réduire les coûts de calcul.
- Sharing workflows— Vous pouvez partager vos flux de travail privés avec d'autres Comptes AWS personnes de la même région.
- Workflow versions— Vous pouvez créer des versions d'un flux de travail privé. La gestion des versions des flux de travail permet aux utilisateurs de choisir à quel moment ils souhaitent commencer à utiliser les fonctionnalités mises à jour. Les versions de flux de travail sont immuables et fournissent le même niveau de provenance des données que les flux de travail.

Pour plus d'informations sur la configuration des autorisations IAM pour les flux de travail, consultezAutorisations IAM pour HealthOmics.

Pour des exemples complets d'utilisation de flux de travail HealthOmics privés, consultez les didacticiels HealthOmics Github ou le didacticiel de bout en bout de l'atelier AWS pour HealthOmics.

Rubriques

- Création de flux de travail privés dans HealthOmics
- Versionnage des flux de travail dans HealthOmics
- Utiliser des HealthOmics courses

- Utilisation de groupes d' HealthOmics exécution
- Mise en cache des appels pour les exécutions HealthOmics
- Partage HealthOmics de workflows

# Création de flux de travail privés dans HealthOmics

Les flux de travail privés dépendent de diverses ressources que vous créez et configurez avant de créer le flux de travail :

- Workflow definition file:Un fichier de définition de flux de travail écrit en WDLNextflow, ouCWL. La définition du flux de travail spécifie les entrées et les sorties pour les exécutions qui utilisent le flux de travail. Il inclut également des spécifications pour les exécutions et les tâches d'exécution pour votre flux de travail, y compris les exigences en matière de calcul et de mémoire. Le fichier de définition du flux de travail doit être au `.zip` format. Pour plus d'informations, consultez la section Fichiers de définition du flux de travail.
  - Vous pouvez utiliser Amazon Q CLI pour créer et valider vos fichiers de définition de flux de travail dans WDL, Nextflow et CWL. Pour plus d'informations, consultez les exemples d'instructions pour Amazon Q CLI et le didacticiel HealthOmics Agentic Generative AI sur. GitHub
- (Optional) Parameter template file:Un fichier de modèle de paramètres écrit enJSON. Créez le fichier pour définir les paramètres d'exécution ou HealthOmics générez le modèle de paramètres pour vous. Pour plus d'informations, consultez la section Fichiers modèles de paramètres pour les HealthOmics flux de travail.
- Amazon ECR container images:Créez un référentiel Amazon ECR privé pour le flux de travail. Créez des images de conteneur dans le référentiel privé ou synchronisez le contenu d'un registre en amont pris en charge avec votre référentiel privé Amazon ECR.
- (Optional) Sentieon licenses:Demandez une Sentieon licence pour utiliser le Sentieon logiciel dans des flux de travail privés.

Vous pouvez éventuellement exécuter un linter sur la définition du flux de travail avant ou après la création du flux de travail. La linter rubrique décrit les linters disponibles dans. HealthOmics

Rubriques

- HealthOmics intégration du flux de travail avec les référentiels basés sur Git
- Fichiers de définition du flux de travail dans HealthOmics

- Fichiers modèles de paramètres pour les HealthOmics flux de travail
- Images de conteneur pour les flux de travail privés
- HealthOmics Fichiers README du flux de travail
- Demande de licences Sentieon pour des flux de travail privés
- Linters de flux de travail dans HealthOmics
- HealthOmics opérations de flux de travail

## HealthOmics intégration du flux de travail avec les référentiels basés sur Git

Lorsque vous créez un flux de travail (ou une version de flux de travail), vous fournissez une définition de flux de travail pour spécifier les informations relatives au flux de travail, aux exécutions et aux tâches. HealthOmics peut récupérer la définition du flux de travail sous forme d'archive .zip (stockée localement ou dans un compartiment Amazon S3) ou à partir d'un référentiel Git compatible.

L' HealthOmics intégration avec les référentiels basés sur Git permet les fonctionnalités suivantes :

- Création directe de flux de travail à partir d'instances publiques, privées et autogérées.
- Intégration de fichiers README de flux de travail et de modèles de paramètres à partir de référentiels.
- Support pour GitHub GitLab, et les référentiels Bitbucket.

En utilisant un référentiel basé sur Git, vous évitez les étapes manuelles consistant à télécharger des fichiers de définition de flux de travail et des fichiers de modèles de paramètres d'entrée, à créer une archive .zip, puis à transférer l'archive vers S3. Cela simplifie la création de flux de travail pour des scénarios tels que les exemples suivants :

1. Vous souhaitez commencer rapidement à utiliser un flux de travail open source courant, tel que nf-core. HealthOmicsrécupère automatiquement tous les fichiers de définition de flux de travail et de modèles de paramètres d'entrée à partir du référentiel nf-core GitHub et utilise ces fichiers pour créer votre nouveau flux de travail.
2. Vous utilisez un flux de travail public depuis GitHub, et de nouvelles mises à jour sont disponibles. Vous pouvez facilement créer une nouvelle version de HealthOmics flux de travail en utilisant la définition de flux de travail mise à jour GitHub comme source. Les utilisateurs de votre flux de travail peuvent choisir entre le flux de travail d'origine ou la nouvelle version de flux de travail que vous avez créée.

3. Votre équipe est en train de créer un pipeline propriétaire qui n'est pas public. Vous conservez votre code dans un dépôt git privé et vous utilisez cette définition de flux de travail pour vos HealthOmics flux de travail. L'équipe met fréquemment à jour la définition du flux de travail dans le cadre d'un cycle de développement itératif du flux de travail. Vous pouvez facilement créer de nouvelles versions de flux de travail selon les besoins à partir de votre référentiel privé.

Rubriques



## Référentiels basés sur Git pris en charge

HealthOmics prend en charge les référentiels publics et privés pour les fournisseurs Git suivants :

- GitHub
- GitLab
- Bitbucket

HealthOmics prend en charge les référentiels autogérés pour les fournisseurs Git suivants :

- GitHubEnterpriseServer
- GitLabSelfManaged

HealthOmics prend en charge l'utilisation de connexions entre comptes pour GitHub GitLab, et Bitbucket. Configurez des autorisations partagées via AWS Resource Access Manager. Pour un exemple, voir Connexions partagées dans le guide de CodePipeline l'utilisateur.

## Configuration des connexions aux référentiels de code externes

Connectez vos flux de travail à des référentiels basés sur Git à l'aide d'AWS. CodeConnection HealthOmics utilise cette connexion pour accéder à vos référentiels de code source.

> **Note**
>
> Le CodeConnections service AWS n'est pas disponible dans la région il-central-1. Pour cette région, configurez le service us-east-1 pour créer des flux de travail ou des versions de flux de travail à partir d'un référentiel.

Créez une connexion

Avant de créer des connexions, suivez les instructions de la section Configuration des connexions du Guide de l'utilisateur des outils de la Developer Console.

Pour créer une connexion, suivez les instructions de la section Créer une connexion du Guide de l'utilisateur des outils de la Developer Console.

Configurer l'autorisation pour la connexion

Vous devez autoriser la connexion en utilisant le OAuth flux du fournisseur. Assurez-vous que l'état de la connexion est `AVAILABLE` correct avant de l'utiliser.

Pour des exemples, consultez le billet de blog Comment créer un AWS HealthOmics flux de travail à partir du contenu dans Git.

## Accès aux référentiels autogérés

Pour configurer des connexions à un référentiel GitLab autogéré, utilisez un jeton d'accès personnel d'administrateur lors de la création d'un hôte. La création de connexion suivante permet d'accéder à Oauth avec le compte du client.

L'exemple suivant établit une connexion à un référentiel GitLab autogéré :

1. Configurez l'accès au jeton d'accès personnel d'un utilisateur administrateur.

   Pour configurer un PAT dans un référentiel GitLab autogéré, voir Jetons d'accès personnels dans GitLab Docs.
2. Création d'un hôte

   a. Accédez à CodePipeline>Paramètres>Connexions.

   b. Choisissez l'onglet Hosts, puis Create Host.

   c. Configurez les champs suivants :

- Entrez le nom de l'hôte
- Pour le type de fournisseur, choisissez GitLab Self Managed
- Entrez l'URL de l'hôte
- Entrez les informations du VPC si l'hôte est défini dans un VPC

d. Choisissez Create Host, ce qui crée l'hôte dans l'état PENDING.

e. Pour terminer la configuration, choisissez Set up Host.

f. Entrez le jeton d'accès personnel (PAT) d'un utilisateur administrateur, puis choisissez Continuer.

3. Créez la connexion

a. Choisissez Créer des connexions dans l'onglet Connexions.

b. Pour le type de fournisseur, sélectionnez GitLab Autogéré.

c. Sous Paramètres de connexion > Entrez le nom de la connexion, entrez l'URL de l'hôte que vous avez créée précédemment.

d. Si votre instance GitLab autogérée est uniquement accessible via un VPC, configurez les détails du VPC.

e. Choisissez Mettre à jour la connexion en attente. La fenêtre modale vous redirige vers la page de GitLab connexion.

f. Entrez le nom d'utilisateur et le mot de passe du compte client et terminez le processus d'autorisation.

g. Pour la première configuration, choisissez Authorize AWS Connector for Gitlab Self Managed.

## Quotas liés aux référentiels de code externes

Pour HealthOmics l'intégration avec des référentiels de code externes, il existe une taille maximale pour un référentiel, chaque fichier de référentiel et chaque fichier README. Pour en savoir plus, consultez HealthOmics quotas de taille fixe du flux de travail.

## Autorisations IAM requises

Ajoutez les actions suivantes à votre politique IAM basée sur l'identité :

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

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

## Fichiers de définition du flux de travail dans HealthOmics

Vous utilisez une définition de flux de travail pour spécifier des informations sur le flux de travail, les exécutions et les tâches des exécutions. Vous créez des définitions de flux de travail dans un ou plusieurs fichiers à l'aide d'un langage de définition de flux de travail. HealthOmics prend en charge les définitions de flux de travail écrites en WDL, Nextflow ou CWL.

HealthOmics prend en charge les choix suivants pour les définitions de flux de travail WDL :

- WDL — Fournit un moteur WDL conforme aux spécifications.
- WDL lenient — Conçu pour gérer les flux de travail migrés depuis Cromwell. Il prend en charge les directives Cromwell du client et certaines logiques non conformes. Pour en savoir plus, consultez Conversion de type implicite dans WDL Lenient.

Pour plus d'informations sur chacune des langues du flux de travail, consultez les sections détaillées spécifiques à chaque langue ci-dessous.

Vous spécifiez les types d'informations suivants dans la définition du flux de travail :

- Language version— La langue et la version de la définition du flux de travail.
- Compute and memory— Les besoins en calcul et en mémoire pour les tâches du flux de travail.
- Inputs— Emplacement des entrées pour les tâches du flux de travail. Pour de plus amples informations, veuillez consulter HealthOmics exécuter les entrées.
- Outputs— Emplacement pour enregistrer les résultats générés par les tâches.
- Task resources— Besoins en calcul et en mémoire pour chaque tâche.
- Accelerators— les autres ressources requises par les tâches, telles que les accélérateurs.

Rubriques

- Ressources de tâches dans une définition HealthOmics de flux de travail
- Accélérateurs de tâches dans une définition de HealthOmics flux de travail
- Spécificités de définition du flux de travail WDL
- Caractéristiques de la définition du flux de travail Nextflow
- Spécificités de définition du flux de travail CWL
- Exemples de définitions de flux de travail

## HealthOmics exigences de définition du flux de travail

Les fichiers HealthOmics de définition du flux de travail doivent répondre aux exigences suivantes :

- Les tâches doivent définir input/output les paramètres, les référentiels de conteneurs Amazon ECR et les spécifications d'exécution telles que l'allocation de mémoire ou de CPU.
- Vérifiez que vos rôles IAM disposent des autorisations requises.
  - Votre flux de travail a accès aux données d'entrée provenant de AWS ressources, telles qu'Amazon S3.
  - Votre flux de travail a accès à des services de référentiel externes en cas de besoin.
- Déclarez les fichiers de sortie dans la définition du flux de travail. Pour copier des fichiers d'exécution intermédiaires vers l'emplacement de sortie, déclarez-les en tant que sorties de flux de travail.
- Les emplacements d'entrée et de sortie doivent se trouver dans la même région que le flux de travail.
- HealthOmics les entrées du flux de travail de stockage doivent être en `ACTIVE` état. HealthOmics n'importera pas les entrées avec un `ARCHIVED` statut, ce qui entraînera l'échec du flux de travail. Pour plus d'informations sur les entrées d'objets Amazon S3, consultezHealthOmics exécuter les entrées.
- L'mainemplacement du flux de travail est facultatif si votre archive ZIP contient une seule définition de flux de travail ou un fichier nommé « principal ».
  - Exemple de chemin : `workflow-definition/main-file.wdl`
- Avant de créer un flux de travail à partir d'Amazon S3 ou de votre disque local, créez une archive zip contenant les fichiers de définition du flux de travail et toutes les dépendances, telles que les sous-flux de travail.
- Nous vous recommandons de déclarer les conteneurs Amazon ECR dans le flux de travail comme paramètres d'entrée pour la validation des autorisations Amazon ECR.

Autres considérations relatives à Nextflow :

- /bin

  Les définitions de flux de travail Nextflow peuvent inclure un dossier /bin contenant des scripts exécutables. Ce chemin dispose d'un accès en lecture seule et d'un accès exécutable aux tâches. Les tâches qui s'appuient sur ces scripts doivent utiliser un conteneur créé avec les interpréteurs de script appropriés. La meilleure pratique consiste à appeler directement l'interprète. Par exemple :

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

- includeConfig

  Les définitions de flux de travail basées sur Nextflow peuvent inclure des fichiers nextflow.config qui permettent d'abstraire les définitions de paramètres ou de traiter les profils de ressources. Pour prendre en charge le développement et l'exécution de pipelines Nextflow sur plusieurs environnements, utilisez une configuration HealthOmics spécifique que vous ajoutez à la configuration globale à l'aide de la directive IncludeConfig. Pour garantir la portabilité, configurez le flux de travail pour inclure le fichier uniquement lors de son exécution en HealthOmics utilisant le code suivant :

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

- Reports

  HealthOmics ne prend pas en charge les rapports DAG, de trace et d'exécution générés par le moteur. Vous pouvez générer des alternatives aux rapports de suivi et d'exécution à l'aide d'une combinaison GetRun d'appels d' GetRunTask API.

Considérations supplémentaires sur le CWL :

- Container image uri interpolation

  HealthOmics permet à la propriété DockerPull du d' DockerRequirement être une expression javascript en ligne. Par exemple :

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

  Cela vous permet de spécifier l'image du conteneur URIs comme paramètre d'entrée du flux de travail.

- Javascript expressions

  Les expressions Javascript doivent être `strict mode` conformes.

- Operation process

  HealthOmics ne prend pas en charge les processus d'opération CWL.

## Support de version pour les langages HealthOmics de définition des flux de travail

HealthOmics prend en charge les fichiers de définition de flux de travail écrits en Nextflow, WDL ou CWL. Les sections suivantes fournissent des informations sur HealthOmics la prise en charge des versions pour ces langues.

Rubriques



### Support des versions WDL

HealthOmics prend en charge les versions 1.0, 1.1 et la version de développement de la spécification WDL.

Chaque document WDL doit inclure une déclaration de version pour spécifier la version (majeure et mineure) de la spécification à laquelle il adhère. Pour plus d'informations sur les versions, voir Gestion des versions WDL

Les versions 1.0 et 1.1 de la spécification WDL ne prennent pas en charge ce `Directory` type. Pour utiliser le `Directory` type pour les entrées ou les sorties, définissez la development version sur la première ligne du fichier :

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

Support de la version CWL

HealthOmics supporte les versions 1.0, 1.1 et 1.2 du langage CWL.

Vous pouvez spécifier la version linguistique dans le fichier de définition du flux de travail CWL. Pour plus d'informations sur CWL, consultez le guide de l'utilisateur de CWL

Support de la version Nextflow

HealthOmics supporte trois versions stables de Nextflow. Nextflow publie généralement une version stable tous les six mois. HealthOmics ne prend pas en charge les versions mensuelles « Edge ».

HealthOmics prend en charge les fonctionnalités publiées dans chaque version, mais pas les fonctionnalités de prévisualisation.

Versions prises en charge

HealthOmics prend en charge les versions suivantes de Nextflow :

- Nextflow v22.04.01 DSL 1 et DSL 2
- Nextflow v23.10.0 DSL 2 (par défaut)
- Nextflow v24.10.8 DSL 2

Pour migrer votre flux de travail vers la dernière version prise en charge (v24.10.8), suivez le guide de mise à niveau de Nextflow.

La migration de Nextflow v23 vers v24 entraîne des modifications majeures, comme décrit dans les sections suivantes du guide de migration de Nextflow :

- Changements les plus marquants de la version 24.04

- Changements majeurs dans la version 24.10

### Détecter et traiter les versions de Nextflow

HealthOmics détecte la version DSL et la version Nextflow que vous spécifiez. Il détermine automatiquement la meilleure version de Nextflow à exécuter en fonction de ces entrées.

#### Version DSL

HealthOmics détecte la version DSL demandée dans votre fichier de définition de flux de travail. Par exemple, vous pouvez spécifier :`nextflow.enable.dsl=2`.

HealthOmics prend en charge le DSL 2 par défaut. Il fournit une rétrocompatibilité avec DSL 1, si cela est spécifié dans le fichier de définition de votre flux de travail.

- Si vous spécifiez DSL 2, HealthOmics exécute Nextflow v23.10.0, sauf si vous spécifiez Nextflow v22.04.0 ou v24.10.8.
- Si vous spécifiez DSL 1, HealthOmics exécute Nextflow v22.04 DSL1 (la seule version prise en charge qui exécute DSL 1).
- Si vous ne spécifiez pas de version DSL, ou si HealthOmics vous ne parvenez pas à analyser les informations DSL pour quelque raison que ce soit (par exemple, des erreurs de syntaxe dans le fichier de définition de votre flux de travail), utilisez HealthOmics par défaut DSL 2 et exécutez Nextflow v23.10.0.
- Pour mettre à niveau votre flux de travail du DSL 1 au DSL 2 afin de tirer parti des dernières versions et fonctionnalités logicielles de Nextflow, voir Migration depuis DSL 1.

#### Versions de Nextflow

HealthOmics détecte la version de Nextflow demandée dans le fichier de configuration de Nextflow (nextflow.config), si vous fournissez ce fichier. Nous vous recommandons d'ajouter la `nextflowVersion` clause à la fin du fichier pour éviter tout remplacement inattendu des configurations incluses. Pour plus d'informations, consultez la section Configuration de Nextflow.

Vous pouvez spécifier une version de Nextflow ou une série de versions à l'aide de la syntaxe suivante :

```
// 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 traite les informations de version de Nextflow comme suit :

- Si vous spécifiez une version exacte qui HealthOmics prend en charge, HealthOmics utilise cette version. =
- Si vous ! spécifiez une version exacte ou une plage de versions qui ne sont pas prises en charge, HealthOmics déclenche une exception et échoue. Envisagez d'utiliser cette option si vous souhaitez être strict en ce qui concerne les demandes de version et échouer rapidement si la demande inclut des versions non prises en charge.
- Si vous spécifiez une plage de versions, HealthOmics utilise la dernière version prise en charge dans cette plage, sauf si la plage inclut la version 24.10.8. Dans ce cas, HealthOmics donne la préférence à une version antérieure. Par exemple, si la plage couvre à la fois les versions 23.10.0 et 24.10.8, choisissez la version 23.10.0. HealthOmics
- S'il n'existe aucune version demandée, ou si les versions demandées ne sont pas valides ou ne peuvent pas être analysées pour une quelconque raison :
  - Si vous avez spécifié DSL 1, HealthOmics exécute Nextflow v22.04.
  - Sinon, HealthOmics exécute Nextflow v23.10.0.

Vous pouvez récupérer les informations suivantes concernant la version de Nextflow HealthOmics utilisée pour chaque exécution :

- Les journaux d'exécution contiennent des informations sur la version réelle de Nextflow HealthOmics utilisée pour l'exécution.
- HealthOmics ajoute des avertissements dans les journaux d'exécution s'il n'y a pas de correspondance directe avec la version demandée ou s'il est nécessaire d'utiliser une version différente de celle que vous avez spécifiée.

- La réponse à l'opération d'GetRunAPI inclut un champ (`engineVersion`) avec la version réelle de Nextflow HealthOmics utilisée pour l'exécution. Par exemple :

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

## Exigences en matière de calcul et de mémoire pour les HealthOmics tâches

HealthOmics exécute vos tâches de flux de travail privées dans une instance omics. HealthOmics fournit une variété de types d'instances pour s'adapter à différents types de tâches. Chaque type d'instance possède une configuration de mémoire et de vCPU fixes (et une configuration GPU fixe pour les types d'instances de calcul accéléré). Le coût d'utilisation d'une instance omics varie en fonction du type d'instance. Pour plus de détails, consultez la page de HealthOmics tarification.

Pour les tâches d'un flux de travail, vous spécifiez la mémoire requise et la valeur v CPUs dans le fichier de définition du flux de travail. Lorsqu'une tâche de flux de travail s'exécute, HealthOmics alloue la plus petite instance omics pouvant accueillir la mémoire demandée et v. CPUs Par exemple, si une tâche nécessite 64 GiB de mémoire et 8 VCPUs, HealthOmics sélectionne. `omics.r.2xlarge`

Nous vous recommandons de passer en revue les types d'instances et de définir le v CPUs et la taille de mémoire demandés pour correspondre à l'instance qui répond le mieux à vos besoins. Le conteneur de tâches utilise le nombre de v CPUs et la taille de mémoire que vous spécifiez dans le fichier de définition de votre flux de travail, même si le type d'instance dispose de v CPUs et de mémoire supplémentaires.

La liste suivante contient des informations supplémentaires sur les vCPU et l'allocation de mémoire :

- Les allocations de ressources aux conteneurs sont des limites strictes. Si une tâche manque de mémoire ou tente d'utiliser un v supplémentaireCPUs , la tâche génère un journal des erreurs et s'arrête.
- Si vous ne spécifiez aucune exigence en termes de calcul ou de mémoire HealthOmics , omics.c.large sélectionnez et utilisez par défaut une configuration avec 1 vCPU et 1 GiB de mémoire.
- La configuration minimale que vous pouvez demander est de 1 vCPU et 1 GiB de mémoire.
- Si vous spécifiez vCPUs, memory ou GPUs une valeur supérieure aux types d'instance pris en charge, un message d' HealthOmics erreur s'affiche et le flux de travail échoue aux validations

- Si vous spécifiez des unités fractionnaires, HealthOmics arrondissez à l'entier supérieur le plus proche.
- HealthOmics réserve une petite quantité de mémoire (5 %) aux agents de gestion et de journalisation, de sorte que l'allocation complète de mémoire n'est peut-être pas toujours disponible pour l'application associée à la tâche.
- HealthOmics fait correspondre les types d'instances aux exigences de calcul et de mémoire que vous spécifiez, et peut utiliser une combinaison de générations matérielles. Pour cette raison, il peut y avoir des variations mineures dans les durées d'exécution des tâches pour une même tâche.

Ces rubriques fournissent des informations détaillées sur les types d'instances pris HealthOmics en charge.

Rubriques



> **Note**
>
> Pour les instances standard, optimisées pour le calcul et la mémoire, augmentez la taille de bande passante de l'instance si celle-ci nécessite un débit plus élevé. Les instances Amazon EC2 dotées de moins de 16 vCPU (taille 4xl ou moins) peuvent connaître des pics de débit. Pour plus d'informations sur le débit des instances Amazon EC2, consultez la section Bande passante disponible pour les instances Amazon EC2.

Types d'instances standard

Pour les types d'instances standard, les configurations visent à trouver un équilibre entre la puissance de calcul et la mémoire.

HealthOmics prend en charge les instances 32xlarge et 48xlarge dans les régions suivantes : USA Ouest (Oregon) et USA Est (Virginie du Nord).

| Instance | Numéro de v CPUs | Mémoire |
|---|---|---|
| omics.m.large | 2 | 8 GiO |
| omics.m.xlarge | 4 | 16 GiO |
| omics.m.2xlarge | 8 | 32 GiO |
| omics.m.4xlarge | 16 | 64 Go |
| omics.m.8xlarge | 32 | 128 Gio |
| omics.m. 12 x large | 48 | 192 Go |
| omics.m. 16 x large | 64 | 256 Gio |
| omics.m. 24 x large | 96 | 384 Go |
| omics.m. 32 x large | 128 | 512 Gio |
| omics.m. 48 x large | 192 | 768 Gio |

Instances optimisées pour le calcul

Pour les types d'instances optimisés pour le calcul, les configurations ont plus de puissance de calcul et moins de mémoire.

HealthOmics prend en charge les instances 32xlarge et 48xlarge dans les régions suivantes : USA Ouest (Oregon) et USA Est (Virginie du Nord).

| Instance | Numéro de v CPUs | Mémoire |
|---|---|---|
| omics.c.large | 2 | 4 GiB |
| omics.c.xlarge | 4 | 8 GiO |
| omics.c.2 x large | 8 | 16 GiO |
| omics.c.4 x large | 16 | 32 GiO |

| Instance | Numéro de v CPUs | Mémoire |
|---|---|---|
| omics.c.8xlarge | 32 | 64 Go |
| omics.c. 12 x large | 48 | 96 GiB |
| omics.c. 16 x large | 64 | 128 Gio |
| omics.c. 24 x large | 96 | 192 Go |
| omics.c. 32 x large | 128 | 256 Gio |
| omics.c. 48 x large | 192 | 384 Go |

Instances optimisées pour la mémoire

Pour les types d'instances optimisés pour la mémoire, les configurations disposent de moins de puissance de calcul et de plus de mémoire.

HealthOmics prend en charge les instances 32xlarge et 48xlarge dans les régions suivantes : USA Ouest (Oregon) et USA Est (Virginie du Nord).

| Instance | Numéro de v CPUs | Mémoire |
|---|---|---|
| omics s.r.large | 2 | 16 GiO |
| omics.r.xlarge | 4 | 32 GiO |
| omics s.r.l. 2 x large | 8 | 64 Go |
| omics s.r.4 x large | 16 | 128 Gio |
| omics s.r.l. 8 x large | 32 | 256 Gio |
| omics s.r.l. 12 x large | 48 | 384 Go |
| omics s.r.l. 16 x large | 64 | 512 Gio |
| omics s.r.l. 24 x large | 96 | 768 Gio |

| Instance | Numéro de v CPUs | Mémoire |
|---|---|---|
| omics s.r.l. 32 x large | 128 | 1024 GiB |
| omics s.r.48 x large | 192 | 1536 GiB |

Instances de calcul accéléré

Vous pouvez éventuellement spécifier des ressources GPU pour chaque tâche d'un flux de travail, afin d' HealthOmicsallouer une instance de calcul accéléré à la tâche. Pour plus d'informations sur la manière de spécifier les informations du GPU dans le fichier de définition du flux de travail, consultezAccélérateurs de tâches dans une définition de HealthOmics flux de travail.

Si vous spécifiez un GPU qui prend en charge plusieurs types d'instances, HealthOmics sélectionne le type d'instance en fonction de sa disponibilité. Si les deux types d'instance sont disponibles, HealthOmics donne la préférence à l'instance la moins coûteuse.

Les instances G4 ne sont pas prises en charge dans la région d'Israël (Tel Aviv). Les instances G5 ne sont pas prises en charge dans la région Asie-Pacifique (Singapour).

Rubriques

- Types d'instances G6 et G6e
- Instances G4 et G5

Types d'instances G6 et G6e

HealthOmics prend en charge les configurations d'instances de calcul accéléré G6 suivantes. Toutes les instances omics.g6 utilisent Nvidia L4 ou Nvidia L4 A10G. GPUs

HealthOmics prend en charge les instances G6 et G6e dans les régions suivantes : USA Ouest (Oregon) et USA Est (Virginie du Nord).

| Instance | Numéro de v CPUs | Mémoire | Nombre de GPUs | Mémoire GPU | |
|---|---|---|---|---|---|
| omics.g6. xlarge | 4 | 16 GiO | 1 | 24 GiO | |

| Instance | Numéro de v CPUs | Mémoire | Nombre de GPUs | Mémoire GPU | |
|---|---|---|---|---|---|
| omics.g6. 2xlarge | 8 | 32 GiO | 1 | 24 GiO | |
| omics.g6.4 x large | 16 | 64 Go | 1 | 24 GiO | |
| omics.g 6,8 x large | 32 | 128 Gio | 1 | 24 GiO | |
| omics.g 6,12 x large | 48 | 192 Go | 4 | 96 GiB | |
| omics.g6.16 x large | 64 | 256 Gio | 1 | 24 GiO | |
| omics.g 6.24 x large | 96 | 384 Go | 4 | 96 GiB | |

Toutes les instances omics.g6e utilisent Nvidia L40s. GPUs

| Instance | Numéro de v CPUs | Mémoire | Nombre de GPUs | Mémoire GPU | |
|---|---|---|---|---|---|
| omics.g6e .xlarge | 4 | 32 GiO | 1 | 48 GiO | |
| omics.g6e .2xlarge | 8 | 64 Go | 1 | 48 GiO | |
| omics.g6e .4xlarge | 16 | 128 Gio | 1 | 48 GiO | |
| omics.g6e .8xlarge | 32 | 256 Gio | 1 | 48 GiO | |

| Instance | Numéro de v CPUs | Mémoire | Nombre de GPUs | Mémoire GPU | |
|---|---|---|---|---|---|
| omics.g6e .12xlarge | 48 | 384 Go | 4 | 192 Go | |
| omics.g6e.16 x large | 64 | 512 Gio | 1 | 48 GiO | |
| omics.g6e .24xlarge | 96 | 768 Gio | 4 | 192 Go | |

Instances G4 et G5

HealthOmics prend en charge les configurations d'instances de calcul accéléré G4 et G5 suivantes.

Toutes les instances omics.g5 utilisent le Nvidia L4 A10G, le Nvidia Tesla A10G ou le Nvidia Tesla T4 A10G. GPUs

| Instance | Numéro de v CPUs | Mémoire | Nombre de GPUs | Mémoire GPU | |
|---|---|---|---|---|---|
| omics.g5. xlarge | 4 | 16 GiO | 1 | 24 GiO | |
| omics.g5.2 x large | 8 | 32 GiO | 1 | 24 GiO | |
| omics.g 5,4 x large | 16 | 64 Go | 1 | 24 GiO | |
| omics.g 5,8 x large | 32 | 128 Gio | 1 | 24 GiO | |
| omics.g5. 12xlarge | 48 | 192 Go | 4 | 96 GiB | |
| omics.g5.16 x large | 64 | 256 Gio | 1 | 24 GiO | |

| Instance | Numéro de v CPUs | Mémoire | Nombre de GPUs | Mémoire GPU | |
|---|---|---|---|---|---|
| omics.g 5.24 x large | 96 | 384 Go | 4 | 96 GiB | |

Toutes les instances omics.g4dn utilisent le Nvidia Tesla T4 ou le Nvidia Tesla T4 A10G. GPUs

| Instance | Numéro de v CPUs | Mémoire | Nombre de GPUs | Mémoire GPU | |
|---|---|---|---|---|---|
| omics.g4d n.xlarge | 4 | 16 GiO | 1 | 16 GiO | |
| omics.g4d n.2xlarge | 8 | 32 GiO | 1 | 16 GiO | |
| omics.g4d n.4xlarge | 16 | 64 Go | 1 | 16 GiO | |
| omics.g4d n.8xlarge | 32 | 128 Gio | 1 | 16 GiO | |
| omics.g4d n.12xlarge | 48 | 192 Go | 4 | 64 Go | |
| omics.g4d n.16xlarge | 64 | 256 Gio | 1 | 24 GiO | |

## Résultats des tâches dans une définition HealthOmics de flux de travail

Vous spécifiez les résultats des tâches dans la définition du flux de travail. Par défaut, HealthOmics supprime tous les fichiers de tâches intermédiaires lorsque le flux de travail est terminé. Pour exporter un fichier intermédiaire, vous devez le définir comme sortie.

Si vous utilisez la mise en cache des appels, HealthOmics enregistre les résultats des tâches dans le cache, y compris les fichiers intermédiaires que vous définissez comme sorties.

Les rubriques suivantes incluent des exemples de définition de tâches pour chacun des langages de définition de flux de travail.

Rubriques



## Sorties de tâches pour WDL

Pour les définitions de flux de travail écrites en WDL, définissez vos sorties dans la outputs section de flux de travail de niveau supérieur.

HealthOmics

Rubriques



### Sortie de tâche pour STDOUT

Cet exemple crée une tâche nommée `SayHello` qui renvoie le contenu STDOUT au fichier de sortie de la tâche. La stdout fonction WDL capture le contenu STDOUT (dans cet exemple, la chaîne d'entrée Hello World ! ) dans le fichierstdout_file.

Dans la mesure où il HealthOmics crée des journaux pour tout le contenu STDOUT, le résultat apparaît également dans CloudWatch les journaux, avec les autres informations de journalisation STDERR relatives à la tâche.

```
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()
    }
}
```

Sortie de tâche pour STDERR

Cet exemple crée une tâche nommée `SayHello` qui renvoie le contenu STDERR au fichier de sortie de la tâche. La stderr fonction WDL capture le contenu STDERR (dans cet exemple, la chaîne d'entrée Hello World ! ) dans le fichierstderr_file.

Dans la mesure où il HealthOmics crée des journaux pour tout le contenu STDERR, le résultat apparaîtra dans les CloudWatch journaux, avec les autres informations de journalisation STDERR relatives à la tâche.

```
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()
```

```
    }
}
```

Sortie de tâche vers un fichier

Dans cet exemple, la SayHello tâche crée deux fichiers (message.txt et info.txt) et déclare explicitement ces fichiers en tant que sorties nommées (message_file et 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"
    }
}
```

Sortie de tâches vers un tableau de fichiers

Dans cet exemple, la `GenerateGreetings` tâche génère un tableau de fichiers comme résultat de la tâche. La tâche génère dynamiquement un fichier d'accueil pour chaque membre du tableau d'entrée`names`. Comme les noms de fichiers ne sont pas connus avant l'exécution, la définition de sortie utilise la fonction WDL glob () pour afficher tous les fichiers correspondant au modèle. `*_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")
    }
}
```

Sorties de tâches pour Nextflow

Pour les définitions de flux de travail écrites dans Nextflow, définissez une directive PublishDir pour exporter le contenu des tâches vers votre compartiment Amazon S3 de sortie. Définissez la valeur PublishDir sur. `/mnt/workflow/pubdir`

HealthOmics Pour exporter des fichiers vers Amazon S3, les fichiers doivent se trouver dans ce répertoire.

Si une tâche produit un groupe de fichiers de sortie à utiliser comme entrées pour une tâche ultérieure, nous vous recommandons de regrouper ces fichiers dans un répertoire et d'émettre le répertoire en tant que sortie de tâche. L'énumération de chaque fichier individuel peut entraîner un engorgement des E/S dans le système de fichiers sous-jacent. Par exemple :

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

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

```
  }
```

Sorties de tâches pour CWL

Pour les définitions de flux de travail écrites en CWL, vous pouvez spécifier les résultats des tâches à l'aide de `CommandLineTool` tâches. Les sections suivantes présentent des exemples de `CommandLineTool` tâches qui définissent différents types de sorties.

Rubriques

- Sortie de tâche pour STDOUT
- Sortie de tâche pour STDERR
- Sortie de tâche vers un fichier
- Sortie de tâches vers un tableau de fichiers

Sortie de tâche pour STDOUT

Cet exemple crée une `CommandLineTool` tâche qui renvoie le contenu STDOUT dans un fichier de sortie texte nommé. output.txt Par exemple, si vous fournissez l'entrée suivante, le résultat de la tâche est Hello World ! dans le output.txt dossier.

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

La `outputs` directive indique que le nom de sortie est example_out et que son type est`stdout`. Pour qu'une tâche en aval consomme le résultat de cette tâche, elle désignera la sortie sous la forme`example_out`.

Dans la mesure HealthOmics où il crée des journaux pour tout le contenu STDERR et STDOUT, le résultat apparaît également dans les CloudWatch journaux, ainsi que d'autres informations de journalisation STDERR relatives à la tâche.

```
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
```

Sortie de tâche pour STDERR

Cet exemple crée une `CommandLineTool` tâche qui renvoie le contenu STDERR dans un fichier de sortie texte nommé. stderr.txt La tâche modifie le `baseCommand` afin qu'il `echo` écrit sur STDERR (au lieu de STDOUT).

La `outputs` directive indique que le nom de sortie est stderr_out et que son type est`stderr`.

Dans la mesure où il HealthOmics crée des journaux pour tout le contenu STDERR et STDOUT, le résultat apparaîtra dans les CloudWatch journaux, avec les autres informations de journalisation STDERR relatives à la tâche.

```
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
```

Sortie de tâche vers un fichier

Cet exemple crée une `CommandLineTool` tâche qui crée une archive tar compressée à partir des fichiers d'entrée. Vous indiquez le nom de l'archive en tant que paramètre d'entrée (archive_name).

La outputs directive indique que le type `archive_file` de sortie est`File`, et elle utilise une référence au paramètre d'entrée `archive_name` pour établir une liaison avec le fichier de sortie.

```
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
```

Sortie de tâches vers un tableau de fichiers

Dans cet exemple, la `CommandLineTool` tâche crée un tableau de fichiers à l'aide de la `touch` commande. La commande utilise les chaînes du paramètre `files-to-create` d'entrée pour nommer les fichiers. La commande génère un tableau de fichiers. Le tableau inclut tous les fichiers du répertoire de travail qui correspondent au `glob` modèle. Cet exemple utilise un motif générique (« * ») qui correspond à tous les fichiers.

```
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
```

## Ressources de tâches dans une définition HealthOmics de flux de travail

Dans la définition du flux de travail, définissez les éléments suivants pour chaque tâche :

- L'image du conteneur pour la tâche. Pour de plus amples informations, veuillez consulter Images de conteneur pour les flux de travail privés.
- Le nombre CPUs et la mémoire nécessaires pour la tâche. Pour de plus amples informations, veuillez consulter Exigences en matière de calcul et de mémoire pour les HealthOmics tâches.

HealthOmics ignore les spécifications de stockage par tâche. HealthOmics fournit un stockage d'exécution auquel toutes les tâches en cours d'exécution peuvent accéder. Pour de plus amples informations, veuillez consulter Exécuter les types de stockage dans les HealthOmics flux de travail.

WDL

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

Dans le cas d'un flux de travail WDL, deux HealthOmics tentatives au maximum sont effectuées pour une tâche qui échoue en raison d'erreurs de service (la demande d'API renvoie un code d'état HTTP 5XX). Pour plus d'informations sur les nouvelles tentatives de tâches, consultezNouvelles tentatives de tâches.

Vous pouvez désactiver le comportement de nouvelle tentative en spécifiant la configuration suivante pour la tâche dans le fichier de définition 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
```

## Accélérateurs de tâches dans une définition de HealthOmics flux de travail

Dans la définition du flux de travail, vous pouvez éventuellement spécifier les spécifications de l'accélérateur GPU pour une tâche. HealthOmics prend en charge les valeurs de spécification d'accélérateur suivantes, ainsi que les types d'instances pris en charge :

| Spécifications de l'accélérateur | Types d'instances Healthomics | | | | |
|---|---|---|---|---|---|
| nvidia-tesla-t4 | G4 | | | | |
| nvidia-tesla-t4-A 10 g | G4 et G5 | | | | |
| nvidia-tesla-a10 g | G5 | | | | |
| nvidia-l4-a10g | G5 et G6 | | | | |
| nvidia-l4 | G6 | | | | |
| nvidia-l40s | G6e | | | | |

Si vous spécifiez un type d'accélérateur qui prend en charge plusieurs types d'instances, HealthOmics sélectionne le type d'instance en fonction de la capacité disponible. Si les deux types d'instance sont disponibles, HealthOmics donne la préférence à l'instance la moins coûteuse.

Pour plus de détails sur les types d'instances, consultezInstances de calcul accéléré.

Dans l'exemple suivant, la définition du flux de travail indique `nvidia-l4` qu'il s'agit de l'accélérateur :

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"
```

## Spécificités de définition du flux de travail WDL

Les rubriques suivantes fournissent des informations détaillées sur les types et les directives disponibles pour les définitions de flux de travail WDL dans HealthOmics.

Rubriques

- Conversion de type implicite dans WDL Lenient

- Définition de l'espace de noms dans input.json
- Types primitifs dans WDL
- Types complexes dans WDL
- Directives dans WDL
- Métadonnées des tâches dans WDL
- Exemple de définition de flux de travail WDL

Conversion de type implicite dans WDL Lenient

HealthOmics prend en charge la conversion de type implicite dans le fichier input.json et dans la définition du flux de travail. Pour utiliser le casting de type implicite, spécifiez le moteur de flux de travail comme WDL indulgent lorsque vous créez le flux de travail. WDL Lenient est conçu pour gérer les flux de travail migrés depuis Cromwell. Il prend en charge les directives Cromwell du client et certaines logiques non conformes.

WDL Lenient prend en charge la conversion de type pour les éléments suivants de la liste des exceptions limitées de WDL :

- Float jusqu'à Int, où la coercition n'entraîne aucune perte de précision (par exemple, 1,0 correspond à 1).
- String to Int/Float, où la coercition n'entraîne aucune perte de précision.
- Associez [W, X] à Array [Pair [Y, Z]], dans le cas où W est coercible à Y et X est coercible à Z.
- Array [Pair [W, X]] to Map [Y, Z], dans le cas où W est coercible à Y et X est coercible à Z (par exemple, 1.0 correspond à 1).

Pour utiliser le casting de type implicite, spécifiez le moteur de flux de travail sous la forme WDL_LENIENT lorsque vous créez le flux de travail ou la version du flux de travail.

Dans la console, le paramètre du moteur de flux de travail s'appelle Language. Dans l'API, le paramètre du moteur de flux de travail est nommé moteur. Pour plus d'informations, consultez Création d'un flux de travail privé ou Création d'une version de flux de travail.

Définition de l'espace de noms dans input.json

HealthOmics prend en charge les variables entièrement qualifiées dans input.json. Par exemple, si vous déclarez deux variables d'entrée nommées numéro1 et numéro2 dans le flux de travail : SumWorkflow

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

Vous pouvez les utiliser comme variables entièrement qualifiées dans input.json :

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

Types primitifs dans WDL

Le tableau suivant montre comment les entrées de WDL correspondent aux types primitifs correspondants. HealthOmics fournit une prise en charge limitée de la coercition de type. Nous vous recommandons donc de définir des types explicites.

Types primitifs

| Type WDL | Type JSON | Exemple WDL | Exemple de clé et de valeur JSON | Remarques |
|---|---|---|---|---|
| `Boolean` | `boolean` | `Boolean b` | `"b": true` | La valeur doit être en minuscules et sans guillemets. |
| `Int` | `integer` | `Int i` | `"i": 7` | Ne doit pas être entre guillemets. |
| `Float` | `number` | `Float f` | `"f": 42.2` | Ne doit pas être entre guillemets. |
| `String` | `string` | `String s` | `"s": "characters"` | Les chaînes JSON qui sont des URI doivent |

| Type WDL | Type JSON | Exemple WDL | Exemple de clé et de valeur JSON | Remarques |
|---|---|---|---|---|
| | | | | être mappées à un fichier WDL pour être importées. |
| `File` | `string` | `File f` | `"f": "s3://amzn-s3-demo-bucket1/path/to/file"` | Amazon S3 et le HealthOmics stockage URIs sont importés tant que le rôle IAM fourni pour le flux de travail dispose d'un accès en lecture à ces objets. Aucun autre schéma d'URI n'est pris en charge (tel que `file://https://`, et`ftp://`). L'URI doit spécifier un objet. Il ne peut pas s'agir d'un répertoire, ce qui signifie qu'il ne peut pas se terminer par un`/`. |

| Type WDL | Type JSON | Exemple WDL | Exemple de clé et de valeur JSON | Remarques |
|---|---|---|---|---|
| `Directory` | `string` | `Directory d` | `"d": "s3://bucket/path/"` | Le `Directory` type n'est pas inclus dans WDL 1.0 ou 1.1, vous devrez donc l'ajouter `version development` à l'en-tête du fichier WDL. L'URI doit être un URI Amazon S3 et comporter un préfixe se terminant par un «/». Tout le contenu du répertoire sera copié de manière récursive dans le flux de travail sous forme de téléchargement unique. Le ne `Directory` doit contenir que des fichiers liés au flux de travail. |

Types complexes dans WDL

Le tableau suivant montre comment les entrées de WDL correspondent aux types JSON complexes correspondants. Les types complexes dans WDL sont des structures de données composées de types primitifs. Les structures de données telles que les listes seront converties en tableaux.

Types complexes

| Type WDL | Type JSON | Exemple WDL | Exemple de clé et de valeur JSON | Remarques |
|---|---|---|---|---|
| `Array` | `array` | `Array[Int] nums` | `“nums": [1, 2, 3]` | Les membres du tableau doivent suivre le format du type de tableau WDL. |
| `Pair` | `object` | `Pair[String, Int] str_to_i` | `“str_to_i": {"left": "0", "right": 1}` | Chaque valeur de la paire doit utiliser le format JSON du type WDL correspondant. |
| `Map` | `object` | `Map[Int, String] int_to_string` | `"int_to_string": { 2: "hello", 1: "goodbye" }` | Chaque entrée de la carte doit utiliser le format JSON du type WDL correspondant. |
| `Struct` | `object` | `struct SampleBamAndIndex { String sample_name File bam` | `"b_and_i": { "sample_name": "NA12878", "bam": "s3://amz` | Les noms des membres de la structure doivent correspondre exactement aux noms des clés d'objet JSON. |

| Type WDL | Type JSON | Exemple WDL | Exemple de clé et de valeur JSON | Remarques |
|---|---|---|---|---|
| | | File bam_index } SampleBam AndIndex b_and_i | n-s3-demo -bucket1/ NA12878.b am", "bam_inde x": "s3:// amzn- s3-demo- bucket1/ NA12878.b am.bai" } | Chaque valeur doit utiliser le format JSON du type WDL correspondant. |
| `Object` | N/A | N/A | N/A | Le `Object` type WDL est obsolète et doit être remplacé par `Struct` dans tous les cas. |

Directives dans WDL

HealthOmics prend en charge les directives suivantes dans toutes les versions de WDL compatibles HealthOmics .

Configuration des ressources du GPU

HealthOmics prend en charge les attributs d'exécution acceleratorType et acceleratorCount avec toutes les instances de GPU prises en charge. HealthOmics prend également en charge les alias nommés gpuType etgpuCount, qui ont les mêmes fonctionnalités que leurs homologues accélérateurs. Si la définition WDL contient les deux directives, HealthOmics utilise les valeurs de l'accélérateur.

L'exemple suivant montre comment utiliser ces directives :

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

Configurer une nouvelle tentative de tâche pour les erreurs de service

HealthOmics prend en charge jusqu'à deux tentatives pour une tâche qui a échoué en raison d'erreurs de service (codes d'état HTTP 5XX). Vous pouvez configurer le nombre maximum de tentatives (1 ou 2) et vous pouvez désactiver les tentatives en cas d'erreur de service. Par défaut, le nombre maximum de HealthOmics tentatives est de deux tentatives.

L'exemple suivant permet `preemptible` de désactiver les tentatives en cas d'erreur de service :

```
{
  preemptible: 0
}
```

Pour plus d'informations sur les nouvelles tentatives de tâches HealthOmics, consultezNouvelles tentatives de tâches.

Configurer une nouvelle tentative de tâche en cas de mémoire insuffisante

HealthOmics prend en charge les nouvelles tentatives pour une tâche qui a échoué en raison d'un manque de mémoire (code de sortie du conteneur 137, code d'état HTTP 4XX). HealthOmics double la quantité de mémoire à chaque nouvelle tentative.

Par défaut, HealthOmics ne réessaie pas pour ce type d'échec. Utilisez la `maxRetries` directive pour spécifier le nombre maximal de tentatives.

L'exemple suivant définit `maxRetries` la valeur 3, de sorte que quatre HealthOmics tentatives au maximum sont tentées pour terminer la tâche (la première tentative plus trois nouvelles tentatives) :

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

> **Note**
>
> Une nouvelle tentative de tâche en cas de mémoire insuffisante nécessite GNU findutils 4.2.3+. Le conteneur HealthOmics d'images par défaut inclut ce package. Si vous spécifiez

une image personnalisée dans votre définition WDL, assurez-vous qu'elle inclut GNU findutils 4.2.3+.

Configurer les codes de retour

L'attribut ReturnCodes fournit un mécanisme permettant de spécifier un code de retour, ou un ensemble de codes de retour, indiquant l'exécution réussie d'une tâche. Le moteur WDL respecte les codes de retour que vous spécifiez dans la section d'exécution de la définition WDL et définit le statut des tâches en conséquence.

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

HealthOmics prend également en charge un alias nommé continueOnReturnCode, qui possède les mêmes fonctionnalités que ReturnCodes. Si vous spécifiez les deux attributs, HealthOmics utilise la valeur ReturnCodes.

Métadonnées des tâches dans WDL

HealthOmics prend en charge les options de métadonnées suivantes pour les tâches WDL.

Désactiver la mise en cache au niveau des tâches avec l'attribut volatile

L'attribut volatile vous permet de désactiver la mise en cache des appels pour des tâches spécifiques de votre flux de travail WDL. Lorsqu'une tâche est marquée comme volatile, elle s'exécute toujours et n'utilise jamais les résultats mis en cache, même lorsque la mise en cache est activée pour l'exécution.

Ajoutez l'attribut volatile à la méta-section de votre définition de tâche :

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

    input {
        String input_file
    }
```

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

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

Exemple de définition de flux de travail WDL

Les exemples suivants présentent des définitions de flux de travail privés pour la conversion de CRAM vers BAM dans WDL. Le BAM flux CRAM de travail to définit deux tâches et utilise les outils du genomes-in-the-cloud conteneur, qui est illustré dans l'exemple et est accessible au public.

L'exemple suivant montre comment inclure le conteneur Amazon ECR en tant que paramètre. Cela permet HealthOmics de vérifier les autorisations d'accès à votre conteneur avant qu'il ne commence l'exécution.

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

L'exemple suivant montre comment spécifier les fichiers à utiliser lors de votre exécution, lorsque les fichiers se trouvent dans un compartiment Amazon S3.

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

Si vous souhaitez spécifier des fichiers à partir d'un magasin de séquences, indiquez-le comme indiqué dans l'exemple suivant, en utilisant l'URI du magasin de séquences.

```
{
    "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"
}
```

Vous pouvez ensuite définir votre flux de travail dans WDL comme indiqué dans l'exemple suivant.

```
 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}"
     }
}
```

## Caractéristiques de la définition du flux de travail Nextflow

HealthOmics prend en charge DSL1 Nextflow et. DSL2 Pour en savoir plus, consultez Support de la version Nextflow.

Nextflow DSL2 est basé sur le langage de programmation Groovy, les paramètres sont donc dynamiques et la coercition de type est possible en utilisant les mêmes règles que Groovy. Les paramètres et valeurs fournis par le JSON d'entrée sont disponibles dans la carte parameters (`params`) du flux de travail.

Rubriques

- Utiliser les plugins nf-schema et nf-validation
- Spécifier le stockage URIs
- Directives Nextflow
- Exporter le contenu de la tâche

### Utiliser les plugins nf-schema et nf-validation

> **Note**
>
> Résumé de la HealthOmics prise en charge des plugins :

- v22.04 — aucun support pour les plugins
- v23.10 — prend en charge et `nf-schema` `nf-validation`
- v24.10 — prend en charge `nf-schema`

HealthOmics fournit le support suivant pour les plugins Nextflow :

- Pour Nextflow v23.10, HealthOmics préinstalle le plugin nf-validation @1 .1.1.
- Pour Nextflow v23.10 et versions ultérieures, HealthOmics préinstalle le plugin nf-schema @2 .3.0.
- Vous ne pouvez pas récupérer de plug-ins supplémentaires lors de l'exécution d'un flux de travail. HealthOmics ignore toutes les autres versions de plug-in que vous spécifiez dans le `nextflow.config` fichier.
- Pour Nextflow v24 et versions supérieures, `nf-schema` il s'agit de la nouvelle version du plugin obsolète`nf-validation`. Pour plus d'informations, consultez nf-schema dans le référentiel GitHub Nextflow.

Spécifier le stockage URIs

Lorsqu'un Amazon S3 ou un HealthOmics URI est utilisé pour créer un fichier ou un objet de chemin Nextflow, l'objet correspondant est mis à la disposition du flux de travail, à condition que l'accès en lecture soit accordé. L'utilisation de préfixes ou de répertoires est autorisée pour Amazon S3 URIs. Pour obtenir des exemples, consultez Formats de paramètres d'entrée Amazon S3.

HealthOmics prend partiellement en charge l'utilisation de modèles globaux dans Amazon S3 URIs ou HealthOmics Storage URIs. Utilisez des modèles Glob dans la définition du flux de travail pour la création de `path` `file` canaux. Pour le comportement attendu et les cas exacts, voirNextflow Gestion du modèle Glob dans les entrées Amazon S3.

Directives Nextflow

Vous configurez les directives Nextflow dans le fichier de configuration ou la définition du flux de travail Nextflow. La liste suivante indique l'ordre de priorité HealthOmics utilisé pour appliquer les paramètres de configuration, de la priorité la plus faible à la plus élevée :

1. Configuration globale dans le fichier de configuration.
2. Section des tâches de la définition du flux de travail.
3. Sélecteurs spécifiques aux tâches dans le fichier de configuration.

Rubriques



### Stratégie de nouvelle tentative de tâche en utilisant **`errorStrategy`**

Utilisez la `errorStrategy` directive pour définir la stratégie en cas d'erreurs de tâches. Par défaut, lorsqu'une tâche revient avec une indication d'erreur (un statut de sortie différent de zéro), la tâche s'arrête et HealthOmics met fin à l'exécution complète. Si vous définissez cette `errorStrategy` option`retry`, HealthOmics tente une nouvelle tentative de la tâche qui a échoué. Pour augmenter le nombre de tentatives, voirTentatives de nouvelle tentative de tâche en utilisant `maxRetries`.

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

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

Pour plus d'informations sur le mode HealthOmics de gestion des nouvelles tentatives de tâches lors d'une exécution, consultezNouvelles tentatives de tâches.

### Tentatives de nouvelle tentative de tâche en utilisant **`maxRetries`**

Par défaut, HealthOmics ne tente aucune nouvelle tentative d'une tâche qui a échoué, ou tente une nouvelle tentative si vous configurez. `errorStrategy` Pour augmenter le nombre maximum de tentatives, définissez `retry` et configurez le nombre maximum de tentatives `errorStrategy` à l'aide de la `maxRetries` directive.

L'exemple suivant définit le nombre maximum de tentatives à 3 dans la configuration globale.

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

```
    maxRetries = 3
}
```

L'exemple suivant montre comment définir `maxRetries` dans la section des tâches de la définition du flux de travail.

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

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

L'exemple suivant montre comment spécifier une configuration spécifique à une tâche dans le fichier de configuration Nextflow, en fonction du nom ou des sélecteurs d'étiquette.

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

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

Désactiver la tâche, réessayez en utilisant **`omicsRetryOn5xx`**

Pour Nextflow v23 et v24, HealthOmics prend en charge les nouvelles tentatives de tâche si la tâche a échoué en raison d'erreurs de service (codes d'état HTTP 5XX). Par défaut, HealthOmics tente jusqu'à deux tentatives d'une tâche ayant échoué.

Vous pouvez configurer `omicsRetryOn5xx` pour désactiver la rétentative de tâche en cas d'erreur de service. Pour plus d'informations sur la nouvelle tentative d'une tâche HealthOmics, consultezNouvelles tentatives de tâches.

L'exemple suivant permet de configurer `omicsRetryOn5xx` la configuration globale pour désactiver la nouvelle tentative de tâche.

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

L'exemple suivant montre comment procéder à la configuration `omicsRetryOn5xx` dans la section des tâches de la définition du flux de travail.

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

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

L'exemple suivant montre comment définir `omicsRetryOn5xx` une configuration spécifique à une tâche dans le fichier de configuration Nextflow, en fonction du nom ou des sélecteurs d'étiquette.

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

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

Durée de la tâche à l'aide de la **`time`** directive

HealthOmics fournit un quota ajustable (voirHealthOmics quotas de service) pour spécifier la durée maximale d'une course. Pour les flux de travail Nextflow v23 et v24, vous pouvez également spécifier la durée maximale des tâches à l'aide de la directive Nextflow. `time`

Lors du développement d'un nouveau flux de travail, la définition de la durée maximale des tâches vous permet de détecter les tâches intempestives et les tâches de longue durée.

Pour plus d'informations sur la directive temporelle Nextflow, voir directive time dans la référence Nextflow.

HealthOmics fournit le support suivant pour la directive temporelle Nextflow :

1. HealthOmics prend en charge une granularité d'une minute pour la directive horaire. Vous pouvez spécifier une valeur comprise entre 60 secondes et la durée maximale d'exécution.
2. Si vous entrez une valeur inférieure à 60, HealthOmics arrondissez-la à 60 secondes. Pour les valeurs supérieures à 60, HealthOmics arrondissez à la minute inférieure la plus proche.
3. Si le flux de travail prend en charge les nouvelles tentatives pour une tâche, HealthOmics réessayez la tâche si le délai imparti est expiré.
4. Si le délai d'expiration d'une tâche (ou si la dernière tentative expire), elle est HealthOmics annulée. Cette opération peut avoir une durée d'une à deux minutes.
5. En cas d'expiration de la tâche, HealthOmics définit l'exécution et le statut de la tâche sur Échec, et annule les autres tâches en cours d'exécution (pour les tâches en cours d'exécution, en attente ou en cours d'exécution). HealthOmics exporte les sorties des tâches qu'il a terminées avant le délai d'expiration vers l'emplacement de sortie S3 que vous avez désigné.
6. Le temps passé par une tâche en attente n'est pas pris en compte dans la durée de la tâche.
7. Si l'exécution fait partie d'un groupe d'exécution et que le groupe d'exécution expire avant le délai imparti, l'exécution et la tâche passent au statut d'échec.

Spécifiez la durée du délai d'expiration en utilisant une ou plusieurs des unités suivantes :`ms`,`s`, `mh`, oud.

L'exemple suivant montre comment spécifier une configuration globale dans le fichier de configuration Nextflow. Il définit un délai d'expiration global de 1 heure et 30 minutes.

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

L'exemple suivant montre comment spécifier une directive temporelle dans la section des tâches de la définition du flux de travail. Cet exemple définit un délai d'expiration de 3 jours, 5 heures et 4 minutes. Cette valeur a priorité sur la valeur globale du fichier de configuration, mais pas sur une directive temporelle spécifique à une tâche `my_label` dans le fichier de configuration.

```
process myTask {
```

```
    label 'my_label'
    time '3d5h4m'

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

L'exemple suivant montre comment spécifier des directives temporelles spécifiques à une tâche dans le fichier de configuration Nextflow, en fonction du nom ou des sélecteurs d'étiquette. Cet exemple définit un délai d'expiration global de la tâche de 30 minutes. Il définit une valeur de 2 heures pour la tâche `myTask` et une valeur de 3 heures pour les tâches avec étiquette`my_label`. Pour les tâches correspondant au sélecteur, ces valeurs ont priorité sur la valeur globale et sur la valeur de la définition du flux de travail.

```
process {
    time = '30m'

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

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

Exporter le contenu de la tâche

Pour les flux de travail écrits dans Nextflow, définissez une directive PublishDir pour exporter le contenu des tâches vers votre compartiment Amazon S3 de sortie. Comme indiqué dans l'exemple suivant, définissez la valeur PublishDir sur. `/mnt/workflow/pubdir` Pour exporter des fichiers vers Amazon S3, les fichiers doivent se trouver dans ce répertoire.

```
 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
    """
  }
```

## Spécificités de définition du flux de travail CWL

Les flux de travail écrits en langage de flux de travail commun, ou CWL, offrent des fonctionnalités similaires à celles des flux de travail écrits en WDL et Nextflow. Vous pouvez utiliser Amazon S3 ou le HealthOmics stockage URIs comme paramètres d'entrée.

Si vous définissez une entrée dans un fichier secondaire dans un sous-flux de travail, ajoutez la même définition dans le flux de travail principal.

HealthOmics les flux de travail ne prennent pas en charge les processus opérationnels. Pour en savoir plus sur les processus opérationnels dans les flux de travail CWL, consultez la documentation CWL.

La meilleure pratique consiste à définir un flux de travail CWL distinct pour chaque conteneur que vous utilisez. Nous vous recommandons de ne pas coder en dur l'entrée DockerPull avec un URI Amazon ECR fixe.

Rubriques



### Convertissez les flux de travail CWL à utiliser HealthOmics

Pour convertir une définition de flux de travail CWL existante à utiliser HealthOmics, apportez les modifications suivantes :

- Remplacez tous les conteneurs Docker URIs par Amazon URIs ECR.
- Assurez-vous que tous les fichiers de flux de travail sont déclarés en entrée dans le flux de travail principal et que toutes les variables sont définies de manière explicite.

- Assurez-vous que tout le JavaScript code est conforme au mode strict.

Désactiver la tâche, réessayez en utilisant **`omicsRetryOn5xx`**

HealthOmics prend en charge les nouvelles tentatives de tâche si la tâche a échoué en raison d'erreurs de service (codes d'état HTTP 5XX). Par défaut, HealthOmics tente jusqu'à deux tentatives d'une tâche qui a échoué. Pour plus d'informations sur la nouvelle tentative d'une tâche HealthOmics, consultezNouvelles tentatives de tâches.

Pour désactiver la nouvelle tentative de tâche en cas d'erreur de service, configurez la `omicsRetryOn5xx` directive dans la définition du flux de travail. Vous pouvez définir cette directive dans la section « exigences » ou « astuces ». Nous vous recommandons d'ajouter la directive comme indication de portabilité.

```
requirements:
  ResourceRequirement:
    omicsRetryOn5xx: false

hints:
  ResourceRequirement:
    omicsRetryOn5xx: false
```

Les exigences l'emportent sur les conseils. Si l'implémentation d'une tâche fournit un besoin en ressources sous forme d'indications qui est également fourni par les exigences d'un flux de travail englobant, les exigences générales ont priorité.

Si la même exigence de tâche apparaît à différents niveaux du flux de travail, HealthOmics utilise l'entrée la plus spécifique provenant de `requirements` (ou`hints`, s'il n'y a aucune entrée`requirements`). La liste suivante indique l'ordre de priorité HealthOmics utilisé pour appliquer les paramètres de configuration, de la priorité la plus faible à la plus élevée :

- Niveau du flux de travail
- Niveau d'étape
- Section des tâches de la définition du flux de travail

L'exemple suivant montre comment configurer la `omicsRetryOn5xx` directive à différents niveaux du flux de travail. Dans cet exemple, l'exigence relative au niveau du flux de travail remplace les

indications relatives au niveau du flux de travail. Les configurations d'exigences au niveau des tâches et des étapes remplacent les configurations indicatives.

```
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
```

Boucler une étape du flux de travail

HealthOmics permet de mettre en boucle une étape du flux de travail. Vous pouvez utiliser des boucles pour exécuter les étapes du flux de travail de manière répétée jusqu'à ce qu'une condition spécifiée soit remplie. Cela est utile pour les processus itératifs dans lesquels vous devez répéter une tâche plusieurs fois ou jusqu'à ce qu'un certain résultat soit atteint.

Remarque : La fonctionnalité de boucle nécessite la version 1.2 ou ultérieure de CWL. Les flux de travail utilisant des versions CWL antérieures à 1.2 ne prennent pas en charge les opérations en boucle.

Pour utiliser des boucles dans votre flux de travail CWL, définissez une exigence de boucle. L'exemple suivant montre la configuration requise pour les boucles :

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

Le `loopWhen` champ contrôle le moment où la boucle se termine. Dans cet exemple, la boucle continue tant que le compteur est inférieur à la valeur maximale. Le `loop` champ définit la manière dont les paramètres d'entrée sont mis à jour entre les itérations. `loopSource`Spécifie le résultat de l'itération précédente qui alimente l'itération suivante. Le `outputMethod` champ défini sur `last` renvoie uniquement le résultat de l'itération finale.

Réessayer des tâches avec une mémoire accrue

HealthOmics permet une nouvelle tentative automatique en cas d'échec des out-of-memory tâches. Lorsqu'une tâche se termine avec le code 137 (out-of-memory), HealthOmics crée une nouvelle tâche avec une allocation de mémoire accrue en fonction du multiplicateur spécifié.

> **Note**
>
> HealthOmics tente une nouvelle tentative d' out-of-memoryéchec jusqu'à 3 fois ou jusqu'à ce que l'allocation de mémoire atteigne 1 536 GiB, selon la première limite atteinte.

L'exemple suivant montre comment configurer la out-of-memory nouvelle tentative :

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

```
    memoryRetryMultiplier: 2.5
```

Lorsqu'une tâche échoue en raison de out-of-memory, HealthOmics calcule l'allocation de mémoire pour les nouvelles tentatives à l'aide de la formule :. `previous_run_memory × memoryRetryMultiplier` Dans l'exemple ci-dessus, si la tâche avec 4 096 Mo de mémoire échoue, la nouvelle tentative utilise 4 096 × 2,5 = 10 240 Mo de mémoire.

Le `memoryRetryMultiplier` paramètre contrôle la quantité de mémoire supplémentaire à allouer pour les nouvelles tentatives :

- Valeur par défaut : si vous ne spécifiez aucune valeur, la valeur par défaut est 2 (double la mémoire)
- Plage valide : doit être un nombre positif supérieur à1. Les valeurs non valides entraînent une erreur de validation 4XX
- Valeur effective minimale : les valeurs comprises entre 1 et `1.5` sont automatiquement augmentées afin `1.5` de garantir une augmentation significative de la mémoire et d'éviter des tentatives de nouvelle tentative excessives

Exemples

Voici un exemple de flux de travail écrit en CWL.

```
cwlVersion: v1.2
class: Workflow

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

out_filename: string
docker_image: string


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

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

Le fichier suivant définit la `copy.cwl` tâche.

```
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)"
```

Voici un exemple de flux de travail écrit en CWL avec une exigence de GPU.

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

inputs: []
outputs: []
```

## Exemples de définitions de flux de travail

L'exemple suivant montre la même définition de flux de travail dans WDL, Nextflow et 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
```

## Fichiers modèles de paramètres pour les HealthOmics flux de travail

Les modèles de paramètres définissent les paramètres d'entrée d'un flux de travail. Vous pouvez définir des paramètres d'entrée pour rendre votre flux de travail plus flexible et plus polyvalent. Par exemple, vous pouvez définir un paramètre pour l'emplacement des fichiers génomiques de référence sur Amazon S3. Les modèles de paramètres peuvent être fournis par le biais d'un service de dépôt basé sur Git ou de votre disque local. Les utilisateurs peuvent ensuite exécuter le flux de travail à l'aide de différents ensembles de données.

Vous pouvez créer le modèle de paramètres pour votre flux de travail ou HealthOmics générer le modèle de paramètres pour vous.

Le modèle de paramètres est un fichier JSON. Dans le fichier, chaque paramètre d'entrée est un objet nommé qui doit correspondre au nom de l'entrée du flux de travail. Lorsque vous lancez une exécution, si vous ne fournissez pas de valeurs pour tous les paramètres requis, l'exécution échoue.

L'objet du paramètre d'entrée inclut les attributs suivants :

- description— Cet attribut obligatoire est une chaîne que la console affiche sur la page Démarrer l'exécution. Cette description est également conservée sous forme de métadonnées d'exécution.

- optional— Cet attribut facultatif indique si le paramètre d'entrée est facultatif. Si vous ne spécifiez pas le optional champ, le paramètre d'entrée est obligatoire.

L'exemple de modèle de paramètres suivant montre comment spécifier les paramètres d'entrée.

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

## Génération de modèles de paramètres

HealthOmics génère le modèle de paramètres en analysant la définition du flux de travail pour détecter les paramètres d'entrée. Si vous fournissez un fichier modèle de paramètres pour un flux de travail, les paramètres de votre fichier remplacent les paramètres détectés dans la définition du flux de travail.

Il existe de légères différences entre la logique d'analyse des moteurs CWL, WDL et Nextflow, comme décrit dans les sections suivantes.

Rubriques

- Détection de paramètres pour CWL
- Détection de paramètres pour WDL
- Détection de paramètres pour Nextflow

### Détection de paramètres pour CWL

Dans le moteur de flux de travail CWL, la logique d'analyse repose sur les hypothèses suivantes :

- Tous les types pris en charge par des valeurs nulles sont marqués comme paramètres d'entrée facultatifs.
- Tous les types pris en charge non nuls sont marqués comme paramètres d'entrée obligatoires.
- Tous les paramètres avec des valeurs par défaut sont marqués comme paramètres d'entrée facultatifs.
- Les descriptions sont extraites de la `label` section de la définition du `main` flux de travail. Si `label` ce n'est pas spécifié, la description sera vide (chaîne vide).

Les tableaux suivants présentent des exemples d'interpolation CWL. Pour chaque exemple, le nom du paramètre est`x`. Si le paramètre est obligatoire, vous devez fournir une valeur pour le paramètre. Si le paramètre est facultatif, il n'est pas nécessaire de fournir de valeur.

Ce tableau présente des exemples d'interpolation CWL pour les types primitifs.

| Entrée | Exemple d'entrée/sortie | Obligatoire |
|---|---|---|
| `x:`<br>`  type: int` | 1 ou 2 ou... | Oui |
| `x:`<br>`  type: int`<br>`  default: 2` | La valeur par défaut est 2.<br>L'entrée valide est 1 ou 2 ou... | Non |
| `x:`<br>`  type: int?` | L'entrée valide est None ou 1 ou 2 ou... | Non |
| `x:`<br>`  type: int?`<br>`  default: 2` | La valeur par défaut est 2.<br>L'entrée valide est None ou 1 ou 2 ou... | Non |

Le tableau suivant présente des exemples d'interpolation CWL pour les types complexes. Un type complexe est un ensemble de types primitifs.

| Entrée | Exemple d'entrée/sortie | Obligatoire |
| --- | --- | --- |
| `x:`<br>`  type: array`<br>`  items: int` | [] ou [1,2,3] | Oui |
| `x:`<br>`  type: array?`<br>`  items: int` | Aucun ou [] ou [1,2,3] | Non |
| `x:`<br>`  type: array`<br>`  items: int?` | [] ou [Aucun, 3, Aucun] | Oui |
| `x:`<br>`  type: array?`<br>`  items: int?` | [Aucun] ou Aucun ou [1,2,3] ou [Aucun, 3] mais pas [] | Non |

Détection de paramètres pour WDL

Dans le moteur de flux de travail WDL, la logique d'analyse repose sur les hypothèses suivantes :

- Tous les types pris en charge par des valeurs nulles sont marqués comme paramètres d'entrée facultatifs.
- Pour les types pris en charge non nullables :
  - Toute variable d'entrée à laquelle sont assignés des littéraux ou des expressions est marquée comme paramètre facultatif. Par exemple :

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

  - Si aucune valeur ou expression n'a été affectée aux paramètres d'entrée, ils seront marqués comme paramètres obligatoires.
- Les descriptions sont extraites de `parameter_meta` la définition du `main` flux de travail. Si `parameter_meta` ce n'est pas spécifié, la description sera vide (chaîne vide). Pour plus d'informations, consultez la spécification WDL pour les métadonnées des paramètres.

Les tableaux suivants présentent des exemples d'interpolation WDL. Pour chaque exemple, le nom du paramètre est`x`. Si le paramètre est obligatoire, vous devez fournir une valeur pour le paramètre. Si le paramètre est facultatif, il n'est pas nécessaire de fournir de valeur.

Ce tableau présente des exemples d'interpolation WDL pour les types primitifs.

| Entrée | Exemple d'entrée/sortie | Obligatoire |
|---|---|---|
| Int x | 1 ou 2 ou... | Oui |
| Int x = 2 | 2 | Non |
| Int x = 1+2 | 3 | Non |
| Int x = y+z | y+z | Non |
| Int ? x | Aucun, 1 ou 2 ou... | Oui |
| Int ? x = 2 | Aucun ou 2 | Non |
| Int ? x = 1+2 | Aucun ou 3 | Non |
| Int ? x = y+z | Aucun ou y+z | Non |

Le tableau suivant présente des exemples d'interpolation WDL pour les types complexes. Un type complexe est un ensemble de types primitifs.

| Entrée | Exemple d'entrée/sortie | Obligatoire | | |
|---|---|---|---|---|
| Tableau [Int] x | [1,2,3] ou [] | Oui | | |
| Tableau [Int] + x | [1], mais pas [] | Oui | | |
| Tableau [Int] ? x | Aucun ou [] ou [1,2,3] | Non | | |
| Tableau [Int ?] x | [] ou [Aucun, 3, Aucun] | Oui | | |

| Entrée | Exemple d'entrée/sortie | Obligatoire | | |
|---|---|---|---|---|
| Tableau [Int ?] = ? x | [Aucun] ou Aucun ou [1,2,3] ou [Aucun, 3] mais pas [] | Non | | |
| Exemple de structure {String a, Int y}<br><br>plus loin dans les entrées : Sample MySample | `String a = mySample.a Int y = mySample.y` | Oui | | |
| Exemple de structure {String a, Int y}<br><br>plus tard dans les entrées : Sample ? Mon échantillon | `if (defined( mySample)) { String a = mySample.a Int y = mySample.y }` | Non | | |

Détection de paramètres pour Nextflow

Pour Nextflow, HealthOmics génère le modèle de paramètres en analysant le `nextflow_schema.json` fichier. Si la définition du flux de travail n'inclut pas de fichier de schéma, HealthOmics analyse le fichier de définition du flux de travail principal.

Rubriques

- Analyse du fichier de schéma
- Analyse du fichier principal
- Paramètres imbriqués
- Exemples d'interpolation Nextflow

### Analyse du fichier de schéma

Pour que l'analyse fonctionne correctement, assurez-vous que le fichier de schéma répond aux exigences suivantes :

- Le fichier de schéma est nommé `nextflow_schema.json` et se trouve dans le même répertoire que le fichier de flux de travail principal.
- Le fichier de schéma est un fichier JSON valide tel que défini dans l'un des schémas suivants :
  - schéma json. org/draft/2020-12/schema.
  - schéma json. org/draft-07/schema.

HealthOmics analyse le `nextflow_schema.json` fichier pour générer le modèle de paramètres :

- Extrait tout properties ce qui est défini dans le schéma.
- Comprend la propriété description si disponible pour la propriété.
- Indique si chaque paramètre est facultatif ou obligatoire, en fonction du required champ de la propriété.

L'exemple suivant montre un fichier de définition et le fichier de paramètres généré.

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

Le modèle de paramètres généré :

```
{
    "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
    }
}
```

Analyse du fichier principal

Si la définition du flux de travail n'inclut aucun `nextflow_schema.json` fichier, HealthOmics analyse le fichier de définition du flux de travail principal.

HealthOmics analyse les `params` expressions présentes dans le fichier de définition du flux de travail principal et dans le `nextflow.config` fichier. Tous ceux `params` dont les valeurs par défaut sont marqués comme facultatifs.

Pour que l'analyse fonctionne correctement, tenez compte des exigences suivantes :

- HealthOmics analyse uniquement le fichier de définition du flux de travail principal. Pour garantir que tous les paramètres sont capturés, nous vous recommandons de les connecter à tous params les sous-modules et aux flux de travail importés.
- Le fichier de configuration est facultatif. Si vous en définissez un, nommez-le `nextflow.config` et placez-le dans le même répertoire que le fichier de définition du flux de travail principal.

L'exemple suivant montre un fichier de définition et le modèle de paramètres généré.

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

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

Le modèle de paramètres généré :

```
{
    "input_file": {
        "description": None,
```

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

Pour les valeurs par défaut définies dans nextflow.config, HealthOmics collecte les `params` assignations et les paramètres déclarés dans le fichier`params {}`, comme indiqué dans l'exemple suivant. Dans les instructions d'affectation, elles `params` doivent apparaître dans la partie gauche de la déclaration.

```
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}"
```

Le modèle de paramètres généré :

```
{
    // 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
    }
}
```

Paramètres imbriqués

Les deux `nextflow_schema.json` et `nextflow.config` autorisent les paramètres imbriqués. Toutefois, le modèle de HealthOmics paramètres ne nécessite que les paramètres de niveau supérieur. Si votre flux de travail utilise un paramètre imbriqué, vous devez fournir un objet JSON en entrée pour ce paramètre.

Paramètres imbriqués dans les fichiers de schéma

HealthOmics sauts imbriqués params lors de l'analyse d'un fichier. `nextflow_schema.json` Par exemple, si vous définissez le `nextflow_schema.json` fichier suivant :

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

HealthOmics ignore `input_file` et `input_num` lorsqu'il génère le modèle de paramètres :

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

Lorsque vous exécutez ce flux de travail, HealthOmics vous vous attendez à un `input.json` fichier similaire au suivant :

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

Paramètres imbriqués dans les fichiers de configuration

HealthOmics ne collecte pas les données imbriquées params dans un `nextflow.config` fichier et les ignore lors de l'analyse. Par exemple, si vous définissez le `nextflow.config` fichier suivant :

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

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

HealthOmics ignore `params.nested.beta` et `params.group.delta` lorsqu'il génère le modèle de paramètres :

```
{
```

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

Exemples d'interpolation Nextflow

Le tableau suivant présente des exemples d'interpolation Nextflow pour les paramètres du fichier principal.

| Paramètres | Obligatoire |
| --- | --- |
| params.input_file | Oui |
| params.input_file = "s3://bucket/data.json » | Non |
| params.nested.input_file | N/A |
| params.nested.input_file = "s3://bucket/data.json » | N/A |

Le tableau suivant présente des exemples d'interpolation Nextflow pour les paramètres du fichier. `nextflow.config`

| Paramètres | Obligatoire |
| --- | --- |
| `params.input_file = "s3://bucket/data.json"` | Non |
| `params { input_file = "s3://bucket/data.json" }` | Non |

| Paramètres | Obligatoire |
| --- | --- |
| `params { nested { input_file = "s3://bucket/data.json" } }` | N/A |
| `input_file = params.input_file` | N/A |

## Images de conteneur pour les flux de travail privés

HealthOmics prend en charge les images de conteneurs hébergées dans les référentiels privés Amazon ECR. Vous pouvez créer des images de conteneur et les télécharger dans le référentiel privé. Vous pouvez également utiliser votre registre privé Amazon ECR comme cache d'extraction pour synchroniser le contenu des registres en amont.

Votre référentiel Amazon ECR doit résider dans la même AWS région que le compte appelant le service. Une autre personne Compte AWS peut être propriétaire de l'image du conteneur, à condition que le référentiel d'images source fournisse les autorisations appropriées. Pour de plus amples informations, veuillez consulter Politiques relatives à l'accès multicompte à Amazon ECR.

Nous vous recommandons de définir votre image de conteneur Amazon ECR en URIs tant que paramètre de votre flux de travail afin que l'accès puisse être vérifié avant le début de l'exécution. Cela facilite également l'exécution d'un flux de travail dans une nouvelle région en modifiant le paramètre Région.

> **Note**
>
> HealthOmics ne prend pas en charge les conteneurs ARM et ne prend pas en charge l'accès aux référentiels publics.

Pour plus d'informations sur la configuration des autorisations IAM pour accéder HealthOmics à Amazon ECR, consultez. HealthOmics Autorisations relatives aux ressources

Rubriques



## Synchronisation avec des registres de conteneurs tiers

Vous pouvez utiliser les règles de cache d'extraction d'Amazon ECR pour synchroniser les référentiels d'un registre en amont pris en charge avec vos référentiels privés Amazon ECR. Pour plus d'informations, consultez Synchroniser un registre en amont dans le guide de l'utilisateur Amazon ECR.

Le cache d'extraction crée automatiquement le référentiel d'images dans votre registre privé lorsque vous créez le cache, et il se synchronise automatiquement avec l'image mise en cache lorsque des modifications sont apportées à l'image en amont.

HealthOmics prend en charge le cache d'extraction pour les registres en amont suivants :

- Amazon ECR Public
- Registre d'images de conteneurs Kubernetes
- Quay
- Docker Hub
- Microsoft Azure Container Registry
- GitHub Registre des conteneurs
- GitLab Registre des conteneurs

HealthOmics ne prend pas en charge le cache d'extraction pour un référentiel privé Amazon ECR en amont.

Les avantages de l'utilisation du cache d'extraction Amazon ECR sont les suivants :

1. Vous évitez d'avoir à migrer manuellement les images de conteneur vers Amazon ECR ou à synchroniser les mises à jour depuis le référentiel tiers.

2. Les flux de travail accèdent aux images de conteneur synchronisées de votre référentiel privé, ce qui est plus fiable que le téléchargement de contenu depuis un registre public au moment de l'exécution.
3. Comme les caches d'extraction Amazon ECR utilisent une structure d'URI prévisible, le HealthOmics service peut automatiquement mapper l'URI privé Amazon ECR avec l'URI de registre en amont. Vous n'êtes pas obligé de mettre à jour et de remplacer les valeurs d'URI dans la définition du flux de travail.

Rubriques



## Configuration du cache d'extraction

Amazon ECR fournit un registre pour vous Compte AWS dans chaque région. Assurez-vous de créer la configuration Amazon ECR dans la même région que celle où vous prévoyez d'exécuter le flux de travail.

Les sections suivantes décrivent les tâches de configuration du cache d'extraction.

Tâches de configuration



### Création d'une règle de cache d'extraction

Créez une règle de cache d'extraction Amazon ECR pour chaque registre en amont contenant des images que vous souhaitez mettre en cache. Une règle spécifie un mappage entre un registre en amont et le référentiel privé Amazon ECR.

Pour un registre en amont qui nécessite une authentification, vous devez fournir vos informations d'identification à l'aide d'AWS Secrets Manager.

> **Note**
>
> Ne modifiez pas une règle de cache d'extraction lorsqu'une exécution active utilise le référentiel privé. L'exécution pourrait échouer ou, plus grave encore, entraîner l'utilisation d'images inattendues dans votre pipeline.

Pour plus d'informations, consultez la section Création d'une règle de cache d'extraction dans le guide de l'utilisateur d'Amazon Elastic Container Registry.

Création d'une règle de cache d'extraction à l'aide de la console

Pour configurer le cache d'extraction, procédez comme suit à l'aide de la console Amazon ECR :

1. Ouvrez la console Amazon ECR : https://console.aws.amazon.com /ecr
2. Dans le menu de gauche, sous Registre privé, développez Fonctionnalités et paramètres, puis choisissez Pull through cache.
3. Sur la page du cache d'extraction, choisissez Ajouter une règle.
4. Dans le panneau de registre Upstream, choisissez le registre amont à synchroniser avec votre registre privé, puis choisissez Next.
5. Si le registre en amont nécessite une authentification, la console ouvre une nouvelle page dans laquelle vous spécifiez le secret SageMaker AI qui contient vos informations d'identification. Choisissez Suivant.
6. Sous Spécifier les espaces de noms, dans le panneau de l'espace de noms du cache, choisissez de créer les référentiels privés à l'aide d'un préfixe de référentiel spécifique ou sans préfixe. Si vous choisissez d'utiliser un préfixe, spécifiez le nom du préfixe dans le préfixe du référentiel de cache.
7. Dans le panneau de l'espace de noms Upstream, choisissez si vous souhaitez extraire des référentiels en amont en utilisant un préfixe de référentiel spécifique ou sans préfixe. Si vous choisissez d'utiliser un préfixe, spécifiez le nom du préfixe dans le préfixe du référentiel Upstream.

   Le panneau d'exemple Namespace affiche un exemple de pull request, une URL en amont et l'URL du référentiel de cache créé.
8. Choisissez Suivant.
9. Vérifiez la configuration et choisissez Create pour créer la règle.

Pour plus d'informations, voir Création d'une règle de cache d'extraction (console AWS de gestion).

Création d'une règle de cache d'extraction à l'aide de la CLI

Utilisez la create-pull-through-cache-rule commande Amazon ECR pour créer une règle de cache d'extraction. Pour les registres en amont qui nécessitent une authentification, stockez les informations d'identification dans un secret Secrets Manager.

Les sections suivantes fournissent des exemples pour chaque registre en amont pris en charge.

Pour Amazon ECR Public

L'exemple suivant crée une règle de mise en cache par extraction pour le registre public Amazon ECR. Il spécifie un préfixe de référentiel de `ecr-public`, ce qui fait que chaque référentiel créé à l'aide de la règle de mise en cache par extraction aura le schéma de dénomination de `ecr-public/`*`upstream-repository-name`*.

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

Pour Kubernetes Container Registry

L'exemple suivant crée une règle de mise en cache par extraction pour le registre public Kubernetes. Il spécifie un préfixe de référentiel de `kubernetes`, ce qui fait que chaque référentiel créé à l'aide de la règle de mise en cache par extraction aura le schéma de dénomination de `kubernetes/`*`upstream-repository-name`*.

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

Pour Quay

L'exemple suivant crée une règle de mise en cache par extraction pour le registre public Quay. Il spécifie un préfixe de référentiel de `quay`, ce qui fait que chaque référentiel créé à l'aide de la règle de mise en cache par extraction aura le schéma de dénomination de `quay/`*`upstream-repository-name`*.

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

Pour Docker Hub

L'exemple suivant crée une règle de mise en cache par extraction pour le registre Docker Hub. Il spécifie un préfixe de référentiel de `docker-hub`, ce qui fait que chaque référentiel créé à l'aide de la règle de mise en cache par extraction aura le schéma de dénomination de `docker-hub/`*`upstream-repository-name`*. Vous devez spécifier l'Amazon Resource Name (ARN) complet du secret contenant vos informations d'identification Docker Hub.

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

Pour le registre des GitHub conteneurs

L'exemple suivant crée une règle de cache d'extraction pour le registre des GitHub conteneurs. Il spécifie un préfixe de référentiel de `github`, ce qui fait que chaque référentiel créé à l'aide de la règle de mise en cache par extraction aura le schéma de dénomination de `github/`*`upstream-repository-name`*. Vous devez spécifier le nom Amazon Resource Name (ARN) complet du secret contenant vos informations d'identification du registre des GitHub conteneurs.

```
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
```

Pour Microsoft Azure Container Registry

L'exemple suivant crée une règle de cache d'extraction pour le Microsoft Azure Container Registry. Il spécifie un préfixe de référentiel de `azure`, ce qui fait que chaque référentiel créé à l'aide de la règle de mise en cache par extraction aura le schéma de dénomination de `azure/`*`upstream-`*

*repository-name*. Vous devez spécifier l'Amazon Resource Name (ARN) complet du secret contenant vos informations d'identification Azure Container Registry.

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

Pour le registre des GitLab conteneurs

L'exemple suivant crée une règle de cache d'extraction pour le registre des GitLab conteneurs. Il spécifie un préfixe de référentiel de `gitlab`, ce qui fait que chaque référentiel créé à l'aide de la règle de mise en cache par extraction aura le schéma de dénomination de `gitlab/`*upstream-repository-name*. Vous devez spécifier le nom Amazon Resource Name (ARN) complet du secret contenant vos informations d'identification du registre des GitLab conteneurs.

```
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
```

Pour plus d'informations, consultez la section Créer une règle de cache d'extraction (CLI) dans le guide de l'utilisateur Amazon ECR.

Vous pouvez utiliser la commande get-run-task CLI pour récupérer des informations sur l'image du conteneur utilisée pour une tâche spécifique :

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

La sortie inclut les informations suivantes concernant l'image du conteneur :

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

```
}
```

Autorisations de registre pour le registre en amont

Utilisez les autorisations de registre HealthOmics pour autoriser l'utilisation du cache d'extraction et l'extraction des images du conteneur dans le registre privé Amazon ECR. Ajoutez une politique de registre Amazon ECR au registre qui fournit les conteneurs utilisés lors des exécutions.

La politique suivante autorise le HealthOmics service à créer des référentiels avec le ou les préfixes de cache d'extraction spécifiés et à lancer des extractions en amont dans ces référentiels.

1. Depuis la console Amazon ECR, ouvrez le menu de gauche, sous Registre privé, développez les autorisations du registre, puis choisissez Generate statement.
2. En haut à droite, choisissez JSON. Entrez une politique similaire à la suivante :

   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/*"
            ]
        }
    ]
}
```

Modèles de création de référentiels

Pour utiliser la mise en cache par extraction HealthOmics, le référentiel Amazon ECR doit disposer d'un modèle de création de référentiel. Le modèle définit les paramètres de configuration lorsque vous ou Amazon ECR créez un référentiel privé pour un registre en amont.

Chaque modèle contient un préfixe d'espace de noms de référentiel, qu'Amazon ECR utilise pour associer les nouveaux référentiels à un modèle spécifique. Les modèles spécifient la configuration de tous les paramètres du référentiel, notamment les politiques d'accès basées sur les ressources, l'immuabilité des balises, le chiffrement et les politiques de cycle de vie.

Pour plus d'informations, consultez les modèles de création de référentiels dans le guide de l'utilisateur d'Amazon Elastic Container Registry.

Comment créer un modèle de création de référentiel :

1. Depuis la console Amazon ECR, ouvrez le menu de gauche, sous Registre privé, développez Fonctionnalités et paramètres, puis choisissez Modèles de création de référentiel.
2. Sélectionnez Create template (Créer un modèle).
3. Dans Détails du modèle, choisissez Pull through cache.
4. Choisissez d'appliquer ce modèle à un préfixe spécifique ou à tous les référentiels qui ne correspondent pas à un autre modèle.

   Si vous choisissez Un préfixe spécifique, entrez la valeur du préfixe de l'espace de noms dans Préfixe. Vous avez spécifié ce préfixe lors de la création de la règle PTC.
5. Choisissez Suivant.
6. Dans la page de configuration de la création d'un référentiel, entrez les autorisations du référentiel. Utilisez l'un des exemples de déclarations de politique ou saisissez-en un similaire à l'exemple suivant :

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "PTCRepoCreationTemplate",
            "Effect": "Allow",
```

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

7. Vous pouvez éventuellement ajouter des paramètres de référentiel tels que la politique de cycle de vie et les balises. Amazon ECR applique ces règles à toutes les images de conteneur créées pour le cache d'extraction qui utilisent le préfixe spécifié.
8. Choisissez Suivant.
9. Vérifiez la configuration et choisissez Next.

### Création du flux de travail

Lorsque vous créez un nouveau flux de travail ou une nouvelle version de flux de travail, passez en revue les mappages de registre et mettez-les à jour si nécessaire. Pour en savoir plus, consultez Création d'un flux de travail privé.

### Mappages de registres

Vous définissez des mappages de registre pour mapper les préfixes de votre registre Amazon ECR privé et les noms de registre en amont.

Pour plus d'informations sur les mappages de registre Amazon ECR, consultez Création d'une règle de cache d'extraction dans Amazon ECR.

L'exemple suivant montre les mappages de registre vers Docker Hub, Quay et Amazon ECR Public.

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

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

Mappages d'images

Vous définissez des mappages d'images pour établir une correspondance entre les noms d'images tels que définis dans vos flux de travail Amazon ECR privés et les noms d'images dans le registre en amont.

Vous pouvez utiliser des mappages d'images avec des registres qui prennent en charge le pull through cache. Vous pouvez également utiliser des mappages d'images avec des registres en amont où le cache d'extraction HealthOmics n'est pas pris en charge. Vous devez synchroniser manuellement le registre en amont avec votre dépôt privé.

Pour plus d'informations sur les mappages d'images Amazon ECR, consultez Création d'une règle de cache d'extraction dans Amazon ECR.

L'exemple suivant montre les mappages entre des images Amazon ECR privées et une image génomique publique et la dernière image Ubuntu.

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

## Considérations générales relatives aux images de conteneurs Amazon ECR

- Architecture

  HealthOmics prend en charge les conteneurs x86_64. Si votre machine locale est basée sur ARM, comme Apple Mac, utilisez une commande telle que la suivante pour créer une image de conteneur x86_64 :

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

- Point d'entrée et coque

  HealthOmics les moteurs de flux de travail injectent des scripts bash pour remplacer les commandes des images de conteneur utilisées par les tâches de flux de travail. Ainsi, les images de conteneur doivent être créées sans point d'entrée spécifié, de sorte qu'un shell bash soit le shell par défaut.

- Chemins montés

  Un système de fichiers partagé est monté sur les tâches du conteneur dans /tmp. Toutes les données ou tous les outils intégrés à l'image du conteneur à cet emplacement seront remplacés.

  La définition du flux de travail est accessible aux tâches via un montage en lecture seule sur /mnt/workflow.

- Taille de l'image

  Voir HealthOmics quotas de taille fixe du flux de travail pour les tailles maximales des images des conteneurs.

## Variables d'environnement pour les HealthOmics flux de travail

HealthOmics fournit des variables d'environnement contenant des informations sur le flux de travail exécuté dans le conteneur. Vous pouvez utiliser les valeurs de ces variables dans la logique de vos tâches de flux de travail.

Toutes les variables HealthOmics de flux de travail commencent par le `AWS_WORKFLOW_` préfixe. Ce préfixe est un préfixe de variable d'environnement protégée. N'utilisez pas ce préfixe pour vos propres variables dans les conteneurs de flux de travail.

HealthOmics fournit les variables d'environnement de flux de travail suivantes :

AWS_REGION

Cette variable correspond à la région dans laquelle le conteneur est exécuté.

AWS_WORKFLOW_EXÉCUTER

Cette variable est le nom de l'exécution en cours.

AWS_WORKFLOW_RUN_ID

Cette variable est l'identifiant de l'exécution en cours.

AWS_WORKFLOW_EXÉCUTER_UUID

Cette variable est l'UUID d'exécution de l'exécution en cours.

AWS_WORKFLOW_TÂCHE

Cette variable est le nom de la tâche en cours.

AWS_WORKFLOW_IDENTIFIANT DE TÂCHE

Cette variable est l'identifiant de la tâche en cours.

AWS_WORKFLOW_UUID DE TÂCHE

Cette variable est l'UUID de la tâche en cours.

L'exemple suivant montre les valeurs typiques de chaque variable d'environnement :

```
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
```

## Utilisation de Java dans les images de conteneurs Amazon ECR

Si une tâche de flux de travail utilise une application Java telle que GATK, tenez compte des exigences de mémoire suivantes pour le conteneur :

- Les applications Java utilisent de la mémoire en pile et de la mémoire en tas. Par défaut, la mémoire de segment maximale est un pourcentage de la mémoire totale disponible dans le

conteneur. Cette valeur par défaut dépend de la distribution et de la version de la JVM spécifiques. Consultez donc la documentation correspondante à votre machine virtuelle Java ou définissez explicitement le maximum de mémoire à l'aide des options de ligne de commande Java (telles que `-Xmx`).

- Ne définissez pas la mémoire de segment maximale à 100 % de l'allocation de mémoire du conteneur, car la pile JVM nécessite également de la mémoire. De la mémoire est également requise pour le ramasse-miettes de la machine virtuelle Java et pour tout autre processus du système d'exploitation exécuté dans le conteneur.
- Certaines applications Java, telles que GATK, peuvent utiliser des invocations de méthodes natives ou d'autres optimisations telles que des fichiers de mappage de mémoire. Ces techniques nécessitent des allocations de mémoire effectuées « hors segment », qui ne sont pas contrôlées par le paramètre de mémoire maximale de la JVM.

  Si vous savez (ou pensez) que votre application Java alloue de la mémoire hors segment, assurez-vous que l'allocation de mémoire aux tâches inclut les exigences en matière de mémoire externe.

  Si ces allocations hors segment entraînent un manque de mémoire du conteneur, vous ne verrez généralement pas d'OutOfMemoryerreur Java, car la JVM ne contrôle pas cette mémoire.

## Ajouter des entrées de tâches à une image de conteneur Amazon ECR

Ajoutez tous les exécutables, bibliothèques et scripts nécessaires pour exécuter une tâche de flux de travail dans l'image Amazon ECR utilisée pour exécuter la tâche.

Il est recommandé d'éviter d'utiliser des scripts, des fichiers binaires et des bibliothèques externes à une image de conteneur de tâches. Cela est particulièrement important lorsque vous utilisez `nf-core` des flux de travail qui utilisent un `bin` répertoire dans le cadre du package de flux de travail. Bien que ce répertoire soit disponible pour la tâche de flux de travail, il est monté en tant que répertoire en lecture seule. Les ressources requises dans ce répertoire doivent être copiées dans l'image de la tâche et mises à disposition lors de l'exécution ou lors de la création de l'image de conteneur utilisée pour la tâche.

Voir HealthOmics quotas de taille fixe du flux de travail pour connaître la taille maximale de l'image de conteneur prise HealthOmics en charge.

# HealthOmics Fichiers README du flux de travail

Vous pouvez télécharger un fichier README.md contenant des instructions, des diagrammes et des informations essentielles pour votre flux de travail. Chaque version du flux de travail prend en charge un fichier README, que vous pouvez mettre à jour à tout moment.

Les exigences du README incluent :

- Le fichier README doit être au format Markdown (.md)
- Taille de fichier maximale : 500 KiB

Rubriques



## Utiliser un fichier README existant

READMEs exportés depuis les référentiels Git contiennent des liens relatifs qui ne fonctionnent généralement pas en dehors du référentiel. HealthOmics L'intégration Git les convertit automatiquement en liens absolus pour un rendu correct dans la console, éliminant ainsi le besoin de mises à jour manuelles des URL.

Pour les images READMEs importées depuis Amazon S3 ou des lecteurs locaux, les images et les liens doivent être soit publics, URLs soit avoir leurs chemins relatifs mis à jour pour un rendu correct.

> **Note**
>
> Les images doivent être hébergées publiquement pour être affichées dans la HealthOmics console. Les images stockées dans GitHub Enterprise Server ou dans GitLab Self-Managed des référentiels ne peuvent pas être rendues.

## Conditions de rendu

La HealthOmics console interpole les images et les liens accessibles au public à l'aide de chemins absolus. Pour effectuer un rendu à URLs partir de référentiels privés, l'utilisateur doit avoir accès au référentiel. GitLab Self-ManagedLes référentiels For GitHub Enterprise Server ou, qui utilisent

des domaines personnalisés, HealthOmics ne peuvent pas résoudre les liens relatifs ni afficher les images stockées dans ces référentiels privés.

Le tableau suivant indique les éléments Markdown pris en charge par la vue README de la AWS console.

| Element | AWS console |
| --- | --- |
| Alerts (Alertes) | Oui, mais sans icônes |
| Badges | Oui |
| Formatage de texte de base | Oui |
| Blocs de code | Oui, mais ne possède pas de fonctionnalité de surlignage syntaxique et de bouton de copie |
| Sections pliables | Oui |
| Rubriques | Oui |
| Formats d'image | Oui |
| Image (cliquable) | Oui |
| Sauts de ligne | Oui |
| Schéma de la sirène | Peut uniquement ouvrir le graphique, déplacer la position du graphique et copier le code |
| Quotes | Oui |
| Indice et exposant | Oui |
| Tables | Oui, mais ne prend pas en charge l'alignement du texte |
| Alignement du texte | Oui |

Utilisation de l'image et du lien URLs

En fonction de votre fournisseur de source, structurez votre base URLs de pages et d'images dans les formats suivants.

- `{username}`: nom d'utilisateur où le dépôt est hébergé.
- `{repo}`: nom du référentiel.
- `{ref}`: référence source (branche, balise et ID de validation).
- `{path}`: chemin du fichier vers la page ou l'image dans le référentiel.

| Fournisseur de source | URL de la page | URL de l'image |
| --- | --- | --- |
| GitHub | `https://github.com/{username}/{repo}/blob/{ref}/{path}` | `https://github.com/{username}/{repo}/blob/{ref}/{path}?raw=true`<br><br>`https://raw.githubusercontent.com/{username}/{repo}/{ref}/{path}` |
| GitLab | `https://gitlab.com/{username}/{repo}/-/blob/{ref}/{path}` | `https://gitlab.com/{username}/{repo}/-/raw/{ref}/{path}` |
| Bitbucket | `https://bitbucket.org/{username}/{repo}/src/{ref}/{path}` | `https://bitbucket.org/{username}/{repo}/raw/{ref}/{path}` |

GitHubGitLab, et Bitbucket prend en charge à la fois les pages et URLs les images qui renvoient vers un dépôt public. Le tableau suivant indique la prise en charge par chaque fournisseur de source pour le rendu d'images et de liens URLs pour les référentiels privés.

| Support pour les référentiels privés | | |
|---|---|---|
| Fournisseur de source | URL de la page | URL de l'image |
| GitHub | Uniquement avec accès au référentiel | Non |
| GitLab | Uniquement avec accès au référentiel | Non |
| Bitbucket | Uniquement avec accès au référentiel | Non |

## Demande de licences Sentieon pour des flux de travail privés

Si votre flux de travail privé utilise le logiciel Sentieon, vous avez besoin d'une licence Senieon. Pour demander et configurer une licence pour le logiciel Sentieon, procédez comme suit :

- Demandez une licence Sentieon
  - Envoyez un e-mail au groupe de support Sentieon (support@sentieon.com) pour demander une licence logicielle.
    - Indiquez votre nom d'utilisateur AWS canonique dans l'e-mail.
    - Trouvez votre nom d'utilisateur AWS canonique en suivant ces instructions.
- Mettez à jour votre rôle de HealthOmics service pour lui accorder l'accès au proxy du serveur de licences Sentieon et au bucket Sentieon Omics de votre région. L'exemple suivant autorise l'accès à`us-east-1`. Si nécessaire, remplacez ce texte par votre région.

  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/*"
            ]
        }
    ]
}
```

- Générez un dossier d' AWS assistance pour accéder au proxy du serveur de licences Sentieon.
  - Pour créer un dossier d'assistance, rendez-vous sur support.console.aws.amazon.com.
  - Indiquez votre région Compte AWS et votre région dans le dossier d'assistance. Votre compte est ajouté à la liste des autorisations pour le proxy du serveur de licences.
- Créez votre flux de travail privé à l'aide du conteneur Sentieon et du script de licence Sentieon.
  - Pour des instructions supplémentaires sur l'utilisation des outils Sentieon dans des flux de travail privés, voir Sentieon-Amazon-Omics dans. GitHub
- La version 202112.07 et supérieure du logiciel Sentieon prend en charge le proxy du HealthOmics serveur de licences. Pour utiliser les versions du logiciel Sentieon antérieures au 202112.07, contactez le support Sentieon.

## Linters de flux de travail dans HealthOmics

Après avoir créé un flux de travail, nous vous recommandons d'exécuter un linter sur le flux de travail avant de commencer la première exécution. Le linter détecte les erreurs susceptibles de provoquer l'échec de l'exécution.

Pour WDL, exécute HealthOmics automatiquement un linter lorsque vous créez le flux de travail. La sortie Linter est disponible dans le `statusMessage` champ de la get-workflow réponse. Utilisez la commande CLI suivante pour récupérer la sortie d'état (utilisez l'ID du flux de travail WDL que vous avez créé) :

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

HealthOmics fournit des linters que vous pouvez exécuter sur la définition du flux de travail avant de créer le flux de travail. Exécutez ces linters sur les pipelines existants vers lesquels vous effectuez la migration. HealthOmics

- WDL— Une image Amazon ECR publique pour exécuter un linter WDL.

- Nextflow— Une image Amazon ECR publique pour exécuter les règles Linter pour Nextflow. Vous pouvez accéder au code source de ce linter à partir de GitHub.
- CWL— non disponible

## HealthOmics opérations de flux de travail

Pour créer un flux de travail privé, vous devez :

- Workflow definition file:Un fichier de définition de flux de travail écrit en WDLNextflow, ouCWL. La définition du flux de travail spécifie les entrées et les sorties pour les exécutions qui utilisent le flux de travail. Il inclut également des spécifications pour les exécutions et les tâches d'exécution pour votre flux de travail, y compris les exigences en matière de calcul et de mémoire. Le fichier de définition du flux de travail doit être au `.zip` format. Pour plus d'informations, consultez la section Fichiers de définition du flux de travail dans HealthOmics.
  - Vous pouvez utiliser Amazon Q CLI pour créer et valider vos fichiers de définition de flux de travail dans WDL, Nextflow et CWL. Pour plus d'informations, consultez les exemples d'instructions pour Amazon Q CLI et le didacticiel HealthOmics Agentic Generative AI sur. GitHub
- (Optional) Parameter template file:Un fichier de modèle de paramètres écrit enJSON. Créez le fichier pour définir les paramètres d'exécution ou HealthOmics générez le modèle de paramètres pour vous. Pour plus d'informations, consultez la section Fichiers modèles de paramètres pour les HealthOmics flux de travail.
- Amazon ECR container images:Créez des référentiels Amazon ECR privés pour chaque conteneur utilisé dans le flux de travail. Créez des images de conteneur pour le flux de travail et stockez-les dans un référentiel privé, ou synchronisez le contenu d'un registre en amont pris en charge avec votre référentiel privé ECR.
- (Optional) Sentieon licenses:Demandez une Sentieon licence pour utiliser le Sentieon logiciel dans des flux de travail privés.

Pour les fichiers de définition de flux de travail supérieurs à 4 Mo (compressés), choisissez l'une des options suivantes lors de la création du flux de travail :

- Téléchargez-le dans un dossier Amazon Simple Storage Service et spécifiez l'emplacement.

- Téléchargez vers un référentiel externe (taille maximale de 1 GiB) et spécifiez les détails du référentiel.

Après avoir créé un flux de travail, vous pouvez mettre à jour les informations de flux de travail suivantes avec l'`UpdateWorkflow`opération :

- Nom
- Description
- Type de stockage par défaut
- Capacité de stockage par défaut (avec ID de flux de travail)
- fichier README.md

Pour modifier d'autres informations du flux de travail, créez un nouveau flux de travail ou une nouvelle version de flux de travail.

Utilisez le versionnement des flux de travail pour organiser et structurer vos flux de travail. Les versions vous aident également à gérer l'introduction de mises à jour itératives du flux de travail. Pour plus d'informations sur les versions, consultez Création d'une version de flux de travail.

Rubriques



## Création d'un flux de travail privé

Créez un flux de travail à l'aide de la HealthOmics console, des commandes de la AWS CLI ou de l'une des AWS SDKs.

> **Note**
>
> N'incluez aucune information personnellement identifiable (PII) dans les noms des flux de travail. Ces noms sont visibles dans les CloudWatch journaux.

Lorsque vous créez un flux de travail, HealthOmics attribuez un identifiant unique universel (UUID) au flux de travail. L'UUID du flux de travail est un identifiant global unique (GUID) unique pour tous les flux de travail et toutes les versions de flux de travail. À des fins de provenance des données, nous vous recommandons d'utiliser l'UUID du flux de travail pour identifier les flux de travail de manière unique.

Si vos tâches de flux de travail utilisent des outils externes (exécutables, bibliothèques ou scripts), vous créez ces outils dans une image de conteneur. Vous disposez des options suivantes pour héberger l'image du conteneur :

- Hébergez l'image du conteneur dans le registre privé ECR. Les conditions requises pour cette option sont les suivantes :
  - Créez un référentiel privé ECR ou choisissez un référentiel existant.
  - Configurez la politique de ressources ECR comme décrit dansAutorisations Amazon ECR.
  - Téléchargez l'image de votre conteneur dans le référentiel privé.
- Synchronisez l'image du conteneur avec le contenu d'un registre tiers pris en charge. Les conditions requises pour cette option sont les suivantes :
  - Dans le registre privé ECR, configurez une règle de cache d'extraction pour chaque registre en amont. Pour de plus amples informations, veuillez consulter Mappages d'images.
  - Configurez la politique de ressources ECR comme décrit dansAutorisations Amazon ECR.
  - Créez des modèles de création de référentiels. Le modèle définit les paramètres à partir desquels Amazon ECR crée le référentiel privé pour un registre en amont.
  - Créez des mappages de préfixes pour remapper les références d'images de conteneur dans la définition du flux de travail avec les espaces de noms du cache ECR.

Lorsque vous créez un flux de travail, vous fournissez une définition de flux de travail contenant des informations sur le flux de travail, les exécutions et les tâches. HealthOmics peut récupérer la définition du flux de travail sous forme d'archive .zip stockée localement ou dans un compartiment Amazon S3, ou à partir d'un référentiel Git compatible.

Rubriques

- Création d'un flux de travail à l'aide de la console
- Création d'un flux de travail à l'aide de la CLI
- Création d'un flux de travail à l'aide d'un SDK

## Création d'un flux de travail à l'aide de la console

### Étapes de création d'un flux de travail

1. Ouvrez la HealthOmics console.
2. Si nécessaire, ouvrez le volet de navigation de gauche (≡). Choisissez des flux de travail privés.
3. Sur la page Flux de travail privés, choisissez Créer un flux de travail.
4. Sur la page Définir le flux de travail, fournissez les informations suivantes :
    1. Nom du flux de travail : nom distinctif pour ce flux de travail. Nous vous recommandons de définir des noms de flux de travail pour organiser vos exécutions dans la AWS HealthOmics console et CloudWatch les journaux.
    2. Description (facultatif) : description de ce flux de travail.
5. Dans le panneau de définition du flux de travail, fournissez les informations suivantes :
    1. Langue du flux de travail (facultatif) : sélectionnez la langue de spécification du flux de travail. Dans le cas contraire, HealthOmics détermine la langue à partir de la définition du flux de travail.
    2. Pour la source de définition du flux de travail, choisissez d'importer le dossier de définition à partir d'un référentiel Git, d'un emplacement Amazon S3 ou d'un lecteur local.
        a. Pour l'importation depuis un service de référentiel :

> **Note**
>
> HealthOmics prend en charge les référentiels publics et privés pour GitHubGitLab,, BitbucketGitHub self-managed,GitLab self-managed.

        i. Choisissez une connexion pour connecter vos AWS ressources au référentiel externe. Pour créer une connexion, voirConnectez-vous à des référentiels de code externes.

> **Note**
>
> Les clients de la TLV région doivent créer une connexion dans la région IAD (us-east-1) pour créer un flux de travail.

ii. Dans ID de référentiel complet, entrez votre ID de référentiel sous forme de nom d'utilisateur/nom de dépôt. Vérifiez que vous avez accès aux fichiers de ce dépôt.

iii. Dans Référence de source (facultatif), entrez une référence de source de référentiel (branche, balise ou ID de validation). HealthOmics utilise la branche par défaut si aucune référence de source n'est spécifiée.

iv. Dans Exclure les modèles de fichiers, entrez les modèles de fichiers pour exclure des dossiers, des fichiers ou des extensions spécifiques. Cela permet de gérer la taille des données lors de l'importation de fichiers de référentiel. Il y a un maximum de 50 modèles, et les modèles doivent suivre la syntaxe du modèle global. Par exemple :

A. `tests/`

B. `*.jpeg`

C. `large_data.zip`

b. Pour Sélectionner le dossier de définition depuis S3 :

i. Entrez l'emplacement Amazon S3 qui contient le dossier de définition du flux de travail compressé. Le compartiment Amazon S3 doit se trouver dans la même région que le flux de travail.

ii. Si votre compte ne possède pas le compartiment Amazon S3, entrez l'ID de AWS compte du propriétaire du compartiment dans l'ID de compte du propriétaire du compartiment S3. Ces informations sont nécessaires pour vérifier HealthOmics la propriété du compartiment.

c. Pour Sélectionner le dossier de définition à partir d'une source locale :

i. Entrez l'emplacement du lecteur local du dossier de définition du flux de travail compressé.

3. Chemin du fichier de définition du flux de travail principal (facultatif) : entrez le chemin du fichier depuis le dossier ou le référentiel de définition du flux de travail compressé vers le `main` fichier. Ce paramètre n'est pas obligatoire s'il n'existe qu'un seul fichier dans le dossier de définition du flux de travail ou si le fichier principal est nommé « main ».

6. Dans le panneau du fichier README (facultatif), sélectionnez la source du fichier README et fournissez les informations suivantes :

- Pour Importer depuis un service de référentiel, dans le champ Chemin du fichier README, entrez le chemin du fichier README dans le référentiel.
- Pour Select file from S3, dans le fichier README dans S3, entrez l'URI Amazon S3 du fichier README.

- Pour Sélectionner un fichier à partir d'une source locale : dans README, facultatif, choisissez Choisir un fichier pour sélectionner le fichier Markdown (.md) à télécharger.

7. Dans le panneau de configuration du stockage d'exécution par défaut, indiquez le type de stockage d'exécution par défaut et la capacité pour les exécutions utilisant ce flux de travail :

   1. Type de stockage d'exécution : choisissez d'utiliser le stockage statique ou dynamique par défaut pour le stockage d'exécution temporaire. La valeur par défaut est le stockage statique.
   2. Capacité de stockage d'exécution (facultatif) : pour le type de stockage d'exécution statique, vous pouvez entrer la quantité de stockage d'exécution par défaut requise pour ce flux de travail. La valeur par défaut de ce paramètre est de 1200 GiB. Vous pouvez remplacer ces valeurs par défaut lorsque vous démarrez une course.

8. Balises (facultatif) : vous pouvez associer jusqu'à 50 balises à ce flux de travail.
9. Choisissez Suivant.
10. Sur la page Ajouter des paramètres de flux de travail (facultatif), sélectionnez la source du paramètre :

    1. Pour Parse from fichier de définition de flux de travail, HealthOmics analysera automatiquement les paramètres du flux de travail à partir du fichier de définition de flux de travail.
    2. Pour Provide parameter template from Git repository, utilisez le chemin d'accès au fichier de modèle de paramètres depuis votre dépôt.
    3. Pour Select JSON file from local source, téléchargez un JSON fichier depuis une source locale qui spécifie les paramètres.
    4. Pour saisir manuellement les paramètres du flux de travail, entrez manuellement les noms et les descriptions des paramètres.

11. Dans le panneau d'aperçu des paramètres, vous pouvez consulter ou modifier les paramètres de cette version du flux de travail. Si vous restaurez le JSON fichier, vous perdrez toutes les modifications locales que vous avez apportées.
12. Choisissez Suivant.
13. Sur la page de remappage des URI du conteneur, dans le panneau Règles de mappage, vous pouvez définir des règles de mappage des URI pour votre flux de travail.

    Pour Source du fichier de mappage, sélectionnez l'une des options suivantes :

    - Aucune : aucune règle de mappage n'est requise.

- Sélectionnez le fichier JSON dans S3 — Spécifiez l'emplacement S3 du fichier de mappage.
- Sélectionnez un fichier JSON à partir d'une source locale : spécifiez l'emplacement du fichier de mappage sur votre appareil local.
- Entrez les mappages manuellement : entrez les mappages de registre et les mappages d'images dans le panneau Mappages.

14. La console affiche le panneau Mappings. Si vous avez choisi un fichier source de mappage, la console affiche les valeurs du fichier.

    a. Dans les mappages de registre, vous pouvez modifier les mappages ou ajouter des mappages (maximum de 20 mappages de registre).

       Chaque mappage de registre contient les champs suivants :

       - URL du registre en amont : URI du registre en amont.
       - Préfixe du référentiel ECR : préfixe du référentiel à utiliser dans le référentiel privé Amazon ECR.
       - (Facultatif) Préfixe du référentiel en amont : préfixe du référentiel dans le registre en amont.
       - (Facultatif) ID de compte ECR : ID de compte du compte propriétaire de l'image du conteneur en amont.

    b. Dans Mappages d'images, vous pouvez modifier les mappages d'images ou ajouter des mappages (maximum de 100 mappages d'images).

       Chaque mappage d'image contient les champs suivants :

       - Image source — Spécifie l'URI de l'image source dans le registre en amont.
       - Image de destination — Spécifie l'URI de l'image correspondante dans le registre Amazon ECR privé.

15. Choisissez Suivant.
16. Vérifiez la configuration du flux de travail, puis choisissez Créer un flux de travail.

### Création d'un flux de travail à l'aide de la CLI

Si vos fichiers de flux de travail et le fichier de modèle de paramètres se trouvent sur votre machine locale, vous pouvez créer un flux de travail à l'aide de la commande CLI suivante.

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

L'`create-workflow`opération renvoie la réponse suivante :

```
{
  "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"
}
```

Paramètres facultatifs à utiliser lors de la création d'un flux de travail

Vous pouvez spécifier n'importe quel paramètre facultatif lorsque vous créez un flux de travail. Pour plus de détails sur la syntaxe, consultez CreateWorkflowle manuel de référence des HealthOmics API AWS.

Rubriques

- Spécifiez la définition du flux de travail (emplacement Amazon S3)
- Utiliser la définition du flux de travail à partir d'un référentiel basé sur Git
- Spécifier un fichier Readme
- Spécifiez le fichier main de définition
- Spécifiez le type de stockage d'exécution
- Spécifiez la configuration du GPU
- Configuration des paramètres de mappage du cache à extraction

Spécifiez la définition du flux de travail (emplacement Amazon S3)

Si le fichier de définition de votre flux de travail se trouve dans un dossier Amazon S3, spécifiez l'emplacement à l'aide du `definition-uri` paramètre, comme indiqué dans l'exemple suivant. Si votre compte ne possède pas le compartiment Amazon S3, fournissez l' Compte AWS ID du propriétaire.

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

Utiliser la définition du flux de travail à partir d'un référentiel basé sur Git

Pour utiliser la définition du flux de travail provenant d'un référentiel Git compatible, utilisez le `definition-repository` paramètre dans votre demande. Ne fournissez aucun autre `definition` paramètre, car une demande échoue si elle inclut plusieurs sources d'entrée.

Le `definition-respository` paramètre contient les champs suivants :

- connectionArn— ARN de la connexion au code qui connecte vos ressources AWS au référentiel externe.
- fullRepositoryId— Entrez l'ID du référentiel sous la forme`owner-name/repo-name`. Vérifiez que vous avez accès aux fichiers de ce dépôt.
- sourceReference(Facultatif) — Entrez un type de référence de référentiel (BRANCH, TAG ou COMMIT) et une valeur.

  HealthOmics utilise le dernier commit sur la branche par défaut si vous ne spécifiez pas de référence de source.
- excludeFilePatterns(Facultatif) — Entrez les modèles de fichiers pour exclure des dossiers, des fichiers ou des extensions spécifiques. Cela permet de gérer la taille des données lors de l'importation de fichiers de référentiel. Fournissez un maximum de 50 modèles. Les modèles doivent suivre la syntaxe du modèle global. Par exemple :
  - `tests/`
  - `*.jpeg`
  - `large_data.zip`

Lorsque vous spécifiez la définition du flux de travail à partir d'un référentiel basé sur Git, utilisez-la `parameter-template-path` pour spécifier le fichier de modèle de paramètres. Si vous ne fournissez pas ce paramètre, HealthOmics crée le flux de travail sans modèle de paramètres.

L'exemple suivant montre les paramètres liés au contenu d'un dépôt privé basé sur 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" \
```

Pour d'autres exemples, consultez le billet de blog How To Create an AWS HealthOmics Workflows from Content in Git.

Spécifier un fichier Readme

Vous pouvez spécifier l'emplacement du fichier README à l'aide de l'un des paramètres suivants :

- readme-markdown— Entrée d'une chaîne ou d'un fichier sur votre machine locale.
- readme-uri— L'URI d'un fichier stocké sur S3.
- readme-path — Le chemin d'accès au fichier README dans le référentiel.

Utilisez readme-path uniquement en conjonction avec definition-respository. Si vous ne spécifiez aucun paramètre README, HealthOmics importe le fichier README.md de niveau racine dans le référentiel (s'il existe).

Les exemples suivants montrent comment spécifier l'emplacement du fichier README en utilisant readme-path et 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"
```

Pour de plus amples informations, veuillez consulter HealthOmics Fichiers README du flux de travail.

Spécifiez le fichier main de définition

Si vous incluez plusieurs fichiers de définition de flux de travail, utilisez le `main` paramètre pour spécifier le fichier de définition principal de votre flux de travail.

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

Spécifiez le type de stockage d'exécution

Vous pouvez spécifier le type de stockage d'exécution par défaut (DYNAMIC ou STATIC) et la capacité de stockage d'exécution (requise pour le stockage statique). Pour plus d'informations sur les types de stockage d'exécution, consultezExécuter les types de stockage dans les HealthOmics flux de travail.

```
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  \
```

Spécifiez la configuration du GPU

Utilisez le paramètre accelerators pour créer un flux de travail qui s'exécute sur une instance de calcul accéléré. L'exemple suivant montre comment utiliser le `accelerators` paramètre. Vous spécifiez la configuration du GPU dans la définition du flux de travail. Consultez Instances de calcul accéléré.

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

Configuration des paramètres de mappage du cache à extraction

Si vous utilisez la fonctionnalité de mappage de cache par extraction d'Amazon ECR, vous pouvez remplacer les mappages par défaut. Pour plus d'informations sur les paramètres de configuration du conteneur, consultezImages de conteneur pour les flux de travail privés.

Dans l'exemple suivant, le fichier `mappings.json` contient le contenu suivant :

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

Spécifiez les paramètres de mappage dans la commande create-workflow :

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

Vous pouvez également spécifier l'emplacement S3 du fichier de paramètres de mappage :

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

Création d'un flux de travail à l'aide d'un SDK

Vous pouvez créer un flux de travail à l'aide de l'un des SDKs. L'exemple suivant montre comment créer un flux de travail à l'aide du SDK Python

```
import boto3

omics = boto3.client('omics')

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

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

## Mettre à jour un flux de travail privé

Vous pouvez mettre à jour un flux de travail à l'aide de la HealthOmics console, des commandes de la AWS CLI ou de l'une des AWS SDKs.

> **Note**
>
> N'incluez aucune information personnellement identifiable (PII) dans les noms des flux de travail. Ces noms sont visibles dans les CloudWatch journaux.

Rubriques



## Mettre à jour un flux de travail à l'aide de la console

Étapes pour mettre à jour un flux de travail

1. Ouvrez la HealthOmics console.
2. Si nécessaire, ouvrez le volet de navigation de gauche (≡). Choisissez des flux de travail privés.
3. Sur la page Flux de travail privés, choisissez le flux de travail à mettre à jour.
4. Sur la page Workflow :
   - Si le flux de travail comporte des versions, assurez-vous de sélectionner la version par défaut.
   - Choisissez Modifier dans la liste des actions.
5. Sur la page Modifier le flux de travail, vous pouvez modifier l'une des valeurs suivantes :
   - Nom du flux de travail.
   - Description du flux de travail.
   - Type de stockage Run par défaut pour le flux de travail.
   - Capacité de stockage d'exécution par défaut (si le type de stockage d'exécution est un stockage statique). Pour plus d'informations sur la configuration d'exécution du stockage par défaut, consultezCréation d'un flux de travail à l'aide de la console.
6. Choisissez Enregistrer les modifications pour appliquer les modifications.

## Mettre à jour un flux de travail à l'aide de la CLI

Comme le montre l'exemple suivant, vous pouvez mettre à jour le nom et la description du flux de travail. Vous pouvez également modifier le type de stockage d'exécution par défaut (STATIC ou DYNAMIC) et la capacité de stockage d'exécution (pour le type de stockage statique). Pour plus d'informations sur les types de stockage d'exécution, consultezExécuter les types de stockage dans les HealthOmics flux de travail.

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

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

Vous ne recevez pas de réponse à cette `update-workflow` demande.

Mettre à jour un flux de travail à l'aide d'un SDK

Vous pouvez mettre à jour un flux de travail à l'aide de l'un des SDKs.

L'exemple suivant montre comment mettre à jour un flux de travail à l'aide du SDK Python

```
import boto3

omics = boto3.client('omics')

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

## Supprimer un flux de travail privé

Lorsque vous n'avez plus besoin d'un flux de travail, vous pouvez le supprimer à l'aide de la HealthOmics console, des commandes de la AWS CLI ou de l'une des commandes AWS SDKs. Vous pouvez supprimer un flux de travail répondant aux critères suivants :

- Son statut est ACTIF ou ÉCHEC.
- Elle ne possède aucune action active.
- Vous avez supprimé toutes les versions du flux de travail.

La suppression d'un flux de travail n'a aucune incidence sur les exécutions en cours utilisant le flux de travail.

Rubriques

- Supprimer un flux de travail à l'aide de la console
- Supprimer un flux de travail à l'aide de la CLI
- Supprimer un flux de travail à l'aide d'un SDK

Supprimer un flux de travail à l'aide de la console

Pour supprimer un flux de travail

1. Ouvrez la HealthOmics console.
2. Si nécessaire, ouvrez le volet de navigation de gauche (≡). Choisissez des flux de travail privés.
3. Sur la page Flux de travail privés, choisissez le flux de travail à supprimer.
4. Sur la page Workflow, choisissez Supprimer la sélection dans la liste des actions.
5. Dans le mode Supprimer le flux de travail, entrez « confirmer » pour confirmer la suppression.
6. Sélectionnez Delete (Supprimer).

Supprimer un flux de travail à l'aide de la CLI

L'exemple suivant montre comment utiliser la AWS CLI commande pour supprimer un flux de travail. Pour exécuter l'exemple, remplacez le *workflow id* par l'ID du flux de travail que vous souhaitez supprimer.

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

HealthOmics n'envoie pas de réponse à la `delete-workflow` demande.

Supprimer un flux de travail à l'aide d'un SDK

Vous pouvez supprimer un flux de travail à l'aide de l'un des SDKs.

L'exemple suivant montre comment supprimer un flux de travail à l'aide du SDK Python.

```
import boto3

omics = boto3.client('omics')

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

## Vérifier l'état du flux de travail

Après avoir créé votre flux de travail, vous pouvez vérifier son statut et consulter d'autres détails du flux de travail à l'aide de get-workflow, comme indiqué.

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

La réponse inclut les détails du flux de travail, y compris le statut, comme indiqué.

```
{
    "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"
  }
```

Vous pouvez démarrer une exécution à l'aide de ce flux de travail une fois que le statut est passé à`ACTIVE`.

## Référencer des fichiers génomiques à partir d'une définition de flux de travail

Un objet de magasin de HealthOmics référence peut être référencé avec un URI comme celui-ci. Utilisez le vôtre *`account ID`*`reference store ID`, et *`reference ID`* là où cela est indiqué.

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

Certains flux de travail nécessiteront à la fois `INDEX` les fichiers `SOURCE` et pour le génome de référence. L'URI précédent est la forme abrégée par défaut et sera le fichier SOURCE par défaut. Pour spécifier l'un ou l'autre des fichiers, vous pouvez utiliser le format URI long, comme suit.

```
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
```

L'utilisation d'un ensemble de lecture de séquences aurait un schéma similaire, comme indiqué.

```
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
```

Certains ensembles de lecture, tels que ceux basés sur FASTQ, peuvent contenir des lectures jumelées. Dans les exemples suivants, ils sont appelés SOURCE1 et SOURCE2. Les formats tels que BAM et CRAM n'auront qu'un SOURCE1 fichier. Certains ensembles de lecture contiendront des fichiers INDEX tels que `crai` des fichiers `bai` ou. L'URI précédent est la forme abrégée par défaut et sera le SOURCE1 fichier par défaut. Pour spécifier le fichier ou l'index exact, vous pouvez utiliser le format URI long, comme suit.

```
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
```

Voici un exemple de fichier JSON d'entrée qui utilise deux 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>"
}
```

Référencez le fichier JSON d'entrée dans le AWS CLI en l'ajoutant `--inputs file://<input_file.json>` à votre demande de démarrage.

# Versionnage des flux de travail dans HealthOmics

Si vous devez apporter des modifications à un flux de travail, vous pouvez créer un nouveau flux de travail ou une nouvelle version de flux de travail. Les versions sont immuables, à l'exception des modifications de configuration autorisées qui n'ont aucun impact sur la logique d'exécution.

Les versions Workflow offrent les avantages suivants :

- Les versions forment un groupe logique de flux de travail liés entre eux. Vous pouvez ajouter un nom défini par l'utilisateur à chaque version du flux de travail afin de les gérer plus facilement (en particulier pour un flux de travail comportant un grand nombre de versions).
- Vous pouvez exécuter plusieurs versions d'un flux de travail en même temps.
- Toutes les versions d'un flux de travail partagent le même ID de flux de travail et le même ARN de base, ce qui peut simplifier la gestion du pipeline une fois que vous avez modifié un flux de travail.
- Les versions de flux de travail fournissent le même niveau de provenance des données que les flux de travail. Les versions sont immuables et HealthOmics créent un ARN unique pour chaque version du flux de travail. L'ARN de la version inclut l'ID du flux de travail et le nom de la version, comme indiqué dans l'exemple suivant :

  ```
  arn:aws:omics:us-west-2:123456789012:workflow/1234567/version/
  myUniqueVersionName
  ```
- Si vous possédez un flux de travail partagé, vous pouvez le mettre à jour sans perturber les abonnés (qui peuvent continuer à utiliser la version précédente). Les abonnés peuvent accéder à toutes les versions du flux de travail. Si vous créez une nouvelle version, il n'est pas nécessaire de partager à nouveau le flux de travail.
- Lorsque vous lancez une exécution de flux de travail, vous pouvez spécifier la version du flux de travail.
  - Les utilisateurs peuvent choisir de rester sur une version stable pour les cycles de production et d'essayer la dernière version pour un essai.
  - Les utilisateurs peuvent revenir à la version précédente d'un flux de travail s'ils rencontrent des problèmes avec la nouvelle version.
  - Les abonnés d'un flux de travail partagé peuvent choisir la version à utiliser.

Rubriques

## Version du flux de travail par défaut

Après avoir créé une ou plusieurs versions d'un flux de travail, HealthOmics traite le flux de travail d'origine comme la version par défaut. Lorsque vous démarrez une exécution, vous pouvez éventuellement spécifier une version de flux de travail pour l'exécution. Si vous ne spécifiez pas de version lorsque vous lancez une exécution, HealthOmics utilise la version par défaut.

Dans la console, HealthOmics indique le flux de travail d'origine avec une étiquette de version par défaut. La console utilise cette étiquette uniquement après avoir créé une ou plusieurs versions du flux de travail. Le flux de travail d'origine reste toujours la version par défaut. Vous ne pouvez attribuer aucune autre version comme version par défaut.

Vous ne pouvez pas supprimer la version par défaut d'un flux de travail si d'autres versions sont associées au flux de travail. Pour de plus amples informations, veuillez consulter Supprimer un flux de travail privé.

## Création d'une version de flux de travail

Lorsque vous créez une nouvelle version d'un flux de travail, vous devez spécifier les valeurs de configuration pour cette nouvelle version. Il n'hérite d'aucune valeur de configuration du flux de travail.

Lorsque vous créez la version, fournissez un nom de version unique pour ce flux de travail. Vous ne pouvez pas modifier le nom après avoir HealthOmics créé la version.

Le nom de version doit commencer par une lettre ou un chiffre et peut inclure des lettres majuscules et minuscules, des chiffres, des traits d'union, des points et des traits de soulignement. La longueur maximale est de 64 caractères. Par exemple, vous pouvez utiliser un schéma de dénomination simple, tel que version1, version2, version3. Vous pouvez également faire correspondre les versions de votre flux de travail à vos propres conventions de version internes, telles que 2.7.0, 2.7.1, 2.7.2.

Vous pouvez éventuellement utiliser le champ de description de la version pour ajouter des remarques sur cette version. Par exemple : Fix for syntax error in workflow definition.

> **Note**
>
> N'incluez aucune information personnellement identifiable (PII) dans le nom de la version. Les noms de version apparaissent dans l'ARN de la version du flux de travail.

HealthOmics attribue un ARN unique à la version du flux de travail. L'ARN est unique en fonction de la combinaison de l'ID du flux de travail et du nom de version.

> **Warning**
>
> Après avoir supprimé une version de flux de travail, HealthOmics vous pouvez réutiliser le nom de version pour une autre version de flux de travail. La meilleure pratique consiste à ne pas réutiliser les noms de version. Si vous réutilisez un nom, le flux de travail et chaque version possèdent un UUID unique que vous pouvez utiliser pour la provenance.

Rubriques



## Création d'une version du flux de travail à l'aide de la console

Étapes pour créer une version de flux de travail

1. Ouvrez la HealthOmics console.
2. Si nécessaire, ouvrez le volet de navigation de gauche (≡). Choisissez des flux de travail privés.
3. Sur la page Flux de travail privés, choisissez le flux de travail pour la nouvelle version.
4. Sur la page des détails du flux de travail, choisissez Créer une nouvelle version.
5. Sur la page Créer une version, fournissez les informations suivantes :
    1. Nom de version : entrez un nom unique pour la version du flux de travail dans le flux de travail.
    2. Description de la version (facultatif) : vous pouvez utiliser le champ de description pour ajouter des remarques sur cette version.
6. Dans le panneau de définition du flux de travail, fournissez les informations suivantes :
    1. Langue du flux de travail (facultatif) : sélectionnez la langue de spécification pour la version du flux de travail. Dans le cas contraire, HealthOmics détermine la langue à partir de la définition du flux de travail.

2. Pour la source de définition du flux de travail, choisissez d'importer le dossier de définition à partir d'un référentiel Git, d'un emplacement Amazon S3 ou d'un lecteur local.

   a. Pour l'importation depuis un service de référentiel :

   > **Note**
   >
   > HealthOmics prend en charge les référentiels publics et privés pour GitHubGitLab,, BitbucketGitHub self-managed,GitLab self-managed.

      i. Choisissez une connexion pour connecter vos AWS ressources au référentiel externe. Pour créer une connexion, voirConnectez-vous à des référentiels de code externes.

      > **Note**
      >
      > Les clients de la TLV région doivent créer une connexion dans la région IAD (us-east-1) pour créer un flux de travail.

      ii. Dans ID de référentiel complet, entrez votre ID de référentiel sous forme de nom d'utilisateur/nom de dépôt. Vérifiez que vous avez accès aux fichiers de ce dépôt.

      iii. Dans Référence de source (facultatif), entrez une référence de source de référentiel (branche, balise ou ID de validation). HealthOmics utilise la branche par défaut si aucune référence de source n'est spécifiée.

      iv. Dans Exclure les modèles de fichiers, entrez les modèles de fichiers pour exclure des dossiers, des fichiers ou des extensions spécifiques. Cela permet de gérer la taille des données lors de l'importation de fichiers de référentiel. Il y a un maximum de 50 modèles, et les modèles doivent suivre la syntaxe du modèle global. Par exemple :

         A. `tests/`

         B. `*.jpeg`

         C. `large_data.zip`

   b. Pour Sélectionner le dossier de définition depuis S3 :

      i. Entrez l'emplacement Amazon S3 qui contient le dossier de définition du flux de travail compressé. Le compartiment Amazon S3 doit se trouver dans la même région que le flux de travail.

ii. Si votre compte ne possède pas le compartiment Amazon S3, entrez l'ID de AWS compte du propriétaire du compartiment dans l'ID de compte du propriétaire du compartiment S3. Ces informations sont nécessaires pour vérifier HealthOmics la propriété du compartiment.

c. Pour Sélectionner le dossier de définition à partir d'une source locale :

i. Entrez l'emplacement du lecteur local du dossier de définition du flux de travail compressé.

3. Chemin du fichier de définition du flux de travail principal (facultatif) : entrez le chemin du fichier depuis le dossier ou le référentiel de définition du flux de travail compressé vers le `main` fichier. Ce paramètre n'est pas obligatoire s'il n'existe qu'un seul fichier dans le dossier de définition du flux de travail ou si le fichier principal est nommé « main ».

7. Dans le panneau du fichier README (facultatif), sélectionnez la source du fichier README et fournissez les informations suivantes :

- Pour Importer depuis un service de référentiel, dans le champ Chemin du fichier README, entrez le chemin du fichier README dans le référentiel.
- Pour Select file from S3, dans le fichier README dans S3, entrez l'URI Amazon S3 du fichier README.
- Pour Sélectionner un fichier à partir d'une source locale : dans README, facultatif, choisissez Choisir un fichier pour sélectionner le fichier Markdown (.md) à télécharger.

8. Dans le panneau de configuration du stockage d'exécution par défaut, indiquez le type de stockage d'exécution par défaut et la capacité pour les exécutions utilisant ce flux de travail :

1. Type de stockage d'exécution : choisissez d'utiliser le stockage statique ou dynamique par défaut pour le stockage d'exécution temporaire. La valeur par défaut est le stockage statique.
2. Capacité de stockage d'exécution (facultatif) : pour le type de stockage d'exécution statique, vous pouvez entrer la quantité de stockage d'exécution par défaut requise pour ce flux de travail. La valeur par défaut de ce paramètre est de 1200 GiB. Vous pouvez remplacer ces valeurs par défaut lorsque vous démarrez une course.

9. Balises (facultatif) : vous pouvez associer jusqu'à 50 balises à cette version du flux de travail.

10. Choisissez Suivant.

11. Sur la page Ajouter des paramètres de flux de travail (facultatif), sélectionnez la source du paramètre :

1. Pour Parse from fichier de définition de flux de travail, HealthOmics analysera automatiquement les paramètres du flux de travail à partir du fichier de définition de flux de travail.
2. Pour Provide parameter template from Git repository, utilisez le chemin d'accès au fichier de modèle de paramètres depuis votre dépôt.
3. Pour Select JSON file from local source, téléchargez un JSON fichier depuis une source locale qui spécifie les paramètres.
4. Pour saisir manuellement les paramètres du flux de travail, entrez manuellement les noms et les descriptions des paramètres.

12. Dans le panneau d'aperçu des paramètres, vous pouvez consulter ou modifier les paramètres de cette version du flux de travail. Si vous restaurez le JSON fichier, vous perdrez toutes les modifications locales que vous avez apportées.
13. Sur la page de remappage des URI du conteneur, dans le panneau Règles de mappage, vous pouvez définir des règles de mappage des URI pour votre flux de travail.

    Pour Source du fichier de mappage, sélectionnez l'une des options suivantes :

    - Aucune : aucune règle de mappage n'est requise.
    - Sélectionnez le fichier JSON dans S3 — Spécifiez l'emplacement S3 du fichier de mappage.
    - Sélectionnez un fichier JSON à partir d'une source locale : spécifiez l'emplacement du fichier de mappage sur votre appareil local.
    - Entrez les mappages manuellement : entrez les mappages de registre et les mappages d'images dans le panneau Mappages.

14. La console affiche le panneau Mappings. Si vous avez choisi un fichier source de mappage, la console affiche les valeurs du fichier.

    a. Dans les mappages de registre, vous pouvez modifier les mappages ou ajouter des mappages (maximum de 20 mappages de registre).

       Chaque mappage de registre contient les champs suivants :

       - URL du registre en amont : URI du registre en amont.
       - Préfixe du référentiel ECR : préfixe du référentiel à utiliser dans le référentiel privé Amazon ECR.

- (Facultatif) Préfixe du référentiel en amont : préfixe du référentiel dans le registre en amont.
- (Facultatif) ID de compte ECR : ID de compte du compte propriétaire de l'image du conteneur en amont.

b. Dans Mappages d'images, vous pouvez modifier les mappages d'images ou ajouter des mappages (maximum de 100 mappages d'images).

Chaque mappage d'image contient les champs suivants :

- Image source — Spécifie l'URI de l'image source dans le registre en amont.
- Image de destination — Spécifie l'URI de l'image correspondante dans le registre Amazon ECR privé.

15. Choisissez Suivant.
16. Vérifiez la configuration de la version, puis choisissez Créer une version.

Lorsque la version est créée, la console revient à la page détaillée du flux de travail et affiche la nouvelle version dans le tableau des flux de travail et des versions.

## Création d'une version de flux de travail à l'aide de la CLI

Vous pouvez créer une version du flux de travail à l'aide de `CreateWorkflowVersion` l'opération API. Pour les paramètres facultatifs, HealthOmics utilise les valeurs par défaut suivantes :

| Paramètre | Par défaut |
| --- | --- |
| Engine | Déterminé à partir de la définition du flux de travail |
| Type de stockage | STATIC |
| Capacité de stockage (pour le stockage statique) | 1200 GiB |
| Principal | Déterminé en fonction du contenu du dossier de définition du flux de travail. Pour en savoir plus, consultez HealthOmics exigences de définition du flux de travail. |

| Paramètre | Par défaut |
| --- | --- |
| Accélérateurs | none |
| Étiquettes | none |

L'exemple de CLI suivant crée une version de flux de travail avec le stockage statique comme stockage d'exécution par défaut :

```
aws omics create-workflow-version \
--workflow-id 1234567  \
--version-name "my_version" \
--engine WDL \
--definition-zip fileb://workflow-crambam.zip \
--description "my version description" \
--main file://workflow-params.json \
--parameter-template file://workflow-params.json \
--storage-type='STATIC'   \
--storage-capacity 1200   \
--tags example123=string  \
--accelerators GPU
```

Si le fichier de définition de votre flux de travail se trouve dans un dossier Amazon S3, entrez l'emplacement en utilisant le `definition-uri` paramètre au lieu de`definition-zip`. Pour plus d'informations, consultez CreateWorkflowVersionle manuel de référence des HealthOmics API AWS.

Vous recevez la réponse suivante à la `create-workflow-version` demande.

```
{
  "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"
}
```

## Création d'une version de flux de travail à l'aide d'un SDK

Vous pouvez créer un flux de travail à l'aide de l'un des SDKs.

L'exemple suivant montre comment créer une version de flux de travail à l'aide du SDK Python

```
import boto3

omics = boto3.client('omics')

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

response = omics.create_workflow_version(
   workflowId='1234567',
   versionName='my_version',
   requestId='my_request_1'
   definitionZip=definition,
   parameterTemplate={ ... }
)
```

## Vérifier le statut d'une version du flux de travail

Après avoir créé la version de votre flux de travail, vous pouvez vérifier le statut et consulter d'autres détails du flux de travail à l'aide get-workflow-versionde ce qui suit :

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

La réponse vous donne les détails de votre flux de travail, y compris le statut, comme indiqué.

```
{
  "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"
}
```

Avant de pouvoir démarrer une exécution avec cette version du flux de travail, le statut doit passer à`ACTIVE`.

## Mettre à jour une version du flux de travail

Vous pouvez mettre à jour la description et la configuration de stockage d'exécution par défaut pour une version de flux de travail privé. Pour modifier toute autre information dans la version du flux de travail, créez une nouvelle version.

Rubriques

- Mettre à jour une version du flux de travail à l'aide de la console
- Mettre à jour une version de flux de travail à l'aide de la CLI
- Mettre à jour une version de flux de travail à l'aide d'un SDK

### Mettre à jour une version du flux de travail à l'aide de la console

Pour mettre à jour une version d'un flux de travail

1. Ouvrez la HealthOmics console.
2. Si nécessaire, ouvrez le volet de navigation de gauche (≡). Choisissez des flux de travail privés.
3. Sur la page Flux de travail privés, choisissez le flux de travail.
4. Sur la page Flux de travail, choisissez la version du flux de travail à mettre à jour, puis sélectionnez Modifier dans la liste Actions.
   - Si vous choisissez la version par défaut, la console ouvre la page Modifier le flux de travail. Pour de plus amples informations, veuillez consulter Mettre à jour un flux de travail privé.
   - Si vous choisissez une version définie par l'utilisateur, la console ouvre la page Modifier la version.
5. Sur la page Modifier la version, fournissez les informations suivantes
   - Description de la version (facultatif) - Description de cette version.
6. Dans le panneau de configuration du stockage par défaut, indiquez les valeurs par défaut suivantes pour les exécutions utilisant cette version du flux de travail. Vous pouvez remplacer les valeurs par défaut lorsque vous démarrez une course :
   - Pour le type de stockage Exécuter, sélectionnez Statique ou Dynamique.

- Pour le stockage statique, sélectionnez la quantité par défaut de capacité de stockage d'exécution pour les exécutions utilisant cette version de flux de travail. La valeur par défaut de ce paramètre est de 1200 GiB.

7. Sélectionnez Enregistrer les modifications.

La console revient à la page détaillée du flux de travail et affiche une bannière de page avec la version mise à jour du flux de travail.

## Mettre à jour une version de flux de travail à l'aide de la CLI

Vous pouvez mettre à jour les paramètres d'une version de flux de travail à l'aide de la commande CLI suivante. La combinaison de l'ID du flux de travail et du nom de version identifie la version de manière unique.

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

Vous ne recevez aucune réponse à cette `update-workflow-version` demande.

## Mettre à jour une version de flux de travail à l'aide d'un SDK

Vous pouvez mettre à jour une version du flux de travail à l'aide de l'un des SDKs. L'exemple de SDK Python suivant montre comment mettre à jour le type de stockage et la description d'une version de flux de travail.

```
import boto3

omics = boto3.client('omics')

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

# Supprimer une version du flux de travail

Vous pouvez supprimer une version de flux de travail définie par l'utilisateur à l'aide de la console, de la CLI ou de l' SDKsune des. La suppression d'une version de flux de travail n'affecte pas les exécutions en cours utilisant cette version de flux de travail.

Vous ne pouvez pas supprimer leVersion du flux de travail par défaut. Vous supprimez toutes les versions définies par l'utilisateur, puis vous supprimez le flux de travail.

Rubriques

- Supprimer une version du flux de travail à l'aide de la console
- Supprimer une version du flux de travail à l'aide de la CLI
- Supprimer une version du flux de travail à l'aide d'un SDK

## Supprimer une version du flux de travail à l'aide de la console

Pour supprimer une version du flux de travail

1. Ouvrez la HealthOmics console.
2. Si nécessaire, ouvrez le volet de navigation de gauche (≡). Choisissez des flux de travail privés.
3. Sur la page Flux de travail privés, choisissez le flux de travail.
4. Sur la page Flux de travail, choisissez la version du flux de travail à supprimer, puis sélectionnez Supprimer dans la liste Actions.
5. Dans le modal Supprimer la version du flux de travail, entrez « confirmer » pour confirmer la suppression.
6. Sélectionnez Delete (Supprimer).

La console affiche une bannière de page avec la version du flux de travail supprimée.

## Supprimer une version du flux de travail à l'aide de la CLI

Vous pouvez supprimer une version de flux de travail définie par l'utilisateur à l'aide de la commande CLI suivante. La combinaison de l'ID du flux de travail et du nom de version identifie la version de manière unique.

```
aws omics delete-workflow-version
```

```
--workflow-id 9876543
--version-name "my_version"
```

Vous ne recevez aucune réponse à cette `delete-workflow-version` demande.

### Supprimer une version du flux de travail à l'aide d'un SDK

Vous pouvez supprimer un flux de travail à l'aide de l'un des SDKs.

L'exemple suivant montre comment supprimer un flux de travail à l'aide du SDK Python.

```
import boto3

omics = boto3.client('omics')

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

## Utiliser des HealthOmics courses

Après avoir créé un flux de travail, vous pouvez démarrer des exécutions à l'aide de ce flux de travail.

Lorsque vous démarrez une exécution, HealthOmics alloue un espace de stockage temporaire au moteur de flux de travail à utiliser pendant l'exécution. Pour garantir l'isolation et la sécurité des données, HealthOmics provisionne le stockage au début de chaque exécution et le déprovisionne à la fin de l'exécution.

HealthOmics fournit plusieurs quotas liés aux exécutions et aux tâches des flux de travail. Les valeurs par défaut sont volontairement prudentes, afin de vous aider à éviter des dépassements de coûts imprévus. Vous pouvez demander une augmentation de ces quotas. Pour de plus amples informations, veuillez consulter HealthOmics quotas de service.

Lorsque vous lancez une exécution, HealthOmics attribuez un ID d'exécution et un uuid d'exécution à l'exécution. Les courses d'un compte ont un cycle unique IDs. Cependant, il HealthOmics réutilise une exécution supprimée IDs, de sorte qu'une exécution et une exécution supprimée peuvent avoir le même ID d'exécution. En outre, il est rare mais possible qu'un flux de travail partagé ait le même identifiant d'exécution qu'un flux de travail dans votre compte.

run uuidII s'agit d'un identifiant global unique (GUID) que vous pouvez utiliser pour identifier les essais sur plusieurs comptes ou pour faire la distinction entre deux essais de votre compte ayant le même identifiant de course.

> **Note**
>
> Pour des raisons de provenance des données, nous vous recommandons d'utiliser le run uuid pour identifier les séries de manière unique. run uuidII s'agit également du meilleur identifiant à associer à votre système interne de gestion des informations de laboratoire (LIMs) ou à votre système de suivi des échantillons.

Vous pouvez utiliser Amazon Q CLI pour optimiser vos exécutions et analyser les performances d'exécution. Pour plus d'informations, consultez les exemples d'instructions pour Amazon Q CLI et le didacticiel HealthOmics Agentic Generative AI sur. GitHub

Rubriques



## Exécuter les types de stockage dans les HealthOmics flux de travail

Lorsque vous démarrez une exécution, HealthOmics alloue un espace de stockage temporaire au moteur de flux de travail à utiliser pendant l'exécution. HealthOmicsfournit le stockage d'exécution temporaire sous forme de système de fichiers.

Pour un flux de travail ou une exécution de flux de travail donné, vous pouvez choisir le stockage des exécutions dynamiques ou statiques. Par défaut, HealthOmics fournit un stockage d'exécution DYNAMIQUE.

> **Note**
>
> L'utilisation de l'espace de stockage entraîne des frais sur votre compte. Pour obtenir des informations sur les tarifs relatifs au stockage statique et dynamique, consultez la section HealthOmicsTarification.

Les sections suivantes fournissent des informations à prendre en compte lors du choix du type de stockage d'exécution à utiliser.

## Stockage dynamique

Nous recommandons d'utiliser le stockage dynamique pour la plupart des exécutions, y compris les exécutions qui nécessitent des temps de démarrage plus rapides, les exécutions pour lesquelles vous ne connaissez pas à l'avance les besoins de stockage et pour les cycles de test de développement itératifs.

Il n'est pas nécessaire d'estimer le stockage ou le débit requis pour l'exécution. HealthOmics augmente ou diminue dynamiquement la taille du stockage, en fonction de l'utilisation du système de fichiers pendant l'exécution. HealthOmics adapte également le débit de manière dynamique en fonction des besoins du flux de travail. Une exécution n'échoue jamais en raison d'une erreur de stockage insuffisant pour le système de fichiers.

Le stockage à exécution dynamique permet de réduire le provisioning/deprovisioning temps de stockage par exécution par rapport au stockage à exécution statique. Une configuration plus rapide est un avantage pour la plupart des flux de travail, mais également pendant development/test les cycles.

Une fois l'exécution terminée (chemin de réussite ou chemin d'échec), l'opération d'API GetRun renvoie le stockage maximal utilisé par l'exécution dans le champ StorageCapacity. Vous pouvez également trouver ces informations dans les journaux du manifeste d'exécution situés dans le groupe de omics journaux. Pour une exécution de stockage dynamique qui se termine dans les 2 heures, la valeur de stockage maximale peut ne pas être disponible.

Pour le stockage dynamique, l'exécution approvisionne un système de fichiers utilisant le protocole NFS. NFS considère les opérations CREATE, DELETE et RENAME comme non idempotentes, ce qui peut parfois créer des conditions de concurrence pour ces opérations que votre code doit gérer correctement. Par exemple, votre code ne doit pas échouer s'il tente de supprimer un fichier qui n'existe pas. Avant d'adopter le stockage à exécution dynamique, nous vous recommandons d'ajuster

le code de votre flux de travail pour le rendre résilient aux opérations de fichiers non idempotentes. Consultez Exemples de code pour une gestion sûre des opérations non idempotentes.

### Exemples de code pour une gestion sûre des opérations non idempotentes

L'exemple python suivant montre comment supprimer un fichier sans échec s'il n'existe pas.

```
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")
```

Les exemples suivants utilisent le shell Bash. Pour supprimer un fichier en toute sécurité même s'il n'existe pas, utilisez :

```
rm -f my_file
```

Pour déplacer (renommer) un fichier en toute sécurité, exécutez la commande move uniquement si le fichier `old_name` existe dans le répertoire actuel.

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

Pour créer un répertoire, utilisez la commande suivante :

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

### Stockage statique

Pour le stockage statique, l'exécution approvisionne un système de fichiers utilisant le protocole Lustre. Ce protocole est résilient aux opérations de fichiers non idempotentes par défaut. Il n'est

pas nécessaire d'ajuster le code de votre flux de travail pour gérer les opérations de fichiers non idempotentes.

HealthOmics alloue une quantité fixe de stockage d'exécution. Vous spécifiez cette valeur lorsque vous démarrez l'exécution. Le stockage d'exécution par défaut est de 1200 GiB, si vous ne spécifiez aucune valeur. Lorsque vous spécifiez une valeur pour la taille de stockage dans la demande d' StartRun API, le système arrondit la valeur au multiple le plus proche de 1 200 GiB. Si cette taille de stockage n'est pas disponible, elle est arrondie au multiple le plus proche de 2 400 GiB.

Pour le stockage statique, HealthOmics fournit les valeurs de débit suivantes :

- Débit de référence de 200 MB/s par TiB de capacité de stockage provisionnée.
- Débit en rafale jusqu'à 1 300 MB/s par TiB de capacité de stockage provisionnée.

Si la taille de stockage spécifiée est trop faible, l'exécution échoue avec un message d'erreur « Stockage insuffisant pour le système de fichiers ». Le stockage statique convient parfaitement aux flux de travail prévisibles dont les exigences de stockage sont connues.

Le stockage statique convient aux charges de travail volumineuses et surchargées avec une grande simultanéité des tâches (par exemple, un grand volume d' RNASeq échantillons traités en parallèle). Il fournit un débit de système de fichiers supérieur par GiB et un coût par GiB inférieur à celui du stockage à exécution dynamique.

## Calcul de l'espace de stockage statique requis

Un flux de travail nécessite une capacité supplémentaire lorsqu'il utilise un stockage d'exécution statique (par rapport au stockage d'exécution dynamique), car l'installation du système de fichiers de base utilise 7 % de la capacité du système de fichiers statique.

Si vous exécutez un flux de travail de stockage dynamique pour mesurer le stockage maximal utilisé par l'exécution, utilisez le calcul suivant pour déterminer la quantité minimale de stockage statique requise :

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

Par exemple :

```
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.
```

Par conséquent, 1 200 Go (capacité minimale pour le stockage statique) sont suffisants pour cette exécution.

## Exécuter le mode de rétention pour les HealthOmics courses

Une fois l'exécution terminée, HealthOmics archive les métadonnées de l'exécution dans CloudWatch. Par défaut, CloudWatch conserve les données d'exécution indéfiniment, sauf si vous modifiez la politique de CloudWatch conservation. Les sorties d'exécution sont également stockées dans Amazon S3 jusqu'à ce que vous les supprimiez.

L'un des réglages HealthOmics quotas de service est celui maximum number of runs (active and inactive) dans une région. HealthOmics conserve les métadonnées d'exécution jusqu'à ce nombre d'exécutions pour les utiliser par la console et les opérations d'API (ListRuns et GetRun). Lorsque vous démarrez une exécution, vous pouvez définir le paramètre du mode de rétention d'exécution pour indiquer le comportement de rétention pour l'exécution. Le paramètre prend en charge les valeurs REMOVE et RETAIN.

Pour une nouvelle exécution avec le mode de rétention défini sur REMOVE, HealthOmics si vous essayez d'ajouter l'exécution après avoir déjà enregistré le nombre maximum d'exécutions, les métadonnées de la plus ancienne exécution pour laquelle le mode REMOVE est défini sont automatiquement supprimées. Cette suppression n'affecte pas les données stockées dans CloudWatch ou Amazon S3.

RETAIN est la valeur par défaut pour exécuter le mode de rétention. Pour les exécutions dans ce mode, le système ne supprime pas les métadonnées d'exécution. Si le nombre maximum d'essais HealthOmics atteint, tous définis sur RETAIN, vous ne pourrez pas créer d'essais supplémentaires tant que vous n'en aurez pas supprimé certains.

Si vous prévoyez d'exécuter un lot dont le nombre d'exécutions est supérieur au nombre maximal d'exécutions en même temps, assurez-vous de définir le mode de rétention des exécutions sur REMOVE. Dans le cas contraire, le lot échoue lorsqu'il HealthOmics essaie de démarrer le cycle suivant après le maximum.

Considérations supplémentaires relatives à l'utilisation du mode de rétention REMOVE :

- Lorsque vous commencez à utiliser REMOVE comme mode de rétention, pensez à supprimer une ou plusieurs exécutions utilisant le mode RETAIN, afin de libérer des emplacements. Lorsque vous lancez des sessions REMOVE supplémentaires, la suppression automatique prend le relais, de sorte que suffisamment d'emplacements sont disponibles pour de nouvelles exécutions.
- Si vous souhaitez réexécuter une exécution archivée (ou un ensemble d'exécutions), utilisez l'outil de HealthOmics réexécution de la CLI. Pour plus d'informations et des exemples d'utilisation de cet outil, consultez la section Réexécution d'Omics dans le GitHub référentiel HealthOmics d'outils.
- Nous vous recommandons de configurer un nom unique pour chaque exécution. Après HealthOmics avoir supprimé une exécution, vous ne pouvez pas utiliser la console ou l'API pour trouver le nom ou l'ID d'exécution. Cependant, vous pouvez l'utiliser CloudWatch pour rechercher le nom de l'exécution. Utilisez donc des noms uniques pour obtenir les meilleurs résultats de recherche.
- Vous pouvez utiliser la CloudWatch start-query commande pour obtenir des informations sur une exécution archivée. Si le nom d'exécution n'est pas unique, la requête peut renvoyer plusieurs manifestes. Les paramètres d'heure de début et de fin définissent la plage de temps pour la recherche.

```
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>
```

  La start-query commande renvoie un ID de requête. La transmission de l'ID de requête à la get-query-results commande renvoie les résultats de la requête.

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

## HealthOmics exécuter les entrées

Si la définition du flux de travail spécifie des fichiers d'entrée pour le flux de travail ou les tâches HealthOmics du flux de travail, place les fichiers dans un volume temporaire dédié à l'exécution du flux de travail. Ces fichiers d'entrée sont en lecture seule, ce qui empêche les tâches de modifier les entrées potentielles pour d'autres tâches du flux de travail. Pour les importations de répertoires, les répertoires sont également en lecture seule.

De nombreuses applications génomiques supposent que les fichiers d'index sont situés au même endroit que les fichiers de séquence (comme un `bai` fichier d'accompagnement pour un `bam` fichier). Pour inclure des fichiers d'index, spécifiez-les en tant qu'entrées de tâche dans la définition du flux de travail.

Rubriques



## Gestion de la taille des paramètres d'exécution

Lorsque vous lancez une exécution, vous spécifiez les entrées d'exécution dans l'objet ou le fichier JSON des paramètres d'exécution. Vous pouvez spécifier jusqu'à 50 Ko de paramètres d'exécution pour le flux de travail. Vous pouvez utiliser les techniques suivantes pour respecter cette contrainte de taille :

- Utiliser les importations d'annuaires

  Pour spécifier un grand nombre de fichiers d'entrée, spécifiez un paramètre comme emplacement Amazon S3 contenant tous les fichiers, plutôt que de spécifier un paramètre pour chaque emplacement de fichier. Pour plus d'informations, consultez la rubrique suivante (formats de paramètres d'entrée Amazon S3).
- Utiliser une feuille d'exemple

  Une feuille d'exemple est un fichier CSV ou TSV comportant une colonne pour l'adresse fastq.gz (ou deux pour la lecture couplée) et des colonnes supplémentaires pour les métadonnées telles que les noms d'échantillons. Vous spécifiez la feuille d'exemple comme paramètre d'entrée d'exécution plutôt que comme paramètre pour chaque fichier d'entrée.

  Votre flux de travail définit la façon dont votre feuille d'exemple correspond aux structures de données du flux de travail. Bien que vous puissiez écrire du code pour des feuilles d'exemples dans WDL et CWL, elles sont plus courantes dans. NextFlow Pour un exemple, voir la feuille d'exemple sur le site nf-core GitHub .

## Formats de paramètres d'entrée Amazon S3

Pour un paramètre d'entrée qui accepte un emplacement Amazon S3, le paramètre peut spécifier l'emplacement d'un fichier ou d'un répertoire complet de fichiers. L'utilisation d'un annuaire présente les avantages suivants :

- Commodité — Vous spécifiez le nom du répertoire en tant que paramètre. Vous ne listez pas chaque nom de fichier.
- Compacité — La taille de fichier maximale des paramètres d'entrée est de 50 Ko. Si vous fournissez une longue liste de noms de fichiers d'entrée, vous pouvez dépasser ce maximum.

Amazon S3 est un système de stockage d'objets plat, il ne prend donc pas en charge les annuaires. Vous regroupez les fichiers dans un « répertoire » en attribuant à chaque fichier le même préfixe de clé d'objet. Pour plus d'informations sur les préfixes de clé d'objet Amazon S3, consultez Organisation des objets à l'aide de préfixes.

HealthOmics interprète la valeur du paramètre d'entrée comme suit :

- Si l'emplacement Amazon S3 ne se termine pas par une barre oblique ou n'utilise pas le modèle global, HealthOmics attendez-vous à ce que la valeur du paramètre soit la clé d'un objet Amazon S3.

  Par exemple, vous spécifiez de `s3://myfiles/runs/inputs/a/file1.fastq` saisir file1.fastq
- Si l'emplacement Amazon S3 se termine par une barre oblique, HealthOmics interprète la valeur du paramètre comme un préfixe Amazon S3. Il charge tous les objets Amazon S3 avec ce préfixe.

  Par exemple, vous pouvez spécifier `s3://myfiles/runs/inputs/a/` de charger tous les objets dont les clés commencent par ce préfixe.
- Pour Nextflow, prend HealthOmics partiellement en charge le modèle global d'Amazon S3 URIs dans les paramètres d'entrée.

  Par exemple, vous pouvez spécifier de `"s3://myfiles/runs/inputs/a/*.gz"` saisir tous les fichiers .gz dont les clés commencent par ce préfixe.

Nextflow Gestion du modèle Glob dans les entrées Amazon S3

| Motif Glob | HealthOmics Comportement du match | Remarques |
| --- | --- | --- |
| s3://bucket/directory/*.txt | Correspond à tous les `.txt` objets, quelle que soit leur profondeur, sous le préfixe s3 ://bucket/directory/. For example, matches s3://bucket/directory/abc.txt or s3://bucket/directory/subDir/123.txt, etc. | |
| s3://bucket/directory/**/*.txt | Correspond à tous les `.txt` objets, quelle que soit leur profondeur, sous le préfixe s3 ://bucket/directory/. For example, matches s3://bucket/directory/abc.txt or s3://bucket/directory/subDir/123.txt, etc. | Dans S3, `**` est équivalent à`*`. |
| s3://bucket/directory/{a,b}.txt | s3 ://bucket/directory/a.txt, s3://bucket/directory/b.txt | |
| s3://bucket/directory/ ?. txt | Fait correspondre les objets situés à la racine du préfixe dont le nom de fichier est un seul caractère suivi `.txt` de. Par exemple, il correspond à s3 ://bucket/directory/a.txt but not s3://bucket/directory/someDir/a.txt or s3://bucket/directory/someDir/subDir/a.txt | |
| s3://bucket/directory/[0-9].txt | s3 ://bucket/directory/0.txt, s3://bucket/directory/1.txt , ... ,s3://bucket/directory/9.txt | |

| Motif Glob | HealthOmics Comportement du match | Remarques |
| --- | --- | --- |
| s3://bucket/directory/[0-9].txt | s3 ://bucket/directory/1.txt, s3://bucket/directory/2.txt, s3://bucket/directory/3.txt | |
| s3://bucket/directory/[0-9].txt | s3 ://bucket/directory/b.txt, s3://bucket/directory/c.txt , ... ,s3://bucket/directory/Y.txt | |

Gestion spécifique à la langue de la double barre oblique dans les entrées Amazon S3

HealthOmics conserve le comportement natif de chaque moteur de flux de travail lors de la gestion des barres obliques doubles dans Amazon S3 URIs, de sorte que vous n'avez pas besoin de modifier vos flux de travail lorsque vous les migrez vers. HealthOmics Les sections suivantes décrivent comment chaque moteur gère différents scénarios.

WDL

Si le paramètre d'entrée inclut une double barre oblique au milieu ou à la fin de l'URI, le moteur WDL conserve la double barre oblique.

| Paramètre d'entrée | Emplacement attendu | |
| --- | --- | --- |
| s3 ://myfiles/runs/inputs//file1.fastq | s3 ://myfiles/runs/inputs//file1.fastq | |
| s3 ://myfiles/runs/inputs// | s3 ://myfiles/runs/inputs// | |

Flux suivant

Si le paramètre d'entrée inclut une double barre oblique au milieu de l'URI, le moteur Nextflow conserve la double barre oblique. Pour une double barre oblique à la fin de l'URI, le moteur Nextflow la résout en une seule barre oblique.

| Paramètre d'entrée | Emplacement attendu | |
|---|---|---|
| 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

Si le paramètre d'entrée inclut une double barre oblique au milieu ou à la fin de l'URI, le moteur CWL conserve la double barre oblique.

| Paramètre d'entrée | Emplacement attendu | |
|---|---|---|
| s3://myfiles// runs/inputs//file1.fastq | s3://myfiles// runs/inputs//file1.fastq | |
| s3://myfiles//runs/inputs// | s3://myfiles//runs/inputs// | |

### États de l'archive d'entrée Amazon S3

HealthOmics peut récupérer des objets Amazon S3 fournis par S3 en temps réel. Pour les objets dont l'état de stockage archivé est le suivant, restore les objets auxquels ils doivent être mis à disposition HealthOmics :

- Classes de stockage Flexible Retrieval ou Deep Archive dans Amazon S3 Glacier.
- Archived Access ou Deep Archive Access propose des niveaux dans la hiérarchisation intelligente.

Pour plus d'informations sur la restauration d'objets, consultez la section Restauration d'un objet archivé dans le guide de l'utilisateur Amazon S3.

## Exécuter le cycle de vie dans un HealthOmics flux de travail

Vous pouvez suivre la progression d'une course en surveillant son état. HealthOmics met à jour l'état de l'exécution au fur et à mesure de son cycle de vie.

Vous pouvez récupérer le statut d'exécution à l'aide de l'une des méthodes suivantes :

- La HealthOmics console affiche le statut de chaque exécution sur la Runs page.
- L'opération GetRun d'API renvoie l'état d'exécution actuel.
- Vous pouvez surveiller l'état de l'exécution à l'aide d' EventBridge événements. Pour de plus amples informations, veuillez consulter Utilisation EventBridge avec AWS HealthOmics.

Rubriques



## Exécuter les valeurs d'état

Lorsque vous lancez une exécution, HealthOmics définit le statut d'exécution surPending. Au fur et à mesure que l'exécution avance dans son cycle de vie, la valeur du statut est mise à HealthOmics jour pour refléter sa progression actuelle.

> **Note**
>
> Aucuns frais ne vous seront facturés lors d'une course autre que celle en cours d'exécution. Pour en savoir plus, consultez la section suivante.

HealthOmics prend en charge les valeurs d'état d'exécution suivantes :

En attente

La course est dans la file d'attente et attend de commencer. Les exécutions restent généralement en attente pendant une brève période avant de commencer.

- Les essais peuvent rester en attente plus longtemps si vous soumettez plusieurs jobs en même temps.
- Les essais restent en attente une fois que votre compte a atteint le nombre maximum d'essais simultanés.
- Une exécution reste en attente si elle fait partie d'un groupe d'exécutions qui a atteint l'une de ses valeurs maximales de ressources.

- Vous pouvez ajuster les priorités d'exécution afin que certaines séries en file d'attente commencent avant les autres. Pour plus d'informations sur la priorité d'exécution, consultezPriorité d'exécution.

Démarrage en cours

HealthOmics crée l'exécution et provisionne les ressources nécessaires à l'exécution (telles que le stockage temporaire des exécutions et le nœud du moteur).

- HealthOmics provisionne le stockage d'exécution temporaire au début de l'exécution et déprovisionne le stockage d'exécution lorsque l'exécution s'arrête.

En cours d'exécution

Une exécution reste en cours d'exécution pendant le processus d'importation, le traitement de chaque tâche et le processus d'exportation.

- HealthOmics importe les fichiers d'entrée dans le système de fichiers de stockage temporaire. Les fichiers d'entrée sont en lecture seule, afin d'empêcher les tâches de modifier les entrées des autres tâches d'un flux de travail.
- Lors de l'exportation de fichiers, HealthOmics exporte les fichiers de sortie du système de fichiers de stockage exécuté vers l'emplacement S3.
- HealthOmics fournit les journaux d'exécution et les journaux de tâches CloudWatch en temps réel lorsque l'état d'exécution est En cours d'exécution. Pour de plus amples informations, veuillez consulter Se connecte CloudWatch.

Arrêt en cours

Une fois le processus d'exportation terminé, l'exécution passe à l'état d'arrêt.

- HealthOmics déprovisionne toutes les ressources (y compris le système de fichiers de stockage exécuté et le nœud du moteur).

Terminé

L'exécution passe à Terminé une fois le déprovisionnement des ressources HealthOmics terminé.

- HealthOmics a terminé toutes les tâches exécutées et exporté les données de sortie sans erreur.
- Les sorties d'exécution sont disponibles dans l'emplacement de sortie URI Amazon S3 spécifié. Pour WDL et CWL, HealthOmics génère un fichier récapitulatif de sortie d'exécution, qui fournit des informations sur le. HealthOmics exécuter les sorties
- Les journaux du manifeste d'exécution final et les journaux du moteur (le cas échéant) sont disponibles dans CloudWatch.

- Pour les exécutions qui prennent en charge de nouvelles tentatives de tâches, une exécution avec le statut Terminé peut inclure une ou plusieurs tâches qui ont échoué. Tant qu'une nouvelle tentative de tâche a réussi pour chaque tâche ayant échoué, fait passer HealthOmics l'exécution à Terminé. HealthOmics attribue un nouvel ID de tâche à chaque nouvelle tentative, de sorte que l'exécution inclut une tâche IDs pour les tentatives infructueuses et pour la tentative terminée.

Échec

HealthOmics a rencontré une ou plusieurs erreurs et n'a pas réussi à terminer toutes les tâches exécutées.

- Une exécution échouée passe à l'état d'arrêt pendant le HealthOmics déprovisionnement des ressources.

Annulée

Un utilisateur a lancé une demande d'annulation de l'exécution.

- HealthOmics arrête toutes les tâches en cours d'exécution et déprovisionne toutes les ressources.
- HealthOmics n'exporte aucune donnée de sortie d'exécution lorsqu'un utilisateur annule une exécution. Vous n'avez accès à aucun fichier intermédiaire en cas d'annulation d'une exécution.
- Votre compte est débité pour les tâches et les ressources consommées par l'exécution pendant qu'elle était en cours d'exécution avant l'annulation.
- Aucuns frais ne sont facturés si vous annulez une course en attente ou en cours.

## Nouvelles tentatives de tâches

HealthOmics prend en charge les nouvelles tentatives pour les tâches qui échouent en raison d'erreurs de service (codes d'état HTTP 5XX).

Si toutes les tâches de l'exécution finissent par se terminer, même si elles ont nécessité de nouvelles tentatives, HealthOmics passe l'exécution à Terminée. HealthOmics attribue un nouvel ID de tâche à chaque nouvelle tentative, de sorte que l'exécution inclut une tâche IDs pour les tentatives infructueuses et pour la tentative terminée.

Le comportement de nouvelle tentative par défaut dépend de la langue de définition utilisée par le flux de travail. La valeur par défaut pour Nextflow est de ne pas réessayer. Pour WDL et CWL, vous pouvez HealthOmics tenter jusqu'à deux tentatives d'une tâche ayant échoué, mais vous pouvez désactiver la rétentative pour des tâches spécifiques ou pour toutes les tâches d'un flux de travail.

Une nouvelle tentative de tâche est utile pour corriger les erreurs de service intermittentes. Toutefois, vous pouvez envisager de désactiver une tâche idempotente.

Pour obtenir des informations spécifiques sur chaque langage de définition de flux de travail, consultez les rubriques suivantes :

- WDL — Configurez le comportement des nouvelles tentatives de tâche dans la définition du flux de travail. Reportez-vous à la section Configurer le comportement des nouvelles tentatives d'une tâche WDL.
- Nextflow — Configurez le comportement des nouvelles tentatives de tâche dans le fichier de configuration Nextflow ou dans la définition du flux de travail. Voir Configurer le comportement de nouvelle tentative des tâches Nextflow.
- CWL — Configurez le comportement des nouvelles tentatives de tâche dans la définition du flux de travail. Voir Configurer le comportement des nouvelles tentatives d'une tâche CWL.

## Incidences tarifaires du statut de course

Votre compte peut être débité tant que le statut de course est en cours. Aucun frais ne vous sera facturé lors d'un autre statut de course. Par exemple, les ressources ne sont pas facturées lorsque l'exécution démarre ou s'arrête.

Une course avec le statut En cours a les conséquences de facturation suivantes :

- Votre compte est facturé pour l'utilisation du système de fichiers de stockage lorsque le statut d'exécution est En cours d'exécution. Pour plus d'informations sur les types de stockage exécutés, voirExécuter les types de stockage dans les HealthOmics flux de travail.
- Votre compte entraîne des frais pour l'exécution des tâches, en fonction des ressources de calcul et de mémoire que vous avez spécifiées pour chaque tâche dans la définition du flux de travail et en fonction de la durée de la tâche. Pour de plus amples informations, veuillez consulter Exigences en matière de calcul et de mémoire pour les HealthOmics tâches.
- Chaque tâche a un seuil de facturation minimum d'une minute. Si vous exécutez une tâche pendant moins d'une minute, des frais vous seront facturés pour la durée minimale d'utilisation d'une minute. Si possible, regroupez les petites tâches pour optimiser les coûts. Le regroupement des tâches réduit également le temps d'exécution en évitant le lancement de plusieurs tâches séquentielles.

Pour plus d'informations sur la HealthOmics tarification, consultez la section HealthOmics Tarification.

# HealthOmics exécuter les sorties

Lorsqu'une exécution WDL ou CWL est terminée, les sorties incluent un fichier récapitulatif des sorties (au format JSON) qui répertorie toutes les sorties produites par l'exécution. Vous pouvez utiliser le fichier récapitulatif de sortie aux fins suivantes :

- Déterminez par programme les fichiers de sortie générés par l'exécution.
- Vérifiez que l'exécution a produit tous les résultats attendus.

Rubriques



## Exécuter le résumé de sortie pour WDL

Lorsqu'une exécution WDL est terminée, HealthOmics crée un fichier récapitulatif de sortie nomméoutput.json.

Pour chaque sortie du flux de travail, il existe une key/value paire correspondante dans le fichier. La clé contient le nom du flux de travail et le nom de sortie au format suivant :`WorkflowName.output_name`. Pour une sortie de fichier, la valeur est un URI S3 pointant vers l'emplacement de sortie dans S3 où le fichier est stocké. Pour une sortie Array [File], la valeur est un tableau de S3 URIs.

L'exemple suivant montre le output.json fichier d'un flux de travail nomméBWAMappingWorkflow.

```
{
  "BWAMappingWorkflow.bam_indexes": [
    "s3://omics-outputs/8886192/out/bam_indexes/0/
pbmc8k_S1_L007_R1_001.sorted.bam.bai",
    "s3://omics-outputs/8886192/out/bam_indexes/1/pbmc8k_S1_L008_R1_001.sorted.bam.bai"
  ],
  "BWAMappingWorkflow.mapping_stats": "s3://omics-outputs/8886192/out/mapping_stats/
genome_mapping_final_stats.txt",
  "BWAMappingWorkflow.merged_bam": "s3://omics-outputs/8886192/out/merged_bam/
genome_mapping.merged.bam",
  "BWAMappingWorkflow.merged_bam_index": "s3://omics-outputs/8886192/out/
merged_bam_index/genome_mapping.merged.bam.bai",
```

```
  "BWAMappingWorkflow.reference_index_tar": "s3://omics-outputs/8886192/out/
reference_index_tar/reference_index.tar",
  "BWAMappingWorkflow.sorted_bams": [
    "s3://omics-outputs/8886192/out/sorted_bams/0/pbmc8k_S1_L007_R1_001.sorted.bam",
    "s3://omics-outputs/8886192/out/sorted_bams/1/pbmc8k_S1_L008_R1_001.sorted.bam"
  ],
  "BWAMappingWorkflow.unmapped_bams": [
    "s3://omics-outputs/8886192/out/unmapped_bams/0/
pbmc8k_S1_L007_R1_001.unmapped.bam",
    "s3://omics-outputs/8886192/out/unmapped_bams/1/pbmc8k_S1_L008_R1_001.unmapped.bam"
  ]
}
```

Si le flux de travail produit des sorties avec des types autres que des fichiers (tels que String, Int, Float ou Bool), la valeur du champ est une primitive JSON. Par exemple :

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

## Exécuter le résumé de sortie pour CWL

Lorsqu'une exécution de CWL est terminée, HealthOmics crée un fichier récapitulatif de sortie nommé outputs.json à l'emplacement suivant :

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

Le fichier récapitulatif des sorties inclut une liste des sorties. Chaque sortie est une key/value paire, dont la clé est le nom de la sortie. La valeur est un objet qui inclut les propriétés suivantes :

- location — Le chemin complet vers le fichier de sortie
- basename — La partie du chemin contenant le nom de fichier
- class — Le type de sortie, qui est généralement un fichier
- size — Taille du fichier en octets

Dans l'exemple suivant, le fichier output.json contient une liste de deux fichiers de sortie.

```
{
```

```
  "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
  }
}
```

## Raisons d'échec de l'exécution

Si une exécution échoue, utilisez l'opération GetRunAPI pour récupérer la raison de l'échec.

Passez en revue la raison de l'échec pour vous aider à résoudre les problèmes liés à l'échec de l'exécution. Le tableau suivant répertorie chaque cause de défaillance ainsi qu'une description de l'erreur.

| Raison de l'échec | Description de l'erreur. |
| --- | --- |
| ÉCHEC DE LA PRISE DE RÔLE | HealthOmics n'est pas autorisé à assumer le rôle. Spécifiez le HealthOmics principal dans la relation de confiance pour le rôle. |
| IMPOSSIBLE DE DÉMARRER UNE ERREUR DE CONTENEUR | Impossible de démarrer la tâche de flux de travail :*name*, id : *ID* container à l'aide de l'image :*image name*. Vérifiez que l'image est valide et réessayez. |
| IMPOSSIBLE DE DÉMARRER UNE ERREUR DE TAILLE DU CONTENEUR | Impossible de démarrer la tâche de flux de travail :*name*, id : *ID* container à l'aide de l'image :*image name*. Assurez-vous que la taille de l'image est inférieure à 45 GiB (95 GiB pour une instance de GPU) et réessayez. |
| ECR_PERMISSION_ERROR | HealthOmics n'est pas autorisé à accéder à l'URI de l'image. Vérifiez que le référentiel privé Amazon ECR existe et qu'il a accordé l'accès au principal du HealthOmics service. |

| Raison de l'échec | Description de l'erreur. |
| --- | --- |
| ECHEC DE L'EXPORTATION | L'exportation a échoué. Vérifiez que le compartiment de sortie existe et que le rôle d'exécution dispose d'une autorisation d'écriture sur le compartiment. |
| SYSTÈME DE FICHIERS OUT_OF_SPACE | Le système de fichiers ne dispose pas de suffisamment d'espace. Augmentez la taille du système de fichiers et réexécutez. |
| ÉCHEC DE LA VÉRIFICAT ION DE L'IMAGE | Impossible de vérifier l'image*image name*. Pour corriger le problème, essayez d'extraire l'image, puis de la transférer à nouveau vers votre référentiel ECR. |
| ECHEC DE L'IMPORTATION | L'importation a échoué. Vérifiez que le fichier d'entrée existe et que le rôle d'exécution peut accéder à l'entrée. |
| RESOURCE_DE STOCKAGE INACTIVE_OMICS | L'URI HealthOmics de stockage n'est pas à l'état ACTIF. Activez le kit de lecture et réessayez. Pour en savoir plus sur l'activat ion des ensembles de lecture, consultezL'activation de la lecture s'installe dans HealthOmics. |
| INPUT_URI_NOT_FOUND | L'URI fourni n'existe pas :*uri*. Vérifiez que le chemin de l'URI existe et que le rôle peut accéder à l'objet. |
| INSTANCE_RESERVATI ON_ECHEC | La capacité d'instance est insuffisante pour terminer l'exécution du flux de travail. Patientez et réessayez d'exécuter le flux de travail. |
| URI D'IMAGE_ECR_INVALI DE | La structure d'URI de l'image Amazon ECR n'est pas valide. Entrez un URI valide et réessayez. |
| VALEUR_RESSOURCE DE TÂCHE INVALIDE | Le GPU, le processeur ou la mémoire requis est soit trop élevé par rapport à la capacité de calcul disponible, soit inférieur à la valeur minimale de 1 pour la tâche*ID*. |
| INPUT_URI_INPUT INVALIDE | La structure de l'URI n'est pas valide*uri*. Vérifiez la structure de l'URI et réessayez. |

| Raison de l'échec | Description de l'erreur. |
| --- | --- |
| RESSOURCE_ENTRÉE_MODIFIÉE | L'URI fourni *uri* a été modifié après le début de l'exécution. Réessayez de courir. |
| ERREUR DE SORTIE DE MÉMOIRE | La mémoire de la tâche *ID* de flux de travail était insuffisante. Augmentez la valeur de mémoire dans la définition du flux de travail et réessayez l'exécution. |
| ÉCHEC DE L'EXÉCUTION DE LA TÂCHE | L'exécution a échoué car la tâche a échoué. Pour corriger l'échec de la tâche, utilisez l'opération GetRunTaskAPI et le flux Amazon CloudWatch Logs. |
| RUN_TIMED_OUT | Expiration du délai d'exécution après *number* quelques minutes. |
| ERREUR DE SERVICE | Une erreur temporaire s'est produite dans le service. Réessayez d'exécuter le flux de travail. |
| DÉLAI D'EXPIRATION DE LA TÂCHE | La tâche *id* a expiré au bout de *number* quelques secondes. |
| TAILLE D'ENTRÉE NON PRISE EN CHARGE | La taille d'entrée totale est trop élevée. Diminuez la taille de saisie et réessayez. |
| WORKFLOW_RUN_FAILED | L'exécution du flux de travail a échoué. Consultez le flux de journal du moteur CloudWatch Logs : *ID* pour corriger la panne. |
| ÉCHEC DU WORKFLOW_VER_VALIDATION_FAILED | HealthOmics ne prend pas en charge la version de Nextflow demandée : *version* --. La dernière version prise en charge est*version*. Modifiez votre version de Nextflow pour une version compatible et réessayez. |
| TYPE_INSTANCE_GPU NON PRIS EN CHARGE | Le type d'instance demandé n'est pas pris en charge dans*Region*. Réessayez l'exécution avec un type d'instance de GPU pris en charge dans cette région. Les types d'instances disponibles sont*GPU instance types*. |

## Conseils pour les essais qui ne répondent pas

Lorsque vous développez de nouveaux flux de travail, des exécutions ou des tâches spécifiques peuvent être « bloquées » ou « bloquées » en cas de problème avec votre code, et les tâches ne parviennent pas à quitter correctement les processus. Cela peut être difficile à résoudre et à attraper, car il est normal que les tâches s'exécutent pendant de longues périodes. Pour éviter et identifier les exécutions qui ne répondent pas, suivez les meilleures pratiques suggérées dans les sections suivantes.

Meilleures pratiques pour éviter les essais sans réponse

- Assurez-vous de fermer tous les fichiers ouverts dans votre code de tâche. L'ouverture d'un trop grand nombre de fichiers peut parfois entraîner des problèmes de thread dans les moteurs de flux de travail.
- Les processus d'arrière-plan créés par une tâche de flux de travail doivent se terminer à la fin de la tâche. Toutefois, si un processus en arrière-plan ne se termine pas correctement, vous devez l'arrêter explicitement dans votre code de tâche.
- Assurez-vous que vos processus ne s'exécutent pas en boucle sans sortir. Cela peut entraîner une absence de réponse et nécessite une modification du code de définition de votre flux de travail pour y remédier.
- Fournissez une allocation de mémoire et de processeur appropriée à vos tâches. Analysez les CloudWatch journaux ou utilisez les Exécuter l'analyseur exécutions réussies de votre flux de travail pour vérifier que vous disposez d'une allocation de calcul optimale. Utilisez le `headroom` paramètre Run Analyzer pour inclure une marge de manœuvre supplémentaire, afin de garantir que les processus disposent de suffisamment de ressources pour être menés à bien. Incluez au moins 5 % d'espace libre dans la mémoire et le processeur alloués, afin de tenir compte des processus du système d'exploitation en arrière-plan.
  - En outre, augmentez la taille de bande passante de l'instance si celle-ci nécessite un débit plus élevé. EC2 Les instances Amazon de moins de 16 V CPUs (taille 4xl ou inférieure) peuvent connaître des pics de débit. Pour plus d'informations sur le débit des EC2 instances Amazon, consultez la section Bande passante d'instance EC2 disponible sur Amazon.
- Assurez-vous d'utiliser la bonne taille de système de fichiers pour vos exécutions. Pour les exécutions qui ne répondent pas et qui utilisent le stockage statique, envisagez d'augmenter l'allocation de stockage pour les exécutions statiques afin d'augmenter le débit d'E/S et la capacité de stockage du système de fichiers. Analysez le manifeste d'exécution pour connaître le stockage maximal du système de fichiers, utilisez l'analyseur d'exécution pour déterminer si l'allocation du système de fichiers doit être augmentée.

Meilleures pratiques pour détecter les courses qui ne répondent pas

- Lorsque vous développez de nouveaux flux de travail, utilisez un groupe d'exécution dont la durée d'exécution maximale est définie pour intercepter le code en fuite. Par exemple, si une exécution doit prendre 1 heure, placez-la dans un groupe d'exécutions qui expire au bout de 2 ou 3 heures (ou d'une période différente en fonction de votre cas d'utilisation) afin de détecter les tâches en cours d'exécution. Appliquez également une zone tampon pour tenir compte de la variation des délais de traitement.
- Configurez une série de groupes d'exécution avec différentes limites d'exécution maximales. Par exemple, vous pouvez attribuer des séries courtes à un groupe d'exécution qui met fin aux séries au bout de quelques heures, et à un groupe de séries longues qui met fin aux séries après quelques jours, en fonction de la durée prévue de votre flux de travail.
- HealthOmics a une limite de service de durée d'exécution maximale par défaut de 604 800 secondes, soit 7 jours, qui est ajustable par le biais d'une demande dans l'outil de quotas. Ne demandez une augmentation de la limite de service de ce quota que si vous avez des courses d'une durée d'environ une semaine. Si vous utilisez à la fois des séries courtes et longues et que vous n'utilisez pas de groupes de séries, envisagez de placer les séries de longue durée dans un compte distinct avec une limite de service de durée maximale d'exécution plus élevée.
- Vérifiez dans les CloudWatch journaux les tâches susceptibles de ne pas répondre. Si une tâche produit normalement des instructions de journal régulières et qu'elle ne le fait pas depuis longtemps, elle est probablement bloquée ou bloquée.

Que faire si vous constatez une course qui ne répond pas

- Annulez la course pour éviter d'encourir des frais supplémentaires.
- Consultez les journaux des tâches pour vérifier si certains processus n'ont pas réussi à se fermer correctement.
- Inspectez les journaux du moteur pour identifier tout comportement anormal du moteur.
- Comparez les journaux des tâches et du moteur de l'exécution sans réponse à ceux des exécutions identiques terminées avec succès. Cela peut aider à identifier les différences susceptibles d'être à l'origine du comportement de non-réponse.
- Si vous ne parvenez pas à déterminer la cause première, soumettez un dossier d'assistance et incluez les éléments suivants :
  - ARN de l'exécution bloquée et ARN d'une exécution identique terminée avec succès.
  - Journaux du moteur (disponibles une fois que l'exécution a été annulée ou échoue)

- Journaux de tâches pour la tâche qui ne répond pas. Nous n'avons pas besoin de journaux de tâches pour toutes les tâches du flux de travail à des fins de dépannage.

## Cycle de vie des tâches en une seule HealthOmics exécution

Une tâche est un processus unique au cours d'une exécution. HealthOmics associe chaque tâche de votre flux de travail à un type d'instance de calcul omique qui correspond le mieux aux ressources requises pour la tâche. Vous spécifiez les ressources requises dans la définition du flux de travail. Pour plus d'informations, voirExigences en matière de calcul et de mémoire pour les HealthOmics tâches.

HealthOmics fournit un stockage d'exécution temporaire pour la tâche à utiliser. HealthOmics copie les fichiers d'entrée des tâches dans le stockage d'exécution temporaire sous forme de fichiers en lecture seule. HealthOmics fournit des liens symboliques permettant à la tâche d'accéder aux fichiers d'entrée depuis le répertoire de travail. La tâche a accès uniquement aux fichiers que vous déclarez dans le fichier de définition du flux de travail.

### Valeurs d'état des tâches

Vous pouvez suivre la progression d'une tâche en surveillant son état. Lorsque vous lancez une exécution, HealthOmics définit le statut de la tâche sur Pending pour chaque tâche de l'exécution. Lorsque la tâche démarre et progresse tout au long de son cycle de vie, HealthOmics met à jour la valeur de statut pour refléter sa progression actuelle.

Vous pouvez récupérer le statut d'une tâche à l'aide de l'une des méthodes suivantes :

- La HealthOmics console affiche le statut de chaque tâche lors d'une exécution sur la Run details page.
- L'opération GetRunTask d'API renvoie le statut de la tâche.
- Vous pouvez surveiller l'état des tâches à l'aide d' EventBridge événements. Pour de plus amples informations, veuillez consulter Utilisation EventBridge avec AWS HealthOmics.

Vous pouvez récupérer le statut actuel d'une tâche à l'aide de l'opération GetRunTask API. La HealthOmics console affiche le statut de chaque tâche lors d'une exécution sur la Run details page.

HealthOmics prend en charge les valeurs d'état des tâches suivantes :

En attente

Votre tâche est dans la file d'attente et attend de démarrer. Les tâches restent en attente pendant une brève période avant de commencer.

- Les tâches restent en attente une fois que votre compte a atteint le nombre maximum de tâches simultanées.
- Les tâches restent en attente si l'exécution fait partie d'un groupe d'exécution ayant atteint l'une de ses valeurs maximales de ressources.
- Vous pouvez ajuster les priorités d'exécution afin que les exécutions spécifiques en file d'attente et leurs tâches démarrent avant les autres exécutions en file d'attente. Pour plus d'informations sur la priorité d'exécution, voir Priorité d'exécution

Démarrage en cours

HealthOmics crée la tâche et fournit les ressources nécessaires à la tâche, telles que le nœud de tâche du flux de travail.

En cours d'exécution

L'état de la tâche HealthOmics est En cours d'exécution pendant le traitement de la tâche.

Arrêt en cours

Une fois le traitement de la tâche terminé et les données de sortie exportées, la tâche passe au mode Arrêt.

- HealthOmics déprovisionne le nœud des tâches du flux de travail.

Terminé

HealthOmics a terminé le traitement de la tâche et a transféré les données de sortie vers le système de fichiers de stockage exécuté.

Échec

HealthOmics a rencontré une erreur lors du traitement de la tâche et ne l'a pas terminée.

- La tâche passe au statut Arrêt (HealthOmics déprovisionne les ressources) puis au statut Échoué.
- Si l'erreur est une erreur de service (code d'état HTTP 5XX) et que le flux de travail prend en charge les nouvelles HealthOmics tentatives pour cette tâche, tente à nouveau de traiter la tâche. HealthOmics attribue un nouvel ID de tâche à la nouvelle tentative.

Annulée

HealthOmics arrête la tâche après une demande d'annulation de l'exécution initiée par l'utilisateur.

- La tâche passe au statut Arrêt (HealthOmics déprovisionne les ressources) puis au statut Annulé.

### Résolution des problèmes de flux de travail

Vous trouverez ci-dessous les meilleures pratiques et les considérations relatives au dépannage de vos tâches.

- Les journaux des tâches reposent sur la tâche `STDOUT` et `STDERR` sont produits par celle-ci. Si l'application utilisée dans la tâche ne produit aucun de ces éléments, il n'y aura pas de journal des tâches. Pour faciliter le débogage, utilisez les applications en `verbose` mode.
- Pour afficher les commandes exécutées dans une tâche ainsi que leurs valeurs interpolées, utilisez la commande `set -x` Bash. Cela peut aider à déterminer si la tâche utilise les entrées correctes et à identifier les domaines dans lesquels des erreurs ont pu empêcher la tâche de s'exécuter comme prévu.
- Utilisez la `echo` commande pour afficher les valeurs des variables vers `STDOUT` ou`STDERR`. Cela vous permet de confirmer qu'ils sont définis comme prévu.
- Utilisez des commandes telles que `ls -l <name_of_input_file>` pour confirmer que les entrées sont présentes et qu'elles ont la taille attendue. Si ce n'est pas le cas, cela peut révéler un problème lié à une tâche précédente produisant des sorties vides en raison d'un bogue.
- Utilisez la commande `df -Ph . | awk 'NR==2 {print $4}'` dans un script de tâches pour déterminer l'espace actuellement disponible pour la tâche et aider à identifier les situations dans lesquelles vous pourriez avoir besoin d'exécuter le flux de travail avec une allocation de stockage supplémentaire.

L'inclusion de l'une des commandes précédentes dans un script de tâche suppose que le conteneur de tâches inclut également ces commandes et qu'elles se trouvent dans l'environnement `path` du conteneur.

## Exécuter l'optimisation pour un HealthOmics flux de travail privé

Vous pouvez optimiser les cycles en fonction du coût total, de la durée d'exécution totale ou d'une combinaison des deux. HealthOmics fournit des données et des outils pour vous aider à prendre des

décisions d'optimisation. L'optimisation d'exécution ne s'applique pas aux flux de travail Ready2Run, car vous n'avez aucun contrôle sur la façon dont le service gère le provisionnement des ressources pour ces flux de travail.

La première étape consiste à comprendre l'utilisation actuelle des ressources des tâches et le coût des tâches en cours d'exécution, puis à appliquer des méthodes pour optimiser le coût d'exécution et les performances.

Rubriques



## Exécuter l'analyseur

HealthOmics fournit un outil open source nommé Run Analyzer. Cet outil extrait les informations d'utilisation des ressources au niveau des tâches pour une exécution et suggère des opportunités d'optimisation en termes de coûts et de performances d'exécution.

> **Note**
>
> Run Analyzer estime le coût des tâches et les économies potentielles en fonction AWS des prix catalogue au moment de l'exécution de l'outil. Évaluez les recommandations d'optimisation et mettez en œuvre celles qui conviennent à vos cas d'utilisation. Testez les optimisations que vous adoptez pour vous assurer qu'elles conviennent à votre course.

Run Analyzer exécute les tâches suivantes :

- Évalue les goulots d'étranglement liés à la mémoire et au calcul.
- Identifie les tâches surprovisionnées en mémoire ou en processeur, et recommande de nouvelles tailles d'instance susceptibles de réduire les coûts.
- Calcule les estimations de coûts pour des tâches individuelles et calcule les économies potentielles si vous appliquez les recommandations.

- Fournit une vue chronologique des tâches afin que vous puissiez vérifier les dépendances entre les tâches et la séquence de traitement. La chronologie vous aide également à identifier les tâches à long terme.
- Fournit des recommandations concernant la taille du système de fichiers pour le stockage d'exécution.
- Vous indique les délais de provisionnement des tâches afin que vous puissiez identifier les zones dans lesquelles les chargements de conteneurs importants peuvent ralentir le temps de provisionnement.
- L'outil inclut un paramètre d'entrée (marge de manœuvre) que vous pouvez utiliser pour contrôler l'agressivité des recommandations d'optimisation.

Les sections suivantes contiennent des suggestions spécifiques pour utiliser Run Analyzer afin d'optimiser les exécutions.

## Déterminer les coûts de fonctionnement

Vous pouvez utiliser les méthodes et directives suivantes pour déterminer les coûts d'exploitation :

- Pour consulter le total des coûts d'exploitation pour une période de facturation, procédez comme suit :

  1. Ouvrez la console Billing and Cost Management et sélectionnez Bills.
  2. Dans Charges par service, développez Omics.
  3. Élargissez la région, puis visualisez le coût de toutes vos exécutions détaillé par type d'instance omics, type de stockage d'exécution et flux de travail Ready2Run.

- Pour générer un rapport de coûts incluant des informations pour chaque cycle, procédez comme suit :

  1. Ouvrez la console Billing and Cost Management et choisissez Data Exports.
  2. Choisissez Créer pour créer une nouvelle exportation de données.
  3. Entrez un nom d'exportation pour l'exportation de données. Conservez les valeurs par défaut des autres champs pour créer un rapport CUR (coût et utilisation).
  4. Pour la granularité horaire, sélectionnez horaire ou quotidien.
  5. Sous Paramètres de stockage des exportations de données, effectuez les étapes de configuration suivantes :

a. Configurez un compartiment Amazon S3 pour l'exportation des données.

b. Pour le contrôle des versions de fichiers, indiquez si vous souhaitez remplacer le fichier d'exportation existant ou créer un nouveau fichier à chaque fois.

Le système génère le premier rapport dans les 24 heures qui suivent et les rapports suivants une fois par jour.

6. Pour plus d'informations sur la création de l'exportation de données, voir Création d'exportations de données dans le Guide de l'utilisateur AWS des exportations de données.

- Vous pouvez étiqueter vos courses pour surveiller et optimiser les coûts par catégorie, par exemple par équipe ou par projet. Si vous utilisez des balises, procédez comme suit pour afficher les coûts d'exploitation par catégorie de balises :

  1. Ouvrez la console Billing and Cost Management et choisissez Cost Explorer.
  2. Dans Paramètres du rapport > Regrouper par, choisissez Tag comme dimension et sélectionnez le nom de balise souhaité.

- Pour connaître l'utilisation des ressources pour les tâches, consultez le manifeste d'exécution pour vous connecter CloudWatch. Pour de plus amples informations, veuillez consulter Surveillance à HealthOmics l'aide de CloudWatch journaux.
- Utilisez l'Exécuter l'analyseuroutil pour extraire les informations d'utilisation des ressources d'une tâche pour une exécution.

## Déterminer l'utilisation du temps d'exécution

Vous pouvez utiliser les méthodes suivantes pour étudier l'utilisation du temps d'exécution :

- Sur la page Exécutions de la console, vous pouvez consulter le temps d'exécution total d'une exécution.
- Sur la page Détails de l'exécution, vous pouvez consulter les éléments suivants :
  - Afficher la durée totale d'une exécution.
  - Affichez le temps d'exécution de chaque tâche en cours d'exécution.
  - Choisissez l'un des liens pour afficher les journaux dans Amazon S3, ou pour consulter les journaux d'exécution ou les journaux d'exécution du manifeste CloudWatch.
- Dans la liste Exécuter les tâches, cliquez sur le lien Afficher les journaux d'une tâche pour afficher les connexions de la tâche CloudWatch.

- La réponse à l'opération d'`listRuns`API inclut l'heure de début et l'heure de fin de l'exécution, afin que vous puissiez calculer la durée d'exécution totale.
- L'Exécuter l'analyseuroutil affiche la durée des tâches sur une vue chronologique. Cet outil fournit une représentation visuelle de la séquence de traitement des tâches, que vous pouvez faire correspondre à l'ordre attendu.

## Méthodes pour optimiser les courses

HealthOmics provisionne, gère et optimise automatiquement les ressources qui assurent le transfert des données (telles que les importations et les exportations de données). HealthOmics démarre et exécute également le moteur de flux de travail pour votre flux de travail. Cependant, vous pouvez influencer les heures de début d'exécution, les heures de début des tâches et la durée globale d'exécution des tâches en définissant différentes configurations d'exécution. Votre approche globale de la définition et de la conception du flux de travail a également un impact sur le temps d'exécution des tâches. La liste suivante décrit les facteurs susceptibles d'affecter les performances d'exécution et de tâche :

Type de stockage d'exécution

Le type de stockage d'exécution a un impact sur les performances d'exécution et le temps de provisionnement de l'exécution. Le stockage à exécution dynamique s'approvisionne plus rapidement et ne manque jamais de mémoire, car il évolue de manière dynamique en fonction de vos besoins en stockage d'exécution. Le stockage dynamique convient également aux flux de travail en cours de développement, dans lesquels vous pouvez souvent démarrer et arrêter un flux de travail pour résoudre des problèmes.

Le stockage à exécution statique nécessite des temps de provisionnement du système de fichiers plus longs, mais peut effectuer certaines exécutions plus rapidement, généralement si les exécutions comportent une grande simultanéité des tâches ou nécessitent une capacité de système de fichiers supérieure à 9,6 TiB. Le stockage statique convient parfaitement aux flux de travail de longue durée soumis à des I/O exigences élevées.

Pour vous aider à évaluer le coût par rapport aux performances de chaque type de stockage exécuté pour un cycle donné, vous pouvez effectuer des tests A/B pour déterminer quel type de stockage d'exécution offre les meilleures performances. Pensez également à utiliser le stockage dynamique pour vos cycles de développement, puis à utiliser le stockage statique pour les cycles de production à grande échelle.

Pour plus d'informations sur les types de stockage d'exécution Exécuter les types de stockage dans les HealthOmics flux de travail

Surprovisionner le stockage statique

Si le calcul des tâches de votre flux de travail est limité par des goulots d'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/Oétranglement, l'augmentation de la taille du système de fichiers pour réduire le temps d'exécution des tâches peut réduire le coût global.

Réduire la taille des images des conteneurs

Lorsque chaque tâche démarre, HealthOmics charge le conteneur que vous avez spécifié pour la tâche. Les grands conteneurs prennent plus de temps à charger. Optimisez vos conteneurs pour qu'ils soient aussi petits que possible afin d'améliorer l'efficacité du lancement de nouvelles tâches. Si vous ajoutez de grands ensembles de données à vos conteneurs, envisagez de les stocker dans S3 et de demander à votre flux de travail d'importer les données depuis S3. Pour les tailles maximales de conteneurs HealthOmics compatibles, voirHealthOmics quotas de taille fixe du flux de travail.

Taille de la tâche

Vous pouvez combiner de petites tâches séquentielles en une seule tâche afin de gagner du temps dans le provisionnement des tâches. De plus, la durée minimale des tâches HealthOmics est facturée d'une minute, de sorte que la combinaison des tâches peut réduire les coûts. Dans le cadre de la tâche combinée, vous pourrez peut-être utiliser des canaux Unix pour éviter les I/O coûts liés à la sérialisation et à la désérialisation des fichiers.

Compression de fichiers

Évitez de trop compresser les fichiers intermédiaires du flux de travail. La plupart des formats génomiques utilisent la compression « gzip » ou « block gzip ». La décompression du fichier d'entrée de tâche et la recompression du fichier de sortie de tâche peuvent consommer un pourcentage important de l'utilisation totale du processeur de la tâche. Certaines applications de génomique vous permettent de définir le niveau de compression lors de la sérialisation des sorties. En réduisant le niveau de compression, vous pouvez réduire le temps du processeur, même si des fichiers plus volumineux augmentent le temps passé à écrire sur le disque.
En fonction de la tâche et de l'application, vous pouvez trouver le niveau de compression optimal pour les fichiers intermédiaires dont le temps d'exécution est le plus court. Nous vous recommandons de commencer par cibler les tâches dont les fichiers de sortie sont les plus

volumineux. Un niveau de compression de 2 fonctionne bien pour plusieurs scénarios. Vous pouvez commencer par ce niveau pour votre cas d'utilisation et comparer les résultats en essayant d'autres niveaux de compression.

Nombre de fils

Si vous spécifiez des threads dans votre définition de tâche, définissez le nombre de threads sur la même valeur que le nombre de threads demandésCPUs.

Spécifier le calcul et la mémoire

Si vous ne spécifiez pas de mémoire ou de ressources de calcul dans votre tâche, HealthOmics attribuez le type d'instance le plus petit (`omics.c.large`) par défaut. Déclarez explicitement vos besoins en mémoire et en calcul si vous HealthOmics souhaitez attribuer un type d'instance plus important.

HealthOmics alloue le nombre de ressources vCPUs, de mémoire et de GPU que vous demandez. Par exemple, si vous demandez 15 V CPUs et 33 Go, HealthOmics alloue une instance omics.m.4xl (16 V, 64 Go) à votre tâcheCPUs, mais celle-ci ne peut utiliser que 15 V et 33 Go. CPUs Par conséquent, nous vous recommandons de demander des ressources v CPUs et mémoire qui correspondent à une instance omics.

Batch de plusieurs échantillons en une seule fois

Comme le provisionnement du système de fichiers prend du temps au début de l'exécution, vous pouvez gagner du temps en regroupant plusieurs échantillons dans le même cycle. Tenez compte des facteurs suivants avant de choisir cette approche :

- Un seul échantillon défectueux peut entraîner l'échec d'un flux de travail. Le traitement par lots d'échantillons peut donc augmenter le nombre de flux de travail défaillants. Si vous n'êtes pas sûr que votre flux de travail réussira la plupart du temps, une exécution par échantillon pourrait être une meilleure approche.
- HealthOmics alloue un système de fichiers de stockage en une seule exécution pour l'ensemble du flux de travail. Pour un lot d'échantillons, assurez-vous de spécifier une quantité de stockage d'essais suffisante pour traiter tous les échantillons.
- La quantité maximale de stockage par flux de travail est limitée, ce qui peut limiter le nombre d'échantillons que vous pouvez ajouter au lot.
- La taille de stockage minimale est de 1,2 TiB. Le traitement par lots peut donc réduire les coûts si le flux de travail utilise beaucoup moins d'espace de stockage que le minimum requis pour chaque échantillon.

- Le stockage d'exécution peut gérer plusieurs connexions simultanées. Le fait que plusieurs tâches utilisent le même stockage d'exécution ne devrait donc pas provoquer de goulots d' I/O étranglement.
- Chaque course possède son propre ensemble de balises. Si vous balisez les flux de travail avec des informations à des fins de budgétisation ou de suivi, il peut être préférable d'utiliser des cycles séparés.
- Les rôles IAM s'appliquent à l'ensemble de la course. Chaque utilisateur a accès à toutes les données d'un lot d'échantillons. La séparation des flux de travail vous permet d'utiliser des autorisations plus précises.
- HealthOmics définit des quotas au niveau du compte pour le nombre maximum de flux de travail simultanés et le nombre maximum de tâches simultanées dans un flux de travail. Pour plus d'informations sur la procédure à suivre pour demander une augmentation de ces quotas, voirHealthOmics quotas de service.

Utiliser des paramètres pour les images de conteneurs

Paramétrez les images de vos conteneurs plutôt que de les intégrer URIs dans le flux de travail. Lorsqu'il s'agit de paramètres d'exécution, cela HealthOmics confirme que l'exécution a accès à vos conteneurs avant le début de l'exécution. Dans le cas contraire, la tâche échoue pendant l'exécution, lorsque vous avez engagé des frais pour toutes les tâches terminées. De plus, comme il s'agit d'entrées paramétrées, il HealthOmics génère une somme de contrôle dans le manifeste d'exécution, ce qui améliore la provenance des exécutions.

Utilisez un linter

Utilisez un linter pour détecter les erreurs de flux de travail courantes avant d'exécuter un nouveau flux de travail. Pour de plus amples informations, veuillez consulter Linters de flux de travail dans HealthOmics.

EventBridge À utiliser pour signaler les problèmes

Utilisez des alertes EventBridge personnalisées pour détecter les anomalies spécifiques à votre logique métier.

Utiliser des magasins de séquences

Envisagez d'utiliser un magasin de séquences pour vos données sources afin de réduire les coûts de stockage. Pour plus d'informations, consultez le billet de HealthOmics blog intitulé Store omics data de manière rentable à n'importe quelle échelle.

## Impact de la variation de taille de fichier entre les exécutions

Les utilisateurs conçoivent et testent souvent des séries en utilisant un petit ensemble de données de test, puis rencontrent une grande variété de données présentant une variation significative de la taille des fichiers lors des cycles de production. Assurez-vous de prendre en compte cet écart lorsque vous optimisez la course.

La liste suivante décrit les recommandations d'optimisation en cas de variation significative de la taille des fichiers :

Variez la taille des fichiers dans vos données de test

Essayez d'utiliser des données de test présentant une variance représentative pendant le développement.

Utiliser Run Analyzer

Utilisez l'outil Run Analyzer sur divers échantillons pour tenir compte de la variation de la taille des données.

Vous pouvez utiliser l'analyseur de cycles pour comprendre la variance entre les cycles dans vos échantillons de données de production. Utilisez `--batch` le mode dans Run Analyzer pour générer des statistiques pour un lot d'exécutions et analyser les ressources de calcul maximales requises pour gérer les valeurs aberrantes de vos ensembles de données.

Par exemple, vous pouvez attribuer à Run Analyzer une cellule à flux complet de données en mode batch afin de comprendre les pics d'utilisation du vCPU et de la mémoire pour la cellule à flux complet.

Réduire la variation de taille des ensembles de données en entrée

Si vous constatez une forte variation de la taille des échantillons, vous pouvez diviser les échantillons en amont HealthOmics et sélectionner des tailles de système de fichiers différentes pour chaque lot afin de réduire les coûts de stockage.

Dans WDL, utilisez la `size` fonction pour bifurquer l'allocation des ressources pour les tâches individuelles pour les grands échantillons par rapport aux petits échantillons. Appliquez cette stratégie à vos tâches les plus coûteuses afin d'avoir le plus d'impact possible.

Dans Nextflow, utilisez des ressources conditionnelles pour hiérarchiser l'allocation des ressources en fonction de la taille ou du nom du fichier. Pour plus d'informations, consultez la section Ressources relatives aux processus conditionnels sur le GitHub site Nextflow.

N'optimisez pas trop tôt

Finalisez le code et la logique de votre flux de travail avant d'investir dans d'importants efforts d'optimisation des performances. La modification de votre code peut avoir un impact significatif sur les ressources requises. Si vous optimisez une exécution trop tôt dans le processus de développement, vous risquez de suroptimiser ou de devoir l'optimiser à nouveau si la définition du flux de travail change ultérieurement.

Réexécutez régulièrement l'outil Run Analyzer

Si vous modifiez la définition de votre flux de travail au fil du temps ou si la variance de votre échantillon change, exécutez régulièrement l'outil Run Analyzer pour vous aider à effectuer des optimisations supplémentaires.

### Méthodes pour optimiser la simultanéité des ressources

HealthOmics fournit les fonctionnalités suivantes pour vous aider à contrôler et à gérer les coûts lors de cycles de traitement à grande échelle :

- Utilisez des groupes d'exécution pour contrôler vos coûts et votre utilisation des ressources. Vous pouvez définir des valeurs maximales dans le groupe d'exécutions pour le nombre d'exécutions simultanées, v CPUs GPUs, et le temps d'exécution total par tâche. Si des équipes ou des groupes distincts utilisent le même compte, vous pouvez créer un groupe de course distinct pour chaque équipe. Vous pouvez contrôler l'utilisation des ressources et les coûts par équipe et en configurant les valeurs maximales des groupes d'exécution. Pour de plus amples informations, veuillez consulter Utilisation de groupes d' HealthOmics exécution.
- Au cours du développement, vous pouvez configurer un groupe d'exécution distinct avec des valeurs maximales inférieures pour intercepter les tâches intempestives.
- Les Quotas de Service aident également à protéger votre compte contre les demandes de ressources excessives. Pour plus d'informations sur les Quotas de Service, notamment sur la manière de demander une augmentation de la valeur des quotas, voir HealthOmics quotas de service

## Exécuter des opérations dans HealthOmics

Vous pouvez démarrer, réexécuter, cloner, annuler ou supprimer une exécution :

- Start— HealthOmics crée une nouvelle exécution en utilisant les paramètres de configuration que vous spécifiez, puis lance l'exécution.
- Rerun— HealthOmics crée une nouvelle exécution qui est une copie de celle que vous spécifiez. Vous pouvez réexécuter une exécution supprimée à l'aide de l' HealthOmics rerunoutil.
- Clone— Vous pouvez cloner une exécution existante à l'aide de la console. La console ouvre la page d'exécution du clonage et préremplit les champs de configuration en utilisant les valeurs de l'exécution existante. Vous pouvez modifier les valeurs selon vos besoins et démarrer l'exécution clonée.
- Cancel— Vous pouvez annuler une course qui n'est pas encore terminée. Lorsque vous annulez une exécution, aucune des sorties d'exécution HealthOmics n'est enregistrée.
- Delete— Vous pouvez supprimer les essais terminés manuellement ou définir le mode de rétention des essais HealthOmics pour supprimer automatiquement les essais les plus anciens. Pour plus d'informations sur le mode de rétention, consultezthe section called “Exécuter les modes de rétention”.

Rubriques

- Commencez une course HealthOmics
- Réexécuter un run in HealthOmics
- Cloner un run in HealthOmics
- Annuler un run in HealthOmics
- Supprimer un run in HealthOmics

## Commencez une course HealthOmics

Lorsque vous lancez une exécution, vous spécifiez les ressources HealthOmics allouées à utiliser pendant l'exécution.

Spécifiez le type de stockage d'exécution et la quantité de stockage (pour le stockage statique). Pour garantir l'isolation et la sécurité des données, HealthOmics provisionne le stockage au début de chaque exécution et le déprovisionne à la fin de l'exécution. Pour plus d’informations, consultez Exécuter les types de stockage dans les HealthOmics flux de travail.

Spécifiez un emplacement Amazon S3 pour les fichiers de sortie. Si vous exécutez un grand nombre de flux de travail simultanément, utilisez une sortie Amazon S3 distincte URIs pour chaque flux de

travail afin d'éviter la limitation des compartiments. Pour plus d'informations, consultez Organiser les objets à l'aide de préfixes dans le guide de l'utilisateur Amazon S3 et Scale Storage Connections horizontalement dans le livre blanc sur l'optimisation des performances d'Amazon S3.

Vous pouvez également spécifier la priorité d'exécution. L'impact de la priorité sur l'exécution dépend du fait que l'exécution est associée ou non à un groupe d'exécution. Pour plus d'informations, consultez Priorité d'exécution.

Si un flux de travail comporte une ou plusieurs versions, vous pouvez spécifier une version lorsque vous démarrez l'exécution. Si vous ne spécifiez aucune version, HealthOmics démarre la version du flux de travail par défaut.

Lorsque vous utilisez l' HealthOmics API, vous pouvez fournir un identifiant de demande unique pour chaque exécution. L'ID de demande est un jeton d'idempuissance HealthOmics utilisé pour identifier les demandes dupliquées et ne démarre l'exécution qu'une seule fois.

> **Note**
>
> Vous spécifiez un rôle de service IAM lorsque vous lancez une exécution. La console peut éventuellement créer le rôle de service pour vous. Pour de plus amples informations, veuillez consulter Rôles de service pour AWS HealthOmics.

Rubriques



## HealthOmics paramètres d'exécution

Lorsque vous lancez une exécution, vous spécifiez les entrées d'exécution dans le fichier JSON des paramètres d'exécution ou vous pouvez saisir les valeurs des paramètres en ligne. Pour plus d'informations sur la gestion de la taille du fichier JSON des paramètres d'exécution, consultezGestion de la taille des paramètres d'exécution.

HealthOmics prend en charge les types JSON suivants pour les valeurs de paramètres.

| Type JSON | Exemple de clé et de valeur | Remarques |
|---|---|---|
| boolean | « b » : vrai | La valeur n'est pas entre guillemets, elle est entièrement en minuscules. |
| entier | « i » :7 | La valeur n'est pas entre guillemets. |
| nombre | « f » :42,3 | La valeur n'est pas entre guillemets. |
| chaîne | « s » ="caractères » | La valeur est entre guillemets. Utilisez le type chaîne pour les valeurs de texte et URIs. L'URI cible doit être le type d'entrée attendu. |
| array | « a » : [1,2,3] | La valeur n'est pas entre guillemets. Les membres du tableau doivent chacun avoir le type défini par le paramètre d'entrée. |
| objet | « o » : {"gauche » « a », « droite » :1} | Dans WDL, l'objet correspond à une paire, une carte ou une structure WDL |

Démarrer une course à l'aide de la console

Pour démarrer une course

1. Ouvrez la HealthOmics console.
2. Si nécessaire, ouvrez le volet de navigation de gauche (≡). Choisissez Runs.
3. Sur la page Exécutions, choisissez Démarrer l'exécution.
4. Dans le panneau Détails de l'exécution, fournissez les informations suivantes :

- Source du flux de travail : choisissez un flux de travail propriétaire ou un flux de travail partagé.
- ID de flux de travail : ID de flux de travail associé à cette exécution.
- Version du flux de travail (facultatif) - Sélectionnez la version du flux de travail à utiliser pour cette exécution. Si vous ne sélectionnez aucune version, l'exécution utilise la version par défaut du flux de travail.
- Nom de l'exécution : nom distinctif de cette exécution.
- Priorité d'exécution (facultatif) : priorité de cette exécution. Les nombres les plus élevés indiquent une priorité plus élevée, et les tâches les plus prioritaires sont exécutées en premier.
- Type de stockage d'exécution : spécifiez le type de stockage ici pour remplacer le type de stockage d'exécution par défaut spécifié pour le flux de travail. Le stockage statique alloue une quantité fixe de stockage pour l'exécution. Le stockage dynamique augmente ou diminue en fonction des besoins de chaque tâche en cours d'exécution.
- Capacité de stockage d'exécution : pour le stockage statique, spécifiez la quantité de stockage nécessaire pour l'exécution. Cette entrée remplace la quantité de stockage d'exécution par défaut spécifiée pour le flux de travail.
- Sélectionnez la destination de sortie S3 : emplacement S3 où les sorties d'exécution seront enregistrées.
- ID de compte du propriétaire du bucket de sortie (facultatif) - Si votre compte ne possède pas le bucket de sortie, entrez l' Compte AWS ID du propriétaire du bucket. Ces informations sont nécessaires pour vérifier HealthOmics la propriété du bucket.
- Mode de conservation des métadonnées d'exécution : choisissez de conserver les métadonnées pour toutes les exécutions ou de demander au système de supprimer les métadonnées d'exécution les plus anciennes lorsque votre compte atteint le nombre maximum d'exécutions. Pour de plus amples informations, veuillez consulter Exécuter le mode de rétention pour les HealthOmics courses.

5. Sous Rôle de service, vous pouvez utiliser un rôle de service existant ou en créer un nouveau.
6. (Facultatif) Pour les balises, vous pouvez attribuer jusqu'à 50 balises à exécuter.
7. Choisissez Suivant.
8. Sur la page Ajouter des valeurs de paramètres, indiquez les paramètres d'exécution. Vous pouvez soit télécharger un fichier JSON qui spécifie les paramètres, soit saisir les valeurs manuellement.
9. Choisissez Suivant.

10. Dans le panneau Exécuter un groupe, vous pouvez éventuellement spécifier un groupe d'exécution pour cette exécution. Pour de plus amples informations, veuillez consulter Utilisation de groupes d' HealthOmics exécution.
11. Dans le panneau Exécuter le cache, vous pouvez éventuellement spécifier un cache d'exécution pour cette exécution. Pour de plus amples informations, veuillez consulter Configuration d'une exécution avec cache d'exécution à l'aide de la console.
12. Choisissez Review and start run (Vérifier et démarrer l'exécution).
13. Après avoir examiné la configuration d'exécution, choisissez Démarrer l'exécution.

Démarrer une course à l'aide de l'API

Utilisez l'opération d'API start-run pour créer et démarrer une exécution.

L'exemple suivant indique l'ID du flux de travail et le rôle de service. Cet exemple définit le mode de rétention sur`REMOVE`. Pour plus d'informations sur le mode de rétention, consultezExécuter le mode de rétention pour les HealthOmics courses.

```
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
```

En réponse, vous obtenez le résultat suivant. Le `uuid` est propre à l'exécution et `outputUri` peut également être utilisé pour suivre l'endroit où les données de sortie sont écrites.

```
{
    "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"
}
```

Inclure un fichier de paramètres

Si le modèle de paramètres d'un flux de travail déclare des paramètres obligatoires, vous pouvez fournir un fichier JSON local contenant les entrées lorsque vous démarrez l'exécution d'un flux de

travail. Le fichier JSON contient le nom exact de chaque paramètre d'entrée et une valeur pour le paramètre.

Référencez le fichier JSON d'entrée dans le en l' AWS CLI ajoutant `--parameters file://<input_file.json>` à votre `start-run` demande. Pour plus d'informations sur les paramètres d'exécution, consultezHealthOmics exécuter les entrées.

Fournir un identifiant de demande

Vous pouvez fournir un numéro unique `requestId` pour chaque course. L'ID de demande est un jeton d'impuissance HealthOmics utilisé pour intercepter les demandes dupliquées. Il ne démarrera pas une exécution si l'ID de demande est un double d'une exécution précédente.

Si vous utilisez une infrastructure (telle que des fonctions Lambda ou des fonctions par étapes) pour orchestrer les démarrages, la meilleure pratique consiste à fournir un identifiant de demande unique pour chaque demande. StartRun Cela garantit que si votre infrastructure lance par inadvertance une exécution déjà lancée, HealthOmics elle ne démarrera pas l'exécution dupliquée. Par exemple, si l'infrastructure tente de se rétablir après une erreur en amont, elle peut réexécuter un script qui tente de lancer des exécutions correspondant à des demandes dupliquées.

Choisissez une version du flux de travail

Vous pouvez spécifier une version du flux de travail pour l'exécution. Si vous ne spécifiez aucune version, HealthOmics lance l'exécution avec la version du flux de travail par défaut.

```
aws omics start-run
     --workflow-id workflow id \
      ...
     --workflow-version-name '1.2.1'
```

Remplacer le type de stockage d'exécution

Vous pouvez remplacer le type de stockage d'exécution par défaut défini dans le flux de travail.

```
aws omics start-run
       --workflow-id workflow id \
        ...
       --storage-type STATIC
       --storage-capacity 2400
```

Exécuter un flux de travail GPU

Vous pouvez également spécifier un ID de flux de travail GPU, comme illustré dans l'exemple suivant :

```
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
```

Obtenir des informations sur une course

Vous pouvez utiliser l'ID dans la réponse avec l'API get-run pour vérifier l'état d'une exécution, comme indiqué.

```
aws omics get-run --id run id
```

La réponse de cette opération d'API vous indique l'état du flux de travail exécuté. Les statuts possibles sont`PENDING`, `STARTINGRUNNING`, et. `COMPLETED` Lorsqu'il s'agit d'une exécution`COMPLETED`, vous pouvez trouver un fichier de sortie appelé `outfile.txt` dans votre compartiment de sortie Amazon S3, dans un dossier nommé d'après l'ID d'exécution.

L'opération d'API get-run renvoie également d'autres informations, telles que la question de savoir si le flux de travail est `Ready2Run` ou`PRIVATE`, le moteur du flux de travail et les détails de l'accélérateur. L'exemple suivant montre la réponse de get-run pour l'exécution d'un flux de travail privé, décrite dans WDL avec un accélérateur GPU et aucune balise n'étant attribuée à l'exécution.

```
{
    "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": {}
}
```

Vous pouvez voir l'état de toutes les exécutions avec l'opération d'API list-runs, comme indiqué.

```
aws omics list-runs
```

Pour voir toutes les tâches effectuées pour une exécution spécifique, utilisez l'list-run-tasksAPI.

```
aws omics list-run-tasks --id task ID
```

Pour obtenir les détails d'une tâche spécifique, utilisez l' get-run-taskAPI.

```
aws omics get-run-task --id <run_id> --task-id task ID
```

Une fois l'exécution terminée, les métadonnées sont envoyées CloudWatch sous le flux**`manifest/run/<run ID>/<run UUID>`**.

Voici un exemple de manifeste.

```
{
    "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"
```

```
  },
  ...
```

Les métadonnées d'exécution ne sont pas supprimées si elles ne figurent pas dans les CloudWatch journaux.

## Réexécuter un run in HealthOmics

Pour les exécutions que vous n'avez pas encore supprimées, utilisez la console ou l'API pour réexécuter l'exécution. Pour les courses que vous avez supprimées, utilisez l' HealthOmics rerunoutil.

Rubriques

- Réexécuter une exécution à l'aide de la console
- Réexécuter une exécution à l'aide de l'API
- Utilisation de l'outil Rerun

### Réexécuter une exécution à l'aide de la console

Depuis la console, procédez comme suit pour relancer une exécution :

1. Ouvrez la HealthOmics console.
2. Si nécessaire, ouvrez le volet de navigation de gauche (≡). Choisissez Runs.
3. Sur la page Exécutions, sélectionnez l'exécution à exécuter à nouveau.
4. Dans le menu d'action situé au-dessus du tableau, choisissez Re-exécuter.

### Réexécuter une exécution à l'aide de l'API

Utilisez l'opération StartRun API pour réexécuter une exécution existante. Fournissez les entrées requises suivantes :

- Un rôle de service ARN (`roleArn`).
- L'ID de l'exécution à dupliquer (`runId`).
- Un emplacement Amazon S3 où l'exécution enregistre les sorties d'exécution (`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
```

Utilisation de l'outil Rerun

Pour une exécution supprimée, vous pouvez télécharger et utiliser l' HealthOmics rerunoutil pour la réexécuter. L'outil extrait les informations d'exécution à partir du manifeste CloudWatch Logs. Téléchargez l'rerunoutil depuis le GitHub référentiel HealthOmics d'outils.

L'exemple suivant montre comment utiliser l'rerunoutil.

```
aws-healthomics-rerun 9876543
```

Si l'exécution existe dans CloudWatch, vous recevez une réponse similaire à l'exemple de sortie suivant. Si le flux de travail n'existe plus, vous recevez un message d'erreur.

```
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": {}
}
```

## Cloner un run in HealthOmics

Vous pouvez cloner une exécution existante à l'aide de la HealthOmics console. Le clonage crée une nouvelle exécution en utilisant les valeurs de configuration de l'exécution clonée. Vous pouvez modifier ces valeurs par défaut et ajouter d'autres entrées facultatives.

1. Ouvrez la HealthOmics console.
2. Si nécessaire, ouvrez le volet de navigation de gauche (≡). Choisissez Runs.
3. Sur la page Exécutions, sélectionnez l'exécution à cloner.
4. Dans le menu d'actions situé au-dessus du tableau, choisissez Clone run. La console ouvre le formulaire d'exécution du clone. Le formulaire est identique à Start run, sauf que la console le remplit avec toutes les valeurs pertinentes de l'exécution clonée.

   La console crée un nouvel ID d'exécution pour le clone d'exécution et ajoute cet ID d'exécution en tant que suffixe au nom de l'exécution.

   Au fur et à mesure que vous parcourez les pages du formulaire, vous pouvez ajuster les valeurs de configuration selon vos besoins.
5. Après avoir examiné la configuration d'exécution, choisissez Démarrer l'exécution.

## Annuler un run in HealthOmics

Vous pouvez annuler une course si son statut est `PENDING`,`STARTING`,`RUNNING`, ou`STOPPING`.

> **Note**
>
> Lorsque vous annulez une exécution, aucune des sorties d'exécution HealthOmics n'est enregistrée.

Depuis la console, procédez comme suit pour annuler une exécution :

1. Ouvrez la HealthOmics console.
2. Si nécessaire, ouvrez le volet de navigation de gauche (≡). Choisissez Runs.
3. Sur la page Exécutions, choisissez l'exécution à annuler.
4. La console ouvre la page de détails de l'exécution. Dans le bandeau d'état en haut de la page, choisissez Arrêter l'exécution.
5. Entrez « Confirmer » pour arrêter l'exécution.

Pour annuler une exécution à l'aide de l'API, utilisez l'opération CancelRun API.

L'exemple suivant montre comment annuler une course à l'aide du AWS CLI . Pour exécuter cet exemple, remplacez le *run id* par l'ID de l'exécution que vous souhaitez annuler. En cas de succès, il n'y aura aucune réponse.

```
aws omics cancel-run --id run id
```

## Supprimer un run in HealthOmics

Lorsque vous n'avez plus besoin d'une exécution, vous pouvez la supprimer à l'aide de l' AWS CLI API ou de la console. Vous pouvez supprimer une course lorsque son statut est `COMPLETED` ou`CANCELED`.

Depuis la console, procédez comme suit pour supprimer une exécution :

1. Ouvrez la HealthOmics console.
2. Si nécessaire, ouvrez le volet de navigation de gauche (≡). Choisissez Runs.
3. Sur la page Exécutions, sélectionnez une ou plusieurs séries à supprimer.
4. Dans le menu d'action situé au-dessus du tableau, choisissez Supprimer.
5. Dans le formulaire modal, tapez confirm pour confirmer la suppression.

La AWS CLI commande suivante supprime une exécution. Pour exécuter l'exemple, remplacez le *run id* par l'ID de l'exécution que vous souhaitez supprimer. Il n'y a aucune réponse si l'exécution est correctement supprimée.

```
aws omics delete-run --id run id
```

# Utilisation de groupes d' HealthOmics exécution

Vous pouvez éventuellement créer un groupe d'exécution afin de plafonner les ressources de calcul pour les exécutions que vous ajoutez au groupe. Les groupes de course peuvent vous aider à :

- Mettez vos courses en file d'attente afin de ne pas dépasser les limites de service.
- Identifiez les tâches inattendues en définissant une durée d'exécution maximale.
- Gérez la priorité de chaque exécution afin que les séries les plus importantes soient terminées en premier.

Si vous définissez le nombre maximal de vCPU, de GPU ou d'exécutions simultanées, les tâches d'exécution seront mises en file d'attente lorsque le maximum sera atteint. Si vous définissez une durée d'exécution maximale, l'exécution échoue si elle dépasse la durée maximale.

Utilisez le paramètre de priorité d'exécution pour établir la priorité au sein d'un groupe d'exécution.

Les limites de service ont priorité sur les limites de groupes d'exécution. Par exemple, si vous définissez un maximum de groupe d'exécution sur une valeur supérieure à votre maximum de service dans une région, HealthOmics applique le maximum de service.

Rubriques



## Priorité d'exécution

Vous pouvez utiliser la priorité d'exécution pour définir la priorité des exécutions dans un groupe d'exécution.

Si plusieurs exécutions ont la même priorité, l'exécution démarrée en premier a la priorité la plus élevée.

Vous pouvez également définir une priorité pour une exécution qui ne fait pas partie d'un groupe de courses. La priorité est comparée aux priorités de tous les autres essais qui ne font pas partie d'un groupe d'essais

Vous définissez la priorité d'exécution lorsque vous démarrez la course. Pour de plus amples informations, veuillez consulter Commencez une course HealthOmics.

## Création d'un groupe d'exécution à l'aide de la console

Pour créer un groupe de course

1. Ouvrez la HealthOmics console.
2. Si nécessaire, ouvrez le volet de navigation de gauche (≡). Choisissez Exécuter les groupes.
3. Sur la page Exécuter des groupes, choisissez Créer un groupe d'exécution.
4. Sur la page de détails de la création d'un groupe d'exécution, fournissez les informations suivantes
   - Nom du groupe d'exécution : nom unique pour ce groupe d'exécution.
   - Nombre maximal de processeurs virtuels pour les exécutions simultanées : nombre maximal de processeurs v CPUs pouvant être exécutés simultanément sur toutes les exécutions actives du groupe d'exécution.
   - Max GPUs : nombre maximum d'exécutions GPUs simultanées sur toutes les exécutions actives du groupe d'exécutions.
   - Durée maximale (minutes) par exécution : durée maximale de chaque exécution (en minutes). Si une exécution dépasse la durée d'exécution maximale, elle échoue automatiquement.
   - Nombre maximal d'exécutions simultanées : nombre maximal d'exécutions pouvant être exécutées simultanément.
5. (facultatif) Vous pouvez ajouter jusqu'à 50 balises au groupe d'exécution.
6. Choisissez Create run group.

## Création d'un groupe d'exécution à l'aide de la CLI

Pour créer un groupe d'exécution, utilisez l'opération create-run-groupd'API pour créer un groupe d'exécution nommé`TestRunGroup`. L'exemple suivant définit un maximum de 20 CPUs GPUs, 10 ou 5 essais et une durée d'exécution maximale de 600 minutes.

```
aws omics create-run-group --name TestRunGroup \
--max-cpus 20 \
--max-gpus 10 \
--max-duration 600 \
--max-runs 5
```

La réponse de cette opération d'API inclut l'ID de la personne nouvellement créée`RunGroup`.

```
{
    "arn": "arn:aws:omics:us-west-2:12345678901:runGroup/2839621",
    "id": "2839621",
    "tags": {}
}
```

Pour obtenir des informations supplémentaires sur le groupe d'exécution, utilisez cet ID avec l'opération d'get-run-groupAPI, comme illustré dans l'exemple suivant.

```
aws omics get-run-group --id run group id
```

La réponse inclut les paramètres de limite pour le groupe d'exécution et les balises attribuées.

```
{
    "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
}
```

Vous pouvez également utiliser l'opération list-run-groupAPI pour afficher tous les groupes d'exécution créés.

```
aws omics list-run-groups
```

## Supprimer un groupe d'exécution à l'aide de la console

Vous pouvez supprimer un groupe d'exécution si aucun essai n'est associé à ce groupe d'exécution ayant le statut `PENDINGSTARTING`,`RUNNING`, ou`STOPPING`.

Pour supprimer un groupe d'exécution, procédez comme suit.

1. Ouvrez la HealthOmics console.
2. Si nécessaire, ouvrez le volet de navigation de gauche (≡). Choisissez Exécuter les groupes.
3. Sur la page Exécuter les groupes, choisissez le groupe d'exécution à supprimer, puis choisissez Supprimer dans le xx.

## Supprimer un groupe d'exécution à l'aide de la CLI

Vous pouvez supprimer un groupe d'exécution si aucun essai n'est associé à ce groupe d'exécution ayant le statut `PENDINGSTARTING`,`RUNNING`, ou`STOPPING`.

L'exemple suivant montre comment vous pouvez utiliser le AWS CLI pour supprimer un groupe d'exécution. Vous ne recevrez pas de réponse. Pour exécuter l'exemple, remplacez le *`run group id`* par l'ID du groupe d'exécution que vous souhaitez supprimer.

```
aws omics delete-run-group --id run group id
```

# Mise en cache des appels pour les exécutions HealthOmics

AWS HealthOmics prend en charge la mise en cache des appels, également appelée reprise, pour les flux de travail privés. La mise en cache des appels enregistre les résultats des tâches de flux de travail terminées une fois l'exécution terminée. Les exécutions suivantes peuvent utiliser les résultats des tâches du cache, plutôt que de les calculer à nouveau. La mise en cache des appels réduit l'utilisation des ressources informatiques, ce qui se traduit par des durées d'exécution plus courtes et des économies de coûts de calcul.

Vous pouvez accéder aux fichiers de sortie des tâches mis en cache une fois l'exécution terminée. Pour effectuer un débogage et un dépannage avancés des tâches, vous pouvez mettre en cache des fichiers de tâches intermédiaires en les spécifiant comme résultats de tâches dans la définition du flux de travail.

Vous pouvez utiliser la mise en cache des appels pour enregistrer les résultats des tâches terminées en cas d'échec d'exécution. La prochaine exécution commence à partir de la dernière tâche terminée avec succès, plutôt que de calculer à nouveau les tâches terminées.

Si HealthOmics aucune entrée de cache correspondante n'est trouvée pour une tâche, l'exécution n'échoue pas. HealthOmics recalcule la tâche et ses tâches dépendantes.

Pour plus d'informations sur la résolution des problèmes de mise en cache des appels, consultezRésolution des problèmes de mise en cache des appels.

Rubriques



## Comment fonctionne la mise en cache des appels

Pour utiliser la mise en cache des appels, vous créez un cache d'exécution et vous le configurez pour qu'il soit associé à un emplacement Amazon S3 pour les données mises en cache. Lorsque vous lancez une exécution, vous spécifiez le cache d'exécution. Un cache d'exécution n'est pas dédié à un seul flux de travail. Les exécutions à partir de plusieurs flux de travail peuvent utiliser le même cache.

Pendant la phase d'exportation d'une exécution, le système exporte les résultats des tâches terminées vers l'emplacement Amazon S3. Pour exporter des fichiers de tâches intermédiaires, déclarez ces fichiers en tant que résultats de tâches dans la définition du flux de travail. La mise en cache des appels enregistre également les métadonnées en interne et crée des hachages uniques pour chaque entrée du cache.

Pour chaque tâche d'une exécution, le moteur de flux de travail détecte s'il existe une entrée de cache correspondante pour cette tâche. Si aucune entrée de cache ne correspond, HealthOmics calcule la tâche. Si une entrée de cache correspond, le moteur récupère les résultats mis en cache.

Pour faire correspondre les entrées du cache, HealthOmics utilise le mécanisme de hachage inclus dans les moteurs de flux de travail natifs. HealthOmicsétend ces implémentations de hachage

existantes pour tenir compte de HealthOmics variables, telles que les ETags S3 et les résumés de conteneurs ECR.

HealthOmics prend en charge la mise en cache des appels pour les versions linguistiques du flux de travail suivantes :

- Versions WDL 1.0, 1.1 et version de développement
- Nextflow versions 23.10 et 24.10
- Toutes les versions de CWL

> **Note**
>
> HealthOmics ne prend pas en charge la mise en cache des appels pour les flux de travail Ready2Run.

Rubriques



## Modèle de responsabilité partagée

La responsabilité est partagée entre les utilisateurs et consiste AWS à déterminer si les tâches et les exécutions sont de bons candidats pour la mise en cache des appels. La mise en cache des appels donne les meilleurs résultats lorsque toutes les tâches sont idempotentes (les exécutions répétées d'une tâche utilisant les mêmes entrées produisent les mêmes résultats).

Toutefois, si une tâche inclut des éléments non déterministes (tels que des générations de nombres aléatoires ou l'heure du système), des exécutions répétées de la tâche utilisant les mêmes entrées peuvent entraîner des sorties différentes. Cela peut avoir un impact sur l'efficacité de la mise en cache des appels des manières suivantes :

- Si elle HealthOmics utilise une entrée de cache (créée lors d'une exécution précédente) qui n'est pas identique à la sortie que l'exécution de la tâche produirait pour l'exécution en cours, l'exécution peut donner des résultats différents de ceux de la même exécution sans mise en cache.

- HealthOmics peut ne pas trouver d'entrée de cache correspondante pour une tâche qui devrait correspondre, en raison de sorties de tâche non déterministes. Si elle ne trouve pas l'entrée de cache valide, l'exécution recalcule inutilement la tâche, ce qui réduit les économies liées à l'utilisation de la mise en cache des appels.

Les comportements de tâches connus suivants peuvent entraîner des résultats non déterministes affectant les résultats de la mise en cache des appels :

- Utilisation de générateurs de nombres aléatoires.
- Dépendance de l'heure du système.
- Utilisation de la simultanéité (les conditions de course peuvent entraîner une variation de sortie).
- Extraction de fichiers locaux ou distants au-delà de ce qui est spécifié dans les paramètres d'entrée de la tâche.

Pour d'autres scénarios susceptibles de provoquer un comportement non déterministe, consultez la section Entrées de processus non déterministes sur le site de documentation de Nextflow.

Si vous pensez qu'une tâche produit des résultats non déterministes, pensez à utiliser les fonctionnalités du moteur de flux de travail pour éviter de mettre en cache des tâches spécifiques non déterministes. Pour savoir comment désactiver la mise en cache pour des tâches individuelles dans chaque langue de flux de travail prise en charge, consultezFonctionnalités de mise en cache spécifiques au moteur.

Nous vous recommandons de revoir attentivement vos exigences spécifiques en matière de flux de travail et de tâches avant d'activer la mise en cache des appels dans les environnements dans lesquels une mise en cache inefficace ou des résultats différents de ceux attendus peuvent présenter un risque. Par exemple, les limites potentielles de la mise en cache des appels doivent être soigneusement prises en compte pour déterminer si la mise en cache des appels est appropriée pour les cas d'utilisation cliniques.

## Exigences de mise en cache pour les tâches

HealthOmics met en cache les résultats des tâches répondant aux exigences suivantes :

- La tâche doit définir un conteneur. HealthOmics ne mettra pas en cache les sorties pour une tâche sans conteneur.

- La tâche doit produire une ou plusieurs sorties. Vous spécifiez les résultats des tâches dans la définition du flux de travail.
- La définition du flux de travail ne doit pas utiliser de valeurs dynamiques. Par exemple, si vous transmettez un paramètre à une tâche dont la valeur augmente à chaque exécution, les résultats de la tâche HealthOmics ne sont pas mis en cache.

> **Note**
>
> Si plusieurs tâches en cours d'exécution utilisent la même image de conteneur, HealthOmics fournit la même version d'image pour toutes ces tâches. Une fois HealthOmics l'image extraite, elle ignore toute mise à jour de l'image du conteneur pendant toute la durée de l'exécution. Cette approche fournit une expérience prévisible et cohérente et permet d'éviter les problèmes potentiels susceptibles de découler des mises à jour de l'image du conteneur déployées en cours d'exécution.

## Exécuter les performances du cache

Lorsque vous activez la mise en cache des appels pour une exécution, vous pouvez constater les impacts suivants sur les performances d'exécution :

- Lors de la première exécution, HealthOmics enregistre les données du cache pour les tâches en cours d'exécution. Les délais d'exportation peuvent être plus longs pour cette exécution, car la mise en cache des appels augmente la quantité de données d'exportation.
- Lors des exécutions suivantes, lorsque vous reprenez une exécution depuis le cache, le nombre d'étapes de traitement peut être réduit et le temps d'exécution peut être réduit.
- Si vous choisissez également de déclarer des fichiers intermédiaires en tant que sorties, vos délais d'exportation peuvent être encore plus longs car ces données peuvent être plus détaillées.

## Événements de conservation et d'invalidation des données du cache

L'objectif principal d'un cache d'exécution est d'optimiser le calcul des tâches en cours d'exécution. S'il existe une entrée de cache correspondante valide pour une tâche, HealthOmics utilise l'entrée de cache au lieu de recalculer la tâche. Sinon, HealthOmics revient au comportement de service par défaut, qui consiste à recalculer la tâche et ses tâches dépendantes. En utilisant cette approche, les erreurs de cache n'entraînent pas l'échec de l'exécution.

Nous vous recommandons de gérer la taille du cache d'exécution. Au fil du temps, les entrées du cache peuvent ne plus être valides en raison de mises à jour du moteur de flux de travail ou du HealthOmics service ou en raison de modifications apportées lors de l'exécution ou des tâches d'exécution. Les sections suivantes fournissent des informations supplémentaires.

Rubriques



## Mises à jour des versions du manifeste et fraîcheur des données

Régulièrement, le HealthOmics service peut introduire de nouvelles fonctionnalités ou des mises à jour du moteur de flux de travail qui invalident certaines ou toutes les entrées du cache d'exécution. Dans ce cas, le cache peut manquer une seule fois lors de vos exécutions.

HealthOmics crée un fichier manifeste JSON pour chaque entrée de cache. Pour les exécutions démarrées après le 12 février 2025, le fichier manifeste inclut un paramètre de version. Si une mise à jour du service invalide des entrées de cache, le numéro de version est HealthOmics incrémenté afin que vous puissiez identifier les anciennes entrées de cache à supprimer.

L'exemple suivant montre un fichier manifeste dont la version est définie sur 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,
}
```

Pour les exécutions avec des entrées de cache qui ne sont plus valides, reconstruisez le cache pour créer de nouvelles entrées valides. Procédez comme suit pour chaque exécution :

1. Lancez l'exécution une fois avec la rétention du cache définie sur CACHE ALWAYS. Cette exécution crée les nouvelles entrées de cache.
2. Pour les exécutions suivantes, réglez la rétention du cache sur son ancien paramètre (CACHE ALWAYS ou CACHE ON FAILURE).

Pour nettoyer les entrées de cache qui ne sont plus valides, vous pouvez les supprimer du compartiment de cache Amazon S3. HealthOmics ne réutilise jamais ces entrées de cache. Si vous choisissez de conserver les entrées non valides, cela n'a aucun impact sur vos courses.

> **Note**
>
> La mise en cache des appels enregistre les données de sortie des tâches dans l'emplacement Amazon S3 spécifié pour le cache, ce qui entraîne des frais pour vous. Compte AWS

### Exécuter le comportement du cache

Vous pouvez définir le comportement du cache d'exécution pour enregistrer les résultats des tâches pour les exécutions qui échouent (cache en cas d'échec) ou pour toutes les exécutions (cache toujours). Lorsque vous créez un cache d'exécution, vous définissez le comportement de cache par défaut pour toutes les exécutions utilisant ce cache. Vous pouvez modifier le comportement par défaut lorsque vous lancez une course.

Cache on failureest utile si vous débugguez un flux de travail qui échoue après l'exécution réussie de plusieurs tâches. L'exécution suivante reprend à partir de la dernière tâche terminée avec succès si toutes les variables uniques prises en compte par le hachage sont identiques à celles de l'exécution précédente.

Cache alwaysest utile si vous mettez à jour une tâche dans un flux de travail qui s'exécute correctement. Nous vous recommandons de suivre les étapes suivantes :

1. Créez une nouvelle course. Définissez le comportement du cache sur Cache always, puis lancez l'exécution.
2. Une fois l'exécution terminée, mettez à jour la tâche dans le flux de travail et lancez une nouvelle exécution avec le comportement défini Cache always. Cette exécution traite la tâche mise à jour et toutes les tâches suivantes qui dépendent de la tâche mise à jour. Toutes les autres tâches utilisent les résultats mis en cache.
3. Répétez l'étape 2 selon les besoins, jusqu'à ce que le développement de la tâche mise à jour soit terminé.
4. Utilisez la tâche mise à jour selon vos besoins lors de futures exécutions. N'oubliez pas de basculer les exécutions suivantes vers le cache en cas d'échec si vous prévoyez d'utiliser des entrées nouvelles ou différentes pour ces exécutions.

> **Note**
>
> Nous recommandons de toujours utiliser le mode Cache lorsque vous utilisez le même ensemble de données de test, mais pas pour un lot d'exécutions. Si vous définissez ce mode pour un grand nombre d'exécutions, le système peut exporter de grandes quantités de données vers Amazon S3, ce qui augmente les délais d'exportation et les coûts de stockage.

### Contrôler la taille du cache d'exécution

HealthOmics ne supprime ni n'archive automatiquement les données d'exécution du cache et n'applique pas les règles de nettoyage d'Amazon S3 pour gérer les données du cache. Nous vous recommandons d'effectuer des nettoyages de cache réguliers afin de réduire les coûts de stockage d'Amazon S3 et de maintenir la taille de votre cache d'exécution à un niveau gérable. Vous pouvez supprimer des fichiers directement ou définir des retention/replication politiques de données sur le bucket de cache d'exécution.

Par exemple, vous pouvez configurer une politique de cycle de vie Amazon S3 pour faire expirer les objets après 90 jours, ou vous pouvez nettoyer manuellement les données du cache à la fin de chaque projet de développement.

Les informations suivantes peuvent vous aider à gérer la taille des données du cache :

- Vous pouvez voir la quantité de données dans le cache en consultant Amazon S3. HealthOmics ne surveille ni ne rend compte de la taille du cache.

- Si vous supprimez une entrée de cache valide, l'exécution suivante n'échoue pas. HealthOmics recalcule la tâche et ses tâches dépendantes.
- Si vous modifiez les noms de cache ou les structures de répertoires de telle sorte qu'aucune entrée correspondante ne HealthOmics puisse être trouvée pour une tâche, HealthOmics recalcule la tâche.

Si vous devez vérifier si une entrée de cache est toujours valide, vérifiez le numéro de version du manifeste de cache. Pour de plus amples informations, veuillez consulter Mises à jour des versions du manifeste et fraîcheur des données.

## Création d'un cache d'exécution

Lorsque vous créez un cache d'exécution, vous spécifiez un emplacement Amazon S3 pour les données du cache. Ces données doivent être immédiatement accessibles. La mise en cache des appels ne permet pas de récupérer les objets archivés dans Glacier (tels que les classes de stockage GFR et GDA).

Si le compartiment Amazon S3 contenant les données du cache appartient à une autre personne Compte AWS, fournissez cet ID de compte lorsque vous créez le cache d'exécution.

### Création d'un cache d'exécution à l'aide de la console

À partir de la console, procédez comme suit pour créer un cache d'exécution.

1. Ouvrez la HealthOmics console.
2. Si nécessaire, ouvrez le volet de navigation de gauche (≡). Choisissez Exécuter les caches.
3. Sur la page Exécuter les caches, choisissez Create run cache.
4. Dans le panneau Détails du cache d'exécution de la page Créer un cache d'exécution, configurez les champs suivants :

    a. Entrez un nom pour le cache d'exécution.

    b. (Facultatif) Entrez une description.

    c. Entrez un emplacement S3 pour la sortie mise en cache. Choisissez un bucket dans la même région que votre flux de travail.

    d. (Facultatif) Entrez le nom Compte AWS du propriétaire du compartiment pour vérifier qu'il est bien le propriétaire du compartiment. Si vous ne saisissez aucune valeur, la valeur par défaut est votre numéro de compte.

e. Sous Comportement du cache, configurez le comportement par défaut (pour mettre en cache les sorties en cas d'échec ou pour toutes les exécutions). Lorsque vous lancez une exécution, vous pouvez éventuellement modifier le comportement par défaut.

5. (Facultatif) Associez une ou plusieurs balises au cache d'exécution.
6. Choisissez Create run cache. La console affiche le nouveau cache d'exécution dans le tableau des caches d'exécution.

### Création d'un cache d'exécution à l'aide de la CLI

Utilisez la commande create-run-cacheCLI pour créer un cache d'exécution. Le comportement du cache par défaut est`CACHE_ON_FAILURE`.

```
aws omics create-run-cache \
      --name "workflow 123 run cache" \
      --description "my run cache" \
      --cache-s3-location "s3://amzn-s3-demo-bucket" \
      --cache-behavior "CACHE_ALWAYS"                \
      --cache-bucket-owner-id  "111122223333"
```

Si la création est réussie, vous recevez une réponse contenant les champs suivants.

```
{
  "arn": "string",
  "id": "string",
  "status": "ACTIVE"
  "tags": {}
  }
```

## Mettre à jour un cache d'exécution

Vous pouvez modifier le nom, la description, les balises ou le comportement du cache, mais pas l'emplacement S3 du cache.

### Mise à jour d'un cache d'exécution à l'aide de la console

Depuis la console, procédez comme suit pour mettre à jour un cache d'exécution.

1. Ouvrez la HealthOmics console.

2. Si nécessaire, ouvrez le volet de navigation de gauche (≡). Choisissez Exécuter les caches.
3. Dans le tableau Exécuter les caches, choisissez le cache d'exécution à mettre à jour, puis sélectionnez Modifier.
4. Dans le panneau Détails du cache d'exécution, vous pouvez mettre à jour les champs du nom, de la description et du comportement du cache d'exécution.
5. (Facultatif) Associez une ou plusieurs nouvelles balises au cache d'exécution ou supprimez les balises existantes.
6. Choisissez Enregistrer le cache d'exécution.

### Mise à jour d'un cache d'exécution à l'aide de la CLI

Utilisez la commande update-run-cacheCLI pour mettre à jour un cache d'exécution.

```
aws omics update-run-cache \
      --name "workflow 123 run cache" \
      --id "workflow id" \
      --description "my run cache" \
      --cache-behavior "CACHE_ALWAYS"
```

Si la mise à jour est réussie, vous recevez une réponse sans champs de données.

## Supprimer un cache d'exécution

Vous pouvez supprimer un cache d'exécution s'il n'est utilisé par aucun essai actif. Si des exécutions utilisent le cache d'exécution, attendez qu'elles soient terminées ou annulez-les.

La suppression d'un cache d'exécution supprime la ressource et ses métadonnées, mais ne supprime pas les données dans Amazon S3. Une fois le cache supprimé, vous ne pouvez pas le rattacher ou l'utiliser pour les exécutions suivantes.

Les données mises en cache restent dans Amazon S3 pour que vous puissiez les inspecter. Vous pouvez supprimer les anciennes données du cache à l'aide Delete des opérations S3 standard. Vous pouvez également créer une politique de cycle de vie Amazon S3 pour faire expirer les données mises en cache que vous n'utilisez plus.

### Supprimer un cache d'exécution à l'aide de la console

Depuis la console, procédez comme suit pour supprimer un cache d'exécution.

1. Ouvrez la HealthOmics console.
2. Si nécessaire, ouvrez le volet de navigation de gauche (≡). Choisissez Exécuter les caches.
3. Dans le tableau Exécuter les caches, choisissez le cache d'exécution à supprimer.
4. Dans le menu du tableau Exécuter les caches, choisissez Supprimer.
5. Dans la boîte de dialogue modale, enregistrez le lien de données du cache Amazon S3 pour référence future, puis confirmez que vous souhaitez supprimer le cache d'exécution.

   Vous pouvez utiliser le lien Amazon S3 pour inspecter les données mises en cache, mais vous ne pouvez pas les relier à un autre cache d'exécution. Supprimez les données du cache lorsque vous avez terminé l'inspection.

### Suppression d'un cache d'exécution à l'aide de la CLI

Utilisez la commande delete-run-cacheCLI pour supprimer un cache d'exécution.

```
aws omics delete-run-cache \
      --id "my cache id"
```

Si la suppression est réussie, vous recevez une réponse sans champs de données.

## Contenu d'un cache d'exécution

HealthOmics organise votre cache d'exécution selon la structure suivante dans votre compartiment S3 :

```
s3://{cache.S3location}/{cache.uuid}/runID/taskID/{cacheentry.uuid}/
```

Le cache.uuid est l'identifiant unique mondial du cache. Le cacheentry.uuid est l'uuid unique au monde pour une tâche mise en cache. HealthOmics attribue les UUID aux caches et aux tâches.

Pour tous les moteurs de flux de travail, le cache contient les fichiers suivants :

- Le {cacheentryuuid}.json fichier — HealthOmics crée ce fichier manifeste, qui contient des informations sur le cache, y compris une liste de tous les éléments du cache et la version du cache.
- Fichiers de sortie de tâche : chaque sortie de tâche comprend un ou plusieurs fichiers, tels que définis par la tâche.

Pour un flux de travail utilisant Nextflow, le moteur Nextflow crée les fichiers supplémentaires suivants dans le cache :

- Le command.out fichier — Ce fichier contient le contenu standard de l'exécution de la tâche.
- Le .exitcode fichier : ce fichier contient le code de sortie de la tâche (un entier).

> **Note**
>
> Si vous souhaitez accéder aux fichiers de tâches intermédiaires de votre cache d'exécution pour un dépannage avancé, déclarez ces fichiers en tant que résultats de tâches dans la définition du flux de travail.

## Fonctionnalités de mise en cache spécifiques au moteur

HealthOmics essaie de fournir une implémentation cohérente de la mise en cache des appels dans tous les moteurs de flux de travail. Il existe certaines différences selon la façon dont chaque moteur de flux de travail gère des cas spécifiques :

- Flux suivant
  - La mise en cache entre les différentes versions de Nextflow n'est pas garantie. Par exemple, si vous exécutez une tâche dans la version 23.10.0 et que vous exécutez ensuite la même tâche dans la version 24.10.8, vous pouvez HealthOmics considérer la deuxième exécution comme un échec du cache.
  - Vous pouvez désactiver la mise en cache pour des tâches individuelles à l'aide de la false directive cache. Pour plus d'informations sur cette directive, consultez les processus de la spécification Nextflow.
  - HealthOmics utilise le mode indulgent de Nextflow, mais ne prend pas en charge le mode de mise en cache approfondie.
  - La mise en cache évalue chaque objet S3 individuel si vous utilisez un modèle global dans le chemin S3 vers les entrées d'une tâche. Si vous ajoutez un nouvel objet, HealthOmics recalcule uniquement les tâches utilisant le nouvel objet.
  - HealthOmics ne met pas en cache les nouvelles tentatives de tâches. Ce comportement est cohérent avec le comportement par défaut de Nextflow.
- WDL

  - HealthOmics prend en charge le nouveau type de « répertoire » pour les entrées lorsque vous utilisez la version de développement du flux de travail WDL. Pour la mise en cache des appels, si un objet du répertoire change, toutes les tâches entrées dans le répertoire HealthOmics sont recalculées.
  - HealthOmics prend en charge la mise en cache au niveau des tâches, mais pas la mise en cache au niveau du flux de travail.
  - Vous pouvez désactiver la mise en cache pour des tâches individuelles à l'aide de l'attribut volatile. Pour de plus amples informations, veuillez consulter Désactiver la mise en cache au niveau des tâches avec l'attribut volatile.
- CWL
  - Les résultats constants des tâches ne sont pas explicitement visibles dans les manifestes. HealthOmics met en cache les sorties constantes sous forme de fichiers intermédiaires.
  - Vous pouvez contrôler la mise en cache pour des tâches individuelles à l'aide de WorkReusecette fonctionnalité.

## Utilisation du cache d'exécution

Par défaut, les exécutions n'utilisent pas de cache d'exécution. Pour utiliser un cache pour l'exécution, vous devez spécifier le cache d'exécution et le comportement du cache d'exécution lorsque vous démarrez l'exécution.

Une fois l'exécution terminée, vous pouvez utiliser la console, les CloudWatch journaux ou les opérations de l'API pour suivre les accès au cache ou résoudre les problèmes liés au cache. Pour plus d'informations, consultez Suivi des informations de mise en cache des appels et Résolution des problèmes de mise en cache des appels.

Si une ou plusieurs tâches d'une exécution génèrent des résultats non déterministes, nous vous recommandons vivement de ne pas utiliser la mise en cache des appels pour l'exécution ou de désactiver ces tâches spécifiques de la mise en cache. Pour de plus amples informations, veuillez consulter Modèle de responsabilité partagée.

> **Note**
>
> Vous fournissez un rôle de service IAM lorsque vous lancez une exécution. Pour utiliser la mise en cache des appels, le rôle de service doit être autorisé à accéder à l'emplacement du

cache d'exécution Amazon S3. Pour de plus amples informations, veuillez consulter Rôles de service pour AWS HealthOmics.

Vous pouvez utiliser la CLI Amazon Q pour analyser et gérer les données de votre cache d'exécution. Pour plus d'informations, consultez les exemples d'instructions pour Amazon Q CLI et le didacticiel HealthOmics Agentic Generative AI sur. GitHub

Rubriques

- Configuration d'une exécution avec cache d'exécution à l'aide de la console
- Configuration d'une exécution avec cache d'exécution à l'aide de la CLI
- Cas d'erreur pour les caches d'exécution
- Suivi des informations de mise en cache des appels

## Configuration d'une exécution avec cache d'exécution à l'aide de la console

Depuis la console, vous configurez le cache d'exécution pour une exécution lorsque vous démarrez l'exécution.

1. Ouvrez la HealthOmics console.
2. Si nécessaire, ouvrez le volet de navigation de gauche (≡). Choisissez Runs.
3. Sur la page Exécutions, choisissez la course à démarrer.
4. Choisissez Démarrer l'exécution et effectuez les étapes 1 et 2 de Démarrer l'exécution comme décrit dansDémarrer une course à l'aide de la console.
5. À l'étape 3 de Démarrer l'exécution, choisissez Sélectionner un cache d'exécution existant.
6. Sélectionnez le cache dans la liste déroulante Run cache ID.
7. Pour remplacer le comportement de cache d'exécution par défaut, choisissez le comportement de cache pour l'exécution. Pour de plus amples informations, veuillez consulter Exécuter le comportement du cache.
8. Passez à l'étape 4 de Démarrer l'exécution.

## Configuration d'une exécution avec cache d'exécution à l'aide de la CLI

Pour démarrer une exécution utilisant un cache d'exécution, ajoutez le paramètre cache-id à la commande start-run CLI. Utilisez éventuellement le `cache-behavior` paramètre pour remplacer

le comportement par défaut que vous avez configuré pour le cache d'exécution. L'exemple suivant montre uniquement les champs de cache de la commande :

```
aws omics start-run \
        ...
      --cache-id "xxxxxx"    \
      --cache-behavior  CACHE_ALWAYS
```

Si l'opération est réussie, vous recevez une réponse sans champs de données.

## Cas d'erreur pour les caches d'exécution

Dans les scénarios suivants, il est HealthOmics possible que les résultats des tâches ne soient pas mis en cache, même dans le cas d'une exécution avec un comportement de cache défini sur Toujours mettre en cache.

- Si l'exécution rencontre une erreur avant que la première tâche ne soit terminée correctement, il n'y a aucune sortie de cache à exporter.
- Si le processus d'exportation échoue, les résultats de la tâche HealthOmics ne sont pas enregistrés dans l'emplacement du cache Amazon S3.
- Si l'exécution échoue en raison d'une filesystem out of space erreur, la mise en cache des appels n'enregistre aucun résultat de tâche.
- Si vous annulez une exécution, la mise en cache des appels n'enregistre aucun résultat de tâche.
- Si le délai d'exécution est dépassé, la mise en cache des appels n'enregistre aucun résultat de tâche, même si vous avez configuré l'exécution pour utiliser le cache en cas d'échec.

## Suivi des informations de mise en cache des appels

Vous pouvez suivre les événements de mise en cache des appels (tels que les accès au cache d'exécution) à l'aide de la console, de la CLI ou CloudWatch des journaux.

Rubriques

- Suivez les accès au cache à l'aide de la console
- Suivez la mise en cache des appels à l'aide de la CLI
- Suivez la mise en cache des appels à l'aide des journaux CloudWatch

### Suivez les accès au cache à l'aide de la console

Dans la page des détails de l'exécution d'une exécution, le tableau Exécuter les tâches affiche les informations relatives aux accès au cache pour chaque tâche. Le tableau inclut également un lien vers l'entrée de cache associée. Utilisez la procédure suivante pour afficher les informations relatives aux accès au cache lors d'une exécution.

1. Ouvrez la HealthOmics console.
2. Si nécessaire, ouvrez le volet de navigation de gauche (≡). Choisissez Runs.
3. Sur la page Exécutions, choisissez l'exécution à inspecter.
4. Sur la page des détails de l'exécution, choisissez l'onglet Exécuter les tâches pour afficher le tableau des tâches.
5. Si une tâche a un accès au cache, la colonne Accès au cache contient un lien vers l'emplacement d'entrée du cache d'exécution dans Amazon S3.
6. Cliquez sur le lien pour inspecter l'entrée du cache d'exécution.

### Suivez la mise en cache des appels à l'aide de la CLI

Utilisez la commande get-run CLI pour vérifier si l'exécution a utilisé un cache d'appels.

```
aws omics get-run --id 1234567
```

Dans la réponse, si le `cacheId` champ est défini, l'exécution utilise ce cache.

Utilisez la commande list-run-tasksCLI pour récupérer l'emplacement des données du cache pour chaque tâche mise en cache en cours d'exécution.

```
aws omics list-run-tasks --id 1234567
```

Dans la réponse, si le champ CacheHit d'une tâche est vrai, le champ Caches3URI indique l'emplacement des données du cache pour cette tâche.

Vous pouvez également utiliser la commande get-run-taskCLI pour récupérer l'emplacement des données du cache pour une tâche spécifique :

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

Suivez la mise en cache des appels à l'aide des journaux CloudWatch

HealthOmics crée des journaux d'activité du cache dans le groupe de `/aws/omics/WorkflowLog` CloudWatch journaux. <cache_id><cache_uuid>Il existe un flux de journal pour chaque cache d'exécution : RunCache//.

Pour les exécutions utilisant la mise en cache des appels, HealthOmics génère CloudWatch des entrées de journal pour les événements suivants :

- création d'une entrée de cache (CACHE_ENTRY_CREATED)
- correspondant à une entrée de cache (CACHE_HIT)
- ne correspond pas à une entrée de cache (CACHE_MISS)

Pour plus d'informations sur ces journaux, consultezSe connecte CloudWatch.

Utilisez la requête CloudWatch Insights suivante sur le groupe de `/aws/omics/WorkflowLog` journaux pour renvoyer le nombre de visites au cache par exécution pour ce cache :

```
filter @logStream like 'runCache/<CACHE_ID>/'
 fields @timestamp, @message
 filter logMessage like 'CACHE_HIT'
 parse "run: *," as run
 stats count(*) as cacheHits by run
```

Utilisez la requête suivante pour renvoyer le nombre d'entrées de cache créées par chaque exécution :

```
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
```

# Partage HealthOmics de workflows

En tant que propriétaire d'un flux de travail privé, vous pouvez partager le flux de travail avec un Compte AWS utilisateur de la même région. Pour partager un flux de travail avec plusieurs personnes Compte AWS, vous devez créer plusieurs partages du même flux de travail.

En tant que propriétaire, vous pouvez révoquer l'accès à un flux de travail partagé en supprimant le partage.

> **Note**
>
> HealthOmics permet automatiquement à un flux de travail partagé d'accéder au référentiel Amazon ECR pendant que le flux de travail est exécuté dans le compte de l'abonné. Il n'est pas nécessaire d'accorder un accès supplémentaire au référentiel pour les flux de travail partagés.

Lorsque vous partagez un flux de travail, l'abonné peut utiliser n'importe quelle version du flux de travail. Si vous avez besoin d'un contrôle d'accès au niveau des versions pour un flux de travail partagé, nous vous recommandons de créer des flux de travail distincts plutôt que d'utiliser des versions de flux de travail.

Rubriques



## Abonnement à un flux de travail partagé

Pour vous abonner à un flux de travail partagé, vous devez suivre ces étapes générales pour accepter et utiliser le flux de travail :

1. Utilisez la console ou l'API pour accepter le partage. Définissez votre région actuelle sur la même région que celle de la demande de partage.
   - Pour trouver la demande de partage dans la console, accédez à la page Tous les partages de ressources, puis choisissez l'onglet Partagé avec moi.
2. Utilisez la console ou l'API pour créer une exécution pour le flux de travail partagé.

- Pour trouver la page des détails du flux de travail dans la console, accédez à Partagé avec moi (voir étape 1), puis cliquez sur le lien Ressource pour le flux de travail partagé.

3. Vous fournissez vos propres données d'entrée pour le flux de travail.
4. Le flux de travail partagé s'exécute dans votre Compte AWS.

En tant qu'abonné à un flux de travail partagé, le système vous empêche d'effectuer les actions de flux de travail suivantes :

- Exportation d'un flux de travail partagé
- Réexécution du flux de travail partagé
  - Vous créez une nouvelle exécution pour le flux de travail partagé.
- Partage à nouveau du flux de travail.
- Affectation d'une balise au flux de travail.
- Suppression du flux de travail.
  - Lorsque vous n'avez plus besoin du flux de travail, vous supprimez le partage du flux de travail.

Consultez Partage de ressources entre comptes dans AWS HealthOmics pour plus d'informations sur le partage des ressources.

## Surveillance de l'état d'un partage de flux de travail

HealthOmics envoie un événement à chaque changement EventBridge de statut d'un partage de flux de travail. Si vous souhaitez recevoir des notifications concernant des changements de statut spécifiques, configurez une EventBridge règle pour surveiller les événements de changement de statut du partage de flux de travail. Par exemple :

- Vous souhaitez recevoir une notification chaque fois que vous recevez une demande de partage de flux de travail et chaque fois qu'un utilisateur révoque un partage de flux de travail.
- Après avoir lancé une demande de partage de flux de travail, vous souhaitez recevoir une notification lorsque l'utilisateur accepte ou refuse la demande.

Pour plus de détails sur l'utilisation des événements, consultezUtilisation EventBridge avec AWS HealthOmics.

## Partage d'un flux de travail privé à l'aide de la console

Depuis la console, vous pouvez partager un flux de travail privé avec un utilisateur situé Compte AWS dans la même région que le flux de travail.

Pour partager un flux de travail privé

1. Ouvrez la HealthOmics console.
2. Si nécessaire, ouvrez le volet de navigation de gauche (≡). Choisissez des flux de travail privés.
3. Dans le tableau des flux de travail de la page des flux de travail privés, sélectionnez le flux de travail à partager, puis choisissez Partager.
4. Dans le panneau Détails du partage de la page du flux de travail de partage, entrez un nom descriptif pour le partage et entrez le nom Compte AWS de l'abonné.
5. Choisissez Partager la ressource. La console affiche les partages de ressources sur la page Tous les partages de ressources.

L'état initial du partage est en attente. Une fois que l'abonné a accepté le partage, l'état devient actif.

## Partage d'un flux de travail privé à l'aide de la CLI

Utilisez l'opération d'API create-share pour créer un partage de flux de travail. L'abonné principal est celui Compte AWS de l'utilisateur qui aura accès au flux de travail.

```
aws omics create-share \
  --resource-arn "arn:aws:omics:us-west-2:555555555555:workflow/123456" \
  --principal-subscriber "123456789012" \
  --name "my_Share-123"
```

Si la création est réussie, vous recevez une réponse avec l'ID et le statut du partage.

```
{
"shareId": "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a",
"name": "my_Share-123",
"status": "PENDING"
}
```

Le partage reste en attente jusqu'à ce que l'abonné l'accepte à l'aide de l'opération `accept-share` API.

Voir Partage de ressources entre comptes dans AWS HealthOmics pour d'autres exemples d'utilisation de l'API.

## Acceptation d'un flux de travail partagé à l'aide de la console

Vous pouvez utiliser la console pour accepter un partage de flux de travail proposé. Assurez-vous de configurer la console sur la même région que le flux de travail.

1. Ouvrez la HealthOmics console.
2. Si nécessaire, ouvrez le volet de navigation de gauche (≡). Choisissez Tous les partages de ressources, puis choisissez l'onglet Partagé avec moi.
3. Dans le tableau Ressources partagées avec moi, sélectionnez le partage du flux de travail, puis choisissez Accepter.

Après avoir accepté le flux de travail, cliquez sur le lien Ressource du flux de travail partagé pour en afficher les détails.

## Exécution d'un flux de travail partagé à l'aide de la console

Après avoir accepté le partage d'un flux de travail, vous pouvez démarrer une exécution du flux de travail.

1. Ouvrez la HealthOmics console.
2. Si nécessaire, ouvrez le volet de navigation de gauche (≡). Choisissez Tous les partages de ressources, puis choisissez l'onglet Partagé avec moi.
3. Dans le tableau Ressources partagées avec moi, cliquez sur le lien Ressource pour le flux de travail partagé.
4. Sur la page des détails du flux de travail, choisissez Create run.

   La console ouvre la page Créer une exécution, avec le type de flux de travail (partagé) et l'ID de flux de travail préremplis.
5. Configurez les champs restants dans le formulaire Créer une exécution. Pour plus d'informations, consultez Démarrer une course à l'aide de la console.

## Exécution d'un flux de travail partagé à l'aide de l'API

Utilisez get-workflow pour récupérer l'ARN du flux de travail partagé.

```
aws omics get-workflow --id 1234567 \
--workflow-owner-id 55555555555
```

Lorsque vous exécutez le flux de travail, fournissez l' Compte AWS ID du propriétaire du flux de travail et l'ARN du flux de travail partagé.

```
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
```

# Workflows Ready2Run dans HealthOmics

Les flux de travail Ready2Run sont des flux de travail préconfigurés publiés par des éditeurs tiers. Certains éditeurs, tels que Sentieon Inc, proposent des flux de travail basés sur des abonnements. Les autres flux de travail Ready2Run ne nécessitent pas d'abonnement, et certains flux de travail sont open source, tels que les flux de travail NF-Core.

Les flux de travail Ready2Run sont parfaitement adaptés aux scénarios suivants :

- Vous souhaitez vous concentrer sur l'analyse de la sortie du pipeline et sur la génération de résultats, sans avoir à configurer l'infrastructure sous-jacente.
- Vous souhaitez reproduire vos résultats à l'aide de flux de travail établis.
- En tant que développeur de logiciels, vous souhaitez intégrer votre application directement au HealthOmics SDK.

HealthOmics prend en charge la gestion des versions pour les flux de travail Ready2Run. Pour un flux de travail Ready2Run proposant des versions, vous pouvez spécifier le nom de la version lorsque vous démarrez une exécution.

Tous les flux de travail Ready2Run fournissent des journaux, y compris CloudWatch des journaux, que vous pouvez utiliser pour le dépannage.

> **Note**
>
> Les flux de travail Sentineon Ready2Run sont basés sur un abonnement. Lorsque vous exécutez un flux de travail Sentieon Ready2Run pour la première fois dans un compte, Sentieon crée automatiquement une licence d'évaluation de deux semaines pour votre. Compte AWS La licence est valide pour tous les flux de travail Sentieon Ready2Run. Une fois la période d'évaluation terminée, vous pouvez demander une licence permanente ou une extension de la licence d'évaluation. Consultez Subscribing to Sentieon Ready2Run workflows pour plus de détails.

Rubriques

- Démarrer des flux de travail HealthOmics Ready2Run à l'aide de la console
- Démarrer des flux de travail HealthOmics Ready2Run à l'aide de l'API

# Workflows Ready2Run disponibles dans HealthOmics

Le tableau suivant répertorie les flux de travail Ready2Run disponibles dans. HealthOmics

Vous pouvez vous connecter à la HealthOmicsconsole pour consulter des informations détaillées sur ces flux de travail, notamment les paramètres d'entrée et les diagrammes de flux de travail. Pour obtenir des informations sur les tarifs des flux de travail Ready2Run, consultez HealthOmics la section Tarification.

> **Note**
>
> Chaque flux de travail Ready2Run possède une taille de fichier d'entrée maximale. Ces tailles de fichier maximales ne sont pas ajustables.

| Nom du flux de travail | Éditeur | Abonnement requis ? | Taille maximale du fichier d'entrée (GiB) | Durée de fonctionnement estimée (HH:MM) |
|---|---|---|---|---|
| AlphaFold pour 601 à 1200 résidus | Google DeepMind | Non | 1 | 11h15 |
| AlphaFold pour un maximum de 600 résidus | Google DeepMind | Non | 1 | 7 h 30 |
| Bases2Fastq pour 2x150 | Biosciences des éléments | Non | 1 000 | 1:45 |
| Bases2Fastq pour 2x300 | Biosciences des éléments | Non | 1 000 | 1H30 |

| Nom du flux de travail | Éditeur | Abonnement requis ? | Taille maximale du fichier d'entrée (GiB) | Durée de fonctionnement estimée (HH:MM) |
|---|---|---|---|---|
| Bases2Fastq pour 2x75 | Biosciences des éléments | Non | 500 | 0:45 |
| ESMFold pour un maximum de 800 résidus | Méta-recherche | Non | 1 | 0:15 |
| GATK-BP fq2bam | Institut Broad | Non | 64 | 10h10 |
| GATK-BP Germline bam2vcf pour un génome 30x | Institut Broad | Non | 39 | 2:45 |
| Lignée germinale GATK-BP fq2vcf pour le génome 30x | Institut Broad | Non | 64 | 12 h 30 |
| GATK-BP Somatic WES bam2vcf | Institut Broad | Non | 86 | 1H30 |
| NVIDIA Parabricks BAM2 FQ2 BAM WGS jusqu'à 30 fois | NVIDIA Corporation | Non | 80 | 1:39 |

| Nom du flux de travail | Éditeur | Abonnement requis ? | Taille maximale du fichier d'entrée (GiB) | Durée de fonctionnement estimée (HH:MM) |
|---|---|---|---|---|
| NVIDIA Parabricks BAM2 FQ2 BAM WGS jusqu'à 50 fois | NVIDIA Corporation | Non | 120 | 2:45 |
| NVIDIA Parabricks BAM2 FQ2 BAM WGS pour un maximum de 5 fois | NVIDIA Corporation | Non | 20 | 0:18 |
| NVIDIA Parabricks FQ2 BAM WGS jusqu'à 30 fois | NVIDIA Corporation | Non | 71 | 1h00 |
| NVIDIA Parabricks FQ2 BAM WGS jusqu'à 50 fois | NVIDIA Corporation | Non | 137 | 1:45 |
| NVIDIA Parabricks FQ2 BAM WGS pour un maximum de 5 fois | NVIDIA Corporation | Non | 13 | 0:15 |

| Nom du flux de travail | Éditeur | Abonnement requis ? | Taille maximale du fichier d'entrée (GiB) | Durée de fonctionnement estimée (HH:MM) |
|---|---|---|---|---|
| NVIDIA Parabricks Germline DeepVariant WGS jusqu'à 30 fois | NVIDIA Corporation | Non | 71 | 2h00 |
| NVIDIA Parabricks Germline DeepVariant WGS jusqu'à 50 fois | NVIDIA Corporation | Non | 137 | 3h30 |
| NVIDIA Parabricks Germline DeepVariant WGS pour un maximum de 5 fois | NVIDIA Corporation | Non | 12 | 0h30 |
| NVIDIA Parabricks Germline HaplotypeCaller WGS jusqu'à 30 fois | NVIDIA Corporation | Non | 71 | 1:15 |

| Nom du flux de travail | Éditeur | Abonnement requis ? | Taille maximale du fichier d'entrée (GiB) | Durée de fonctionnement estimée (HH:MM) |
|---|---|---|---|---|
| NVIDIA Parabricks Germline HaplotypeCaller WGS jusqu'à 50 fois | NVIDIA Corporation | Non | 137 | 2h00 |
| NVIDIA Parabricks Germline HaplotypeCaller WGS pour un maximum de 5 fois | NVIDIA Corporation | Non | 13 | 0:15 |
| NVIDIA Parabricks Somatic Mutect2 WGS jusqu'à 50 fois | NVIDIA Corporation | Non | 196 | 0:45 |
| sc RNAseq avec Kallisto BUStools | NF-Core | Non | 119 | 1H30 |
| sc RNAseq avec saumon sauté aux alevins | NF-Core | Non | 119 | 2h30 |
| sc RNAseq avec STARsolo | NF-Core | Non | 119 | 2h30 |

| Nom du flux de travail | Éditeur | Abonnement requis ? | Taille maximale du fichier d'entrée (GiB) | Durée de fonctionnement estimée (HH:MM) |
|---|---|---|---|---|
| Sentieon Germline BAM WES jusqu'à 300 fois | Sentieon, Inc. | Oui | 9 | 1h00 |
| Sentieon Germline BAM WGS jusqu'à 32 fois | Sentieon, Inc. | Oui | 18 | 1H30 |
| Sentieon Germline FASTQ WES jusqu'à 100x | Sentieon, Inc. | Oui | 5 | 0:45 |
| Sentieon Germline FASTQ WES jusqu'à 300 fois | Sentieon, Inc. | Oui | 26 | 2h00 |
| Sentieon Germline FASTQ WGS jusqu'à 32 fois | Sentieon, Inc. | Oui | 51 | 3h30 |
| Sentieon LongRead pour ONT | Sentieon, Inc. | Oui | 25 | 1H30 |
| Sentieon LongRead pour PacBio HiFi | Sentieon, Inc. | Oui | 58 | 4h00 |

| Nom du flux de travail | Éditeur | Abonnement requis ? | Taille maximale du fichier d'entrée (GiB) | Durée de fonctionnement estimée (HH:MM) |
|---|---|---|---|---|
| Sentieon Somatic WES | Sentieon, Inc. | Oui | 50 | 2h30 |
| Sentieon Somatic WGS | Sentieon, Inc. | Oui | 113 | 4 h 30 |
| Ultima Genomics jusqu' DeepVariant à 40 fois | Ultima Genomics | Non | 91 | 1:55 |

Lorsque vous utilisez un flux de travail Ready2Run, celui-ci est préconfiguré et ne peut pas être modifié. Contrairement aux flux de travail privés, les flux de travail Ready2Run ne prennent pas en charge les éléments suivants :

- Augmenter la taille maximale du fichier d'entrée
- Modification des ressources de calcul ou du stockage d'exécution
- Modification de la définition du flux de travail ou des conteneurs
- Ajouter des courses à un groupe de courses
- Partage du flux de travail

Si l'éditeur a partagé le flux de travail Ready2Run sur GitHub, vous pouvez créer votre propre flux de travail privé sur la base du flux de travail Ready2Run. Le tableau suivant fournit des liens vers les GitHub flux de travail de chaque éditeur.

| Éditeur | Workflows sur GitHub |
|---|---|
| Google DeepMind, Méta-recherche | Workflows de repliement des protéines |
| Biosciences des éléments | Pour plus d'informations, contactez Element Biosciences |

| Éditeur | Workflows sur GitHub |
| --- | --- |
| Institut Broad | Flux de travail GATK |
| NVIDIA Corporation | Flux de travail Parabricks |
| NF-Core | Flux de travail NF-Core |
| Sentieon | Flux de travail Sentineon |
| Ultima Genomics | Flux de travail Ultima Genomics |

## Abonnement aux flux de travail Sentineon Ready2Run

Les flux de travail Sentineon Ready2Run sont basés sur un abonnement. Lorsque vous exécutez un flux de travail Sentieon Ready2Run pour la première fois dans un compte, Sentieon crée automatiquement une licence d'évaluation de deux semaines pour votre. Compte AWS La licence est valide pour tous les flux de travail Sentieon Ready2Run. Une fois la période d'évaluation terminée, vous pouvez demander une licence permanente ou une extension de la licence d'évaluation.

Suivez ces étapes pour vous abonner aux flux de travail Sentieon Ready2Run :

- Trouvez votre nom d'utilisateur AWS canonique en suivant ces instructions.
- Envoyez un e-mail au groupe de support Sentieon (support@sentieon.com) pour demander une licence logicielle. Indiquez votre nom d'utilisateur AWS canonique dans l'e-mail.

## Démarrer des flux de travail HealthOmics Ready2Run à l'aide de la console

L'utilisation de flux de travail Ready2Run dans la console est similaire à l'utilisation d'un flux de travail privé. L'une des principales différences réside dans le fait que l'éditeur de flux de travail fournit des exemples de données, afin que vous puissiez tester le flux de travail sans créer vos propres données.

Pour utiliser un flux de travail Ready2Run dans la console

1. Ouvrez la HealthOmics console.

2. Si nécessaire, ouvrez le volet de navigation de gauche (≡). Choisissez les flux de travail Ready2Run.
3. Sur la page des flux de travail Ready2Run, choisissez le flux de travail que vous souhaitez utiliser. La console ouvre la page de détails de ce flux de travail.
4. L'onglet Détails répertorie des informations telles que le nom, le prix catalogue par cycle, la description, le type de langue du flux de travail, la capacité de stockage du cycle, le statut, la date de création et les paramètres avec descriptions. L'onglet Détails indique également si le flux de travail nécessite un abonnement.
5. Pour utiliser le flux de travail, choisissez Create run
6. Dans la page Spécifier les détails de l'exécution, entrez un nom d'exécution. Vous pouvez éventuellement spécifier la version du flux de travail. Vous pouvez également ajouter une priorité d'exécution à l'exécution.
7. Entrez ou sélectionnez un emplacement Amazon S3 pour la sortie d'exécution.
8. Pour le mode Exécuter la conservation des métadonnées, choisissez de conserver ou de supprimer les métadonnées d'exécution.
9. Dans le panneau Rôle de service, choisissez d'utiliser un rôle de service existant ou d'en créer un nouveau.
10. (Facultatif) Ajoutez des balises pour identifier et gérer votre course.
11. Choisissez Suivant.
12. Sur la page Ajouter des paramètres, choisissez l'une des options pour ajouter les valeurs des paramètres d'exécution :

    - Sélectionnez un fichier de paramètres (au format JSON) depuis un emplacement Amazon S3.
    - Sélectionnez un fichier de paramètres (au format JSON) sur votre disque local.
    - Entrez manuellement les valeurs des paramètres.
    - Exécutez le flux de travail avec les exemples de données Ready2Run fournis par l'éditeur du flux de travail.

13. Si vous chargez un fichier JSON, la console analyse le fichier et effectue une validation en ligne. Vous pouvez ensuite mettre à jour manuellement les valeurs de vos paramètres selon vos besoins.
14. Choisissez Suivant.
15. Passez en revue vos entrées, puis choisissez Démarrer l'exécution.

# Démarrer des flux de travail HealthOmics Ready2Run à l'aide de l'API

La plupart des opérations d'API se comportent de la même manière pour les flux de travail Ready2Run et les flux de travail privés.

Pour renvoyer une liste des flux de travail Ready2Run disponibles, utilisez list-workflows avec le `type` paramètre défini sur RUN. READY2

```
aws omics list-workflows --type READY2RUN
```

Après avoir identifié le flux de travail à exécuter à partir de la réponse list-workflows, vous pouvez utiliser get-workflow avec le `--id` paramètre pour obtenir plus de détails.

```
aws omics get-workflow --type READY2RUN --id workflow id
```

Pour exécuter un flux de travail Ready2Run, vous pouvez utiliser l'opération d'API Start-Run avec le paramètre de type de flux de travail défini sur`READY2RUN`, comme indiqué dans l'exemple suivant

```
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
```

Pour spécifier une version de flux de travail, utilisez le paramètre workflow-version, comme illustré dans cet exemple.

```
aws-omics start-run \
  --workflow-type READY2RUN \
  ...
  --version-name '3.0.0'
```

Pour surveiller votre course, vous pouvez utiliser l'opération d'API get-run, comme indiqué.

```
aws-omics get-run \
```

```
  --id run id
```

# HealthOmics rangement

Utilisez le HealthOmics stockage pour stocker, récupérer, organiser et partager les données génomiques de manière efficace et à moindre coût. HealthOmics le stockage comprend les relations entre les différents objets de données, ce qui vous permet de définir quels ensembles de lecture proviennent des mêmes données sources. Cela vous permet de connaître la provenance des données.

Les données stockées dans `ACTIVE` l'état sont immédiatement récupérables. Les données qui n'ont pas été consultées depuis 30 jours ou plus sont stockées dans leur `ARCHIVE` état actuel. Pour accéder aux données archivées, vous pouvez les réactiver via les opérations ou la console de l'API.

HealthOmics les magasins de séquences sont conçus pour préserver l'intégrité du contenu des fichiers. Cependant, l'équivalence au niveau du bit entre les fichiers de données importés et les fichiers exportés n'est pas préservée en raison de la compression lors de la hiérarchisation active et de la hiérarchisation des archives.

Lors de l'ingestion, HealthOmics génère une balise d'entité HealthOmics ETag, ou pour permettre de valider l'intégrité du contenu de vos fichiers de données. Les parties de séquençage sont identifiées et capturées ETag au niveau de la source d'un ensemble de lecture. Le ETag calcul ne modifie pas le fichier réel ni les données génomiques. Une fois qu'un jeu de lecture est créé, il ETag ne devrait pas changer tout au long du cycle de vie de la source du jeu de lecture. Cela signifie que la réimportation du même fichier entraîne le calcul de la même ETag valeur.

Rubriques

# HealthOmics ETags et provenance des données

Une HealthOmics ETag (étiquette d'entité) est un hachage du contenu ingéré dans un magasin de séquences. Cela simplifie la récupération et le traitement des données tout en préservant l'intégrité du contenu des fichiers de données ingérés. Cela ETag reflète les modifications apportées au contenu sémantique de l'objet, et non à ses métadonnées. Le type d'ensemble de lecture et l'algorithme spécifiés déterminent le ETag mode de calcul. Le ETag calcul ne modifie pas le fichier réel ni les données génomiques. Lorsque le schéma de type de fichier de l'ensemble de lecture le permet, le magasin de séquences met à jour les champs liés à la provenance des données.

Les fichiers ont une identité binaire et une identité sémantique. L'identité binaire signifie que les bits d'un fichier sont identiques, et l'identité sémantique signifie que le contenu d'un fichier est identique. L'identité sémantique résiste aux modifications des métadonnées et aux modifications de compression car elle capture l'intégrité du contenu du fichier.

Les ensembles de lecture placés dans des magasins de HealthOmics séquences sont soumis à compression/decompression des cycles et à un suivi de la provenance des données tout au long du cycle de vie d'un objet. Au cours de ce traitement, l'identité binaire d'un fichier ingéré peut changer et devrait changer chaque fois qu'un fichier est activé ; toutefois, l'identité sémantique du fichier est conservée. L'identité sémantique est capturée sous forme de balise d' HealthOmics entité, ou ETag calculée lors de l'ingestion du magasin de séquences et disponible sous forme de métadonnées d'ensemble de lecture.

Lorsque le schéma de type de fichier de l'ensemble de lecture le permet, les champs de mise à jour du magasin de séquences sont liés à la provenance des données. Pour les fichiers uBam, BAM et CRAM, une nouvelle `Comment` balise `@CO` or est ajoutée à l'en-tête. Le commentaire contient l'identifiant du magasin de séquences et l'horodatage d'ingestion.

## Amazon S3 ETags

Lorsque vous accédez à un fichier à l'aide de l'URI Amazon S3, les opérations de l'API Amazon S3 peuvent également renvoyer des valeurs Amazon S3 ETag et des valeurs de somme de contrôle. Les valeurs d'Amazon S3 ETag et de checksum diffèrent de celles-ci HealthOmics ETags car elles représentent l'identité binaire du fichier. Pour en savoir plus sur les métadonnées descriptives et les objets, consultez la documentation de l'API Amazon S3 Object. ETag Les valeurs Amazon S3 peuvent changer à chaque cycle d'activation d'un ensemble de lecture et vous pouvez les utiliser pour valider la lecture d'un fichier. Cependant, ne mettez pas en cache ETag les valeurs Amazon S3 à utiliser pour la validation de l'identité du fichier pendant le cycle de vie du fichier, car elles ne restent

pas cohérentes. En revanche, ils HealthOmics ETag restent cohérents tout au long du cycle de vie du jeu de lecture.

## Comment HealthOmics calcule ETags

Le ETag est généré à partir d'un hachage du contenu du fichier ingéré. La famille d' ETag algorithmes est définie sur MD5up par défaut, mais elle peut être configurée différemment lors de la création du magasin de séquences. Lorsque le ETag est calculé, l'algorithme et les hachages calculés sont ajoutés à l'ensemble de lecture. Les MD5 algorithmes pris en charge pour les types de fichiers sont les suivants.

- FASTQ_ MD5up — Calcule le MD5 hachage d'une source d'ensemble de lecture FASTQ complète et non compressée.
- BAM_ MD5up — Calcule le MD5 hachage de la section d'alignement d'une source de jeu de lecture BAM ou UbAM non compressée telle que représentée dans le SAM, sur la base de la référence liée, le cas échéant.
- CRAM_ MD5up — Calcule le MD5 hachage de la section d'alignement de la source du jeu de lecture CRAM non compressée telle que représentée dans le SAM, sur la base de la référence liée.

> **Note**
>
> MD5 le hachage est connu pour être vulnérable aux collisions. De ce fait, deux fichiers différents peuvent avoir les mêmes caractéristiques ETag s'ils ont été fabriqués pour exploiter la collision connue.

Les algorithmes suivants sont pris en charge pour la SHA256 famille. Les algorithmes sont calculés comme suit :

- FASTQ_ SHA256up — Calcule le hachage SHA-256 d'une source de jeu de lecture FASTQ complète et non compressée.
- BAM_ SHA256up — Calcule le hachage SHA-256 de la section d'alignement d'un ensemble de lecture BAM ou UbAM non compressé tel que représenté dans le SAM, sur la base de la référence liée, le cas échéant.
- CRAM_ SHA256up — Calcule le hachage SHA-256 de la section d'alignement d'une source de jeu de lecture CRAM non compressée telle que représentée dans le SAM, sur la base de la référence liée.

Les algorithmes suivants sont pris en charge pour la SHA512 famille. Les algorithmes sont calculés comme suit :

- FASTQ_ SHA512up — Calcule le hachage SHA-512 d'une source de jeu de lecture FASTQ complète et non compressée.
- BAM_ SHA512up — Calcule le hachage SHA-512 de la section d'alignement d'un ensemble de lecture BAM ou UbAM non compressé tel que représenté dans le SAM, sur la base de la référence liée, le cas échéant.
- CRAM_ SHA512up — Calcule le hachage SHA-512 de la section d'alignement d'une source de jeu de lecture CRAM non compressée telle que représentée dans le SAM, sur la base de la référence liée.

# Création d'un magasin HealthOmics de référence

Un magasin de référence dans HealthOmics est un magasin de données destiné au stockage de génomes de référence. Vous pouvez avoir un seul magasin de référence dans chaque Compte AWS région. Vous pouvez créer un magasin de référence à l'aide de la console ou de la CLI.

Rubriques



## Création d'un magasin de référence à l'aide de la console

Pour créer un magasin de références

1. Ouvrez la HealthOmics console.
2. Si nécessaire, ouvrez le volet de navigation de gauche (≡). Choisissez Reference store.
3. Choisissez Génomes de référence dans les options de stockage des données génomiques.
4. Vous pouvez soit choisir un génome de référence précédemment importé, soit en importer un nouveau. Si vous n'avez pas importé de génome de référence, choisissez Importer le génome de référence en haut à droite.
5. Sur la page Créer une tâche d'importation du génome de référence, choisissez l'option Création rapide ou Création manuelle pour créer un magasin de référence, puis fournissez les informations suivantes.

- Nom du génome de référence : nom unique pour ce magasin.
- Description (facultatif) : description de ce magasin de référence.
- Rôle IAM - Sélectionnez un rôle ayant accès à votre génome de référence.
- Référence provenant d'Amazon S3 : sélectionnez votre fichier de séquence de référence dans un compartiment Amazon S3.
- Balises (facultatif) - Fournissez jusqu'à 50 balises pour ce magasin de référence.

## Création d'un magasin de référence à l'aide de la CLI

L'exemple suivant montre comment créer un magasin de référence à l'aide du AWS CLI. Vous pouvez avoir un magasin de référence par AWS région.

Les magasins de référence prennent en charge le stockage de fichiers FASTA avec les extensions `.fasta .fa.fas,.fsa,,.faa,.fna,.ffn,.frn,, .mpfa.seq,.txt`. La `bgzip` version de ces extensions est également prise en charge.

Dans l'exemple suivant, remplacez *`reference store name`* par le nom que vous avez choisi pour votre boutique de référence.

```
aws omics create-reference-store --name "reference store name"
```

Vous recevez une réponse JSON avec l'ID et le nom du magasin de référence, l'ARN et l'horodatage de la création de votre magasin de référence.

```
{
    "id": "3242349265",
    "arn": "arn:aws:omics:us-west-2:555555555555:referenceStore/3242349265",
    "name": "MyReferenceStore",
    "creationTime": "2022-07-01T20:58:42.878Z"
}
```

Vous pouvez utiliser l'ID du magasin de référence dans des AWS CLI commandes supplémentaires. Vous pouvez récupérer la liste des magasins de référence IDs liés à votre compte à l'aide de la list-reference-storescommande, comme illustré dans l'exemple suivant.

```
aws omics list-reference-stores
```

En réponse, vous recevez le nom du magasin de référence que vous venez de créer.

```
{
    "referenceStores": [
        {
              "id": "3242349265",
              "arn": "arn:aws:omics:us-west-2:555555555555:referenceStore/3242349265",
              "name": "MyReferenceStore",
             "creationTime": "2022-07-01T20:58:42.878Z"
         }
     ]
}
```

Après avoir créé un magasin de référence, vous pouvez créer des tâches d'importation pour y charger des fichiers de référence génomiques. Pour ce faire, vous devez utiliser ou créer un rôle IAM pour accéder aux données. Voici un exemple de politique .

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/*"
            ]
        }
    ]
}
```

Vous devez également disposer d'une politique de confiance similaire à celle décrite dans l'exemple suivant.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": [
                   "omics.amazonaws.com"
                ]
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

Vous pouvez désormais importer un génome de référence. Cet exemple utilise Human Build 38 (hg38) du Genome Reference Consortium, qui est en libre accès et disponible sur AWS le registre des données ouvertes le. Le bucket qui héberge ces données est basé dans l'est des États-Unis (Ohio). Pour utiliser des compartiments dans d'autres AWS régions, vous pouvez copier les données dans un compartiment Amazon S3 hébergé dans votre région. Utilisez la AWS CLI commande suivante pour copier le génome dans votre compartiment Amazon S3.

```
aws s3 cp s3://broad-references/hg38/v0/Homo_sapiens_assembly38.fasta s3://amzn-s3-
demo-bucket
```

Vous pouvez ensuite commencer votre tâche d'importation. Remplacez *reference store IDrole ARN*, et *source file path* par votre propre entrée.

```
aws omics start-reference-import-job --reference-store-id reference store ID --role-
arn role ARN --sources source file path
```

Une fois les données importées, vous recevez la réponse suivante au format JSON.

```
{
        "id": "7252016478",
        "referenceStoreId": "3242349265",
        "roleArn": "arn:aws:iam::111122223333:role/OmicsReferenceImport",
```

```
        "status": "CREATED",
        "creationTime": "2022-07-01T21:15:13.727Z"
}
```

Vous pouvez contrôler l'état d'une tâche à l'aide de la commande suivante. Dans l'exemple suivant, remplacez *reference store ID* et *job ID* par votre identifiant de magasin de référence et l'identifiant de tâche sur lesquels vous souhaitez en savoir plus.

```
aws omics get-reference-import-job --reference-store-id reference store ID --id job ID
```

En réponse, vous recevez une réponse contenant les détails de ce magasin de référence et son statut.

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

Vous pouvez également trouver la référence importée en répertoriant vos références et en les filtrant en fonction du nom de la référence. *reference store ID*Remplacez-le par l'identifiant de votre boutique de référence et ajoutez un filtre facultatif pour affiner la liste.

```
aws omics list-references --reference-store-id reference store ID --filter
 name=MyReference
```

En réponse, vous recevez les informations suivantes.

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

Pour en savoir plus sur les métadonnées de référence, utilisez l'opération get-reference-metadataAPI. Dans l'exemple suivant, remplacez-le *reference store ID* par l'ID de votre boutique de référence et *reference ID* par l'ID de référence sur lequel vous souhaitez en savoir plus.

```
aws omics get-reference-metadata --reference-store-id reference store ID --id reference
 ID
```

Vous recevez les informations suivantes en réponse.

```
{
    "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
        }
    }
}
```

Vous pouvez également télécharger des parties du fichier de référence à l'aide de get-reference. Dans l'exemple suivant, remplacez-le *reference store ID* par l'ID de votre magasin de référence et *reference ID* par l'ID de référence à partir duquel vous souhaitez effectuer le téléchargement.

```
aws omics get-reference --reference-store-id reference store ID --id reference ID --
part-number 1 outfile.fa
```

# Création d'un magasin HealthOmics de séquences

HealthOmics les magasins de séquences prennent en charge le stockage de fichiers génomiques dans les formats non alignés FASTQ (gzip uniquement) et. uBAM Il prend également en charge les formats alignés de BAM etCRAM.

Les fichiers importés sont stockés sous forme de jeux de lecture. Vous pouvez ajouter des balises aux ensembles de lecture et utiliser les politiques IAM pour contrôler l'accès aux ensembles de lecture. Les ensembles de lecture alignés nécessitent un génome de référence pour aligner les séquences génomiques, mais c'est facultatif pour les ensembles de lecture non alignés.

Pour stocker des ensembles de lecture, vous devez d'abord créer un magasin de séquences. Lorsque vous créez un magasin de séquences, vous pouvez spécifier un compartiment Amazon S3 facultatif comme emplacement de secours et comme emplacement de stockage des journaux d'accès S3. L'emplacement de secours est utilisé pour stocker tous les fichiers qui ne parviennent pas à créer un ensemble de lecture lors d'un téléchargement direct. Des emplacements de secours sont disponibles pour les magasins de séquences créés après le 15 mai 2023. Vous spécifiez l'emplacement de secours lorsque vous créez le magasin de séquences.

Vous pouvez spécifier jusqu'à cinq clés de balise Read Set. Lorsque vous créez ou mettez à jour un ensemble de lecture avec une clé de balise correspondant à l'une de ces clés, les balises d'ensemble de lecture sont propagées à l'objet Amazon S3 correspondant. Les balises système créées par HealthOmics sont propagées par défaut.

Rubriques

## Création d'un magasin de séquences à l'aide de la console

Pour créer un magasin de séquences

1. Ouvrez la HealthOmics console.
2. Si nécessaire, ouvrez le volet de navigation de gauche (≡). Choisissez Sequence stores.
3. Sur la page Créer un magasin de séquences, fournissez les informations suivantes

   - Nom du magasin de séquences : nom unique pour ce magasin.
   - Description (facultatif) : description de ce magasin de séquences.
4. Pour l'emplacement de secours dans S3, spécifiez un emplacement Amazon S3. HealthOmics utilise l'emplacement de secours pour stocker tous les fichiers qui ne parviennent pas à créer un ensemble de lectures lors d'un téléchargement direct. Vous devez accorder au HealthOmics service un accès en écriture à l'emplacement de secours d'Amazon S3. Pour un exemple de politique, consultez Configuration d'un emplacement de secours.

   Les emplacements de secours ne sont pas disponibles pour les magasins de séquences créés avant le 16 mai 2023.
5. (Facultatif) Pour les clés de balise Read Set pour la propagation S3, vous pouvez entrer jusqu'à cinq clés de lecture à propager d'un ensemble de lectures aux objets S3 sous-jacents. En propageant les balises d'un ensemble de lecture vers l'objet S3, vous pouvez accorder des autorisations d'accès à S3 en fonction des balises. Les utilisateurs and/or finaux peuvent ainsi voir les balises propagées via l'opération d' getObjectTagging API Amazon S3.

   a. Entrez une valeur clé dans la zone de texte. La console crée une nouvelle zone de texte pour ajouter la touche suivante.

   b. (Facultatif) Choisissez Supprimer pour supprimer toutes les clés.
6. Sous Chiffrement des données, indiquez si vous souhaitez que le chiffrement des données soit détenu et géré par AWS ou qu'il utilise une clé CMK gérée par le client.

7. (Facultatif) Sous Accès aux données S3, indiquez si vous souhaitez créer un nouveau rôle et une nouvelle politique pour accéder au magasin de séquences via Amazon S3.
8. (Facultatif) Pour la journalisation des accès S3, indiquez `Enabled` si vous souhaitez qu'Amazon S3 collecte les enregistrements des journaux d'accès.

   Pour l'emplacement de journalisation des accès dans S3, spécifiez un emplacement Amazon S3 pour stocker les journaux. Ce champ n'est visible que si vous avez activé la journalisation des accès S3.
9. Balises (facultatif) - Fournissez jusqu'à 50 balises pour ce magasin de séquences. Ces balises sont distinctes des balises d'ensemble de lecture définies lors de la mise à import/tag jour de l'ensemble de lecture

Une fois que vous avez créé le magasin, il est prêt pourImportation de fichiers génomiques.

## Création d'un magasin de séquences à l'aide de la CLI

Dans l'exemple suivant, remplacez *`sequence store name`* par le nom que vous avez choisi pour votre magasin de séquences.

```
aws omics create-sequence-store --name sequence store name --fallback-location "s3://
amzn-s3-demo-bucket"
```

Vous recevez la réponse suivante au format JSON, qui inclut le numéro d'identification du magasin de séquences que vous venez de créer.

```
{
    "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"
}
```

Vous pouvez également afficher tous les magasins de séquences associés à votre compte à l'aide de la list-sequence-storescommande, comme indiqué ci-dessous.

```
aws omics list-sequence-stores
```

Vous recevez la réponse suivante.

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

Vous pouvez utiliser get-sequence-storepour en savoir plus sur un magasin de séquences en utilisant son identifiant, comme illustré dans l'exemple suivant :

```
aws omics get-sequence-store --id sequence store ID
```

Vous recevez la réponse suivante :

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

Après la création, plusieurs paramètres de la boutique peuvent également être mis à jour. Cela peut être effectué via la console ou l'`updateSequenceStore`opération API.

## Mettre à jour un magasin de séquences

Pour mettre à jour un magasin de séquences, procédez comme suit :

1. Ouvrez la HealthOmics console.
2. Si nécessaire, ouvrez le volet de navigation de gauche (≡). Choisissez Sequence stores.
3. Choisissez le magasin de séquences à mettre à jour.
4. Dans le panneau Détails, choisissez Modifier.
5. Sur la page Modifier les détails, vous pouvez mettre à jour les champs suivants :

   - Nom du magasin de séquences : nom unique pour ce magasin.
   - Description : description de ce magasin de séquences.
   - Emplacement de secours dans S3, spécifiez un emplacement Amazon S3. HealthOmics utilise l'emplacement de secours pour stocker tous les fichiers qui ne parviennent pas à créer un ensemble de lectures lors d'un téléchargement direct.
   - Clés de balise Read Set pour la propagation dans S3 Vous pouvez saisir jusqu'à cinq clés de lecture à propager vers Amazon S3.
   - (Facultatif) Pour la journalisation des accès S3, indiquez `Enabled` si vous souhaitez qu'Amazon S3 collecte les enregistrements des journaux d'accès.

     Pour l'emplacement de journalisation des accès dans S3, spécifiez un emplacement Amazon S3 pour stocker les journaux. Ce champ n'est visible que si vous avez activé la journalisation des accès S3.
   - Balises (facultatif) - Fournissez jusqu'à 50 balises pour ce magasin de séquences.

## Mise à jour des balises de jeu de lecture pour un magasin de séquences

Pour mettre à jour les balises de jeu de lecture ou les autres champs d'un magasin de séquences, procédez comme suit :

1. Ouvrez la HealthOmics console.
2. Si nécessaire, ouvrez le volet de navigation de gauche (≡). Choisissez Sequence stores.
3. Choisissez le magasin de séquences que vous souhaitez mettre à jour.
4. Cliquez sur l'onglet Détails.
5. Choisissez Modifier.
6. Ajoutez de nouvelles balises readset ou supprimez des balises existantes, selon les besoins.
7. Mettez à jour le nom, la description, l'emplacement de secours ou l'accès aux données S3, selon les besoins.
8. Sélectionnez Enregistrer les modifications.

## Importation de fichiers génomiques

Pour importer des fichiers génomiques dans un magasin de séquences, procédez comme suit :

Pour importer un fichier génomique

1. Ouvrez la HealthOmics console.
2. Si nécessaire, ouvrez le volet de navigation de gauche (≡). Choisissez Choisissez Sequence stores.
3. Sur la page Magasins de séquences, choisissez le magasin de séquences dans lequel vous souhaitez importer vos fichiers.
4. Sur la page du magasin de séquences individuelles, choisissez Importer des fichiers génomiques.
5. Sur la page Spécifier les détails de l'importation, fournissez les informations suivantes
   - Rôle IAM : rôle IAM qui peut accéder aux fichiers génomiques sur Amazon S3.
   - Génome de référence - Le génome de référence pour ces données génomiques.
6. Sur la page Spécifier le manifeste d'importation, spécifiez le fichier manifeste d'informations suivant. Le fichier manifeste est un fichier JSON ou YAML qui décrit les informations essentielles de vos données génomiques. Pour plus d'informations sur le fichier manifeste, consultezImportation de jeux de lecture dans un magasin de HealthOmics séquences.
7. Cliquez sur Créer une tâche d'importation.

# Suppression des HealthOmics référentiels et des magasins de séquences

Les magasins de référence et de séquences peuvent être supprimés. Les magasins de séquences ne peuvent être supprimés que s'ils ne contiennent pas de jeux de lecture, et les magasins de référence ne peuvent être supprimés que s'ils ne contiennent pas de références. La suppression d'une séquence ou d'un magasin de référence supprime également toutes les balises associées à ce magasin.

L'exemple suivant montre comment supprimer un magasin de référence à l'aide du AWS CLI. Si l'action aboutit, vous ne recevrez pas de réponse. Dans l'exemple suivant, remplacez par *reference store ID* l'ID de votre boutique de référence.

```
aws omics delete-reference-store --id reference store ID
```

L'exemple suivant montre comment supprimer un magasin de séquences. Vous ne recevez pas de réponse si l'action aboutit. Dans l'exemple suivant, remplacez-le *sequence store ID* par votre identifiant de magasin de séquences.

```
aws omics delete-sequence-store --id sequence store ID
```

Vous pouvez également supprimer une référence dans un magasin de références, comme indiqué dans l'exemple suivant. Les références ne peuvent être supprimées que si elles ne sont pas utilisées dans un ensemble de lectures, un magasin de variantes ou un magasin d'annotations. Dans l'exemple suivant, remplacez *reference store ID* par l'ID de votre magasin de référence et remplacez *reference ID* par l'ID de la référence que vous souhaitez supprimer.

```
aws omics delete-reference  --id reference ID --reference-store-id reference store ID
```

# Importation de jeux de lecture dans un magasin de HealthOmics séquences

Après avoir créé votre magasin de séquences, créez des tâches d'importation pour télécharger des ensembles de lectures dans le magasin de données. Vous pouvez charger vos fichiers depuis un

compartiment Amazon S3, ou vous pouvez les télécharger directement à l'aide des opérations d'API synchrones. Votre compartiment Amazon S3 doit se trouver dans la même région que votre magasin de séquences.

Vous pouvez télécharger n'importe quelle combinaison d'ensembles de lecture alignés et non alignés dans votre magasin de séquences. Toutefois, si l'un des ensembles de lecture de votre importation est aligné, vous devez inclure un génome de référence.

Vous pouvez réutiliser la politique d'accès IAM que vous avez utilisée pour créer le magasin de référence.

Les rubriques suivantes décrivent les principales étapes à suivre pour importer un jeu de lectures dans votre magasin de séquences, puis obtenir des informations sur les données importées.

Rubriques



## Charger des fichiers sur Amazon S3

L'exemple suivant montre comment déplacer des fichiers dans votre compartiment Amazon S3.

```
aws s3 cp s3://1000genomes/phase1/data/HG00100/alignment/
HG00100.chrom20.ILLUMINA.bwa.GBR.low_coverage.20101123.bam s3://your-bucket
aws s3 cp s3://1000genomes/phase3/data/HG00146/sequence_read/SRR233106_1.filt.fastq.gz
 s3://your-bucket
aws s3 cp s3://1000genomes/phase3/data/HG00146/sequence_read/SRR233106_2.filt.fastq.gz
 s3://your-bucket
aws s3 cp s3://1000genomes/data/HG00096/alignment/
HG00096.alt_bwamem_GRCh38DH.20150718.GBR.low_coverage.cram s3://your-bucket
aws s3 cp s3://gatk-test-data/wgs_ubam/NA12878_20k/NA12878_A.bam s3://your-bucket
```

L'échantillon BAM CRAM utilisé dans cet exemple nécessite des références génomiques différentes, Hg19 etHg38. Pour en savoir plus ou pour accéder à ces références, voir The Broad Genome References dans le registre des données ouvertes sur AWS.

## Création d'un fichier manifeste

Vous devez également créer un fichier manifeste au format JSON pour modéliser la tâche d'importation `import.json` (voir l'exemple suivant). Si vous créez un magasin de séquences dans la console, il n'est pas nécessaire de spécifier le `sequenceStoreId` ou`roleARN`. Votre fichier manifeste commence donc par l'`sources`entrée.

API manifest

L'exemple suivant importe trois ensembles de lecture à l'aide de l'API : un FASTQBAM, un et unCRAM.

```
{
  "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

Cet exemple de code est utilisé pour importer un seul ensemble de lectures à l'aide de la console.

```
[
```

```
  {
      "sourceFiles":
      {
          "source1": "s3://amzn-s3-demo-bucket/
HG00100.chrom20.ILLUMINA.bwa.GBR.low_coverage.20101123.bam"
      },
      "sourceFileType": "BAM",
      "subjectId": "mySubject",
      "sampleId": "mySample",
      "name": "HG00100",
      "description": "BAM for HG00100",
      "generatedFrom": "1000 Genomes"
  },
  {
      "sourceFiles":
      {
          "source1": "s3://amzn-s3-demo-bucket/SRR233106_1.filt.fastq.gz",
          "source2": "s3://amzn-s3-demo-bucket/SRR233106_2.filt.fastq.gz"
      },
      "sourceFileType": "FASTQ",
      "subjectId": "mySubject",
      "sampleId": "mySample",
      "name": "HG00146",
      "description": "FASTQ for HG00146",
      "generatedFrom": "1000 Genomes"
  },
  {
      "sourceFiles":
      {
          "source1": "s3://your-bucket/
HG00096.alt_bwamem_GRCh38DH.20150718.GBR.low_coverage.cram"
      },
      "sourceFileType": "CRAM",
      "subjectId": "mySubject",
      "sampleId": "mySample",
      "name": "HG00096",
      "description": "CRAM for HG00096",
      "generatedFrom": "1000 Genomes"
  },
  {
      "sourceFiles":
      {
          "source1": "s3://amzn-s3-demo-bucket/NA12878_A.bam"
      },
```

```
      "sourceFileType": "UBAM",
      "subjectId": "mySubject",
      "sampleId": "mySample",
      "name": "NA12878_A",
      "description": "uBAM for NA12878",
      "generatedFrom": "GATK Test Data"
  }
]
```

Vous pouvez également télécharger le fichier manifeste au format YAML.

## Démarrage de la tâche d'importation

Pour démarrer la tâche d'importation, utilisez la AWS CLI commande suivante.

```
aws omics start-read-set-import-job --cli-input-json file://import.json
```

Vous recevez la réponse suivante, qui indique une création d'emploi réussie.

```
{
  "id": "3660451514",
  "sequenceStoreId": "3936421177",
  "roleArn": "arn:aws:iam::111122223333:role/OmicsImport",
  "status": "CREATED",
  "creationTime": "2022-07-13T22:14:59.309Z"
}
```

## Surveiller la tâche d'importation

Une fois la tâche d'importation lancée, vous pouvez suivre sa progression à l'aide de la commande suivante. Dans l'exemple suivant, remplacez *sequence store id* par l'ID de votre magasin de séquences et remplacez *job import ID* par l'ID d'importation.

```
aws omics get-read-set-import-job --sequence-store-id sequence store id --id job import
 ID
```

Voici les statuts de toutes les tâches d'importation associées à l'ID de magasin de séquences spécifié.

```
{
```

```
  "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"
      }
  ]
}
```

## Trouvez les fichiers de séquence importés

Une fois le travail terminé, vous pouvez utiliser l'opération list-read-setsAPI pour rechercher les fichiers de séquence importés. Dans l'exemple suivant, remplacez-le *`sequence store id`* par votre identifiant de magasin de séquences.

```
aws omics list-read-sets --sequence-store-id sequence store id
```

Vous recevez la réponse suivante.

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

## Obtenir des informations sur un kit de lecture

Pour afficher plus de détails sur un ensemble de lectures, utilisez l'opération GetReadSetMetadataAPI. Dans l'exemple suivant, remplacez *sequence store id* par votre identifiant de magasin de séquences et remplacez *read set id* par votre identifiant de jeu de lecture.

```
aws omics get-read-set-metadata --sequence-store-id sequence store id --id read set id
```

Vous recevez la réponse suivante.

```
{
"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"
}
```


## Téléchargez les fichiers de données du jeu de lecture

Vous pouvez accéder aux objets d'un ensemble de lecture actif à l'aide de l'opération d'GetObjectAPI Amazon S3. L'URI de l'objet est renvoyé dans la réponse de l'GetReadSetMetadataAPI. Pour de plus

amples informations, veuillez consulter Accès aux ensembles de HealthOmics lecture avec Amazon S3 URIs.

Vous pouvez également utiliser l'opération HealthOmics GetReadSet API. Vous pouvez GetReadSet utiliser le téléchargement en parallèle en téléchargeant des parties individuelles. Ces composants sont similaires aux composants Amazon S3. Voici un exemple de téléchargement de la partie 1 à partir d'un jeu de lecture. Dans l'exemple suivant, remplacez *sequence store id* par votre identifiant de magasin de séquences et remplacez *read set id* par votre identifiant de jeu de lecture.

```
aws omics get-read-set --sequence-store-id sequence store id --id read set id  --part-
number 1 outfile.bam
```

Vous pouvez également utiliser le gestionnaire de HealthOmics transfert pour télécharger des fichiers à des fins de HealthOmics référence ou de lecture. Vous pouvez télécharger le gestionnaire HealthOmics de transferts ici. Pour plus d'informations sur l'utilisation et la configuration du gestionnaire de transfert, consultez ce GitHubréférentiel.

# Téléchargement direct vers un magasin de HealthOmics séquences

Nous vous recommandons d'utiliser le gestionnaire HealthOmics de transfert pour ajouter des fichiers à votre magasin de séquences. Pour plus d'informations sur l'utilisation de Transfer Manager, consultez ce GitHubréférentiel. Vous pouvez également télécharger vos ensembles de lecture directement dans un magasin de séquences via les opérations d'API de téléchargement direct.

Les ensembles de lecture à téléchargement direct existent en `PROCESSING_UPLOAD` premier. Cela signifie que les parties du fichier sont en cours de téléchargement et que vous pouvez accéder aux métadonnées du jeu de lecture. Une fois les parties téléchargées et les checksums validés, le jeu de lecture devient `ACTIVE` et se comporte de la même manière qu'un jeu de lecture importé.

Si le téléchargement direct échoue, l'état du jeu de lecture est affiché sous la forme`UPLOAD_FAILED`. Vous pouvez configurer un compartiment Amazon S3 comme emplacement de secours pour les fichiers qui ne parviennent pas à être chargés. Des emplacements de secours sont disponibles pour les magasins de séquences créés après le 15 mai 2023.

Rubriques

## Téléchargement direct vers un magasin de séquences à l'aide du AWS CLI

Pour commencer, lancez un téléchargement en plusieurs parties. Vous pouvez le faire en utilisant le AWS CLI, comme indiqué dans l'exemple suivant.

Pour effectuer un téléchargement direct à l'aide de AWS CLI commandes

1. Créez les parties en séparant vos données, comme indiqué dans l'exemple suivant.

```
split -b 100MiB SRR233106_1.filt.fastq.gz source1_part_
```

2. Une fois que vos fichiers source sont divisés en plusieurs parties, créez un téléchargement de jeux de lecture en plusieurs parties, comme indiqué dans l'exemple suivant. Remplacez *sequence store ID* les autres paramètres par votre identifiant de magasin de séquences et d'autres valeurs.

```
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"
```

Vous obtenez les métadonnées `uploadID` et les autres dans la réponse. Utilisez le `uploadID` pour l'étape suivante du processus de téléchargement.

```
{
"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. Ajoutez vos ensembles de lecture au téléchargement. Si votre fichier est suffisamment petit, vous ne devez effectuer cette étape qu'une seule fois. Pour les fichiers plus volumineux, vous devez effectuer cette étape pour chaque partie du fichier. Si vous chargez une nouvelle pièce en utilisant un numéro de pièce déjà utilisé, il remplace la pièce précédemment téléchargée.

   Dans l'exemple suivant, remplacez *sequence store IDupload ID*, et les autres paramètres par vos valeurs.

   ```
   aws omics upload-read-set-part \
   --sequence-store-id sequence store ID \
   --upload-id upload ID \
   --part-source SOURCE1 \
   --part-number part number \
   --payload  source1/source1_part_aa.fastq.gz
   ```

   La réponse est un identifiant que vous pouvez utiliser pour vérifier que le fichier téléchargé correspond au fichier que vous vouliez.

   ```
   {
   "checksum": "984979b9928ae8d8622286c4a9cd8e99d964a22d59ed0f5722e1733eb280e635"
   }
   ```

4. Poursuivez le téléchargement des parties de votre fichier, si nécessaire. Pour vérifier que vos ensembles de lecture ont été téléchargés, utilisez l'opération d'API list-read-set-upload-parts, comme indiqué ci-dessous. Dans l'exemple suivant, remplacez *sequence store ID upload ID*, et *part source* par votre propre entrée.

   ```
   aws omics list-read-set-upload-parts \
    --sequence-store-id sequence store ID \
    --upload-id upload ID \
    --part-source SOURCE1
   ```

   La réponse renvoie le nombre d'ensembles de lecture, la taille et l'horodatage de la dernière mise à jour.

   ```
   {
   "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. Pour afficher tous les téléchargements de sets de lecture partitionnés actifs, utilisez list-multipart-read-set-uploads, comme indiqué ci-dessous. *sequence store ID*Remplacez-le par l'ID de votre propre magasin de séquences.

```
aws omics list-multipart-read-set-uploads --sequence-store-id
            sequence store ID
```

Cette API renvoie uniquement les téléchargements de jeux de lecture partitionnés en cours. Une fois les ensembles de lecture ingérésACTIVE, ou si le téléchargement a échoué, le téléchargement ne sera pas renvoyé dans la réponse à l'API list-multipart-read-set-uploads. Pour afficher les ensembles de lecture actifs, utilisez l'list-read-setsAPI. Un exemple de réponse pour list-multipart-read-set-uploads est présenté ci-dessous.

```
{
```

```
"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. Après avoir chargé toutes les parties de votre fichier, utilisez complete-multipart-read-set-upload pour terminer le processus de téléchargement, comme indiqué dans l'exemple suivant.

Remplacez *sequence store IDupload ID*, et le paramètre des pièces par vos propres valeurs.

```
aws omics complete-multipart-read-set-upload \
--sequence-store-id sequence store ID \
--upload-id upload ID \
--parts '[{"checksum":"gaCBQMe+rpCFZxLpoP6gydBoXaKKDA/
Vobh5zBDb4W4=","partNumber":1,"partSource":"SOURCE1"}]'
```

La réponse pour complete-multipart-read-set-upload est le jeu de lecture IDs pour vos ensembles de lecture importés.

```
{
"readSetId": "0000000001"
}
```

7. Pour arrêter le téléchargement, utilisez abort-multipart-read-set-upload avec l'ID de téléchargement pour terminer le processus de téléchargement. Remplacez *sequence store ID* et *upload ID* par vos propres valeurs de paramètres.

```
aws omics abort-multipart-read-set-upload \
--sequence-store-id sequence store ID \
--upload-id upload ID
```

8. Une fois le téléchargement terminé, récupérez vos données depuis le set de lecture en utilisant get-read-set, comme indiqué ci-dessous. Si le téléchargement est toujours en cours de traitement, get-read-setrenvoie des métadonnées limitées et les fichiers d'index générés ne sont pas disponibles. Remplacez *sequence store ID* les autres paramètres par vos propres données.

```
aws omics get-read-set
 --sequence-store-id sequence store ID \
 --id read set ID \
 --file SOURCE1 \
 --part-number 1 myfile.fastq.gz
```

9. Pour vérifier les métadonnées, y compris le statut de votre téléchargement, utilisez l'opération get-read-set-metadataAPI.

```
aws omics get-read-set-metadata --sequence-store-id sequence store ID --id read set
 ID
```

La réponse inclut des détails de métadonnées tels que le type de fichier, l'ARN de référence, le nombre de fichiers et la longueur des séquences. Il inclut également le statut. Les statuts possibles sont PROCESSING_UPLOADACTIVE, et. 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"
}
```

## Configuration d'un emplacement de secours

Lorsque vous créez ou mettez à jour un magasin de séquences, vous pouvez configurer un compartiment Amazon S3 comme emplacement de secours pour les fichiers qui ne parviennent pas

à être chargés. Les parties du fichier correspondant à ces ensembles de lecture sont transférées vers l'emplacement de secours. Des emplacements de secours sont disponibles pour les magasins de séquences créés après le 15 mai 2023.

Créez une politique de compartiment Amazon S3 pour accorder un accès en HealthOmics écriture à l'emplacement de secours d'Amazon S3, comme illustré dans l'exemple suivant :

```
{
    "Effect": "Allow",
    "Principal": {
        "Service": "omics.amazonaws.com"
    },
    "Action": "s3:PutObject",
    "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
}
```

Si le compartiment Amazon S3 pour les journaux de secours ou d'accès utilise une clé gérée par le client, ajoutez les autorisations suivantes à la politique en matière de clés :

```
 {
    "Sid": "Allow use of key",
    "Effect": "Allow",
    "Principal": {
        "Service": "omics.amazonaws.com"
    },
    "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey*"
    ],
    "Resource": "*"
}
```

# Exporter HealthOmics des ensembles de lectures vers un compartiment Amazon S3

Vous pouvez exporter des ensembles de lectures sous forme de tâche d'exportation par lots vers un compartiment Amazon S3. Pour ce faire, créez d'abord une stratégie IAM dotée d'un accès en écriture au bucket, comme dans l'exemple de stratégie IAM suivant.

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"
  }
]
}
```

Une fois la politique IAM en place, commencez votre travail d'exportation du set de lecture. L'exemple suivant montre comment procéder à l'aide de l'opération start-read-set-exportd'API

-job. Dans l'exemple suivant, remplacez tous les paramètres, tels que *sequence store IDdestination*,*role ARN*, et*sources*, par votre entrée.

```
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
```

Vous recevez la réponse suivante contenant des informations sur le magasin de séquences d'origine et le compartiment Amazon S3 de destination.

```
{
"id": <job-id>,
"sequenceStoreId": <sequence-store-id>,
"destination": <destination-s3-uri>,
"status": "SUBMITTED",
"creationTime": "2022-10-22T01:33:38.079000+00:00"
}
```

Une fois la tâche démarrée, vous pouvez déterminer son statut à l'aide de l'opération get-read-set-exportd'API -job, comme indiqué ci-dessous. Remplacez le *sequence store ID* et *job ID* par votre identifiant de magasin de séquences et votre identifiant de tâche, respectivement.

```
aws omics get-read-set-export-job --id job-id --sequence-store-id sequence store ID
```

Vous pouvez afficher toutes les tâches d'exportation initialisées pour un magasin de séquences à l'aide de l'opération  list-read-set-exportd'API -jobs, comme illustré ci-dessous. Remplacez le *sequence store ID* par votre identifiant de magasin de séquences.

```
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"
  }
]
}
```

En plus d'exporter vos ensembles de lecture, vous pouvez également les partager en utilisant l'accès Amazon S3 URIs. Pour en savoir plus, consultez Accès aux ensembles de HealthOmics lecture avec Amazon S3 URIs.

# Accès aux ensembles de HealthOmics lecture avec Amazon S3 URIs

Vous pouvez utiliser les chemins d'URI Amazon S3 pour accéder aux ensembles de lecture de votre magasin de séquences actif.

Avec le chemin d'URI Amazon S3, vous pouvez utiliser les opérations Amazon S3 pour répertorier, partager et télécharger vos ensembles de lectures. L'accès via le S3 APIs accélère la collaboration et l'intégration des outils étant donné que de nombreux outils du secteur sont déjà conçus pour lire depuis S3. En outre, vous pouvez partager l'accès au S3 APIs avec d'autres comptes et fournir un accès en lecture aux données entre les régions.

HealthOmics ne prend pas en charge l'accès par URI Amazon S3 aux ensembles de lecture archivés. Lorsque vous activez un ensemble de lecture, il est restauré dans le même chemin d'URI à chaque fois.

Lorsque les données sont chargées dans HealthOmics les magasins, étant donné que l'URI Amazon S3 est basée sur les points d'accès Amazon S3, vous pouvez l'intégrer directement aux outils standard du secteur qui lisent Amazon S3 URIs, tels que les suivants :

- Applications d'analyse visuelle telles que Integrative Genomics Viewer (IGV) ou UCSC Genome Browser.
- Workflows courants avec les extensions Amazon S3 telles que CWL, WDL et Nextflow.
- Tout outil capable d'authentifier et de lire depuis le point d'accès Amazon S3 URIs ou de lire un Amazon S3 présigné. URIs
- Utilitaires Amazon S3 tels que Mountpoint ou. CloudFront

Amazon S3 Mountpoint vous permet d'utiliser un compartiment Amazon S3 comme système de fichiers local. Pour en savoir plus sur Mountpoint et pour l'installer en vue de son utilisation, consultez Mountpoint pour Amazon S3.

Amazon CloudFront est un service de réseau de diffusion de contenu (CDN) conçu pour optimiser les performances, la sécurité et le confort des développeurs. Pour en savoir plus sur l'utilisation d'Amazon CloudFront, consultez la CloudFront documentation Amazon. Pour configurer CloudFront un magasin de séquences, contactez l' AWS HealthOmics équipe.

Le compte root du propriétaire des données est activé pour les actions S3 :GetObject, S3 : GetObjectTagging et S3:List Bucket sur le préfixe du magasin de séquences. Pour qu'un utilisateur du compte puisse accéder aux données, vous devez créer une politique IAM et l'associer à l'utilisateur ou au rôle. Pour un exemple de politique, consultez Autorisations d'accès aux données à l'aide d'Amazon S3 URIs.

Vous pouvez utiliser les opérations d'API Amazon S3 suivantes sur les ensembles de lecture actifs pour répertorier et récupérer vos données. Vous pouvez accéder aux ensembles de lecture archivés via Amazon S3 une URIs fois qu'ils ont été activés.

- GetObject— Récupère un objet depuis Amazon S3.
- HeadObject— L'opération HEAD récupère les métadonnées d'un objet sans renvoyer l'objet lui-même. Cette opération est utile si vous souhaitez uniquement les métadonnées d'un objet.
- ListObjects et ListObject v2 — Renvoie une partie ou la totalité (jusqu'à 1 000) des objets d'un compartiment.
- CopyObject— Crée une copie d'un objet déjà stocké dans Amazon S3. HealthOmicsprend en charge la copie vers un point d'accès Amazon S3, mais pas l'écriture sur un point d'accès.

HealthOmics les magasins de séquences conservent l'identité sémantique des fichiers via ETags. Tout au long du cycle de vie d'un fichier, l'Amazon S3 ETag, qui est basé sur l'identité bit à bit, peut changer, mais cela HealthOmics ETag reste le même. Pour en savoir plus, veuillez consulter la section HealthOmics ETags et provenance des données.

Rubriques

- Structure d'URI Amazon S3 dans le HealthOmics stockage
- Utilisation d'un IGV hébergé ou local pour accéder aux ensembles de lecture
- À l'aide de Samtools ou HTSlib dans HealthOmics

- Utilisation de Mountpoint HealthOmics
- Utilisation CloudFront avec HealthOmics

## Structure d'URI Amazon S3 dans le HealthOmics stockage

Tous les fichiers associés à Amazon S3 URIs possèdent `omics:subjectId` des balises de `omics:sampleId` ressource. Vous pouvez utiliser ces balises pour partager l'accès en utilisant des politiques IAM via un modèle tel que`"s3:ExistingObjectTag/omics:subjectId": "pattern desired"`.

La structure du fichier est la suivante :

`.../`*`account_id`*`/sequenceStore/`*`seq_store_id`*`/readSet/`*`read_set_id`*`/`*`files`*`.`

Pour les fichiers importés dans des magasins de séquences depuis Amazon S3, le magasin de séquences tente de conserver le nom de source d'origine. En cas de conflit entre les noms, le système ajoute des informations relatives aux ensembles de lecture pour garantir que les noms de fichiers sont uniques. Par exemple, pour les ensembles de lecture fastq, si les deux noms de fichiers sont identiques, afin de rendre les noms uniques, ils sont insérés avant `sourceX` .fastq.gz ou .fq.gz. Pour un téléchargement direct, les noms de fichiers suivent les modèles suivants :

- Pour FASTQ— *`read_set_name`* _ *`sourcex`* .fastq.gz
- Pour uBAM/BAM/CRAM —*`read_set_name`*. *`file extension`*avec des extensions de `.bam` ou`.cram`. Par exemple : `NA193948.bam`.

Pour les ensembles de lecture BAM ou CRAM, les fichiers d'index sont automatiquement générés pendant le processus d'ingestion. Pour les fichiers d'index générés, l'extension d'index appropriée à la fin du nom de fichier est appliquée. Il a le modèle *`<name of the Source the index is on>.<file index extension>.`* Les extensions d'index sont `.bai` ou`.crai`.

## Utilisation d'un IGV hébergé ou local pour accéder aux ensembles de lecture

IGV est un navigateur génomique utilisé pour analyser les fichiers BAM et CRAM. Il nécessite à la fois le fichier et l'index car il n'affiche qu'une partie du génome à la fois. L'IGV peut être téléchargé et utilisé localement, et il existe des guides pour créer un IGV hébergé par AWS. La version Web publique n'est pas prise en charge car elle nécessite CORS.

L'IGV local s'appuie sur la AWS configuration locale pour accéder aux fichiers. Assurez-vous que le rôle utilisé dans cette configuration est associé à une politique qui active les GetObject autorisations kms: Decrypt et s3 : sur l'URI s3 des ensembles de lecture auxquels vous accédez. Ensuite, dans IGV, vous pouvez utiliser « Fichier > charger à partir de l'URL » et coller l'URI pour la source et l'index. Sinon, le pré-signé URLs peut être généré et utilisé de la même manière, ce qui permettra de contourner la configuration AWS. Notez que CORS n'est pas pris en charge avec l'accès aux URI Amazon S3. Les demandes basées sur CORS ne sont donc pas prises en charge.

L'exemple AWS Hosted IGV s'appuie sur AWS Cognito pour créer les configurations et les autorisations appropriées au sein de l'environnement. Assurez-vous de créer une politique qui active les autorisations KMS:Decrypt et s3 : sur GetObject l'URI Amazon S3 des ensembles de lecture auxquels vous accédez, et ajoutez cette politique au rôle attribué au groupe d'utilisateurs Cognito. Ensuite, dans IGV, vous pouvez utiliser « Fichier > charger à partir de l'URL » et saisir l'URI de la source et de l'index. Sinon, le pré-signé URLs peut être généré et utilisé de la même manière, sans passer par la configuration AWS.

Notez que le magasin de séquences n'apparaîtra pas sous l'onglet « Amazon » car il affiche uniquement les buckets dont vous êtes propriétaire dans la région dans laquelle le AWS profil est configuré.

## À l'aide de Samtools ou HTSlib dans HealthOmics

HTSlib est la bibliothèque principale partagée par plusieurs outils tels que Samtools, RSAMTools et autres PySam. Utilisez HTSlib la version 1.20 ou ultérieure pour bénéficier d'une prise en charge fluide des points d'accès Amazon S3. Pour les anciennes versions de la HTSlib bibliothèque, vous pouvez utiliser les solutions suivantes :

- Définissez la variable d'environnement pour l'hôte HTS Amazon S3 avec :`export HTS_S3_HOST="s3.region.amazonaws.com"`.
- Générez une URL présignée pour les fichiers que vous souhaitez utiliser. Si un BAM ou un CRAM est utilisé, assurez-vous qu'une URL présignée est générée à la fois pour le fichier et pour l'index. Ensuite, les deux fichiers peuvent être utilisés avec les bibliothèques.
- Utilisez Mountpoint pour monter le magasin de séquences ou le préfixe du set de lecture dans le même environnement que celui dans lequel vous utilisez des bibliothèques. HTSlib À partir de là, les fichiers sont accessibles en utilisant les chemins de fichiers locaux.

## Utilisation de Mountpoint HealthOmics

Mountpoint pour Amazon S3 est un client de fichiers simple à haut débit permettant de monter un compartiment Amazon S3 en tant que système de fichiers local. Avec Mountpoint pour Amazon S3, vos applications peuvent accéder aux objets stockés dans Amazon S3 par le biais d'opérations de fichier telles que l'ouverture et la lecture. Mountpoint for Amazon S3 traduit automatiquement ces opérations en appels d'API d'objets Amazon S3, permettant ainsi à vos applications d'accéder au stockage et au débit élastiques d'Amazon S3 via une interface de fichier.

Mountpoint peut être installé à l'aide des instructions d'installation de Mountpoint. Mountpoint utilise le profil AWS local à l'installation et fonctionne au niveau du préfixe Amazon S3. Assurez-vous que le profil utilisé dispose d'une politique qui active les autorisations s3 :GetObject, s3 : ListBucket et kms: Decrypt sur le préfixe d'URI Amazon S3 du ou des ensembles de lecture ou du magasin de séquences auxquels vous accédez. Ensuite, le godet peut être monté en utilisant le chemin suivant :

```
mount-s3 access point arn local path to mount --prefix prefix to sequence store or read
 set --region region
```

## Utilisation CloudFront avec HealthOmics

Amazon CloudFront est un service de réseau de diffusion de contenu (CDN) conçu pour optimiser les performances, la sécurité et le confort des développeurs. Les clients qui souhaitent l'utiliser CloudFront doivent contacter l'équipe du service pour activer la CloudFront distribution. Collaborez avec l'équipe chargée de votre compte pour impliquer l'équipe HealthOmics de service.

# L'activation de la lecture s'installe dans HealthOmics

Vous pouvez activer les ensembles de lecture archivés à l'aide de l'opération start-read-set-activationd'API -job ou via le AWS CLI, comme indiqué dans l'exemple suivant. Remplacez le *sequence store ID* et *read set id* par votre identifiant de magasin de séquences et votre ensemble de lecture 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
```

Vous recevez une réponse contenant les informations relatives à la tâche d'activation, comme indiqué ci-dessous.

```
{
    "id": "12345678",
    "sequenceStoreId": "1234567890",
    "status": "SUBMITTED",
    "creationTime": "2022-10-22T00:50:54.670000+00:00"
}
```

Une fois la tâche d'activation lancée, vous pouvez suivre sa progression à l'aide de l'opération get-read-set-activationd'API -job. Voici un exemple d'utilisation du pour vérifier le statut AWS CLI de votre tâche d'activation. Remplacez *job ID* et *sequence store ID* par votre identifiant de magasin de séquences et votre tâche IDs, respectivement.

```
aws omics get-read-set-activation-job --id job ID --sequence-store-id sequence store ID
```

La réponse résume la tâche d'activation, comme indiqué ci-dessous.

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

Vous pouvez vérifier l'état d'une tâche d'activation à l'aide de l'opération get-read-set-metadataAPI. Les statuts possibles sont `ACTIVEACTIVATING`, et. `ARCHIVED` Dans l'exemple suivant, remplacez *sequence store ID* par votre identifiant de magasin de séquences et remplacez *read set ID* par votre identifiant de jeu de lecture.

```
aws omics get-read-set-metadata --sequence-store-id sequence store ID --id read set ID
```

La réponse suivante indique que le set de lecture est actif.


```
{
    "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"
    }
}
```

Vous pouvez afficher toutes les tâches d'activation des ensembles de lecture à l'aide de list-read-set-activation-jobs, comme illustré dans l'exemple suivant. Dans l'exemple suivant, remplacez-le *sequence store ID* par votre identifiant de magasin de séquences.

```
aws omics list-read-set-activation-jobs --sequence-store-id sequence store ID
```

Vous recevez la réponse suivante.

```
{
    "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 analyses

> **Important**
>
> AWS HealthOmics les magasins de variantes et les magasins d'annotations ne sont plus ouverts aux nouveaux clients. Les clients existants peuvent continuer à utiliser le service normalement. Pour de plus amples informations, veuillez consulter AWS HealthOmics modification de la disponibilité du magasin de variantes et du magasin d'annotations.

HealthOmics analytics prend en charge le stockage et l'analyse des variantes génomiques et des annotations. Analytics fournit deux types de ressources de stockage : les magasins de variantes et les magasins d'annotations. Vous utilisez ces ressources pour stocker, transformer et interroger des données de variants génomiques et des données d'annotation. Après avoir importé des données dans une banque de données, vous pouvez utiliser Athena pour effectuer des analyses avancées sur les données.

Vous pouvez utiliser la HealthOmics console ou l'API pour créer et gérer des boutiques, importer des données et partager des données analytiques de boutique avec des collaborateurs.

Variant stocke les données de support au format VCF, tandis que les annotations stockent le support TSV/CSV et les GFF3 formats. Les coordonnées génomiques sont représentées sous forme d'intervalles de base zéro, semi-fermés, semi-ouverts. Lorsque vos données se trouvent dans le magasin de données HealthOmics d'analyse, l'accès aux fichiers VCF est géré via AWS Lake Formation. Vous pouvez ensuite interroger les fichiers VCF à l'aide d'Amazon Athena. Les requêtes doivent utiliser le moteur de requêtes Athena version 3. Pour en savoir plus sur les versions du moteur de requête Athena, consultez la documentation Amazon Athena.

Rubriques

- Interrogation de HealthOmics données analytiques
- Partage de magasins HealthOmics d'analyse

# Création de boutiques de HealthOmics variantes

> **Important**
>
> AWS HealthOmics les magasins de variantes et les magasins d'annotations ne sont plus ouverts aux nouveaux clients. Les clients existants peuvent continuer à utiliser le service normalement. Pour de plus amples informations, veuillez consulter AWS HealthOmics modification de la disponibilité du magasin de variantes et du magasin d'annotations.

Les rubriques suivantes décrivent comment créer des magasins de HealthOmics variantes à l'aide de la console et de l'API.

Rubriques

- Création d'un magasin de variantes à l'aide de la console
- Création d'un magasin de variantes à l'aide de l'API

## Création d'un magasin de variantes à l'aide de la console

Vous pouvez créer un magasin de variantes à l'aide de la HealthOmics console.

1. Ouvrez la HealthOmics console.
2. Si nécessaire, ouvrez le volet de navigation de gauche (≡). Choisissez Variant stores.
3. Sur la page Créer un magasin de variantes, fournissez les informations suivantes
   - Variante du nom de la boutique : nom unique pour cette boutique.
   - Description (facultatif) : description de ce magasin de variantes.
   - Génome de référence - Le génome de référence pour cette banque de variants.
   - Chiffrement des données - Choisissez si vous souhaitez que le chiffrement des données soit détenu et géré par vous-même AWS ou par vous-même.
   - Balises (facultatif) - Fournissez jusqu'à 50 balises pour cette variante de boutique.
4. Choisissez Créer un magasin de variantes.

# Création d'un magasin de variantes à l'aide de l'API

Utilisez le fonctionnement de l' HealthOmics `CreateVariantStore`API pour créer des magasins de variantes. Vous pouvez également effectuer cette opération avec le AWS CLI.

Pour créer un magasin de variantes, vous devez fournir un nom au magasin et l'ARN d'un magasin de référence. Le magasin de variantes est prêt à ingérer des données lorsque son statut passe à READY.

L'exemple suivant utilise le AWS CLI pour créer un magasin de variantes.

```
aws omics create-variant-store --name myvariantstore \
    --reference referenceArn="arn:aws:omics:us-
west-2:555555555555:referenceStore/123456789/reference/5987565360"
```

Pour confirmer la création de votre boutique de variantes, vous recevez la réponse suivante.

```
{
    "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"
}
```

Pour en savoir plus sur un magasin de variantes, utilisez l'get-variant-storeAPI.

```
aws omics get-variant-store --name myvariantstore
```

Vous recevez la réponse suivante.

```
{
    "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
}
```

Pour afficher toutes les variantes de magasins associées à un compte, utilisez l'list-variant-storesAPI.

```
aws omics list-variant-stores
```

Vous recevez une réponse répertoriant toutes les variantes de magasins, ainsi que leurs IDs statuts et autres informations, comme indiqué dans l'exemple de réponse suivant.

```
{
    "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
        }
    ]
}
```

Vous pouvez également filtrer les réponses pour l'list-variant-storesAPI en fonction des statuts ou d'autres critères.

Les fichiers VCF importés dans des magasins d'analyse créés le 15 mai 2023 ou après cette date ont défini des schémas pour les annotations VEP (Variant Effect Predictor). Cela facilite l'interrogation

et l'analyse des données VCF importées. La modification n'a aucun impact sur les magasins créés avant le 15 mai 2023, sauf si le `annotation fields` paramètre est inclus dans l'appel d'API ou de CLI. Pour ces magasins, l'utilisation du `annotation fields` paramètre entraînera l'échec de la demande.

# Création de tâches d'importation de magasins de HealthOmics variantes

> **Important**
>
> AWS HealthOmics les magasins de variantes et les magasins d'annotations ne sont plus ouverts aux nouveaux clients. Les clients existants peuvent continuer à utiliser le service normalement. Pour de plus amples informations, veuillez consulter AWS HealthOmics modification de la disponibilité du magasin de variantes et du magasin d'annotations.

L'exemple suivant montre comment utiliser le AWS CLI pour créer une tâche d'importation pour un magasin de variantes.

```
aws omics start-variant-import-job \
       --destination-name myvariantstore \
       --runLeftNormalization false \
       --role-arn  arn:aws:iam::55555555555:role/roleName \
       --items source=s3://my-omics-bucket/sample.vcf.gz source=s3://my-omics-bucket/
sample2.vcf.gz
```

```
{
    "destinationName": "store_a",
    "roleArn": "....",
    "runLeftNormalization": false,
    "items": [
        {"source": "s3://my-omics-bucket/sample.vcf.gz"},
        {"source": "s3://my-omics-bucket/sample2.vcf.gz"}
    ]
}
```

Pour les boutiques créées après le 15 mai 2023, l'exemple suivant montre comment ajouter le `--annotation-fields` paramètre. Les champs d'annotation sont définis lors de l'importation.

```
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"
}
```

get-variant-import-jobÀ utiliser pour vérifier le statut.

```
aws omics get-variant-import-job --job-id 08279950-a9e3-4cc3-9a3c-a574f9c9e229
```

Vous recevrez une réponse JSON indiquant le statut de votre tâche d'importation. Les annotations VEP dans le VCF sont analysées pour les informations stockées dans la colonne INFO sous forme de paire. ID/Value L'ID par défaut pour la colonne INFO des annotations d'Ensembl Variant Effect Predictor est CSQ, mais vous pouvez utiliser le `--annotation-fields` paramètre pour indiquer une valeur personnalisée utilisée dans la colonne INFO. L'analyse syntaxique est actuellement prise en charge pour les annotations VEP.

Pour une boutique créée avant le 15 mai 2023 ou pour les fichiers VCF qui n'incluent pas d'annotation VEP, la réponse n'inclut aucun champ d'annotation.

```
{
    "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",
```

```
}
```

Les annotations VEP qui font partie des fichiers VCF sont stockées sous forme de schéma prédéfini avec la structure suivante. Le champ extras peut être utilisé pour stocker tous les champs VEP supplémentaires qui ne sont pas inclus dans le schéma par défaut.

```
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>
    >>
>
```

L'analyse est effectuée selon une approche basée sur le meilleur effort. Si l'entrée VEP ne respecte pas les spécifications standard VEP, elle ne sera pas analysée et la ligne du tableau sera vide.

Pour un nouveau magasin de variantes, la réponse pour get-variant-import-jobinclurait les champs d'annotation, comme indiqué.

```
aws omics get-variant-import-job --job-id 08279950-a9e3-4cc3-9a3c-a574f9c9e229
```

Vous recevez une réponse JSON qui indique le statut de votre tâche d'importation.

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

Vous pouvez l'utiliser list-variant-import-jobspour voir toutes les tâches d'importation et leur statut.

```
aws omics list-variant-import-jobs --ids 7a1c67e3-b7f9-434d-817b-9c571fd63bea
```

La réponse contient les informations suivantes.

```
{
    "variantImportJobs": [
    {
        "creationTime": "2023-04-11T17:52:37.241958+00:00",
        "destinationName": "annotationparsingvariantstore",
        "id": "7a1c67e3-b7f9-434d-817b-9c571fd63bea",
        "roleArn": "arn:aws:iam::55555555555:role/roleName",
        "runLeftNormalization": false,
        "status": "COMPLETED",
        "updateTime": "2023-04-11T17:58:22.676043+00:00",
        "annotationFields" : {"VEP": "CSQ"}
        }
    ]
  }
```

```
}
```

Si nécessaire, vous pouvez annuler une tâche d'importation à l'aide de la commande suivante.

```
aws omics cancel-variant-import-job
     --job-id edd7b8ce-xmpl-47e2-bc99-258cac95a508
```

# Création de magasins HealthOmics d'annotations

> **Important**
>
> AWS HealthOmics les magasins de variantes et les magasins d'annotations ne sont plus ouverts aux nouveaux clients. Les clients existants peuvent continuer à utiliser le service normalement. Pour de plus amples informations, veuillez consulter AWS HealthOmics modification de la disponibilité du magasin de variantes et du magasin d'annotations.

Un magasin d'annotations est un magasin de données représentant une base de données d'annotations, telle qu'une base de données provenant d'un fichier TSV, VCF ou GFF. Si le même génome de référence est spécifié, les magasins d'annotations sont mappés sur le même système de coordonnées que les magasins de variantes lors d'une importation. Les rubriques suivantes expliquent comment utiliser la HealthOmics console et comment créer et AWS CLI gérer des magasins d'annotations.

Rubriques



## Création d'un magasin d'annotations à l'aide de la console

Pour créer des banques d'annotations avec la HealthOmics console, procédez comme suit.

Pour créer un magasin d'annotations

1. Ouvrez la HealthOmics console.
2. Si nécessaire, ouvrez le volet de navigation de gauche (≡). Choisissez Annotation Stores.

3. Sur la page Stockages d'annotations, choisissez Créer un magasin d'annotations.
4. Sur la page Créer un magasin d'annotations, fournissez les informations suivantes

   - Nom du magasin d'annotations : nom unique pour ce magasin.
   - Description (facultatif) - Description de ce génome de référence.
   - Format des données et détails du schéma : sélectionnez le format du fichier de données et téléchargez la définition du schéma pour ce magasin.
   - Génome de référence - Le génome de référence pour cette annotation.
   - Chiffrement des données : choisissez si vous souhaitez que le chiffrement des données soit détenu et géré par vous-même AWS ou par vous-même.
   - Balises (facultatif) : fournissez jusqu'à 50 balises pour ce magasin d'annotations.
5. Choisissez Créer un magasin d'annotations.

## Création d'un magasin d'annotations à l'aide de l'API

L'exemple suivant montre comment créer un magasin d'annotations à l'aide du AWS CLI. Pour toutes les opérations AWS CLI et les opérations d'API, vous devez spécifier le format de vos données.

```
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
```

Vous recevez la réponse suivante pour confirmer la création de votre magasin d'annotations.

```
{
          "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"
      }
```

Pour en savoir plus sur un magasin d'annotations, utilisez l'get-annotation-storeAPI.

```
aws omics get-annotation-store --name my_annotation_store
```

Vous recevez la réponse suivante.

```
{
          "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
      }
```

Pour afficher tous les magasins d'annotations associés à un compte, utilisez l'opération list-annotation-storesAPI.

```
aws omics list-annotation-stores
```

Vous recevez une réponse répertoriant tous les magasins d'annotations IDs, ainsi que leurs statuts et autres informations, comme indiqué dans l'exemple de réponse suivant.

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

Vous pouvez également filtrer les réponses en fonction du statut ou d'autres critères.

# Création de tâches d'importation pour les magasins HealthOmics d'annotations

> **Important**
>
> AWS HealthOmics les magasins de variantes et les magasins d'annotations ne sont plus ouverts aux nouveaux clients. Les clients existants peuvent continuer à utiliser le service normalement. Pour de plus amples informations, veuillez consulter AWS HealthOmics modification de la disponibilité du magasin de variantes et du magasin d'annotations.

Rubriques



## Création d'une tâche d'importation d'annotations à l'aide de l'API

L'exemple suivant montre comment utiliser le AWS CLI pour démarrer une tâche d'importation d'annotations.

```
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"}'
```

Les magasins d'annotations créés avant le 15 mai 2023 renvoient un message d'erreur si les champs d'annotation sont inclus. Ils ne renvoient aucun résultat pour les opérations d'API impliquées dans les tâches d'importation de magasins d'annotations.

Vous pouvez ensuite utiliser l'opération get-annotation-import-jobd'API et le `job ID` paramètre pour en savoir plus sur la tâche d'importation d'annotations.

```
aws omics get-annotation-import-job --job-id 9e4198fb-fa85-446c-9301-9b823a1a8ba8
```

Vous recevez la réponse suivante, y compris les champs d'annotation.

```
{
          "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"}
       }
```

Pour afficher toutes les tâches d'importation du magasin d'annotations, utilisez list-annotation-import-jobs.

```
aws omics list-annotation-import-jobs --ids 9e4198fb-fa85-446c-9301-9b823a1a8ba8
```

La réponse inclut les détails et les statuts des tâches d'importation de votre magasin d'annotations.

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

## Paramètres supplémentaires pour les formats TSV et VCF

Pour les formats TSV et VCF, des paramètres supplémentaires indiquent à l'API comment analyser votre entrée.

> **Important**
>
> Les données d'annotation CSV exportées à l'aide de moteurs de requête renvoient directement les informations issues de l'importation du jeu de données. Si les données importées contiennent des formules ou des commandes, le fichier peut être soumis à une injection CSV. Par conséquent, les fichiers exportés à l'aide de moteurs de requête peuvent provoquer des avertissements de sécurité. Pour éviter toute activité malveillante, désactivez les liens et les macros lors de la lecture des fichiers d'exportation.

L'analyseur TSV effectue également des opérations bioinformatiques de base, telles que la normalisation à gauche et la standardisation des coordonnées génomiques, répertoriées dans le tableau suivant.

| Type de format | Description |
| --- | --- |
| Générique | Fichier texte générique. Aucune information génomique. |
| `CHR_POS` | Position de départ - 1, ajouter une position de fin, qui est identique à`POS`. |

| Type de format | Description |
| --- | --- |
| CHR_POS_REF_ALT | Contient des informations sur les allèles contig, 1 base, ref et alt. |
| CHR_START_END_REF_ALT_ONE_BASE | Contient des informations sur les allèles contig, start, end, ref et alt. Les coordonnées sont basées sur 1. |
| CHR_START_END_ZERO_BASE | Contient les positions contig, de début et de fin. Les coordonnées sont basées sur 0. |
| CHR_START_END_ONE_BASE | Contient les positions contig, de début et de fin. Les coordonnées sont basées sur 1. |
| CHR_START_END_REF_ALT_ZERO_BASE | Contient des informations sur les allèles contig, start, end, ref et alt. Les coordonnées sont basées sur 0. |

Une demande de magasin d'annotations d'importation TSV ressemble à l'exemple suivant.

```
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"
        }
    }
}'
```

## Création de magasins d'annotations au format TSV

L'exemple suivant crée un magasin d'annotations à l'aide d'un fichier limité à onglets contenant un en-tête, des lignes et des commentaires. Les coordonnées sontCHR_START_END_ONE_BASED, et elle contient la carte HG19 génétique tirée du synopsis de la carte des gènes humains de l'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}]}'
```

Vous pouvez importer des fichiers avec ou sans en-tête. Pour l'indiquer dans une demande CLI, utilisez`header=false`, comme indiqué dans l'exemple de tâche d'importation suivant.

```
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="#"}}'
```

L'exemple suivant crée un magasin d'annotations pour un fichier bed. Un fichier bed est un simple fichier délimité par des tabulations. Dans cet exemple, les colonnes sont le chromosome, le début, la fin et le nom de la région. Les coordonnées sont basées sur zéro et les données n'ont pas d'en-tête.

```
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}]}'
```

Vous pouvez ensuite importer le fichier bed dans le magasin d'annotations à l'aide de la commande CLI suivante.

```
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="#"}}'
```

L'exemple suivant crée un magasin d'annotations pour un fichier délimité par des tabulations qui contient les premières colonnes d'un fichier VCF, suivies de colonnes contenant des informations d'annotation. Il contient les positions du génome ainsi que des informations sur le chromosome, les allèles de départ, de référence et alternatifs, et il contient un en-tête.

```
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}]}'
```

Vous devez ensuite importer le fichier dans le magasin d'annotations à l'aide de la commande CLI suivante.

```
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="#"}}'
```

L'exemple suivant montre comment un client peut créer un magasin d'annotations pour un fichier mim2gene. Un fichier mim2gene fournit les liens entre les gènes d'OMIM et un autre identifiant de gène. Il est délimité par des tabulations et contient des commentaires.

```
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}]}'
```

Vous pouvez ensuite importer des données dans votre boutique comme suit.

```
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="#"}}'
```

## Démarrage de tâches d'importation au format VCF

Pour les fichiers VCF, il existe deux entrées supplémentaires qui ignorent ou incluent ces paramètres`ignoreFilterField`, comme indiqué. `ignoreQualField`

```
aws omics start-annotation-import-job --destination-name annotation_example\
  --role-arn arn:aws:iam::555555555555:role/demoRole \
  --items source=s3://demodata/example.garvan.vcf \
  --format-options '{ "vcfOptions": {
    "ignoreQualField": false,
    "ignoreFilterField": false
    }
   }'
```

Vous pouvez également annuler l'importation d'un magasin d'annotations, comme indiqué. Si l'annulation aboutit, vous ne recevrez pas de réponse à cet AWS CLI appel. Toutefois, si l'ID de la tâche d'importation est introuvable ou si la tâche d'importation est terminée, vous recevez un message d'erreur.

```
aws omics cancel-annotation-import-job --job-id edd7b8ce-xmpl-47e2-bc99-258cac95a508
```

> **Note**
>
> L'historique de vos tâches d'importation de métadonnées pour get-annotation-import-jobget-variant-import-job, list-annotation-import-jobs, et list-variant-import-jobsest automatiquement supprimé au bout de deux ans. Les données de variante et d'annotation importées ne sont pas supprimées automatiquement et restent dans vos magasins de données.

# Création de versions de magasin HealthOmics d'annotations

> **Important**
>
> AWS HealthOmics les magasins de variantes et les magasins d'annotations ne sont plus ouverts aux nouveaux clients. Les clients existants peuvent continuer à utiliser le service normalement. Pour de plus amples informations, veuillez consulter AWS HealthOmics modification de la disponibilité du magasin de variantes et du magasin d'annotations.

Vous pouvez créer de nouvelles versions de magasins d'annotations pour collecter différentes versions de vos bases de données d'annotations. Cela vous permet d'organiser vos données d'annotation, qui sont mises à jour régulièrement.

Pour créer une nouvelle version d'un magasin d'annotations existant, utilisez l'create-annotation-store-versionAPI comme indiqué dans l'exemple suivant.

```
aws omics create-annotation-store-version \
     --name my_annotation_store \
     --version-name my_version
```

Vous recevrez la réponse suivante avec l'ID de version du magasin d'annotations, confirmant qu'une nouvelle version de votre annotation a été créée.

```
{
     "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"
}
```

Pour mettre à jour la description d'une version du magasin d'annotations, vous pouvez l'utiliser update-annotation-store-versionpour ajouter des mises à jour à une version du magasin d'annotations.

```
aws omics update-annotation-store-version \
    --name my_annotation_store \
    --version-name my_version \
    --description "New Description"
```

Vous recevrez la réponse suivante, confirmant que la version du magasin d'annotations a été mise à jour.

```
{
     "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"
}
```

Pour afficher les détails d'une version du magasin d'annotations, utilisez get-annotation-store-version.

```
aws omics get-annotation-store-version --name my_annotation_store --version-name
 my_version
```

Vous recevrez une réponse avec le nom de la version, le statut et d'autres détails.

```
{
     "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
    }
```

Pour afficher toutes les versions d'un magasin d'annotations, vous pouvez utiliser list-annotation-store-versions, comme indiqué dans l'exemple suivant.

```
aws omics list-annotation-store-versions --name my_annotation_store
```

Vous recevrez une réponse contenant les informations suivantes

```
{
  "annotationStoreVersions": [
    {
     "storeId": "4934045d1c6d",
     "id": "2a3f4a44aa7b",
     "status": "CREATING",
     "versionArn": "arn:aws:omics:us-west-2:555555555555:annotationStore/
my_annotation_store/version/my_version_2",
```

```
     "name": "my_annotation_store",
     "versionName": "my_version_2",
     "creation Time": "2023-07-21T17:20:59.380043+00:00",
     "versionSizeBytes": 0
    },
    {
     "storeId": "4934045d1c6d",
     "id": "4934045d1c6d",
     "status": "ACTIVE",
     "versionArn": "arn:aws:omics:us-west-2:555555555555:annotationStore/
my_annotation_store/version/my_version_1",
     "name": "my_annotation_store",
     "versionName": "my_version_1",
     "creationTime": "2023-07-21T17:15:49.251040+00:00",
     "updateTime": "2023-07-21T17:15:56.434223+00:00",
     "statusMessage": "",
     "versionSizeBytes": 0
    }
}
```

Si vous n'avez plus besoin d'une version du magasin d'annotations, vous pouvez l'utiliser delete-annotation-store-versionspour supprimer une version du magasin d'annotations, comme indiqué dans l'exemple suivant.

```
aws omics delete-annotation-store-versions --name my_annotation_store --versions
 my_version
```

Si la version du magasin est supprimée sans erreur, vous recevrez la réponse suivante.

```
{
  "errors": []
}
```

S'il y a des erreurs, vous recevrez une réponse avec les détails des erreurs, comme indiqué.

```
{
  "errors": [
    {
      "versionName": "my_version",
      "message": "Version with versionName: my_version was not found."
    }
```

```
  ]
}
```

Si vous essayez de supprimer une version du magasin d'annotations pour laquelle une tâche d'importation est active, vous recevrez une réponse contenant une erreur, comme indiqué.

```
{
  "errors": [
    {
      "versionName": "my_version",
      "message": "version has an inflight import running"
    }
  ]
}
```

Dans ce cas, vous pouvez forcer la suppression de la version du magasin d'annotations, comme illustré dans l'exemple suivant.

```
aws omics delete-annotation-store-versions --name my_annotation_store --versions
 my_version --force
```

# Supprimer des magasins HealthOmics d'analyse

> **Important**
>
> AWS HealthOmics les magasins de variantes et les magasins d'annotations ne sont plus ouverts aux nouveaux clients. Les clients existants peuvent continuer à utiliser le service normalement. Pour de plus amples informations, veuillez consulter AWS HealthOmics modification de la disponibilité du magasin de variantes et du magasin d'annotations.

Lorsque vous supprimez un magasin de variantes ou d'annotations, le système supprime également toutes les données importées dans ce magasin et toutes les balises associées.

L'exemple suivant montre comment supprimer un magasin de variantes à l'aide du AWS CLI. Si l'action est réussie, le statut du magasin de variantes passe à`DELETING`.

```
aws omics delete-variant-store --id <variant-store-id>
```

L'exemple suivant montre comment supprimer un magasin d'annotations. Si l'action aboutit, le statut du magasin d'annotations passe à`DELETING`. Les magasins d'annotations ne peuvent pas être supprimés s'il existe plusieurs versions.

```
aws omics delete-annotation-store --id <annotation-store-id>
```

# Interrogation de HealthOmics données analytiques

> **Important**
>
> AWS HealthOmics les magasins de variantes et les magasins d'annotations ne sont plus ouverts aux nouveaux clients. Les clients existants peuvent continuer à utiliser le service normalement. Pour de plus amples informations, veuillez consulter AWS HealthOmics modification de la disponibilité du magasin de variantes et du magasin d'annotations.

Vous pouvez effectuer des requêtes sur vos variantes de boutiques à l'aide AWS Lake Formation d'Amazon Athena ou d'Amazon EMR. Avant d'exécuter des requêtes, suivez les procédures de configuration (décrites dans les sections suivantes) pour Lake Formation et Amazon Athena.

Pour plus d'informations sur Amazon EMR, consultez Tutoriel : Commencer à utiliser Amazon EMR

Pour les variantes de magasins créées après le 26 septembre 2024, HealthOmics partitionnez le magasin par ID d'échantillon. Ce partitionnement signifie qu'il HealthOmics utilise l'identifiant de l'échantillon pour optimiser le stockage des informations sur les variantes. Les requêtes qui utilisent des exemples d'informations comme filtres renvoient des résultats plus rapidement, car elles analysent moins de données.

HealthOmics utilise un échantillon IDs comme nom de fichier de partition. Avant d'ingérer des données, vérifiez si l'identifiant d'échantillon contient des données PHI. Si c'est le cas, modifiez l'identifiant de l'échantillon avant d'ingérer les données. Pour plus d'informations sur le contenu à inclure et à ne pas inclure dans l'échantillon IDs, consultez les instructions sur la page Web de conformité à la loi AWS HIPAA.

Rubriques

- Configuration de Lake Formation à utiliser HealthOmics
- Configuration d'Athena pour les requêtes

- Exécution de requêtes sur des magasins de HealthOmics variantes

# Configuration de Lake Formation à utiliser HealthOmics

> **Important**
>
> AWS HealthOmics les magasins de variantes et les magasins d'annotations ne sont plus ouverts aux nouveaux clients. Les clients existants peuvent continuer à utiliser le service normalement. Pour de plus amples informations, veuillez consulter AWS HealthOmics modification de la disponibilité du magasin de variantes et du magasin d'annotations.

Avant d'utiliser Lake Formation pour gérer les magasins de HealthOmics données, effectuez les procédures de configuration de Lake Formation suivantes.

Rubriques

- Création ou vérification des administrateurs de Lake Formation
- Création de liens vers des ressources à l'aide de la console Lake Formation
- Configuration des autorisations pour les partages AWS RAM de ressources

## Création ou vérification des administrateurs de Lake Formation

Avant de créer un lac de données dans Lake Formation, vous devez définir un ou plusieurs administrateurs.

Les administrateurs sont des utilisateurs et des rôles autorisés à créer des liens vers des ressources. Vous configurez les administrateurs de data lake par compte et par région.

Créez un utilisateur administrateur dans la console Lake Formation

1. Ouvrez la console AWS Lake Formation : console Lake Formation
2. Si la console affiche le panneau Welcome to Lake Formation, choisissez Get started.

   Lake Formation vous ajoute à la table des administrateurs du Data Lake.
3. Sinon, dans le menu de gauche, sélectionnez Rôles et tâches d'administration.
4. Ajoutez des administrateurs supplémentaires si nécessaire.

## Création de liens vers des ressources à l'aide de la console Lake Formation

Pour créer une ressource partagée que les utilisateurs peuvent interroger, les contrôles d'accès par défaut doivent être désactivés. Pour en savoir plus sur la désactivation des contrôles d'accès par défaut, consultez la section Modification des paramètres de sécurité par défaut pour votre lac de données dans la documentation de Lake Formation. Vous pouvez créer des liens vers des ressources individuellement ou en groupe, afin d'accéder aux données d'Amazon Athena ou d'autres AWS services (tels qu'Amazon EMR).

Création de liens vers des ressources dans la console AWS Lake Formation et partage de ces liens avec les utilisateurs HealthOmics d'Analytics

1. Ouvrez la console AWS Lake Formation : console Lake Formation
2. Dans la barre de navigation principale, sélectionnez Bases de données.
3. Dans le tableau Bases de données, sélectionnez la base de données souhaitée.
4. Dans le menu Créer, choisissez Lien vers la ressource.
5. Entrez le nom du lien vers la ressource. Si vous prévoyez d'accéder à la base de données depuis Athéna, entrez un nom en minuscules uniquement (256 caractères maximum).
6. Choisissez Créer.
7. Le nouveau lien vers la ressource est désormais répertorié sous Bases de données.

Accorder l'accès à la ressource partagée à l'aide de la console Lake Formation

Un administrateur de base de données Lake Formation peut accorder l'accès à la ressource partagée en suivant la procédure suivante.

1. Ouvrez la console AWS Lake Formation : https://console.aws.amazon.com/lakeformation/
2. Dans la barre de navigation principale, sélectionnez Bases de données.
3. Sur la page Bases de données, sélectionnez le lien de ressource que vous avez créé précédemment.
4. Dans le menu Actions, choisissez Grant on target.
5. Sur la page Accorder les autorisations relatives aux données, sous Principaux, sélectionnez Utilisateurs ou rôles IAM.
6. Dans le menu déroulant Utilisateurs ou rôles IAM, recherchez l'utilisateur auquel vous souhaitez accorder l'accès.

7. Ensuite, sous Étiquettes LF ou carte de ressources de catalogue, sélectionnez l'option Ressources de catalogue de données nommées.
8. Dans le menu déroulant facultatif Tableaux, sélectionnez Toutes les tables ou la table que vous avez créée précédemment.
9. Dans la fiche Autorisations du tableau, sous Autorisations du tableau, choisissez Décrire et sélectionner.
10. Ensuite, choisissez Grant.

Pour consulter les autorisations de Lake Formation, choisissez Data Lake Permissions dans le volet de navigation principal. Le tableau indique les bases de données et les liens vers les ressources disponibles.

## Configuration des autorisations pour les partages AWS RAM de ressources

Dans la console AWS Lake Formation, consultez les autorisations en choisissant Autorisations du lac de données dans la barre de navigation principale. Sur la page Autorisations relatives aux données, vous pouvez consulter un tableau qui indique les types de ressources, les bases de données et **ARN** les informations relatives à une ressource partagée sous RAM Resource Share. Si vous devez accepter un partage de ressources AWS Resource Access Manager (AWS RAM), vous en AWS Lake Formation informe dans la console.

HealthOmics peut accepter implicitement les partages de AWS RAM ressources lors de la création du magasin. Pour accepter le partage de AWS RAM ressources, l'utilisateur ou le rôle IAM qui appelle les opérations `CreateVariantStore` ou `CreateAnnotationStore` API doit autoriser les actions suivantes :

- `ram:GetResourceShareInvitations`- Cette action permet HealthOmics de retrouver les invitations.
- `ram:AcceptResourceShareInvitation`- Cette action permet d' HealthOmics accepter l'invitation en utilisant un jeton FAS.

Sans ces autorisations, une erreur d'autorisation s'affiche lors de la création de la boutique.

Voici un exemple de politique qui inclut ces actions. Ajoutez cette politique à l'utilisateur ou au rôle IAM qui accepte le partage de AWS RAM ressources.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "omics:*",
        "ram:AcceptResourceShareInvitation",
        "ram:GetResourceShareInvitations"
      ],
      "Resource": "*"
    }
  ]
}
```

## Configuration d'Athena pour les requêtes

> **Important**
>
> AWS HealthOmics les magasins de variantes et les magasins d'annotations ne sont plus ouverts aux nouveaux clients. Les clients existants peuvent continuer à utiliser le service normalement. Pour de plus amples informations, veuillez consulter AWS HealthOmics modification de la disponibilité du magasin de variantes et du magasin d'annotations.

Vous pouvez utiliser Athena pour interroger des variantes et des annotations. Avant d'exécuter des requêtes, effectuez les tâches de configuration suivantes :

Rubriques

- Configuration de l'emplacement des résultats d'une requête à l'aide de la console Athena
- Configuration d'un groupe de travail avec le moteur Athena v3

### Configuration de l'emplacement des résultats d'une requête à l'aide de la console Athena

Pour configurer l'emplacement des résultats d'une requête, procédez comme suit.

1. Ouvrez la console Athena : Athena console
2. Dans la barre de navigation principale, choisissez l'éditeur de requêtes.
3. Dans l'éditeur de requêtes, choisissez l'onglet Paramètres, puis sélectionnez Gérer.
4. Entrez le préfixe S3 d'un emplacement pour enregistrer le résultat de la requête.

### Configuration d'un groupe de travail avec le moteur Athena v3

Pour configurer un groupe de travail, procédez comme suit.

1. Ouvrez la console Athena : Athena console
2. Dans la barre de navigation principale, choisissez Groupes de travail, puis Créer un groupe de travail.
3. Entrez un nom pour le groupe de travail.
4. Sélectionnez Athena SQL comme type de moteur.
5. Sous Mettre à niveau le moteur de requête, sélectionnez Manuel.
6. Sous Query version engine, sélectionnez Athena version 3.
7. Choisissez Créer un groupe de travail.

## Exécution de requêtes sur des magasins de HealthOmics variantes

> **Important**
>
> AWS HealthOmics les magasins de variantes et les magasins d'annotations ne sont plus ouverts aux nouveaux clients. Les clients existants peuvent continuer à utiliser le service normalement. Pour de plus amples informations, veuillez consulter AWS HealthOmics modification de la disponibilité du magasin de variantes et du magasin d'annotations.

Vous pouvez effectuer des requêtes sur votre boutique de variantes à l'aide d'Amazon Athena. Notez que les coordonnées génomiques dans les magasins de variants et d'annotations sont représentées sous forme d'intervalles de base zéro, mi-fermés, mi-ouverts.

## Exécutez une requête simple à l'aide de la console Athena

L'exemple suivant montre comment exécuter une requête simple.

1. Ouvrez l'éditeur de requêtes Athena : éditeur de requêtes Athena
2. Sous Groupe de travail, sélectionnez le groupe de travail que vous avez créé lors de la configuration.
3. Vérifiez que la source de données est AwsDataCatalog.
4. Pour Base de données, sélectionnez le lien de ressource de base de données que vous avez créé lors de la configuration de Lake Formation.
5. Copiez la requête suivante dans l'éditeur de requêtes sous l'onglet Requête 1 :

```
SELECT * from omicsvariants limit 10
```

6. Choisissez Exécuter pour exécuter la requête. La console remplit le tableau des résultats avec les 10 premières lignes du omicsvariants tableau.

## Exécuter une requête complexe à l'aide de la console Athena

L'exemple suivant montre comment exécuter une requête complexe. Pour exécuter cette requête, importez-la `ClinVar` dans le magasin d'annotations.

Exécuter une requête complexe

1. Ouvrez l'éditeur de requêtes Athena : éditeur de requêtes Athena
2. Sous Groupe de travail, sélectionnez le groupe de travail que vous avez créé lors de la configuration.
3. Vérifiez que la source de données est AwsDataCatalog.
4. Pour Base de données, sélectionnez le lien de ressource de base de données que vous avez créé lors de la configuration de Lake Formation.
5. Cliquez + sur le bouton en haut à droite pour créer un nouvel onglet de requête nommé Requête 2.

6. Copiez la requête suivante dans l'éditeur de requêtes sous l'onglet Requête 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. Choisissez Exécuter pour démarrer l'exécution de la requête.

# Partage de magasins HealthOmics d'analyse

> **Important**
>
> AWS HealthOmics les magasins de variantes et les magasins d'annotations ne sont plus ouverts aux nouveaux clients. Les clients existants peuvent continuer à utiliser le service normalement. Pour de plus amples informations, veuillez consulter AWS HealthOmics modification de la disponibilité du magasin de variantes et du magasin d'annotations.

En tant que propriétaire d'un magasin de variantes ou d'un magasin d'annotations, vous pouvez partager le magasin avec d'autres comptes AWS. Le propriétaire peut révoquer l'accès à la ressource partagée en supprimant le partage.

En tant qu'abonné à une boutique partagée, vous devez d'abord accepter le partage. Vous pouvez ensuite définir des flux de travail utilisant le magasin partagé. Les données apparaissent sous forme de tableau à la fois dans Lake Formation AWS Glue et dans Lake Formation.

Lorsque vous n'avez plus besoin d'accéder à la boutique, vous supprimez le partage.

Consultez Partage de ressources entre comptes dans AWS HealthOmics pour plus d'informations sur le partage des ressources.

## Création d'un partage de boutique

Pour créer un partage de boutique, utilisez l'opération d'API create-share. L'abonné principal est celui Compte AWS de l'utilisateur qui va souscrire au partage. L'exemple suivant crée un partage pour un magasin de variantes. Pour partager une boutique avec plusieurs comptes, vous devez créer plusieurs partages de la même boutique.

```
aws omics create-share  \
        --resource-arn "arn:aws:omics:us-west-2:555555555555:variantStore/
omics_dev_var_store" \
        --principal-subscriber "123456789012" \
        --name "my_Share-123"
```

Si la création est réussie, vous recevez une réponse avec l'ID et le statut du partage.

```
{
       "shareId": "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a",
       "name": "my_Share-123",
       "status": "PENDING"
  }
```

Le partage reste en attente jusqu'à ce que l'abonné l'accepte à l'aide de l'opération d'API accept-share.

# Partage de ressources entre comptes dans AWS HealthOmics

Utilisez le partage entre comptes pour partager des ressources avec des collaborateurs sans créer de copies ni modifier les politiques de ressources IAM. Les ressources suivantes prennent en charge le partage entre comptes :

- HealthOmics boutiques de variantes
- HealthOmics magasins d'annotations
- Flux de travail privés

Le partage d'une ressource inclut les étapes suivantes :

1. Le propriétaire de la ressource crée un partage et spécifie l'ARN de la ressource et celui Compte AWS de l'abonné prévu. Le partage de ressources reste en attente jusqu'à ce que l'abonné accepte le partage.
2. L'abonné accepte le partage de ressources pour avoir accès à la ressource. Le partage de ressources passe à l'état d'activation.
3. Le HealthOmics service fournit à un compte d'abonné un accès à la ressource.
4. Le propriétaire de la ressource peut supprimer le partage ou l'abonné peut révoquer son accès au partage. L'abonné ne peut pas supprimer le partage ou la ressource associée.

Rubriques

# Création d'un partage

Vous pouvez utiliser l'opération d'API create-share pour créer un partage. L'abonné principal est celui Compte AWS de l'utilisateur qui s'abonnera à la ressource partagée. L'exemple suivant crée un partage pour un magasin de variantes.

```
aws omics create-share \
    --resource-arn "arn:aws:omics:us-west-2:555555555555:variantStore/
omics_dev_var_store" \
    --principal-subscriber "123456789012"  \
    --name "my_Share-123"
```

Si la création est réussie, vous recevez une réponse avec l'ID et le statut du partage.

```
{
"shareId": "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a",
     "name": "my_Share-123",
     "status": "PENDING"
}
```

Le partage reste en attente jusqu'à ce que l'abonné l'accepte à l'aide de l'opération accept-share API.

```
 aws omics accept-share \
    --share-id "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a"
```

Une fois que l'abonné a accepté le partage, celui-ci passe à l'état actif.

```
{
"status": "ACTIVATING"
}
```

# Récupérer des informations sur un partage

Utilisez l'opération d'API get-share pour récupérer des informations sur le partage.

```
aws omics get-share --share-id "495c21bedc889d07d0ab69d710a6841e-
dd75ab7a1a9c384fa848b5bd8e5a7e0a"
```

La réponse de l'API inclut des informations de métadonnées concernant le partage.

```
{
  "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"
    }
}
```

## Afficher les actions que vous détenez

Utilisez l'API list-shares pour récupérer des informations sur chacun des partages que vous possédez.

```
aws omics list-shares  --resource-owner SELF
```

La réponse de l'API inclut les métadonnées de chaque partage dont vous êtes propriétaire.

## Afficher les actions acceptées depuis d'autres comptes

Utilisez l'API list-shares pour afficher tous les partages que vous avez acceptés depuis d'autres comptes.

```
aws omics list-shares  --resource-owner OTHER
```

La réponse de l'API inclut les métadonnées pour chaque partage que vous avez accepté.

## Supprimer un partage

Utilisez l'API delete-share pour supprimer un partage lorsque vous n'en avez plus besoin.

```
aws omics delete-share \
```

```
    --share-id "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a"
```

# Marquage des ressources dans HealthOmics

Rubriques



## Avis important

HealthOmics protège les données des clients conformément aux politiques du modèle de responsabilité partagée d'AWS. Cela signifie que toutes les données des clients sont cryptées à la fois pendant la transition et au repos. Cependant, les noms saisis par le client pour les ressources telles que les magasins de données ou les opérations basées sur des tâches ne sont pas tous cryptés. Ils ne doivent jamais contenir d'informations personnelles identifiables ou d'informations de santé protégées. Pour de plus amples informations, veuillez consulter Sécurité dans AWS HealthOmics.

## Ressources de balisage HealthOmics

Vous pouvez attribuer des métadonnées à vos ressources AWS à l'aide de balises. Chaque balise est une étiquette composée d'une clé définie par l'utilisateur et d'une valeur. Les balises peuvent vous aider à gérer, identifier, organiser, rechercher et filtrer des ressources.

Cette rubrique décrit les catégories et stratégies de balisage couramment utilisées pour vous aider à mettre en œuvre une stratégie de balisage cohérente et efficace. Les sections suivantes supposent des connaissances de base sur les ressources AWS, le balisage, la facturation détaillée et Gestion des identités et des accès AWS.

Chaque balise se compose de deux parties :

- Une clé de balise (par exemple CostCenter, Environnement ou Projet). Les touches de tag distinguent les majuscules et minuscules.

- Une valeur de balise (par exemple, 111122223333 ou Production). Les valeurs de balise sont sensibles à la casse, tout comme les clés de balise.

Vous pouvez utiliser des balises pour classer les ressources en fonction de leur objectif, de leur propriétaire, de leur environnement ou d'autres critères. Pour plus d'informations, consultez Stratégies de balisage AWS.

Vous pouvez ajouter, modifier ou supprimer des balises pour une ressource depuis la console de service de la ressource, l'API de service ou le AWS CLI.

Pour activer le balisage, assurez-vous qu' TagResources il est autorisé. Vous pouvez autoriser TagResources en joignant une politique IAM comme dans l'exemple suivant.

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": "*"
        }
    ]
}
```

## Bonnes pratiques

Lorsque vous créez une stratégie de balisage pour les ressources AWS, suivez les meilleures pratiques :

- Ne stockez pas d'informations personnelles identifiables (PII), d'informations de santé protégées (PHI) ou d'autres informations sensibles dans des balises.
- Utilisez un format standardisé et sensible à la casse pour les balises et appliquez-le de manière cohérente à tous les types de ressources.
- Optez pour des directives de balisage qui prennent en charge plusieurs objectifs, comme la gestion du contrôle d'accès aux ressources, le suivi des coûts, l'automatisation et l'organisation.
- Utilisez des outils automatisés pour gérer les balises de ressources. AWS Resource Groups et l'API Resource Groups Tagging permettent le contrôle programmatique des balises, ce qui permet de gérer, de rechercher et de filtrer automatiquement les balises et les ressources.
- Le balisage est plus efficace lorsque vous utilisez un plus grand nombre de balises.
- Les balises peuvent être éditées ou modifiées en fonction de l'évolution des besoins des utilisateurs. Toutefois, pour mettre à jour les balises de contrôle d'accès, vous devez également mettre à jour les politiques qui font référence à ces balises afin de contrôler l'accès à vos ressources.

## Balisage des exigences

Les balises possèdent les exigences suivantes :

- Les clés ne peuvent pas être préfixées par aws :.
- Les clés doivent être uniques par ensemble de balises.
- Une clé doit comporter entre 1 et 128 caractères autorisés.
- Une valeur doit comprendre entre 0 et 256 caractères autorisés.
- Les valeurs n'ont pas besoin d'être uniques par ensemble de balises.

- Les caractères autorisés pour les clés et les valeurs sont les lettres Unicode, les chiffres, les espaces et les symboles suivants : _ . : / = + - @.
- Les clés et les valeurs sont sensibles à la casse.

## Sequence : stockage, lecture et définition des balises

Pour les magasins de séquences, les balises créées sur le jeu de lecture se situent au niveau des ressources du jeu de lecture. Les ensembles de lecture contiennent également des objets situés en dessous qui peuvent être consultés, recherchés et restreints à l'aide de S3 APIs. Par défaut, l'identifiant de l'échantillon (omics:sampleId) et l'identifiant du sujet (omics:subjectid) sont ajoutés à l'objet.

En outre, jusqu'à cinq balises peuvent être synchronisées entre le jeu de lecture et les objets situés en dessous. La configuration des balises à synchroniser est une configuration au niveau du magasin définie lors de la création ou de la mise à jour du magasin à l'aide du `propogatedSetLevelTags` paramètre.

Si des données se trouvent déjà dans le magasin, la mise à jour des clés peut prendre du temps. Au cours de cette mise à jour, HealthOmics change le statut de la boutique enUpdating. À la fin, HealthOmics définit le statut du magasin surActive. Pendant la propagation des balises, les autorisations basées sur les balises peuvent ne pas être appliquées. Les autorisations seront appliquées une fois la propagation des balises terminée.

Lorsque des balises sont définies ou mises à jour sur le jeu de lecture, le système décide s'il convient de mettre à jour les objets pour ce jeu de lecture, en fonction de la configuration du magasin.

## Ajouter un tag à une HealthOmics ressource

L'ajout de balises à une ressource peut vous aider à identifier et à organiser vos ressources AWS et à gérer l'accès à celles-ci. Tout d'abord, vous ajoutez une ou plusieurs balises (paires clé-valeur) à une ressource. Vous pouvez utiliser jusqu'à 50 balises par ressource. Il existe également des restrictions concernant les caractères que vous pouvez utiliser dans les champs de clé et de valeur.

Après avoir ajouté des balises, vous pouvez créer des politiques IAM pour gérer l'accès à la AWS ressource en fonction de ces balises. Vous pouvez utiliser la HealthOmics console ou le AWS CLI pour ajouter des balises à une ressource. L'ajout de balises à un référentiel peut avoir un impact sur l'accès à ce référentiel. Avant d'ajouter une balise à un magasin de données, passez en revue les

politiques IAM susceptibles d'utiliser des balises pour contrôler l'accès aux ressources telles que les magasins de données.

Les étiquettes de service sont générées automatiquement pour un sujet et un identifiant d'échantillon pour les magasins de séquences.

Procédez comme suit pour utiliser le AWS CLI pour ajouter une balise à une HealthOmics ressource. Par exemple, pour ajouter des balises à un magasin de séquences lors de sa création, vous devez utiliser la commande suivante dans le AWS CLI. Le nom du magasin de séquences est MySequenceStore, et les deux balises ajoutées avec des clés sont key1 et key2 avec des valeurs comme value1 et value2 respectivement :

```
aws omics create-sequence-store --name "MySequenceStore" --tags key1=value1,key2=value2
```

La sortie ne répertorie pas les balises. Elle renvoie la réponse suivante.

```
{
    "id": "6860403586",
    "referenceStoreId": "4889894479",
    "roleArn": "arn:aws:iam::555555555555:role/ImportTest",
    "status": "CREATED",
    "creationTime": "2022-07-21T01:19:07.194Z"
}
```

Pour ajouter des balises à une ressource existante, vous devez exécuter l'exemple de commande suivant.

```
aws omics tag-resource --resource-arn arn:aws:omics:us-
west-2:555555555555:sequenceStore/2275234794 --tags key1=value1,key2=value2
```

En cas de succès, cette commande ne renvoie aucune réponse.

## Répertorier les tags d'une ressource

Procédez comme suit pour utiliser le AWS CLI pour afficher la liste des AWS balises d'une HealthOmics ressource. Si aucune balise n'a été ajoutée, la liste renvoyée est vide.

Sur le terminal ou sur la ligne de commande, exécutez la list-tags-for-resource commande comme indiqué dans l'exemple suivant.

```
aws omics list-tags-for-resource --resource-arn arn:aws:omics:us-
west-2:555555555555:sequenceStore/2275234794
```

Vous recevrez une liste de balises en réponse, au format JSON.

```
 {
    "tags": {
        "key1": "value1",
        "key2": "value2"
    }
}
```

## Supprimer des balises d'un magasin de données

Vous pouvez supprimer une ou plusieurs balises associées à une ressource. La suppression d'une balise ne supprime pas la balise des autres ressources AWS associées à cette balise.

Sur le terminal ou sur la ligne de commande, exécutez la commande untag-resource en spécifiant le nom de ressource Amazon (ARN) de la ressource dont vous souhaitez supprimer les balises et la clé de balise de la balise que vous souhaitez supprimer.

```
aws omics untag-resource --resource-arn arn:aws:omics:us-
west-2:555555555555:sequenceStore/2275234794 --tag-keys key1,key2
```

En cas de succès, cette commande ne renvoie aucune réponse. Pour vérifier les balises associées à la ressource, exécutez la commande list-tags-for-resource.

# Autorisations IAM pour HealthOmics

Vous pouvez utiliser Gestion des identités et des accès AWS (IAM) pour gérer l'accès à l' HealthOmics API et aux ressources telles que les magasins et les flux de travail. Pour les utilisateurs et les applications de votre compte qui les utilisent HealthOmics, vous gérez les autorisations dans le cadre d'une politique d'autorisation que vous pouvez appliquer aux utilisateurs, aux groupes ou aux rôles IAM.

Pour gérer les autorisations accordées aux utilisateurs et aux applications de vos comptes, utilisez les politiques HealthOmics fournies ou rédigez les vôtres. La HealthOmics console utilise plusieurs services pour obtenir des informations sur la configuration et les déclencheurs de votre fonction. Vous pouvez utiliser les politiques fournies telles quelles ou comme point de départ pour des politiques plus restrictives.

HealthOmics utilise les rôles de service IAM pour accéder à d'autres services en votre nom. Par exemple, vous pouvez créer ou choisir un rôle de service lorsque vous exécutez un flux de travail qui lit des données depuis Amazon S3. Pour certaines fonctionnalités, vous devez également configurer les autorisations sur les ressources d'autres services. Passez en revue ces exigences avant de commencer à travailler avec HealthOmics

Pour plus d'informations sur IAM, consultez Qu'est-ce qu'IAM ? dans le Guide de l'utilisateur IAM.

Rubriques



## Politiques IAM basées sur l'identité pour HealthOmics

Pour autoriser les utilisateurs de votre compte à accéder à HealthOmics, vous utilisez des politiques basées sur l'identité dans Gestion des identités et des accès AWS (IAM). Les politiques basées sur l'identité peuvent s'appliquer directement aux utilisateurs IAM ou aux groupes et rôles IAM associés à un utilisateur. Vous pouvez également accorder aux utilisateurs d'un autre compte l'autorisation d'assumer un rôle dans votre compte et d'accéder à vos ressources HealthOmics.

Pour autoriser les utilisateurs à effectuer des actions sur une version de flux de travail, vous devez ajouter le flux de travail et la version de flux de travail spécifique à la liste des ressources.

La politique IAM suivante permet à un utilisateur d'accéder à toutes les actions de HealthOmics l'API et de leur transmettre des rôles de HealthOmics service.

Example Stratégie utilisateur

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "omics:*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "iam:PassRole"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": "omics.amazonaws.com"
        }
      }
    }
  ]
}
```

Lorsque vous les utilisez HealthOmics, vous interagissez également avec d'autres AWS services. Pour accéder à ces services, utilisez les politiques gérées fournies par chaque service. Pour restreindre l'accès à un sous-ensemble de ressources, vous pouvez utiliser les politiques gérées comme point de départ pour créer vos propres politiques plus restrictives.

- AmazonS3 FullAccess — Accès aux compartiments et aux objets Amazon S3 utilisés par les tâches.
- Amazon EC2 ContainerRegistryFullAccess — Accès aux registres et référentiels Amazon ECR pour les images des conteneurs de flux de travail.
- AWSLakeFormationDataAdmin— Accès aux bases de données et aux tables de Lake Formation créées par les magasins d'analyse.
- ResourceGroupsandTagEditorFullAccess— Marquez HealthOmics les ressources à l'aide des opérations d'API HealthOmics de balisage.

Les politiques précédentes n'autorisent pas un utilisateur à créer des rôles IAM. Pour qu'un utilisateur disposant de ces autorisations puisse exécuter une tâche, un administrateur doit créer le rôle de service qui accorde l' HealthOmics autorisation d'accéder aux sources de données. Pour de plus amples informations, veuillez consulter Rôles de service pour AWS HealthOmics.

## Définissez des autorisations IAM personnalisées pour les exécutions

Vous pouvez inclure n'importe quel flux de travail, exécution ou groupe d'exécution référencé par la `StartRun` demande dans une demande d'autorisation. Pour ce faire, listez la combinaison souhaitée de flux de travail, d'exécutions ou de groupes d'exécution dans la politique IAM. Par exemple, vous pouvez limiter l'utilisation d'un flux de travail à une exécution ou à un groupe d'exécution spécifique. Vous pouvez également spécifier qu'un flux de travail ne doit être utilisé qu'avec un groupe d'exécution.

Voici un exemple de politique IAM qui autorise un flux de travail unique avec un seul groupe d'exécution.

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/*"
            ]
        }
    ]
}
```

# Rôles de service pour AWS HealthOmics

Un rôle de service est un rôle Gestion des identités et des accès AWS (IAM) qui autorise un AWS service à accéder aux ressources de votre compte. Vous attribuez un rôle de service AWS HealthOmics lorsque vous démarrez une tâche d'importation ou que vous lancez une exécution.

La HealthOmics console peut créer le rôle requis pour vous. Si vous utilisez l' HealthOmics API pour gérer les ressources, créez le rôle de service à l'aide de la console IAM. Pour plus d'informations, voir Créer un rôle pour déléguer des autorisations à un Service AWS.

Les rôles de service doivent respecter la politique de confiance suivante.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "omics.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

La politique de confiance permet au HealthOmics service d'assumer le rôle.

Rubriques

- Exemples de politiques de service IAM
- Exemple de CloudFormation modèle

## Exemples de politiques de service IAM

Dans ces exemples, les noms des ressources et IDs les comptes sont des espaces réservés que vous pouvez remplacer par des valeurs réelles.

L'exemple suivant montre la politique d'un rôle de service que vous pouvez utiliser pour démarrer une exécution. La politique accorde des autorisations pour accéder à l'emplacement de sortie Amazon S3, au groupe de journaux du flux de travail et au conteneur Amazon ECR pour l'exécution.

> **Note**
>
> Si vous utilisez la mise en cache des appels pour l'exécution, ajoutez l'emplacement du cache d'exécution Amazon S3 en tant que ressource dans les autorisations s3.

Example Politique de rôle de service pour le démarrage d'une exécution

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/*"
            ]
        }
    ]
}
```

L'exemple suivant montre la politique d'un rôle de service que vous pouvez utiliser pour une tâche d'importation de boutique. La politique accorde des autorisations pour accéder à l'emplacement d'entrée Amazon S3.

Example Rôle de service pour une tâche dans un magasin de référence

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"
            ]
        }

    ]
}
```

## Exemple de CloudFormation modèle

L'exemple de CloudFormation modèle suivant crée un rôle de service qui donne HealthOmics l'autorisation d'accéder aux compartiments Amazon S3 dont le nom est préfixé par et de télécharger des `omics-` journaux de flux de travail.

Example Autorisations relatives au magasin de référence, Amazon S3 et CloudWatch aux journaux

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

# Autorisations Amazon ECR

Avant que le HealthOmics service puisse exécuter un flux de travail dans un conteneur à partir de votre référentiel Amazon ECR privé, vous devez créer une politique de ressources pour le référentiel. La politique autorise le HealthOmics service à utiliser le conteneur. Vous ajoutez cette politique de ressources à chaque référentiel privé référencé par le flux de travail.

> **Note**
>
> Le référentiel privé et le flux de travail doivent se trouver dans la même région.

Si différents AWS comptes possèdent le flux de travail et le référentiel, vous devez configurer les autorisations entre comptes.

Il n'est pas nécessaire d'accorder un accès supplémentaire au référentiel pour les flux de travail partagés. Vous pouvez toutefois créer des politiques qui autorisent ou interdisent à des flux de travail spécifiques l'accès à l'image du conteneur.

Pour utiliser la fonction de cache d'extraction d'Amazon ECR, vous devez créer une politique d'autorisation de registre.

Les sections suivantes décrivent comment configurer les autorisations des ressources Amazon ECR pour ces scénarios. Pour plus d'informations sur les autorisations dans Amazon ECR, consultez la section Autorisations de registre privé dans Amazon ECR.

Rubriques

- Création d'une politique de ressources pour le référentiel Amazon ECR
- Exécution de flux de travail avec des conteneurs multi-comptes
- Politiques Amazon ECR pour les flux de travail partagés
- Politiques relatives au cache d'extraction Amazon ECR

# Création d'une politique de ressources pour le référentiel Amazon ECR

Créez une politique de ressources pour permettre au HealthOmics service d'exécuter un flux de travail à l'aide d'un conteneur dans le référentiel. La politique autorise le directeur du HealthOmics service à accéder aux actions Amazon ECR requises.

Pour créer la politique, procédez comme suit :

1. Ouvrez la page des référentiels privés dans la console Amazon ECR et sélectionnez le référentiel auquel vous accordez l'accès.
2. Dans la barre de navigation latérale, sélectionnez Autorisations.
3. Choisissez Modifier.
4. Choisissez Modifier la politique JSON.
5. Ajoutez la déclaration de politique suivante, puis sélectionnez Enregistrer.

   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": "*"
        }
    ]
}
```

# Exécution de flux de travail avec des conteneurs multi-comptes

Si différents AWS comptes possèdent le flux de travail et le conteneur, vous devez configurer les autorisations inter-comptes suivantes :

1. Mettez à jour la politique Amazon ECR pour le référentiel afin d'accorder explicitement l'autorisation au compte propriétaire du flux de travail.
2. Mettez à jour le rôle de service du compte propriétaire du flux de travail pour lui accorder l'accès à l'image du conteneur.

L'exemple suivant illustre une politique de ressources Amazon ECR qui accorde l'accès au compte propriétaire du flux de travail.

Dans cet exemple :

- ID du compte Workflow : 111122223333
- ID du compte du dépôt de conteneurs : 444455556666
- Nom du conteneur : 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": "*"
        }
    ]
}
```

Pour terminer la configuration, ajoutez la déclaration de politique suivante au rôle de service du compte propriétaire du flux de travail. La politique autorise le rôle de service à accéder à l'image du conteneur « samtools ». Assurez-vous de remplacer les numéros de compte, le nom du conteneur et la région par vos propres valeurs.

```
{
    "Sid": "CrossAccountEcrRepoPolicy",
    "Effect": "Allow",
    "Action": ["ecr:BatchCheckLayerAvailability", "ecr:BatchGetImage",
 "ecr:GetDownloadUrlForLayer"],
    "Resource": "arn:aws:ecr:us-west-2:444455556666:repository/samtools"
}
```

## Politiques Amazon ECR pour les flux de travail partagés

> **Note**
>
> HealthOmics permet automatiquement à un flux de travail partagé d'accéder au référentiel Amazon ECR dans le compte du propriétaire du flux de travail, pendant que le flux de travail s'exécute dans le compte de l'abonné. Il n'est pas nécessaire d'accorder un accès supplémentaire au référentiel pour les flux de travail partagés. Pour plus d'informations, voir Partage de HealthOmics flux de travail.

Par défaut, les abonnés n'ont pas accès au référentiel Amazon ECR pour utiliser les conteneurs sous-jacents. Vous pouvez éventuellement personnaliser l'accès au référentiel Amazon ECR en

ajoutant des clés de condition à la politique de ressources du référentiel. Les sections suivantes fournissent des exemples de politiques.

## Restreindre l'accès à des flux de travail spécifiques

Vous pouvez répertorier les flux de travail individuels dans une déclaration de condition, de sorte que seuls ces flux de travail peuvent utiliser des conteneurs dans le référentiel. La clé de SourceArncondition spécifie l'ARN du flux de travail partagé. L'exemple suivant autorise le flux de travail spécifié à utiliser ce référentiel.

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"
          }
       }
     }
  ]
}
```

## Restreindre l'accès à des comptes spécifiques

Vous pouvez répertorier les comptes d'abonnés dans un énoncé de condition, afin que seuls ces comptes soient autorisés à utiliser les conteneurs du référentiel. La clé de SourceAccountcondition

indique le Compte AWS nom de l'abonné. L'exemple suivant autorise le compte spécifié à utiliser ce référentiel.

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"
        }
      }
    }
  ]
}
```

Vous pouvez également refuser les autorisations Amazon ECR à des abonnés spécifiques, comme indiqué dans l'exemple de politique suivant.

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"
              }
            }
        }
    ]
}
```

## Politiques relatives au cache d'extraction Amazon ECR

Pour utiliser le cache d'extraction Amazon ECR, vous devez créer une politique d'autorisation de registre. Vous créez également un modèle de création de référentiel, qui définit les autorisations pour les référentiels créés par le cache d'extraction Amazon ECR.

Les sections suivantes contiennent des exemples de ces politiques. Pour plus d'informations sur le cache d'extraction, consultez Synchroniser un registre en amont avec un registre privé Amazon ECR dans le guide de l'utilisateur du registre Amazon Elastic Container Registry.

### Politique d'autorisation du registre

Pour utiliser le cache d'extraction Amazon ECR, créez une politique d'autorisation de registre. La politique d'autorisations du registre permet de contrôler les autorisations de réplication et d'extraction du cache.

Pour la réplication entre comptes, vous devez explicitement autoriser chacun d'entre eux Compte AWS à répliquer ses référentiels dans votre registre.

Par défaut, lorsque vous créez une règle de cache d'extraction, tout principal IAM autorisé à extraire des images d'un registre privé peut également utiliser la règle de cache d'extraction. Vous pouvez utiliser les autorisations de registre pour limiter davantage ces autorisations à des référentiels spécifiques.

Ajoutez une politique d'autorisation de registre au compte propriétaire de l'image du conteneur.

Dans l'exemple suivant, la politique permet au HealthOmics service de créer des référentiels pour chaque registre en amont et de lancer des pull requests en amont à partir des référentiels créés.

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/*"
            ]
        }
    ]
}
```

## Modèle de création de référentiel

Pour utiliser le cache pull through HealthOmics, le référentiel Amazon ECR doit disposer d'un modèle de création de référentiel. Le modèle définit les paramètres de configuration pour les référentiels privés créés pour un registre en amont.

Chaque modèle contient un préfixe d'espace de noms de référentiel, qu'Amazon ECR utilise pour associer les nouveaux référentiels à un modèle spécifique. Les modèles peuvent spécifier la configuration de tous les paramètres du référentiel, y compris les politiques d'accès basées sur les ressources, l'immuabilité des balises, le chiffrement et les politiques de cycle de vie. Pour plus d'informations, consultez la section Modèles de création de référentiels dans le guide de l'utilisateur d'Amazon Elastic Container Registry.

Dans l'exemple suivant, la politique autorise le HealthOmics service à lancer des pull requests en amont à partir des référentiels en amont.

JSON

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

## Politiques relatives à l'accès multicompte à Amazon ECR

Pour l'accès entre comptes, le propriétaire du référentiel privé met à jour la politique d'autorisation du registre et le modèle de création du référentiel afin d'autoriser l'accès à l'autre compte et au rôle d'exécution de ce compte.

Dans la politique d'autorisation du registre, ajoutez une déclaration de politique pour autoriser le rôle d'exécution de l'autre compte à accéder aux actions Amazon ECR :

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowCrossAccountPTCinRegPermissions",
            "Effect": "Allow",
```

```
            "Principal": {
                "AWS": "arn:aws:iam::123456789012:role/RUN_ROLE"},
            "Action": [
                "ecr:CreateRepository",
                "ecr:BatchGetImage",
                "ecr:BatchImportUpstreamImage"
            ],
            "Resource": "arn:aws:ecr:us-east-1:123456789012:repository/path/*"
        }
    ]
}
```

Dans le modèle de création de référentiel, ajoutez une déclaration de politique pour autoriser le rôle d'exécution de l'autre compte à accéder aux nouvelles images du conteneur. Vous pouvez éventuellement ajouter des instructions de condition pour limiter l'accès à des flux de travail spécifiques :

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"
                }
            }
        }
    ]
```

```
}
```

Ajoutez des autorisations pour deux actions supplémentaires (CreateRepository et BatchImportUpstreamImage) dans le rôle d'exécution et spécifiez la ressource à laquelle le rôle d'exécution peut accéder.

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 Autorisations relatives aux ressources

AWS HealthOmics crée et accède aux ressources d'autres services en votre nom lorsque vous exécutez une tâche ou créez une boutique. Dans certains cas, vous devez configurer des autorisations dans d'autres services pour accéder aux ressources ou pour autoriser HealthOmics l'accès à celles-ci.

Pour les autorisations relatives aux ressources liées à Amazon ECR, consultezAutorisations Amazon ECR.

# Autorisations Lake Formation

Avant d'utiliser les fonctionnalités d'analyse HealthOmics, configurez les paramètres de base de données par défaut dans Lake Formation.

Pour configurer les autorisations relatives aux ressources dans Lake Formation

1. Ouvrez la page des paramètres du catalogue de données dans la console Lake Formation.
2. Décochez les exigences de contrôle d'accès IAM pour les bases de données et les tables sous Autorisations par défaut pour les bases de données et les tables nouvellement créées.
3. Choisissez Enregistrer.

HealthOmics Analytics accepte automatiquement les données si votre politique de service dispose des autorisations de RAM appropriées, comme dans l'exemple suivant.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "omics:*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ram:AcceptResourceShareInvitation",
        "ram:GetResourceShareInvitations"
      ],
      "Resource": "*"
    }
  ]
}
```

# Autorisations d'accès aux données à l'aide d'Amazon S3 URIs

Vous pouvez accéder aux données du magasin de séquences à l'aide HealthOmics d'opérations d'API ou d'opérations d'API Amazon S3.

Pour HealthOmics l'accès aux API, HealthOmics les autorisations sont gérées par le biais d'une politique IAM. Cependant, S3 Access nécessite deux niveaux de configuration : une autorisation explicite dans la politique d'accès S3 du Store et une politique IAM. Pour en savoir plus sur l'utilisation des politiques IAM avec HealthOmics, consultez la section Rôles de service pour HealthOmics.

Il existe trois manières de partager la capacité de lecture d'objets à l'aide d'Amazon S3 APIs :

1. Partage basé sur des politiques : ce partage nécessite d'activer le principal IAM à la fois dans la politique d'accès S3 et d'écrire une politique IAM et de l'associer au principal IAM. Consultez la rubrique suivante pour plus de détails.
2. Présigné URLs — Vous pouvez également générer une URL pré-signée partageable pour un fichier dans le magasin de séquences. Pour en savoir plus sur la création de présignés URLs à l'aide d'Amazon S3, consultez la section Utilisation de présignés URLs dans la documentation Amazon S3. La politique d'accès S3 du magasin de séquences prend en charge les instructions visant à limiter les capacités des URL présignées.
3. Rôles assumés : créez un rôle dans le compte du propriétaire des données doté d'une politique d'accès permettant aux utilisateurs d'assumer ce rôle.

Rubriques



## Partage basé sur des politiques

Si vous accédez aux données du magasin de séquences à l'aide d'une URI S3 directe, HealthOmics fournit des mesures de sécurité renforcées pour la politique d'accès au compartiment S3 associée.

Les règles suivantes s'appliquent aux nouvelles politiques d'accès S3. Pour les politiques existantes, les règles s'appliquent lors de la prochaine mise à jour de la politique :

- Les politiques d'accès S3 prennent en charge les éléments de politique suivants

  - Version, identifiant, déclaration, side, effet, principal, action, ressource, condition
- Les politiques d'accès S3 prennent en charge les clés de condition suivantes :
  - s3 :ExistingObjectTag/<key>, s3 : préfixe, s3 : version de signature, s3 : TlsVersion
  - Les politiques prennent également en charge aws : PrincipalArn avec les opérateurs de condition suivants : ArnEquals et ArnLike

Si vous essayez d'ajouter ou de mettre à jour une politique afin d'inclure un élément ou une condition non pris en charge, le système rejette la demande.

Rubriques

- Politique d'accès S3 par défaut
- Personnalisation de la politique d'accès
- Politique IAM
- Contrôle d'accès basé sur les étiquettes

## Politique d'accès S3 par défaut

Lorsque vous créez un magasin de séquences, HealthOmics crée une politique d'accès S3 par défaut accordant au compte racine du propriétaire du magasin de données les autorisations suivantes pour tous les objets accessibles du magasin de séquences : S3 : GetObjectGetObjectTagging, S3 et S3 :ListBucket. La politique créée par défaut est la suivante :

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/*"
        }
    ]
}
```

## Personnalisation de la politique d'accès

Si la politique d'accès S3 est vide, aucun accès S3 n'est autorisé. S'il existe une politique existante et que vous devez supprimer l'accès s3, utilisez-la `deleteS3AccessPolicy` pour supprimer tous les accès.

Pour ajouter des restrictions au partage ou pour accorder l'accès à d'autres comptes, vous pouvez mettre à jour la politique à l'aide de l'`PutS3AccessPolicy`API. Les mises à jour de la politique ne peuvent pas aller au-delà du préfixe du magasin de séquences ou des actions spécifiées.

## Politique IAM

Pour autoriser un utilisateur ou un utilisateur principal IAM à l'aide d'Amazon S3 APIs, en plus de l'autorisation prévue dans la politique d'accès S3, une politique IAM doit être créée et attachée au principal pour accorder l'accès. Une politique autorisant l'accès à l'API Amazon S3 peut être appliquée au niveau du magasin de séquences ou au niveau du jeu de lecture. Au niveau de l'ensemble de lecture, les autorisations peuvent être restreintes soit par le biais du préfixe, soit à l'aide de filtres de balises de ressources pour les modèles d'identification des échantillons ou des sujets.

Si le magasin de séquences utilise une clé gérée par le client (CMK), le principal doit également avoir le droit d'utiliser la clé KMS pour le déchiffrement. Pour plus d'informations, consultez la section Accès KMS entre comptes dans le Guide du AWS Key Management Service développeur.

L'exemple suivant donne à un utilisateur l'accès à un magasin de séquences. Vous pouvez affiner l'accès à l'aide de conditions supplémentaires ou de filtres basés sur les ressources.

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/*"
```

```
        }
      }
    }
  ]
}
```

## Contrôle d'accès basé sur les étiquettes

Pour utiliser le contrôle d'accès basé sur les balises, le magasin de séquences doit d'abord être mis à jour pour propager les clés de balise qui seront utilisées. Cette configuration est définie lors de la création ou de la mise à jour du magasin de séquences. Une fois les balises propagées, les conditions des balises peuvent être utilisées pour ajouter des restrictions supplémentaires. Les restrictions peuvent être placées dans la politique d'accès S3 ou dans la politique IAM. Voici un exemple de politique d'accès S3 basée sur des onglets qui serait définie :

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

# Exemple de restriction

Scénario : création d'un partage dans lequel le propriétaire des données peut restreindre la capacité d'un utilisateur à télécharger des données « retirées ».

Dans ce scénario, un propriétaire de données (compte #111111111111) gérait un magasin de données. Ce propriétaire de données partage les données avec un large éventail d'utilisateurs tiers, y compris un chercheur (compte #999999999999). Dans le cadre de la gestion des données, le propriétaire des données reçoit régulièrement des demandes de retrait des données d'un participant. Pour gérer ce retrait, le propriétaire des données restreint d'abord l'accès direct au téléchargement à la réception de la demande et supprime finalement les données selon ses besoins.

Pour répondre à ce besoin, le propriétaire des données met en place un magasin de séquences et chaque ensemble de lecture reçoit une étiquette de « statut » qui sera définie sur « retiré » si la demande de retrait est acceptée. Pour les données dont la balise est définie sur cette valeur, ils veulent s'assurer qu'aucun utilisateur ne peut exécuter « GetObject » sur ce fichier. Pour effectuer cette configuration, le propriétaire des données doit s'assurer que deux étapes sont suivies.

Étape 1. Pour le magasin de séquences, assurez-vous que la balise d'état est mise à jour pour être propagée. Cela se fait en ajoutant la touche « status » dans le champ `propogatedSetLevelTags` lors de l'appel `createSequenceStore` ou `updateSequenceStore.`

Étape 2. Mettez à jour la politique d'accès s3 du magasin pour restreindre GetObject aux objets dont la balise d'état est définie sur « retiré ». Cela se fait en mettant à jour la politique d'accès aux boutiques à l'aide de l'`PutS3AccesPolicy`API. La politique suivante permettrait aux clients de toujours voir les fichiers retirés lorsqu'ils mettent en vente des objets, mais les empêcherait d'y accéder :

- Déclaration 1 (restrictedGetWithdrawal) : Le compte 999999999999 ne peut pas récupérer les objets retirés.
- Déclaration 2 (ownerGetAll) : Le compte 111111111111, propriétaire des données, peut récupérer tous les objets, y compris les objets retirés.
- Déclaration 3 (everyoneListAll) : Tous les comptes partagés, 111111111111 et 999999999999, peuvent exécuter l'opération sur l'ensemble du préfixe. 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/*"
                }
            }
        }
    ]
}
```

# Sécurité dans AWS HealthOmics

La sécurité du cloud AWS est la priorité absolue. En tant que AWS client, vous bénéficiez de centres de données et d'architectures réseau conçus pour répondre aux exigences des entreprises les plus sensibles en matière de sécurité.

La sécurité est une responsabilité partagée entre vous AWS et vous. Le modèle de responsabilité partagée décrit ceci comme la sécurité du cloud et la sécurité dans le cloud :

- Sécurité du cloud : AWS est chargée de protéger l'infrastructure qui exécute les AWS services dans le AWS Cloud. AWS vous fournit également des services que vous pouvez utiliser en toute sécurité. Des auditeurs tiers testent et vérifient régulièrement l'efficacité de notre sécurité dans le cadre des programmes de AWS conformité Programmes de de conformité. Pour en savoir plus sur les programmes de conformité qui s'appliquent à AWS HealthOmics, consultez la section AWS Services concernés par programme de conformitéAWS .
- Sécurité dans le cloud — Votre responsabilité est déterminée par le AWS service que vous utilisez. Vous êtes également responsable d'autres facteurs, y compris de la sensibilité de vos données, des exigences de votre entreprise, ainsi que de la législation et de la réglementation applicables.

Cette documentation vous aide à comprendre comment appliquer le modèle de responsabilité partagée lors de l'utilisation d'AWS HealthOmics. Les rubriques suivantes expliquent comment configurer AWS pour répondre HealthOmics à vos objectifs de sécurité et de conformité. Vous apprendrez également à utiliser d'autres AWS services qui vous aident à surveiller et à sécuriser vos HealthOmics ressources AWS.

Rubriques

# Protection des données dans AWS HealthOmics

Le modèle de responsabilité AWS partagée de s'applique à la protection des données dans AWS HealthOmics. Comme décrit dans ce modèle, AWS est chargé de protéger l'infrastructure mondiale qui gère tous les AWS Cloud. La gestion du contrôle de votre contenu hébergé sur cette infrastructure relève de votre responsabilité. Vous êtes également responsable des tâches de configuration et de gestion de la sécurité des Services AWS que vous utilisez. Pour plus d'informations sur la confidentialité des données, consultez Questions fréquentes (FAQ) sur la confidentialité des données. Pour en savoir plus sur la protection des données en Europe, consultez le billet de blog Modèle de responsabilité partagée d'AWS et RGPD (Règlement général sur la protection des données) sur le Blog de sécuritéAWS .

À des fins de protection des données, nous vous recommandons de protéger les Compte AWS informations d'identification et de configurer les utilisateurs individuels avec AWS IAM Identity Center ou Gestion des identités et des accès AWS (IAM). Ainsi, chaque utilisateur se voit attribuer uniquement les autorisations nécessaires pour exécuter ses tâches. Nous vous recommandons également de sécuriser vos données comme indiqué ci-dessous :

- Utilisez l'authentification multifactorielle (MFA) avec chaque compte.
- SSL/TLS À utiliser pour communiquer avec AWS les ressources. Nous exigeons TLS 1.2 et recommandons TLS 1.3.
- Configurez l'API et la journalisation de l'activité des utilisateurs avec AWS CloudTrail. Pour plus d'informations sur l'utilisation des CloudTrail sentiers pour capturer AWS des activités, consultez la section Utilisation des CloudTrail sentiers dans le guide de AWS CloudTrail l'utilisateur.
- Utilisez des solutions de AWS chiffrement, ainsi que tous les contrôles de sécurité par défaut qu'ils contiennent Services AWS.
- Utilisez des services de sécurité gérés avancés tels qu'Amazon Macie, qui contribuent à la découverte et à la sécurisation des données sensibles stockées dans Amazon S3.
- Si vous avez besoin de modules cryptographiques validés par la norme FIPS 140-3 pour accéder AWS via une interface de ligne de commande ou une API, utilisez un point de terminaison FIPS. Pour plus d'informations sur les points de terminaison FIPS disponibles, consultez Norme FIPS (Federal Information Processing Standard) 140-3.

Nous vous recommandons fortement de ne jamais placer d'informations confidentielles ou sensibles, telles que les adresses e-mail de vos clients, dans des balises ou des champs de texte libre tels que le champ Nom. Cela inclut lorsque vous travaillez avec AWS HealthOmics ou une autre entreprise

Services AWS à l'aide de la console AWS CLI, de l'API ou AWS SDKs. Toutes les données que vous entrez dans des balises ou des champs de texte de forme libre utilisés pour les noms peuvent être utilisées à des fins de facturation ou dans les journaux de diagnostic. Si vous fournissez une adresse URL à un serveur externe, nous vous recommandons fortement de ne pas inclure d'informations d'identification dans l'adresse URL permettant de valider votre demande adressée à ce serveur.

## Chiffrement au repos

Rubriques



Pour protéger les données sensibles des clients au repos, AWS HealthOmics fournit un chiffrement par défaut à l'aide d'une clé AWS Key Management Service (AWS KMS) appartenant au service. Les clés gérées par le client sont également prises en charge. Pour en savoir plus sur les clés gérées par le client, consultez Amazon Key Management Service.

Tous les magasins de HealthOmics données (stockage et analyse) prennent en charge l'utilisation de clés gérées par le client. La configuration du chiffrement ne peut pas être modifiée après la création d'un magasin de données. Si un magasin de données utilise un Clé détenue par AWS, il sera désigné comme tel AWS_OWNED_KMS_KEY et vous ne verrez pas la clé spécifique utilisée pour le chiffrement au repos.

Pour les HealthOmics flux de travail, les clés gérées par le client ne sont pas prises en charge par le système de fichiers temporaire ; toutefois, toutes les données sont chiffrées automatiquement au repos à l'aide de l'algorithme de chiffrement par blocs XTS-AES-256 pour chiffrer le système de fichiers. L'utilisateur et le rôle IAM utilisés pour démarrer l'exécution d'un flux de travail doivent également avoir accès aux AWS KMS clés utilisées pour les compartiments d'entrée et de sortie du flux de travail. Les flux de travail n'utilisent pas de licences et AWS KMS le chiffrement est limité aux compartiments Amazon S3 en entrée et en sortie. Le rôle IAM utilisé à la fois pour le flux de travail APIs doit également avoir accès aux AWS KMS clés utilisées ainsi qu'aux compartiments Amazon S3 d'entrée et de sortie. Vous pouvez utiliser les rôles et les autorisations IAM pour contrôler l'accès

ou AWS KMS les politiques. Pour en savoir plus, consultez Authentification et contrôle d'accès pour AWS KMS.

Lorsque vous l'utilisez AWS Lake Formation avec HealthOmics Analytics, toutes les autorisations de déchiffrement associées à la Lake Formation sont également accordées aux compartiments Amazon S3 en entrée et en sortie. Vous trouverez plus d'informations sur la façon dont AWS Lake Formation les autorisations sont gérées dans la AWS Lake Formation documentation.

HealthOmics Analytics accorde à Lake Formation kms: Decrypt l'autorisation de lire les données chiffrées dans un compartiment Amazon S3. Tant que vous êtes autorisé à interroger les données via Lake Formation, vous pourrez lire les données cryptées. L'accès aux données est contrôlé par le biais du contrôle d'accès aux données dans Lake Formation, et non par le biais d'une politique de clé KMS. Pour en savoir plus, consultez les demandes de service AWS AWS intégrées dans la documentation de Lake Formation.

### Clés détenues par AWS

Par défaut, HealthOmics utilise Clés détenues par AWS pour chiffrer automatiquement les données au repos, car ces données peuvent contenir des informations sensibles telles que des informations personnelles identifiables (PII) ou des informations de santé protégées (PHI). Clés détenues par AWS ne sont pas enregistrés dans votre compte. Elles font partie d'un ensemble de clés KMS qu'AWS possède et gère pour être utilisées dans plusieurs comptes AWS.

Les services AWS peuvent être utilisés Clés détenues par AWS pour protéger vos données. Vous ne pouvez ni consulter, ni gérer, ni accéder à leur utilisation Clés détenues par AWS, ni en vérifier l'utilisation. Cependant, vous n'avez pas besoin de travailler ou de modifier de programme pour protéger les clés qui chiffrent vos données.

Aucuns frais mensuels ni frais d'utilisation ne vous sont facturés Clés détenues par AWS, et ils ne sont pas pris en compte dans les quotas AWS KMS de votre compte. Pour de plus amples informations, veuillez consulter Clés gérées par AWS.

### Clés gérées par le client

HealthOmics prend en charge l'utilisation de clés symétriques gérées par le client que vous créez, détenez et gérez pour ajouter une deuxième couche de chiffrement au chiffrement existant appartenant à AWS. Étant donné que vous avez le contrôle total de cette couche de chiffrement, vous pouvez effectuer les tâches suivantes :

- Établir et maintenir des politiques clés, des politiques IAM et des subventions

- Rotation des matériaux de chiffrement de clé
- Activation et désactivation des stratégies de clé
- Ajout de balises
- Création d'alias de clé
- Planification des clés pour la suppression

Vous pouvez également l' CloudTrail utiliser pour suivre les demandes HealthOmics envoyées AWS KMS en votre nom. Des AWS KMS frais supplémentaires s'appliquent. Pour plus d'informations, consultez la section Clés gérées par le client.

## Création d'une clé gérée par le client

Vous pouvez créer une clé symétrique gérée par le client à l'aide de l'AWS Management Console ou du AWS KMS APIs.

Suivez les étapes de création de clés symétriques gérées par le client dans le guide du développeur AWS Key Management Service.

Les stratégies de clés contrôlent l'accès à votre clé gérée par le client. Chaque clé gérée par le client doit avoir exactement une stratégie de clé, qui contient des instructions qui déterminent les personnes pouvant utiliser la clé et comment elles peuvent l'utiliser. Lorsque vous créez une clé gérée par le client, vous pouvez définir une politique clé. Pour plus d'informations, consultez la section Gestion de l'accès aux clés gérées par le client dans le guide du développeur AWS Key Management Service.

Pour utiliser une clé gérée par le client avec vos ressources HealthOmics Analytics, le principal appelant a besoin de kms : CreateGrant operations dans la politique des clés. Cela permet au système d'utiliser un jeton FAS pour créer une attribution à une clé gérée par le client qui contrôle l'accès à une clé KMS spécifiée. Cette clé permet à un utilisateur d'accéder aux opérations kms:grant requises. HealthOmics Voir Utilisation des subventions pour plus d'informations.

À des fins HealthOmics d'analyse, les opérations d'API suivantes doivent être autorisées pour le principal appelant :

- kms : CreateGrant ajoute des autorisations à une clé spécifique gérée par le client, ce qui permet d'accéder aux opérations d'attribution dans HealthOmics Analytics.
- kms : DescribeKey fournit les informations clés gérées par le client nécessaires pour valider la clé. Cela est obligatoire pour toutes les opérations.

- kms : GenerateDataKey fournit un accès aux ressources de chiffrement au repos pour toutes les opérations d'écriture. Cette action fournit également des informations clés gérées par le client que le service peut utiliser pour valider que l'appelant a accès à la clé.
- KMS:Decrypt permet d'accéder aux opérations de lecture ou de recherche de ressources chiffrées.

Pour utiliser une clé gérée par le client avec vos ressources HealthOmics de stockage, le principal de HealthOmics service et le principal appelant doivent être autorisés dans la politique des clés. Cela permet au service de valider que l'appelant a accès à la clé et utilise le principal du service pour exécuter la gestion du magasin à l'aide de la clé gérée par le client. Pour le HealthOmics stockage, la politique clé du principal de service doit autoriser les opérations d'API suivantes :

- kms : DescribeKey fournit les informations clés gérées par le client nécessaires pour valider la clé. Cela est obligatoire pour toutes les opérations.
- kms : GenerateDataKey fournit un accès aux ressources de chiffrement au repos pour toutes les opérations d'écriture. Cette action fournit également des informations clés gérées par le client que le service peut utiliser pour valider que l'appelant a accès à la clé.
- KMS:Decrypt permet d'accéder aux opérations de lecture ou de recherche de ressources chiffrées.

L'exemple suivant montre une déclaration de politique qui permet à un directeur de service de créer et d'interagir avec une HealthOmics séquence ou un magasin de référence chiffré à l'aide de la clé gérée par le client :

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
{
    "Effect": "Allow",
    "Principal": {
        "Service": "omics.amazonaws.com"
    },
    "Action": [
        "kms:Decrypt",
        "kms:DescribeKey",
        "kms:Encrypt",
        "kms:GenerateDataKey*"
```

```
    ],
    "Resource": "*"
        }
    ]
}
```

L'exemple suivant montre une politique qui crée des autorisations permettant à un magasin de données de déchiffrer les données d'un compartiment 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"
                  ]
              }
            }
        }
    ]
}
```

## Autorisations IAM requises pour utiliser une clé gérée par le client

Lors de la création d'une ressource telle qu'un magasin de données AWS KMS chiffré à l'aide d'une clé gérée par le client, des autorisations sont requises à la fois pour la politique de clé et pour la stratégie IAM pour l'utilisateur ou le rôle IAM.

Vous pouvez utiliser la clé de ViaService condition kms : pour limiter l'utilisation de la clé KMS aux seules demandes provenant de HealthOmics.

Pour plus d'informations sur les politiques clés, consultez la section Activation des politiques IAM dans le guide du développeur d'AWS Key Management Service.

Rubriques

- Autorisations relatives à l'API Analytics
- Autorisations de l'API de stockage
- Comment HealthOmics utilise les subventions dans AWS KMS
- Surveillance de vos clés de chiffrement pour AWS HealthOmics

### Autorisations relatives à l'API Analytics

Pour les HealthOmics analyses, l'utilisateur ou le rôle IAM qui crée les magasins doit disposer des autorisations kms: CreateGrant kms:GenerateDataKey, kms: Déchiffrer et des autorisations, ainsi que kms: DescribeKey des autorisations nécessaires HealthOmics .

### Autorisations de l'API de stockage

Pour le HealthOmics stockage APIs, l'utilisateur ou le rôle IAM qui appelle les opérations d'API suivantes a besoin des autorisations répertoriées :

CreateReferenceStore, CreateSequenceStore

Pour créer une boutique, l'appelant IAM doit disposer des `kms:DescribeKey` autorisations ainsi que des autorisations nécessaires HealthOmics . Le directeur du HealthOmics service appelle `kms:GenerateDataKeyWithoutPlaintext` pour effectuer des contrôles de validation d'accès pour le chargement et l'accès aux données.

StartReadSetImportJob, StartReferenceImportJob

Pour démarrer des tâches d'importation de données, l'appelant IAM doit disposer `kms:Decrypt` d'`kms:GenerateDataKey`autorisations pour la clé KMS sur le magasin d'importation, ainsi que d'`kms:Decrypt`autorisations sur le compartiment Amazon S3 contenant les objets à importer. En outre, le rôle transmis à l'appel doit disposer d'`kms:Decrypt`autorisations sur le compartiment Amazon S3 contenant les objets à importer. L'appelant IAM doit également être autorisé à transmettre le rôle à la tâche.

CreateMultipartReadSetUpload, UploadReadSetPart, CompleteMultipartReadSetUpload

Pour effectuer un téléchargement en plusieurs parties, l'appelant IAM doit avoir créé, `kms:GenerateDataKey` télécharger `kms:Decrypt` et terminer le téléchargement en plusieurs parties.

StartReadSetExportJob

Pour démarrer une tâche d'exportation de données, l'appelant IAM doit être `kms:Decrypt` autorisé à utiliser la clé KMS sur le magasin à exporter depuis `kms:GenerateDataKey` et `kms:Decrypt` sur le compartiment Amazon S3 qui reçoit les objets. En outre, le rôle transmis à l'appel doit disposer d'`kms:Decrypt`autorisations sur le compartiment Amazon S3 qui reçoit les objets. L'appelant IAM doit également être autorisé à transmettre le rôle à la tâche.

StartReadsetActivationJob

Pour démarrer une tâche d'activation d'un ensemble de lecture, l'appelant IAM doit disposer des `kms:GenerateDataKey` autorisations nécessaires pour `kms:Decrypt` les objets.

GetReference, GetReadSet

Pour lire des objets depuis le magasin, l'appelant IAM doit avoir l'`kms:Decrypt`autorisation d'accéder aux objets.

Set de lecture S3 GetObject

Pour lire des objets depuis le magasin à l'aide de l'`GetObject`API Amazon S3, l'appelant IAM doit avoir `kms:Decrypt` l'autorisation d'accéder aux objets. Définissez cette autorisation pour les clés gérées par le client et pour les Clé détenue par AWS configurations.

Comment HealthOmics utilise les subventions dans AWS KMS

HealthOmics Analytics nécessite une autorisation pour utiliser votre clé KMS gérée par le client. Les subventions ne sont ni requises ni utilisées pour les HealthOmics flux de travail. HealthOmics Le stockage utilise la clé gérée par le client directement auprès du principal du service. N'utilisez donc pas de subvention. Lorsque vous créez un magasin d'analyse chiffré à l'aide d'une clé gérée par le client, HealthOmics Analytics crée une subvention en votre nom en envoyant une CreateGrantdemande à AWS KMS. Les subventions dans AWS KMS sont utilisées pour donner HealthOmics accès à une clé KMS dans un compte client.

Il n'est pas recommandé de révoquer ou de retirer les autorisations créées par HealthOmics Analytics en votre nom. Si vous révoquez ou retirez l' HealthOmics autorisation d'utiliser les clés AWS KMS de votre compte, vous HealthOmics ne pouvez pas accéder à ces données, chiffrer les nouvelles ressources envoyées au magasin de données ou les déchiffrer lorsqu'elles sont extraites.

Lorsque vous révoquez ou retirez une subvention pour HealthOmics, le changement intervient immédiatement. Pour révoquer les droits d'accès, nous vous recommandons de supprimer le magasin de données plutôt que de révoquer l'autorisation. Lorsque vous supprimez le magasin de données, HealthOmics les subventions sont supprimées en votre nom.

Surveillance de vos clés de chiffrement pour AWS HealthOmics

Vous pouvez l'utiliser CloudTrail pour suivre les demandes AWS HealthOmics envoyées en votre AWS KMS nom lorsque vous utilisez une clé gérée par le client. Les entrées du CloudTrail journal indiquent HealthOmics .Amazonaws.com dans le champ UserAgent afin de distinguer clairement les demandes effectuées par. HealthOmics

Les exemples suivants sont CloudTrail des événements pour CreateGrant GenerateDataKey, déchiffrer et surveiller les AWS KMS opérations appelées DescribeKey pour accéder HealthOmics aux données chiffrées par votre clé gérée par le client.

Ce qui suit montre également comment autoriser les HealthOmics analyses CreateGrant à accéder à une clé KMS fournie par le client, ce qui permet HealthOmics d'utiliser cette clé KMS pour chiffrer toutes les données client au repos.

Vous n'êtes pas obligé de créer vos propres subventions. HealthOmics crée une subvention en votre nom en envoyant une CreateGrant demande à AWS KMS. Les subventions AWS KMS sont utilisées pour donner HealthOmics accès à une AWS KMS clé dans un compte client.


```
{
    "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"
}
```

L'exemple suivant montre comment s' GenerateDataKey assurer que l'utilisateur dispose des autorisations nécessaires pour chiffrer les données avant de les stocker.

```
{
    "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"
}
```


## En savoir plus

Les ressources suivantes fournissent des informations supplémentaires sur le chiffrement des données au repos.

Pour plus d'informations sur les concepts de base d'AWS Key Management Service, consultez la AWS KMS documentation.

Pour plus d'informations sur les meilleures pratiques en matière de sécurité, AWS KMS consultez la documentation.

### Chiffrement en transit

AWS HealthOmics utilise le protocole TLS 1.2+ pour chiffrer les données en transit via les points de terminaison publics et via les services principaux.

## Gestion des identités et des accès dans HealthOmics

Gestion des identités et des accès AWS (IAM) est un outil Service AWS qui permet à un administrateur de contrôler en toute sécurité l'accès aux AWS ressources. Les administrateurs IAM contrôlent qui peut être authentifié (connecté) et autorisé (autorisé) à utiliser les ressources AWS HealthOmics . IAM est un Service AWS outil que vous pouvez utiliser sans frais supplémentaires.

Rubriques



### Public ciblé

La façon dont vous utilisez Gestion des identités et des accès AWS (IAM) varie en fonction de votre rôle :

- Utilisateur du service : demandez des autorisations à votre administrateur si vous ne pouvez pas accéder aux fonctionnalités (voir Résolution des problèmes AWS HealthOmics d'identité et d'accès)
- Administrateur du service : déterminez l'accès des utilisateurs et soumettez les demandes d'autorisation (voir Comment AWS HealthOmics fonctionne avec IAM)

- Administrateur IAM : rédigez des politiques pour gérer l'accès (voir Exemples de politiques basées sur l'identité pour AWS HealthOmics)

## Authentification par des identités

L'authentification est la façon dont vous vous connectez à AWS l'aide de vos informations d'identification. Vous devez être authentifié en tant qu'utilisateur IAM ou en assumant un rôle IAM. Utilisateur racine d'un compte AWS

Vous pouvez vous connecter en tant qu'identité fédérée à l'aide d'informations d'identification provenant d'une source d'identité telle que AWS IAM Identity Center (IAM Identity Center), d'une authentification unique ou d'informations d'identification. Google/Facebook Pour plus d'informations sur la connexion, consultez Connexion à votre Compte AWS dans le Guide de l'utilisateur Connexion à AWS .

Pour l'accès par programmation, AWS fournit un SDK et une CLI pour signer les demandes de manière cryptographique. Pour plus d'informations, consultez Signature AWS Version 4 pour les demandes d'API dans le Guide de l'utilisateur IAM.

### Compte AWS utilisateur root

Lorsque vous créez un Compte AWS, vous commencez par une seule identité de connexion appelée utilisateur Compte AWS root qui dispose d'un accès complet à toutes Services AWS les ressources. Il est vivement déconseillé d'utiliser l'utilisateur racine pour vos tâches quotidiennes. Pour les tâches qui requièrent des informations d'identification de l'utilisateur racine, consultez Tâches qui requièrent les informations d'identification de l'utilisateur racine dans le Guide de l'utilisateur IAM.

### Identité fédérée

Il est recommandé d'obliger les utilisateurs humains à utiliser la fédération avec un fournisseur d'identité pour accéder à Services AWS l'aide d'informations d'identification temporaires.

Une identité fédérée est un utilisateur provenant de votre annuaire d'entreprise, de votre fournisseur d'identité Web ou Directory Service qui y accède à Services AWS l'aide d'informations d'identification provenant d'une source d'identité. Les identités fédérées assument des rôles qui fournissent des informations d'identification temporaires.

Pour une gestion des accès centralisée, nous vous recommandons d'utiliser AWS IAM Identity Center. Pour plus d'informations, consultez Qu'est-ce que IAM Identity Center ? dans le Guide de l'utilisateur AWS IAM Identity Center .

### Utilisateurs et groupes IAM

Un utilisateur IAM est une identité qui dispose d'autorisations spécifiques pour une seule personne ou application. Nous vous recommandons d'utiliser ces informations d'identification temporaires au lieu des utilisateurs IAM avec des informations d'identification à long terme. Pour plus d'informations, voir Exiger des utilisateurs humains qu'ils utilisent la fédération avec un fournisseur d'identité pour accéder à AWS l'aide d'informations d'identification temporaires dans le guide de l'utilisateur IAM.

Les groupes IAM spécifient une collection d'utilisateurs IAM et permettent de gérer plus facilement les autorisations pour de grands ensembles d'utilisateurs. Pour plus d'informations, consultez Cas d'utilisation pour les utilisateurs IAM dans le Guide de l'utilisateur IAM.

### Rôles IAM

Un rôle IAM est une identité dotée d'autorisations spécifiques qui fournit des informations d'identification temporaires. Vous pouvez assumer un rôle en passant d'un rôle d'utilisateur à un rôle IAM (console) ou en appelant une opération d' AWS API AWS CLI ou d'API. Pour plus d'informations, consultez Méthodes pour endosser un rôle dans le Guide de l'utilisateur IAM.

Les rôles IAM sont utiles pour l'accès des utilisateurs fédérés, les autorisations temporaires des utilisateurs IAM, les accès entre comptes, les accès entre services et pour les applications exécutées sur Amazon. EC2 Pour plus d'informations, consultez Accès intercompte aux ressources dans IAM dans le Guide de l'utilisateur IAM.

## Gestion de l'accès à l'aide de politiques

Vous contrôlez l'accès en AWS créant des politiques et en les associant à AWS des identités ou à des ressources. Une politique définit les autorisations lorsqu'elles sont associées à une identité ou à une ressource. AWS évalue ces politiques lorsqu'un directeur fait une demande. La plupart des politiques sont stockées AWS sous forme de documents JSON. Pour plus d'informations les documents de politique JSON, consultez Vue d'ensemble des politiques JSON dans le Guide de l'utilisateur IAM.

À l'aide de politiques, les administrateurs précisent qui a accès à quoi en définissant quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

Par défaut, les utilisateurs et les rôles ne disposent d'aucune autorisation. Un administrateur IAM crée des politiques IAM et les ajoute aux rôles, que les utilisateurs peuvent ensuite assumer. Les politiques IAM définissent les autorisations quelle que soit la méthode que vous utilisez pour exécuter l'opération.

## Politiques basées sur l'identité

Les stratégies basées sur l'identité sont des documents de stratégie d'autorisations JSON que vous attachez à une identité (utilisateur, groupe ou rôle). Ces politiques contrôlent les actions que peuvent exécuter ces identités, sur quelles ressources et dans quelles conditions. Pour découvrir comment créer une politique basée sur l'identité, consultez Définition d'autorisations IAM personnalisées avec des politiques gérées par le client dans le Guide de l'utilisateur IAM.

Les politiques basées sur l'identité peuvent être des politiques intégrées (intégrées directement dans une seule identité) ou des politiques gérées (politiques autonomes associées à plusieurs identités). Pour découvrir comment choisir entre des politiques gérées et en ligne, consultez Choix entre les politiques gérées et les politiques en ligne dans le Guide de l'utilisateur IAM.

## Politiques basées sur les ressources

Les politiques basées sur les ressources sont des documents de politique JSON que vous attachez à une ressource. Les exemples incluent les politiques de confiance de rôle IAM et les stratégies de compartiment Amazon S3. Dans les services qui sont compatibles avec les politiques basées sur les ressources, les administrateurs de service peuvent les utiliser pour contrôler l'accès à une ressource spécifique. Vous devez spécifier un principal dans une politique basée sur les ressources.

Les politiques basées sur les ressources sont des politiques en ligne situées dans ce service. Vous ne pouvez pas utiliser les politiques AWS gérées par IAM dans une stratégie basée sur les ressources.

## Autres types de politique

AWS prend en charge des types de politiques supplémentaires qui peuvent définir les autorisations maximales accordées par les types de politiques les plus courants :

- Limites d'autorisations : une limite des autorisations définit le nombre maximum d'autorisations qu'une politique basée sur l'identité peut accorder à une entité IAM. Pour plus d'informations, consultez Limites d'autorisations pour des entités IAM dans le Guide de l'utilisateur IAM.
- Politiques de contrôle des services (SCPs) — Spécifiez les autorisations maximales pour une organisation ou une unité organisationnelle dans AWS Organizations. Pour plus d'informations, consultez Politiques de contrôle de service du Guide de l'utilisateur AWS Organizations .
- Politiques de contrôle des ressources (RCPs) : définissez le maximum d'autorisations disponibles pour les ressources de vos comptes. Pour plus d'informations, voir Politiques de contrôle des ressources (RCPs) dans le guide de AWS Organizations l'utilisateur.

- Politiques de session : politiques avancées que vous passez en tant que paramètre lorsque vous créez par programmation une session temporaire pour un rôle ou un utilisateur fédéré. Pour plus d'informations, consultez Politiques de session dans le Guide de l'utilisateur IAM.

### Plusieurs types de politique

Lorsque plusieurs types de politiques s'appliquent à la requête, les autorisations en résultant sont plus compliquées à comprendre. Pour savoir comment AWS déterminer s'il faut autoriser une demande lorsque plusieurs types de politiques sont impliqués, consultez la section Logique d'évaluation des politiques dans le guide de l'utilisateur IAM.

## Comment AWS HealthOmics fonctionne avec IAM

Avant d'utiliser IAM pour gérer l'accès à AWS HealthOmics, découvrez quelles fonctionnalités IAM peuvent être utilisées avec AWS. HealthOmics

Fonctionnalités IAM que vous pouvez utiliser avec AWS HealthOmics

| Fonctionnalité IAM | HealthOmics soutien |
|---|---|
| Politiques basées sur l'identité | Oui |
| Politiques basées sur les ressources | Non |
| Actions de politique | Oui |
| Ressources de politique | Oui |
| Clés de condition d'une politique | Non |
| ACLs | Non |
| ABAC (balises dans les politiques) | Oui |
| Informations d'identification temporaires | Oui |
| Autorisations de principal | Oui |
| Rôles de service | Oui |

| Fonctionnalité IAM | HealthOmics soutien |
| --- | --- |
| Rôles liés à un service | Non |

Pour obtenir une vue d'ensemble de la façon dont HealthOmics les autres AWS services fonctionnent avec la plupart des fonctionnalités IAM, consultez la section AWS Services compatibles avec IAM dans le Guide de l'utilisateur IAM.

## Prévention du problème de l'adjoint confus entre services

Le problème de député confus est un problème de sécurité dans lequel une entité qui n'est pas autorisée à effectuer une action peut contraindre une entité plus privilégiée à le faire. En AWS, l'usurpation d'identité interservices peut entraîner la confusion des adjoints. L'usurpation d'identité entre services peut se produire lorsqu'un service (le service appelant) appelle un autre service (le service appelé). Le service appelant peut être manipulé pour utiliser ses autorisations afin d'agir sur les ressources d'un autre client de sorte qu'il n'y aurait pas accès autrement. Pour éviter cela, AWS fournit des outils qui vous aident à protéger vos données pour tous les services avec des principaux de service qui ont eu accès aux ressources de votre compte.

Nous recommandons d'utiliser les clés de contexte de condition `aws:SourceAccount`globale `aws:SourceArn`et les clés de contexte dans les politiques de ressources afin de limiter les autorisations qu'AWS HealthOmics accorde à un autre service à la ressource.

Pour éviter de semer la confusion dans les rôles assumés par les adjoints HealthOmics, définissez la valeur de `aws:SourceArn` to `arn:aws:omics:`*`region`*`:`*`accountNumber`*`:*` dans la politique de confiance du rôle. Le caractère générique (`*`) applique la condition à toutes les HealthOmics ressources.

La politique de relation de confiance suivante donne HealthOmics accès à vos ressources et utilise les clés de contexte de condition `aws:SourceAccount` globale `aws:SourceArn` et globale pour éviter le problème de confusion des adjoints. Utilisez cette politique lorsque vous créez un rôle pour 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:*"
        }
      }
    }
  ]
}
```

## Politiques basées sur l'identité pour HealthOmics

Prend en charge les politiques basées sur l'identité : oui

Les politiques basées sur l'identité sont des documents de politique d'autorisations JSON que vous pouvez attacher à une identité telle qu'un utilisateur, un groupe d'utilisateurs ou un rôle IAM. Ces politiques contrôlent quel type d'actions des utilisateurs et des rôles peuvent exécuter, sur quelles ressources et dans quelles conditions. Pour découvrir comment créer une politique basée sur l'identité, consultez Définition d'autorisations IAM personnalisées avec des politiques gérées par le client dans le Guide de l'utilisateur IAM.

Avec les politiques IAM basées sur l'identité, vous pouvez spécifier des actions et ressources autorisées ou refusées, ainsi que les conditions dans lesquelles les actions sont autorisées ou refusées. Pour découvrir tous les éléments que vous utilisez dans une politique JSON, consultez Références des éléments de politique JSON IAM dans le Guide de l'utilisateur IAM.

Exemples de politiques basées sur l'identité pour HealthOmics

Pour consulter des exemples de politiques HealthOmics basées sur l'identité AWS, consultez. Exemples de politiques basées sur l'identité pour AWS HealthOmics

## Politiques basées sur les ressources au sein de HealthOmics

Prend en charge les politiques basées sur les ressources : non

Les politiques basées sur les ressources sont des documents de politique JSON que vous attachez à une ressource. Par exemple, les politiques de confiance de rôle IAM et les politiques de compartiment Amazon S3 sont des politiques basées sur les ressources. Dans les services qui sont compatibles avec les politiques basées sur les ressources, les administrateurs de service peuvent les utiliser pour contrôler l'accès à une ressource spécifique. Pour la ressource dans laquelle se trouve la politique, cette dernière définit quel type d'actions un principal spécifié peut effectuer sur cette ressource et dans quelles conditions. Vous devez spécifier un principal dans une politique basée sur les ressources. Les principaux peuvent inclure des comptes, des utilisateurs, des rôles, des utilisateurs fédérés ou. Services AWS

Pour permettre un accès intercompte, vous pouvez spécifier un compte entier ou des entités IAM dans un autre compte en tant que principal dans une politique basée sur les ressources. Pour plus d'informations, consultez Accès intercompte aux ressources dans IAM dans le Guide de l'utilisateur IAM.

## Actions politiques pour HealthOmics

Prend en charge les actions de politique : oui

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément `Action` d'une politique JSON décrit les actions que vous pouvez utiliser pour autoriser ou refuser l'accès à une politique. Intégration d'actions dans une politique afin d'accorder l'autorisation d'exécuter les opérations associées.

Pour consulter la liste des HealthOmics actions, consultez la section Actions définies par AWS HealthOmics dans le Service Authorization Reference.

Les actions de politique en HealthOmics cours utilisent le préfixe suivant avant l'action :

```
omics
```

Pour indiquer plusieurs actions dans une seule déclaration, séparez-les par des virgules.

```
"Action": [
      "omics:action1",
      "omics:action2"
         ]
```

Pour consulter des exemples de politiques HealthOmics basées sur l'identité AWS, consultez. Exemples de politiques basées sur l'identité pour AWS HealthOmics

## Ressources politiques pour HealthOmics

Prend en charge les ressources de politique : oui

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément de politique JSON `Resource` indique le ou les objets auxquels l'action s'applique. Il est recommandé de définir une ressource à l'aide de son Amazon Resource Name (ARN). Pour les actions qui ne sont pas compatibles avec les autorisations de niveau ressource, utilisez un caractère générique (*) afin d'indiquer que l'instruction s'applique à toutes les ressources.

```
"Resource": "*"
```

Pour consulter la liste des types de HealthOmics ressources et leurs caractéristiques ARNs, consultez la section Ressources définies par AWS HealthOmics dans la référence d'autorisation de service. Pour savoir avec quelles actions vous pouvez spécifier l'ARN de chaque ressource, consultez Actions définies par AWS HealthOmics .

Pour consulter des exemples de politiques HealthOmics basées sur l'identité AWS, consultez. Exemples de politiques basées sur l'identité pour AWS HealthOmics

## Clés de conditions de politique pour HealthOmics

Les clés de condition de politique ne sont pas prises en charge dans HealthOmics.

## Listes de contrôle d'accès (ACLs) dans HealthOmics

Supports ACLs : Non

Les listes de contrôle d'accès (ACLs) contrôlent les principaux (membres du compte, utilisateurs ou rôles) autorisés à accéder à une ressource. ACLs sont similaires aux politiques basées sur les ressources, bien qu'elles n'utilisent pas le format de document de politique JSON.

## Contrôle d'accès basé sur les attributs (ABAC) avec HealthOmics

Prise en charge d'ABAC (balises dans les politiques) : Oui

Le contrôle d'accès par attributs (ABAC) est une stratégie d'autorisation qui définit les autorisations en fonction des attributs appelés balises. Vous pouvez associer des balises aux entités et aux AWS ressources IAM, puis concevoir des politiques ABAC pour autoriser les opérations lorsque la balise du principal correspond à la balise de la ressource.

Pour contrôler l'accès basé sur des étiquettes, vous devez fournir les informations d'étiquette dans l'élément de condition d'une politique utilisant les clés de condition `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` ou `aws:TagKeys`.

Si un service prend en charge les trois clés de condition pour tous les types de ressources, alors la valeur pour ce service est Oui. Si un service prend en charge les trois clés de condition pour certains types de ressources uniquement, la valeur est Partielle.

Pour plus d'informations sur ABAC, consultez Définition d'autorisations avec l'autorisation ABAC dans le Guide de l'utilisateur IAM. Pour accéder à un didacticiel décrivant les étapes de configuration de l'ABAC, consultez Utilisation du contrôle d'accès par attributs (ABAC) dans le Guide de l'utilisateur IAM.

Pour plus d'informations sur le balisage des ressources HealthOmics, consultez Marquage des ressources dans HealthOmics.

L'exemple suivant montre comment vous pouvez écrire une politique IAM refusant l'accès à une ressource sans balise spécifique.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "omics:*"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "Null": {
                  "aws:RequestTag/MyCustomTag": "true"
                }
            }
        }
    ]
}
```

## Utilisation d'informations d'identification temporaires avec HealthOmics

Prend en charge les informations d'identification temporaires : oui

Les informations d'identification temporaires fournissent un accès à court terme aux AWS ressources et sont automatiquement créées lorsque vous utilisez la fédération ou que vous changez de rôle. AWS recommande de générer dynamiquement des informations d'identification temporaires au lieu d'utiliser des clés d'accès à long terme. Pour plus d'informations, consultez Informations d'identification de sécurité temporaires dans IAM et Services AWS compatibles avec IAM dans le Guide de l'utilisateur IAM.

## Autorisations principales interservices pour HealthOmics

Prend en charge les sessions d'accès direct (FAS) : oui

Les sessions d'accès direct (FAS) utilisent les autorisations du principal appelant et Service AWS, combinées Service AWS à la demande d'envoi de demandes aux services en aval. Pour plus de détails sur la politique relative à la transmission de demandes FAS, consultez la section Sessions de transmission d'accès.

### Rôles de service pour HealthOmics

Prend en charge les rôles de service : oui

Un rôle de service est un rôle IAM qu'un service endosse pour accomplir des actions en votre nom. Un administrateur IAM peut créer, modifier et supprimer un rôle de service à partir d'IAM. Pour plus d'informations, consultez Création d'un rôle pour la délégation d'autorisations à un Service AWS dans le Guide de l'utilisateur IAM.

> **Warning**
>
> La modification des autorisations associées à un rôle de service peut perturber HealthOmics les fonctionnalités. Modifiez les rôles de service uniquement lorsque HealthOmics vous recevez des instructions à cet effet.

### Rôles liés à un service pour HealthOmics

Prend en charge les rôles liés à un service : non

Un rôle lié à un service est un type de rôle de service lié à un. Service AWS Le service peut endosser le rôle afin d'effectuer une action en votre nom. Les rôles liés au service apparaissent dans votre Compte AWS fichier et appartiennent au service. Un administrateur IAM peut consulter, mais ne peut pas modifier, les autorisations concernant les rôles liés à un service.

Pour plus d'informations sur la création ou la gestion des rôles liés à un service, consultez Services AWS qui fonctionnent avec IAM. Recherchez un service dans le tableau qui inclut un `Yes` dans la colonne Rôle lié à un service. Choisissez le lien Oui pour consulter la documentation du rôle lié à ce service.

## Exemples de politiques basées sur l'identité pour AWS HealthOmics

Par défaut, les utilisateurs et les rôles ne sont pas autorisés à créer ou à modifier HealthOmics des ressources AWS. Pour octroyer aux utilisateurs des autorisations d'effectuer des actions sur les ressources dont ils ont besoin, un administrateur IAM peut créer des politiques IAM.

Pour apprendre à créer une politique basée sur l'identité IAM à l'aide de ces exemples de documents de politique JSON, consultez Création de politiques IAM (console) dans le Guide de l'utilisateur IAM.

Pour plus de détails sur les actions et les types de ressources définis par AWS HealthOmics, y compris le format ARNs de chaque type de ressource, consultez la section Actions, ressources et clés de condition pour AWS HealthOmics dans la référence d'autorisation de service.

Rubriques

- Bonnes pratiques en matière de politiques
- Utilisation de la console HealthOmics
- Autorisation accordée aux utilisateurs pour afficher leurs propres autorisations

## Bonnes pratiques en matière de politiques

Les politiques basées sur l'identité déterminent si quelqu'un peut créer, accéder ou supprimer HealthOmics des ressources AWS dans votre compte. Ces actions peuvent entraîner des frais pour votre Compte AWS. Lorsque vous créez ou modifiez des politiques basées sur l'identité, suivez ces instructions et recommandations :

- Commencez AWS par les politiques gérées et passez aux autorisations du moindre privilège : pour commencer à accorder des autorisations à vos utilisateurs et à vos charges de travail, utilisez les politiques AWS gérées qui accordent des autorisations pour de nombreux cas d'utilisation courants. Ils sont disponibles dans votre Compte AWS. Nous vous recommandons de réduire davantage les autorisations en définissant des politiques gérées par les AWS clients spécifiques à vos cas d'utilisation. Pour plus d'informations, consultez politiques gérées par AWS ou politiques gérées par AWS pour les activités professionnelles dans le Guide de l'utilisateur IAM.
- Accordez les autorisations de moindre privilège : lorsque vous définissez des autorisations avec des politiques IAM, accordez uniquement les autorisations nécessaires à l'exécution d'une seule tâche. Pour ce faire, vous définissez les actions qui peuvent être entreprises sur des ressources spécifiques dans des conditions spécifiques, également appelées autorisations de moindre privilège. Pour plus d'informations sur l'utilisation d'IAM pour appliquer des autorisations, consultez politiques et autorisations dans IAM dans le Guide de l'utilisateur IAM.
- Utilisez des conditions dans les politiques IAM pour restreindre davantage l'accès : vous pouvez ajouter une condition à vos politiques afin de limiter l'accès aux actions et aux ressources. Par exemple, vous pouvez écrire une condition de politique pour spécifier que toutes les demandes doivent être envoyées via SSL. Vous pouvez également utiliser des conditions pour accorder l'accès aux actions de service si elles sont utilisées par le biais d'un service spécifique Service AWS, tel que CloudFormation. Pour plus d'informations, consultez Conditions pour éléments de politique JSON IAM dans le Guide de l'utilisateur IAM.

- Utilisez l'Analyseur d'accès IAM pour valider vos politiques IAM afin de garantir des autorisations sécurisées et fonctionnelles : l'Analyseur d'accès IAM valide les politiques nouvelles et existantes de manière à ce que les politiques IAM respectent le langage de politique IAM (JSON) et les bonnes pratiques IAM. IAM Access Analyzer fournit plus de 100 vérifications de politiques et des recommandations exploitables pour vous aider à créer des politiques sécurisées et fonctionnelles. Pour plus d'informations, consultez Validation de politiques avec IAM Access Analyzer dans le Guide de l'utilisateur IAM.
- Exiger l'authentification multifactorielle (MFA) : si vous avez un scénario qui nécessite des utilisateurs IAM ou un utilisateur root, activez l'authentification MFA pour une sécurité accrue. Compte AWS Pour exiger la MFA lorsque des opérations d'API sont appelées, ajoutez des conditions MFA à vos politiques. Pour plus d'informations, consultez Sécurisation de l'accès aux API avec MFA dans le Guide de l'utilisateur IAM.

Pour plus d'informations sur les bonnes pratiques dans IAM, consultez Bonnes pratiques de sécurité dans IAM dans le Guide de l'utilisateur IAM.

## Utilisation de la console HealthOmics

Pour accéder à la HealthOmics console AWS, vous devez disposer d'un ensemble minimal d'autorisations. Ces autorisations doivent vous permettre de répertorier et d'afficher des informations détaillées sur les HealthOmics ressources AWS de votre Compte AWS. Si vous créez une politique basée sur l'identité qui est plus restrictive que l'ensemble minimum d'autorisations requis, la console ne fonctionnera pas comme prévu pour les entités (utilisateurs ou rôles) tributaires de cette politique.

Il n'est pas nécessaire d'accorder des autorisations de console minimales aux utilisateurs qui appellent uniquement l'API AWS CLI ou l' AWS API. Autorisez plutôt l'accès à uniquement aux actions qui correspondent à l'opération d'API qu'ils tentent d'effectuer.

## Autorisation accordée aux utilisateurs pour afficher leurs propres autorisations

Cet exemple montre comment créer une politique qui permet aux utilisateurs IAM d'afficher les politiques en ligne et gérées attachées à leur identité d'utilisateur. Cette politique inclut les autorisations permettant d'effectuer cette action sur la console ou par programmation à l'aide de l'API AWS CLI or 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 politiques gérées pour AWS HealthOmics

Une politique AWS gérée est une politique autonome créée et administrée par AWS. AWS les politiques gérées sont conçues pour fournir des autorisations pour de nombreux cas d'utilisation courants afin que vous puissiez commencer à attribuer des autorisations aux utilisateurs, aux groupes et aux rôles.

N'oubliez pas que les politiques AWS gérées peuvent ne pas accorder d'autorisations de moindre privilège pour vos cas d'utilisation spécifiques, car elles sont accessibles à tous les AWS clients.

Nous vous recommandons de réduire encore les autorisations en définissant des politiques gérées par le client qui sont propres à vos cas d'utilisation.

Vous ne pouvez pas modifier les autorisations définies dans les politiques AWS gérées. Si les autorisations définies dans une politique AWS gérée sont AWS mises à jour, la mise à jour affecte toutes les identités principales (utilisateurs, groupes et rôles) auxquelles la politique est attachée. AWS est le plus susceptible de mettre à jour une politique AWS gérée lorsqu'une nouvelle politique Service AWS est lancée ou lorsque de nouvelles opérations d'API sont disponibles pour les services existants.

Pour plus d'informations, consultez Politiques gérées par AWS dans le Guide de l'utilisateur IAM.

## AWS politique gérée : AmazonOmicsFullAccess

Vous pouvez associer la `AmazonOmicsFullAccess` politique à vos identités IAM pour leur donner un accès complet à HealthOmics.

Cette politique accorde des autorisations d'accès complètes à toutes les HealthOmics actions. Lorsque vous créez un magasin d'annotations ou de variantes, Omics vous donne également accès à ce magasin par le biais d'une invitation au partage de ressources dans la console Resource Access Manager (RAM). Pour plus d'informations sur les invitations au partage de ressources via Lake Formation, consultez le partage de données entre comptes dans Lake Formation. Dans le cadre d'une politique d'administration Omics, vous avez également besoin des autorisations suivantes pour accéder à votre compartiment Amazon S3.

- PutObject
- GetObject
- ListBucket
- AbortMultipartUpload
- ListMultipartUploadParts

JSON

```
{
 "Version":"2012-10-17",
 "Statement": [
  {
   "Effect": "Allow",
   "Action": [
    "omics:*"
   ],
   "Resource": "*"
  },
  {
   "Effect": "Allow",
   "Action": [
    "ram:AcceptResourceShareInvitation",
    "ram:GetResourceShareInvitations"
   ],
   "Resource": "*",
   "Condition": {
    "StringEquals": {
     "aws:CalledViaLast": "omics.amazonaws.com"
    }
   }
  },
  {
   "Effect": "Allow",
   "Action": "iam:PassRole",
   "Resource": "*",
   "Condition": {
    "StringEquals": {
     "iam:PassedToService": "omics.amazonaws.com"
    }
   }
  }
 ]
}
```

## AWS politique gérée : AmazonOmicsReadOnlyAccess

Vous pouvez associer la `AWSOmicsReadOnlyAccess` politique à vos identités IAM lorsque vous souhaitez limiter les autorisations associées à cette identité à un accès en lecture seule.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "omics:Get*",
                "omics:List*"
            ],
            "Resource": "*"
        }
    ]
}
```

## HealthOmics mises à jour des politiques AWS gérées

Consultez les détails des mises à jour des politiques AWS gérées HealthOmics depuis que ce service a commencé à suivre ces modifications. Pour recevoir des alertes automatiques concernant les modifications apportées à cette page, abonnez-vous au flux RSS sur la page Historique du HealthOmics document.

| Modifier | Description | Date |
| --- | --- | --- |
| AmazonOmicsFullAccess - Nouvelle politique ajoutée | HealthOmics a ajouté une nouvelle politique pour accorder à un utilisateur un accès complet à toutes les actions et ressources. Pour en savoir plus, veuillez consulter la section AmazonOmicsFullAccess. | 23 février 2023 |

| Modifier | Description | Date |
| --- | --- | --- |
| HealthOmics a commencé à suivre les modifications | HealthOmics a commencé à suivre les modifications apportées AWS à ses politiques gérées. | 29 novembre 2022 |
| AmazonOmicsReadOnlyAccess - Nouvelle politique ajoutée | HealthOmics a ajouté une nouvelle politique qui limite l'accès à la lecture seule. Pour en savoir plus, consultez AmazonOmicsReadOnlyAccess. | 29 novembre 2022 |

## Résolution des problèmes AWS HealthOmics d'identité et d'accès

Utilisez les informations suivantes pour vous aider à diagnostiquer et à résoudre les problèmes courants que vous pouvez rencontrer lorsque vous travaillez avec AWS HealthOmics et IAM.

Rubriques

- Je ne suis pas autorisé à effectuer une action dans HealthOmics
- Je ne suis pas autorisé à effectuer iam : PassRole
- Je souhaite permettre à des personnes extérieures Compte AWS à moi d'accéder à mes HealthOmics ressources

### Je ne suis pas autorisé à effectuer une action dans HealthOmics

Si vous recevez une erreur qui indique que vous n'êtes pas autorisé à effectuer une action, vos politiques doivent être mises à jour afin de vous permettre d'effectuer l'action.

L'exemple d'erreur suivant se produit quand l'utilisateur IAM `mateojackson` tente d'utiliser la console pour afficher des informations détaillées sur une ressource *`my-example-widget`* fictive, mais ne dispose pas des autorisations `omics:`*`GetWidget`* fictives.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 omics:GetWidget on resource: my-example-widget
```

Dans ce cas, la politique qui s'applique à l'utilisateur `mateojackson` doit être mise à jour pour autoriser l'accès à la ressource *`my-example-widget`* à l'aide de l'action `omics:`*`GetWidget`*.

Si vous avez besoin d'aide, contactez votre AWS administrateur. Votre administrateur vous a fourni vos informations d'identification de connexion.

## Je ne suis pas autorisé à effectuer iam : PassRole

Si vous recevez un message d'erreur indiquant que vous n'êtes pas autorisé à effectuer l'`iam:PassRole`action, vos politiques doivent être mises à jour pour vous permettre de transmettre un rôle à AWS HealthOmics.

Certains vous Services AWS permettent de transmettre un rôle existant à ce service au lieu de créer un nouveau rôle de service ou un rôle lié à un service. Pour ce faire, vous devez disposer des autorisations nécessaires pour transmettre le rôle au service.

L'exemple d'erreur suivant se produit lorsqu'un utilisateur IAM nommé `marymajor` essaie d'utiliser la console pour effectuer une action dans AWS HealthOmics. Toutefois, l'action nécessite que le service ait des autorisations accordées par un rôle de service. Mary n'est pas autorisée à transmettre le rôle au service.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

Dans ce cas, les politiques de Mary doivent être mises à jour pour lui permettre d'exécuter l'action `iam:PassRole`.

Si vous avez besoin d'aide, contactez votre AWS administrateur. Votre administrateur vous a fourni vos informations d'identification de connexion.

## Je souhaite permettre à des personnes extérieures Compte AWS à moi d'accéder à mes HealthOmics ressources

Vous pouvez créer un rôle que les utilisateurs provenant d'autres comptes ou les personnes extérieures à votre organisation pourront utiliser pour accéder à vos ressources. Vous pouvez spécifier qui est autorisé à assumer le rôle. Pour les services qui prennent en charge les politiques basées sur les ressources ou les listes de contrôle d'accès (ACLs), vous pouvez utiliser ces politiques pour autoriser les utilisateurs à accéder à vos ressources.

Pour plus d'informations, consultez les éléments suivants :

- Pour savoir si AWS HealthOmics prend en charge ces fonctionnalités, consultezComment AWS HealthOmics fonctionne avec IAM.
- Pour savoir comment fournir l'accès à vos ressources sur celles Comptes AWS que vous possédez, consultez la section Fournir l'accès à un utilisateur IAM dans un autre utilisateur Compte AWS que vous possédez dans le Guide de l'utilisateur IAM.
- Pour savoir comment fournir l'accès à vos ressources à des tiers Comptes AWS, consultez la section Fournir un accès à des ressources Comptes AWS détenues par des tiers dans le guide de l'utilisateur IAM.
- Pour savoir comment fournir un accès par le biais de la fédération d'identité, consultez Fournir un accès à des utilisateurs authentifiés en externe (fédération d'identité) dans le Guide de l'utilisateur IAM.
- Pour en savoir plus sur la différence entre l'utilisation des rôles et des politiques basées sur les ressources pour l'accès intercompte, consultez Accès intercompte aux ressources dans IAM dans le Guide de l'utilisateur IAM.

## Validation de conformité pour AWS HealthOmics

Des auditeurs tiers évaluent la sécurité et AWS HealthOmics la conformité de plusieurs programmes de AWS conformité. Cela inclut HIPAA, FedRAMP et d'autres. Le tableau suivant indique les certifications de conformité du HealthOmics service.

| Certification | Lien |
|---|---|
| HIPAA | Référence des services éligibles à la loi HIPAA |
| HiTrust-LCR | Cadre de sécurité commun de la Health Information Trust Alliance |
| FedRAMP Modérée (Est/Ouest) | Programme fédéral de gestion des risques et des autorisations |
| ÉTOILE ISO/CSA | Certifié ISO et CSA STAR |
| C5 | Catalogue des contrôles de conformité du cloud computing |

| Certification | Lien |
| --- | --- |
| DoD CC SRG IL2 | Guide des exigences de sécurité en matière de cloud computing du ministère de la Défense |
| ENS High | Schéma national de sécurité |
| FINMA | Autorité suisse de surveillance des marchés financiers |
| ISMAP | Programme de gestion et d'évaluation de la sécurité des systèmes d'information |
| OSPAR | Rapport d'audit du fournisseur de services externalisé |
| PCI | Norme de sécurité des données du secteur des cartes de paiement |
| Pinakes | Association bancaire CCI - Qualification par un tiers |
| PiTuKri | Critères d'évaluation de la sécurité des informations des services cloud |
| SOC 1,2,3 | Contrôles du système et de l'organisation |

Pour une liste de tous les AWS services concernés par des programmes de conformité spécifiques, voir Services AWS concernés par programme de conformité . Pour obtenir des renseignements généraux, consultez Programmes de conformitéAWS .

Vous pouvez télécharger des rapports d'audit tiers à l'aide de AWS Artifact. Pour plus d'informations, voir Téléchargement de rapports dans AWS Artifact .

HealthOmics les magasins de données utilisent l'ID d'exemple pour nommer les fichiers internes et pour baliser les ressources. Avant d'ingérer des données, vérifiez si l'identifiant d'échantillon contient des données PHI. Si c'est le cas, modifiez l'identifiant de l'échantillon avant d'ingérer les données. Pour plus d'informations, consultez les instructions sur la page Web de conformité à la loi AWS HIPAA.

Votre responsabilité en matière de conformité lors de l'utilisation AWS HealthOmics est déterminée par la sensibilité de vos données, les objectifs de conformité de votre entreprise et les lois et réglementations applicables. AWS fournit les ressources suivantes pour faciliter la mise en conformité :

- Guides démarrage rapide de la sécurité et de la conformité. Ces guides de déploiement traitent des considérations architecturales et fournissent des étapes pour déployer des environnements de base axés sur la sécurité et la conformité sur AWS.
- Livre blanc sur l'architecture pour la sécurité et la conformité HIPAA — Ce livre blanc décrit comment les entreprises peuvent créer des applications conformes à la loi HIPAA. AWS
- AWS Ressources de https://aws.amazon.com/compliance/resources/ de conformité — Cette collection de classeurs et de guides peut s'appliquer à votre secteur d'activité et à votre région.
- Évaluation des ressources à l'aide des règles énoncées dans le guide du AWS Config développeur : AWS Configévalue dans quelle mesure les configurations de vos ressources sont conformes aux pratiques internes, aux directives du secteur et aux réglementations.
- AWS Security Hub CSPM— Ce AWS service fournit une vue complète de l'état de votre sécurité interne, AWS ce qui vous permet de vérifier votre conformité aux normes et aux meilleures pratiques du secteur de la sécurité.

# Résilience dans HealthOmics

L'infrastructure AWS mondiale est construite autour Régions AWS de zones de disponibilité. Régions AWS fournissent plusieurs zones de disponibilité physiquement séparées et isolées, connectées par un réseau à faible latence, à haut débit et hautement redondant. Avec les zones de disponibilité, vous pouvez concevoir et exploiter des applications et des bases de données qui basculent automatiquement d'une zone à l'autre sans interruption. Les zones de disponibilité sont davantage disponibles, tolérantes aux pannes et ont une plus grande capacité de mise à l'échelle que les infrastructures traditionnelles à un ou plusieurs centres de données.

Pour plus d'informations sur les zones de disponibilité Régions AWS et les zones de disponibilité, consultez la section Infrastructure AWS globale.

Outre l'infrastructure AWS mondiale, AWS HealthOmics propose plusieurs fonctionnalités pour répondre à vos besoins en matière de résilience et de sauvegarde des données.

# AWS HealthOmics et points de terminaison VPC d'interface ()AWS PrivateLink

Vous pouvez établir une connexion privée entre votre VPC et créer un point de AWS HealthOmics terminaison VPC d'interface. Les points de terminaison de l'interface sont alimentés par AWS PrivateLinkune technologie que vous pouvez utiliser pour accéder de manière privée aux opérations d' HealthOmics API sans passerelle Internet, appareil NAT, connexion VPN ou connexion AWS Direct Connect. Les instances de votre VPC n'ont pas besoin d'adresses IP publiques pour communiquer avec les opérations d' HealthOmics API. Le trafic entre votre VPC et celui qui HealthOmics ne sort pas du réseau Amazon.

Chaque point de terminaison d'interface est représenté par une ou plusieurs interfaces réseau Elastic dans vos sous-réseaux.

Pour plus d'informations, consultez la section Interface VPC endpoints (AWS PrivateLink) dans le guide de l'utilisateur Amazon VPC.

Les politiques relatives aux points de terminaison VPC sont prises en charge dans toutes HealthOmics les régions, à l'exception d'Israël (Tel Aviv). Par défaut, l'accès complet à HealthOmics est autorisé via le point de terminaison.

## Considérations relatives aux points de HealthOmics terminaison VPC

Avant de configurer un point de terminaison VPC d'interface pour HealthOmics, assurez-vous de consulter les propriétés et les limites du point de terminaison d'interface dans le guide de l'utilisateur Amazon VPC.

HealthOmics permet d'appeler toutes les actions HealthOmics de l'API de stockage depuis votre VPC.

Les politiques de point de terminaison VPC ne sont pas prises en charge HealthOmics par défaut, mais vous pouvez créer un point de terminaison VPC pour un HealthOmics accès complet aux opérations de stockage. HealthOmics Pour plus d'informations, consultez Contrôle de l'accès aux services avec points de terminaison d'un VPC dans le Guide de l'utilisateur Amazon VPC.

## Création d'un point de terminaison de VPC d'interface pour HealthOmics

Vous pouvez créer un point de terminaison VPC pour le HealthOmics service à l'aide de la console Amazon VPC ou du (). AWS Command Line Interface AWS CLI Pour plus d'informations, consultez Création d'un point de terminaison d'interface dans le Guide de l'utilisateur Amazon VPC.

Créez un point de terminaison VPC pour HealthOmics en utilisant les noms de service suivants :

- com.amazonaws. *region*.storage-omics
- com.amazonaws. *region*. control-storage-omics
- com.amazonaws. *region*.analytics-omics
- com.amazonaws. *region*.workflows-omics
- com.amazonaws. *region*.tags-omics

Les régions USA Est (Virginie du Nord) et USA Ouest (Oregon) prennent en charge les points de terminaison AWS PrivateLink FIPS. Pour ces régions, vous pouvez également utiliser les noms de service suivants :

- com.amazonaws. *region*. storage-omics-fips
- com.amazonaws. *region*. control-storage-omics-fips
- com.amazonaws. *region*. analytics-omics-fips
- com.amazonaws. *region*. workflows-omics-fips
- com.amazonaws. *region*. tags-omics-fips

Si vous activez le DNS privé pour le point de terminaison, vous pouvez envoyer des demandes d' HealthOmics API en utilisant son nom DNS par défaut pour la région, par exemple,`omics.us-east-1.amazonaws.com`.

Pour plus d'informations, consultez Accès à un service via un point de terminaison d'interface dans le Guide de l'utilisateur Amazon VPC.

## Création d'une stratégie de point de terminaison d'un VPC pour HealthOmics

Vous pouvez attacher une stratégie de point de terminaison à votre point de terminaison d'un VPC qui contrôle l'accès à HealthOmics. La politique spécifie les informations suivantes :

- Le principal qui peut exécuter des actions.
- Les actions qui peuvent être effectuées.
- Les ressources sur lesquelles les actions peuvent être exécutées.

Pour plus d'informations, consultez Contrôle de l'accès aux services avec points de terminaison d'un VPC dans le Guide de l'utilisateur Amazon VPC.

Exemple : politique de point de terminaison VPC pour les HealthOmics actions.

Voici un exemple de politique de point de terminaison pour HealthOmics. Lorsqu'elle est attachée à un point de terminaison, cette politique accorde l'accès aux HealthOmics actions pour tous les principaux sur toutes les ressources.

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\":\"*\"}]}"
```

# Considérations spéciales relatives à l'accès aux ensembles de lecture à l'aide d'Amazon S3 URIs

Pour accéder aux ensembles de lecture via Amazon S3 URIs lorsque vous utilisez une connexion privée, configurez les points de terminaison de l' PrivateLinkinterface sur le magasin de séquences. Une fois que vous les avez configurés, les points de terminaison ont les formats suivants :

```
com.amazonaws.region.storage-omics
com.amazonaws.region.control-storage-omics
```

Pour utiliser les points de terminaison de passerelle, suivez le guide Points de terminaison de passerelle pour Amazon S3 afin de configurer vos points de terminaison de passerelle. HealthOmics possède le compartiment Amazon S3, vous n'avez donc pas à créer ou à ajuster la politique du compartiment. Les points de terminaison de la passerelle s'appuient sur la politique attachée à l'utilisateur ou au rôle qui accède aux données, mais vous pouvez également configurer les points de terminaison avec des politiques plus restrictives. Ces politiques peuvent inclure des restrictions d'accès basées sur l'ARN du point d'accès Amazon S3 et les actions Amazon S3.

# Surveillance d'AWS HealthOmics

La surveillance joue un rôle important dans le maintien de la fiabilité, de la disponibilité et des performances d'AWS HealthOmics et de vos autres AWS solutions. AWS fournit les outils de surveillance suivants pour surveiller AWS HealthOmics, signaler un problème et prendre des mesures automatiques le cas échéant :

- Amazon CloudWatch surveille vos AWS ressources et les applications que vous utilisez AWS en temps réel. Vous pouvez collecter et suivre les métriques, créer des tableaux de bord personnalisés, et définir des alarmes qui vous informent ou prennent des mesures lorsqu'une métrique spécifique atteint un seuil que vous spécifiez. Par exemple, vous pouvez CloudWatch suivre l'utilisation du processeur ou d'autres indicateurs de vos EC2 instances Amazon et lancer automatiquement de nouvelles instances en cas de besoin. Pour plus d'informations, consultez le guide de CloudWatch l'utilisateur Amazon.
- Amazon CloudWatch Logs vous permet de surveiller, de stocker et d'accéder à vos fichiers journaux à partir d' EC2 instances Amazon et d'autres sources. CloudTrail CloudWatch Les journaux peuvent surveiller les informations contenues dans les fichiers journaux et vous avertir lorsque certains seuils sont atteints. Vous pouvez également archiver vos données de journaux dans une solution de stockage hautement durable. Pour plus d'informations, consultez le guide de l'utilisateur d'Amazon CloudWatch Logs.
- AWS CloudTrail capture les appels d'API et les événements associés créés par votre Compte AWS ou au nom de celui-ci et livre les fichiers journaux dans un compartiment Amazon S3 que vous spécifiez. Vous pouvez identifier les utilisateurs et les comptes qui ont appelé AWS, l'adresse IP source à partir de laquelle les appels ont été émis, ainsi que le moment où les appels ont eu lieu. Pour de plus amples informations, veuillez consulter le Guide de l'utilisateur AWS CloudTrail.
- Amazon EventBridge est un service de bus d'événements sans serveur qui permet de connecter facilement vos applications à des données provenant de diverses sources. EventBridge fournit un flux de données en temps réel à partir de vos propres applications, applications Software-as-a-Service (SaaS) et AWS services et achemine ces données vers des cibles telles que Lambda. Cela vous permet de surveiller les événements qui se produisent dans les services et de créer des architectures basées sur les événements. Pour plus d'informations, consultez le guide de EventBridge l'utilisateur Amazon.

> **Note**
>
> Pour les mises à jour des services, configurez et surveillez votre Personal Health Dashboard. Pour plus d'informations sur la gestion du tableau de bord, consultez Getting started with your AWS Health Dashboard.

Rubriques



# Journalisation des accès S3

Vous pouvez surveiller l'accès à l'API Amazon S3 pour HealthOmics séquencer les données du magasin à l'aide des journaux d'accès créés par le magasin. Vous pouvez l'utiliser CloudWatch pour surveiller l'accès à S3 à partir des opérations HealthOmics d'API. CloudWatch fournit une visibilité sur l'accès à Amazon S3 à partir de votre propre compte. Si, en tant que propriétaire des données, vous partagez l'accès à un compte tiers, la journalisation des accès n'est pas disponible dans CloudWatch. Utilisez plutôt le journal d'accès S3 du magasin, qui enregistre tous les accès S3 aux données du compartiment Amazon S3 configuré.

Configurez les journaux d'accès S3 à l'aide des opérations de `UpdateSequenceStore` l'API `CreateSequenceStore` or. Assurez-vous également que le principal de HealthOmics service (`omics.amazonaws.com`) dispose `s3:PutObject` des autorisations d'accès au préfixe S3 configuré.

> **Note**
>
> Les journaux utilisent la configuration de chiffrement par défaut du compartiment de destination. Si le bucket utilise une clé gérée par le client, le principal du service doit être autorisé à utiliser la clé pour écrire.

Pour désactiver la journalisation des accès, utilisez `UpdateSequenceStore` et définissez la configuration du journal d'accès sur vide.

# Surveillance à HealthOmics l'aide de CloudWatch métriques

Vous pouvez surveiller HealthOmics l'utilisation CloudWatch, qui collecte les données brutes et les transforme en indicateurs lisibles en temps quasi réel. Ces statistiques sont enregistrées pour une durée de 15 mois ; par conséquent, vous pouvez accéder aux informations historiques et acquérir un meilleur point de vue de la façon dont votre service ou application web s'exécute. Vous pouvez également définir des alarmes qui surveillent certains seuils et envoient des notifications ou prennent des mesures lorsque ces seuils sont atteints. Pour plus d'informations, consultez le guide de CloudWatch l'utilisateur Amazon.

Le AWS HealthOmics service indique les métriques suivantes dans l'espace de `AWS/Omics` noms.

Les statistiques du nombre d'appels d'API sont signalées pour les éléments suivants AWS HealthOmics APIs. Seule la dimension d'opération d'API est signalée.

- Magasin de référence et de référence APIs —CreateReferenceStore, DeleteReferenceStore, StartReferenceImportJob
- Stockage de séquences et ensemble de lecture APIs — CreateSequenceStore DeleteSequenceStore,, StartReadSetImportJob, StartReadSetActivationJob, StartReadSetExportJob
- Variante de magasin APIs — CreateVariantStore DeleteVariantStore, StartVariantImportJob, CancelVariantImportJob
- Magasin d'annotations APIs — CreateAnnotationStore, DeleteAnotationStore, StartAnnotationImportJob, CancelAnnotationImportJob
- Flux de travail, exécution et groupe d'exécution APIs — CreateWorkflow DeleteWorkflow, StartRun, CancelRun, DeleteRun, CreateRunGroup, DeleteRunGroup

## Affichage des métriques *AWS HealthOmics*

CloudWatch les métriques pour AWS HealthOmics sont consultables dans la CloudWatch console.

Pour consulter les métriques (CloudWatch console)

1. Connectez-vous à AWS Management Console et ouvrez la console CloudWatch .

2. Choisissez Metrics, choisissez All Metrics, puis choisissez AWS/Usage.
3. Service de filtrage pour AWS HealthOmics.
4. Choisissez la dimension, le nom de la métrique, puis Ajouter au graphique.
5. Choisissez une valeur pour la plage de dates. Le décompte de la métrique pour la plage de dates sélectionnée est affiché dans le graphique.

## Création d'une alarme à l'aide de CloudWatch

Une CloudWatch alarme surveille une seule métrique sur une période spécifiée et exécute une ou plusieurs actions : envoyer une notification à une rubrique Amazon Simple Notification Service (Amazon SNS) ou à une politique Auto Scaling. L'action ou les actions sont basées sur la valeur de la métrique par rapport à un seuil donné sur un certain nombre de périodes que vous spécifiez. CloudWatch peut également vous envoyer un message Amazon SNS lorsque l'alarme change d'état.

CloudWatch les alarmes appellent des actions uniquement lorsque l'état change et persiste pendant la période que vous spécifiez.

Pour consulter les métriques (CloudWatch console)

1. Connectez-vous à AWS Management Console et ouvrez la console CloudWatch .
2. Choisissez Alarmes, puis Créer une alarme.
3. Choisissez AWS/Usage, puis choisissez une AWS HealthOmics métrique à l'aide de la dimension Service.
4. Pour Période, choisissez un intervalle de temps à surveiller, puis cliquez sur Suivant.
5. Saisissez un Name (Nom) et une Description.
6. Pour Whenever, choisissez **>=**, puis saisissez une valeur maximale.
7. Si vous souhaitez CloudWatch envoyer un e-mail lorsque l'état d'alarme est atteint, dans la section Actions, pour Chaque fois que cette alarme est atteinte, choisissez State is ALARM. Pour Envoyer une notification à, choisissez une liste de diffusion ou choisissez Nouvelle liste et créez une nouvelle liste de diffusion.
8. Affichez un aperçu de l'alarme dans la section Aperçu de l'alarme. Si elle vous convient, choisissez Créer une alarme.

# Surveillance à HealthOmics l'aide de CloudWatch journaux

HealthOmics génère divers journaux pour vous aider à comprendre et à résoudre les problèmes liés à vos courses. Les journaux sont disponibles à deux endroits : CloudWatch et sur Amazon S3.

Par défaut, la journalisation des courses est activée. Vous pouvez éventuellement désactiver la journalisation d'une exécution `LogLevel = OFF` en définissant la startrun demande.

> **Note**
>
> Pour les mises à jour des services, configurez et surveillez votre Personal Health Dashboard. Pour plus d'informations sur la gestion du tableau de bord, consultez Getting started with your AWS Health Dashboard.

Rubriques



## Types de journaux pour les HealthOmics flux de travail

HealthOmics fournit les types de journaux suivants pour les flux de travail :

- Journaux du moteur : les moteurs de flux de travail sous-jacents (Nextflow, WDL et CWL) produisent des journaux du moteur pour les exécutions. Ces journaux peuvent vous aider à résoudre les problèmes de définition des flux de travail.
- Journaux du manifeste d'exécution : ces journaux fournissent des informations de haut niveau sur chaque tâche exécutée, telles que l'état de la tâche, l'heure de début, l'heure de fin et le motif de l'échec (en cas d'échec de la tâche).

  Les journaux des manifestes d'exécution fournissent également des statistiques d'utilisation des ressources qui peuvent être utiles pour comprendre les opportunités d'optimisation des ressources. Ces statistiques incluent :

  - Moyenne du processeur

  - Maximum du processeur
  - CPU réservés
  - GPU réservés
  - memoryAverageGi B
  - memoryMaximumGi B
  - memoryReservedGi B
  - Secondes de fonctionnement
- Journaux d'exécution : les journaux d'exécution fournissent l'état d'exécution global et l'heure à laquelle les tâches individuelles démarrent, s'exécutent, s'arrêtent et se terminent. Les journaux d'exécution vous donnent également une visibilité sur les étapes d'importation et d'exportation de fichiers.
- Journaux de tâches : les journaux de tâches fournissent des informations de journalisation détaillées sur les tâches individuelles de votre exécution. Les résultats de votre journal des tâches dépendent de la définition de la tâche et de l'endroit où vous utilisez les instructions de journal dans votre code. Si vos journaux de tâches ne fournissent pas le niveau d'information dont vous avez besoin, pensez à ajouter des instructions de journal supplémentaires à la définition de vos tâches afin de produire des journaux de tâches plus pertinents.
- Exécuter les journaux du cache : les journaux du cache d'exécution fournissent l'état général des caches d'exécution et de la mise en cache des résultats des tâches. Les journaux d'exécution du cache vous donnent une visibilité sur les accès et les échecs du cache pour chaque exécution utilisant la mise en cache.
- outputs.json — Pour les flux de travail WDL et CWL, HealthOmics fournit un fichier généré par le moteur, nommé, à `outputs.json` votre compartiment Amazon S3 une fois l'exécution terminée. Ce fichier inclut une liste et une carte de toutes les sorties pour l'exécution.

## Se connecte CloudWatch

CloudWatch génère des journaux de flux de travail pour les échecs et les exécutions réussies. Tous les journaux sont disponibles pour les essais ayant échoué et les essais réussis, à l'exception des journaux du moteur qui ne sont disponibles que pour les essais ayant échoué.

Vous pouvez trouver les journaux du CloudWatch flux de travail dans le groupe de journaux suivant :`/aws/omics/WorkflowLog`. En outre, le résultat de l'opération d'API get-run fournit le flux de CloudWatch journal ARNs pour les journaux du moteur et les journaux d'exécution.

Par défaut, AWS conserve les CloudWatch journaux indéfiniment. Vous pouvez ajuster la politique de conservation du groupe de journaux afin de définir une période de conservation comprise entre 10 ans et un jour.

Le tableau suivant fournit un résumé CloudWatch des connexions HealthOmics. Tous les journaux de flux de travail sont disponibles pour les exécutions réussies et les échecs, à l'exception des journaux du moteur qui ne sont disponibles que pour les exécutions ayant échoué.

| Nom du journal | Disponible dans CloudWatch Logs | Quand le journal est-il disponible | Format du flux de journal |
|---|---|---|---|
| Journaux du moteur | Oui, en cas d'échec des courses | Une fois l'exécution terminée | exécuter/ /engine *runID* |
| Exécuter les journaux du manifeste | Oui | Une fois l'exécution terminée | manifest/exécuter/ /*runIDrunUUID* |
| Lancer les journaux | Oui | En temps réel | courrir/ *runID* |
| Journaux de tâches | Oui | En temps réel | exécuter/ /tâche/ *runID taskID* |
| Exécuter les journaux du cache | Oui | En temps réel | Exécuter le cache//*runCacheId runCacheUUID* |
| Outputs.json (WDL et CWL) | Non | N/A | s/o |

## Se connecte à Amazon S3

Seuls les journaux du moteur et le `outputs.json` fichier sont transmis à Amazon S3.

Une fois l'exécution terminée, les journaux du moteur sont envoyés dans votre compartiment S3 et sont disponibles indéfiniment jusqu'à ce que vous les supprimiez. Ces journaux se trouvent dans le répertoire des journaux de l'URI de sortie S3 que vous avez spécifié pour le flux de travail.

Le chemin d'accès au répertoire des journaux est au format suivant :`s3://{user_provided_path}/logs/`.

Le tableau suivant fournit un résumé des HealthOmics journaux disponibles dans votre compartiment Amazon S3.

| Nom du journal | Disponible sur Amazon S3 | Quand le journal est-il disponible | Chemin du flux de log |
|---|---|---|---|
| Journaux du moteur | Oui | Une fois l'exécution terminée | s3 ://*user_provided_path* /logs/engine.log |
| Outputs.json (WDL et CWL) | Oui | Une fois l'exécution terminée | s3 ://*user_provided_path* /*runID*/*runUUID*/logs/outputs.json |
| Exécuter des journaux de manifeste, des journaux d'exécution et des journaux de tâches | Non | N/A | s/o |

## CloudWatch Journaux interactifs dans la CLI

Vous pouvez consulter les CloudWatch journaux de manière interactive à l'aide de la commande Live Tail en mode interactif. Vous pouvez suivre la progression des courses en temps réel et définir jusqu'à 5 mots clés à mettre en évidence dans les journaux :

```
aws logs start-live-tail  \
  --mode interactive  \
  --log-group-identifiers arn:aws:logs:region:account-ID:log-group:/aws/omics/
WorkflowLog
```

Pour plus d'informations, voir Start Live Tail dans le manuel de référence des AWS CLI commandes.

## Accès aux CloudWatch journaux depuis la console

Pour accéder aux journaux d'une exécution, vous pouvez accéder directement à ces journaux depuis la page des détails de l'exécution de la HealthOmics console.

1. Ouvrez la HealthOmics console.
2. Si nécessaire, ouvrez le volet de navigation de gauche (≡). Choisissez Runs.
3. Sélectionnez l'exécution dans le tableau Exécutions.
4. Sur la page des détails de l'exécution, vous pouvez choisir l'une des actions suivantes :

    a. Dans Récapitulatif des exécutions, choisissez Afficher les journaux d'exécution. La console ouvre les journaux d'exécution dans la CloudWatch console.
    b. Dans Exécuter le résumé, choisissez Afficher les journaux dans Amazon S3. La console ouvre le dossier des journaux dans la console Amazon S3.
    c. Dans Exécuter les tâches, choisissez Afficher les journaux, Afficher les journaux d'exécution ou Afficher les journaux du manifeste d'exécution d'une tâche. La console ouvre les journaux dans la CloudWatch console.

Vous pouvez également accéder aux journaux depuis la CloudWatch console :

1. Ouvrez la CloudWatch console https://console.aws.amazon.com/cloudwatch/.
2. Dans le menu de gauche, choisissez Log groups.
3. Sélectionnez le groupe `/aws/omics/WorkflowLog`.

    Si la liste des groupes de journaux est longue, vous pouvez saisir des omiques dans la zone de texte de recherche pour affiner la liste.
4. Lorsque la page des détails du groupe de journaux s'ouvre, choisissez le flux de journaux que vous souhaitez consulter. La console affiche les événements de ce flux de journal.

# Journalisation des appels AWS HealthOmics d'API à l'aide AWS CloudTrail

AWS HealthOmics est intégré à AWS CloudTrail un service qui fournit un enregistrement des actions entreprises par un utilisateur, un rôle ou un AWS service dans HealthOmics. CloudTrail capture tous les appels d'API HealthOmics sous forme d'événements. Les appels capturés incluent des appels provenant de la HealthOmics console et des appels de code vers les opérations de l' HealthOmics API. Si vous créez un suivi, vous pouvez activer la diffusion continue d' CloudTrail événements vers un compartiment Amazon S3, y compris les événements pour HealthOmics. Si vous ne configurez pas de suivi, vous pouvez toujours consulter les événements les plus récents dans la CloudTrail console dans Historique des événements. À l'aide des informations collectées par CloudTrail, vous

pouvez déterminer la demande qui a été faite HealthOmics, l'adresse IP à partir de laquelle la demande a été faite, qui a fait la demande, quand elle a été faite et des détails supplémentaires.

Pour en savoir plus CloudTrail, consultez le guide de AWS CloudTrail l'utilisateur.

## HealthOmics informations dans CloudTrail

CloudTrail est activé sur votre compte Compte AWS lorsque vous créez le compte. Lorsqu'une activité se produit dans HealthOmics, cette activité est enregistrée dans un CloudTrail événement avec d'autres événements de AWS service dans l'historique des événements. Vous pouvez consulter, rechercher et télécharger les événements récents dans votre Compte AWS. Pour plus d'informations, consultez la section Affichage des événements à l'aide de l'historique des CloudTrail événements.

Pour un enregistrement continu des événements de votre région Compte AWS, y compris des événements pour HealthOmics, créez un parcours. Un suivi permet CloudTrail de fournir des fichiers journaux à un compartiment Amazon S3. Par défaut, lorsque vous créez un journal d'activité dans la console, il s'applique à toutes les régions Régions AWS. Le journal enregistre les événements de toutes les régions de la AWS partition et transmet les fichiers journaux au compartiment Amazon S3 que vous spécifiez. En outre, vous pouvez configurer d'autres AWS services pour analyser plus en détail les données d'événements collectées dans les CloudTrail journaux et agir en conséquence. Pour plus d'informations, consultez les ressources suivantes :

- Présentation de la création d'un journal de suivi
- CloudTrail services et intégrations pris en charge
- Configuration des notifications Amazon SNS pour CloudTrail
- Réception de fichiers CloudTrail journaux de plusieurs régions et réception de fichiers CloudTrail journaux de plusieurs comptes

Toutes les HealthOmics actions sont enregistrées CloudTrail et documentées dans la référence de l'AWS HealthOmics API. Par exemple, les appels au`CreateReferenceeStore`, `StartVariantImportJob` et les `CreateWorkflow` actions génèrent des entrées dans les fichiers CloudTrail journaux.

Chaque événement ou entrée de journal contient des informations sur la personne ayant initié la demande. Les informations relatives à l'identité permettent de déterminer les éléments suivants :

- Si la demande a été faite avec les informations d'identification de l'utilisateur IAM.

- Si la demande a été effectuée avec les informations d'identification de sécurité temporaires d'un rôle ou d'un utilisateur fédéré.
- Si la demande a été faite par un autre AWS service.

Pour de plus amples informations, veuillez consulter l'élément userIdentity CloudTrail .

## Comprendre les entrées du fichier HealthOmics journal

Un suivi est une configuration qui permet de transmettre des événements sous forme de fichiers journaux à un compartiment Amazon S3 que vous spécifiez. CloudTrail les fichiers journaux contiennent une ou plusieurs entrées de journal. Un événement représente une demande unique provenant de n'importe quelle source et inclut des informations sur l'action demandée, la date et l'heure de l'action, les paramètres de la demande, etc. CloudTrail les fichiers journaux ne constituent pas une trace ordonnée des appels d'API publics, ils n'apparaissent donc pas dans un ordre spécifique.

L'exemple suivant montre une entrée de CloudTrail journal illustrant l' CreateWorkflow action.

```
{
    "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"
}
```


# Utilisation EventBridge avec AWS HealthOmics

HealthOmics envoie des événements à Amazon EventBridge lorsque le statut des ressources change. Les ressources incluent les tâches d'importation, les tâches d'exportation, les partages de ressources, les flux de travail, les tâches et les exécutions. Pour chaque type de ressource, il existe une liste de changements de statut qui génèrent un événement.

Un bus d'événements est un routeur qui reçoit des événements et les achemine vers des destinations. Votre compte inclut un bus d'événements par défaut qui reçoit automatiquement les événements des AWS services. Vous pouvez créer des bus d'événements personnalisés supplémentaires.

Vous créez des EventBridge règles pour spécifier les actions à effectuer lorsque le bus d'événements reçoit des événements. Par exemple, vous pouvez créer une règle qui vous informe des changements de statut d'une ressource.

Les scénarios courants d'utilisation des événements incluent :

- Pour surveiller le moment où un utilisateur partage une ressource avec vous ou révoque le partage.
- Pour vérifier si une exécution échoue ou se termine correctement.

Pour plus d'informations sur l'utilisation EventBridge, consultez Qu'est-ce qu'Amazon EventBridge ?

Rubriques



## Configurez EventBridge pour HealthOmics

Avant de pouvoir surveiller les EventBridge événements, créez un EventBridge bus et créez des règles pour les événements qui vous intéressent.

### Configuration d'un EventBridge bus

Vous pouvez utiliser le bus d'événements par défaut pour votre bus d'événements Compte AWS ou configurer un bus d'événements personnalisé. Pour configurer un bus d'événements personnalisé, procédez comme suit :

1. Ouvrez la EventBridge console : https://console.aws.amazon.com/events/
2. Dans le menu de navigation de gauche, sélectionnez Event bus.
3. Choisissez Create event bus (Créer un bus d'événement).
4. Dans le formulaire Créer un bus d'événements, entrez le nom du bus.
5. Choisissez Create pour créer le bus.

## Création d'une EventBridge règle

La procédure suivante montre comment créer une règle simple. Pour plus d'informations sur les règles, consultez la section Règles dans EventBridge.

1. Ouvrez la EventBridge console : https://console.aws.amazon.com/events/
2. Dans le volet de navigation de gauche, choisissez Règles.
3. Choisissez Créer une règle. La console ouvre le formulaire Créer une règle.
4. Dans Définir les détails de la règle, donnez un nom à la règle.
   - Dans Nom, entrez le nom du bus.
   - Pour Event bus, sélectionnez le bus pour cette règle.
   - Choisissez Suivant.
5. Dans Créer un modèle d'événement, sous Source de l'événement, sélectionnez les événements AWS ou les événements EventBridge partenaires.
6. Faites défiler l'écran vers le bas jusqu'à Modèle d'événement.
   a. Dans Source de l'événement, sélectionnez les services AWS.
   b. Pour le service AWS, entrez omics dans le filtre de texte et sélectionnez AWS HealthOmicscomme service.
   c. Dans Type d'événement, sélectionnez l'événement qui vous intéresse (ou Tous les événements).
   d. Choisissez Suivant.
7. Dans Sélectionner une ou plusieurs cibles, sélectionnez une cible pour l'événement. Par exemple, choisissez le service AWS, le groupe de CloudWatch journaux choisi et configurez un groupe de journaux.

   Pour de nombreux types de cible, EventBridge a besoin d'une autorisation pour envoyer des événements à la cible. La console crée ces autorisations pour vous.
8. (Facultatif) Dans Configurer les balises, associez les balises à la règle.
9. Dans Révision et mise à jour, passez en revue la configuration et choisissez Créer une règle.

# EventBridge événements à HealthOmics

Le tableau suivant répertorie les événements HealthOmics envoyés à EventBridge et la liste des valeurs de statut possibles pour l'événement.

| Nom de l'événement | Valeurs de statut possibles |
| --- | --- |
| Modification du statut de la tâche d'importation d'annotations | Soumis, en cours, annulé, terminé, en échec ou terminé avec des échecs |
| Modification du statut du partage d'Annotation Store | En attente, activation, activité, suppression, suppression, échec |
| Modification du statut du magasin d'annotations | Création, création, mise à jour, mise à jour, suppression, suppression ou échec de la création |
| Lire la modification du statut du job d'activation | Soumis, en cours, terminé, échoué ou terminé avec des échecs |
| Lire > Définir > Exporter le statut de la tâche | Soumis, en cours, terminé, échoué ou terminé avec des échecs |
| Lire et configurer le changement de statut de la tâche d'importation | Soumis, en cours, terminé, échoué ou terminé avec des échecs |
| Modifier le statut du set de lecture | Traitement du téléchargement, échec du téléchargement, activité, archivage, activation ou suppression |
| Modification du statut du job Reference Import | Soumis, en cours, terminé, échoué ou terminé avec des échecs |
| Modification du statut de référence | Actif ou supprimé |
| Modification du statut du magasin de référence | Créé, mis à jour, actif ou supprimé |
| Modifier le statut d'exécution | En attente, démarrage, exécution, arrêt, terminé, supprimé, échoué ou annulé |

| Nom de l’événement | Valeurs de statut possibles |
| --- | --- |
| Modification du statut du magasin de séquences | Créé, mis à jour, actif ou supprimé |
| Modification du statut de la tâche | En attente, démarrage, exécution, arrêt, terminé, supprimé, échoué ou annulé |
| Modification du statut du job d'importation de variantes | Soumis, en cours, annulé, terminé, en échec ou terminé avec des échecs |
| Variant Store Share : modification du statut | En attente, activation, activité, suppression, suppression, échec |
| Modification du statut du magasin Variant | Création, création, mise à jour, mise à jour, suppression, suppression ou échec de la création |
| Modification du statut du partage du flux de travail | En attente, activation, activité, suppression, suppression, échec |
| Modification du statut du flux de travail | Succès de création, échec de création, succès de suppression ou échec de suppression |

## Structure des messages d’événements

HealthOmics fournit les meilleurs efforts pour envoyer des messages d'événement de changement d'état à EventBridge. L'événement est un objet doté d'une structure JSON qui contient également des détails sur les métadonnées. Vous pouvez utiliser les métadonnées comme entrée pour recréer l'événement ou pour obtenir plus d'informations. Les événements incluent les domaines suivants :

- `version`— Actuellement 0 (zéro) pour tous les événements.
- `id`— Un UUID version 4 généré pour chaque événement.
- `detail-type`— Le type d'événement envoyé.
- `account`— L' Compte AWS identifiant à 12 chiffres du propriétaire du bucket.
- `source`— Identifie le service qui a généré l'événement.
- `time`— Heure à laquelle l'événement s'est produit.

- `region`— Identifie Région AWS le compartiment.
- `resources`— Un tableau JSON contenant le nom de ressource Amazon (ARN) du bucket.
- `detail`— Objet JSON contenant des informations sur l'événement.

Les événements d'exécution incluent les champs suivants :

- `uuid`— L'identifiant unique universel pour la course.
- `workflowId`— Identifiant du flux de travail associé à cette exécution.
- `workflowName`— Nom du flux de travail associé à cette exécution.
- `runId`— Identifiant d'exécution.
- `runName`— Nom de l'exécution.
- `runOutputUri`— L'URI dans lequel l'exécution écrira ses données de sortie.

## Exemples de messages d'événements

L'exemple suivant est un événement de modification du statut d'exécution, montrant les champs supplémentaires.

```
{
    "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"
    }
}
```

L'exemple suivant est un événement lié à une modification du statut d'une tâche.

```
{
    "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"
    }
}
```

Voici un exemple d'événement lié à une modification du statut d'un ensemble de lecture.

```
{
  "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"
  }
}
```

Un événement similaire est créé pour une variante de la tâche d'importation d'un magasin.

```
{
    "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"
    }
}
```

Ce qui suit est un événement lié à une modification du statut de la tâche d'importation.

```
{
    "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"
    }
}
```

# Résolution des problèmes

Les rubriques suivantes peuvent vous aider à résoudre les problèmes que vous rencontrez lors de l'utilisation des HealthOmics flux de travail et des banques de données.

Rubriques



# Dépannage des workflows

Rubriques



## Comment résoudre les problèmes liés à un échec d'exécution ?

Utilisez l'opération GetRunAPI pour récupérer la raison de l'échec. Pour de plus amples informations, veuillez consulter Raisons d'échec de l'exécution.

## Comment résoudre les problèmes liés à l'échec d'une tâche ?

Consultez le code d'erreur indiqué dans le message d'échec de la tâche pour comprendre l'échec. Passez en revue les connexions de la tâche CloudWatch pour voir les messages de journalisation détaillés relatifs à la tâche. Si vous ne recevez pas de messages de journal détaillés, vous pouvez modifier votre flux de travail pour générer des instructions de journal supplémentaires. Pour de plus amples informations, veuillez consulter Surveillance à HealthOmics l'aide de CloudWatch journaux.

## Où puis-je trouver les journaux du moteur indiquant les essais effectués avec succès ?

HealthOmics publie les journaux uniquement CloudWatch pour les échecs d'exécution. Si une exécution est réussie, HealthOmics envoie les journaux du moteur à votre compartiment Amazon S3. Pour de plus amples informations, veuillez consulter Se connecte à Amazon S3.

## Comment puis-je réduire la taille des paramètres d'entrée pour un flux de travail ?

Vous pouvez spécifier jusqu'à 50 Ko de paramètres d'entrée pour un flux de travail. Vous pouvez utiliser des importations de répertoires ou des exemples de feuilles pour respecter cette contrainte de taille. Pour de plus amples informations, veuillez consulter Gestion de la taille des paramètres d'exécution.

## Pourquoi ma course ne se termine-t-elle pas ?

Si votre code présente des problèmes et que les processus ne se terminent pas correctement, il se peut que votre exécution ne réponde pas ou soit « bloquée ». Pour plus d'informations sur la façon de prévenir et de détecter les courses qui ne répondent pas, consultezConseils pour les essais qui ne répondent pas.

# Résolution des problèmes de mise en cache des appels

Les rubriques suivantes peuvent vous aider à résoudre les problèmes que vous rencontrez lors de la mise en cache des appels.

Rubriques

- Pourquoi ma course n'est-elle pas enregistrée dans le cache ?
- Pourquoi une tâche n'utilise-t-elle pas l'entrée du cache ?
- Pourquoi la mise en cache des appels pour une tâche est-elle désactivée ?

## Pourquoi ma course n'est-elle pas enregistrée dans le cache ?

1. Vérifiez que l'exécution est configurée pour utiliser un cache en vérifiant le champ cacheID dans la réponse à l'opération GetRun d'API. À l'aide de la CLI, exécutez cette commande :`aws omics get-run –id <run_id>`.

2. Si l'exécution a réussi, vérifiez que le comportement du cache renvoyé dans la GetRun réponse est CACHE_ALWAYS. Si le comportement du cache est défini sur CACHE_ON_FAILURE, les exécutions ne seront enregistrées dans le cache qu'en cas d'échec.

## Pourquoi une tâche n'utilise-t-elle pas l'entrée du cache ?

<cache_id><cache_uuid>Dans le groupe de `/aws/omics/WorkflowLog` CloudWatch journaux, ouvrez le flux de journal pour le cache d'exécution : RunCache//.

1. Vérifiez qu'une exécution précédente a créé une entrée de cache pour la tâche que vous vous attendiez à voir mise en cache. Les exécutions enregistrées dans le cache seront enregistrées avec un message de journal CACHE_ENTRY_CREATED.
2. Localisez le journal CACHE_MISS de la tâche et exécutez-le complètement. S'il n'y a aucune entrée dans le journal, vérifiez que l'exécution a été configurée pour utiliser le cache.
3. Si une entrée de cache a été créée, vérifiez que le CPUs résumé de la mémoire GPUs et du conteneur sont identiques pour les deux tâches. L'ARN de la tâche qui a créé l'entrée de cache figure dans le message du journal.
4. Si les exigences de calcul pour les deux tâches correspondent, vérifiez que les entrées n'ont pas changé entre les tâches. Pour ce faire, ouvrez les journaux du moteur. Si l'exécution a le statut FAILED, les journaux se trouveront dans le groupe de journaux Cloudwatch/aws/omics/WorkflowLog. Sinon, les journaux du moteur se trouvent dans le répertoire de sortie de l'exécution.

## Pourquoi la mise en cache des appels pour une tâche est-elle désactivée ?

Vérifiez si la tâche est configurée pour désactiver la mise en cache à l'aide des fonctionnalités du moteur de flux de travail :

- Pour les flux de travail WDL : vérifiez si la tâche est définie comme volatile `true` dans la méta-section
- Pour les flux de travail Nextflow : vérifiez si la directive de cache de la tâche est définie sur `false`
- Pour les flux de travail CWL : vérifiez si EnableReuse est défini sur la tâche pour la fonctionnalité `false` WorkReuse

# Dépannage des magasins de données

Rubriques



## Pourquoi S3 GetObject échoue-t-il sur mon set de lecture ?

Le plus souvent, l'échec est dû à une autorisation manquante. L'autorisation de lecture S3 du magasin de séquences est une configuration bidirectionnelle nécessitant à la fois la politique d'accès du magasin de séquences S3 pour autoriser l'accès et le principal IAM pour disposer d'une politique d'accès attachée. Pour plus de détails sur les exigences de la politique, voirAutorisations d'accès aux données à l'aide d'Amazon S3 URIs. Vérifiez que les configurations suivantes sont en place :

- La politique d'accès S3 du magasin de séquences a explicitement autorisé l'accès au principal IAM ou à la racine du compte du principal.
- Vérifiez que le principal IAM dispose d'une politique autorisant explicitement la ressource à laquelle vous accédez. Notez que la politique principale IAM doit utiliser l'ARN du point d'accès et non le chemin basé sur l'alias du point d'accès lors de la définition des autorisations et que l'ARN est conforme à la condition et n'est pas utilisé pour spécifier une ressource.
- Si votre boutique utilise une clé gérée par le client (CMK-KMS), assurez-vous que le principal IAM dispose des autorisations de kms: déchiffrement sur la clé. Consultez le guide d'accès entre comptes KMS pour configurer l'utilisation entre comptes.

Si votre politique utilise des contrôles d'accès basés sur des balises, assurez-vous de ce qui suit :

- Assurez-vous que le magasin de séquences a terminé de synchroniser les balises. Pour cela, le statut du magasin doit être active ou nonupdating.
- Assurez-vous qu'il n'y a aucune faute de frappe dans le tag, la clé ou la valeur de la clé sur le set de lecture et la politique.

## Pourquoi ne puis-je pas voir mon magasin d'annotations ou mon magasin de variantes dans Athena ?

Dans Lake Formation, veillez à créer un lien de ressource basé sur le magasin qui a été partagé avec vous. Une fois que vous avez créé un lien vers une ressource à laquelle vous êtes autorisé à accéder, le magasin devrait être visible dans Athena. Pour de plus amples informations, veuillez consulter Configuration de Lake Formation à utiliser HealthOmics.

## Pourquoi ne puis-je pas accéder à mon magasin de données dans Athena ?

Si votre magasin d'annotations ou de variantes est visible mais que vous recevez un message d'erreur indiquant que l'accès est refusé, vérifiez la version du moteur de requête que vous utilisez. Seules les requêtes exécutées à l'aide de la version 3 du moteur sont prises en charge. Pour en savoir plus sur les versions du moteur de requête Athena, consultez la documentation Amazon Athena.

# Résolution des problèmes avec Amazon Q CLI

La CLI Amazon Q peut vous aider à rationaliser votre processus de dépannage en :

- Analyse des exécutions de flux de travail et des échecs de tâches de débogage
- Collecte de journaux et de messages d'erreur pertinents
- Création de dossiers de AWS support auxquels sont joints tous les journaux de débogage nécessaires
- Supprime les informations personnelles identifiables (PII) des informations soumises au Support AWS

Pour plus d'informations sur l'utilisation d'Amazon Q CLI AWS HealthOmics pour le dépannage et la création de dossiers de support, consultez le didacticiel HealthOmics Agentic Generative AI sur. GitHub

> **Warning**
>
> Lorsque vous travaillez avec Amazon Q CLI, passez en revue tout le contenu généré et les actions proposées avant de continuer. Fournissez des commentaires pour améliorer la qualité

des réponses et répondre aux exigences de votre flux de travail. Pour plus d'informations, consultez Considérations relatives à la sécurité et bonnes pratiques pour Amazon Q.

# Quotas pour AWS HealthOmics

AWS remplit votre compte avec les valeurs par défaut pour les HealthOmics quotas. Sauf indication contraire, chaque valeur de quota est la valeur maximale par région.

> **Important**
>
> Vous pouvez demander l'augmentation de la plupart des quotas de service et des quotas d'API. Consultez les rubriques suivantes pour plus de détails.

Rubriques



## HealthOmics quotas de service

Le tableau ci-dessous répertorie les quotas HealthOmics de service, ainsi que leurs valeurs par défaut. Pour consulter les quotas actuels pour chaque région, ouvrez la console Service Quotas.

> **Important**
>
> Vous pouvez demander l'augmentation d'un quota ajustable à l'aide de la console Service Quotas.

Pour plus d'informations sur les quotas de service, consultez la section Demander une augmentation de quota dans le Guide de l'utilisateur des quotas de service. Pour un quota qui n'est pas disponible dans la console Service Quotas, utilisez le formulaire d'augmentation de quota.

| Name | Par défaut | Ajusta | Description |
| --- | --- | --- | --- |
| Analyses - Nombre maximal de magasins d'annotations | Par région prise en charge : 10 | Oui | Le nombre maximum de magasins d'annotat |

| Name | Par défaut | Ajusta | Description |
|---|---|---|---|
| | | | ions dans la AWS région actuelle |
| Analytics : nombre maximal de tâches simultanées d'importation de variantes ou de magasins d'annotations | Chaque Région prise en charge : 5 | Oui | Le nombre maximum de tâches d'importation simultanées dans la AWS région actuelle |
| Analytics - Nombre maximum de fichiers par variante de la tâche d'importation en magasin | Chaque Région prise en charge : 1 000 | Oui | Le nombre maximum de fichiers par variante de la tâche d'importation dans la AWS région actuelle |
| Analytics - Nombre maximal de parts par magasin d'annotations | Par région prise en charge : 10 | Oui | Nombre maximal de partages par magasin d'annotations dans la AWS région actuelle |
| Analytics - Nombre maximum de parts par variante de magasin | Par région prise en charge : 10 | Oui | Le nombre maximum de partages par variante de magasin dans la AWS région actuelle |
| Analytics - Taille maximale de chaque fichier dans une tâche d'importation de variante | Chaque région prise en charge : 20 gigaoctets | Oui | Taille maximale d'un fichier dans une variante d'une tâche d'importa tion dans la AWS région actuelle |
| Analytics : taille maximale de chaque fichier dans une tâche d'importation d'annotations | Chaque région prise en charge : 20 gigaoctets | Oui | Taille maximale d'un fichier dans une tâche d'importation d'annotat ions dans la AWS région actuelle |

| Name | Par défaut | Ajusta | Description |
| --- | --- | --- | --- |
| Analytics - Nombre maximum de variantes de magasins | Par région prise en charge : 10 | Oui | Le nombre maximum de variantes de magasins dans la AWS région actuelle |
| Analytics - Nombre maximum de versions par magasin d'annotations | Par région prise en charge : 10 | Oui | Le nombre maximum de versions par magasin d'annotations dans la AWS région actuelle |
| Configurations - Configurations maximales | Par région prise en charge : 10 | Oui | Le nombre maximum de configurations dans la AWS région actuelle. |
| Stockage : nombre maximal de tâches simultanées d'activation des ensembles de lecture | Chaque région prise en charge : 25 | Oui | Nombre maximal de tâches d'activation d'ensembles de lecture simultanées dans la AWS région actuelle |
| Stockage : séquence simultanée maximale et tâches d'exportation du magasin de référence | Chaque Région prise en charge : 5 | Oui | Nombre maximal de tâches d'exportation simultanées à partir d'une séquence ou d'un magasin de référence dans la AWS région actuelle |
| Stockage : nombre maximal de tâches d'importation de séquences simultanées ou de magasins de référence | Chaque Région prise en charge : 5 | Oui | Nombre maximal de tâches d'importation simultanées pour une séquence ou un magasin de référence dans la AWS région actuelle |

| Name | Par défaut | Ajusta | Description |
| --- | --- | --- | --- |
| Stockage : nombre maximum d'ensembles de lecture par magasin de séquences | Chaque Région prise en charge : 1 000 000 | Oui | Le nombre maximum d'ensembles de lecture dans un magasin de séquences dans la AWS région actuelle |
| Stockage : nombre maximum de références par magasin de référence | Chaque Région prise en charge : 50 | Oui | Le nombre maximum de références dans un magasin de référence dans la AWS région actuelle |
| Stockage - Nombre maximum de séquences mémorisées | Chaque région prise en charge : 20 | Oui | Le nombre maximum de magasins de séquences dans la AWS région actuelle |
| Flux de travail : nombre maximal d'actifs GPUs | Chaque région prise en charge : 12 | Oui | Le nombre maximum de simultanés actifs GPUs dans la AWS région actuelle. Dans us-east-1 et us-west-2, les demandes d'augment ation de quota pour des valeurs allant jusqu'à 500 sont automatiquement approuvées. |

| Name | Par défaut | Ajusta | Description |
| --- | --- | --- | --- |
| Flux de travail : nombre maximal d'exécutions actives simultanées grâce au stockage dynamique des exécutions | Chaque Région prise en charge : 50 | Oui | Nombre maximal d'exécutions actives utilisant le stockage dynamique des exécution s dans la AWS région actuelle. Les demandes d'augmentation de quota pour des valeurs allant jusqu'à 200 sont automatiquement approuvées. |
| Flux de travail - Nombre maximal d'exécutions actives simultanées en utilisant le stockage statique des exécutions | Par région prise en charge : 10 | Oui | Nombre maximal d'essais actifs utilisant le stockage statique des essais dans la AWS région actuelle. Les demandes d'augment ation de quota pour des valeurs allant jusqu'à 50 sont automatiquement approuvées. |
| Flux de travail : nombre maximal de tâches simultanées par exécution | Chaque région prise en charge : 25 | Oui | Le nombre maximum de tâches simultanées par exécution dans la AWS région actuelle. Dans us-east-1 et us-west-2, les demandes d'augment ation de quota pour des valeurs allant jusqu'à 100 sont automatiquement approuvées. |

| Name | Par défaut | Ajusta | Description |
| --- | --- | --- | --- |
| Workflows - Durée d'exécution maximale | Chaque région prise en charge : 604 800 secondes | Oui | Durée maximale d'exécution du flux de travail dans la AWS région actuelle. |
| Flux de travail - Nombre maximum d'exécutions (actives ou inactives) | Chaque région prise en charge : 100 000 | Oui | Le nombre maximum d'essais (actifs ou inactifs) dans la AWS région actuelle. |
| Flux de travail : nombre maximum de partages par flux de travail | Chaque région prise en charge : 100 | Oui | Le nombre maximum de partages par flux de travail dans la AWS région actuelle |
| Workflows - Capacité maximale de stockage d'essais statiques par cycle | Chaque région prise en charge : 9 600 | Oui | Capacité de stockage statique maximale en gibioctets (GiB) pour chaque exécution dans la région actuelle. AWS Dans us-east-1 et us-west-2, les demandes d'augmentation de quota pour des valeurs allant jusqu'à 50 000 sont automatiquement approuvées. |
| Flux de travail : flux de travail maximaux | Chaque Région prise en charge : 1 000 | Oui | Le nombre maximum de flux de travail dans la AWS région actuelle. |
| Flux de travail - Transactions par seconde (TPS) pour l' StartRun opération | Par région prise en charge : 1 | Oui | Le nombre maximum de transactions par seconde (TPS) pour l' StartRun opération dans la AWS région actuelle. |

# HealthOmics quotas de taille fixe

En plus de celaHealthOmics quotas de service, HealthOmics inclut des quotas dont la taille est fixe. Vous ne pouvez pas demander d'augmentation pour ces valeurs.

Sauf indication contraire, chaque quota indique la valeur maximale par région.

Rubriques

- HealthOmics quotas de taille fixe pour les analyses
- HealthOmics quotas de stockage de taille fixe
- HealthOmics quotas de taille fixe du flux de travail
- HealthOmics Quotas de taille fixe pour le workflow Ready2Run

## HealthOmics quotas de taille fixe pour les analyses

Le tableau suivant indique les valeurs maximales prises en charge pour les quotas d'analyse. Ces valeurs ne sont pas ajustables.

| Nom | Description | Maximum | Réglable Oui/Non |
|---|---|---|---|
| Analytics - Nombre maximum de fichiers par tâche d'importation du magasin d'annotations | Nombre maximal de fichiers par tâche d'importation d'annotations. | 1 | Non |

## HealthOmics quotas de stockage de taille fixe

Le tableau suivant indique les valeurs maximales prises en charge pour les fichiers de stockage. Ces valeurs ne sont pas ajustables.

| Nom | Description | Maximum | Réglable Oui/Non |
|---|---|---|---|
| Stockage : taille maximale de la | La taille maximale de la politique de | 15 KO | Non |

| Nom | Description | Maximum | Réglable Oui/Non |
| --- | --- | --- | --- |
| politique de ressource s d'accès S3 | ressources d'accès S3 | | |
| Stockage : nombre maximal de balises de niveau défini propagées | Le nombre maximum de clés de balise de niveau défini, par magasin, qui se propagent à l'objet S3 | 5 | Non |
| Stockage : nombre maximum de jeux de lecture par tâche d'activation | Le nombre maximum d'ensembles de lecture par tâche d'activation. | 20 | Non |
| Stockage : nombre maximum de jeux de lecture par tâche d'exportation | Le nombre maximum de jeux de lecture par tâche d'exportation. | 100 | Non |
| Stockage : nombre maximum de jeux de lecture par tâche d'importation | Le nombre maximum d'ensembles de lecture par tâche d'importation. | 100 | Non |
| Stockage - Nombre maximum de magasins de référence | Le nombre maximum de magasins de référence. | 1 | Non |
| Stockage : taille maximale des pièces pour un télécharg ement direct | Taille de pièce maximale pour le téléchargement direct vers un magasin de séquences. | 100 Mo | Non |

| Nom | Description | Maximum | Réglable Oui/Non |
|---|---|---|---|
| Stockage - Nombre maximum de pièces dans le fichier pour le téléchargement direct | Nombre maximal de parties d'un fichier pouvant être téléchargées directement vers un magasin de séquences. | 10 000 | Non |
| Stockage - Taille de référence maximale | Taille maximale d'un fichier de référence pouvant être importé dans un magasin de référence. | 15 Go | Non |
| Stockage : taille maximale de la source du set de lecture | Taille maximale d'un seul fichier source dans un ensemble de lectures pouvant être importé dans un magasin de séquences. | 976 GO | Non |

## HealthOmics quotas de taille fixe du flux de travail

Le tableau suivant indique les valeurs maximales prises en charge pour les quotas de flux de travail. Ces valeurs ne sont pas ajustables.

| Nom | Description | Taille maximum | Réglable Oui/Non |
|---|---|---|---|
| Workflows - Nombre maximum de groupes d'exécution | Le nombre maximum de groupes d'essais. | 1 000 | Non |
| Workflows - Nombre maximal de caches d'exécution | Nombre maximal de caches d'exécuti | 1 000 | Non |

| Nom | Description | Taille maximum | Réglable Oui/Non |
| --- | --- | --- | --- |
| | on que vous pouvez créer pour un compte.<br><br>Une ou plusieurs exécutions peuvent partager le même cache d'exécuti on. Il n'y a pas de quota pour le nombre d'exécutions HealthOmics pouvant être mises en cache par compte. | | |
| Workflows - Nombre maximum de versions de workflows | Le nombre maximum de versions de flux de travail par flux de travail. | 1 000 | Non |
| Workflows : taille du conteneur de l'instanc e du processeur | Taille d'image de conteneur maximale pour une instance de processeur. | 45 GiB | Non |
| Workflows : taille du conteneur de l'instanc e GPU | Taille maximale de l'image de conteneur pour une instance de GPU. | 95 GiB | Non |
| Mémoire partagée de l'instance GPU /dev/ shm | La quantité maximale de mémoire partagée par instance de GPU. | 8 Go par GPU | Non |
| Workflows - Exécuter le fichier de paramètre s | Taille maximale d'un fichier de paramètres d'exécution. | 50 000 octets | Non |

| Nom | Description | Taille maximum | Réglable Oui/Non |
| --- | --- | --- | --- |
| Workflows - Fichier modèle de paramètres de flux de travail | Nombre maximum d'entrées et taille de fichier maximale pour un fichier modèle de paramètres de flux de travail. Ce quota s'applique aux flux de travail que vous créez à l'aide de la console ou de l'API. | 1 000 entrées, 400 Ko | Non |
| Flux de travail - Taille du fichier de définition du flux de travail - API | Taille maximale du fichier de définition du flux de travail lorsque vous créez le flux de travail à l'aide de l'opération API ou d'un AWS SDK. | 100 Mo | Non |
| Flux de travail - Taille du fichier de définition du flux de travail - Console (téléchargement direct) | Taille maximale du fichier de définition du flux de travail que vous pouvez fournir sous forme de téléchargement direct, lorsque vous créez le flux de travail à l'aide de la console. | 4,4 MO | Non |

| Nom | Description | Taille maximum | Réglable Oui/Non |
|---|---|---|---|
| Flux de travail - Taille du fichier de définition du flux de travail - Console (téléchargement depuis Amazon S3) | Taille maximale du fichier de définition du flux de travail que vous pouvez fournir sous forme de téléchargement depuis Amazon S3, lorsque vous créez le flux de travail à l'aide de la console. | 100 Mo | Non |
| Workflows - Taille du référentiel | Taille maximale d'un référentiel de code externe. | 1 Gio | Non |
| Workflows - Taille de fichier individuelle du référentiel | Taille maximale d'un fichier individuel provenant d'un référentiel de code externe. | 100 Mio | Non |
| Workflows - Taille du fichier README | Taille maximale d'un fichier README. | 500 KIB | Non |

Pour obtenir des suggestions sur la manière de réduire la taille de votre fichier de paramètres d'exécution, consultezGestion de la taille des paramètres d'exécution.

## HealthOmics Quotas de taille fixe pour le workflow Ready2Run

Chaque flux de travail Ready2Run possède une taille de fichier d'entrée maximale. Dans le tableau suivant, les unités de taille de fichier sont répertoriées en Gibioctets (GiB). Ces tailles de fichier maximales ne sont pas ajustables.

| Nom du flux de travail Ready2Run | Taille maximale du fichier d'entrée (GiB) | Réglable (Oui/Non) |
|---|---|---|
| AlphaFold pour 601 à 1200 résidus | 1 | Non |
| AlphaFold pour un maximum de 600 résidus | 1 | Non |
| Bases2Fastq pour 2x150 | 1 000 | Non |
| Bases2Fastq pour 2x300 | 1 000 | Non |
| Bases2Fastq pour 2x75 | 500 | Non |
| ESMFold pour jusqu'à 800 résidus | 1 | Non |
| GATK-BP fq2bam | 64 | Non |
| GATK-BP Germline bam2vcf pour un génome 30x | 39 | Non |
| Lignée germinale GATK-BP fq2vcf pour le génome 30x | 64 | Non |
| GATK-BP Somatic WES bam2vcf | 86 | Non |
| NVIDIA Parabricks BAM2 FQ2 BAM WGS jusqu'à 30 fois | 80 | Non |
| NVIDIA Parabricks BAM2 FQ2 BAM WGS jusqu'à 50 fois | 120 | Non |
| NVIDIA Parabricks BAM2 FQ2 BAM WGS pour un maximum de 5 fois | 20 | Non |

| Nom du flux de travail Ready2Run | Taille maximale du fichier d'entrée (GiB) | Réglable (Oui/Non) |
|---|---|---|
| NVIDIA Parabricks FQ2 BAM WGS jusqu'à 30 fois | 71 | Non |
| NVIDIA Parabricks FQ2 BAM WGS jusqu'à 50 fois | 137 | Non |
| NVIDIA Parabricks FQ2 BAM WGS pour un maximum de 5 fois | 13 | Non |
| NVIDIA Parabricks Germline DeepVariant WGS jusqu'à 30 fois | 71 | Non |
| NVIDIA Parabricks Germline DeepVariant WGS jusqu'à 50 fois | 137 | Non |
| NVIDIA Parabricks Germline DeepVariant WGS pour un maximum de 5 fois | 12 | Non |
| NVIDIA Parabricks Germline HaplotypeCaller WGS jusqu'à 30 fois | 71 | Non |
| NVIDIA Parabricks Germline HaplotypeCaller WGS jusqu'à 50 fois | 137 | Non |
| NVIDIA Parabricks Germline HaplotypeCaller WGS pour un maximum de 5 fois | 13 | Non |
| NVIDIA Parabricks Somatic Mutect2 WGS jusqu'à 50 fois | 196 | Non |

| Nom du flux de travail Ready2Run | Taille maximale du fichier d'entrée (GiB) | Réglable (Oui/Non) |
|---|---|---|
| sc RNAseq avec Kallisto BUStools | 119 | Non |
| sc RNAseq avec saumon sauté aux alevins | 119 | Non |
| sc RNAseq avec STARsolo | 119 | Non |
| Sentieon Germline BAM WES jusqu'à 300 fois | 9 | Non |
| Sentieon Germline BAM WGS jusqu'à 32 fois | 18 | Non |
| Sentieon Germline FASTQ WES jusqu'à 100x | 5 | Non |
| Sentieon Germline FASTQ WES jusqu'à 300 fois | 26 | Non |
| Sentieon Germline FASTQ WGS jusqu'à 32 fois | 51 | Non |
| Sentieon LongRead pour ONT | 25 | Non |
| Sentieon LongRead pour PacBio HiFi | 58 | Non |
| Sentieon Somatic WES | 50 | Non |
| Sentieon Somatic WGS | 113 | Non |
| Ultima Genomics jusqu' DeepVariant à 40 fois | 91 | Non |

# HealthOmics Quotas d'API

HealthOmics possède les quotas suivants liés aux opérations d'API. Lorsque cela est indiqué, le quota est ajustable. Pour demander une augmentation, utilisez le formulaire d'augmentation de quota.

Pour chaque opération d'API répertoriée, le quota correspond au nombre maximal de transactions par seconde (TPS) pour cette opération d'API dans chaque région.

Rubriques



## Quotas d'API généraux

Le tableau suivant répertorie les opérations d'API générales qui s'appliquent à plusieurs catégories (stockage, flux de travail et analyses).

| Opération API | TPS maximum par défaut | Réglable (Oui/Non) |
|---|---|---|
| AcceptShare, CreateShare, DeleteShare, GetShare, ListShares | 1 TPS | Oui |

## Quotas d'API de stockage

Le tableau suivant répertorie les opérations de l'API de stockage.

| Fonctionnement de l'API de stockage | TPS maximum par défaut | Réglable (Oui/Non) |
|---|---|---|
| CreateSequenceStore, UpdateSequenceStore, DeleteSequenceStore, | 1 TPS | Oui |

| Fonctionnement de l'API de stockage | TPS maximum par défaut | Réglable (Oui/Non) |
|---|---|---|
| CreateReferenceStore, DeleteReferenceStore | | |
| BatchDeleteReadSet, DeleteReference | 1 TPS | Oui |
| CreateMultipartReadSetUpload, CompleteMultipartReadSetUpload, AbortMultipartReadSetUpload | 1 TPS | Non |
| obtient S3AccessPolicy, met 3, supprime 3 AccessPolicy AccessPolicy | 1 TPS | Oui |
| GetReference | 10 TPS | Oui |
| UploadReadSetPart | 10 TPS | Oui |
| GetReadSet | 30 TPS | Oui |
| GetSequenceStore, ListSequenceStores | 5 TPS | Oui |
| GetReadSetMetadata, ListReadSets | 5 TPS | Oui |
| StartReadSetImportJob, GetReadSetImportJob, ListReadSetImportJobs | 5 TPS | Oui |
| StartReadSetExportJob, GetReadSetExportJob, ListReadSetExportJobs | 5 TPS | Oui |
| ListReferenceStores | 5 TPS | Oui |

| Fonctionnement de l'API de stockage | TPS maximum par défaut | Réglable (Oui/Non) |
|---|---|---|
| StartReferencetImportJob, GetReferenceImportJob, ListReferenceImportJobs | 5 TPS | Oui |
| ListReferences, GetReferenceMetadata | 5 TPS | Oui |
| StartReadsetActivationJob | 5 TPS | Oui |
| ListReadsetActivationJobs, GetReadSetActivationJob | 5 TPS | Oui |
| ListMultipartReadSetUploads, ListReadSetUploadParts | 5 TPS | Oui |
| TagResource, UntagResource, ListTagsForResource | 5 TPS | Oui |

## Quotas d'API Workflow

Le tableau suivant répertorie les opérations de l'API de flux de travail.

| Fonctionnement de l'API Workflow | TPS maximum par défaut | Réglable (Oui/Non) |
|---|---|---|
| StartRun | 1 TPS | Oui |
| CreateWorkflow | 5 TPS | Oui |
| CancelRun, DeleteRun, GetRun, GetRunTask, ListRunTasks, ListRuns | 10 TPS | Oui |
| CreateRunGroup, DeleteRunGroup, GetRunGroup, | 10 TPS | Oui |

| Fonctionnement de l'API Workflow | TPS maximum par défaut | Réglable (Oui/Non) |
|---|---|---|
| ListRunGroups, UpdateRun Group | | |
| CreateRunCache, UpdateRun Cache, DeleteRunCache, GetRunCache, ListRunCaches | 10 TPS | Oui |
| DeleteWorkflow, GetWorkfl ow, ListWorkflows, UpdateWor kflow | 10 TPS | Oui |

## Quotas d'API d'analyse

Le tableau suivant répertorie les opérations de l'API d'analyse.

| Fonctionnement de l'API Analytics | TPS maximum par défaut | Réglable (Oui/Non) |
|---|---|---|
| CreateVariantStore, DeleteVar iantStore, GetVariantStore, ListVariantStores, UpdateVar iantStore | 1 TPS | Non |
| StartVariantImportJob, CancelVariantImportJob, GetVariantImportJob, ListVaria ntImportJobs | 1 TPS | Non |
| CreateAnnotationStore, DeleteAnnotationStore, GetAnnotationStore, ListAnnot ationStores, UpdateAnn otationStore | 1 TPS | Non |

| Fonctionnement de l'API Analytics | TPS maximum par défaut | Réglable (Oui/Non) |
| --- | --- | --- |
| StartAnnotationImportJob, ListAnnotationImportJobs, GetAnnotationImportJob, CancelAnnotationImportJob | 1 TPS | Non |

# Historique du document pour le guide de HealthOmics l'utilisateur

Le tableau suivant décrit les versions de documentation pour HealthOmics.

| Modification | Description | Date |
| --- | --- | --- |
| AWS HealthOmics les magasins de variantes et les magasins d'annotations ne sont plus ouverts aux nouveaux clients. | AWS HealthOmics les magasins de variantes et les magasins d'annotations ne sont plus ouverts aux nouveaux clients. Pour plus d'informations, voir Modification de la disponibilité du magasin de AWS HealthOmics variantes et du magasin d'annotations. | 7 novembre 2025 |
| AWS HealthOmics les magasins de variantes et les magasins d'annotations ne seront plus ouverts aux nouveaux clients à compter du 7 novembre 2025. | AWS HealthOmics les magasins de variantes et les magasins d'annotations ne seront plus ouverts aux nouveaux clients à compter du 7 novembre 2025. Si vous souhaitez utiliser des magasins de variantes ou des magasins d'annotations, inscrivez-vous avant cette date. Les clients existants peuvent continuer à utiliser le service normalement. Pour plus d'informations, voir Modification de la disponibilité du magasin de AWS | 7 octobre 2025 |

| | | |
|---|---|---|
| | HealthOmics variantes et du magasin d'annotations. | |
| Nouvelles fonctionnalités | HealthOmics ajout de la prise en charge des flux de travail pour synchroniser un référenti el Amazon ECR privé avec un registre en amont. Pour en savoir plus, consultez la section Images de conteneur s pour les flux de travail privés dans HealthOmics. | 28 août 2025 |
| Nouvelles fonctionnalités d'intégration du README et du référentiel | Ajout de la prise en charge de la création de flux de travail à partir de référentiels de code externes et de fichiers README. | 24 juillet 2025 |
| Nouvelles fonctionnalités | HealthOmics ajout du support pour l'interpolation automatiq ue des paramètres de Nextflow. Pour en savoir plus, consultez la section Fichiers modèles de paramètres pour les HealthOmics flux de travail. | 27 juin 2025 |
| Nouvelles fonctionnalités | HealthOmics ajout de la prise en charge des flux de travail pour interpoler les paramètres d'exécution à partir d'un fichier de définition de flux de travail WDL. Pour en savoir plus, consultez la section Fichiers modèles de paramètres pour les HealthOmics flux de travail. | 30 mai 2025 |

| | | |
|---|---|---|
| Nouvelles fonctionnalités | HealthOmics ajout du support pour le versionnement des flux de travail. Pour en savoir plus, consultez la section Gestion des versions des flux de travail dans HealthOmics. | 18 avril 2025 |
| Nouvelles fonctionnalités | HealthOmics débit élastique ajouté pour le stockage dynamique des exécutions. Pour en savoir plus, consultez la section Exécuter les types de stockage dans HealthOmics. | 16 avril 2025 |
| Nouvelles fonctionnalités | HealthOmics ajout de contrôles d'accès basés sur des attributs pour les emplacements de Sequence Store S3 et possibilité de synchroniser jusqu'à cinq balises de lecture avec un objet Sequence Store S3. Pour en savoir plus, consultez la section Création d'un magasin de HealthOmics séquences. | 22 novembre 2024 |
| Nouvelles fonctionnalités | HealthOmics ajout de la prise en charge de la mise en cache des appels, également connue sous le nom de CV, pour les flux de travail privés. Pour en savoir plus, consultez la section Mise en cache des appels. | 20 novembre 2024 |

| | | |
|---|---|---|
| Nouvelles fonctionnalités | HealthOmics a ajouté de nouveaux champs d'API pour vous aider à établir une correspondance entre les tâches d'entrée du magasin de séquences et les ensembles de lecture. | 29 août 2024 |
| Nouvelles fonctionnalités | HealthOmics ajout d'un support pour la gestion des versions de Nextflow. Pour en savoir plus, consultez les versions de Nextflow. | 14 août 2024 |
| Nouvelles fonctionnalités | HealthOmics prise en charge ajoutée des flux de travail partagés et du stockage dynamique des exécutions. | 30 avril 2024 |
| Nouvelles fonctionnalités | HealthOmics prise en charge ajoutée de l'accès d'Amazon S3 aux magasins de référence et de séquences, et prise en charge de SHA256 ETags. | 15 avril 2024 |
| Nouvelles fonctionnalités | HealthOmics balises d'entité ajoutées (ETags) pour les magasins de séquences. | 6 octobre 2023 |
| Nouvelles fonctionnalités | HealthOmics ajout de la gestion des versions du magasin d'annotations et du partage du magasin d'analyse s. | 15 août 2023 |

| | | |
|---|---|---|
| Nouvelles fonctionnalités | HealthOmics a ajouté le langage CWL (Common Workflow Language) en tant que langage pris en charge pour les HealthOmics flux de travail. | 30 juin 2023 |
| Nouvelles fonctionnalités | HealthOmics a ajouté de nouveaux flux de travail Ready2Run, la prise en charge du GPU pour les flux de travail, l'analyse des données pour les magasins d'annotations, le téléchargement direct dans le HealthOmics stockage et l'intégration avec. EventBridge | 15 mai 2023 |
| Nouvelle politique gérée | HealthOmics a ajouté une nouvelle politique gérée qui fournit un accès complet. Pour en savoir plus, consultez les politiques gérées par AWS. | 23 février 2023 |
| Nouvelle politique gérée | HealthOmics a ajouté une nouvelle politique gérée qui limite l'accès à la lecture seule. Pour en savoir plus, consultez les politiques gérées par AWS. | 29 novembre 2022 |
| Première version | Publication initiale du guide de HealthOmics l'utilisateur | 29 novembre 2022 |

Les traductions sont fournies par des outils de traduction automatique. En cas de conflit entre le contenu d'une traduction et celui de la version originale en anglais, la version anglaise prévaudra.