aws

User Guide

AWS HealthOmics

Version latest

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

AWS HealthOmics: User Guide

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

Die Handelsmarken und die Handelsaufmachung von Amazon dürfen nicht in einer Weise in Verbindung mit nicht von Amazon stammenden Produkten oder Services verwendet werden, durch die Kunden irregeführt werden könnten oder Amazon in schlechtem Licht dargestellt oder diskreditiert werden könnte. Alle anderen Handelsmarken, die nicht Eigentum von Amazon sind, gehören den jeweiligen Besitzern, die möglicherweise zu Amazon gehören oder nicht, mit Amazon verbunden sind oder von Amazon gesponsert werden.

Table of Contents

Was ist AWS HealthOmics? 1
Wichtiger Hinweis 1
HealthOmics features 1
Konzepte 3
Arbeitsabläufe 3
Speicher 3
Analysen 4
Zugehörige Services 4
Wie greife ich zu HealthOmics 5
Regionen und Endpunkte für AWS HealthOmics 6
Weitere Informationen 6
AWS HealthOmics Änderung der Verfügbarkeit von Variantenspeicher und Annotationsspeicher 7
Überblick über die Migrationsoptionen 7
Migrationsoptionen für ETL-Logik 8
Migrationsoptionen für Speicher 8
Analysen 8
AWS Partner 9
Beispiele 9
Athena DDL 9
Erstellen Sie Tabellen mit Python (ohne Athena) 10
Einrichten HealthOmics 13
Melde dich an für AWS-Konto 13
Erstellen Sie IAM-Berechtigungen für HealthOmics 13
Stellen Sie eine Verbindung zu externen Code-Repositorys her 14
Verwenden von Kiro CLI mit HealthOmics 14
Verwenden von Kiro IDE mit HealthOmics 14
Erste Schritte 16
Verwenden eines Ready2Run-Workflows in der Konsole HealthOmics 16
Beispielaufforderungen für Kiro CLI 17
Private Arbeitsabläufe 18
Workflows erstellen 20
Git-Repository-Integration 22
Workflow-Definitionsdateien 26
Parameter-Vorlagendateien 114

Container-Images ... 127
Umgebungsvariablen ... 162
Workflow-README-Dateien ... 164
Optional: Sentieon-Lizenzserver ... 167
Illumina DRAGEN ... 170
Unterstützte Workflow-Sprachen ... 184
Hinweise für Arbeitsabläufe ... 203
Lokale Definitionsvalidierung ... 207
Workflow-Operationen ... 209
Workflow-Versionierung ... 229
Standardversion ... 230
Erstellen einer Version ... 230
Eine Version aktualisieren ... 238
Löschen einer Version ... 240
HealthOmics läuft ... 242
Speichertypen ausführen ... 243
Flüchtiger Speicher ... 247
Führen Sie den Aufbewahrungsmodus aus ... 261
Eingaben ausführen ... 262
Lebenszyklus ausführen ... 267
Ausgaben ausführen ... 272
Gründe für Ausführungsfehler ... 274
Lebenszyklus von Aufgaben ... 282
Einen Lauf debuggen ... 285
Optimierung ausführen ... 292
Operationen ausführen ... 303
HealthOmics Batch-Läufe ... 328
Konzepte für die Batch-Ausführung ... 329
Voraussetzungen ... 330
IAM-Berechtigungen für Batch-Operationen ... 330
Erste Schritte: Reichen Sie Ihren ersten Stapel ein ... 333
Einen Batch-Lauf starten ... 335
Überwachung des Batch-Fortschritts ... 343
Behandlung fehlgeschlagener Läufe ... 347
Einen Batch stornieren ... 351
Löschen von Batchläufen ... 352

Löschen von Batch-Metadaten ... 352
EventBridge Ereignisse für Stapel ... 353
Einschränkungen und Überlegungen ... 357
Ausführungsgruppen ... 358
Priorität ausführen ... 358
Erstellen Sie mit der Konsole eine Ausführungsgruppe ... 359
Erstellen Sie eine Ausführungsgruppe mit der CLI ... 360
Löschen Sie eine Ausführungsgruppe mithilfe der Konsole ... 361
Löschen Sie eine Ausführungsgruppe mit der CLI ... 361
Rufen Sie Caching auf ... 361
So funktioniert das Caching von Anrufen ... 362
Einen Run-Cache erstellen ... 369
Einen Run-Cache aktualisieren ... 371
Löschen eines Run-Caches ... 372
Inhalt eines Run-Caches ... 373
Engine-specific Caching-Funktionen ... 373
Verwenden Sie den Run-Cache ... 374
Arbeitsabläufe teilen ... 379
Einen gemeinsamen Workflow abonnieren ... 380
Überwachung des Status einer Workflow-Freigabe ... 381
Einen privaten Workflow teilen ... 382
Akzeptieren eines gemeinsamen Workflows mithilfe der Konsole ... 383
Einen gemeinsamen Workflow mithilfe der Konsole ausführen ... 383
Ausführen eines gemeinsamen Workflows mithilfe der API ... 384
Workflows mit einer VPC verbinden ... 384
Wann sollte das VPC-Netzwerk verwendet werden ... 385
Netzwerkmodi ... 386
Erste Schritte ... 386
VPC-Anforderungen ... 389
Service-linked Rolle ... 390
Cross-account VPC-Zugriff ... 393
Konfigurations-APIs ... 411
Ausführen von Workflows mit VPC-Netzwerken ... 414
Fehlerbehebung ... 416
Bewährte Methoden ... 426
Kontingente für VPC-Netzwerke ... 428

Internetzugang für VPC-Workflows ... 428
Einrichtung einer VPC mit Internetzugang ... 429
VPC-Endpunkte für AWS service ... 432
Sicherheitsgruppenkonfiguration ... 434
Konfiguration der Routing-Tabelle ... 434
IAM-Berechtigungen für AWS service ... 435
Testen der VPC-Konnektivität ... 437
Beispiele ... 439
Bewährte Methoden ... 441
Ready2Run-Workflows ... 442
Verfügbare Workflows ... 443
Sentieon-Workflows abonnieren ... 449
Ready2Run-Workflows starten (Konsole) ... 449
Ready2Run-Workflows (API) starten ... 451
HealthOmics Lager ... 453
HealthOmics ETags ... 454
Amazon S3-eTags ... 454
Wie HealthOmics berechnet ETags ... 455
Einen Referenzspeicher erstellen ... 456
Erstellen eines Referenzspeichers mithilfe der Konsole ... 456
Erstellen eines Referenzspeichers mit der CLI ... 457
Einen Sequenzspeicher erstellen ... 462
Erstellen eines Sequenzspeichers mithilfe der Konsole ... 463
Einen Sequenzspeicher mit der CLI erstellen ... 464
Aktualisierung eines Sequenzspeichers ... 466
Aktualisierung der Read-Set-Tags für einen Sequenzspeicher ... 467
Genomdateien importieren ... 467
Löschen von Speichern ... 468
Lesesätze in einen Sequenzspeicher importieren ... 468
Laden Sie Dateien auf Amazon S3 hoch ... 469
Erstellen einer Manifestdatei ... 470
Den Importjob starten ... 473
Überwachen Sie den Importauftrag ... 473
Suchen Sie die importierten Sequenzdateien ... 475
Ruft Details zu einem Lesesatz ab ... 478
Laden Sie die Read-Set-Datendateien herunter ... 479

Direkter Upload in einen Sequenzspeicher ... 480
Direkter Upload in einen Sequenzspeicher mit dem AWS CLI ... 480
Konfigurieren Sie einen Ausweichort ... 486
Lesesätze exportieren ... 487
Zugreifen auf Lesesätze mit Amazon S3-URIs ... 489
Amazon S3-URI-Struktur in HealthOmics Lager ... 491
Verwenden von Hosted oder Local IGV für den Zugriff auf Lesesätze ... 492
Verwenden Sie Samtools oder HTSLib in HealthOmics ... 493
Mit Mountpoint HealthOmics ... 493
Verwenden CloudFront mit HealthOmics ... 494
Lesesätze aktivieren ... 494
HealthOmics Analytik ... 498
Variantenspeicher erstellen ... 499
Einen Variantenspeicher mithilfe der Konsole erstellen ... 499
Einen Variantenspeicher mithilfe der API erstellen ... 500
Importjobs für Variantenspeicher erstellen ... 502
Annotationsspeicher erstellen ... 506
Erstellen eines Annotationsspeichers mithilfe der Konsole ... 506
Einen Annotationsspeicher mithilfe der API erstellen ... 507
Importaufträge für Annotationsspeicher werden erstellt ... 509
Einen Job zum Importieren von Anmerkungen mithilfe der API erstellen ... 509
Zusätzliche Parameter für die Formate TSV und VCF ... 511
Annotationsspeicher im TSV-Format erstellen ... 512
Importaufträge im VCF-Format werden gestartet ... 515
Annotation-Store-Versionen erstellen ... 516
Analytics-Speicher löschen ... 520
Abfragen von Analysedaten ... 521
Konfiguration von Lake Formation ... 522
Athena für Abfragen konfigurieren ... 525
Abfragen ausführen ... 526
Analyseshops teilen ... 528
Eine Shop-Freigabe erstellen ... 528
Freigabe von Ressourcen ... 530
Einen Share erstellen ... 531
Ruft Informationen über eine Aktie ab ... 532
Sehen Sie sich die Aktien an, die Sie besitzen ... 532

Sehen Sie sich akzeptierte Aktien von anderen Konten an 532
Löschen Sie eine Aktie 533
Ressourcen taggen in HealthOmics 534
Wichtiger Hinweis 534
Ressourcen taggen HealthOmics 534
Best Practices 536
Anforderungen zum Markieren 536
Sequenzspeicher, gelesene Satz-Tags 537
Hinzufügen von Tags 537
Auflisten von Tags 538
Entfernen von Tags 539
Berechtigungen 540
Benutzerrichtlinien 540
Definieren Sie benutzerdefinierte IAM-Berechtigungen für Läufe 542
Servicerollen 543
Beispiel für IAM-Servicerichtlinien 544
Verwenden Sie Sitzungsrichtlinien, um Berechtigungen einzuschränken 547
Beispiel CloudFormation Vorlage 549
Amazon-ECR-Berechtigungen 551
Erstellen Sie eine Ressourcenrichtlinie für das Amazon ECR-Repository 552
Ausführen von Workflows mit kontoübergreifenden Containern 553
Amazon ECR-Richtlinien für gemeinsame Workflows 554
Richtlinien für Amazon ECR ziehen den Cache durch 557
Richtlinie zur Dienstrolle für die Unterstützung von Containern, die nicht von PTC stammen 562
Ich verifiziere das HealthOmics kann ein Container-Image abrufen 563
Berechtigungen für Ressourcen 568
Lake-Formation-Berechtigungen 569
Amazon S3 S3-URI-Berechtigungen 570
Auf Richtlinien basierendes Teilen 570
Beispiel für eine Einschränkung 575
Sicherheit 578
Datenschutz 579
Verschlüsselung im Ruhezustand 580
Verschlüsselung während der Übertragung 591
Identity and Access Management 591

Zielgruppe ... 592
Authentifizierung mit Identitäten ... 592
Verwalten des Zugriffs mit Richtlinien ... 594
Wie AWS HealthOmics funktioniert mit IAM ... 595
Beispiele für identitätsbasierte Richtlinien ... 603
AWS verwaltete Richtlinien ... 606
Fehlerbehebung ... 610
Verwenden von servicegebundenen Rollen ... 612
Compliance-Validierung ... 614
Ausfallsicherheit ... 616
VPC-Endpunkte (AWS PrivateLink) ... 617
Überlegungen zu HealthOmics VPC-Endpunkten ... 617
Erstellen eines Schnittstellen-VPC-Endpunkts für HealthOmics ... 618
Erstellen einer VPC-Endpunktrichtlinie für HealthOmics ... 618
Besondere Überlegungen für den Zugriff auf Lesesätze mithilfe von Amazon S3-URIs ... 620
Überwachung von AWS HealthOmics ... 621
S3-Zugriffsprotokollierung ... 622
CloudWatch Metriken ... 623
AWS HealthOmics Metriken anzeigen ... 623
Erstellen eines -Alarms ... 624
Führen Sie Metriken für private Workflows aus ... 625
Aktivieren von -Metriken ... 626
Verfügbare Metriken ... 626
Metriken abfragen ... 634
Erstellen eines -Alarms ... 635
Hinzufügen von Metriken zu einem Dashboard ... 635
MCP-Server ... 636
Callouts ... 636
Fakturierung ... 637
Abmeldung ... 637
CloudWatch Logs ... 638
Protokolltypen für HealthOmics Workflows ... 638
Loggt sich ein CloudWatch ... 640
Loggt sich in Amazon S3 ein ... 641
Interaktiv CloudWatch Loggt sich in der CLI ein ... 642
Zugreifen CloudWatch Protokolle von der Konsole ... 642

CloudTrail protokolliert ... 643
HealthOmics Informationen in CloudTrail ... 643
HealthOmics Logdateieinträge verstehen ... 645
EventBridge ... 646
Aufstellen EventBridge zum HealthOmics ... 647
EventBridge Ereignisse in HealthOmics ... 648
Struktur von Ereignismeldungen ... 650
Beispiele für Event-Meldungen ... 651
Fehlerbehebung ... 655
Problembehandlung bei Workflows ... 655
Wie behebe ich Fehler bei einem fehlgeschlagenen Lauf? ... 655
Wie behebe ich eine fehlgeschlagene Aufgabe? ... 656
Wo finde ich die Engine-Logs? ... 656
Wie kann ich die Größe der Eingabeparameter für einen Workflow reduzieren? ... 656
Warum wird mein Lauf nicht abgeschlossen? ... 656
Warum ist mein Lauf mit INSTANCE_RESERVATION_FAILED oder ALL_PROFILES_INSTANCE_RESERVATION_FAILED fehlgeschlagen? ... 656
Fehlerbehebung bei Laufmetriken ... 657
Warum läuft meine Aufgabe langsamer als erwartet? ... 657
Warum ist mein Lauf fehlgeschlagen, weil der Speicherplatz knapp wurde? ... 657
Wie dimensioniere ich die Rechenleistung und den Speicher für meinen Workflow richtig? ... 658
Warum sehe ich keine Metriken für meinen Lauf? ... 658
Wie unterscheidet sich das von den Ressourcenstatistiken für das Run-Manifestprotokoll? ... 659
Wie unterscheiden sich die Ausführungsmetriken in Query Studio vom Nextflow-Ausführungsbericht? ... 659
Behebung von Problemen beim Zwischenspeichern von Anrufen ... 660
Warum wird mein Lauf nicht im Cache gespeichert? ... 660
Warum verwendet eine Aufgabe den Cache-Eintrag nicht? ... 660
Warum ist das Caching von Aufrufen für eine Aufgabe deaktiviert? ... 661
Problembehandlung bei Datenspeichern ... 661
Warum ist S3 GetObject schlägt mein Lesesatz fehl? ... 661
Warum kann ich meinen Annotations- oder Variantenspeicher in Athena nicht sehen? ... 662
Warum kann ich nicht auf meinen Datenspeicher in Athena zugreifen? ... 662
Problembehandlung mit Kiro CLI ... 662
Kontingente ... 664
Servicekontingente ... 664

Feste Größenkontingente 669
Kontingente für Analytics-Dateien 670
Kontingente für die Größe von Speicherdateien 670
Feste Größenkontingente für Workflows 672
Chargenkontingente 676
Feste Größenkontingente für den Ready2Run-Workflow 676
API-Kontingente 679
Allgemeine API-Kontingente 680
Speicher-API-Kontingente 680
Workflow-API-Kontingente 682
Kontingente für Analytics-APIs 683
Dokumentverlauf 685
..... dcxciii

Was ist AWS HealthOmics?

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

Topics

- Wichtiger Hinweis
- HealthOmics features
- HealthOmics Konzepte
- Zugehörige Services
- Wie greife ich zu HealthOmics
- Regionen und Endpunkte für AWS HealthOmics
- Weitere Informationen

Wichtiger Hinweis

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

HealthOmics features

Primäre Anwendungsfälle für: HealthOmics

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

Die wichtigsten Vorteile von HealthOmics:

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

HealthOmics besteht aus drei Hauptkomponenten:

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

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

HealthOmics Konzepte

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

Themen

- Arbeitsabläufe
- Speicher
- Analysen

Arbeitsabläufe

Mit HealthOmics Workflows können Sie Ihre Genomdaten verarbeiten und analysieren.

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

Speicher

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

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

Analysen

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

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

Zugehörige Services

Die folgenden Dienste funktionieren mit HealthOmics.

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

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

Wie greife ich zu HealthOmics

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

- AWS Managementkonsole — Stellt eine Weboberfläche bereit, über die Sie darauf zugreifen können HealthOmics.
- AWS Command Line Interface (AWS CLI) — Stellt Befehle für eine Vielzahl von AWS Diensten bereit, darunter Windows AWS HealthOmics, MacOS und Linux, und wird unter diesen unterstützt. Weitere Informationen zur Installation von finden Sie unter AWS Command Line Interface. AWS CLI
- AWS SDKs — AWS stellt SDKs (Software Development Kits) bereit, die aus Bibliotheken und Beispielcode für verschiedene Programmiersprachen und Plattformen (einschließlich Java, Python, Ruby, .NET, iOS und Android) bestehen. SDKs Sie bieten eine bequeme Möglichkeit, sie HealthOmics programmgesteuert zu verwenden. Weitere Informationen finden Sie im AWS SDK Developer Center.
- AWS API — Sie können API-Operationen verwenden, um HealthOmics programmgesteuert darauf zuzugreifen und diese zu verwalten. Weitere Informationen finden Sie in der HealthOmics -API-Referenz.
- Kiro Power for HealthOmics — Ein kuratierter und vorkonfigurierter MCP-Server mit Steuerungsdateien und Agent-Hooks, der den Kiro-Agenten Fachwissen bei der Erstellung und Optimierung von Workflows vermittelt. HealthOmics Laden Sie Kiro Power von HealthOmics Kiro Powers herunter.
- Kiro IDE-Erweiterung für HealthOmics — Bietet Syntaxhervorhebung, Codevervollständigung und Anleitung zur Fehlerbehebung für HealthOmics Workflows sowie Engine-Kompatibilitätsprüfungen, Empfehlungen zur Leistungsoptimierung, automatische Laufanalyse mit Fehlerdiagnose und Workflow-Funktionen. import/export Laden Sie die Erweiterung von Open VSX Registry herunter.

Regionen und Endpunkte für AWS HealthOmics

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

Zusätzlich zu den AWS Regionen, die standardmäßig aktiv sind, gibt es auch Opt-in-Regionen, die aktiviert werden müssen. Weitere Informationen zur Aktivierung oder Deaktivierung einer Region findest du im Leitfaden zur Kontoverwaltung unter „Geben Sie an, welche AWS Regionen Ihr AWS Konto verwenden kann“.

Weitere Informationen

In diesen Workshops und Tutorials erfährst du mehr darüber HealthOmics :

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

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

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

AWS HealthOmics Änderung der Verfügbarkeit von Variantenspeicher und Annotationsspeicher

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

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

Themen

- Überblick über die Migrationsoptionen
- Migrationsoptionen für ETL-Logik
- Migrationsoptionen für Speicher
- Analysen
- AWS Partner
- Beispiele

Überblick über die Migrationsoptionen

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

1. Verwenden Sie die von HealthOmics -bereitgestellte Referenzimplementierung von ETL logic on GitHub.

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

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

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

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

Migrationsoptionen für ETL-Logik

Beachten Sie die folgenden Migrationsoptionen für die ETL-Logik:

1. HealthOmics stellt den Quellcode für die ETL-Pipeline für den Variantenspeicher als Referenzimplementierung bereit GitHub. Sie können diese Pipeline verwenden, um denselben ETL-Prozess für Variantendaten wie den Variantenspeicher zu starten, jedoch mit voller Kontrolle über die ETL-Logik. Weitere Informationen finden Sie unter Glue Variant ETL on GitHub.
2. Um Variantendaten zu transformieren, können Sie benutzerdefinierte Glue-ETL-Jobs schreiben oder Open-Source-HAIL- oder GLOW-Code auf EMR verwenden.

Migrationsoptionen für Speicher

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

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

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

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

Analysen

Verwenden Sie für Datenanalysen weiterhin AWS Analysedienste wie Amazon Athena, AmazonEMR, AmazonRedshift oder Amazon Quick.

AWS Partner

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

Beispiele

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

Athena DDL

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

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

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

Erstellen Sie Tabellen mit Python (ohne Athena)

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

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


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

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

    return load_catalog("s3tables", **catalog_config)


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

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


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

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

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

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


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

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

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

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

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

    return table


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

    main(bucket_arn, namespace, table_name)
```

Einrichten HealthOmics

Zum Einrichten AWS HealthOmics, melde dich für eine an AWS-Konto, erstellen Sie einen Administratorbenutzer und verwalten Sie den Zugriff für weitere Benutzer sicher.

Themen

- Melde dich an für AWS-Konto
- Erstellen Sie IAM-Berechtigungen für HealthOmics
- Stellen Sie eine Verbindung zu externen Code-Repositorys her
- Verwenden von Kiro CLI mit HealthOmics
- Verwenden von Kiro IDE mit HealthOmics

Melde dich an für AWS-Konto

Um loszulegen mit AWS, du brauchst eine AWS-Konto. Es gibt zwei Möglichkeiten, sich anzumelden AWS. Wenn Sie die vollständige Kontrolle über die Kontokonfiguration und den Speicherort Ihrer Inhalte haben möchten oder wenn Sie regulierte Workloads wie HIPAA oder FedRAMP haben, verwenden Sie Melden Sie sich an für AWS (fortgeschritten). Wenn Sie ein neuer Benutzer sind und ein Konto verwenden möchten, das Sie bereits besitzen, um sich anzumelden AWS und erstellen Sie eine AWS Umgebung mit vorkonfigurierten Standardwerten in select AWS-Regionen, verwenden Melde dich an für AWS.

Erstellen Sie IAM-Berechtigungen für HealthOmics

Zu verwenden HealthOmics, konfigurieren Sie die folgenden IAM-Berechtigungen:

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

Weitere Informationen zur Konfiguration von IAM-Berechtigungen für HealthOmics, siehe IAM-Berechtigungen für HealthOmics.

Stellen Sie eine Verbindung zu externen Code-Repositorys her

Mit AWS HealthOmics, können Sie Ihre Workflows verwalten mit Git-based Repositorys über AWS CodeConnections. HealthOmics verwendet diese Verbindung, um auf Ihre Quellcode-Repositorys zuzugreifen.

Bevor Sie mit externen Code-Repositorys arbeiten, folgen Sie den Verbindungen einrichten Anleitung zum Einstieg in die Arbeit AWS CodeConnections. Stellen Sie sicher, dass Sie die richtigen IAM-Richtlinien und -Berechtigungen für Ihre erstellt haben AWS Konto. Eine Liste der unterstützten Git-Anbieter und weitere Informationen finden Sie unter Für welche Drittanbieter kann ich Verbindungen herstellen?.

Stellen Sie eine Verbindung her

Um eine Verbindung mit Ihrem bevorzugten Repository-Anbieter herzustellen, folgen Sie den Erstellen Sie eine Verbindung Tutorial.

Verwenden von Kiro CLI mit HealthOmics

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

Das HealthOmics Tutorial zur generativen KI von Agentic bietet eine schrittweise Anleitung zur Konfiguration von Kontextdateien und zur Aktivierung von Kiro CLI zur Erstellung, Ausführung und Optimierung Ihrer AWS HealthOmics Arbeitsabläufe.

Verwenden von Kiro IDE mit HealthOmics

Zusätzlich zur Kiro CLI können Sie die Kiro-IDE mit verwenden HealthOmics:

- Kiro Power für HealthOmics — Ein kuratierter und vorgefertigter MCP-Server mit Steuerungsdateien und Agenten-Hooks, der dem Kiro-Agenten Fachwissen in folgenden Bereichen bietet HealthOmics Erstellung und Optimierung von Arbeitsabläufen.
- Kiro IDE-Erweiterung für HealthOmics — Bietet Anleitungen zur Syntaxhervorhebung, Codevervollständigung und Problembehebung für HealthOmics Workflows sowie Engine-

Kompatibilitätsprüfungen, Empfehlungen zur Leistungsoptimierung, automatische Laufanalyse mit Fehlerdiagnose und Arbeitsablauf import/export Fähigkeiten.

Erste Schritte mit HealthOmics

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

Verwenden eines Ready2Run-Workflows in der Konsole HealthOmics

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

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

Beispielaufforderungen für Kiro CLI

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

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

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

Private Workflows in HealthOmics

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

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

Arbeiten mit HealthOmics Private Workflows umfassen vier Phasen. Wählen Sie eine Registerkarte für Details.

Setting up

Sie können einen Workflow und eine optionale Konfiguration für den Internetzugang erstellen.

- Workflows erstellen und Workflow-Versionen — Ein Workflow ist Ihre Pipeline-Definition. Er definiert seine Aufgaben, Ausgaben und Konfiguration. Sie können auch Versionen eines privaten Workflows erstellen, um die Pipeline-Definition zu aktualisieren. Die Workflow-Versionierung bietet Benutzern die Möglichkeit, zu wählen, wann sie mit der Nutzung der aktualisierten Funktionen beginnen möchten. Workflow-Versionen sind unveränderlich und bieten den gleichen Grad an Datenherkunft wie Workflows.
- Konfiguration — Dies ist eine optionale Einstellung, die Sie erstellen, wenn ein Workflow einen Internetzugang benötigt.
- IAM-Berechtigungen — Bevor Sie einen Workflow ausführen, konfigurieren Sie die IAM-Berechtigungen, die Workflows benötigen. HealthOmics verwendet IAM-Servicerollen, um auf andere Dienste zuzugreifen (wie Amazon S3, Amazon ECR und CloudWatch) in Ihrem Namen, um die in Ihrem Workflow definierten Prozesse abzuschließen.

Execution

Nachdem Sie einen privaten Workflow erstellt haben, können Sie einen einzelnen Lauf oder mehrere Läufe zusammen starten.

- Starte einen Lauf — Sie können einen einzelnen Lauf eines privaten Workflows starten, den Sie mit Ihrem Eingabe-Datensatz erstellt haben. HealthOmics führt Ihre private Workflow-Definition mit jeder Aufgabe als einem einzelnen Prozess innerhalb dieses Laufs aus.

- Startet eine Batch-Ausführung — Bei einer Batchausführung werden viele Durchläufe (z. B. mehr als vier Läufe gleichzeitig) desselben Workflows zusammen in einer einzigen Einreichung ausgeführt. Anstatt Läufe einzeln zu starten, senden Sie eine Batchausführung und HealthOmics orchestriert sie.
- Die Überwachung läuft — HealthOmics sendet Logs und Metriken in Echtzeit aus, mit denen du deine Läufe überwachen kannst CloudWatch und EventBridge. Sie können Kennzahlen erfassen und verfolgen, benutzerdefinierte Dashboards erstellen und Alarme festlegen, die Sie benachrichtigen oder Maßnahmen ergreifen, wenn eine bestimmte Metrik einen von Ihnen festgelegten Schwellenwert erreicht.

Optimization and governance

Sie können die Rechenauslastung kontrollieren, Läufe priorisieren oder frühere Laufergebnisse für einen oder mehrere Ihrer Läufe wiederverwenden.

- Gruppen ausführen — Sie können einer Ausführungsgruppe private Workflows hinzufügen, um die Computernutzung zu kontrollieren. Eine Ausführungsgruppe ist eine Sammlung von Workflow-Ausführungen, die eine Reihe von Ressourcenbeschränkungen gemeinsam haben, wie z. B. die maximale Anzahl gleichzeitiger Läufe und die maximale Ausführungsdauer. Sie legen diese Grenzwerte fest, um die Rechenressourcen zu steuern, die die Ausführungsgruppe verbraucht.
- Rufen Sie das Caching auf — Sie können einen Anrufcache verwenden, um Aufgabenausgaben zu speichern und wiederzuverwenden, was zu kürzeren Laufzeiten und Kosteneinsparungen führt.
- Führen Sie die Optimierung aus — Sie können Läufe hinsichtlich der Gesamtkosten, der Gesamtlaufzeit oder einer Kombination aus beidem optimieren. HealthOmics bietet Daten und Tools, die Ihnen bei Entscheidungen zur Laufoptimierung helfen.

Collaboration

Sie können mit anderen zusammenarbeiten AWS-Konten indem Sie Ihre privaten Workflows teilen.

- Workflows teilen — Sie können Ihre privaten Workflows mit anderen teilen AWS-Konten in derselben Region. Ein Abonnentenkonto kann dann die Ausführung eines Workflows starten, dessen Eigentümer es nicht ist. Einrichtung, Ausführung und Optimierung erfolgen alle innerhalb eines Kontos, wohingegen das Teilen die Kontogrenze überschreitet.

Vollständige Anwendungsbeispiele finden Sie HealthOmics private Workflows finden Sie im Repositorium für aws-healthomics-tutorials auf GitHub oder der AWS HealthOmics umfassender Workshop in der AWS Builder Center-Werkstattkatalog.

Themen

- Private Workflows erstellen in HealthOmics
- Workflow-Versionierung in HealthOmics
- Verwenden HealthOmics läuft
- Batch läuft in HealthOmics
- HealthOmics Run-Gruppen verwenden
- Aufruf-Caching für Läufe HealthOmics
- Teilen HealthOmics Workflows
- HealthOmics Workflows mit einer VPC verbinden
- Internetzugang für VPC-connected Workflows

Private Workflows erstellen in HealthOmics

Sie können Ihren eigenen Workflow mitbringen AWS HealthOmics als Privater Arbeitsablauf und führen Sie Ihre Bioinformatik-Workloads für Forschung oder Produktion auf HealthOmics. Um dorthin zu gelangen, bereiten Sie eine Workflow-Definition vor und konfigurieren die Ressourcen, von denen sie abhängt.

- Workflow definition file: Eine Workflow-Definitionsdatei, geschrieben in WDL, Nextflow, oder CWL. Die Workflow-Definition spezifiziert die Eingaben und Ausgaben für Läufe, die den Workflow verwenden. Sie enthält auch Spezifikationen für die Ausführungen und Ausführungsaufgaben für Ihren Workflow, einschließlich Rechen- und Speicheranforderungen. Die Workflow-Definitionsdatei muss sich in befinden `.zip` formatieren. Weitere Informationen finden Sie unter Workflow-Definitionsdateien.
  - Du kannst verwenden Kiro CLI um Ihre Workflow-Definitionsdateien in WDL, Nextflow und CWL zu erstellen und zu validieren. Weitere Informationen finden Sie unter Beispiel für Eingabeaufforderungen für Kiro CLI und der HealthOmics Tutorial zur generativen KI von Agentic auf GitHub.
- Container images: Wenn Ihr Workflow Container-Images aus öffentlichen Registern oder aus Ihren eigenen Repositorys benötigt, können Sie Amazon Elastic Container Registry (Amazon ECR)

konfigurieren und AWS Identity and Access Management (IAM) -Berechtigungen für Ihren Lauf. Mit HealthOmics, können Sie Ihre eigenen privaten Container und Container-Images verwenden von Docker Hub, Amazon ECR Public, GitHub Container-Register, GitLab Container-Register, Google Artefakt- oder Container-Register, Kubernetes Container-Register, Microsoft Azure Container-Register, NVIDIA NGC, Quay, und Seqera Containers. Weitere Informationen zu unterstützten Registrierungen finden Sie unter Container-Images für private Workflows.

- (Optional) Parameter template file: Eine Parameter-Vorlagendatei, geschrieben in JSON. Erstellen Sie die Datei, um die Laufparameter zu definieren, oder HealthOmics generiert die Parametervorlage für Sie. Weitere Informationen finden Sie unter Parameter-Vorlagendateien für HealthOmics Workflows.
- (Optional) README file: Mithilfe von README-Dateien können Sie umfassende Dokumentationen, Implementierungsdetails und Diagramme direkt an Ihre Bioinformatik-Workflows und Workflow-Versionen anhängen. README-Dateien verbessern das Workflow-Management, indem sie einen zentralen Ort für wichtige Workflow-Dokumentationen bieten und so einen effektiveren Wissensaustausch zwischen Forschungsteams ermöglichen. Weitere Informationen finden Sie unter Workflow-README-Dateien.
- (Optional) Sentieon licenses: Beantragen Sie eine Sentieon Lizenz zur Nutzung der Sentieon Software in privaten Arbeitsabläufen.
- (Optional) Illumina DRAGEN: Führen Sie aus Illumina DRAGEN Softwaremodus als Aufgabe in einem privaten Workflow. Weitere Informationen finden Sie unter Illumina DRAGEN auf HealthOmics private Arbeitsabläufe.

Sie können Probleme in Ihren Workflows erkennen, bevor Sie einen Lauf starten, was Zeit spart. HealthOmics bietet zwei Ressourcen, die Ihnen bei der Suche nach Problemen in Ihrer Workflow-Definition helfen.

- Hinweise zum Arbeitsablauf: Verknüpfen Sie Ihre Workflow-Definition, um Fehler und Warnungen frühzeitig zu erkennen. HealthOmics enthält integrierte Linter, die während der Workflow-Erstellung ausgeführt werden. Sie können optional einen Linter lokal ausführen, bevor Sie den Workflow erstellen. Wenn der integrierte Linter Probleme feststellt, werden die Ergebnisse des strukturierten Lints in der `statusMessage` damit Sie sie überprüfen und beheben können.
- Unterstützte Sprachspezifikationen und Engine-Versionen: HealthOmics unterstützt Workflow-Definitionen, die in drei Workflow-Sprachen geschrieben sind. Jede Sprache hat ihre eigene Syntax, Funktionen und Versionsunterstützung.

Wenn Ihre Workflow-Definition und die Container-Images fertig sind, erstellen Sie den Workflow in der Konsole mit AWS Command Line Interface (AWS CLI), oder mit einem AWS SDK. Der Workflow beginnt in `CREATING` und wird `ACTIVE` wenn es fertig ist. Nach dem Arbeitsablauf ist `ACTIVE`, du kannst einen Lauf starten. Weitere Informationen finden Sie unter HealthOmics Workflow-Operationen.

Themen

- HealthOmics Workflow-Integration mit Git-basierten Repositorys
- Workflow-Definitionsdateien in HealthOmics
- Parametervorlagendateien für HealthOmics Workflows
- Container-Images für private Workflows
- Umgebungsvariablen für HealthOmics Workflows
- HealthOmics Workflow-README-Dateien
- Verbindung herstellen HealthOmics läuft auf einem Sentieon-Lizenzserver
- Illumina DRAGEN auf HealthOmics private Arbeitsabläufe
- Unterstützte Workflow-Sprachen für HealthOmics
- Der Arbeitsablauf zeigt an HealthOmics
- Workflow-Definitionen lokal validieren
- HealthOmics Workflow-Operationen

HealthOmics Workflow-Integration mit Git-basierten Repositorys

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

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

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

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

1. Sie möchten schnell mit einem gängigen Open-Source-Workflow wie nf-core loslegen. HealthOmicsruft automatisch alle Workflow-Definitions- und Eingabeparameter-Vorlagendateien aus dem NF-Core-Repository ab GitHub und verwendet diese Dateien, um Ihren neuen Workflow zu erstellen.
2. Sie verwenden einen öffentlichen Workflow von GitHub, und einige neue Updates sind verfügbar. Sie können ganz einfach eine neue HealthOmics Workflow-Version erstellen, indem Sie die aktualisierte Workflow-Definition GitHub als Quelle verwenden. Benutzer Ihres Workflows können zwischen dem ursprünglichen Workflow und der neuen Workflow-Version wählen, die Sie erstellt haben.
3. Ihr Team baut eine eigene Pipeline auf, die nicht öffentlich ist. Sie speichern Ihren Code in einem privaten Git-Repository und verwenden diese Workflow-Definition für Ihre HealthOmics Workflows. Das Team aktualisiert die Workflow-Definition häufig als Teil eines iterativen Workflow-Entwicklungszyklus. Sie können ganz einfach neue Workflow-Versionen nach Bedarf aus Ihrem privaten Repository erstellen.

Themen

- Unterstützte Git-basierte Repositorys
- Konfigurieren Sie Verbindungen zu externen Code-Repositorys
- Zugreifen auf selbstverwaltete Repositorys
- Kontingente im Zusammenhang mit externen Code-Repositorys
- Erforderliche IAM-Berechtigungen

Unterstützte Git-basierte Repositorys

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

- GitHub
- GitLab
- Bitbucket

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

- GitHubEnterpriseServer
- GitLabSelfManaged

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

Konfigurieren Sie Verbindungen zu externen Code-Repositorys

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

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

Eine Verbindung erstellen

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

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

Konfigurieren Sie die Autorisierung für die Verbindung

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

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

Zugreifen auf selbstverwaltete Repositorys

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

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

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

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

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

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

   d. Wählen Sie Create Host, wodurch der Host im Status PENDING erstellt wird.
   e. Um die Einrichtung abzuschließen, wählen Sie Host einrichten.
   f. Geben Sie das Personal Access Token (PAT) eines Admin-Benutzers ein und wählen Sie dann Weiter.
3. Stellen Sie die Verbindung her

   a. Wählen Sie auf der Registerkarte Verbindungen die Option Verbindungen erstellen aus.
   b. Wählen Sie als Anbietertyp die Option GitLab Selbstverwaltet aus.
   c. Geben Sie unter Verbindungseinstellungen > Verbindungsnamen eingeben die Host-URL ein, die Sie zuvor erstellt haben.
   d. Wenn auf Ihre GitLab selbstverwaltete Instanz nur über eine VPC zugegriffen werden kann, konfigurieren Sie die VPC-Details.
   e. Wählen Sie „Ausstehende Verbindung aktualisieren“. Das modale Fenster leitet Sie zur GitLab Anmeldeseite weiter.

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

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

Kontingente im Zusammenhang mit externen Code-Repositorys

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

Erforderliche IAM-Berechtigungen

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

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

Workflow-Definitionsdateien in HealthOmics

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

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

- WDL — Stellt eine spezifikationskonforme WDL-Engine bereit.
- WDL nachsichtig — Konzipiert für Workflows, die von Cromwell migriert wurden. Es unterstützt Kundenrichtlinien von Cromwell und einige nicht konforme Logiken. Details hierzu finden Sie unter Implizite Typkonvertierung in WDL (milde).

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

In der Workflow-Definition geben Sie die folgenden Informationstypen an:

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

Themen

- HealthOmics Anforderungen an die Workflow-Definition
- Verpacken und liefern Sie eine HealthOmics Workflow-Definition
- Versionsunterstützung für HealthOmics Workflow-Definitionssprachen
- Aufgabenausgaben in einer HealthOmics Workflow-Definition
- Aufgaben, Rechenleistung und Ressourcen
- Besonderheiten der WDL-Workflow-Definition
- Besonderheiten der Nextflow-Workflow-Definition
- Besonderheiten der CWL-Workflow-Definition
- Beispiele für Workflow-Definitionen

HealthOmics Anforderungen an die Workflow-Definition

Die HealthOmics Workflow-Definitionsdateien müssen die folgenden Anforderungen erfüllen:

- Aufgaben müssen definiert werden input/output Parameter, Amazon ECR-Container-Repositorys und Laufzeitspezifikationen wie Speicher- oder CPU-Zuweisung.
- Stellen Sie sicher, dass Ihre IAM-Rollen über die erforderlichen Berechtigungen verfügen.
  - Ihr Workflow hat Zugriff auf Eingabedaten von AWS Ressourcen wie Amazon S3.
  - Ihr Workflow hat bei Bedarf Zugriff auf externe Repository-Services.
- Deklarieren Sie die Ausgabedateien in der Workflow-Definition. Um Zwischenlaufdateien an den Ausgabespeicherort zu kopieren, deklarieren Sie sie als Workflow-Ausgaben.

- Die Eingabe- und Ausgabespeicherorte müssen sich in derselben Region wie der Workflow befinden.
- HealthOmics Die Eingaben für den Speicher-Workflow müssen sich in befinden `ACTIVE` Status. HealthOmics importiert keine Eingaben mit einem `ARCHIVED` Status, wodurch der Workflow fehlschlägt. Informationen zu Amazon S3-Objekteingaben finden Sie unter HealthOmics Eingaben ausführen.
- EIN main Der Speicherort des Workflows ist optional, wenn Ihr ZIP-Archiv entweder eine einzelne Workflow-Definition oder eine Datei mit dem Namen „main“ enthält.
  - Beispielpfad: `workflow-definition/main-file.wdl`
- Bevor Sie einen Workflow von Amazon S3 oder Ihrem lokalen Laufwerk aus erstellen, erstellen Sie ein ZIP-Archiv der Workflow-Definitionsdateien und aller Abhängigkeiten, wie z. B. Unterworkflows.
- Wir empfehlen, Amazon ECR-Container im Workflow als Eingabeparameter für die Validierung der Amazon ECR-Berechtigungen zu deklarieren.

Zusätzliche Überlegungen zu Nextflow:

- /bin

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

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

- includeConfig

  Nextflow-based Workflow-Definitionen können nextflow.config-Dateien enthalten, mit deren Hilfe Parameterdefinitionen oder Prozessressourcenprofile abstrahiert werden können. Um die Entwicklung und Ausführung von Nextflow-Pipelines in mehreren Umgebungen zu unterstützen, verwenden Sie eine HealthOmics-specific Konfiguration, die Sie mithilfe der IncludeConfig-Direktive

zur globalen Konfiguration hinzufügen. Um die Portabilität aufrechtzuerhalten, konfigurieren Sie den Workflow so, dass er die Datei nur einbezieht, wenn er ausgeführt wird HealthOmics mithilfe des folgenden Codes:

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

- Reports

  HealthOmics unterstützt den Nextflow-Ausführungsbericht, die Zeitleiste, die Ablaufverfolgung und die DAG-Berichte. Konfiguriere jeden Bericht so, dass seine Ausgabe unter `/mnt/workflow/output/` so dass HealthOmics exportiert es an den Amazon S3-Ausgabespeicherort Ihres Laufs. Weitere Informationen finden Sie unter Generieren Sie Nextflow-Ausführungsberichte.
- Automatische Syntaxvalidierung — Für WDL und Nextflow HealthOmics führt während der Workflow-Erstellung integrierte Linters aus. Weitere Informationen finden Sie unter Der Arbeitsablauf zeigt an HealthOmics.

Weitere Informationen zur lokalen Validierung einer Definition vor dem Erstellen des Workflows finden Sie unter Workflow-Definitionen lokal validieren.

Zusätzliche Überlegungen zu CWL:

- Container image uri interpolation

  HealthOmics ermöglicht die DockerPull-Eigenschaft des DockerRequirement um ein Inline-Javascript-Ausdruck zu sein. Beispiel:

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

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

  Javascript-Ausdrücke müssen `strict mode` konform.
- Operation process

HealthOmics unterstützt keine CWL-Betriebsprozesse.

Verpacken und liefern Sie eine HealthOmics Workflow-Definition

Wenn Sie einen Workflow oder eine Workflow-Version erstellen, liefern Sie die Definition aus genau einer Quelle: einem ZIP-Archiv, das Sie mit der Anfrage senden, einem ZIP-Archiv in Amazon S3 oder einem Git-based Repositorium. Eine Anfrage, die mehr als eine Definitionsquelle enthält, schlägt fehl. Die Parameter in diesem Thema gelten sowohl für `CreateWorkflow` und `CreateWorkflowVersion` Operationen. Die vollständigen Verfahren finden Sie unter Erstellen Sie einen privaten Workflow und Eine Workflow-Version erstellen.

> **Note**
>
> HealthOmics entfernt einige Workflow-Definitionen, wenn Sie einen Workflow erstellen. Informationen darüber, welche Engines verwendet werden, ob Ergebnisse den Workflow blockieren und wie Sie sie lesen können, finden Sie unter Der Arbeitsablauf zeigt an HealthOmics.

Wählen Sie eine Workflow-Definitionsquelle

Verwenden Sie diese Quelle	API-Parameter	Maximale Größe	AWS CLI Fahne
Ein ZIP-Archiv auf Ihrem lokalen Computer.	`definitionZip`	4,4 MB	`--definition-zip fileb://my-definition.zip`
Ein ZIP-Archiv in einem Amazon S3-Bucket. Der Bucket muss sich im selben befinden AWS Region wie der Workflow.	`definitionUri`	100 MB	`--definition-uri s3://amzn-s3-demo-bucket1/my-definition.zip`

Verwenden Sie diese Quelle	API-Parameter	Maximale Größe	AWS CLI Fahne
Ein unterstützter Git-based Repositorium, über ein AWS CodeConnections Verbindung.	`definitionRepository`	1 GiB-Repository, 100 MiB pro Datei	`--definition-repository` mit einem JSON-Wert

Weitere Überlegungen:

- Diese Größenkontingente können nicht erhöht werden. Weitere Informationen finden Sie unter HealthOmics feste Größenkontingente für Workflows.
- Wenn Ihr Konto nicht Eigentümer des Amazon S3-Buckets ist, der die Definition enthält, geben Sie die des Bucket-Besitzers an AWS-Konto ID ein `workflowBucketOwnerId` (`--workflow-bucket-owner-id`).
- Mit `definitionRepository`, HealthOmics liest die Definitionsdateien aus dem Repository und `main`, `parameterTemplatePath`, und `readmePath` sind Pfade innerhalb des Repositorys und nicht innerhalb eines Archivs. Für weitere Informationen über `definitionRepository` Felder finden Sie unter Verwenden Sie die Workflow-Definition aus einem Git-based Endlager. Weitere Informationen zum Verbindungsaufbau und zu den unterstützten Anbietern finden Sie unter HealthOmics Workflow-Integration mit Git-basierten Repositorys.

ZIP-Archiv-Layout und die Hauptdefinitionsdatei

Schließen Sie alle Dateien ein, die die Hauptdefinition importiert, z. B. Unterworkflows, `nextflow.config`, und beliebig `bin` Skripte.

Eine Definitionsdatei ist eine Datei, deren Erweiterung einer unterstützten Engine entspricht, z. B. `.wdl`, `.nf`, oder `.cwl`. Wenn Sie weglassen `main`, HealthOmics wählt es aus den Definitionsdateien aus, die dem Stammverzeichnis des Archivs am nächsten liegen. Es ignoriert versteckte Dateien und Ordner und es ignoriert Definitionsdateien, die sich tiefer im Archiv befinden. Für die automatische Auswahl ist auch eine einzelne Engine erforderlich, entweder weil das Archiv die Definitionsdateien einer Engine enthält oder weil Sie Folgendes festlegen `engine`. Von den Dateien auf der nächsten Ebene:

- Wenn es genau eine Definitionsdatei gibt, HealthOmics benutzt sie, wie auch immer sie heißt.

- Wenn es mehrere gibt, HealthOmics verwendet den genannten `main.wdl`, `main.nf`, oder `main.cwl`.

In zwei Fällen HealthOmics kann das Archiv nicht auflösen. Die Anfrage selbst ist erfolgreich und der Workflow oder die Workflow-Version beginnt in `CREATING`, wechselt dann aber zu `FAILED`, und die Statusmeldung gibt den Grund an:

- `Zip file contains multiple workflow definition types` — das Archiv enthält Definitionsdateien für mehr als eine Engine, die Sie nicht gesetzt haben `engine`. Dies gilt unabhängig davon, ob Sie Folgendes einstellen `main`. Einstellen `engine`, oder entfernen Sie die Dateien der anderen Engine. HealthOmics meldet diesen Grund zuerst, wenn beide Fälle zutreffen.
- `Zip file contains multiple top-level workflow definition files` — du hast es weggelassen `main`, mehrere Definitionsdateien befinden sich auf der nächsten Ebene, und nicht genau eine von ihnen ist benannt `main.wdl`, `main.nf`, oder `main.cwl`. Einstellen `main` auf den Pfad desjenigen, den du willst.

Um die Hauptdefinitionsdatei selbst auszuwählen, setzen Sie `main` (`--main`) zu seinem Pfad, relativ zum Stammverzeichnis des Archivs. Der Pfad kann bis zu 128 Zeichen lang sein.

Zum Beispiel für ein Archiv, dessen Hauptdefinitionsdatei `workflow-definition/main-file.wdl`:

```
--main workflow-definition/main-file.wdl
```

Einstellen `engine` (`--engine`), um die Engine selbst zu benennen, entweder um ein Archiv aufzulösen, das mehrere Definitionsdateien einer Engine enthält, oder um die WDL Lenient Engine zu verwenden.

Stellen Sie sich einen Workflow-Ordner auf Ihrem lokalen Computer mit dem folgenden Layout vor:

```
my-workflow/
    main.wdl
    tasks/align.wdl
    tasks/call-variants.wdl
```

Erstellen Sie das Archiv in diesem Ordner, sodass `main.wdl` und der `tasks` Der Ordner befindet sich im Stammverzeichnis des Archivs. Mit dem folgenden Befehl werden auch versteckte Dateien und Ordner hinzugefügt, z. B. `.git` Ordner in einem geklonten Repository. Diese werden auf die

Definitionsgröße angerechnet. Schließen Sie also diejenigen aus, die Sie nicht benötigen, z. B. mit `-x '.*' -x '*/.*'`:

```
cd my-workflow
zip -r ../my-definition.zip .
cd ..
```

`main.wdl` ist jetzt die einzige Definitionsdatei im Stammverzeichnis des Archivs, also kannst du weglassen `main`:

```
main.wdl
tasks/align.wdl
tasks/call-variants.wdl
```

Senden Sie das Archiv online oder über Amazon S3

`definitionZip` ist ein binärer Parameter. Wenn Sie das verwenden AWS CLI, übergeben Sie das Archiv mit dem `fileb://` Präfix und das AWS CLI liest die Datei und codiert sie für Sie. Das AWS SDKs übernehmen die Kodierung für Sie. Wenn Sie die API direkt über HTTP aufrufen, senden Sie das Archiv als Base64-kodierte Zeichenfolge.

Die Parametervorlage in den folgenden Beispielen ist optional. Wenn Sie sie weglassen, HealthOmics leitet die Ausführungsparameter aus Ihrer Workflow-Definition ab. Weitere Informationen finden Sie unter Parametervorlagendateien für HealthOmics Workflows.

Folgendes AWS CLI Beispiel erstellt einen Workflow, indem das ZIP-Archiv mit der Anfrage gesendet wird. Ersetzen Sie jeden Wert kursiv:

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

Für ein Archiv, das größer ist als `definitionZip` einschränken in Wählen Sie eine Workflow-Definitionsquelle, laden Sie es in einen Amazon S3-Bucket in derselben Region wie der Workflow hoch und verwenden Sie `definitionUri` stattdessen. Ersetzen Sie jeden Wert kursiv und ersetzen Sie den Bucket durch Ihren eigenen:

```
aws s3 cp my-definition.zip s3://amzn-s3-demo-bucket1/my-definition.zip
```

```
aws omics create-workflow \
  --name my-workflow-name \
  --definition-uri s3://amzn-s3-demo-bucket1/my-definition.zip \
  --parameter-template file://my-parameter-template.json
```

Stellen Sie den Standard-Run-Speicher ein

Wir empfehlen für die meisten Workflows dynamischen Laufspeicher. Wählen Sie in einem der folgenden Fälle den statischen Run-Speicher:

- Der Workflow verwendet mehr als 9,6 TiB an Daten, und seine Aufgaben profitieren von einer hohen Dauerhaftigkeit I/O.
- Mehr als 100 verstreute Aufgaben werden gleichzeitig aus dem gemeinsam genutzten Laufspeicher gelesen.

Weitere Informationen zu den einzelnen Laufspeichertypen, einschließlich ihrer Durchsatzeigenschaften und zur Größe des statischen Laufspeichers, finden Sie unter Speichertypen in HealthOmics Workflows ausführen und Führen Sie die Optimierung für ein privates aus HealthOmics Arbeitsablauf.

Das `storageType` (`--storage-type`) und `storageCapacity` (`--storage-capacity`) - Parameter legen die Standardwerte für Läufe fest, die den Workflow verwenden. Sie können beide zur Laufzeit überschreiben, wie unter beschrieben Starte einen Lauf in HealthOmics. Kombiniere sie wie folgt:

- `storageType` von `DYNAMIC` — HealthOmics skaliert den Laufspeicher je nach Auslastung des Dateisystems nach oben und unten. Nicht einstellen `storageCapacity`. Wenn Sie dies tun, schlägt die Anfrage fehl: `CreateWorkflow` Berichte `You cannot set the storage capacity for workflows that use DYNAMIC run storage type`, und `CreateWorkflowVersion` Berichte `You cannot set the storage capacity for workflow version that use DYNAMIC run storage type`.
- `storageType` von `STATIC` — HealthOmics weist eine feste Menge an Run-Speicher zu, also auch gesetzt `storageCapacity`, in GiB. Wenn weder der Workflow noch der Lauf eine Kapazität angeben, verbraucht der Lauf 1.200 GiB. Das Maximum ist ein Limit pro Konto. HealthOmics stellt statischen Laufspeicher nur als 1.200 GiB oder als Vielfaches von 2.400 GiB bereit und rundet jeden anderen Wert auf die nächste unterstützte Größe auf. Zum Beispiel HealthOmics stellt 1.200 GiB für eine Kapazität unter 1.200 GiB bereit.

Im folgenden Beispiel wird der Standard-Run-Speicher auf 1.200 GiB statischen Speicher festgelegt:

```
aws omics create-workflow \
  --name my-workflow-name \
  --definition-zip fileb://my-definition.zip \
  --storage-type STATIC \
  --storage-capacity 1200
```

Versionsunterstützung für HealthOmics Workflow-Definitionssprachen

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

Themen

- Unterstützung der WDL-Version
- Unterstützung der CWL-Version
- Unterstützung der Nextflow-Version

Unterstützung der WDL-Version

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

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

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

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

Unterstützung der CWL-Version

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

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

Unterstützung der Nextflow-Version

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

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

Unterstützte Versionen

HealthOmics unterstützt die folgenden Nextflow-Versionen:

- Nextflow v22.04.0 DSL 1 und DSL 2
- Nextflow v23.10.0 DSL 2 (Standard)
- Nextflow v24.10.8 DSL 2
- Nextflow v25.10.0 DSL 2
- Nextflow v26.04.0 DSL 2

> **Note**
>
> Nextflow v26.04.0 verwendet standardmäßig den strikten Syntaxparser (v2). Um den Legacy-Parser zu verwenden, setzen `engineSettings.syntaxVersion` Sie ihn beim Starten eines Laufs auf. `v1` Unterstützt für Nextflow v25.10.0 und früher HealthOmics keinen strikten Syntaxmodus und der einzig zulässige Wert ist. `v1`
> HealthOmics Führt separat den integrierten strikten Nextflow-Linter während der Workflow-Erstellung aus. Der Linter gilt für alle DSL2-Versionen (v22.04, v23.10, v24.10, v25.10 und v26.04). DSL1-Workflows sind nicht überlastet. Die Ergebnisse von Lint werden im `statusMessage` Feld der Antwort angezeigt. `GetWorkflow` Weitere Informationen finden Sie unter Der Arbeitsablauf zeigt an HealthOmics.

Um Ihren Workflow auf die neueste unterstützte Version (v26.04.0) zu migrieren, folgen Sie der Nextflow-Upgrade-Anleitung.

Bei der Migration zu Nextflow v24, v25 oder v26 gibt es einige wichtige Änderungen. Folgen Sie der Nextflow-Migrationsanleitung. https://www.nextflow.io/docs/latest/migrations/index.html

Erkennen und verarbeiten Sie Nextflow-Versionen

HealthOmics erkennt die von Ihnen angegebene DSL-Version und Nextflow-Version. Basierend auf diesen Eingaben wird automatisch die beste Nextflow-Version für die Ausführung bestimmt.

DSL-Version

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

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

- Wenn Sie DSL 1 angeben, HealthOmics wird Nextflow v22.04 DSL1 ausgeführt (die einzige unterstützte Version, die DSL 1 ausführt).
- Wenn Sie keine DSL-Version angeben oder die DSL-Informationen aus irgendeinem Grund nicht analysieren HealthOmics können (z. B. aufgrund von Syntaxfehlern in Ihrer Workflow-Definitionsdatei), wird standardmäßig DSL 2 verwendet und Nextflow v23.10.0 ausgeführt. HealthOmics
- Informationen zum Upgrade Ihres Workflows von DSL 1 auf DSL 2, um die neuesten Versionen und Softwarefunktionen von Nextflow nutzen zu können, finden Sie unter Migration von DSL 1. https://nextflow.io/docs/latest/dsl1.html

Nextflow-Versionen

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

Sie können eine Nextflow-Version oder eine Reihe von Versionen mithilfe der folgenden Syntax angeben:

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

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

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

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

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

HealthOmics verarbeitet die Nextflow-Versionsinformationen wie folgt:

- Wenn Sie verwenden=, um eine genaue Version anzugeben, die HealthOmics unterstützt, HealthOmics verwendet diese Version.
- Wenn Sie verwenden!, um eine genaue Version oder einen Bereich von Versionen anzugeben, die nicht unterstützt werden, HealthOmics löst dies eine Ausnahme aus und die Ausführung schlägt fehl. Erwägen Sie, diese Option zu verwenden, wenn Sie bei Versionsanfragen strikt vorgehen möchten und schnell scheitern möchten, wenn die Anforderung nicht unterstützte Versionen enthält.
- Wenn Sie einen Versionsbereich angeben, wird die Version mit der höchsten Präferenz in diesem Bereich HealthOmics verwendet. Die Präferenzreihenfolge von der höchsten zur niedrigsten ist v23.10.0, v22.04.0, v24.10.8, v25.10.0 und v26.04.0. Beispiel:
  - Wenn der Bereich HealthOmics v25.10.0 und v26.04.0 abdeckt, wählen Sie v25.10.0.
  - Wenn der Bereich v24.10.8 und v25.10.0 abdeckt, wählen Sie v24.10.8. HealthOmics
- Wenn es keine angeforderte Version gibt oder wenn die angeforderten Versionen nicht gültig sind oder aus irgendeinem Grund nicht analysiert werden können:
  - Wenn Sie DSL 1 angegeben haben, HealthOmics wird Nextflow v22.04 ausgeführt.
  - Andernfalls wird Nextflow v23.10.0 ausgeführt HealthOmics .

Sie können die automatisch ausgewählte Nextflow-Version überschreiben, indem Sie in der Anfrage eine Einstellung vornehmen. `engineSettings.engineVersion` StartRun Dadurch wird der Lauf an eine bestimmte Nextflow-Version gebunden, unabhängig von der in der Workflow-Datei angegebenen Version. `nextflow.config` Weitere Informationen finden Sie unter Geben Sie die Nextflow-Engine-Einstellungen an.

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

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

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

Aufgabenausgaben in einer HealthOmics Workflow-Definition

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

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

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

Themen

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

Aufgabenausgaben für WDL

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

HealthOmics

Themen

- Aufgabenausgabe für STDOUT
- Aufgabenausgabe für STDERR

- Ausgabe der Aufgabe in eine Datei
- Ausgabe der Aufgabe in ein Array von Dateien

Aufgabenausgabe für STDOUT

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

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

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

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

    output {
        File stdout_file = SayHello.stdout_file
    }
}

task SayHello {
    input {
        String message
        String container
    }

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

```
    >>>

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

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

Aufgabenausgabe für STDERR

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

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

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

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

    output {
        File stderr_file = SayHello.stderr_file
    }
}

task SayHello {
```

```
    input {
        String message
        String container
    }

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

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

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

Ausgabe der Aufgabe in eine Datei

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

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

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

    output {
        File message_file = SayHello.message_file
```

```
        File info_file = SayHello.info_file
    }
}

task SayHello {
    input {
        String message
        String container
    }

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

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

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

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

Ausgabe der Aufgabe in ein Array von Dateien

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

```
version 1.0
 workflow HelloArray {
    input {
        Array[String] names = ["World", "Friend", "Developer"]
```

```
        String ubuntu_container = "123456789012.dkr.ecr.us-east-1.amazonaws.com/
dockerhub/library/ubuntu:20.04"
    }

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

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

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

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

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

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

Task-Ausgaben für Nextflow

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

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

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

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

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

Aufgabenausgaben für CWL

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

Themen

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

Aufgabenausgabe für STDOUT

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

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

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

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

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

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

Aufgabenausgabe für STDERR

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

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

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

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

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

Ausgabe der Aufgabe in eine Datei

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

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

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

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

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

Ausgabe der Aufgabe in ein Array von Dateien

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

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

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

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

Aufgaben, Rechenleistung und Ressourcen

Wenn Sie Aufgaben definieren in einem HealthOmics In einem Workflow geben Sie die Rechen-, Speicher- und Containerressourcen an, die jede Aufgabe benötigt. HealthOmics weist den entsprechenden Instanztyp für jede Aufgabe auf der Grundlage der von Ihnen angeforderten Ressourcen zu.

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

- CPU und Arbeitsspeicher — Die Anzahl der vCPUs und die Speichermenge für standardmäßige, rechenoptimierte oder speicheroptimierte Instances. Siehe Aufgabenressourcen in einem HealthOmics Workflow-Definition.
- Container-Image — Das Amazon ECR-Container-Image für die Aufgabe. Siehe Container-Images für private Workflows.
- GPU-Beschleuniger — Optional ein GPU-Beschleunigertyp zur Zuweisung einer Instanz für beschleunigtes Rechnen (G4, G5, G6 oder G6e) für Aufgaben, die von der GPU-Verarbeitung profitieren. Siehe Taskbeschleuniger in einem HealthOmics Workflow-Definition.

- Erweiterte Ressourcenkonfiguration — Optional eine geordnete Liste von Beschleunigertypen (GPU), die zur Laufzeit gesucht und ausgeführt werden sollen. HealthOmics probiert jedes Beschleunigerprofil der Reihe nach aus, bis eines erfolgreich ist. Dadurch wird ein Fallback von einem Beschleunigertyp auf einen anderen oder von Beschleunigern auf die CPU ermöglicht. Siehe Erweiterte Ressourcenkonfiguration.

> **Note**
>
> HealthOmics passt die Instanztypen an die von Ihnen angegebenen Rechen- und Speicheranforderungen an. Wenn Sie keine Rechen- oder Speicheranforderungen angeben, HealthOmics standardmäßig 1 vCPU und 1 GiB Arbeitsspeicher für eine CPU-Instanz.

Themen

- Aufgabenressourcen in einem HealthOmics Workflow-Definition
- Rechen- und Speicheranforderungen für HealthOmics Aufgaben
- Taskbeschleuniger in einem HealthOmics Workflow-Definition
- Erweiterte Ressourcenkonfiguration

Aufgabenressourcen in einem HealthOmics Workflow-Definition

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

- Das Container-Image für die Aufgabe. Weitere Informationen finden Sie unter Container-Images für private Workflows.
- Die Anzahl der CPUs und der Arbeitsspeicher, die für die Aufgabe benötigt werden. Weitere Informationen finden Sie unter Rechen- und Speicheranforderungen für HealthOmics Aufgaben.
- Der optionale temporäre Speicher pro Aufgabe, den die Aufgabe verwenden kann `/tmp` wenn für den Lauf aktiviert. Weitere Informationen finden Sie unter Kurzlebiger Speicher für Workflow-Aufgaben HealthOmics.

HealthOmics bietet Laufspeicher, auf den alle Aufgaben im Lauf zugreifen können. Weitere Informationen finden Sie unter Speichertypen in HealthOmics Workflows ausführen.

WDL

```
task my_task {
   runtime {
      container: "<aws-account-id>.dkr.ecr.<aws-region>.amazonaws.com/<image-name>"
      cpu: 2
      memory: "4 GB"
      disks: "100 GiB"                # optional
   }
   ...
}
```

Für einen WDL-Workflow HealthOmics versucht bis zu zwei Wiederholungen für eine Aufgabe, die aufgrund von Servicefehlern fehlschlägt (die API-Anforderung gibt einen 5XX-HTTP-Statuscode zurück). Weitere Hinweise zu Aufgabenwiederholungen finden Sie unter Die Aufgabe wird erneut versucht.

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

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

NextFlow

```
process my_task {
   container "<aws-account-id>.dkr.ecr.<aws-region>.amazonaws.com/<image-name>"
   cpus 2
   memory "4 GiB"
   disk '100 GB'                       // optional
   ...
}
```

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
        tmpdirMin: 102400 # 100 gibibytes expressed in mebibytes
```

Rechen- und Speicheranforderungen für HealthOmics Aufgaben

HealthOmics führt Ihre privaten Workflow-Aufgaben in einer Omics-Instanz aus. HealthOmics bietet eine Vielzahl von Instanztypen für verschiedene Aufgabentypen. Jeder Instanztyp hat eine feste Speicher- und vCPU-Konfiguration (und eine feste GPU-Konfiguration für Instanztypen mit beschleunigter Datenverarbeitung). Die Kosten für die Verwendung einer Omics-Instanz variieren je nach Instanztyp. Einzelheiten finden Sie auf der Seite mit den HealthOmics Preisen.

Für Aufgaben in einem Workflow geben Sie den erforderlichen Arbeitsspeicher und die vCPUs in der Workflow-Definitionsdatei an. Wenn eine Workflow-Aufgabe ausgeführt wird, wird die kleinste HealthOmics Omics-Instanz zugewiesen, die den angeforderten Speicher und die vCPUs beherbergt. Wenn eine Aufgabe beispielsweise 64 GiB Arbeitsspeicher und 8 vCPUs benötigt, wählt aus. HealthOmics `omics.r.2xlarge`

Wir empfehlen Ihnen, die Instance-Typen zu überprüfen und die angeforderten vCPUs und die Speichergröße so einzustellen, dass sie der Instance entsprechen, die Ihren Anforderungen am besten entspricht. Der Aufgabencontainer verwendet die Anzahl der vCPUs und die Speichergröße, die Sie in Ihrer Workflow-Definitionsdatei angeben, auch wenn der Instanztyp über zusätzliche vCPUs und Arbeitsspeicher verfügt.

Die folgende Liste enthält zusätzliche Informationen zu vCPU und Speicherzuweisung:

- Bei der Zuweisung von Container-Ressourcen handelt es sich um feste Grenzwerte. Wenn einer Aufgabe der Arbeitsspeicher ausgeht oder versucht wird, zusätzliche vCPUs zu verwenden, generiert die Aufgabe ein Fehlerprotokoll und wird beendet.
- Wenn Sie keine Rechen- oder Speicheranforderungen angeben, HealthOmics wählt eine Konfiguration mit 1 vCPU omics.c.large und 1 GiB Arbeitsspeicher aus und verwendet diese standardmäßig.
- Die Mindestkonfiguration, die Sie anfordern können, ist 1 vCPU und 1 GiB Arbeitsspeicher.
- Wenn Sie vCPUs, Arbeitsspeicher oder GPUs angeben, die die unterstützten Instanztypen überschreiten, wird eine Fehlermeldung ausgegeben und der Workflow HealthOmics schlägt die Validierungen fehl

- Wenn Sie Brucheinheiten angeben, wird auf die nächste Ganzzahl aufgerundet HealthOmics .
- HealthOmics reserviert eine kleine Speichermenge (5%) für Verwaltungs- und Protokollierungsagenten, sodass der Anwendung in der Aufgabe möglicherweise nicht immer der gesamte Speicherzuweisung zur Verfügung steht.
- HealthOmics passt die Instanztypen an die von Ihnen angegebenen Rechen- und Speicheranforderungen an und verwendet möglicherweise eine Mischung aus Hardwaregenerationen. Aus diesem Grund kann es für dieselbe Aufgabe zu geringfügigen Abweichungen bei den Aufgabenlaufzeiten kommen.

Diese Themen enthalten Einzelheiten zu den HealthOmics unterstützten Instanztypen.

Themen

- Standard-Instance-Typen
- Compute-optimized Instanzen
- Memory-optimized Instanzen
- Accelerated-computing Instanzen

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

Standard-Instance-Typen

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

HealthOmics unterstützt die 32xlarge- und 48xlarge-Instances in den folgenden Regionen: USA West (Oregon), USA Ost (Nord-Virginia), Asien-Pazifik (Seoul), Asien-Pazifik (Tokio) und USA Ost (Ohio).

Instance	Anzahl der vCPUs	Arbeitsspeicher
omics.m.large	2	8 GiB
omics.m.xlarge	4	16 GiB
omics.m.2xlarge	8	32 GiB
omics.m.4xlarge	16	64 GiB
omics.m.8xlarge	32	128 GiB
omics.m.12xlarge	48	192 GiB
omics.m.16xlarge	64	256 GiB
omics.m.24xlarge	96	384 GiB
omics.m.32xlarge	128	512 GiB
omics.m.48xlarge	192	768 GiB

Compute-optimized Instanzen

Bei rechenoptimierten Instanztypen verfügen die Konfigurationen über mehr Rechenleistung und weniger Arbeitsspeicher.

HealthOmics unterstützt die 32xlarge- und 48xlarge-Instances in den folgenden Regionen: USA West (Oregon), USA Ost (Nord-Virginia), Asien-Pazifik (Seoul), Asien-Pazifik (Tokio) und USA Ost (Ohio).

Instance	Anzahl der vCPUs	Arbeitsspeicher
omics.c.large	2	4 GiB
omics.c.xlarge	4	8 GiB
omics.c.2xlarge	8	16 GiB
omics.c.4xlarge	16	32 GiB

Instance	Anzahl der vCPUs	Arbeitsspeicher
omics.c.8xlarge	32	64 GiB
omics.c.12xlarge	48	96 GiB
Omics.c.16x groß	64	128 GiB
omics.c.24xlarge	96	192 GiB
omics.c.32xlarge	128	256 GiB
omics.c.48xlarge	192	384 GiB

Memory-optimized Instanzen

Bei speicheroptimierten Instanztypen haben die Konfigurationen weniger Rechenleistung und mehr Speicher.

HealthOmics unterstützt die 32xlarge- und 48xlarge-Instances in den folgenden Regionen: USA West (Oregon), USA Ost (Nord-Virginia), Asien-Pazifik (Seoul), Asien-Pazifik (Tokio) und USA Ost (Ohio).

Instance	Anzahl der vCPUs	Arbeitsspeicher
omics.r.large	2	16 GiB
omics.r.xlarge	4	32 GiB
omics.r.2xgroß	8	64 GiB
omics.r.4x groß	16	128 GiB
omics.r.8 x groß	32	256 GiB
omics.r. 12 x groß	48	384 GiB
omics.r. 16x groß	64	512 GiB
omics.r. 24x groß	96	768 GiB

Instance	Anzahl der vCPUs	Arbeitsspeicher
omics.r. 32 x groß	128	1024 GiB
omics.r. 48x groß	192	1536 GiB

Accelerated-computing Instanzen

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

Wenn Sie einen Task Accelerator angeben, der mehrere Instanztypen unterstützt, HealthOmics wählen Sie den Instanztyp je nach Verfügbarkeit aus. Wenn mehr als ein Instanztyp verfügbar ist, wird HealthOmics der kostengünstigeren Instanz der Vorzug gegeben. Die Ausnahme bildet der nvidia-t4-a10g-l4 Task Accelerator, der der in Ihrer Region verfügbaren Instance der neuesten Generation den Vorzug gibt.

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

Themen

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

G6- und G6e-Instance-Typen

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

HealthOmics unterstützt die G6- und G6e-Instances in diesen Regionen: USA West (Oregon), USA Ost (Nord-Virginia), Asien-Pazifik (Seoul), Asien-Pazifik (Tokio) und USA Ost (Ohio).

Instance	Anzahl der vCPUs	Arbeitsspeicher	Anzahl der GPUs	GPU-Arbeitsspeicher	
omics.g6.xlarge	4	16 GiB	1	24 GiB	
omics.g6.2xlarge	8	32 GiB	1	24 GiB	
omics.g6.4xgroß	16	64 GiB	1	24 GiB	
omics.g6.8xlarge	32	128 GiB	1	24 GiB	
omics.g6.12xlarge	48	192 GiB	4	96 GiB	
omics.g 6, 16 x groß	64	256 GiB	1	24 GiB	
omics.g6.24xlarge	96	384 GiB	4	96 GiB	

Alle omics.g6e-Instanzen verwenden Nvidia L40s-GPUs.

Instance	Anzahl der vCPUs	Arbeitsspeicher	Anzahl der GPUs	GPU-Arbeitsspeicher	
omics.g6e.xlarge	4	32 GiB	1	48 GiB	
omics.g6e.2xlarge	8	64 GiB	1	48 GiB	
omics.g6e.4xlarge	16	128 GiB	1	48 GiB	

Instance	Anzahl der vCPUs	Arbeitsspeicher	Anzahl der GPUs	GPU-Arbeitsspeicher	
omics.g6e.8xlarge	32	256 GiB	1	48 GiB	
omics.g6e.12xlarge	48	384 GiB	4	192 GiB	
omics.g6e.16xlarge	64	512 GiB	1	48 GiB	
omics.g6e.24xlarge	96	768 GiB	4	192 GiB	

G4- und G5-Instanzen

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

Alle omics.g5-Instanzen verwenden Nvidia Tesla A10G-GPUs.

Instance	Anzahl der vCPUs	Arbeitsspeicher	Anzahl der GPUs	GPU-Arbeitsspeicher	
omics.g5.xlarge	4	16 GiB	1	24 GiB	
omics.g5.2xlarge	8	32 GiB	1	24 GiB	
omics.g5.4xgroß	16	64 GiB	1	24 GiB	
omics.g5.8xlarge	32	128 GiB	1	24 GiB	
omics.g5.12xlarge	48	192 GiB	4	96 GiB	

Instance	Anzahl der vCPUs	Arbeitsspeicher	Anzahl der GPUs	GPU-Arbeitsspeicher	
omics.g 5.16x groß	64	256 GiB	1	24 GiB	
omics.g5. 24xlarge	96	384 GiB	4	96 GiB	

Alle omics.g4dn-Instanzen verwenden Nvidia Tesla T4-GPUs.

Instance	Anzahl der vCPUs	Arbeitsspeicher	Anzahl der GPUs	GPU-Arbeitsspeicher	
omics.g4dn.xlarge	4	16 GiB	1	16 GiB	
omics.g4dn.2xlarge	8	32 GiB	1	16 GiB	
omics.g4dn.4xlarge	16	64 GiB	1	16 GiB	
omics.g4dn.8xlarge	32	128 GiB	1	16 GiB	
omics.g4dn.12xlarge	48	192 GiB	4	64 GiB	
omics.g4dn.16xlarge	64	256 GiB	1	16 GiB	

Taskbeschleuniger in einem HealthOmics Workflow-Definition

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

Beschleuniger-Spezifikation	Healthomics-Instanztypen				
nvidia-Tesla-T4	G4				
nvidia-Tesla-T4-a10g	G4 und G5				
nvidia-Tesla-a10g	G5				
nvidia-T4-a10g-l4	G4, G5 und G6				
nvidia-l4-a10g	G5 und G6				
nvidia-l4	G6				
nvidia-l40s	G6e				

Wenn Sie einen Beschleunigertyp angeben, der mehrere Instanztypen unterstützt, HealthOmics wählt den Instanztyp auf der Grundlage der verfügbaren Kapazität aus. Wenn beide Instanztypen verfügbar sind, HealthOmics bevorzugt die kostengünstigere Instanz. Die Ausnahme bildet der nvidia-t4-a10g-l4 Task Accelerator, der der verfügbaren Instanz der neuesten Generation den Vorzug gibt.

Einzelheiten zu den Instanztypen finden Sie unter Accelerated-computing Instanzen.

Im folgenden Beispiel spezifiziert die Workflow-Definition `nvidia-l4` als Beschleuniger:

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"
```

Erweiterte Ressourcenkonfiguration

Sie können eine Aufgabe auch so konfigurieren, dass mehrere Beschleuniger in der von Ihnen bevorzugten Prioritätsreihenfolge reserviert werden. Verwenden Sie dazu omicsResourceFallbackOrder Richtlinie. Diese Direktive ist für WDL und Nextflow verfügbar. HealthOmics probiert jedes Beschleunigerprofil in der von Ihnen definierten Reihenfolge aus, bis eines erfolgreich ist, einschließlich eines optionalen endgültigen CPU-Profils.

Dies ist in den folgenden Szenarien nützlich:

- Fallback von GPU zu GPU — Listet die GPU-Typen in der Reihenfolge ihrer Priorität auf (versuchen Sie es beispielsweise `nvidia-l40s` zuerst, dann greifen Sie zurück zu `nvidia-l4`).
- Fallback von GPU zu CPU — Füge ein Finale hinzu CPU-only Profil, sodass die Aufgabe auch dann abgeschlossen wird, wenn in der Region keine GPU-Kapazität verfügbar ist.

Für vollständige Informationen über omicsResourceFallbackOrder, einschließlich Profil-Feldverweise, Validierungsregeln und vollständiger WDL- und Nextflow-Beispiele, finden Sie unter Erweiterte Ressourcenkonfiguration.

Erweiterte Ressourcenkonfiguration

Mit dem omicsResourceFallbackOrder Direktive können Sie eine geordnete Liste von Ressourcenprofilen (z. B. Beschleuniger und CPU) für eine Aufgabe in Ihrem Workflow deklarieren. Sie geben diese Anweisung auf Aufgabenebene in WDL oder auf Prozess- oder Konfigurationsebene in Nextflow an. HealthOmics sucht nach den einzelnen Profilen in der von Ihnen angegebenen Reihenfolge, um die Verfügbarkeit zu reservieren. Wenn innerhalb des Warte-Timeouts keine Kapazität verfügbar ist, HealthOmics wechselt zum nächsten Ressourcenprofil in der Liste.

Dies ist nützlich, wenn Ihre bevorzugte Beschleunigerkapazität (z. B. G6e mit `nvidia-l40s`) ist nicht verfügbar und Sie möchten lieber auf einen anderen Beschleunigertyp oder auf eine CPU zurückgreifen, anstatt den Lauf zu scheitern. Der Ressourcenrückgriff ist eine von mehreren Strategien, um den Erfolg der GPU-Akquisition zu verbessern. Weitere Informationen zu diesen bewährten Methoden finden Sie unter Behebung von INSTANCE_RESERVATION_FAILED-Fehlern.

Funktionsweise

1. Sie definieren eine geordnete Liste von Ressourcenprofilen in der omicsResourceFallbackOrder Direktive.
2. Zur Laufzeit HealthOmics versucht, Kapazität für das erste Profil in der Liste zu reservieren.
3. Wenn innerhalb des Wartezeitraums keine Kapazität verfügbar ist, HealthOmics wechselt zum nächsten Profil.
4. Die Aufgabe wird auf dem Profil ausgeführt, das zuerst erfolgreich ist.
5. Wenn alle Profile in der Liste fehlschlagen, schlägt die Aufgabe aus einem bestimmten Grund fehl `ALL_PROFILES_INSTANCE_RESERVATION_FAILED`. In WDL werden keine Engine-Wiederholungsversuche durchgeführt, wenn alle Profile nicht verfügbar sind. In Nextflow errorStrategy und maxRetries entscheiden Sie, ob die Aufgabe erneut versucht wird.

> **Note**
>
> Verwenden Sie in WDL omicsResourceFallbackOrder statt der üblichen Aufgabe acceleratorType, acceleratorCount, cpu, memory, und omicsResourceWaitTimeoutInMin Felder. Diese dürfen nicht auf der obersten Ebene festgelegt werden, wenn die Richtlinie vorliegt. In Nextflow ist ein Prozess cpus und memory Direktiven gelten für jedes Profil, das

sie weglässt. Details hierzu finden Sie unter Konfigurieren Sie den Ressourcenfallback mit ComicsResourceFallbackOrder.

Anwendungsfälle

Szenario	Description
Fallback von GPU zu GPU	Listet die Beschleunigertypen (oder GPU-Typen) in der Reihenfolge ihrer Priorität auf. Versuchen Sie es zum Beispiel `nvidia-l40s` zuerst, dann greifen Sie zurück zu `nvidia-l4` . Es sind keine Befehlsän derungen erforderlich, wenn die Arbeitslast bei allen Beschleunigertypen gleich ist.
Fallback zwischen GPU und CPU	Fügen Sie ein endgültiges Profil hinzu, das Folgendes auslässt acceleratorType zum CPU-only Ausweichlösung. Benutze das AWS_HEALTHOMICS_RESOURCE_TYPE Umgebungsvariable, um den Befehl nach Ressourcentyp zu verzweigen.

Task-level Laufzeitfelder

> **Note**
>
> Auf dieser Seite Profil bezieht sich auf eine einzelne Beschleuniger- oder CPU-Ressourcenkonfiguration, die in deklariert ist omicsResourceFallbackOrder das besteht aus den 5 Feldern pro Profil.

Bei der Verwendung omicsResourceFallbackOrder, Laufzeitfelder auf Aufgabenebene sind in zwei Gruppen aufgeteilt:

- Per-profile Felder (acceleratorType, acceleratorCount, cpu, memory, omicsResourceWaitTimeoutInMin) — unabhängig konfigurierbar für jedes Profil in der Liste.

- Gemeinsam genutzte Felder (alle anderen Laufzeitfelder, wie docker, maxRetries) — wird einmal auf der obersten Ebene gesetzt und auf jedes Profil auf die gleiche Weise angewendet.

In WDL ist das CPU-Feld cpu. In Nextflow ist es cpus, und memory ist wie für den Nextflow geschrieben memory Direktive, zum Beispiel `'32 GB'`. Die gemeinsam genutzten Felder in Nextflow sind die anderen Direktiven des Prozesses, wie container, errorStrategy, und maxRetries.

Beispiele

WDL

Die folgende WDL-Aufgabe sucht nach `nvidia-l40s` zuerst (bis zu 45 Minuten auf Kapazität warten), dann `nvidia-l4` (Standard-Wartefenster), dann fällt auf ein CPU-only Profil (32 vCPUs, 128 GiB), wenn keiner der Beschleunigertypen verfügbar ist.

```
task align {
  command <<<
    # Branch based on which resource type was allocated
    if [ "$AWS_HEALTHOMICS_RESOURCE_TYPE" = "cpu" ]; then
      sentieon bwa mem -t 32 ~{reference} ~{fastq}
    else
      pbrun fq2bam --ref ~{reference} --in-fq ~{fastq}
    fi
  >>>

  runtime {
    docker: "my-registry/align-multi-arch:latest"
    maxRetries: 2

    omicsResourceFallbackOrder: [
      {"acceleratorType": "nvidia-l40s", "acceleratorCount": 1,
       "cpu": 8, "memory": "32 GiB",
       "omicsResourceWaitTimeoutInMin": 45},

      {"acceleratorType": "nvidia-l4", "acceleratorCount": 1,
       "cpu": 8, "memory": "32 GiB"},

      {"cpu": 32, "memory": "128 GiB"}
    ]
  }
}
```

Nextflow

Der folgende Nextflow-Prozess sucht nach `nvidia-l40s` zuerst (bis zu 45 Minuten auf Kapazität warten), dann `nvidia-l4` (Standard-Wartefenster), dann fällt auf ein CPU-only Profil (32 vCPUs, 128 GB), wenn keiner der Beschleunigertypen verfügbar ist. Schreiben Sie die Direktive in einen Prozesskörper als Methodenaufruf mit Klammern.

```
process ALIGN {
    container 'my-registry/align-multi-arch:latest'
    errorStrategy 'retry'
    maxRetries 2

    omicsResourceFallbackOrder([
        [acceleratorType: 'nvidia-l40s', acceleratorCount: 1, cpus: 8, memory: '32
 GB',
         omicsResourceWaitTimeoutInMin: 45],
        [acceleratorType: 'nvidia-l4', acceleratorCount: 1, cpus: 8, memory: '32
 GB'],
        [cpus: 32, memory: '128 GB']
    ])

    script:
    """
    # Branch based on which resource type was allocated
    if [ "\$AWS_HEALTHOMICS_RESOURCE_TYPE" = "cpu" ]; then
      sentieon bwa mem -t \$(nproc) ${reference} ${fastq}
    else
      pbrun fq2bam --ref ${reference} --in-fq ${fastq}
    fi
    """
}
```

Sie können die Direktive auch in setzen `nextflow.config`, entweder in der `process` Umfang für jeden Prozess oder mit einem Prozessselektor. Weisen Sie der Liste in einer Konfigurationsdatei Folgendes zu =:

```
process {
    withLabel: 'gpu_align' {
        omicsResourceFallbackOrder = [
            [acceleratorType: 'nvidia-l40s', acceleratorCount: 1, cpus: 8, memory:
 '32 GB'],
            [cpus: 32, memory: '128 GB']
```

```
        ]
    }
}
```

In diesen Beispielen AWS_HEALTHOMICS_RESOURCE_TYPE teilt dem Befehl mit, welcher Ressourcenpfad ausgewählt wurde (zum Beispiel `"nvidia-l40s"` oder `"cpu"`).

> **Note**
>
> Das Docker-Image muss sowohl Beschleuniger- als auch CPU-Codepfade unterstützen, wenn Ihre Fallback-Bestellung ein CPU-Profil enthält. Stellen Sie sicher, dass Ihr Container die erforderlichen Tools für alle Ressourcenprofile in der Liste enthält.

Per-profile Referenz zum Feld

Jeder Eintrag in der omicsResourceFallbackOrder Die Liste ist eine Karte, die ein Ressourcenprofil beschreibt. Alle Felder sind optional; ein Profil kann eine Teilspezifikation sein. Jedes Profil muss Schlüssel (Zeichenfolge) in Anführungszeichen verwenden. Das cpu und memory Die Felder unterscheiden sich zwischen WDL und Nextflow, daher ist in der folgenden Tabelle jeweils eine eigene Zeile aufgeführt.

Feld	Typ	Standard, wenn weggelassen	Hinweise
`acceleratorType`	Zeichenfolge	Wenn nicht angegeben, wird das Profil als CPU betrachtet	Muss einer der 7 unterstützten Beschleunigertypen sein. Siehe Taskbeschleuniger in einem HealthOmics Workflow-Definition. Lassen Sie dieses Feld aus, um eine anzugeben CPU-only Profil. Stellen Sie es

Feld	Typ	Standard, wenn weggelassen	Hinweise
			für das CPU-Profil nicht auf ein "".
`acceleratorCount`	Ganzzahl	Feld fehlt wann `acceleratorType` ist auch abwesend	Muss zusammen mit angegeben werden `acceleratorType` . Ein Profil kann nicht das eine ohne das andere haben.
`cpu` (WDL)	Integer oder Float	1 vCPU oder der GPU-Instanztyp (Standard) für ein GPU-Profil	Auf eine ganze vCPU aufgerundet, mindestens 1. Gleiche fraktionierte Unterstützung wie auf der obersten Ebene runtime.cpu.
`cpus` (Nächster Fluss)	Integer, Float oder ein String, der eine ganze Zahl enthält (z. B. `'8'`)	Der Prozess ist cpus, falls gesetzt. Andernfalls der GPU-Instanztyp als Standard für ein GPU-Profil.	Auf eine ganze vCPU gekürzt, wie cpus Die Direktive tut es. Ein Wert unter 1 wird mit 1 vCPU ausgeführt. Ein negativer Wert wird zurückgewiesen.
`memory` (WDL)	Zeichenfolge, zum Beispiel `"32 GiB"`	1 GiB oder der GPU-Instanztyp (Standard) für ein GPU-Profil	Dasselbe Format wie auf der obersten Ebene runtime.memory. Auf ein ganzes GiB aufgerundet.

Feld	Typ	Standard, wenn weggelassen	Hinweise
`memory` (Nächster Fluss)	Zeichenfolge oder Nextflow-Speicherwert, zum Beispiel `'32 GB'` oder `32.GB`	Der Prozess ist `memory`, falls gesetzt. Andernfalls der GPU-Instanztyp als Standard für ein GPU-Profil.	Dasselbe Format wie das `memory` Direktive: B, KB, MB, GB oder TB. Auf ein ganzes GB abgerundet. Ein Wert unter 1 GB entspricht 1 GB.
`omicsResourceWaitTimeoutInMin`	Ganzzahl	20 Minuten für ein Single-GPU-Beschleunigerpaket und 30 Minuten für Multi-GPU-Beschleunigerpakete. Dies sind ebenfalls empfohlene Mindestwerte.	Es gibt keine Obergrenze. Steuert, wie lange HealthOmics sucht nach einem Profil, bevor zum nächsten Profil gewechselt wird. Siehe Timeout-Verhalten.

> **Note**
>
> Ausgelassene Felder erhalten den Standardwert, keine Werte, die von einem früheren Profil in der Liste übernommen wurden. Ein Feld, das in einem Profil weggelassen wird, erhält seinen dokumentierten Standardwert (wie in der obigen Tabelle angegeben), nicht einen Wert, der aus einem früheren Profil in der Liste kopiert wurde.

Timeout-Verhalten

omicsResourceWaitTimeoutInMin steuert, wie lang HealthOmics wartet auf die Beschleunigerkapazität eines bestimmten Profils, bevor es zum nächsten Profil übergeht.

- Per-profile, nicht global. Jedes Beschleunigerprofil kann seinen eigenen Timeout angeben. Konfigurieren Sie eine längere Wartezeit für einen bevorzugten High-End-Beschleuniger und eine kürzere Wartezeit für einen Fallback-Typ.

- Es wird ein Minimum von 20 Minuten empfohlen. Werte unter 20 Minuten (30 für Pakete mit mehreren GPUs) werden akzeptiert, erzeugen jedoch eine Validierungswarnung.
- Das Timeout schreitet voran, schlägt nicht fehl. Wenn der Timeout abgelaufen ist, HealthOmics wechselt zum nächsten Profil — die Aufgabe schlägt nicht fehl. Ein Fehler tritt erst auf, wenn alle Profile erschöpft sind.
- Gilt nicht für CPU-only Profile. Ein CPU-Profil hat keine Kapazitätsbeschränkungen. Auslassen omicsResourceWaitTimeoutInMin auf dem endgültigen CPU-Profil.
- Bei Wiederholungen wird ein neues Timeout-Fenster angezeigt. Jeder OOM- oder Servicefehler-Wiederholungsversuch beginnt von selbst. omicsResourceWaitTimeoutInMin Fenster auf demselben Profil, anstatt die Zeit zu übernehmen, die bereits bei einem früheren Versuch aufgewendet wurde.

Umgebungsvariablen

HealthOmics legt die folgende Umgebungsvariable im Aufgabencontainer fest, sodass Ihr Befehl je nach zugewiesenem Ressourcenprofil verzweigen kann:

Variable	Wert	Beispielwerte
`AWS_HEALTHOMICS_RESOURCE_TYPE`	Die `acceleratorType` des aktiven Profils, oder `"cpu"` für ein CPU-only Profil.	`"nvidia-l40s"` , `"nvidia-l4"` , `"cpu"`

Validierungskriterien

HealthOmics validiert Folgendes bei der Workflow-Erstellung und überprüft es erneut zur Laufzeit der Aufgabe. Alle Regeln lehnen den Workflow oder die Aufgabe ab, sofern nicht anders angegeben.

1. Kann nicht mit einzelnen Ressourcenrichtlinien kombiniert werden. Wenn omicsResourceFallbackOrder angegeben ist, die oberste Ebene acceleratorType, acceleratorCount, cpu, memory, und omicsResourceWaitTimeoutInMin darf nicht in derselben Aufgabe angegeben werden.
2. Muss eine Liste sein. omicsResourceFallbackOrder muss als eine Reihe von Profilen geschrieben werden.
3. Darf nicht leer sein. Die Liste muss mindestens ein Profil enthalten.

4. Schlüssel in Anführungszeichen erforderlich. Jeder Feldname muss eine Zeichenfolge in Anführungszeichen sein, zum Beispiel `{"acceleratorType": "nvidia-l4", "acceleratorCount": 1}`. Die Überprüfung von Bareword-Schlüsseln schlägt fehl.
5. Non-empty Profile. Ein leeres Profil (`{}`) ist nicht zulässig.
6. acceleratorType und acceleratorCount geh zusammen. Ein Profil, das das eine festlegt, muss das andere festlegen. Ein CPU-Profil muss beide weglassen.
7. Nur unterstützte Beschleunigertypen. acceleratorType muss ein unterstützter Beschleunigertyp sein oder weggelassen werden (CPU-Profil). Leere Zeichenfolge `""` wird nicht akzeptiert.
8. Der Beschleunigertyp ist in der Region nicht verfügbar (nur Warnung). Wenn ein Accelerator-Typ in der AWS-Region des Workflows nicht unterstützt wird, HealthOmics überspringt dieses Profil bei der Aufgabenplanung. In den Laufzeitprotokollen wird eine Warnung angezeigt, wenn ein Beschleuniger übersprungen wird, weil er in dieser Region nicht unterstützt (oder nicht angeboten) wird.
9. Minimaler Timeout für die Wartezeit. omicsResourceWaitTimeoutInMin Der empfohlene Wert ist ≥ 20 Minuten (≥ 30 für Pakete mit mehreren GPUs).
10. Doppelte Profile (nur Warnung). Doppelte Profile sind zulässig, führen jedoch zu einer Warnung. Erwägen Sie eine Erhöhung omicsResourceWaitTimeoutInMin stattdessen auf dem früheren Profil.
11. Unbekannte Felder wurden zurückgewiesen. Nur die 5 oben aufgeführten Felder pro Profil sind zulässig.
12. Richtige Typen. Zum Beispiel acceleratorCount muss eine Zahl sein, keine Zeichenfolge.
13. Maximale Ressourcenwerte. cpu, memory, und acceleratorCount haben jeweils ein Maximum — 192, 1536 GiB bzw. 4, was der größten Instanz entspricht HealthOmics unterstützt. Werte, die über diesen Werten liegen, werden bei der Workflow-Erstellung und bei der Ausführung der Aufgabe abgelehnt. Die Höchstwerte für unterstützte Instanzen finden Sie unter Rechenleistung und Arbeitsspeicher.
14. Höchstens ein CPU-Profil. Nur ein Profil, das ausgelassen wird acceleratorType ist erlaubt.
15. Das CPU-Profil muss das letzte sein. Ein CPU-Profil, falls vorhanden, muss das letzte Profil in der Liste sein. Sie sollten alle gewünschten Beschleunigeroptionen angeben, bevor Sie ein CPU-Profil in der Prioritätsreihenfolge angeben.
16. Maximal 10 Profile pro Aufgabe.

Nextflow-specific Kriterien: Bei Nextflow lehnt die Workflow-Erstellung den Workflow nicht ab. Stattdessen wird jedes Ergebnis in der Statusmeldung des Workflows gemeldet, und bei

einem ungültigen Wert schlägt die Aufgabe fehl, wenn sie ausgeführt wird. Regel 1 ist eher eine Warnung als eine Ablehnung. Regel 4 gilt nicht, da Profilschlüssel als Groovy-Map-Schlüssel geschrieben werden. Werte, die nur zur Laufzeit bekannt sind, werden bei der Ausführung der Aufgabe überprüft. Details hierzu finden Sie unter Konfigurieren Sie den Ressourcenfallback mit ComicsResourceFallbackOrder.

Interaktion mit Wiederholungsversuchen

Für Out-of-Memory (OOM) und Servicefehler (5xx ausgenommen) `ALL_PROFILES_INSTANCE_RESERVATION_FAILED`), HealthOmics versucht die Aufgabe wie folgt erneut:

- Wiederholungen erfolgen innerhalb des aktuell aktiven Profils, das zuvor erfolgreich reserviert wurde.
- Wiederholungen wechseln nie zum nächsten Profil in der Ausweichreihenfolge.
- Ein erschöpfender Wiederholungsversuch maxRetries schlägt bei der Aufgabe fehl.

Weitere Informationen zu Aufgabenwiederholungen finden Sie unter HealthOmics, siehe Die Aufgabe wird erneut versucht.

> **Warning**
>
> Wenn in WDL alle Profile beim ersten Engine-Versuch ohne Instanzreservierung erschöpft sind, schlagen die Aufgabe und die Ausführung mit dem Status fehl `ALL_PROFILES_INSTANCE_RESERVATION_FAILED`. Auch wenn Sie Wiederholungen konfigurieren, HealthOmics versucht es bei diesem Fehlercode nicht erneut. Wir empfehlen, Ihre anzupassen omicsResourceWaitTimeoutInMin angemessen. Nextflow-Wiederholungen funktionieren anders; siehe Konfigurieren Sie den Ressourcenfallback mit ComicsResourceFallbackOrder.

Bewährte Methoden

Folgen Sie diesen bewährten Methoden, wenn Sie den Ressourcenfallback konfigurieren:

- Vermeiden Sie Pakettypen mit mehreren GPUs in der Fallback-Reihenfolge. Accelerator-Typen, die sich über mehrere Instance-Familien erstrecken (z. B. `nvidia-t4-a10g-l4`) werden intern nicht empfohlen omicsResourceFallbackOrder. Verwenden Sie stattdessen Einfamilientypen.

Einzelheiten zu den verfügbaren Beschleunigertypen finden Sie unter Taskbeschleuniger in einem HealthOmics Workflow-Definition.

- Stellen Sie angemessene Wartezeiten ein. Erhöhen Sie für Beschleunigerprofile mit hoher Priorität omicsResourceWaitTimeoutInMin zu geben HealthOmics mehr Zeit, um Kapazitäten zu finden.
- Platzieren Sie die CPU-Profile zuletzt. Wenn Sie eine hinzufügen CPU-only Fallback, es muss der letzte Eintrag sein, sodass Beschleuniger bevorzugt werden, sofern verfügbar.
- Verwenden Sie Container-Images mit mehreren Architekturen. Wenn Sie GPU-CPU-Fallback verwenden, stellen Sie sicher, dass Ihr Docker-Image beide unterstützt GPU-accelerated und CPU-only Code-Pfade.

Einschränkungen

Die folgenden Einschränkungen gelten:

- In WDL omicsResourceFallbackOrder wird nicht unterstützt in scatter Blöcke. Es ist nur auf Aufgabenebene verfügbar.
- Die Direktive wird für WDL und Nextflow unterstützt. CWL-Unterstützung ist geplant.
- Instance-type Namen (zum Beispiel `omics.g6e.4xlarge`) werden nicht als Ressourcenwerte akzeptiert. Sie müssen die auf dieser Seite beschriebene Feldsyntax pro Profil verwenden.

Besonderheiten der WDL-Workflow-Definition

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

Themen

- Implizite Typkonvertierung in WDL (milde)
- Namespace-Definition in input.json
- Primitive Typen in WDL
- Komplexe Typen in WDL
- Richtlinien in WDL
- Andere unterstützte Laufzeitattribute
- Aufgaben-Metadaten in WDL
- Beispiel für eine WDL-Workflow-Definition

Implizite Typkonvertierung in WDL (milde)

HealthOmics unterstützt die implizite Typkonvertierung in der Datei input.json und der Workflow-Definition. Um die implizite Typumwandlung zu verwenden, geben Sie bei der Erstellung des Workflows für die Workflow-Engine die Option WDL Delient an. WDL Lenient umfasst alle Standard-WDL-Funktionen sowie zusätzliche Kompatibilitätsverhaltensweisen, die für Workflows entwickelt wurden, die von Cromwell migriert wurden. Es unterstützt Kundenrichtlinien von Cromwell und einige nicht konforme Logiken.

WDL Lenient unterstützt die Typkonvertierung für die folgenden Elemente in der Liste der begrenzten Ausnahmen von WDL: https://github.com/openwdl/wdl/blob/wdl-1.1/SPEC.md#-limited-exceptions

- Float auf Int, wobei der Zwang keinen Genauigkeitsverlust zur Folge hat (z. B. wird 1,0 auf 1 abgebildet).
- Zeichenfolge bis Int/Float, wobei der Zwang keinen Genauigkeitsverlust zur Folge hat.
- Ordnen Sie [W, X] dem Array [Pair [Y, Z]] zu, falls W zu Y und X zu Z erzwingbar ist.
- Array [Pair [W, X]] zu Map [Y, Z], falls W zu Y und X zu Z erzwingbar ist (z. B. 1,0 entspricht 1).

Um die implizite Typumwandlung zu verwenden, geben Sie die Workflow-Engine als WDL_LENIENT an, wenn Sie den Workflow oder die Workflow-Version erstellen.

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

Namespace-Definition in input.json

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

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

Sie können sie als vollständig qualifizierte Variablen in input.json verwenden:

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

Primitive Typen in WDL

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

Primitive Typen

WDL-Typ	JSON-Typ	Beispiel WDL	Beispiel für einen JSON-Schlüssel und -Wert	Hinweise
`Boolean`	`boolean`	`Boolean b`	`"b": true`	Der Wert muss in Kleinbuchstaben und ohne Anführungszeichen geschrieben werden.
`Int`	`integer`	`Int i`	`"i": 7`	Muss nicht in Anführungszeichen stehen.
`Float`	`number`	`Float f`	`"f": 42.2`	Muss nicht in Anführungszeichen stehen.
`String`	`string`	`String s`	`"s": "characters"`	JSON-Zeichenfolgen, die eine URI sind, müssen einer WDL-Datei zugeordnet

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

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

Komplexe Typen in WDL

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

Komplexe Typen

WDL-Typ	JSON-Typ	Beispiel WDL	Beispiel für einen JSON-Schlüssel und -Wert	Hinweise
`Array`	`array`	`Array[Int] nums`	`“nums": [1, 2, 3]`	Die Mitglieder des Arrays müssen dem Format des WDL-Array-Typs entsprechen.
`Pair`	`object`	`Pair[String, Int] str_to_i`	`“str_to_i": {"left": "0", "right": 1}`	Jeder Wert des Paares muss das JSON-Format des entsprechenden WDL-Typs verwenden. Bei Zeichenfolgenschlüsselnamen in WDL-Paar-JSON-Repräsentationen wird nicht zwischen Groß- und Kleinschreibung unterschieden. Beispielsweise werden {"left“: „0", „right“:

WDL-Typ	JSON-Typ	Beispiel WDL	Beispiel für einen JSON-Schlüssel und -Wert	Hinweise
				1} und {"LEFT“: „0", „Right“: 1} bei der Deseriali sierung in einen Pair-Typ als gleichwertig behandelt.
`Map`	`object`	`Map[Int, String] int_to_st ring`	`"int_to_s tring": { 2: "hello", 1: "goodbye" }`	Jeder Eintrag in der Map muss das JSON-Form at des entsprech enden WDL-Typs verwenden.
`Struct`	`object`	`struct SampleBam AndIndex { String sample_na me File bam File bam_index } SampleBam AndIndex b_and_i`	`"b_and_i": { "sample_n ame": "NA12878" , "bam": "s3://amz n-s3-demo -bucket1/ NA12878.b am", "bam_inde x": "s3:// amzn- s3-demo- bucket1/ NA12878.b am.bai" }`	Die Namen der Strukturm itglieder müssen exakt mit den Namen der JSON-Obje ktschlüssel übereinstimmen. Jeder Wert muss das JSON-Form at des entsprech enden WDL-Typs verwenden.

WDL-Typ	JSON-Typ	Beispiel WDL	Beispiel für einen JSON-Schlüssel und -Wert	Hinweise
`Object`	N/A	N/A	N/A	Der `Object` WDL-Typ ist veraltet und sollte `Struct` in jedem Fall durch einen ersetzt werden.

Richtlinien in WDL

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

Konfigurieren Sie GPU-Ressourcen

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

Das folgende Beispiel zeigt, wie diese Direktiven verwendet werden:

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

Konfigurieren Sie den Ressourcenfallback mit Comics ResourceFallbackOrder

HealthOmics unterstützt die omicsResourceFallbackOrder Runtime-Direktive, um eine geordnete Liste von Beschleunigern- (oder GPU-) und CPU-Ressourcenprofilen für eine Aufgabe zu definieren. Wenn der angeforderte Beschleunigertyp nicht verfügbar ist, wechselt HealthOmics automatisch zum nächsten Profil in der Liste. Sie können ein CPU-only Profil als letzte Option hinzufügen.

Das folgende Beispiel versucht es `nvidia-l40s` zuerst`nvidia-l4`, dann und fällt schließlich auf die CPU zurück, wenn keiner der Beschleuniger verfügbar ist:

```
runtime {
    docker: "my-registry/align-multi-arch:latest"
    omicsResourceFallbackOrder: [
        {"acceleratorType": "nvidia-l40s", "acceleratorCount": 1, "cpu": 8, "memory":
 "32 GiB", "omicsResourceWaitTimeoutInMin": 45},
        {"acceleratorType": "nvidia-l4",   "acceleratorCount": 1, "cpu": 8, "memory":
 "32 GiB"},
        {"cpu": 32, "memory": "128 GiB"}
    ]
}
```

HealthOmics legt die AWS_HEALTHOMICS_RESOURCE_TYPE Umgebungsvariable automatisch im Container fest. Ihr Wert ist der Beschleunigertyp des aktiven Profils (z. B.`nvidia-l40s`) oder `cpu` für ein CPU-only Profil. Verwenden Sie ihn, um Ihren Befehl zu verzweigen, wenn sich GPU- und CPU-Codepfade unterscheiden.

Wenn jedes Profil in der Liste erschöpft ist, schlägt die Aufgabe aufgrund eines Fehlers fehl`ALL_PROFILES_INSTANCE_RESERVATION_FAILED`. OOM- und Servicefehlerwiederholungen werden innerhalb des aktuellen Profils ausgeführt und nicht zum nächsten Profil weitergeleitet.

Weitere Informationen finden Sie unterErweiterte Ressourcenkonfiguration. Unterstützte Beschleunigertypen finden Sie unter. Taskbeschleuniger in einem HealthOmics Workflow-Definition

Konfigurieren Sie die Aufgabenwiederholung für Servicefehler

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

Das folgende Beispiel legt fest`preemptible`, dass Wiederholungsversuche aufgrund von Servicefehlern deaktiviert werden:

```
{
  preemptible: 0
}
```

Weitere Hinweise zu Aufgabenwiederholungen in finden Sie HealthOmics unter. Die Aufgabe wird erneut versucht

Konfigurieren Sie die Aufgabenwiederholung bei unzureichendem Arbeitsspeicher

HealthOmics unterstützt Wiederholungen für eine Aufgabe, die fehlgeschlagen ist, weil der Speicher knapp wurde (Container-Exit-Code 137, 4XX HTTP-Statuscode). HealthOmics verdoppelt die Speichermenge für jeden Wiederholungsversuch.

In der Standardeinstellung wird bei dieser Art von Fehler HealthOmics kein erneuter Versuch durchgeführt. Verwenden Sie die `maxRetries` Direktive, um die maximale Anzahl von Wiederholungen anzugeben.

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

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

> **Note**
>
> Für den erneuten Versuch einer Aufgabe wegen unzureichenden Speichers ist GNU findutils 4.2.3+ erforderlich. Der Standard-Image-Container enthält dieses Paket HealthOmics . Wenn Sie in Ihrer WDL-Definition ein benutzerdefiniertes Image angeben, stellen Sie sicher, dass das Image GNU Findutils 4.2.3+ enthält.

Konfigurieren Sie Rückgabecodes

Das ReturnCodes-Attribut bietet einen Mechanismus zum Angeben eines Rückgabecodes oder einer Reihe von Rückgabecodes, der auf eine erfolgreiche Ausführung einer Aufgabe hinweist. Die WDL-Engine berücksichtigt die Rückgabecodes, die Sie im Laufzeitabschnitt der WDL-Definition angeben, und legt den Aufgabenstatus entsprechend fest.

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

HealthOmics unterstützt auch einen Alias namens continue OnReturnCode, der über dieselben Funktionen wie ReturnCodes verfügt. Wenn Sie beide Attribute angeben, wird der ReturnCodes-Wert HealthOmics verwendet.

Unterstützte WDL-Festplattenformulare

HealthOmics akzeptiert alle WDL `disks` 1.1-Standardformulare. Der Bereitstellungspfad und der Festplattentypspezifizierer (`SSD`,`HDD`) werden ignoriert — nur die numerische Größe wird extrahiert. Wenn mehrere Einträge deklariert sind, werden die Größen zu einer einzigen `/tmp` Zuordnung summiert.

Akzeptiertes Formular	Beispiel	Gelöste Größe
Ganzzahl	`disks: 700`	700 GiB
Zeichenfolge	`disks: "700"`	700 GiB
Zeichenfolge mit Einheit	`disks: "700 GiB"` oder `"700 GB"`	700 GiB
Cromwell-style mit Pfad und Typ	`disks: "local-disk 700 SSD"`	700 GiB (Pfad und Typ werden ignoriert)
Comma-separated Einträge	`disks: "local-disk 300 SSD, /scratch 200 SSD"`	500 GiB (summiert)
Zeichenfolgen-Array	`disks: ["/tmp 300 GiB", "/scratch 200 GiB"]`	500 GiB (summiert)

HealthOmics montiert den gesamten kurzlebigen Speicher auf. `/tmp` Wenn Ihre Workflow-Definition mehrere `disks` Einträge deklariert, werden die Größen summiert und die Gesamtmenge wird als einzelnes Volume bereitgestellt. `/tmp` Weitere Informationen zum `disks` WDL-Laufzeitattribut finden Sie in der WDL 1.1-Spezifikation. https://github.com/openwdl/wdl/blob/wdl-1.1/SPEC.md#disks

Andere unterstützte Laufzeitattribute

Konfigurieren Sie das Timeout auf Aufgabenebene mit dem Laufzeitattribut OmicsTimeout

HealthOmics bietet ein benutzerdefiniertes `omicsTimeout` Attribut, mit dem Sie eine maximale Aufgabendauer in Ihrem Workflow festlegen können. Geben Sie die Timeout-Dauer als Ganzzahl (Sekunden) oder als Zeichenfolge mit einer oder mehreren der folgenden Einheiten an:`s`, `mh`, oderd. Beispiel: `"40m"`, `"1h30m"` oder `"1m3s"`.

Das folgende Beispiel zeigt, wie das `omicsTimeout` Attribut im `runtime` Abschnitt einer WDL-Aufgabendefinition angegeben wird. In diesem Beispiel wird ein Timeout von 40 Minuten festgelegt.

```
    runtime {
        omicsTimeout: "40m"
    }
```

HealthOmics bietet die folgende Unterstützung für das `omicsTimeout` WDL-Laufzeitattribut:

1. HealthOmics unterstützt eine Granularität von 1 Minute für den Timeout-Wert. Sie können einen Wert zwischen 60 Sekunden und dem Wert für die maximale Laufdauer angeben.
2. Wenn Sie einen Wert unter 60 eingeben, wird HealthOmics dieser auf 60 Sekunden aufgerundet. Bei Werten über 60 wird auf die nächste Minute HealthOmics abgerundet.
3. Wenn bei einer Aufgabe ein Timeout auftritt, wird die Aufgabe HealthOmics abgebrochen. Dieser Vorgang kann eine Dauer von ein bis zwei Minuten haben.
4. HealthOmics Setzt bei einem Task-Timeout die Ausführung und den Aufgabenstatus auf Fehlgeschlagen und storniert die anderen Aufgaben in der Ausführung (für Aufgaben im Status Gestartet, Ausstehend oder Wird ausgeführt). HealthOmics exportiert die Ausgaben von Aufgaben, die vor dem Timeout abgeschlossen wurden, an den von Ihnen angegebenen S3-Ausgabeort.
5. Die Zeit, die eine Aufgabe im Status „Ausstehend“ verbringt, wird nicht auf die Aufgabendauer angerechnet.
6. Wenn die Ausführung Teil einer Ausführungsgruppe ist und für die Ausführungsgruppe das Zeitlimit früher als der Task-Timer überschritten wird, gehen die Ausführung und die Aufgabe in den Status Fehlgeschlagen über.

Alle Läufe haben eine maximale Dauer, die das einstellbare Kontingent in HealthOmics Servicekontingente Grenzen hält. Sie können eine Erhöhung dieses Kontingents beantragen.

Aufgaben-Metadaten in WDL

HealthOmics unterstützt die folgenden Metadatenoptionen für WDL-Aufgaben.

Deaktivieren Sie das Caching auf Aufgabenebene mit dem Attribut volatile

Das volatile Attribut ermöglicht es Ihnen, das Caching von Anrufen für bestimmte Aufgaben in Ihrem WDL-Workflow zu deaktivieren. Wenn eine Aufgabe als flüchtig markiert ist, wird sie immer ausgeführt und verwendet niemals zwischengespeicherte Ergebnisse, auch wenn das Caching für die Ausführung aktiviert ist.

Fügen Sie das Attribut volatile zum Meta-Abschnitt Ihrer Aufgabendefinition hinzu:

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

    input {
        String input_file
    }

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

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

Beispiel für eine WDL-Workflow-Definition

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

Das folgende Beispiel zeigt, wie der Amazon ECR-Container als Parameter aufgenommen wird. Auf diese Weise können HealthOmics Sie die Zugriffsberechtigungen für Ihren Container überprüfen, bevor er mit der Ausführung beginnt.

```
{
   ...
```

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

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

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

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

```
{
    "input_cram": "omics://429915189008.storage.us-west-2.amazonaws.com/111122223333/
readSet/4500843795/source1",
    "ref_dict": "s3://amzn-s3-demo-bucket1/inputs/Homo_sapiens_assembly38.dict",
    "ref_fasta": "s3://amzn-s3-demo-bucket1/inputs/Homo_sapiens_assembly38.fasta",
    "ref_fasta_index": "s3://amzn-s3-demo-bucket1/inputs/
Homo_sapiens_assembly38.fasta.fai",
    "sample_name": "NA12878"
}
```

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

```
 version 1.0
workflow CramToBamFlow {
    input {
        File ref_fasta
        File ref_fasta_index
        File ref_dict
        File input_cram
        String sample_name
        String gotc_docker = "<account>.dkr.ecr.us-west-2.amazonaws.com/genomes-in-the-
cloud:latest"
    }
```

```
    #Converts CRAM to SAM to BAM and makes BAI.
    call CramToBamTask{
         input:
            ref_fasta = ref_fasta,
            ref_fasta_index = ref_fasta_index,
            ref_dict = ref_dict,
            input_cram = input_cram,
            sample_name = sample_name,
            docker_image = gotc_docker,
     }
     #Validates Bam.
     call ValidateSamFile{
        input:
           input_bam = CramToBamTask.outputBam,
           docker_image = gotc_docker,
     }
     #Outputs Bam, Bai, and validation report to the FireCloud data model.
     output {
         File outputBam = CramToBamTask.outputBam
         File outputBai = CramToBamTask.outputBai
         File validation_report = ValidateSamFile.report
      }
}
#Task definitions.
task CramToBamTask {
    input {
       # Command parameters
       File ref_fasta
       File ref_fasta_index
       File ref_dict
       File input_cram
       String sample_name
       # Runtime parameters
       String docker_image
    }
   #Calls samtools view to do the conversion.
   command {
       set -eo pipefail

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

```
    #Runtime attributes:
    runtime {
        docker: docker_image
    }

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

#Validates BAM output to ensure it wasn't corrupted during the file conversion.
task ValidateSamFile {
   input {
      File input_bam
      Int machine_mem_size = 4
      String docker_image
   }
   String output_name = basename(input_bam, ".bam") + ".validation_report"
   Int command_mem_size = machine_mem_size - 1
   command {
       java -Xmx~{command_mem_size}G -jar /usr/gitc/picard.jar \
       ValidateSamFile \
       INPUT=~{input_bam} \
       OUTPUT=~{output_name} \
       MODE=SUMMARY \
       IS_BISULFITE_SEQUENCED=false
    }
    runtime {
    docker: docker_image
    }
   #A text file is generated that lists errors or warnings that apply.
    output {
        File report = "~{output_name}"
    }
}
```

Besonderheiten der Nextflow-Workflow-Definition

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

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

Weitere Informationen zum Linting einer Nextflow-Definition vor dem Erstellen des Workflows finden Sie unter Workflow-Definitionen lokal validieren.

Themen

- Verwenden Sie die Plugins nf-schema und nf-validation
- Speicher-URIs angeben
- Nextflow-Richtlinien
- Verwenden Sie Nextflow-Profile
- Exportieren Sie Inhalte auf Workflow-Ebene
- Aufgabeninhalt exportieren
- Generieren Sie Nextflow-Ausführungsberichte
- Geben Sie die Nextflow-Syntaxversion an
- Automatische Syntaxvalidierung bei der Workflow-Erstellung
- Effiziente Nutzung von Scratch Storage in Nextflow

Verwenden Sie die Plugins nf-schema und nf-validation

> **Note**
>
> Zusammenfassung von HealthOmics Unterstützung für Plugins:
>
> - v22.04 — keine Unterstützung für Plugins
> - v23.10 — unterstützt `nf-schema` und `nf-validation`
> - v24.10 — unterstützt `nf-schema`
> - v25.10, v26.04 — unterstützt `nf-schema`, `nf-core-utils`, `nf-fgbio`, und `nf-prov`

HealthOmics bietet die folgende Unterstützung für Nextflow-Plugins:

- Für Nextflow v23.10 HealthOmics installiert das nf-validation @1 .1.1-Plugin vor.

- Für Nextflow v23.10 und v24.10 HealthOmics installiert das nf-schema @2 .3.0-Plugin vor.
- Für Nextflow v25.10 HealthOmics installiert die Plugins nf-schema @2 .6.1, nf-core-utils @0 .4.0, nf-prov @1 .7.0 und nf-fgbio @1 .0.1 vor.
- Für Nextflow v26.04 HealthOmics installiert die Plugins nf-schema @2 .7.2, nf-core-utils @0 .4.0, nf-prov @1 .7.0 und nf-fgbio @1 .0.1 vor.
- Sie können während einer Workflow-Ausführung keine zusätzlichen Plugins abrufen. HealthOmics ignoriert alle anderen Plugin-Versionen, die Sie in der `nextflow.config` datei.
- Für Nextflow v24 und höher `nf-schema` ist die neue Version der veralteten `nf-validation` Plugin. Weitere Informationen finden Sie unter nf-Schema im Nextflow GitHub Repositorium.

Speicher-URIs angeben

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

HealthOmics unterstützt teilweise die Verwendung von Glob-Mustern in Amazon S3-URIs oder HealthOmics Speicher-URIs. Verwenden Sie Glob-Muster in der Workflow-Definition für die Erstellung von `path` oder `file` Kanäle. Das erwartete Verhalten und die genauen Fälle finden Sie unter Nextflow: Behandlung des Glob-Musters in Amazon S3-Eingaben.

Nextflow-Richtlinien

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

1. Globale Konfiguration in der Konfigurationsdatei.
2. Aufgabenbereich der Workflow-Definition.
3. Task-specific Selektoren in der Konfigurationsdatei.

Themen

- Strategie zur Wiederholung von Aufgaben mit Fehler-Strategie
- Versuche, Aufgaben erneut zu versuchen, indem MaxRetries
- Deaktivieren Sie die Wiederholung von Aufgaben mit ComicsRetryOn5xx

- Konfigurieren Sie den Ressourcenfallback mit ComicsResourceFallbackOrder
- Dauer der Aufgabe unter Verwendung des Zeit Richtlinie

Strategie zur Wiederholung von Aufgaben mit **`Fehler-Strategie`**

Benutze das `errorStrategy` Direktive, um die Strategie für Aufgabenfehler zu definieren. Wenn eine Aufgabe mit einer Fehleranzeige (einem Exit-Status ungleich Null) zurückkehrt, stoppt die Aufgabe standardmäßig und HealthOmics beendet den gesamten Lauf. Wenn du einstellst `errorStrategy` zu `retry`, HealthOmics versucht, die fehlgeschlagene Aufgabe einmal zu wiederholen. Informationen zum Erhöhen der Anzahl der Wiederholungen finden Sie unter Versuche, Aufgaben erneut zu versuchen, indem `MaxRetries`.

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

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

Für Informationen darüber, wie HealthOmics verarbeitet Task-Wiederholungen während einer Ausführung, siehe Die Aufgabe wird erneut versucht.

Versuche, Aufgaben erneut zu versuchen, indem **`MaxRetries`**

In der Standardeinstellung HealthOmics versucht keine Wiederholungen einer fehlgeschlagenen Aufgabe oder versucht einen erneuten Versuch, wenn Sie Folgendes konfigurieren `errorStrategy`. Um die maximale Anzahl von Wiederholungen zu erhöhen, setzen Sie `errorStrategy` zu `retry` und konfigurieren Sie die maximale Anzahl von Wiederholungen mit dem `maxRetries` Direktive.

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

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

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

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

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

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

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

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

Deaktivieren Sie die Wiederholung von Aufgaben mit **`ComicsRetryOn5xx`**

Für Nextflow v23 und höher HealthOmics unterstützt Aufgabenwiederholungen, wenn die Aufgabe aufgrund von Dienstfehlern fehlgeschlagen ist (5XX HTTP-Statuscodes). In der Standardeinstellung HealthOmics versucht bis zu zwei Wiederholungen einer fehlgeschlagenen Aufgabe.

Sie können konfigurieren `omicsRetryOn5xx` um die Wiederholung von Aufgaben aufgrund von Servicefehlern abzubestellen. Weitere Informationen zur Aufgabenwiederholung finden Sie unter HealthOmics, siehe Die Aufgabe wird erneut versucht.

Das folgende Beispiel konfiguriert `omicsRetryOn5xx` in der globalen Konfiguration, um die Aufgabenwiederholung abzulehnen.

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

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

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

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

Das folgende Beispiel zeigt, wie Folgendes festgelegt wird `omicsRetryOn5xx` als aufgabenspezifische Konfiguration in der Nextflow-Konfigurationsdatei, basierend auf den Namens- oder Label-Selektoren.

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

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

Konfigurieren Sie den Ressourcenfallback mit ComicsResourceFallbackOrder

HealthOmics unterstützt die omicsResourceFallbackOrder Prozessdirektive zur Definition einer geordneten Liste von Beschleuniger- (oder GPU-) und CPU-Ressourcenprofilen für einen Prozess. Wenn der angeforderte Beschleunigertyp nicht verfügbar ist, HealthOmics wechselt automatisch zum nächsten Profil in der Liste. Sie können eine hinzufügen CPU-only Profil als letzte Option.

Das folgende Beispiel versucht `nvidia-l40s` zuerst, dann `nvidia-l4`, und fällt schließlich auf die CPU zurück, wenn keiner der Beschleuniger verfügbar ist:

```
process ALIGN {
    container 'my-registry/align-multi-arch:latest'

    omicsResourceFallbackOrder([
        [acceleratorType: 'nvidia-l40s', acceleratorCount: 1, cpus: 8, memory: '32 GB',
 omicsResourceWaitTimeoutInMin: 45],
        [acceleratorType: 'nvidia-l4',   acceleratorCount: 1, cpus: 8, memory: '32
 GB'],
        [cpus: 32, memory: '128 GB']
    ])

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

Sie können die Direktive auch in der Nextflow-Konfigurationsdatei festlegen. Weisen Sie der Liste in einer Konfigurationsdatei Folgendes zu =. Sie können es auf der obersten Ebene des einstellen `process` Geltungsbereich, der für jeden Prozess gilt, oder innerhalb eines `withName` oder `withLabel` Selektor. Im folgenden Beispiel wird für jeden Prozess eine Ausweichreihenfolge festgelegt, für Aufgaben mit einer Bezeichnung eine andere `my_label`und eine weitere für die Aufgabe `myTask`.

```
process {
    omicsResourceFallbackOrder = [
        [acceleratorType: 'nvidia-l4', acceleratorCount: 1, cpus: 8, memory: '32 GB'],
        [cpus: 16, memory: '64 GB']
    ]

    withLabel: 'my_label' {
        omicsResourceFallbackOrder = [
            [acceleratorType: 'nvidia-l40s', acceleratorCount: 1, cpus: 8, memory: '32
 GB'],
            [cpus: 32, memory: '128 GB']
        ]
    }

    withName: 'myTask' {
        omicsResourceFallbackOrder = [
            [acceleratorType: 'nvidia-l40s', acceleratorCount: 2, cpus: 16, memory: '64
 GB']
```

```
        ]
    }
}
```

Wenn ein Prozess eine Ersatzbestellung von mehr als einer Stelle erhält, HealthOmics verwendet den ersten, der zutrifft:

1. EIN `withName` oder `withLabel` Selektor in der Konfigurationsdatei, der dem Prozess entspricht.
2. Die Direktive in der Prozessdefinition.
3. Die oberste Ebene der `process` Umfang in der Konfigurationsdatei.

HealthOmics legt den fest AWS_HEALTHOMICS_RESOURCE_TYPE Umgebungsvariable im Aufgabencontainer. Ihr Wert entspricht dem Beschleunigertyp des aktiven Profils (z. B. `nvidia-l40s`) oder `cpu` für ein CPU-only Profil. Verwenden Sie es, um Ihren Befehl zu verzweigen, wenn sich GPU- und CPU-Codepfade unterscheiden.

Beachten Sie Folgendes, wenn Sie die Direktive in Nextflow verwenden:

- Schreiben Sie memory zum Beispiel mit einer der Einheiten B, KB, MB, GB oder TB `'32 GB'`. GiB, MiB und TiB werden nicht akzeptiert.
- Ein Profil, das auslässt cpus oder memory verwendet den Prozess cpus oder memory Direktive, falls gesetzt.
- Wenn ein Profil erbt cpus oder memory aus dem Prozess, HealthOmics wendet den des Prozesses an resourceLimits Direktive, die das Maximum festlegt cpus und memory, bevor Sie den Wert in das Profil kopieren.
- `${task.cpus}` und `${task.memory}` im Skript werden die Werte auf Prozessebene angezeigt, nicht die Ressourcen des Profils, das ausgeführt wurde.

Weitere Informationen finden Sie unter Erweiterte Ressourcenkonfiguration, und die unterstützten Beschleunigertypen finden Sie unter Taskbeschleuniger in einem HealthOmics Workflow-Definition.

Interaktion mit Wiederholungsversuchen

Wenn Sie Wiederholungsversuche mit einer Fehlerstrategie konfigurieren (die `errorStrategy` und `maxRetries` Direktiven), hängt das Profil, das bei jeder Wiederholung verwendet wird, davon ab, warum der vorherige Versuch fehlgeschlagen ist.

- Insufficient-capacity Fehler (ICE). HealthOmics probiert jedes Profil in der Liste der Reihe nach aus. Nachdem alle Profile ausgeschöpft wurden, ohne die Kapazität sicherzustellen, beginnt der nächste Versuch erneut mit dem ersten Profil in Ihrer Direktive.
- Out-of-memory (OOM) und Dienstausfälle. Der Wiederholungsversuch bleibt auf demselben Profil, das beim vorherigen Versuch erreicht wurde, statt mit dem ersten Profil neu zu starten.

Dauer der Aufgabe unter Verwendung des **`Zeit`** Richtlinie

HealthOmics bietet eine einstellbare Quote (siehe HealthOmics Servicekontingente), um die maximale Dauer für einen Lauf anzugeben. Für Workflows von Nextflow v23 und höher können Sie mithilfe von Nextflow auch die maximale Aufgabendauer angeben `time` Direktive.

Bei der Entwicklung neuer Workflows hilft Ihnen die Festlegung der maximalen Aufgabendauer dabei, außer Kontrolle geratene Aufgaben und Aufgaben mit langer Laufzeit abzufangen.

Weitere Informationen zur Nextflow-Zeitrichtlinie finden Sie unter Zeit-Direktive in der Nextflow-Referenz.

HealthOmics bietet die folgende Unterstützung für die Nextflow-Zeitdirektive:

1. HealthOmics unterstützt die 1-Minuten-Granularität für die Zeitdirektive. Sie können einen Wert zwischen 60 Sekunden und dem Wert für die maximale Laufdauer angeben.
2. Wenn Sie einen Wert unter 60 eingeben, HealthOmics rundet den Wert auf 60 Sekunden auf. Für Werte über 60 HealthOmics rundet auf die nächste Minute ab.
3. Wenn der Workflow Wiederholungen für eine Aufgabe unterstützt, HealthOmics wiederholt die Aufgabe, wenn das Zeitlimit überschritten wird.
4. Wenn bei einer Aufgabe ein Timeout auftritt (oder bei der letzten Wiederholung ein Timeout auftritt), HealthOmics bricht die Aufgabe ab. Dieser Vorgang kann eine Dauer von ein bis zwei Minuten haben.
5. Bei einem Task-Timeout HealthOmics setzt die Ausführung und den Aufgabenstatus auf Fehlgeschlagen und storniert die anderen Aufgaben in der Ausführung (für Aufgaben im Status Gestartet, Ausstehend oder Wird ausgeführt). HealthOmics exportiert die Ausgaben von Aufgaben, die vor dem Timeout abgeschlossen wurden, an den von Ihnen angegebenen S3-Ausgabeort.
6. Die Zeit, die eine Aufgabe im Status „Ausstehend“ verbringt, wird nicht auf die Aufgabendauer angerechnet.

7. Wenn die Ausführung Teil einer Ausführungsgruppe ist und für die Ausführungsgruppe das Zeitlimit früher als der Task-Timer überschritten wird, gehen die Ausführung und die Aufgabe in den Status Fehlgeschlagen über.

Geben Sie die Timeout-Dauer mit einer oder mehreren der folgenden Einheiten an: `ms`, `s`, `m`,`h`, oder `d`.

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

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

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

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

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

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

```
process {
    time = '30m'
```

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

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

Verwenden Sie Nextflow-Profile

Nextflow-Profile sind benannte Gruppen von Konfigurationseinstellungen, die Sie zur Laufzeit auswählen können. Definieren Sie Profile in der `profiles` Block Ihres `nextflow.config` datei:

```
profiles {
    standard {
        process.cpus = 2
        process.memory = '4 GB'
    }

    production {
        process.cpus = 16
        process.memory = '64 GB'
        params.input = 's3://bucket/production-data.bam'
    }
}
```

Wenn Sie einen Lauf starten, geben Sie ein oder mehrere Profile mit dem `engineSettings` Parameter. HealthOmics übergibt die `-profile` Flagge an die Nextflow-Engine. Weitere Informationen finden Sie unter Geben Sie die Nextflow-Engine-Einstellungen an.

```
aws omics start-run \
  --workflow-id workflow-id \
  --role-arn role-arn \
  --output-uri s3://bucket/prefix/ \
  --engine-settings '{"profile": "production"}'
```

Wenn mehrere Profile angegeben sind (z. B. `"test,docker"`), wendet Nextflow sie in der Reihenfolge an, in der sie in der Befehlszeile angegeben sind. Spätere Profile überschreiben frühere Profile aufgrund widersprüchlicher Einstellungen. Bei Nextflow-Versionen unter 26 werden Profile

in der Reihenfolge angewendet, in der sie in der Konfigurationsdatei definiert sind, und nicht in der Befehlszeilenreihenfolge.

Beachten Sie Folgendes:

- Profilunterstützung ist für alle verfügbar HealthOmics unterstützte Nextflow-Versionen.
- Profile können Parameter, Prozessrichtlinien enthalten, `includeConfig` Anweisungen und Manifestüberschreibungen (einschließlich `manifest.nextflowVersion`).
- Explizite Ausführungsparameter haben Vorrang vor profildefinierten Parameterwerten.
- Wenn Sie ein nicht vorhandenes Profil angeben, HealthOmics gibt einen Validierungsfehler zurück.
- Profile müssen in der Zip-Datei mit der Workflow-Definition definiert werden. HealthOmics unterstützt das Abrufen von Profildefinitionen aus externen Quellen nicht.
- Wenn Sie kein Profil angeben, verwendet der Lauf das `standard` Profil, wenn es in der Workflow-Definition unter Profilen definiert ist. Andernfalls verwendet der Lauf die Standardkonfiguration (oberste Ebene).
- Wenn Sie Profile verwenden, empfehlen wir, die Nextflow-Version in Ihrer Workflow-Definition anzuheften mit `manifest.nextflowVersion` um ein konsistentes Verhalten der Profilanwendung bei allen Läufen sicherzustellen.

Exportieren Sie Inhalte auf Workflow-Ebene

Für Nextflow v25.10 und höher können Sie Dateien exportieren, die außerhalb einzelner Aufgaben erstellt wurden, wie z. B. Herkunftsberichte oder Pipeline-DAGs. Um diese Dateien zu exportieren, schreiben Sie sie in `/mnt/workflow/output/`. HealthOmics exportiert Dateien, die sich in diesem Verzeichnis befinden, in den `output/` Präfix im Amazon S3-Ausgabespeicherort Ihres Laufs.

Das folgende Beispiel zeigt, wie Sie das konfigurieren `nf-prov` Plugin zum Schreiben eines Provenienzberichts `/mnt/workflow/output/`.

```
prov {
    formats {
        bco {
            file = "/mnt/workflow/output/pipeline_info/manifest.bco.json"
        }
    }
}
```

Sie können diesen Pfad auch als Parameter in der JSON-Eingabe Ihres Laufs übergeben. Dieser Ansatz ist bei NF-Core-Workflows üblich, die Folgendes verwenden `params.outdir`.

```
{
    "outdir": "/mnt/workflow/output/"
}
```

Aufgabeninhalt exportieren

Definieren Sie für in Nextflow geschriebene Workflows eine PublishDir Direktive zum Exportieren von Aufgabeninhalten in Ihren Amazon S3-Ausgabe-Bucket. Stellen Sie, wie im folgenden Beispiel gezeigt, den PublishDir Wert bis `/mnt/workflow/pubdir`. Um Dateien nach Amazon S3 zu exportieren, müssen sich die Dateien in diesem Verzeichnis befinden.

```
nextflow.enable.dsl=2

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

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

   publishDir "/mnt/workflow/pubdir"

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

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

   script:
   """
       set -eo pipefail
```

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

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

    publishDir "/mnt/workflow/pubdir"

    input:
        file input_bam

    output:
        path "validation_report"

    script:
    """
        java -Xmx3G -jar /usr/gitc/picard.jar \
        ValidateSamFile \
        INPUT=${input_bam} \
        OUTPUT=validation_report \
        MODE=SUMMARY \
        IS_BISULFITE_SEQUENCED=false
    """
  }
```

Für Nextflow v25.10 und höher, als Alternative zu `publishDir`, Sie können Workflow-Ausgaben verwenden, um Aufgabeninhalte zu exportieren. Das folgende Beispiel zeigt, wie ein Workflow definiert wird `output` Block, der Aufgabenergebnisse nach Amazon S3 exportiert.

```
process myTask {
    input:
    val data

    output:
    path 'result.txt'

    script:
    """
    echo ${data} > result.txt
```

```
    """
}

workflow {
    main:
    output_file = myTask('hello')

    publish:
    results = output_file
}

output {
    results {
        path '.'
    }
}
```

Weitere Informationen zu Workflow-Ausgaben finden Sie unter Workflow-Ausgaben in der Nextflow-Dokumentation.

Generieren Sie Nextflow-Ausführungsberichte

Nextflow kann für jeden Lauf vier integrierte Berichte erstellen: einen Ausführungsbericht (`report`), eine Zeitleiste (`timeline`), eine Trace-Datei (`trace`) und ein Workflow-Diagramm (`dag`). Für HealthOmics Um diese Dateien an den Amazon S3-Ausgabespeicherort Ihres Laufs zu exportieren, konfigurieren Sie jede Datei so, dass ihre Ausgabe unter geschrieben wird `/mnt/workflow/output/` in deinem `nextflow.config` datei:

```
report {
    enabled = true
    file = '/mnt/workflow/output/report.html'
    overwrite = true
}

timeline {
    enabled = true
    file = '/mnt/workflow/output/timeline.html'
    overwrite = true
}

trace {
    enabled = true
```

```
    file = '/mnt/workflow/output/trace.txt'
    overwrite = true
}

dag {
    enabled = true
    file = '/mnt/workflow/output/dag.html'
    overwrite = true
}
```

HealthOmics exportiert Dateien, die unter geschrieben wurden `/mnt/workflow/output/` zum `output/` Präfix im Amazon S3-Ausgabeort Ihres Laufs. Weitere Informationen zu diesem Exportpfad finden Sie unter Exportieren Sie Inhalte auf Workflow-Ebene. Berichte, die draußen geschrieben wurden `/mnt/workflow/output/` werden nicht an den Amazon S3-Ausgabespeicherort Ihres Laufs exportiert.

> **Die Aufgabencontainer müssen Folgendes enthalten `ps`**
>
> Wann der `report`, `timeline`, oder `trace` Wenn der Bericht aktiviert ist, erfasst Nextflow Kennzahlen pro Aufgabe durch Aufrufen `ps` in jedem Aufgabencontainer. Das Container-Image, das Sie mit dem angeben `container` Die Direktive muss Folgendes enthalten `ps` Befehl. Installieren Sie ihn auf den meisten Linux-Distributionen mit dem `procps` (Debian/Ubuntu) oder `procps-ng` Paket (Amazon Linux, Red Hat, Fedora). Wenn ein Prozess keinen deklariert `container` Richtlinie, HealthOmics führt die Aufgabe in einem Standardcontainer aus, der bereits Folgendes enthält `ps`.

> **Format des Workflow-Diagramms**
>
> Das `dag` Der Bericht unterstützt mehrere Ausgabeformate, ausgewählt durch die Erweiterung von `dag.file`. Die Formate HTML, Mermaid und DOT werden direkt von Nextflow gerendert und erfordern keine zusätzlichen Tools. Für die Formate PDF, PNG und SVG ist Graphviz erforderlich, das nicht enthalten ist HealthOmicsDie Nextflow-Engine. Wenn `dag.file` auf einen PDF-, PNG- oder SVG-Pfad gesetzt ist, protokolliert Nextflow eine Warnung und schreibt das Workflow-Diagramm als `.dot` Quelldatei an ihrer Stelle; der Lauf wird trotzdem erfolgreich abgeschlossen. Wir empfehlen die Einstellung `dag.file` zu einem `.html`, `.mmd`, oder `.dot` Pfad, um die Warnung zu vermeiden und das angeforderte Format zu erzeugen.

Geben Sie die Nextflow-Syntaxversion an

Nextflow v26.04.0 verwendet standardmäßig den strikten Syntaxparser (v2). Dies ist eine grundlegende Änderung für Workflows, die mit der Legacy-Syntax (v1) geschrieben wurden, was in Nextflow v25.10.0 und früheren Versionen die Standardeinstellung ist. Hinweise zur v2-Syntax finden Sie unter Strikte Syntax in der Seqera Nextflow-Dokumentation.

Um einen Workflow auszuführen, der für den Legacy-Parser (v1) erstellt wurde, setzen Sie `engineSettings.syntaxVersion` zu `v1` in der StartRun Anfrage:

```
{
  "engineSettings": {
    "syntaxVersion": "v1"
  }
}
```

Für Nextflow v25.10.0 und früher HealthOmics unterstützt den v2-Parser nicht.

Automatische Syntaxvalidierung bei der Workflow-Erstellung

HealthOmics führt automatisch den in Nextflow integrierten Strict DSL2-Linter aus (nf-lang/v2), wenn Sie einen Nextflow DSL2-Workflow erstellen oder aktualisieren. Dieser Linter läuft während `CreateWorkflow` und `CreateWorkflowVersion`. Es gilt für alle unterstützten DSL2-Versionen (v22.04, v23.10, v24.10, v25.10 und v26.04). DSL1-Workflows sind nicht überlastet.

Der Linter arbeitet im nicht blockierenden Modus. Lint-Ergebnisse verhindern nicht, dass der Workflow AKTIV wird. Die Ergebnisse erscheinen als strukturiertes JSON in der `statusMessage` Feld des `GetWorkflow` Antwort.

> **Note**
>
> Der integrierte Linter validiert die Syntax Ihrer Workflow-Definition bei der Erstellung. Sie unterscheidet sich von der strikten Syntax Parser verfügbar für Nextflow v26.04, das gesteuert wird von `engineSettings.syntaxVersion` und beeinflusst das Laufzeitverhalten. Der Linter überprüft bei der Erstellung die Syntax aller DSL2-Versionen, unabhängig davon, welchen Parser der Workflow zur Laufzeit verwendet. Bei Nextflow v22.04, v23.10 und v24.10 (Legacy-Grammatik) sind die Ergebnisse beratend. In Nextflow v25.10 und v26.04 spiegeln die Ergebnisse strenge Anforderungen an die Modussyntax wider.

Weitere Informationen zum Lint-Ausgabeformat und zur Behebung von Ergebnissen finden Sie unter Der Arbeitsablauf zeigt an HealthOmics.

Effiziente Nutzung von Scratch Storage in Nextflow

Nextflows `scratch` Die Direktive steuert, wo ein Prozess seine temporären Arbeitsdateien schreibt. Wenn der kurzlebige Speicher aktiviert ist (`scratchStorageMode: LOCAL`), benutze den `scratch` Direktive zum direkten Kratzen I/O auf die schnelle lokale Lautstärke bei `/tmp`.

Die folgende Tabelle beschreibt die unterstützten `scratch` Direktivenwerte und ihr Verhalten in HealthOmics:

Wert	Verhalten in HealthOmics	Empfehlung
`scratch true`	Nutzt `$TMPDIR`. Kratzen I/O wird auf das lokale kurzlebige Volumen gerichtet, wenn `scratchStorageMode` ist `LOCAL`.	Empfohlen
`scratch '/some/path'`	Verwendet den angegebenen Literalpfad als Scratch-Verzeichnis. Um kurzlebigen Speicher zu verwenden, setzen Sie den Pfad auf `/tmp` oder ein Unterverzeichnis von `/tmp`. Der Pfad muss im Container existieren und beschreibbar sein.	Funktioniert, wenn der Pfad unter `/tmp`
`scratch 'ram-disk'`	Versuche zu verwenden `/dev/shm` (tmpfs im RAM). Dies wird nicht für lokalen Scratch-Speicher in HealthOmics empfohlen.	Nicht empfohlen

Der empfohlene Ansatz besteht darin, Folgendes festzulegen `scratch true` in Ihrer Prozessdefinition, die automatisch verwendet `$TMPDIR` und erfordert keine Pfadkonfiguration:

```
process my_process {
    scratch true
    disk '200 GB'
    script:
    """
    my-tool --input ${input} --output ${output}
    """
}
```

Für weitere Informationen über kurzlebige Speicherung und `disk` Richtlinie finden Sie unter Kurzlebiger Speicher für Workflow-Aufgaben HealthOmics.

Besonderheiten der CWL-Workflow-Definition

In Common Workflow Language (CWL) geschriebene Workflows bieten ähnliche Funktionen wie in WDL und Nextflow geschriebene Workflows. Sie können Amazon S3 verwenden oder HealthOmics Speicher-URIs als Eingabeparameter.

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

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

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

Weitere Informationen zur Validierung einer CWL-Definition vor der Erstellung des Workflows finden Sie unter Workflow-Definitionen lokal validieren.

Themen

- CWL-Workflows zur Verwendung konvertieren HealthOmics
- Deaktivieren Sie die Wiederholung von Aufgaben mit ComicsRetryOn5xx
- Einen Workflow-Schritt wiederholen
- Versuchen Sie Aufgaben mit erhöhtem Arbeitsspeicher erneut
- Beispiele

CWL-Workflows zur Verwendung konvertieren HealthOmics

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

- Ersetzen Sie alle Docker-Container-URIs durch Amazon ECR-URIs.
- Stellen Sie sicher, dass alle Workflow-Dateien im Haupt-Workflow als Eingabe deklariert sind und dass alle Variablen explizit definiert sind.
- Stellen Sie sicher, dass alle JavaScript Beim Code handelt es sich um eine Strict-Mode-Beschwerde.

Deaktivieren Sie die Wiederholung von Aufgaben mit **`ComicsRetryOn5xx`**

HealthOmics unterstützt Aufgabenwiederholungen, wenn die Aufgabe aufgrund von Dienstfehlern fehlgeschlagen ist (5XX HTTP-Statuscodes). In der Standardeinstellung HealthOmics versucht bis zu zwei Wiederholungen einer fehlgeschlagenen Aufgabe. Weitere Informationen zur Aufgabenwiederholung finden Sie unter HealthOmics, siehe Die Aufgabe wird erneut versucht.

Um die Wiederholung von Aufgaben aufgrund von Servicefehlern abzubestellen, konfigurieren Sie das `omicsRetryOn5xx` Direktive in der Workflow-Definition. Sie können diese Direktive unter Anforderungen oder Hinweise definieren. Wir empfehlen, die Richtlinie als Hinweis für die Portabilität hinzuzufügen.

```
requirements:
  ResourceRequirement:
    omicsRetryOn5xx: false

hints:
  ResourceRequirement:
    omicsRetryOn5xx: false
```

Die Anforderungen haben Vorrang vor Hinweisen. Wenn eine Aufgabenimplementierung in Hinweisen eine Ressourcenanforderung vorsieht, die auch durch Anforderungen in einem umschließenden Workflow bereitgestellt wird, haben die einschließenden Anforderungen Vorrang.

Wenn dieselbe Aufgabenanforderung auf verschiedenen Ebenen des Workflows auftaucht, HealthOmics verwendet den spezifischsten Eintrag von `requirements` (oder `hints`, wenn es keine Einträge in `requirements`). Die folgende Liste zeigt die Rangfolge, die HealthOmics verwendet, um Konfigurationseinstellungen von der niedrigsten zur höchsten Priorität anzuwenden:

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

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

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

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

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

Einen Workflow-Schritt wiederholen

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

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

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

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

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

Versuchen Sie Aufgaben mit erhöhtem Arbeitsspeicher erneut

HealthOmics unterstützt die automatische Wiederholung von Aufgabenfehlern, die nicht über genügend Arbeitsspeicher verfügen. Wenn eine Aufgabe mit dem Code 137 (nicht genügend Arbeitsspeicher) beendet wird, HealthOmics erstellt eine neue Aufgabe mit erhöhter Speicherzuweisung auf der Grundlage des angegebenen Multiplikators.

> **Note**
>
> HealthOmics wiederholt Fehler aufgrund unzureichenden Speichers bis zu dreimal oder bis die Speicherzuweisung 1536 GiB erreicht, je nachdem, welcher Grenzwert zuerst erreicht wird.

Das folgende Beispiel zeigt, wie Wiederholungsversuche bei unzureichendem Arbeitsspeicher konfiguriert werden:

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

Wenn eine Aufgabe aufgrund von zu wenig Arbeitsspeicher fehlschlägt, HealthOmics berechnet die Speicherzuweisung für Wiederholungsversuche anhand der folgenden Formel: `previous_run_memory × memoryRetryMultiplier`. Wenn im obigen Beispiel die Aufgabe mit 4096 MB Arbeitsspeicher fehlschlägt, verbraucht der Wiederholungsversuch 4096 × 2,5 = 10.240 MB Speicher.

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

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

Beispiele

Das Folgende ist ein Beispiel für einen in CWL geschriebenen Workflow.

```

cwlVersion: v1.2
class: Workflow

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

out_filename: string
docker_image: string
```

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

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

Die folgende Datei definiert `copy.cwl` Aufgabe.

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

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

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

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

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

Das Folgende ist ein Beispiel für einen in CWL geschriebenen Workflow mit einer GPU-Anforderung.

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

inputs: []
outputs: []
```

Beispiele für Workflow-Definitionen

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

WDL

```
version 1.1

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

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

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

workflow my_workflow {
   inputs {
       File input_file
       String name
       Int threshold = 50
   }

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

Nextflow

```
nextflow.enable.dsl = 2

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

process my_task {
   // <directives>

   input:
     path input_file
     val name
     val threshold
```

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

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


}

workflow MY_WORKFLOW {
   my_task(
       params.input_file,
       params.name,
       params.threshold
   )
}

workflow {
   MY_WORKFLOW()
}
```

CWL

```
cwlVersion: v1.2
class: Workflow

requirements:
    InlineJavascriptRequirement: {}

inputs:
   input_file: File
   name: string
   threshold: int

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

```
steps:
    my_task:
        run:
            class: CommandLineTool
            baseCommand: my_tool
            requirements:
                ...
            inputs:
                name:
                    type: string
                    inputBinding:
                        prefix: "--name"
                threshold:
                    type: int
                    inputBinding:
                        prefix: "--threshold"
                input_file:
                    type: File
                    inputBinding: {}
            outputs:
                results:
                    type: File
                    outputBinding:
                        glob: results.txt
```

Parametervorlagendateien für HealthOmics Workflows

Parametervorlagen definieren die Eingabeparameter für einen Workflow. Sie können Eingabeparameter definieren, um Ihren Workflow flexibler und vielseitiger zu gestalten. Sie können beispielsweise einen Parameter für den Amazon S3 S3-Speicherort der Referenzgenomdateien definieren. Parametervorlagen können über einen Git-basierten Repository-Dienst oder Ihr lokales Laufwerk bereitgestellt werden. Benutzer können den Workflow dann mit verschiedenen Datensätzen ausführen.

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

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

Das Eingabeparameterobjekt umfasst die folgenden Attribute:

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

Die folgende Beispielparametervorlage zeigt, wie die Eingabeparameter angegeben werden.

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

Generieren von Parametervorlagen

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

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

Themen

- Parametererkennung für CWL
- Parametererkennung für WDL

- Parametererkennung für Nextflow

Parametererkennung für CWL

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

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

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

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

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

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

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

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

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

Eingabe	Beispiel für Eingabe/Ausgabe	Erforderlich

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

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

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

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

Parametererkennung für WDL

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

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

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

  - Wenn den Eingabeparametern keine Werte oder Ausdrücke zugewiesen wurden, werden sie als erforderliche Parameter markiert.

- Beschreibungen werden aus `parameter_meta` der `main` Workflow-Definition extrahiert. Wenn nicht angegeben, `parameter_meta` ist die Beschreibung leer (eine leere Zeichenfolge). Weitere Informationen finden Sie in der WDL-Spezifikation für Parameter-Metadaten.

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

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

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

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

Eingabe	Beispiel für Eingabe/Ausgabe	Erforderlich		
Array [Int] x	[1,2,3] oder []	Ja		
Reihe [Int] + x	[1], aber nicht []	Ja		

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

Parametererkennung für Nextflow

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

Themen

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

Analysieren der Schemadatei

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

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

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

- Extrahiert allesproperties, was im Schema definiert ist.
- Schließt die Eigenschaft eindescription, sofern sie für die Eigenschaft verfügbar ist.
- Identifiziert, ob jeder Parameter optional oder erforderlich ist, basierend auf dem required Feld der Eigenschaft.

Das folgende Beispiel zeigt eine Definitionsdatei und die generierte Parameterdatei.

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

```
    "$defs": {
        "input_options": {
            "title": "Input options",
            "type": "object",
            "required": ["input_file"],
            "properties": {
                "input_file": {
                    "type": "string",
                    "format": "file-path",
                    "pattern": "^s3://[a-z0-9.-]{3,63}(?:/\\S*)?$",
                    "description": "description for input_file"
                },
                "input_num": {
                    "type": "integer",
                    "default": 42,
                    "description": "description for input_num"
                }
            }
        },
        "output_options": {
            "title": "Output options",
            "type": "object",
            "required": ["output_dir"],
            "properties": {
                "output_dir": {
                    "type": "string",
                    "format": "file-path",
                    "description": "description for output_dir",
                }
            }
        }
    },
    "properties": {
        "ungrouped_input_bool": {
            "type": "boolean",
            "default": true
        }
    },
    "required": ["ungrouped_input_bool"],
    "allOf": [
        { "$ref": "#/$defs/input_options" },
        { "$ref": "#/$defs/output_options" }
    ]
```

```
}
```

Die generierte Parametervorlage:

```
{
    "input_file": {
        "description": "description for input_file",
        "optional": False
    },
    "input_num": {
        "description": "description for input_num",
        "optional": True
    },
    "output_dir": {
        "description": "description for output_dir",
        "optional": False
    },
    "ungrouped_input_bool": {
        "description": None,
        "optional": False
    }
}
```

Analysieren der Hauptdatei

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

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

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

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

Das folgende Beispiel zeigt eine Definitionsdatei und die generierte Parametervorlage.

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

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

Die generierte Parametervorlage:

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

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

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

params {
    gamma = "gamma"
```

```
    delta = "delta"
}

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

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

Die generierte Parametervorlage:

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

Verschachtelte Parameter

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

Verschachtelte Parameter in Schemadateien

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

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

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

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

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

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

Verschachtelte Parameter in Konfigurationsdateien

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

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

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

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

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

Beispiele für Nextflow-Interpolation

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

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

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

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

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

Container-Images für private Workflows

Wenn Sie einen Workflow erstellen, können Sie die Container-Registrierungen und Images angeben, die für Ihren Workflow erforderlich sind. HealthOmics führt die Container-Images aus, die Ihre Workflow-Aufgaben verwenden. Sie können Bilder aus Ihren eigenen privaten Amazon ECR-Repositorys und aus Container-Registern von Drittanbietern verwenden. HealthOmics unterstützt Container-Images von Amazon ECR Public, Docker Hub, Quay, GitHub Container-Registry,

Kubernetes-Container-Image-Registrierung, Microsoft Azure Container Registry, GitLab Container Registry, Seqera Containers, NVIDIA NGC, Google Artifact Registry und Google Container Registry.

Sie können eine oder mehrere Container-Registries und Images für Ihren Workflow auswählen.

- Verwenden Sie Ihre eigenen Amazon ECR-Bilder: Sie können auf Bilder verweisen, die Sie in privaten Amazon ECR-Repositorys hosten. Informationen zum Erstellen Ihres eigenen privaten Repositorys finden Sie unter Ein privates Amazon ECR-Repository erstellen in der Amazon Elastic Container Registry — Benutzerhandbuch.

  Sie können diese Bilder direkt in Ihrer Workflow-Definition referenzieren, sie als Workflow-Parameter übergeben oder sie Bildzuordnungen zuordnen. Weitere Informationen finden Sie unter Bildzuordnungen.
- Verwenden Sie Registrierungen von Drittanbietern, die Cache-Unterstützungen abrufen: Sie können Bilder aus Registern von Drittanbietern abrufen, die Pull Through Cache (PTC) - Unterstützung bieten, wie Docker Hub, Quay, GitHub Container Registry und Amazon ECR Public. HealthOmics synchronisiert sie automatisch. Weitere Informationen finden Sie unter Verwendung von Drittanbieter-Registern, die vom Pull-Through-Cache unterstützt werden.
- Verwenden Sie Registrierungen von Drittanbietern, die das Durchziehen des Caches nicht unterstützen: Sie können auch Container-Images aus Registern verwenden, die PTC nicht unterstützt, wie Seqera, NVIDIA NGC und Google Container Registry oder Google Artifact Registry. Weitere Informationen finden Sie unter Verwendung von Drittanbieter-Registrierungen, die vom Pull-Through-Cache nicht unterstützt werden.

Allgemeine Überlegungen zu Amazon ECR-Container-Images

- Region — Ihr Amazon ECR-Repository befindet sich in derselben Region AWS Region wie das Konto, das den Service aufruft.
- Cross-account Bilder — Ein anderes AWS-Konto kann das Container-Image besitzen, sofern das Quell-Image-Repository die entsprechenden Berechtigungen bereitstellt. Weitere Informationen finden Sie unter Richtlinien für den kontoübergreifenden Amazon ECR-Zugriff.
- Image-URIs — Wir empfehlen Ihnen, die Uniform Resource Identifiers (URIs) Ihres Amazon ECR-Container-Images als Parameter in Ihrem Workflow zu definieren, damit der Zugriff vor Beginn der Ausführung überprüft werden kann. Dadurch wird es auch einfacher, einen Workflow in einer neuen Region auszuführen, indem Sie den Parameter Region ändern.

- Berechtigungen — Ihre Läufe benötigen die Erlaubnis, Amazon ECR während der Ausführung zu erreichen. Sie konfigurieren IAM-Berechtigungen für HealthOmics um auf Amazon ECR zuzugreifen. Weitere Informationen finden Sie unter HealthOmics Berechtigungen für Ressourcen. Lesen Sie auch die folgenden Themen:
  - Für Registries, die Cache-Unterstützungen per Pull-Through-Cache verwenden: Richtlinien für Amazon ECR ziehen den Cache durch
  - Für Registries, die den Cache durchziehen, wird Folgendes nicht unterstützt: Richtlinie zur Dienstrolle für die Unterstützung von Containern, die nicht von PTC stammen
- Architektur — HealthOmics unterstützt x86_64-Container und unterstützt keine ARM-Container. Wenn Ihr lokaler Computer ARM-based, z. B. Apple Mac, verwenden Sie einen Befehl wie den folgenden, um ein x86_64-Container-Image zu erstellen:

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

- Entrypoint und Shell — HealthOmics Workflow-Engines fügen Bash-Skripte als Befehlsüberschreibung in die Container-Images ein, die von Workflow-Aufgaben verwendet werden. Daher sollten Container-Images ohne einen bestimmten ENTRYPOINT erstellt werden, sodass eine Bash-Shell die Standardeinstellung ist.
- Bereitgestellte Pfade — HealthOmics mountet ein gemeinsam genutztes Dateisystem für Container-Aufgaben unter `/tmp`. HealthOmics überschreibt alle Daten oder Tools, die an dieser Stelle in das Container-Image integriert sind.

  Die Workflow-Definition ist für Aufgaben über einen schreibgeschützten Mount unter verfügbar `/mnt/workflow`.
- Bildgröße — Siehe HealthOmics feste Größenkontingente für Workflows für die maximalen Container-Bildgrößen.
- Preisgestaltung — Für die Bilder in Ihren Repositorys fallen die standardmäßigen Amazon ECR-Speichergebühren an. Für Upstream-Registrierungen, für die Zugangsdaten erforderlich sind, fallen auch Secrets Manager-Gebühren an.
- Bewährte Verfahren — Lesen Sie die Anleitungen zu bewährten Verfahren unter Bewährte Methoden für Container-Images.

Themen

- Verwendung von Drittanbieter-Registern, die vom Pull-Through-Cache unterstützt werden

- Verwendung von Drittanbieter-Registrierungen, die vom Pull-Through-Cache nicht unterstützt werden
- Hinzufügen von privaten Containern oder Containern von Drittanbietern während der Workflow-Erstellung
- Bewährte Methoden für Container-Images

Verwendung von Drittanbieter-Registern, die vom Pull-Through-Cache unterstützt werden

Sie können Amazon ECR-Pull-Through-Cache-Regeln verwenden, um Repositorys in einer unterstützten Upstream-Registry mit Ihren privaten Amazon ECR-Repositorys zu synchronisieren. Weitere Informationen finden Sie unter Synchronisieren Sie eine Upstream-Registrierung in der Amazon Elastic Container Registry — Benutzerhandbuch.

Wenn ein Bild zum ersten Mal abgerufen wird, erstellt der Pull-Through-Cache das Image-Repository in Ihrer privaten Registry und synchronisiert es mit dem Upstream-Image, solange die Image-Tags des Repositorys veränderbar bleiben.

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

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

Für Registries, die den Cache mit Pull-Through-Cache nicht unterstützen, HealthOmics kopiert die Bilder in Ihre privaten Amazon ECR-Repositorys. Weitere Informationen finden Sie unter Verwendung von Drittanbieter-Registrierungen, die vom Pull-Through-Cache nicht unterstützt werden.

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

1. Sie müssen keine Container-Images manuell zu Amazon ECR migrieren oder Updates aus dem Drittanbieter-Repository synchronisieren.

2. Workflows greifen auf die synchronisierten Container-Images in Ihrem privaten Repository zu. Das ist zuverlässiger als das Herunterladen von Inhalten zur Laufzeit aus einem öffentlichen Register.
3. Da Amazon ECR Pull-Through-Caches eine vorhersehbare URI-Struktur verwendet, können Sie die Upstream-Image-URIs in Ihrer Workflow-Definition beibehalten und eine Container-Registry-Map hinzufügen, die sie in Ihren Cache-Repositorys auflöst. Weitere Informationen finden Sie unter Schritt 4: Hinzufügen einer Container-Registry Map (wenn Ihr Workflow Upstream-URIs verwendet).

Themen

- Voraussetzungen
- Pull-Through-Cache konfigurieren
- Re-running die Konfiguration
- Problembehandlung beim Pull-Through-Cache

Voraussetzungen

Wenn eine Upstream-Registrierung eine Authentifizierung erfordert, AWS Secrets Manager Das Geheimnis muss im selben enthalten sein AWS-Konto und Region als Pull-Through-Cache-Regel. Wenn Sie die Regel erstellen, übergeben Sie `--credential-arn` Der ARN des Secrets genau so, wie Secrets Manager ihn zurückgibt, einschließlich des Suffixes, das Secrets Manager an den Namen anhängt. Der Secrets Manager create-secret und describe-secret Antworten geben es beide zurück. Die Beispiel-ARNs in diesem Thema kürzen den geheimen Namen ab.

Upstream-Registrierung	URL der Registrierung	Secrets Manager-Schlüssel erforderlich
Amazon ECR Public	`public.ecr.aws`	Nein
Kubernetes-Container-Image-Registrierung	`registry.k8s.io`	Nein
Quay	`quay.io`	Nein
Docker Hub	`registry-1.docker.io`	Ja

Upstream-Registrierung	URL der Registrierung	Secrets Manager-Schlüssel erforderlich
Microsoft Azure Container Registry	*registry-name* .azurecr.io	Ja
GitHub Container-Registrierung	ghcr.io	Ja
GitLab Container-Registrierung	registry.gitlab.com	Ja

Der geheime Name muss beginnen mit `ecr-pullthroughcache/`. Die anderen Anforderungen finden Sie unter Speichern Ihrer Upstream-Repository-Anmeldeinformationen in einem AWS Secrets Manager Geheimnis in der Amazon Elastic Container Registry — Benutzerhandbuch.

Pull-Through-Cache konfigurieren

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

Führen Sie die folgenden Konfigurationsaufgaben der Reihe nach aus. Schritt 4 gilt nur, wenn Ihre Workflow-Definition die Upstream-Image-URIs verwendet.

Aufgaben zur Konfiguration

- Schritt 1: Erstellen einer Pull-Through-Cache-Regel
- Schritt 2: Berechtigungen für die Upstream-Registrierung einrichten
- Schritt 3: Repository-Vorlagen erstellen
- Schritt 4: Hinzufügen einer Container-Registry Map (wenn Ihr Workflow Upstream-URIs verwendet)
- Schritt 5: Überprüfen Sie Ihre Pull-Through-Cache-Konfiguration

Schritt 1: Erstellen einer Pull-Through-Cache-Regel

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

Für eine Upstream-Registrierung, die eine Authentifizierung erfordert, geben Sie Ihre Anmeldeinformationen wie folgt an AWS Secrets Manager.

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

Weitere Informationen finden Sie unter Eine Pull-Through-Cache-Regel erstellen in der Amazon Elastic Container Registry — Benutzerhandbuch.

Erstellen einer Pull-Through-Cache-Regel mithilfe der Konsole

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

Teil 1: Auswahl der Upstream-Registrierung

1. Öffnen Sie die Amazon-ECR-Konsole.
2. Im Navigationsbereich unter Privates Register, erweitern Funktionen und Einstellungen, dann wähle Durchsuche den Cache.
3. Auf der Durchsuche den Cache Seite, wähle Regel hinzufügen.
4. In der Upstream-Registrierung Panel, wählen Sie die Upstream-Registrierung aus, die mit Ihrer privaten Registrierung synchronisiert werden soll, und wählen Sie dann Weiter.
5. Wenn für die Upstream-Registrierung eine Authentifizierung erforderlich ist, geben Sie auf der nächsten Seite das Secrets Manager-Geheimnis an, das Ihre Anmeldeinformationen enthält. Wählen Sie Weiter aus.

Teil 2: Namespaces angeben und die Regel erstellen

1. Unter Geben Sie Namespaces an, in der Namespace zwischenspeichern Wählen Sie im Bedienfeld, ob die privaten Repositorys mit einem bestimmten Repository-Präfix oder ohne Präfix erstellt werden sollen. Wenn Sie ein Präfix verwenden möchten, geben Sie den Präfixnamen in Repository-Präfix zwischenspeichern.

2. In der Upstream-Namespace Wählen Sie im Bedienfeld aus, ob Sie mit einem bestimmten Repository-Präfix oder ohne Präfix aus Upstream-Repositorys abrufen möchten. Wenn Sie ein Präfix verwenden möchten, geben Sie den Präfixnamen in Präfix für das Upstream-.

   Das Beispiel für einen Namespace Das Panel zeigt einen Beispiel-Pull-Request, eine Upstream-URL und die URL des Cache-Repositorys, das erstellt wird.

3. Wählen Sie Weiter aus.
4. Überprüfen Sie die Konfiguration und wählen Sie Erstellen um die Regel zu erstellen.

Erstellen einer Pull-Through-Cache-Regel mit der AWS CLI

Verwenden Sie den Amazon ECR create-pull-through-cache-rule Befehl zum Erstellen einer Pull-Through-Cache-Regel. Speichern Sie die Anmeldeinformationen für Upstream-Registrierungen, für die eine Authentifizierung erforderlich ist, in einem Secrets Manager-Schlüssel.

Die folgenden Beispiele zeigen, wie eine Regel für jede unterstützte Upstream-Registrierung erstellt wird.

Amazon ECR Public

Im folgenden Beispiel wird eine Pull-Through-Cache-Regel für die öffentliche Registrierung von Amazon ECR erstellt. Sie gibt ein Repository-Präfix von an `ecr-public`, sodass jedes Repository, das die Regel erstellt, das Benennungsschema verwendet `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
```

Kubernetes container image registry

Im folgenden Beispiel wird eine Pull-Through-Cache-Regel für das öffentliche Kubernetes-Registry erstellt. Es spezifiziert ein Repository-Präfix von `kubernetes`, sodass jedes Repository, das die Regel erstellt, das Benennungsschema verwendet `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
```

Quay

Im folgenden Beispiel wird eine Pull-Through-Cache-Regel für die öffentliche Registrierung von Quay erstellt. Es spezifiziert ein Repository-Präfix von `quay`, sodass jedes Repository, das die Regel erstellt, das Benennungsschema verwendet `quay/`*`upstream-repository-name`*.

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

Docker Hub

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

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

GitHub Container Registry

Das folgende Beispiel erstellt eine Pull-Through-Cache-Regel für GitHub Container-Registrierung. Es gibt ein Repository-Präfix von an `github`, sodass jedes Repository, das die Regel erstellt, das Benennungsschema verwendet `github/`*`upstream-repository-name`*. Sie müssen den vollständigen Amazon-Ressourcennamen (ARN) des Secrets angeben, das Ihre enthält GitHub Anmeldeinformationen für Container Registry.

```
aws ecr create-pull-through-cache-rule \
    --ecr-repository-prefix github \
```

```
    --upstream-registry-url ghcr.io \
    --credential-arn arn:aws:secretsmanager:us-east-1:111122223333:secret:ecr-
pullthroughcache/example1234 \
    --region us-east-1
```

Microsoft Azure Container Registry

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

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

GitLab Container Registry

Das folgende Beispiel erstellt eine Pull-Through-Cache-Regel für GitLab Container-Registrierung. Es gibt ein Repository-Präfix von an `gitlab`, sodass jedes Repository, das die Regel erstellt, das Benennungsschema verwendet `gitlab/`*`upstream-repository-name`*. Sie müssen den vollständigen Amazon-Ressourcennamen (ARN) des Secrets angeben, das Ihre enthält GitLab Anmeldeinformationen für Container Registry.

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

Sie können die verwenden get-run-task AWS CLI Befehl zum Abrufen von Informationen über das Container-Image, das für eine bestimmte Aufgabe verwendet wurde:

```
aws omics get-run-task --id run-id --task-id task-id
```

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

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

Schritt 2: Berechtigungen für die Upstream-Registrierung einrichten

Verwenden Sie Registrierungsberechtigungen, um Folgendes zuzulassen HealthOmics um den Pull-Through-Cache zu verwenden und die Container-Images in die private Amazon ECR-Registry abzurufen. Fügen Sie der Registrierung eine Amazon ECR-Registrierungsrichtlinie hinzu, die die in den Läufen verwendeten Container bereitstellt.

Die folgende Richtlinie gewährt die Genehmigung für HealthOmics Dienst, um Repositorys mit den angegebenen Pull-Through-Cache-Präfixen zu erstellen und Upstream-Pulls in diese Repositorys zu initiieren.

Stellen Sie sicher `Resource` deckt jedes Cache-Repository-Präfix ab, das Sie erstellt haben, und legen Sie die Region und die AWS-Konto in jedem Repository-ARN für die Registrierung, die Sie konfigurieren, da eine Registrierungsrichtlinie nur für ihre eigene Region und ihr eigenes Konto gilt.

1. Öffnen Sie die Amazon-ECR-Konsole.
2. Im Navigationsbereich unter Privates Register, erweitern Registrierungsberechtigungen, dann wähle Aussage generieren.
3. Wählen Sie oben rechts JSON aus. Geben Sie eine Richtlinie ein, die der folgenden ähnelt:

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPTCinRegPermissions",
            "Effect": "Allow",
            "Principal": {
                "Service": "omics.amazonaws.com"
            },
```

```
            "Action": [
                "ecr:CreateRepository",
                "ecr:BatchImportUpstreamImage"
            ],
            "Resource": [
                "arn:aws:ecr:us-east-1:123456789012:repository/ecr-public/*",
                "arn:aws:ecr:us-east-1:123456789012:repository/docker-hub/*"
            ]
        }
    ]
}
```

4. Speichern Sie die Richtlinie.

Um die Richtlinie an das anzuhängen AWS CLI Speichern Sie es stattdessen in einer Datei und verwenden Sie dann den Amazon ECR put-registry-policy Befehl:

```
aws ecr put-registry-policy \
     --policy-text file://registry-policy-file.json \
     --region region
```

> **Important**
>
> put-registry-policy ersetzt die gesamte Registrierungsrichtlinie. Wenn Ihre Registrierung bereits über eine Richtlinie verfügt, rufen Sie sie mit get-registry-policy, füge das hinzu HealthOmics Erklärung, und legen Sie die kombinierte Richtlinie zurück.

Weitere Informationen zu den Richtlinien für Amazon ECR Pull Through Cache finden Sie unter Richtlinien für Amazon ECR ziehen den Cache durch.

Schritt 3: Repository-Vorlagen erstellen

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

Jede Vorlage enthält ein Repository-Namespace-Präfix, das Amazon ECR verwendet, um neue Repositorys einer bestimmten Vorlage zuzuordnen. Vorlagen spezifizieren die Konfiguration

für alle Repository-Einstellungen, einschließlich ressourcenbasierter Zugriffsrichtlinien, Tag-Unveränderbarkeit, Verschlüsselung und Lebenszyklusrichtlinien.

Weitere Informationen finden Sie unter Vorlagen zur Repository-Erstellung in der Amazon Elastic Container Registry — Benutzerhandbuch.

> **Note**
>
> Erstellen Sie die Vorlage vor dem ersten Lauf, der das Präfix verwendet. Ein Präfix ohne eigene Vorlage und ohne `ROOT` Vorlage, erhält keine Repository-Richtlinie, also HealthOmics kann die zwischengespeicherten Bilder nicht abrufen. Belassen Sie die Tag-Veränderbarkeit bei `MUTABLE`, sodass Amazon ECR ein zwischengespeichertes Bild aktualisieren kann, wenn sich das Upstream-Tag ändert. Informationen zu Repositorys, die bereits existieren, finden Sie unter Aktualisierung der Repositorys, die vor der Vorlage erstellt wurden.

Gehen Sie mit der Amazon ECR-Konsole wie folgt vor, um eine Vorlage zur Repository-Erstellung zu erstellen:

Teil 1: Definition der Vorlage

1. Öffnen Sie die Amazon-ECR-Konsole.
2. Im Navigationsbereich unter Privates Register, erweitern Funktionen und Einstellungen, dann wähle Vorlagen zur Repository-Erstellung.
3. Wählen Sie Create template (Vorlage erstellen) aus.
4. In Einzelheiten zur Vorlage, wählen Cache durchsuchen.
5. Wählen Sie, ob Sie diese Vorlage auf ein bestimmtes Präfix oder auf alle Repositorys anwenden möchten, die keiner anderen Vorlage entsprechen.

   Wenn du wählst Ein bestimmtes Präfix, geben Sie den Namespace-Präfixwert in Präfix. Sie haben dieses Präfix angegeben, als Sie die Pull-Through-Cache-Regel erstellt haben.
6. Wählen Sie Weiter aus.

Teil 2: Hinzufügen der Konfiguration zur Repository-Erstellung

1. Auf der Konfiguration zur Repository-Erstellung hinzufügen Seite, für Repository-Berechtigungen, verwenden Sie eine der Beispielrichtlinien, oder geben Sie eine ähnliche wie im folgenden Beispiel ein:

JSON

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

2. (Optional) Fügen Sie Repository-Einstellungen wie Lebenszyklusrichtlinien und Tags hinzu. Amazon ECR wendet diese Regeln auf alle Container-Images an, die für den Pull-Through-Cache erstellt wurden und das angegebene Präfix verwenden.
3. Wählen Sie Weiter aus.
4. Überprüfen Sie die Konfiguration und wählen Sie Erstellen um die Vorlage zu erstellen.

Um die Vorlage zu erstellen mit dem AWS CLI Speichern Sie stattdessen die Repository-Richtlinie aus dem vorherigen Verfahren in einer Datei und verwenden Sie dann den Amazon ECR create-repository-creation-template Befehl mit `--prefix` auf das Cache-Repository-Präfix aus Ihrer Pull-Through-Cache-Regel setzen. Die Beschreibung ist optional:

```
aws ecr create-repository-creation-template \
     --prefix ecr-repository-prefix \
     --applied-for PULL_THROUGH_CACHE \
     --repository-policy file://repository-policy-file.json \
     --image-tag-mutability MUTABLE \
     --description "description" \
     --region region
```

Wenn bereits eine Vorlage für das Präfix existiert, create-repository-creation-template schlägt fehl mit TemplateAlreadyExistsException. Aktualisieren Sie stattdessen die vorhandene Vorlage an Ort und Stelle:

```
aws ecr update-repository-creation-template \
     --prefix ecr-repository-prefix \
     --applied-for PULL_THROUGH_CACHE \
     --repository-policy file://repository-policy-file.json \
     --image-tag-mutability MUTABLE \
     --region region
```

> **Important**
>
> update-repository-creation-template ersetzt jeden Wert, den Sie übergeben, also `--applied-for` und `--repository-policy` die bestehende Liste und Richtlinie überschreiben, anstatt sie mit ihnen zusammenzuführen. Rufen Sie die aktuelle Vorlage mit ab describe-repository-creation-templates, führen Sie Ihre Änderungen zusammen und übergeben Sie die vollständigen Werte.

Amazon ECR ist konfiguriert, aber HealthOmics Die Konfiguration ist noch nicht abgeschlossen. Wenn Ihre Workflow-Definition die Upstream-Image-URIs verwendet, fahren Sie mit Schritt 4: Hinzufügen einer Container-Registry Map (wenn Ihr Workflow Upstream-URIs verwendet): Ohne eine Container-Registrierungszuordnung schlägt der Lauf fehl, nachdem er gestartet wurde. Fahren Sie andernfalls mit dem Schritt Schritt 5: Überprüfen Sie Ihre Pull-Through-Cache-Konfiguration fort.

Schritt 4: Hinzufügen einer Container-Registry Map (wenn Ihr Workflow Upstream-URIs verwendet)

Die Image-URIs in Ihrer Workflow-Definition entscheiden, ob Sie eine Container-Registry-Map benötigen:

Image-URIs in Ihrer Workflow-Definition	Übersicht über die Registrierung
Die Upstream-URIs in der Form *upstream-registry-url* /*upstream-repository-name* :*tag*	Fügen Sie eine Karte hinzu. Die Karte verrät HealthOmics welches Cache-Repository-Präfix zu welcher Upstream-Registry gehört, weil HealthOmics liest deine Pull-Through-Cache-Regeln nicht.

Image-URIs in Ihrer Workflow-Definition	Übersicht über die Registrierung
Die privaten Amazon ECR-URIs Ihrer Cache-Repositorys in der folgenden Form *account-id* .dkr.ecr. *region*.amazonaws.com/ *ecr-repository-prefix* /*upstream-repository-name* :*tag*	Sie benötigen keine Karte. HealthOmics verwendet jeden URI wie angegeben.

Ein Image-Name, der ohne einen Registrierungshost geschrieben wird, ist ein Docker-Hub-Kurzname, der als Upstream-URI gilt und eine Map benötigt.

Wenn Ihre Definition beide Arten von URIs verwendet, fügen Sie eine Map für die Bilder hinzu, die die Upstream-URIs verwenden.

> **Important**
>
> Mit den Upstream-URIs `StartRun` ist auch dann erfolgreich, wenn die Map fehlt oder keine Zuordnung für die Upstream-Registrierung eines Images enthält. HealthOmics validiert jedes Image, wenn es die Aufgabe erstellt, die es verwendet, sodass die Ausführung nach dem Start fehlschlägt. In beiden Fällen wird der Grund für den Fehler angegeben INVALID_ECR_IMAGE_URI, und die Statusmeldung des Laufs unterscheidet sie. Ein Mapping, das ein Bild in ein Cache-Repository auflöst, das nicht existiert, bietet stattdessen ECR_REPOSITORY_NOT_FOUND.

Spiegeln Sie Ihre Pull-Through-Cache-Regel im Mapping: set wider `upstreamRegistryUrl` zur Upstream-Registrierungs-URL der Regel, `ecrRepositoryPrefix` zu seinem Cache-Repository-Präfix und `upstreamRepositoryPrefix` zu seinem Upstream-Repository-Präfix, falls die Regel eines verwendet.

Wenn sich Ihre Cache-Repositorys in einem anderen befinden AWS-Konto aus dem Workflow, setzen `ecrAccountId` zu dem Konto, dem die Cache-Repositorys gehören, und fügen Sie die kontoübergreifenden Richtlinien hinzu, die unter beschrieben sind Richtlinien für den kontoübergreifenden Amazon ECR-Zugriff. `ecrAccountId` verwendet standardmäßig das eigene Konto des Workflows.

> **Note**
>
> Für Docker Hub setzen Sie `upstreamRegistryUrl` zu `registry-1.docker.io`, der Wert in Voraussetzungen. HealthOmics Leckereien `registry-1.docker.io` und `docker.io` als derselbe Host, sodass ein Mapping auch Bild-URIs abdeckt, die mit geschrieben wurden `docker.io`.
> HealthOmics erweitert auch einen kurzen Docker-Hub-Namen, bevor er mit dem Mapping übereinstimmt: `ubuntu:22.04` wird `docker.io/library/ubuntu:22.04`, und `myorg/image:tag` wird `docker.io/myorg/image:tag`. HealthOmics Einsätze `library/` nur wenn der Bildname in Ihrer Definition keinen Schrägstrich enthält. Der erweiterte Name bestimmt auch den Cache-Repository-Pfad, also `ubuntu:22.04` wird aufgelöst zu *`ecr-repository-prefix`*`/library/ubuntu:22.04` in Ihrem privaten Register.

Sie stellen die Map bereit, wenn Sie den Workflow oder die Workflow-Version erstellen. Das Kartenformat und die Regeln, nach denen entschieden wird, welche Zuordnungen Ihr Workflow benötigt, finden Sie unter Zuordnungen in der Registrierung. Informationen zu Bildzuordnungen finden Sie unter Bildzuordnungen. Nachdem Sie die Karte hinzugefügt haben, fahren Sie fort mit Schritt 5: Überprüfen Sie Ihre Pull-Through-Cache-Konfiguration.

Schritt 5: Überprüfen Sie Ihre Pull-Through-Cache-Konfiguration

Führen Sie das aus aws ecr Befehle in dem Konto, dem die Cache-Repositorys gehören, weil get-registry-policy und describe-repository-creation-templates lese immer die Registrierung des Anruferkontos und starte den aws omics Befehle in dem Konto, dem der Workflow gehört. Verwenden Sie für beide die Region, in der Sie Ihre Workflows ausführen. Die drei Prüfungen im folgenden Verfahren sind unabhängig. Führen Sie also alle Prüfungen aus und erfassen Sie alle Probleme, bevor Sie sie beheben. Die Überprüfung der Registrierungszuordnung nach dem Verfahren gilt, sobald Sie den Workflow oder die Workflow-Version erstellt haben, wie unter beschrieben Hinzufügen von privaten Containern oder Containern von Drittanbietern während der Workflow-Erstellung. Die anschließende Ende-zu-Ende-Überprüfung erfordert ebenfalls einen gestarteten Lauf.

1. Vergewissern Sie sich, dass die Pull-Through-Cache-Regel existiert und auf die erwartete Upstream-Registrierung verweist:

```
aws ecr describe-pull-through-cache-rules \
     --region region
```

Jede Regel, die Sie erstellt haben, wird mit ihrer `ecrRepositoryPrefix` und `upstreamRegistryUrl`. Eine Regel, die ein Upstream-Repository-Präfix festlegt, wird ebenfalls angezeigt `upstreamRepositoryPrefix`, und eine Regel, die Anmeldeinformationen verwendet, wird ebenfalls angezeigt `credentialArn`; Amazon ECR lässt jedes dieser Felder aus, wenn die Regel es nicht festlegt. Ein leeres `pullThroughCacheRules` Eine Liste bedeutet, dass in dieser Region keine Regel existiert. Kehren Sie zurück zu Schritt 1: Erstellen einer Pull-Through-Cache-Regel.

Wenn die Regel Anmeldeinformationen verwendet, stellen Sie außerdem sicher, dass Amazon ECR das Geheimnis lesen kann, und melden Sie sich bei der Upstream-Registrierung an:

```
aws ecr validate-pull-through-cache-rule \
     --ecr-repository-prefix ecr-repository-prefix \
     --region region
```

Die Ausgabe beinhaltet `"isValid": true` wenn die Authentifizierung erfolgreich ist, und benennt andernfalls den Grund in `failure`. Wenn der Befehl fehlschlägt mit PullThroughCacheRuleNotFoundException, keine Regel verwendet dieses Präfix. Ein AccessDeniedException bedeutet, dass Ihrem Anrufer das fehlt `ecr:ValidatePullThroughCacheRule` Erlaubnis, die in Richtlinien zum Lesen nicht immer enthalten ist, nicht dass die Regel falsch ist. Weitere Informationen finden Sie unter Überprüfung der Pull-Through-Cache-Regeln in der Amazon Elastic Container Registry — Benutzerhandbuch.

2. Bestätigen Sie, dass die Registrierungsrichtlinie Folgendes gewährt HealthOmics Zugriff:

```
aws ecr get-registry-policy \
     --region region
```

Das `policyText` Wert in der Ausgabe, den Amazon ECR zurückgibt JSON-escaped, muss die Aussage von enthalten Schritt 2: Berechtigungen für die Upstream-Registrierung einrichten, gewährend `omics.amazonaws.com` beides `ecr:CreateRepository` und `ecr:BatchImportUpstreamImage`, einzeln oder durch einen Platzhalter und deckt jedes Cache-Repository-Präfix ab, das Sie erstellt haben. Ein Präfix wird durch ein abgedeckt `Resource` Eintrag, der mit den darunter liegenden Cache-Repositorys übereinstimmt, z. B. ein Repository-ARN, der mit dem Präfix endet, gefolgt von `/*`, eine Repository-ARN, die mit endet `/*` allein oder ein `Resource` von `*`. Ein ARN, der am Präfix selbst endet, ohne `/*`, deckt nichts ab: Ein Cache-Repository-Name hat immer einen Pfad unter dem Präfix. Ein

ARN, der eine Region benennt oder AWS-Konto außer der eigenen Registrierung deckt auch nichts ab, wie unter beschrieben Schritt 2: Berechtigungen für die Upstream-Registrierung einrichten. Wenn keine Anweisung ein Präfix abdeckt oder wenn der Befehl fehlschlägt mit RegistryPolicyNotFoundException, zurück zu Schritt 2: Berechtigungen für die Upstream-Registrierung einrichten und stellen Sie sicher, dass Sie die Richtlinie gespeichert haben.

3. Vergewissern Sie sich, dass für jedes Cache-Repository-Präfix eine Vorlage zur Repository-Erstellung existiert. Übergeben Sie alle Präfixe, die Sie erstellt haben:

```
aws ecr describe-repository-creation-templates \
     --prefixes ecr-repository-prefix-1 ecr-repository-prefix-2 \
     --region region
```

Für jedes Präfix muss die Vorlage, die Amazon ECR auf seine Cache-Repositorys anwendet — wie im nächsten Absatz beschrieben — Folgendes enthalten `PULL_THROUGH_CACHE` in `appliedFor` und ein `repositoryPolicy` das gewährt `omics.amazonaws.com` die Pull-Aktionen von Schritt 3: Repository-Vorlagen erstellen, `ecr:BatchGetImage` und `ecr:GetDownloadUrlForLayer`. Eine Richtlinie, die nicht leer ist, reicht nicht aus: Bestätigen Sie sowohl das Prinzip als auch die Aktionen.

Amazon ECR wendet die Vorlage an, deren Präfix die längste Übereinstimmung mit dem Repository ist, oder `ROOT` Vorlage, wenn kein Präfix übereinstimmt. Führen Sie den Befehl ohne `--prefixes` um alle Vorlagen zu sehen, weil `--prefixes` passt genau zu jedem Präfix und nicht nach Pfad, also keines `ROOT` Eine Vorlage oder eine Vorlage mit einem längeren Präfix erscheint in der gefilterten Ausgabe. Kehren Sie für jedes Präfix, dessen entsprechende Vorlage fehlt oder eine der Anforderungen nicht erfüllt, zu diesem Schritt zurück. `imageTagMutability` hat keinen Einfluss darauf, ob HealthOmics kann das Bild abrufen, aber eine Vorlage, die es auf etwas anderes festlegt `MUTABLE` stoppt Amazon ECR, ein zwischengespeichertes Bild zu aktualisieren, wenn das Upstream-Tag verschoben wird, wie unter beschrieben Schritt 3: Repository-Vorlagen erstellen.

Falls bereits Cache-Repositorys existierten, bevor Sie die Vorlage erstellt haben, überprüfen Sie auch die Repository-Richtlinie für jedes dieser Repositorys, wie unter beschrieben Aktualisierung der Repositorys, die vor der Vorlage erstellt wurden.

Wenn Ihre Definition die Upstream-URIs verwendet, bestätigen Sie die Registrierungszuordnung, die Sie hinzugefügt haben Schritt 4: Hinzufügen einer Container-Registry Map (wenn Ihr Workflow Upstream-URIs verwendet). Eine Map gehört zu dem Workflow oder der Workflow-Version, mit der

Sie sie geliefert haben, also überprüfen Sie, welche Ihre Läufe verwenden: die Version, als die Sie übergeben `--workflow-version-name` zu start-run, oder den Workflow, falls deine Läufe keinen benennen. Für einen Lauf, der bereits existiert, `workflowVersionName` in der get-run Die Ausgabe benennt ihre Version.

Für einen Arbeitsablauf:

```
aws omics get-workflow \
     --id workflow-id \
     --region region
```

Für eine Workflow-Version:

```
aws omics get-workflow-version \
     --workflow-id workflow-id \
     --version-name version-name \
     --region region
```

Prüfen Sie das `containerRegistryMap` Wert in der Ausgabe. Wenn es keine Map gibt, wird das Feld in der Antwort weggelassen. Bestätigen Sie das `status` ist `ACTIVE`: wenn es immer noch ist `CREATING`, warte und versuche es erneut, und wenn ja `FAILED`, der Grund ist in `statusMessage`. Sie können eine Map nicht an Ort und Stelle ändern. Wenn sie fehlt oder falsch ist, erstellen Sie eine neue Workflow-Version, wie unter beschrieben Re-running die Konfiguration.

Lesen Sie das Ergebnis eines Laufs, bei dem der Cache verwendet wird, um zu überprüfen, ob bei einem Lauf ein zwischengespeichertes Bild verwendet wurde. Eine Ausführung, die bereits abgeschlossen ist, funktioniert, auch wenn eine fehlgeschlagene Ausführung ausgeführt wurde, aber bei einem abgebrochenen Lauf wurden die Images möglicherweise nie aufgelöst. Starten Sie einen Lauf nur, wenn Sie noch keinen haben:

```
aws omics get-run \
     --id run-id \
     --region region
```

Der `resourceDigests` Das Feld in der Ausgabe zeichnet Ressourcen auf, die durch den Lauf aufgelöst wurden, einschließlich der zugehörigen Container-Images, wenn der Lauf einen Wert meldet. Ein Eintrag, dessen Schlüssel eine private Amazon ECR-URI unter Ihrem Cache-Repository-Präfix ist, mit einem `sha256` digest als Wert bestätigt, dass der Lauf das Bild in Ihrem Cache

aufgelöst hat. Dies ist die schnellste Überprüfung, und sie funktioniert auch bei einem Lauf, der aus einem anderen Grund fehlgeschlagen ist.

Wenn der Lauf erfolglos gestoppt wurde, `failureReason` und `statusMessage` zusammen sagen wir Ihnen, ob es sich um diese Konfiguration handelt. Überprüfe diese, bevor du dir die Aufgaben des Laufs ansiehst, denn ein Lauf, der auf einem Image fehlschlägt, erzeugt möglicherweise gar keine Aufgaben:

- ECR_REPOSITORY_NOT_FOUND, ECR_IMAGE_NICHT_GEFUNDEN oder ECR_PERMISSION_ERROR für eine Bild-URI unter einem Ihrer Cache-Repository-Präfixe — fahren Sie fort mit Problembehandlung beim Pull-Through-Cache. Einer dieser Gründe für eine URI, die nicht unter einem Ihrer Präfixe steht, ist eher ein Amazon ECR-Problem mit diesem Repository als ein Pull-Through-Cache-Problem, da keine Cache-Regel für sie gilt. Für ein Repository, das Sie selbst in Amazon ECR verwalten, geben Sie HealthOmics Zugriff mit einer Repository-Richtlinie, wie beschrieben unter Erstellen Sie eine Ressourcenrichtlinie für das Amazon ECR-Repository.
- UNGÜLTIGE_ECR_IMAGE-URI — Die Registry Map fehlt, sie hat keine Zuordnung für die Upstream-Registry des Images, es ist `upstreamRepositoryPrefix` deckt den Pfad des Bildes nicht ab, oder die Bild-URI ist nicht gültig. Lesen Sie die Statusmeldung, um diese zu unterscheiden, und überprüfen Sie die URI, bevor Sie die Karte ändern. Siehe Schritt 4: Hinzufügen einer Container-Registry Map (wenn Ihr Workflow Upstream-URIs verwendet).
- CONTAINER_ERROR KANN NICHT GESTARTET WERDEN oder IMAGE_VERIFICATION_FAILURE für eine Bild-URI unter einem Ihrer Präfixe — das Bild hat Ihr Cache-Repository erreicht, aber die Aufgabe konnte es nicht verwenden. Wenn Sie die Cache-Regel geändert oder neu erstellt haben, kann dies durch ein veraltetes, zwischengespeichertes Bild verursacht werden; siehe Änderung einer Cache-Regel, während eine Ausführung aktiv ist. Andernfalls ist das zwischengespeicherte Image selbst das Problem und nicht diese Konfiguration, und das Upstream-Image oder die Aufgabe, die es verwendet, ist der Ort, an dem gesucht werden muss.
- Irgendein anderer Fehlergrund oder gar kein Grund, wie ein abgebrochener Lauf meldet — der Lauf wurde aus einem Grund gestoppt, der nichts mit dieser Konfiguration zu tun hat. Informationen zum Ausführungsstatus und zu den anderen Fehlergründen finden Sie unter Holen Sie sich Informationen zum Lauf und Gründe für Ausführungsfehler.

Nicht jeder Lauf meldet sich `resourceDigests`, und ein Lauf, dessen Bilder noch nicht aufgelöst wurden, hat keinen Eintrag für sie. Wenn das Feld oder der Eintrag fehlt oder wenn Sie überprüfen möchten, welches Image eine einzelne Aufgabe abgerufen hat, listen Sie die Aufgaben des Laufs auf:

```
aws omics list-run-tasks \
     --id run-id \
     --region region
```

Ein leeres `items` Eine Liste bei einem Lauf, der noch läuft, bedeutet HealthOmics hat die Aufgaben noch nicht erstellt, also warte und liste sie erneut auf. Bei einem Lauf, der gestoppt wurde, bedeutet das, dass noch nie eine Aufgabe erstellt wurde: Bei einem fehlgeschlagenen Lauf erklären die vorherigen Fehlergründe, warum, und ein Lauf, der vor dem Start abgebrochen wurde, hat nichts zu bestätigen. Verwenden Sie also einen anderen Lauf. Rufen Sie andernfalls die Image-Details für eine der Aufgaben ab. Verwenden Sie dazu `taskId` aus der vorherigen Ausgabe:

```
aws omics get-run-task \
     --id run-id \
     --task-id task-id \
     --region region \
     --query imageDetails
```

In `imageDetails`, `sourceImage` ist die Bild-URI aus Ihrer Workflow-Definition und `image` ist die private Amazon ECR-URI, die von der Aufgabe abgerufen wurde. Ein `image` Wert unter Ihrem Cache-Repository-Präfix, in der privaten URI-Form, wie in Schritt 4: Hinzufügen einer Container-Registry Map (wenn Ihr Workflow Upstream-URIs verwendet) und möglicherweise mit einer angehängten Bildübersicht, bestätigt, dass die Aufgabe Ihren Pull-Through-Cache verwendet hat. `imageDetails` kann leer sein, für eine Aufgabe, die ihr Image noch nicht abgerufen hat, und für einige Aufgaben, die bereits abgeschlossen wurden. Eine Aufgabe, für die keine Bilddetails aufgezeichnet wurden, hat auch keine im Ausführungsmanifestprotokoll. Überprüfen Sie daher eine andere Aufgabe derselben Ausführung. Wenn keine Aufgabe in einem abgeschlossenen Lauf sie aufgezeichnet hat, können Sie trotzdem bestätigen, dass das Image zwischengespeichert wurde, wie unter beschrieben Behebung einer fehlenden oder unvollständigen Registrierungsrichtlinie. Das bestätigt eher die Konfiguration als das Image, das eine bestimmte Aufgabe abgerufen hat, da ein früherer Lauf den Cache möglicherweise gefüllt hat.

Nach Abschluss einer Ausführung wiederholt das Ausführungsmanifestprotokoll diese Image-Informationen, sodass Sie die Aufgaben der Ausführung sofort überprüfen können, anstatt sie aufzurufen get-run-task für jeden. Im Datensatz jeder Aufgabe das benannte Feld `originalImage`

entspricht `sourceImage` hier, und `image` ist derselbe Wert wie oben. Wo Sie das Protokoll finden, finden Sie unter Überwachung HealthOmics mit CloudWatch Logs, und den Rest des Datensatzes einer Aufgabe finden Sie unter Holen Sie sich Informationen zum Lauf.

Nachdem Sie den Pull-Through-Cache konfiguriert haben, können Sie einen neuen Workflow oder eine neue Workflow-Version erstellen, um die Container aus konfigurierten Registries zu verwenden. Weitere Informationen finden Sie unter Hinzufügen von privaten Containern oder Containern von Drittanbietern während der Workflow-Erstellung.

Re-running die Konfiguration

Sie können diese Schritte wiederholen, um beispielsweise ein Präfix hinzuzufügen oder eine Richtlinie zu korrigieren. Die Create-Befehle schlagen mit einem Exit-Code ungleich Null fehl, wenn die Ressource bereits existiert, ohne sie zu ändern. put-registry-policy schlägt nicht fehl, aber es ersetzt die gesamte Registrierungsrichtlinie, wie unter beschrieben Schritt 2: Berechtigungen für die Upstream-Registrierung einrichten.

- Cache-Regel durchziehen — create-pull-through-cache-rule schlägt fehl mit PullThroughCacheRuleAlreadyExistsException. Wenn für die Registrierung, die Sie hinzufügen, eine Authentifizierung erforderlich ist, erstellen Sie zuerst das Secrets Manager-Geheimnis, wie unter beschrieben Voraussetzungen.
- Richtlinie für Registrierungsberechtigungen — fügen Sie eine hinzu `Resource` Eintrag für jedes neue Cache-Repository-Präfix, wie unter beschrieben Schritt 2: Berechtigungen für die Upstream-Registrierung einrichten; eine Richtlinie, die Präfixe einzeln aufzählt, gilt nicht für neue. Bestätigen Sie den Versicherungsschutz mit get-registry-policy, wie beschrieben unter Schritt 5: Überprüfen Sie Ihre Pull-Through-Cache-Konfiguration.
- Vorlage zur Repository-Erstellung — Erstellen Sie eine Vorlage für ein Cache-Repository-Präfix, das keines hat, und aktualisieren Sie die vorhandene Vorlage, um ein Präfix zu ändern, das bereits eines hat. Beides ist unter beschrieben Schritt 3: Repository-Vorlagen erstellen. Wenn Sie ein Update für die Vorlage eines anderen Präfixes ausführen, wird diese Vorlage überschrieben. Um Repositorys, die Amazon ECR bereits erstellt hat, auf den neuesten Stand zu bringen, siehe Aktualisierung der Repositorys, die vor der Vorlage erstellt wurden.
- Container-Registry Map — Sie geben die Map an, wenn Sie einen Workflow oder eine Workflow-Version erstellen, und update-workflow und update-workflow-version kann es nicht ändern. Bevor Sie eine Version erstellen, sollten Sie die komplette Map bereithalten: Eine neue Version erbt nicht die Map der vorherigen Version, also listen Sie alle Mappings auf, die der Workflow noch benötigt, nicht nur die neue. create-workflow-version verwendet die gesamte Version und nicht nur die Map, daher benötigen Sie auch die Workflow-Definition und die anderen Parameter der

Version, wie unter beschrieben Eine Workflow-Version erstellen. Erstellen Sie dann die Version mit der aktualisierten Karte, wie unter beschrieben Schritt 4: Hinzufügen einer Container-Registry Map (wenn Ihr Workflow Upstream-URIs verwendet), und starte deine Runs mit dieser Version. Ein Lauf, der keine Version benennt, verwendet die Standardversion des Workflows, die die Map, mit der er erstellt wurde, beibehält, sodass er auf die gleiche Weise fehlschlägt wie zuvor. Weitere Informationen finden Sie unter Standard-Workflow-Version.

Problembehandlung beim Pull-Through-Cache

Die folgenden Konfigurationsprobleme führen dazu, dass ein Lauf fehlschlägt, wenn Sie Container aus dem Pull-Through-Cache verwenden. Wenn Ihre Workflow-Definition die Upstream-Image-URIs verwendet und ein Lauf fehlschlägt mit INVALID_ECR_IMAGE_URI, beginnen Sie mit der Registrierungsübersicht, wie unter beschrieben Schritt 4: Hinzufügen einer Container-Registry Map (wenn Ihr Workflow Upstream-URIs verwendet). Weitere Gründe für Lauffehler finden Sie unter Gründe für Ausführungsfehler.

Führen Sie das aus aws ecr Befehle in diesem Abschnitt im selben Konto wie in Schritt 5: Überprüfen Sie Ihre Pull-Through-Cache-Konfiguration, oder bestehen `--registry-id` *`ecr-account-id`*. Verwenden Sie die Repository-Namen genau wie describe-repositories gibt sie zurück: Ein Cache-Repository-Name ist das Cache-Repository-Präfix, gefolgt vom Upstream-Repository-Pfad, der für einen Docker-Hub-Kurznamen Folgendes beinhaltet `library/` Segment, beschrieben in Schritt 4: Hinzufügen einer Container-Registry Map (wenn Ihr Workflow Upstream-URIs verwendet).

Behebung einer fehlenden oder unvollständigen Registrierungsrichtlinie

Symptom— Die Ausführung für den Image-URI schlägt mit einem der folgenden Fehlergründe fehl:

- ECR_REPOSITORY_NOT_FOUND — Das Cache-Repository existiert nicht in Ihrer privaten Registrierung.
- ECR_IMAGE_NOT_FOUND — Das Cache-Repository existiert, aber das Bild wurde nie in es importiert.
- ECR_PERMISSION_ERROR — Amazon ECR verweigert den Zugriff.

Wenn das Cache-Repository fehlt, gibt der folgende Befehl kein Repository für das Präfix zurück:

```
aws ecr describe-repositories \
     --region region \
     --query "repositories[?starts_with(repositoryName, 'ecr-repository-prefix/')]"
```

Ein leeres Ergebnis für ein Präfix, aus dem noch nichts abgerufen wurde, wird erwartet und ist kein eigenständiges Konfigurationsproblem. Wenn ein Rechenlauf für den Image-URI fehlgeschlagen ist, bedeutet ein fehlendes Repository in der Regel, dass Amazon ECR es nicht erstellen konnte. Wenn das Repository existiert, überprüfen Sie, ob das Image es erreicht hat:

```
aws ecr describe-images \
     --repository-name cache-repository-name \
     --image-ids imageTag=tag \
     --region region
```

Ein zurückgegebenes Bild bedeutet, dass das Tag zwischengespeichert ist. ImageNotFoundException bedeutet, dass das Repository existiert, aber dieses Tag nicht darin ist, und RepositoryNotFoundException bedeutet, dass der Repository-Name falsch ist oder dass noch nichts das Präfix durchgesetzt hat. Durchgehen `--image-ids`: Ohne ihn meldet der Befehl jedes Tag im Projektarchiv, sodass das Vorhandensein eines anderen Tags als Erfolg für das Tag, das dein Lauf benötigt, aussehen kann.

Ursache—Die Registrierungsrichtlinie gewährt nicht HealthOmics was sie benötigt: Es gibt keine Richtlinie, der Richtlinie fehlt eine Aktion oder `Resource` Die Liste deckt dieses Cache-Präfix nicht ab.

Lösung— Fügen Sie die Registrierungsrichtlinie hinzu oder korrigieren Sie sie wie unter beschrieben Schritt 2: Berechtigungen für die Upstream-Registrierung einrichten, und bestätigen Sie sie wie unter beschrieben Schritt 5: Überprüfen Sie Ihre Pull-Through-Cache-Konfiguration. Wenn die Registrierungsrichtlinie das Cache-Repository-Präfix bereits abdeckt, ist die Registrierungsrichtlinie nicht das Problem. Für ECR_IMAGE_NOT_FOUND, wo das Repository existiert, aber das Image nie angekommen ist, ist der Upstream-Import fehlgeschlagen: Bestätigen Sie die Anmeldeinformationen der Regel mit validate-pull-through-cache-rule, wie beschrieben unter Schritt 5: Überprüfen Sie Ihre Pull-Through-Cache-Konfiguration. Für ECR_PERMISSION_ERROR fahren Sie in einem vorhandenen Repository fort mit Behebung einer fehlenden Vorlage zur Repository-Erstellung oder einer Repository-Richtlinie. Dann starte den Lauf erneut.

Behebung einer fehlenden Vorlage zur Repository-Erstellung oder einer Repository-Richtlinie

Symptom— Das Cache-Repository ist mit dem importierten Image vorhanden, und die Ausführung schlägt immer noch fehl ECR_PERMISSION_ERROR für die Bild-URI.

Ursache— Keine Vorlage zur Repository-Erstellung entspricht dem Cache-Repository-Präfix, die Vorlage enthält keine `PULL_THROUGH_CACHE` in `appliedFor`, oder die Vorlage hat keine

Repository-Richtlinie. Das Repository hat daher keine Richtlinie, die Folgendes ermöglicht HealthOmics ziehe davon ab.

Lösung— Erstellen oder aktualisieren Sie die Vorlage wie unter beschrieben Schritt 3: Repository-Vorlagen erstellen. Da die Repositorys bereits existieren, wenden Sie die Richtlinie auch auf sie an, wie unter Aktualisierung der Repositorys, die vor der Vorlage erstellt wurden.

Aktualisierung der Repositorys, die vor der Vorlage erstellt wurden

Symptom— Sie haben die Vorlage zur Repository-Erstellung erstellt oder korrigiert, describe-repository-creation-templates zeigt sie als korrekt an, und eine Ausführung schlägt immer noch fehl und es wird ein Amazon ECR-Zugriffsfehler für dasselbe Image angezeigt. Bestätigen Sie die fehlende Richtlinie für ein betroffenes Repository:

```
aws ecr get-repository-policy \
     --repository-name cache-repository-name \
     --region region
```

Die `policyText` Wert muss gewähren `omics.amazonaws.com` mindestens die Aktionen, die die Repository-Richtlinie der Vorlage gewährt. Wenn das Repository überhaupt keine Richtlinie hat, schlägt der Befehl fehl mit RepositoryPolicyNotFoundException. Wenn es fehlschlägt mit RepositoryNotFoundException, der Repository-Name ist falsch oder das Repository existiert noch nicht.

Ursache— Eine Vorlage gilt nur, wenn Amazon ECR ein Repository erstellt. Repositorys, die vor dem Hinzufügen der Vorlage erstellt wurden, haben daher keine Repository-Richtlinie für HealthOmics.

Lösung— Bringen Sie die vorhandenen Repositorys auf eine der folgenden Arten auf den neuesten Stand.

> **Warning**
>
> Durch das Löschen eines Repositorys werden auch alle darin zwischengespeicherten Bilder gelöscht, und Sie können es nicht rückgängig machen. Lösche ein Repository nicht, während es von einem Run verwendet wird.

- Lege die Richtlinie direkt für jedes bestehende Repository fest. Speichern Sie die Repository-Richtlinie der Vorlage in einer Datei und führen Sie dann den folgenden Befehl für jedes betroffene Repository aus:

```
aws ecr set-repository-policy \
     --repository-name cache-repository-name \
     --policy-text file://repository-policy-file.json \
     --region region
```

- Löschen Sie jedes betroffene Repository. Der Pull-Through-Cache erstellt es aus der Vorlage neu, wenn eine Aufgabe das Image das nächste Mal abruft. Das zwischengespeicherte Image ist also erst verfügbar, wenn der Abruf abgeschlossen ist.

```
aws ecr delete-repository \
     --repository-name cache-repository-name \
     --force \
     --region region
```

Änderung einer Cache-Regel, während eine Ausführung aktiv ist

Symptom—Nachdem Sie eine Pull-Through-Cache-Regel geändert oder neu erstellt haben, kann ein Lauf, der das betroffene Repository bereits verwendet, fehlschlagen mit CANNOT_START_CONTAINER_ERROR oder IMAGE_VERIFICATION_FAILURE, oder kann mit einem Bild abgeschlossen werden, das Sie nicht erwartet haben.

Ursache— Das Cache-Repository-Präfix wird jetzt in ein anderes Upstream-Repository aufgelöst, während Läufe bereits aus den Repositorys abgerufen werden, die mit der alten Regel erstellt wurden.

Lösung—Warten Sie, bis die aktiven Durchläufe abgeschlossen sind, bevor Sie die Regel ändern, wie der Hinweis in Schritt 1: Erstellen einer Pull-Through-Cache-Regel beschreibt. Um zu überprüfen, welches Image eine Aufgabe tatsächlich verwendet hat, überprüfen Sie `imageDetails` in der get-run-task Ausgabe, wie beschrieben unter Schritt 5: Überprüfen Sie Ihre Pull-Through-Cache-Konfiguration. Wenn bei einem Lauf ein veraltetes zwischengespeichertes Image verwendet wurde, löschen Sie dieses Image aus dem Cache-Repository, sodass es beim nächsten Abruf aus der Upstream-Registrierung aktualisiert wird.

Verwendung von Drittanbieter-Registrierungen, die vom Pull-Through-Cache nicht unterstützt werden

Ihr Workflow kann Container-Images von Upstream-Registern verwenden, die Amazon ECR Pull Through Cache (PTC) nicht unterstützt. Wenn eine Aufgabe aus Ihrem Lauf zum ersten Mal eines

dieser Images benötigt, HealthOmics kopiert es aus der Upstream-Registrierung in ein privates Amazon ECR-Repository in Ihrem Konto und startet die Aufgabe von der Kopie aus. Spätere Aufgaben und Läufe verwenden die Kopie wieder, sodass der Download nur einmal erfolgt. Wenn mehrere Läufe dasselbe Image gleichzeitig benötigen, z. B. stapelweise, HealthOmics kopiert das Bild einmal und die Läufe warten auf diese Kopie. Ihre Workflow-Definition behält die ursprünglichen Upstream-Image-URIs bei, sodass Sie die Image-URIs nicht selbst abrufen, übertragen oder neu schreiben müssen.

Wir empfehlen, dass Sie diese Bilder nicht nach Tag, sondern nach Digest oder Version referenzieren. Weitere Informationen finden Sie unter Verwenden Sie für Bilder, die nicht von PTC stammen, einen Digest oder eine Version anstelle eines Tags.

HealthOmics unterstützt die folgenden Upstream-Registrierungen, die PTC nicht unterstützt:

- Seqera-Container (`community.wave.seqera.io` und `wave.seqera.io`)
- NVIDIA GC (`nvcr.io`)
- Google-Container-Registrierung (`gcr.io`)
- Google-Artefaktregister (*`region`*`-docker.pkg.dev`). Verwenden Sie zum Beispiel den vollständigen regionalen Host `us-central1-docker.pkg.dev`.

Themen

- Konfiguration der Unterstützung von Nicht-PTC-Containern
- Benennung des Ziel-Repositories
- Überlegungen zur Unterstützung von Containern, die nicht von PTC unterstützt werden

Konfiguration der Unterstützung von Nicht-PTC-Containern

Bevor Sie Container aus Registries verwenden, die PTC nicht unterstützt, speichern Sie Ihre Anmeldeinformationen für die Registrierung und richten Sie Berechtigungen ein HealthOmics.

Schritt 1: Speichern Sie Ihre Registrierungsdaten in Secrets Manager

HealthOmics authentifiziert und kopiert Container-Images aus Upstream-Registern mithilfe Ihrer Anmeldeinformationen. Sie speichern Ihre Anmeldeinformationen als Geheimnis darin AWS Region wie Ihre HealthOmics läuft. Sie können auch ein Geheimnis aus einem anderen verwenden AWS-Konto oder teile dein Geheimnis mit einem anderen Konto. Wenn du ein Geheimnis von einem anderen verwendest AWS-Konto, die Servicerolle, an die Sie übergehen StartRun muss das

Geheimnis lesen dürfen. Weitere Informationen finden Sie unter Greifen Sie von einem anderen Konto aus auf Secrets Manager-Geheimnisse zu in der Secrets Manager-Benutzerhandbuch.

Jede unterstützte Registrierung benötigt Anmeldeinformationen, einschließlich Registrierungen, die anonyme Abrufe zulassen, wie `community.wave.seqera.io`.

Die folgenden Registerkarten zeigen das Format der Anmeldeinformationen und ein Beispiel für jede unterstützte Registrierung.

Seqera Containers and Wave

Ein JSON-Objekt mit einem `accessToken` Schlüssel, der ein Seqera Platform-Zugriffstoken enthält.

Folgendes AWS CLI Der Befehl erstellt ein Geheimnis für Seqera Containers:

```
aws secretsmanager create-secret \
     --name omics-registry/seqera \
     --secret-string '{"accessToken":"seqera-platform-access-token"}' \
     --region us-east-1
```

NVIDIA NGC

Ein JSON-Objekt mit `username` auf das Literal gesetzt `$oauthtoken` und `accessToken` auf Ihren NGC-API-Schlüssel setzen.

Folgendes AWS CLI Der Befehl erstellt ein Geheimnis für NVIDIA NGC:

```
aws secretsmanager create-secret \
     --name omics-registry/nvidia-ngc \
     --secret-string '{"username":"$oauthtoken","accessToken":"ngc-api-key"}' \
     --region us-east-1
```

Wenn du ein Geheimnis für NVIDIA NGC erstellst, behalte das `--secret-string` Wert in einfachen Anführungszeichen, damit Ihre Shell nicht expandiert `$oauthtoken`.

Google Container Registry and Google Artifact Registry

Erstellen Sie einen JSON-Schlüssel für ein Google Cloud-Dienstkonto, das Bilder aus Ihrer Registrierung lesen kann, und speichern Sie die Schlüsseldatei (z. B. `key.json`). Anweisungen dazu finden Sie unter Dienstkontoschlüssel erstellen und löschen in der Google Cloud-Dokumentation.

Einstellen `username` zu `_json_key` und `accessToken` zum Inhalt der Schlüsseldatei als einzelne JSON-Zeichenfolge. Folgendes AWS CLI Der Befehl erstellt das Geheimnis und verwendet das `jq` Tool zum Erstellen der JSON-Zeichenfolge:

```
aws secretsmanager create-secret \
     --name omics-registry/google \
     --secret-string "$(jq -n --rawfile key key.json '{username: "_json_key",
 accessToken: $key}')" \
     --region us-east-1
```

Sie können den Amazon-Ressourcennamen (ARN) des Geheimnisses aus der Antwort aufzeichnen und ihn als `credentialArn` Wert in der Registrierungszuordnung.

Schritt 2: Erteilen Sie der Servicerolle Berechtigungen

HealthOmics verwendet die Servicerolle, an die Sie übergeben `StartRun` um das Geheimnis zu lesen, erstellen Sie das Ziel-Repository und übertragen Sie das kopierte Image. Die Servicerolle muss dieselbe sein AWS-Konto wie der Lauf. Ansonsten HealthOmics lehnt die Rolle mit dem Fehler ab Die IAM-Rolle entspricht nicht dem anfordernden AWS-Konto.

Informationen zur Richtlinie finden Sie unter Richtlinie zur Dienstrolle für die Unterstützung von Containern, die nicht von PTC stammen.

Nachdem Sie diese Berechtigungen erteilt haben, fügen Sie eine Registrierungszuordnung mit dem geheimen ARN als `credentialArn`. Weitere Informationen finden Sie unter Zuordnungen in der Registrierung.

Benennung des Ziel-Repositories

Sie geben Registrierungszuordnungen an, wenn Sie einen Workflow erstellen, und HealthOmics kopiert die Bilder während einer Ausführung. Weitere Informationen finden Sie unter Hinzufügen von privaten Containern oder Containern von Drittanbietern während der Workflow-Erstellung.

Jedes kopierte Bild landet bei einer URI, die Sie anhand des Registrierungs-Mappings berechnen können:

```
account-id.dkr.ecr.region.amazonaws.com/ecrRepositoryPrefix/image-path:tag
```

`image-path` ist der Upstream-Image-URI ohne den Registrierungshost und ohne `upstreamRepositoryPrefix` wenn Sie einen festlegen. Behalten

`upstreamRepositoryPrefix` Sonnenuntergang für Wave (`wave.seqera.io`). Das Konto ist das Konto des Workflows, es sei denn, Sie legen Folgendes fest `ecrAccountId`. Wenn du festlegst `ecrAccountId`, HealthOmics erstellt das Repository und überträgt die Kopie mithilfe Ihrer Servicerolle in dieses Konto. Die Servicerolle benötigt dann dieselben Amazon ECR-Berechtigungen für die Repositorys dieses Kontos, und die Registrierungs- und Repository-Richtlinien dieses Kontos müssen dies zulassen.

Zum Beispiel `community.wave.seqera.io/library/samtools:1.22.1--eccb42ff8fb55509` mit Präfix `wave-seqera` landet bei `111122223333.dkr.ecr.us-east-1.amazonaws.com/wave-seqera/library/samtools:1.22.1--eccb42ff8fb55509`.

Um zu bestätigen, dass ein Kopiervorgang erfolgreich war, überprüfen Sie, ob das Bild an seinem Ziel-URI existiert, z. B. mit den folgenden Angaben AWS CLI Befehl:

```
aws ecr describe-images \
     --repository-name wave-seqera/library/samtools \
     --image-ids imageTag=1.22.1--eccb42ff8fb55509 \
     --region us-east-1
```

Wenn das Repository im Konto des Workflows nicht existiert, HealthOmics erstellt es. Wenn das Repository keine Repository-Richtlinie hat, HealthOmics hängt eine an, die Folgendes ermöglicht HealthOmics ziehe das Bild. Wenn das Repository bereits eine Richtlinie hat, HealthOmics lässt es unverändert, also füge eine Anweisung hinzu, die Folgendes ermöglicht HealthOmics Bilder abrufen. Ein Beispiel finden Sie unter Erstellen Sie eine Ressourcenrichtlinie für das Amazon ECR-Repository.

`ecrRepositoryPrefix/image-path` wird zu einem Amazon ECR-Repository-Namen, daher muss das Präfix Kleinbuchstaben, Zahlen und die folgenden Zeichen enthalten `/`, `-`, `_`, und `..`

Wenn sich Ihre Workflows im selben Konto eine teilen `ecrRepositoryPrefix`, sie teilen sich die kopierten Bilder. Wenn Sie ein kopiertes Bild löschen, wirkt sich dies auf jeden Workflow aus, der dieses Bild mit diesem Präfix verwendet.

Überlegungen zur Unterstützung von Containern, die nicht von PTC unterstützt werden

Beachten Sie Folgendes, wenn Sie die Unterstützung von Nicht-PTC-Containern verwenden:

- Architektur — HealthOmics kopiert nur x86_64-Bilder. ARM-based und Images mit mehreren Architekturen werden nicht unterstützt.
- Gemeinsame Nutzung von Arbeitsabläufen — Non-PTC Container-Unterstützung wird für gemeinsam genutzte Workflows nicht unterstützt. Wenn ein anderes Konto einen Workflow

ausführt, den Sie geteilt haben, HealthOmics ignoriert Registrierungszuordnungen, die Folgendes beinhalten `credentialArn`.

- Bild-Tags — HealthOmics kopiert das Bild, das ein Tag enthält, wie `latest`, zeigt auf die Kopie und aktualisiert sie nicht, wenn sich der Upstream-Inhalt ändert. Um den neuen Inhalt zu übernehmen, löschen Sie das kopierte Bild aus Amazon ECR. Nachdem Sie es gelöscht haben, HealthOmics kopiert das Image bei der nächsten Ausführung, bei der es benötigt wird, erneut aus der Upstream-Registrierung. Wenn Ihr Workflow auf ein neues Tag oder einen neuen Digest verweist, HealthOmics kopiert dieses Bild, wenn es für eine Aufgabe zum ersten Mal benötigt wird.
- Kopierfehler — Wenn die Registrierungsinformationen ungültig sind oder die Servicerolle nicht über die entsprechenden Berechtigungen verfügt HealthOmics benötigt, das Kopieren des Images schlägt fehl und die Aufgabe und die Ausführung schlagen fehl. Wenn eine Kopie des Images bereits in Amazon ECR vorhanden ist, HealthOmics verwendet diese Kopie und überprüft die Registrierungsdaten nicht.

Hinzufügen von privaten Containern oder Containern von Drittanbietern während der Workflow-Erstellung

Wenn Sie einen neuen Workflow erstellen, können Sie die Registrierungs- und Image-Zuordnungen definieren, die für Ihren Workflow erforderlich sind. Weitere Informationen finden Sie unter Erstellen Sie einen privaten Workflow. Sie können Registrierungs- und Image-Zuordnungen auch aktualisieren, wenn Sie eine neue Workflow-Version erstellen. Weitere Informationen finden Sie unter Eine Workflow-Version erstellen.

In den folgenden Abschnitten werden Registrierungs- und Image-Zuordnungen beschrieben.

Zuordnungen in der Registrierung

Sie definieren Registrierungszuordnungen, um Präfixe in Ihrer privaten Amazon ECR-Registrierung und den Namen der Upstream-Registry zuzuordnen. Eine Registrierungsübersicht kann sowohl Pull-Through-Cache-Zuordnungen als auch Zuordnungen für Registries enthalten, die HealthOmics Kopien von.

Das folgende Beispiel ordnet Docker Hub, Quay und Amazon ECR Public über den Pull-Through-Cache zu und ordnet Seqera Containers zu (`community.wave.seqera.io`) mit einem Berechtigungsnachweis, sodass HealthOmics kopiert seine Bilder.

```
{
```

```
    "registryMappings": [
        {
            "upstreamRegistryUrl": "registry-1.docker.io",
            "ecrRepositoryPrefix": "docker-hub"
        },
        {
            "upstreamRegistryUrl": "quay.io",
            "ecrRepositoryPrefix": "quay"
        },
        {
            "upstreamRegistryUrl": "public.ecr.aws",
            "ecrRepositoryPrefix": "ecr-public"
        },
        {
            "upstreamRegistryUrl": "community.wave.seqera.io",
            "ecrRepositoryPrefix": "wave-seqera",
            "credentialArn": "arn:aws:secretsmanager:us-
east-1:111122223333:secret:omics-registry/seqera-EXAMPLE"
        }
    ]
}
```

Die folgenden Regeln gelten für Registrierungszuordnungen:

- `upstreamRegistryUrl` muss exakt mit dem Registrierungshost am Anfang der Image-URI übereinstimmen. Zum Beispiel ein Mapping für `wave.seqera.io` deckt keine Bilder ab `community.wave.seqera.io`.
- Jeder Upstream-Registrierungshost, auf den der Workflow verweist, benötigt eine Zuordnung. Bilder, auf die Sie mit ihrer privaten Amazon ECR-URI verweisen, benötigen keine.
- Pull-Through-Cache-Zuordnungen werden nicht verwendet `credentialArn`. Wenn du hinzufügst `credentialArn` zu einem Pull-Through-Cache-Mapping, HealthOmics ignoriert es.
- Zuordnungen für Registries, die HealthOmics Kopien von müssen Folgendes enthalten `credentialArn`. Wenn eine dieser Zuordnungen keine hat `credentialArn`schlägt der Workflow mit einem Fehler fehl, der die Registrierung benennt.
- Behalten `upstreamRepositoryPrefix` Sonnenuntergang für Wave (`wave.seqera.io`).

`CreateWorkflow` und `CreateWorkflowVersion` sind asynchron. Wenn der Status lautet `FAILED`, der Fehler erscheint in `statusMessage` in der `GetWorkflow` Antwort für einen Workflow, oder `GetWorkflowVersion` Antwort für eine Workflow-Version.

Stellen Sie die Karte im Inline-Format bereit mit `containerRegistryMap`oder als Amazon S3-URI mit `containerRegistryMapUri`, nicht beides. CLI-Beispiele finden Sie unter Konfigurieren Sie die Parameter für die Pull-through-Cache-Zuordnung und Konfigurieren Sie Cache-Mapping-Parameter ohne Pull-Through-Cache.

Der Lauf verwendet die Registrierungsübersicht des Workflows oder der Workflow-Version, den Sie ausführen, sodass Sie nichts hinzufügen `StartRun`.

Bildzuordnungen

Sie definieren Image-Zuordnungen zur Zuordnung zwischen den Image-Namen, wie sie in Ihren privaten Amazon ECR-Workflows definiert sind, und den Image-Namen in der Upstream-Registrierung.

Sie können Image-Zuordnungen mit Registries verwenden, die Amazon ECR Pull Through-Cache-Unterstützungen wie Quay unterstützt, und mit Registrierungen, die Pull-Through-Cache nicht unterstützen, wie Seqera. HealthOmics kopiert keine Bilder für Image-Zuordnungen, sodass Sie Ihr privates Repository selbst mit der Upstream-Registrierung synchronisieren.

> **Note**
>
> Eine Image-Zuordnung hat Vorrang vor Registrierungszuordnungen.

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

```
{
    "imageMappings": [
        {
            "sourceImage": "public.ecr.aws/aws-genomics/broadinstitute/gatk:4.6.0.2",
            "destinationImage": "111122223333.dkr.ecr.us-east-1.amazonaws.com/
broadinstitute/gatk:4.6.0.2"
        },
        {
            "sourceImage": "ubuntu:latest",
            "destinationImage": "111122223333.dkr.ecr.us-east-1.amazonaws.com/custom/
ubuntu:latest"
        }
    ]
```

```
}
```

Weitere Informationen zur Verwendung von Registrierungs- und Image-Zuordnungen beim Erstellen eines Workflows finden Sie unter Konfigurieren Sie die Parameter für die Pull-through-Cache-Zuordnung.

Bewährte Methoden für Container-Images

Folgen Sie diesen bewährten Methoden, wenn Sie Container-Images für erstellen HealthOmics Arbeitsabläufe.

Verwendung von Java in Amazon ECR-Container-Images

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

- Java-Anwendungen verwenden Stack-Speicher und Heap-Speicher. Standardmäßig ist der maximale Heap-Speicher ein Prozentsatz des gesamten verfügbaren Speichers im Container. Diese Standardeinstellung hängt von der jeweiligen JVM-Distribution und der JVM-Version ab. Konsultieren Sie daher die entsprechende Dokumentation für Ihre JVM oder legen Sie das Heap-Speichermaximum explizit mithilfe von Java-Befehlszeilenoptionen fest (wie `-Xmx`).
- Stellen Sie den maximalen Heap-Speicher nicht auf 100% der Speicherzuweisung des Containers ein, da der JVM-Stack ebenfalls Speicher benötigt. Speicher wird auch für den JVM-Garbage Collector und alle anderen Betriebssystemprozesse benötigt, die im Container ausgeführt werden.
- Einige Java-Anwendungen, wie GATK, können native Methodenaufrufen oder andere Optimierungen wie Speicherzuordnungsdateien verwenden. Diese Techniken erfordern Speicherzuweisungen, die „off-Heap“ ausgeführt werden, was durch den JVM-Parameter für den maximalen Heap nicht gesteuert wird.

  Wenn Sie wissen (oder vermuten), dass Ihre Java-Anwendung Off-Heap-Speicher zuweist, stellen Sie sicher, dass Ihre Aufgabenspeicherzuweisung den Off-Heap-Speicheranforderungen entspricht.

  Wenn diese Off-Heap-Zuweisungen dazu führen, dass dem Container der Arbeitsspeicher ausgeht, wird in der Regel kein Java-Code angezeigt `OutOfMemory` Fehler, weil die JVM diesen Speicher nicht kontrolliert.

Fügen Sie Aufgabeneingaben zu einem Amazon ECR-Container-Image hinzu

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

Es empfiehlt sich, die Verwendung von Skripten, Binärdateien und Bibliotheken zu vermeiden, die sich außerhalb des Container-Images einer Aufgabe befinden. Dies ist besonders wichtig bei der Verwendung `nf-core` Workflows, die eine verwenden `bin` Verzeichnis als Teil des Workflow-Pakets. Dieses Verzeichnis steht zwar für die Workflow-Aufgabe zur Verfügung, es ist jedoch als schreibgeschütztes Verzeichnis bereitgestellt. Die erforderlichen Ressourcen in diesem Verzeichnis sollten in das Task-Image kopiert und zur Laufzeit oder beim Erstellen des für die Aufgabe verwendeten Container-Images verfügbar gemacht werden.

Siehe HealthOmics feste Größenkontingente für Workflows für die maximale Größe des Container-Images HealthOmics unterstützt.

Verwenden Sie für Bilder, die nicht von PTC stammen, einen Digest oder eine Version anstelle eines Tags

Verweisen Sie bei Container-Images aus Registern, die keinen Pull-Through-Cache unterstützen, wie Seqera, NVIDIA NGC und Google Artifact Registry, auf einen Digest oder eine Version statt auf ein Tag wie `latest`. HealthOmics kopiert ein Bild, wenn eine Aufgabe es zum ersten Mal benötigt, und aktualisiert die Kopie nicht, sodass ein Tag, das sich im Upstream ändert, immer wieder in die frühere Kopie aufgelöst wird. Weitere Informationen finden Sie unter Verwendung von Drittanbieter-Registrierungen, die vom Pull-Through-Cache nicht unterstützt werden.

Das folgende Beispiel referenziert ein Seqera Container-Image nach Version und Digest:

```
community.wave.seqera.io/library/htslib_samtools:1.24--
d697cfb9dce007cd@sha256:a55ddea590e567a91df592300a960aa534cfc1bd16e7623e3938ec21f4f3df15
```

Umgebungsvariablen für HealthOmics Workflows

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

Alle HealthOmics Workflow-Variablen außer `AWS_REGION` und `AWS_HEALTHOMICS_RESOURCE_TYPE` beginne mit dem `AWS_WORKFLOW_` Präfix. Dieses Präfix

ist ein geschütztes Umgebungsvariablenpräfix. Verwenden Sie dieses Präfix nicht für Ihre eigenen Variablen in Workflow-Containern.

HealthOmics stellt die folgenden Workflow-Umgebungsvariablen bereit:

AWS_HEALTHOMICS_RESOURCE_TYPE

Diese Variable ist der Beschleunigertyp des aktiven Ressourcenprofils, oder `cpu` für ein CPU-only Profil. HealthOmics legt diese Variable für Aufgaben fest, die das verwenden `omicsResourceFallbackOrder` Direktive. Weitere Informationen finden Sie unter Umgebungsvariablen in Erweiterte Ressourcenkonfiguration.

AWS_REGION

Diese Variable ist die Region, in der der Container läuft.

AWS_WORKFLOW_RUN

Diese Variable ist der Name des aktuellen Laufs.

AWS_WORKFLOW_RUN_ID

Diese Variable ist die Lauf-ID des aktuellen Laufs.

AWS_WORKFLOW_RUN_UUID

Diese Variable ist die Run-UUID des aktuellen Laufs.

AWS_WORKFLOW_TASK

Diese Variable ist der Name der aktuellen Aufgabe.

AWS_WORKFLOW_TASK_ID

Diese Variable ist die Aufgabenkennung der aktuellen Aufgabe.

AWS_WORKFLOW_TASK_UUID

Diese Variable ist die Aufgaben-UUID der aktuellen Aufgabe.

Das folgende Beispiel zeigt typische Werte für jede Umgebungsvariable:

```
AWS Region: us-east-1
Workflow Run: arn:aws:omics:us-east-1:123456789012:run/6470304
```

```
Workflow Run ID: 6470304
Workflow Run UUID: a1b2c3d4-5678-90ab-cdef-EXAMPLE22222
Workflow Task: arn:aws:omics:us-east-1:123456789012:task/4192063
Workflow Task ID: 4192063
Workflow Task UUID: a1b2c3d4-5678-90ab-cdef-EXAMPLE33333
```

HealthOmics Workflow-README-Dateien

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

Zu den README-Anforderungen gehören:

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

Themen

- Verwenden Sie eine vorhandene README-Datei
- Bedingungen für das Rendern

Verwenden Sie eine vorhandene README-Datei

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

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

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

Bedingungen für das Rendern

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

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

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

Element	AWS Konsole
Textausrichtung	Ja

Bild und Link werden verwendet URLs

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

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

Quellanbieter	Seiten-URL	Bild-URL
GitHub	`https://github.com/{username}/{repo}/blob/{ref}/{path}`	`https://github.com/{username}/{repo}/blob/{ref}/{path}?raw=true` `https://raw.githubusercontent.com/{username}/{repo}/{ref}/{path}`
GitLab	`https://gitlab.com/{username}/{repo}/-/blob/{ref}/{path}`	`https://gitlab.com/{username}/{repo}/-/raw/{ref}/{path}`
Bitbucket	`https://bitbucket.org/{username}/{repo}/src/{ref}/{path}`	`https://bitbucket.org/{username}/{repo}/raw/{ref}/{path}`

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

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

Verbindung herstellen HealthOmics läuft auf einem Sentieon-Lizenzserver

HealthOmics unterstützt VPC-connected Workflows, das ist der empfohlene Ansatz für die Verbindung mit einem Lizenzserver eines Drittanbieters, z. B. einem Sentieon-Lizenzserver. Standardmäßig wird der Workflow wie folgt ausgeführt `RESTRICTED` Netzwerkmodus, der keinen Lizenzserver erreichen kann. Die folgenden Schritte zeigen, wie Sie Ihre Läufe mit einer VPC verbinden, damit sie Ihren Sentieon-Lizenzserver erreichen können.

Schritt 1: Holen Sie sich Ihre Lizenz von Sentieon

Senden Sie eine E-Mail an die Sentieon-Supportgruppe (support@sentieon.com), um eine Softwarelizenz anzufordern. Sentieon stellt eine Lizenz zur Verfügung, die Sie in Schritt 2 auf dem Lizenzserver installieren.

Schritt 2: Richten Sie Ihren Sentieon-Lizenzserver ein

Stellen Sie einen Sentieon-Lizenzserver auf einer Amazon EC2-Instance in Ihrer VPC bereit. Weitere Einzelheiten finden Sie unter Starten Sie einen Sentieon-Lizenzserver in Ihrer AWS-VPC in GitHub.

Schritt 3: Richten Sie Ihre VPC ein

Stellen Sie eine VPC bereit, indem Sie den Anweisungen folgen Beispiel für eine Healthomics-VPC-Konfiguration AWS CDK-Projekt. Ihre Läufe verwenden diese VPC, um den Lizenzserver aus Schritt 2 zu erreichen. Dieses Projekt stellt eine VPC, Subnetze, NAT-Gateways und eine Sicherheitsgruppe bereit und erstellt optional eine HealthOmics Konfiguration, wenn Sie die festlegen `networkingConfigurationName` Parameter. Außerdem werden VPC Flow Logs aktiviert, mit denen Sie Verbindungsprobleme zu Ihrem Lizenzserver überwachen und beheben können.

Ihre VPC muss die folgenden Anforderungen erfüllen:

- Ihre Sicherheitsgruppe muss ausgehendes TCP auf Port 8990 (dem Standardport des Sentieon-Lizenzservers) an die IP-Adresse oder den FQDN des Lizenzservers zulassen.
- Wenn Ihr Sentieon-Lizenzserver lokal oder auf einem anderen Server läuft AWS-Konto, konfigurieren Sie das Routing über VPN, AWS Direct Connect, oder Kontoübergreifender VPC-Zugriff.
- Ihre VPC muss sich in derselben befinden AWS-Region wie der Lauf.

Schritt 4: Erstellen Sie eine HealthOmics Aufbau

Wenn Sie die einstellen `networkingConfigurationName` Parameter in Schritt 3, die Konfiguration ist bereits vorhanden, und Sie können mit Schritt 5 fortfahren. Erstellen Sie andernfalls eine Konfiguration, die auf Ihre VPC-Subnetze und Sicherheitsgruppen verweist:

```
aws omics create-configuration \
  --name sentieon-vpc-config \
  --description "VPC config for Sentieon license access" \
  --run-configurations '{
    "vpcConfig": {
      "securityGroupIds": ["sg-0123456789abcdef0"],
      "subnetIds": [
        "subnet-0a1b2c3d4e5f6g7h8",
        "subnet-1a2b3c4d5e6f7g8h9"
      ]
    }
  }' \
  --region us-west-2
```

Warten Sie, bis der Konfigurationsstatus lautet `ACTIVE` bevor du einen Lauf startest. Dies dauert in der Regel bis zu 15 Minuten. Führen Sie den folgenden Befehl aus, um den Status zu überprüfen:

```
aws omics get-configuration \
  --name sentieon-vpc-config
```

> **Note**
>
> Wenn der Status angezeigt wird `FAILED`, überprüfen Sie die Konfigurationsdetails. Eine häufige Ursache bei der ersten Verwendung ist ein Fehlen `iam:CreateServiceLinkedRole` Erlaubnis. Ihre erste Konfiguration erstellt automatisch die HealthOmics dienstgebundene Rolle, für die diese Berechtigung erforderlich ist.

Schritt 5: Aktualisieren Sie Ihre Workflow-Aufgaben, um Ihren Lizenzserver zu verwenden

Aktualisieren Sie jede Aufgabe, die die Sentieon-Software verwendet, um Folgendes einzustellen `SENTIEON_LICENSE` Umgebungsvariable für die IP-Adresse oder den FQDN und den Port Ihres Lizenzservers (z. B. `licsrvr.your-domain:8990`). Die Aufgabe stellt über Ihr VPC-Netzwerk eine Verbindung zum Lizenzserver her. Es hat sich bewährt, die Adresse als Workflow-Parameter zu übergeben, anstatt sie fest zu codieren. Dadurch wird der Workflow auf Konten und VPC-Konfigurationen übertragbarer.

Nachdem Sie den Workflow aktualisiert haben, erstellen Sie einen Workflow oder eine Workflow-Version, bevor Sie die Ausführung starten. Beispiele finden Sie unter Sentieon-Amazon-Omics in GitHub.

Schritt 6: Starte deinen Lauf mit VPC-Netzwerken

Wenn du anrufst `start-run`, fügen Sie die folgenden Parameter hinzu:

- `--networking-mode VPC`
- `--configuration-name` mit dem Namen der Konfiguration, die Sie in Schritt 4 erstellt haben
- `--parameters` mit Ihrer Lizenzserveradresse, die Ihre Aufgaben aus Schritt 5 verwenden

HealthOmics verwendet die VPC-Einstellungen in dieser Konfiguration (Subnetze und Sicherheitsgruppen), sodass der Lauf Ihren Sentieon-Lizenzserver erreichen kann.

```
aws omics start-run \
  --workflow-id 1234567 \
  --role-arn arn:aws:iam::123456789012:role/OmicsWorkflowRole \
  --output-uri s3://amzn-s3-demo-bucket/outputs/ \
  --parameters '{"sentieon_license":"licsrvr.your-domain:8990"}' \
  --networking-mode VPC \
  --configuration-name sentieon-vpc-config \
  --region us-west-2
```

Schritt 7: Überprüfen Sie die Konnektivität

Starten Sie einen Testlauf und vergewissern Sie sich, dass er erfolgreich abgeschlossen wurde. Verwenden Sie dazu eine gültige Sentieon-Lizenz, die Sie über die VPC erworben haben. Prüfen Sie die Protokolle in der `/aws/omics/WorkflowLog` CloudWatch Protokolliert die Protokollgruppe für Erfolgs- oder Fehlermeldungen der Verbindung.

Wenn eine Sentieon-Aufgabe während der Lizenzvalidierung fehlschlägt, stellen Sie sicher, dass Ihre Sicherheitsgruppe ausgehendes TCP auf Port 8990 zulässt, und stellen Sie sicher, dass die VPC-Subnetze in Ihrem HealthOmics Die Konfiguration hat eine Route zum Lizenzserver. Wenn sich der Lizenzserver in einer anderen VPC oder einem anderen Konto befindet, stellen Sie sicher, dass die Peering- oder Transit-Gateway-Route aktiv ist. Bei lizenzspezifischen Problemen, wie z. B. abgelaufenen Lizenzen, wenden Sie sich an den Sentieon-Support unter support@sentieon.com.

Illumina DRAGEN auf HealthOmics private Arbeitsabläufe

Illumina DRAGEN v4.6 beinhaltet einen Softwaremodus (SW-Modus), der auf Standard-x86-CPU-Instances läuft und passende Ergebnisse liefert FPGA-based DRAGEN. Du kannst rennen DRAGEN SW-Modus als Aufgabe in einem HealthOmics privater Workflow in allen Regionen, in denen HealthOmics ist verfügbar. DRAGEN Der SW-Modus verwendet keine FPGA-Instanzen, und Sie geben keinen Task Accelerator an.

> **Note**
>
> DRAGEN ist Drittanbietersoftware, lizenziert von Illumina. HealthOmics bietet nicht DRAGEN lizenzen.

Wie arbeitet Illumina DRAGEN mit HealthOmics

Jeder DRAGEN Aufgabe validiert ihre Lizenz mit Illumina wenn die Aufgabe beginnt. Für die Lizenzvalidierung ist ein ausgehender Internetzugang erforderlich. In der Standardeinstellung HealthOmics führt use aus `RESTRICTED` Netzwerkmodus, der nur den Zugriff auf Amazon S3 und Amazon ECR in derselben Region ermöglicht und das öffentliche Internet blockiert. Um zu laufen DRAGEN, du startest den Einlauf `VPC` Netzwerkmodus mit einer VPC, die ausgehenden Datenverkehr über ein NAT-Gateway weiterleitet. Ein Lauf, der verwendet `RESTRICTED` Der Modus kann die Lizenz nicht validieren und DRAGEN Aufgabe schlägt fehl. DRAGEN Der SW-Modus wird auf x86-CPU-Instances ausgeführt und ist in allen Regionen verfügbar, in denen HealthOmics unterstützt private Workflows.

Um zu sehen, welche DRAGEN Pipelines werden unterstützt in DRAGEN SW-Modus, siehe die neueste Version Illumina DRAGEN Versionshinweise.

Voraussetzungen

- EIN DRAGEN Lizenz von Illumina.
- Ein privates Amazon ECR-Repository in der Region, in der Sie den Workflow ausführen.
- Eine VPC mit privaten Subnetzen, einem NAT-Gateway und einer Sicherheitsgruppe, die ausgehendes HTTPS (Port 443) zulässt. Ein Beispiel für ein VPC-Setup finden Sie im HealthOmics VPC-Konfiguration auf GitHub.
- EIN HealthOmics Servicerolle mit Zugriff auf Ihr Amazon ECR-Repository und Ihre Amazon S3-Eingabe- und Ausgabespeicherorte. Siehe Servicerollen für AWS HealthOmics.

Holen Sie sich eine DRAGEN-Lizenz

Abonnieren Sie DRAGEN durch Illumina. Nachdem Sie sich angemeldet haben, Illumina stellt die Anmeldeinformationen bereit, die DRAGEN Der Container verwendet, um die Lizenz zu validieren.

Erstellen Sie das DRAGEN-Container-Image und übertragen Sie es auf Amazon ECR

HealthOmics führt Container-Images aus privaten Amazon ECR-Repositorys in derselben Region wie der Workflow aus. Sie müssen das erstellen DRAGEN Bild, bevor Sie es in Ihr Repository übertragen. HealthOmics unterstützt `x86_64` Behälter. Installationsanweisungen finden Sie in der DRAGEN Referenz zum Softwaremodus.

Laden Sie das herunter DRAGEN RPM im SW-Modus für DRAGEN 4.6.2 oder höher von Illumina. Das Paket für Version 4.6.2 ist beispielsweise `dragen-softwaremode-4.6.2-4.6.2-1.el9.x86_64.rpm`.

Extrahieren Sie vor dem Erstellen des Images DRAGEN Installationsprogramm (`.run` file), ohne sie auszuführen, und überprüfen Sie die darin enthaltenen RPMs:

```
sh dragen-4.6.2-*.multi.el9.x86_64.run --noexec --keep --target ./extracted
ls ./extracted            # RPMs, flavors/, installer/, wrapper
```

Installieren Sie nur die Pakete, die für den SW-Modus erforderlich sind, und überspringen Sie den FPGA-Treiber, die Hostdienste und die Bitstream-Pakete:

Installieren	Überspringen (und warum)
`dragen-paths`	`dragen_drv` FPGA-XDMA-Kerneltreiber (ein Container kann keine Kernelmodule laden)
`dragen-host`	`dragen-host-services` (nein systemd in einer Aufgabe)
`dragen`	`dragen-bitstream-*` (im SW-Modus unbenutzt)
`dragen-platform-aws`	Keine

Beispiel für eine Docker-Datei:

```
FROM oraclelinux:9
ARG DRAGEN_VERSION=4.6.2

# dnf resolves the RPM's declared deps (bzip2-libs, procps-ng, libgomp, ...)
COPY dragen-softwaremode-${DRAGEN_VERSION}-${DRAGEN_VERSION}-1.el9.x86_64.rpm /tmp/
RUN dnf -y install /tmp/dragen-softwaremode-*.rpm unzip \
    && rm -f /tmp/*.rpm \
    && dnf clean all && rm -rf /var/cache/dnf

# AWS CLI v2, used by tasks that fetch the API key from Secrets Manager
RUN curl -sSL "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o /tmp/
awscli.zip \
    && unzip -q /tmp/awscli.zip -d /tmp && /tmp/aws/install && rm -rf /tmp/aws /tmp/
awscli.zip
```

```
ENV PATH="/opt/dragen/${DRAGEN_VERSION}/bin:${PATH}"
CMD ["dragen", "--version"]
```

HealthOmics erfordert `linux/amd64` Bilder. Erstellen Sie das Bild mit dem `--platform` Mit Flag können Sie von einer Apple-Silicon-Maschine aus kreuzkompilieren.

```
docker build --platform linux/amd64 -t dragen:4.6.2-sw .
```

Erstellen Sie ein privates Amazon ECR-Repository und pushen Sie dann Ihr DRAGEN Docker-Image in Ihr privates Amazon ECR-Repository. Weitere Informationen finden Sie unter Ein Docker-Image in ein privates Amazon ECR-Repository übertragen in der Amazon ECR-Benutzerhandbuch.

Fügen Sie eine Repository-Richtlinie hinzu, damit HealthOmics kann das Container-Image abrufen.

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

Fügen Sie die Datei hinzu, die alle Richtlinienerklärungen enthält.

```
aws ecr set-repository-policy --repository-name repository-name --region us-east-2 \
  --policy-text file://ecr-omics-policy.json
```

Für mehr HealthOmics Anforderungen finden Sie unter Container-Images für private Workflows.

Konfigurieren Sie das VPC-Netzwerk für die Lizenzvalidierung

Erstellen Sie die Netzwerkressourcen und eine HealthOmics Konfigurationsressource, wie beschrieben unter HealthOmics Workflows mit einer VPC verbinden. Deine Läufe müssen verwendet werden VPC Netzwerkmodus mit einem aktiven HealthOmics Konfiguration, die die folgenden Anforderungen erfüllt:

Einstellung	Anforderung
Subnets	Private Subnetze, eines pro Availability Zone, in Availability Zones HealthOmics funktioniert.
Routing	Route `0.0.0.0/0` von den privaten Subnetzen zu einem NAT-Gateway, sodass die Aufgabe erreicht `license.dragen.illumina.com` .
Sicherheitsgruppe	Ausgehendes TCP 443 zulassen. Um die Regel auf bestimmte Ziele zu beschränken, verwenden Sie den Illumina Endpunkte lizenzieren.

Stellen Sie sicher, dass Ihre Konfiguration die oben genannten Anforderungen erfüllt:

```
aws omics get-configuration --name my-vpc-config --region us-east-2
```

Sie können die Konfiguration für alle wiederverwenden DRAGEN läuft in der Region.

Lizenzschlüssel verwalten

Sie können eine Kopie Ihres Lizenzschlüssels in AWS Secrets Manager aufbewahren. Weil HealthOmics fügt keine Geheimnisse in die Aufgabenumgebung ein. Die Aufgabe selbst muss zur Laufzeit den Secrets Manager aufrufen, um den Schlüssel abzurufen, bevor sie ausgeführt wird `dragen`.

Speichere den Schlüssel als Geheimnis:

```
aws secretsmanager create-secret \
  --name dragen/illumina-api-key \
  --secret-string file://illumina-api-key.txt \
  --region us-east-2
```

Erteilen Sie Ihrer Servicerolle die Erlaubnis, das Geheimnis zu lesen (siehe Rolle bei der Ausführung des IAM-Dienstes), und rufen Sie dann den Schlüssel in der Aufgabe ab, bevor Sie sie aufrufen `dragen`. Das DRAGEN Das Container-Image muss das enthalten AWS CLI (oder ein AWS SDK), um den Anruf zu tätigen.

Fügen Sie beispielsweise Folgendes zum Befehlsblock einer Aufgabe hinzu, vor dem `dragen` anrufen. Es erstellt eine Klartextdatei mit dem Namen `illumina-api-key.txt` in der `/tmp/` Verzeichnis. Pass `/tmp/illumina-api-key.txt` zu `--api-key-file`.

```
# Fetch the API key at runtime; never echo it.
KEY_FILE=/tmp/illumina-api-key.txt
REGION=$(echo "~{api_key_secret_arn}" | cut -d: -f4)
(umask 077; aws secretsmanager get-secret-value \
  --secret-id "~{api_key_secret_arn}" --region "${REGION}" \
  --query SecretString --output text > "${KEY_FILE}")
```

Weil die Aufgabe bereits läuft VPC Netzwerkmodus, um den zu erreichen Illumina Lizenzserver, dieselbe NAT-Gateway-Route bietet Zugriff auf den Secrets Manager-API-Endpunkt. Um zu vermeiden, dass dieser Anruf über das NAT-Gateway weitergeleitet wird, fügen Sie stattdessen einen VPC-Schnittstellenendpunkt für Secrets Manager hinzu.

Sie können den Lizenzschlüssel auch als SSE-encrypted Amazon S3-Objekt und geben Sie den Schlüssel in den Eingabedateien für den Workflow oder die Workflow-Aufgaben an. Dadurch wird sichergestellt, dass der Schlüssel niemals in der Workflow-Definition oder in Amazon erscheint CloudWatch.

Amazon S3-Dateieingabe:

```
aws s3 cp illumina-api-key.txt \
  s3://amzn-s3-demo-bucket/secrets/illumina-api-key.txt \
  --sse AES256 --region us-east-2
```

Gewähren Sie den HealthOmics Service-Rolle Lesezugriff auf dieses Objekt (siehe Rolle bei der Ausführung des IAM-Dienstes) und referenzieren Sie es dann als Workflow-Parameter:

```
{
  "api_key_file": "s3://amzn-s3-demo-bucket/secrets/illumina-api-key.txt"
}
```

Stellen Sie sicher, dass Illumina Der API-Schlüssel in den Workflow-Eingaben ist auf Folgendes beschränkt DRAGEN Nur Aufgaben.

Definieren Sie die DRAGEN-Aufgabe

Das folgende WDL-Beispiel zeigt die Struktur eines Beispiels DRAGEN Aufgabe. Sie können den Lizenzschlüssel als Teil der Workflow-Eingabeparameter entweder als JSON-Datei oder als Inline-Werte übergeben. Sie sollten den Bild-URI auch als Workflow-Eingabe übergeben, damit HealthOmics kann den Zugriff überprüfen, bevor der Lauf beginnt. Weitere Informationen finden Sie unter HealthOmics Eingaben ausführen.

Der `dragen` Der Befehlszeilenprozess und die Ressourcenwerte hängen von Ihrer Analyse ab. Für die Ausführung des Prozesses sind mindestens 16 vCPUs und 128 GiB Arbeitsspeicher erforderlich. Illumina empfiehlt für einige Aufgaben bis zu 64 vCPUs und 256 GiB Arbeitsspeicher. Weitere Informationen finden Sie unter Aufgabenressourcen in einem HealthOmics Workflow-Definition.

DRAGEN Analyse ist I/O intensiv, und der Festplattendurchsatz kann die Leistung einschränken und sich auf die Gesamtlaufzeit auswirken. Illumina empfiehlt lokale temporäre Speichervolumes für optimale Leistung. Sie können mehr temporären Speicher pro Aufgabe anfordern, indem Sie den `disks` Direktive (oder gleichwertig) in Ihrer Workflow-Definition. Siehe Konfiguration der Größe des kurzlebigen Speichers.

Weitere Informationen finden Sie im IlluminaDRAGEN v4.6 Benutzerhandbuch für weitere Informationen zu den benötigten Ressourcen.

Beispiel für eine DRAGEN-Aufgabe (WDL)

```
version 1.0

workflow DragenSwGermline {
    input {
        String dragen_container
        File? hashtable_tar             # reuse an existing hash table; built from
 ref_fasta when absent
        File? ref_fasta
        File fastq_1
        File fastq_2
        String sample_id
        String api_key_secret_arn       # Secrets Manager ARN; the key itself never
 becomes a WDL value
        Boolean metering_preview = true # validate the API key without analysis or
 billing
    }

    call CheckLicenseSecret {
```

```
        input:
            secret_arn = api_key_secret_arn,
            dragen_container = dragen_container
    }

    if (!defined(hashtable_tar)) {
        call BuildHashTable {
            input:
                ref_fasta = select_first([ref_fasta]),
                dragen_container = dragen_container
        }
    }

    call DragenGermline {
        input:
            hashtable_tar = select_first([hashtable_tar, BuildHashTable.hashtable]),
            fastq_1 = fastq_1,
            fastq_2 = fastq_2,
            sample_id = sample_id,
            api_key_secret_arn = CheckLicenseSecret.checked_secret_arn,
            metering_preview = metering_preview,
            dragen_container = dragen_container
    }

    output {
        File? hashtable = BuildHashTable.hashtable
        Array[File] dragen_outputs = DragenGermline.outputs
        File dragen_log = DragenGermline.log
    }
}

# Fails fast if the run role can't read the secret or it's empty. Outputs only the ARN.
task CheckLicenseSecret {
    meta {
        volatile: true  # re-check access on every run
    }

    input {
        String secret_arn
        String dragen_container
    }

    command <<<
        set -euo pipefail
```

```
        REGION=$(echo "~{secret_arn}" | cut -d: -f4)
        KEY=$(aws secretsmanager get-secret-value \
            --secret-id "~{secret_arn}" --region "${REGION}" \
            --query SecretString --output text)
        if [ -z "${KEY}" ] || [ "${KEY}" = "None" ]; then
            echo "Secret has no SecretString; store it with --secret-string" >&2
            exit 1
        fi
        echo "API key secret is readable (${#KEY} characters)"
    >>>

    runtime {
        container: dragen_container
        cpu: 1
        memory: "2 GiB"
    }

    output {
        String checked_secret_arn = secret_arn
    }
}

# No license check, so this task is safe to call-cache.
task BuildHashTable {
    input {
        File ref_fasta
        String dragen_container
    }

    command <<<
        set -euo pipefail
        mkdir -p hashtable
        dragen --sw-mode --build-hash-table true \
            --ht-reference "~{ref_fasta}" \
            --ht-alt-liftover /opt/dragen/4.6.2/resources/ht_builder/hg38/bwa-
kit_hs38DH_liftover.sam \
            --ht-num-threads 16 \
            --output-directory hashtable
        tar -cf hashtable.tar -C hashtable .
    >>>

    runtime {
        container: dragen_container
        cpu: 16
```

```
        memory: "128 GiB"
        disks: "200 GiB"
    }

    output {
        File hashtable = "hashtable.tar"
    }
}

task DragenGermline {
    meta {
        volatile: true  # contacts the license server; never reuse cached results
    }

    input {
        File hashtable_tar
        File fastq_1
        File fastq_2
        String sample_id
        String api_key_secret_arn
        Boolean metering_preview
        String dragen_container
    }

    command <<<
        set -euo pipefail
        mkdir -p output /tmp/reference /tmp/dragen-intermediate

        # Fetch the API key at runtime; never echo it.
        KEY_FILE=/tmp/illumina-api-key.txt
        REGION=$(echo "~{api_key_secret_arn}" | cut -d: -f4)
        (umask 077; aws secretsmanager get-secret-value \
            --secret-id "~{api_key_secret_arn}" --region "${REGION}" \
            --query SecretString --output text > "${KEY_FILE}")

        tar -xf "~{hashtable_tar}" -C /tmp/reference

        set +e
        dragen --sw-mode \
            --api-key-file "${KEY_FILE}" \
            -r /tmp/reference \
            -1 "~{fastq_1}" -2 "~{fastq_2}" \
            --RGID "~{sample_id}-rg" --RGSM "~{sample_id}" \
            --enable-map-align true --enable-map-align-output true \
```

```
            --enable-variant-caller true \
            --validate-pangenome-reference false \
            --intermediate-results-dir /tmp/dragen-intermediate \
            --output-directory output --output-file-prefix "~{sample_id}" \
            ~{if metering_preview then "--metering-preview" else ""} \
            2>&1 | tee dragen.log
        RC=${PIPESTATUS[0]}
        set -e
        rm -f "${KEY_FILE}"
        [ "${RC}" -eq 0 ] || { echo "DRAGEN failed (${RC})" >&2; exit "${RC}"; }
        touch output/.keep  # metering preview writes no outputs
    >>>

    runtime {
        container: dragen_container
        cpu: 16
        memory: "128 GiB"
        disks: "300 GiB"
    }

    output {
        Array[File] outputs = glob("output/*")
        File log = "dragen.log"
    }
}
```

Erstellen Sie Ihren Arbeitsablauf

Folgen Sie den Schritten in Erstellen Sie einen privaten Workflow mit der Konsole, AWS CLI Befehle oder einen der AWS SDKs.

```
aws omics create-workflow --name dragen-sw-germline \
  --definition-zip fileb://workflow.zip --engine WDL --main main.wdl ...
```

Sie können das verwenden HealthOmics Linter für miniwdl oder Nextflow, um alle Fehler vor der Bereitstellung zu beheben.

Rolle bei der Ausführung des IAM-Dienstes

Erstellen Sie eine dedizierte Rolle mit den geringsten Rechten. Weitere detaillierte Anweisungen finden Sie unter IAM-Berechtigungen für HealthOmics.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "S3",
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:PutObject",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket",
        "arn:aws:s3:::amzn-s3-demo-bucket/*"
      ]
    },
    {
      "Sid": "Logs",
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "logs:DescribeLogStreams"
      ],
      "Resource": "arn:aws:logs:us-east-2:123456789012:log-group:/aws/omics/*"
    },
    {
      "Sid": "ECRPull",
      "Effect": "Allow",
      "Action": [
        "ecr:BatchCheckLayerAvailability",
        "ecr:GetDownloadUrlForLayer",
        "ecr:BatchGetImage"
      ],
      "Resource": "arn:aws:ecr:us-east-2:123456789012:repository/dragen"
    },
    {
      "Sid": "ECRAuth",
      "Effect": "Allow",
      "Action": "ecr:GetAuthorizationToken",
      "Resource": "*"
    }
```

```
  ]
}
```

Wenn Sie AWS Secrets Manager verwenden, benötigen Sie die Erlaubnis, das Geheimnis zu lesen.

```
{
  "Sid": "ReadDragenApiKeySecret",
  "Effect": "Allow",
  "Action": "secretsmanager:GetSecretValue",
  "Resource": "arn:aws:secretsmanager:us-east-2:123456789012:secret:secret-name-*"
}
```

Starten Sie den Lauf mit VPC-Netzwerk

Starte einen Lauf in VPC Netzwerkmodus mit Ihrer angewendeten Konfiguration und `scratch-storage-mode` gesetzt auf `LOCAL` um Scratch umzuleiten I/O auf die lokale Festplatte.

```
aws omics start-run \
  --configuration-name my-vpc-config \
  --name dragen-sw-germline-run \
  --networking-mode VPC \
  --output-uri s3://amzn-s3-demo-bucket/runs/ \
  --parameters file://parameters.json \
  --region us-east-2 \
  --role-arn arn:aws:iam::123456789012:role/dragen-sw-omics-role \
  --scratch-storage-mode LOCAL \
  --storage-type DYNAMIC \
  --workflow-id workflow-id
```

Weitere Informationen zum Einstellen Ihres Run-Speichertyps finden Sie unter Speichertypen in HealthOmics Workflows ausführen.

Überlegungen

Lokaler Datenträger

Wir empfehlen, dass du rennst DRAGEN mit schneller lokaler Festplatte auf dem Rechenknoten. Sie können den flüchtigen Scratch-Speicher aktivieren, der unter eingehängt ist `/tmp` durch Einstellung `scratchStorageMode` zu `LOCAL`. Weitere Informationen finden Sie unter Kurzlebiger Speicher für Workflow-Aufgaben HealthOmics.

Öffentliche Internetverbindung

DRAGEN validiert seine Lizenz über das Internet, aber die Standardlizenz HealthOmics `RESTRICTED` Der Netzwerkmodus blockiert ausgehende Anrufe ins öffentliche Internet. Um zu laufen DRAGEN, starte den Lauf mit `VPC` Netzwerkmodus aktiviert und eine Konfiguration mit Subnetzen bereitstellen, die ausgehenden Datenverkehr über ein NAT-Gateway weiterleiten, um `license.dragen.illumina.com` über HTTPS. Weitere Informationen hierzu finden Sie unter Internetzugang für VPC-connected Workflows.

Caching aufrufen

Wenn Sie das Caching von Anrufen verwenden, sollten Sie Aufgaben ausschließen, für die Lizenzprüfungen erforderlich sind. Bei der Lizenzüberprüfung wird ein externer Dienst kontaktiert, und ein ausgeführter Cache könnte sich auf die Aufgabenausgaben auswirken. Sie können eine Aufgabe vom Caching ausschließen, indem Sie eine Anweisung in der Workflow-Definition verwenden. Anweisungen zum Ausschließen einer Aufgabe vom Call-Caching finden Sie unter Engine-specific Caching-Funktionen.

Fehlerbehebung

Symptom	Wahrscheinliche Ursache	Action
Die Aufgabe schlägt beim Start mit einem Lizenz- oder Verbindungsfehler im Amazon fehl CloudWatch Protokolle, Protokoll	Der Lauf wurde gestartet in `RESTRICTED` Modus	Starte den Lauf mit `--networking-mode VPC` und einen Konfigurationsname n.
Derselbe Fehler in `VPC` Modus	Das private Subnetz hat keine Route zu einem NAT-Gatew ay, oder die Sicherheitsgruppe blockiert Port 443	Überprüfen Sie die Routing-Tabelle und die Sicherhei tsgruppe. Führen Sie den Konnektivitätstest in aus Testen der VPC-Konnektivität.
Lizenz abgelehnt	Testversion abgelaufen, erschöpft iCredits, oder falsche Anmeldeinformationen	Prüfen Sie den Status Ihres Abonnements mit Illumina.

Symptom	Wahrscheinliche Ursache	Action
`AccessDenied` das Bild ziehen	Der Servicerolle oder der Repository-Richtlinie fehlen die Amazon ECR-Berechtigungen	Siehe Amazon-ECR-Berechtigungen.
Der Lizenzschritt schlägt mit HTTP 403 sofort fehl	Benutzt `--lic-server` (ältere Anmeldeinformationen) statt `--api-key-file`	Wechseln Sie zu `--api-key-file` (oder `--lic-credentials`). `--lic-server` wird im SW-Modus nicht unterstützt.
Die Aufgabe schlägt vor jeder Lizenzprüfung fehl, selbst bei minimalen Eingaben oder Testeingaben	DRAGEN erzwingt eine Mindestanzahl von 16 vCPUs und über 120 GiB Arbeitsspeicher, nur um die Befehlszeile zu überprüfen	Benutze mindestens `omics.r.4xlarge` (16 vCPUs, 128 GiB) für beliebige DRAGEN Aufgabe im SW-Modus, einschließlich Probeläufen.

Verwandte Themen

- HealthOmics Workflows mit einer VPC verbinden
- Internetzugang für VPC-connected Workflows
- Container-Images für private Workflows
- Verbindung herstellen HealthOmics läuft auf einem Sentieon-Lizenzserver

Unterstützte Workflow-Sprachen für HealthOmics

HealthOmics Private Workflows unterstützen Workflow-Definitionen, die in drei Workflow-Sprachen verfasst sind. Jede Sprache hat ihre eigene Syntax, Funktionen und Versionsunterstützung HealthOmics.

Workflow-Engines auf einen Blick

In der folgenden Tabelle sind die unterstützten Workflow-Sprachen, Versionen und Hauptmerkmale zusammengefasst. In der Tabelle bezieht sich DSL auf die domänenspezifische Sprachversion, die jede Engine unterstützt.

Engine	Unterstützte Versionen	DSL//Spezifische Versionen	Schlüsselmerkmale
Nächster Fluss	v22.04, v23.10, v24.10, v25.10, v26.04	DSL 1 (nur v22.04), DSL 2	Groovy-based Datenflusssprache mit kanalgesteuerter Parallelität. In der Genomik in der NF-Core-Community weit verbreitet. Unterstützt Plugins (nf-schema, nf-prov, nf-fgbio, nf-core-utils).
WDL	—	Sprachspezifikation 1.0, 1.1, Entwicklung	Task-oriented Sprache mit explizite r input/output Typisierung. Zwei Motormodi: WDL (strikte Einhaltung der Spezifikationen) und WDL Milient (Cromwell-compatible mit impliziter Typumwandlung).
CWL	—	Sprachspezifikation 1.0, 1.1, 1.2	Standards-based, YAML-defined Workflow-Sprache, die von der Common Workflow Language-Community verwaltet wird. Portabel auf mehrere Ausführungsplattformen.

Auswahl einer Workflow-Sprache

Alle drei Sprachen werden vollständig unterstützt HealthOmics und können auf dieselben Rechen-, Speicher- und Netzwerkfunktionen zugreifen. Die folgenden Überlegungen könnten Ihnen bei der Auswahl helfen:

- Bestehende Pipelines — Wenn Ihre Organisation bereits Pipelines in einer Sprache unterhält, können Sie sie HealthOmics ohne Umschreiben verwenden. HealthOmicsunterstützt alle drei Sprachen mit derselben zugrunde liegenden Rechen-, Speicher- und Netzwerkinfrastruktur.

- Sprachmerkmale — Nextflow verwendet ein Groovy-based Datenflussmodell mit kanalgesteuerter Parallelität und unterstützt Plugins. WDL verwendet explizite Typisierung mit einer aufgabenorientierten Struktur. input/output CWL verwendet einen deklarativen YAML-based Ansatz, der für plattformübergreifende Portabilität konzipiert ist.
- Open-source Ökosystem — Jede Sprache hat eine aktive Open-Source-Community mit öffentlich zugänglichen Pipelines und Tools. Weitere Informationen finden Sie in den Spezifikationen für WDL, Nextflow oder CWL. https://www.commonwl.org/

Versionsdetails nach Sprache

Detaillierte Versionsunterstützung, sprachspezifische Funktionen und Anleitungen zur Konfiguration finden Sie auf den folgenden Seiten:

- Versionsunterstützung für HealthOmics Workflow-Definitionssprachen— Vollständige Versionsmatrix für alle drei Sprachen, einschließlich der Art und Weise, wie Versionen HealthOmics erkannt und ausgewählt werden.
- Besonderheiten der Nextflow-Workflow-Definition— Nextflow-DSL-Syntax, Direktiven, Plugins, Profile, Ausführungsberichte und Engine-Einstellungen.
- Versionshinweise zur Nextflow-Version für HealthOmics— Versionshinweise für jede unterstützte Nextflow-Version auf. HealthOmics
- Besonderheiten der WDL-Workflow-Definition— Unterstützung der WDL-Version, Typzwang, WDL-Nachsichtigkeitsmodus und Strukturbehandlung.
- WDL-Motoränderungsprotokoll für HealthOmics— Änderungsprotokoll der neuen Funktionen, Verbesserungen und veralteten Funktionen für die WDL-Engine aktiviert. HealthOmics
- Besonderheiten der CWL-Workflow-Definition— CWL-Versionsunterstützung, Eingabeformate und Docker-Anforderungen.

Lebenszyklus der Nextflow-Engine-Version

HealthOmics unterhält einen regelmäßigen Rhythmus für die Unterstützung neuer Versionen der Nextflow-Engine. Nextflow veröffentlicht in der Regel alle sechs Monate eine stabile Version und HealthOmics fügt fortlaufend Unterstützung für diese Versionen hinzu. Die vollständige Richtlinie zum Lebenszyklus der Nextflow-Versionen, einschließlich der Zeitpläne für veraltete Versionen und der Preise für die erweiterte Versionsverfügbarkeit, finden Sie auf der folgenden Seite:

- Aufbewahrungsrichtlinie für Nextflow-Versionen— Unterstützte Versionen, wie lange sie noch verfügbar sind, Veraltungsphasen, Preise für die erweiterte Versionsverfügbarkeit, Plugin-Management und Anleitungen zum Testen.

Verwandte Themen

- Workflow-Definitionsdateien in HealthOmics— So strukturieren Sie Ihre Workflow-Definition ZIP, Anforderungen und Beispiele.
- Private Workflows erstellen in HealthOmics— End-to-end Anleitung zur Erstellung eines Workflows auf HealthOmics.
- Starte einen Lauf in HealthOmics— So starten Sie einen Workflow-Lauf, einschließlich Engine-Einstellungen.

Themen

- Aufbewahrungsrichtlinie für Nextflow-Versionen
- Versionshinweise zur Nextflow-Version für HealthOmics
- WDL-Motoränderungsprotokoll für HealthOmics

Aufbewahrungsrichtlinie für Nextflow-Versionen

Unterstützte Versionen der Nextflow-Engine

HealthOmics unterstützt mehrere stabile Versionen der Nextflow-Engine. Jede Version unterstützt eine oder mehrere domänenspezifische Sprachversionen (DSL). Die folgende Tabelle zeigt die derzeit unterstützten Versionen.

Version	DSL-Unterstützung	HealthOmics Datum der Veröffentlichung	Status
v22.04	DSL 1 UND DSL 2	November 2022	Das Programm ist für den 30. November 2026 als veraltet geplant
v23.10	DSL 2	Oktober 2024	Das Programm wird voraussichtlich am 30. November 2027 außer Betrieb genommen

Version	DSL-Unterstützung	HealthOmics Datum der Veröffentlichung	Status
v24.10	DSL 2	August 2025	Wird mindestens bis November 2027 unterstützt
v25.10	DSL 2	April 2026	Wird mindestens bis Oktober 2028 unterstützt
v26.04	DSL 2	Juni 2026	Aktuell, wird mindestens bis April 2029 unterstützt

HealthOmics unterstützt die monatlichen „Edge“ -Veröffentlichungen nicht. HealthOmics unterstützt veröffentlichte Funktionen in jeder Version, jedoch keine Vorschaufunktionen.

Eine Version kann für die Ausführung von Workflows verfügbar bleiben, nachdem sie nicht mehr unterstützt wird. Unterstützte Versionen (das aktuelle Kalenderjahr plus die beiden Vorjahre) erhalten neue Funktionen und Kompatibilitätsverbesserungen. Deprecated-but-available Versionen erhalten nur Sicherheitspatches.

Wie lange bleiben die Versionen der Nextflow-Engine verfügbar

HealthOmics unterstützt Nextflow-Versionen, die im aktuellen Kalenderjahr veröffentlicht wurden, sowie Versionen, die in den beiden Vorjahren veröffentlicht wurden. Nur diese unterstützten Versionen erhalten Feature-Releases und Verbesserungen der Abwärtskompatibilität. In der Praxis:

- Im Jahr 2026: Familien im Alter von 26.x, 25.x und 24.x werden unterstützt.
- 2027: Wenn 27.04 veröffentlicht wird, werden Familien, für die ein Jahr unterstützt wird, auf 27.x, 26.x und 25.x umgestellt.

Wenn die erste stabile Version eines neuen Kalenderjahres allgemein verfügbar ist HealthOmics, ist die Familie mit dem ältesten Jahrgang als veraltet eingestuft.

Wenn eine Version eine Benachrichtigung über veraltete Versionen erhält (beginnt die Kündigungsfrist von T—12 bis T—0), beschränkt sich die Wartung ausschließlich auf Sicherheitspatches. Veralteten Versionen werden keine neuen Funktionen hinzugefügt.

Lebenszyklus veralteter Versionen

Wenn eine Nextflow-Version voraussichtlich als veraltet markiert ist HealthOmics, erhalten Sie eine Kündigungsfrist von mindestens 12 Monaten. Darauf folgt eine erweiterte Versionsverfügbarkeit von bis zu 24 Monaten in drei verschiedenen Phasen.

Phase	Zeitplan	Verfügbarkeit der Versionen	Wartur	Preisgestaltung	Erstellung von Arbeitsabläufen	Einen Lauf starten	Deine Aktion
Phase 1: Kündigungsfrist	T—12 bis T—0	Voll funktionsfähig für bestehende Workflows zum Standardpreis	Nur Sicherh tspatch	Standardtarif (keine Änderung)	In veraltete n Versionen nicht verfügbar	Voll funktionsfähig für bestehende Workflows	Migrieren Sie, um Preise für erweiterte Versionsverfügbarkeit zu vermeiden
Phase 2: Erweiterte Versionsverfügbarkeit	T — 0 bis T+24 Monate	Voll funktionsfähig für bestehende Workflows zum Preis der erweiterten Versionsverfügbarkeit	Nur Sicherh tspatch	2x (Monate 0—12), 4x (Monate 12—24)	In veraltete n Versionen nicht verfügbar	Voll funktionsfähig für bestehende Workflows, vorbehaltlich der Preise für erweiterte Versionsverfügbarkeit	Starten Sie neue Läufe mit unterstützten Versionen oder aktualisieren Sie die Workflow-Standardversion, um die Preisgestaltung für erweiterte Versionsverfügbarkeit zu beenden

Phase	Zeitplan	Verfügbarkeit der Versionen	Wartur	Preisgestaltung	Erstellung von Arbeitsabläufen	Einen Lauf starten	Deine Aktion
Phase 3: Ende des Lebens	Nach T+24 Monaten	Aus dem Dienst genommen	Keine	N/A — Läufe schlagen fehl	In veralteten Versionen nicht verfügbar	Die Ausführung schlägt auf nicht unterstützten (oder zurückgezogenen) Versionen fehl	Sie müssen ein Upgrade durchführen, um die Ausführung fortzusetzen oder neue Läufe auf unterstützten Versionen zu starten

Phase 1: Kündigungsfrist (T—12 bis T—0)

Während der Kündigungsfrist:

- Wenn Sie die veraltete Version verwenden, erhalten Sie per E-Mail, AWS Health Dashboard und Bannern in der Konsole Informationen über den Zeitplan und die betroffenen Workflows.
- Die Version funktioniert weiterhin für bestehende Workflows zum Standardpreis.
- Sie können keine neuen Workflows für veraltete Versionen erstellen, aber Sie können neue Läufe für bestehende Workflows zum Standardpreis starten.
- Es werden nur Sicherheitspatches angewendet. Für die veraltete Version werden keine neuen Funktionen oder Bugfixes bereitgestellt.
- Wir empfehlen, dass Sie testen und auf eine neuere unterstützte Version migrieren.
- Sie werden daran erinnert, Workflows zu migrieren, die die veraltete Version verwenden.

Phase 2: Erweiterte Versionsverfügbarkeit (T—0 bis T+24 Monate)

Nach dem Verfallsdatum (T):

- Für alle neuen Läufe, die mit veralteten Versionen gestartet wurden, fallen Preise für erweiterte Versionsverfügbarkeit an. Sie können diese Preisgestaltung vermeiden, indem Sie neue Läufe mit unterstützten Versionen starten oder Ihren Workflow aktualisieren, um eine unterstützte Version zu verwenden.
- Sie können keine neuen Workflows für die veralteten Versionen erstellen.
- Es werden nur Sicherheitspatches angewendet. Für die veraltete Version werden keine neuen Funktionen oder Bugfixes bereitgestellt.
- Sie erhalten Benachrichtigungen über die Preise für erweiterte Versionsverfügbarkeit, wenn Ihre Workflows weiterhin veraltete Versionen verwenden. Preise für erweiterte Versionsverfügbarkeit gelten nur, wenn in dieser Phase neue Läufe mit veralteten Versionen gestartet werden.
- Informationen zum Starten einer Ausführung auf unterstützten Versionen finden Sie unter Anheften der Nextflow-Engine-Version. Geben Sie die Nextflow-Engine-Einstellungen an

In der folgenden Tabelle sind die Preise für erweiterte Versionsverfügbarkeit zusammengefasst.

Zeitraum	Preisgestaltung
T — 0 bis T+12 Monate	Doppelter Basistarif für neue Läufe auf nicht unterstützten Versionen
T+12 bis T+24 Monate	4-facher Basistarif für neue Läufe auf nicht unterstützten Versionen

Phase 3: Ende der Nutzungsdauer (nach T+24 Monaten)

Nach Ablauf der Verfügbarkeitsdauer der erweiterten Version:

- Sie erhalten eine endgültige Benachrichtigung über die Außerbetriebnahme der Version, falls ein Workflow die zurückgezogene Version noch verwendet.
- Die Version wurde vollständig aus dem Dienst entfernt. In der veralteten Version können Sie keine neuen Workflows erstellen oder neue Läufe starten.
- Auf bestehende Workflow-Definitionen kann weiterhin zugegriffen werden, sie können jedoch in der veralteten Version nicht ausgeführt werden.

- Sie müssen Ihren Workflow aktualisieren, um eine unterstützte Engine-Version zu verwenden, oder eine unterstützte Version einbinden`StartRun`. Weitere Informationen zum Versions-Pinning beim Start der Ausführung finden Sie unterGeben Sie die Nextflow-Engine-Einstellungen an.

⚠ Versionssperre für aktive Läufe

Ein Workflow-Lauf wird immer mit der Engine-Version und den Plugin-Versionen abgeschlossen, mit denen er begonnen hat. Dies gilt für alle Versionsübergänge, einschließlich Patch-Updates, Versions-Außerbetriebnahme und Plugin-Updates. Versionsänderungen gelten nur für neue Läufe, die nach dem Update initiiert wurden, niemals für bereits laufende Läufe.

⚠ Es wird empfohlen, die Nextflow-Version beim Start anzuheften

Wir empfehlen dringend, dass Sie zur Laufzeit eine exakte Nextflow-Version anheften und rechtzeitig vor dem Verfallsdatum auf neueren Versionen testen. Weitere Informationen finden Sie unter Geben Sie die Nextflow-Engine-Einstellungen an.

Beschleunigte Außerbetriebnahme der Engine-Version

AWS behält sich das Recht vor, in seltenen Fällen, in denen eine Version nicht den Sicherheits- und Betriebsstandards entspricht, eine beschleunigte Außerbetriebnahme einer Nextflow-Engine-Version einzuleiten. AWS

In solchen Fällen:

- Wenn Sie betroffene Versionen verwenden, werden Sie sofort benachrichtigt, sobald eine beschleunigte Außerdienststellung festgestellt wird.
- Es gilt eine verkürzte Frist für die Außerbetriebnahme mit einer Kündigungsfrist von mindestens 90 Tagen, sofern dies betrieblich machbar ist.
- HealthOmics bietet Anleitungen zur Migration und, wo möglich, automatische Upgrade-Tools, um Sie zu unterstützen.

Versionen der Patch-Engine

Patch-Versionen (z. B. v24.10.0 auf v24.10.8) werden im HealthOmics Rahmen der routinemäßigen Servicewartung transparent aktualisiert. Patch-Versionen enthalten Sicherheitsupdates, Bugfixes und Abhängigkeitsupdates. Sie führen keine grundlegenden Änderungen ein.

Für Patch-Upgrades gilt weder eine Benachrichtigung des Kunden noch ein Kündigungsverfahren. Sie führen immer den neuesten Sicherheitspatch in der von Ihnen ausgewählten Version aus.

Versionen der DSL-Spezifikation

Da Nextflow die Spezifikationssprache und die Engine in einer Anwendung bündelt, ist die Unterstützung der DSL-Version direkt an die Unterstützung der Engine-Versionen gebunden:

- DSL 1 — Wird ausschließlich in Version 22.04 unterstützt. Wenn Version 22.04 eingestellt wird, endet die DSL 1-Unterstützung. Sie müssen auf DSL 2 migrieren.
- DSL 2 — Legacy Parser (Syntax V1) — Wird auf Version 22.04 bis Version 26.04 unterstützt. In Version 26.04 erfordert der Legacy-Parser eine explizite Anmeldung. `engineSettings.syntaxVersion = v1`
- DSL 2 — Strict Parser (Syntax V2) — Standard bei v26.04 und allen zukünftigen Versionen.

Vor dem Upgrade testen

Bevor Sie auf eine neuere Nextflow-Version aktualisieren, empfehlen wir Ihnen:

- Du kannst HealthOmics MCP-Server verwenden, um deine Workflows zu aktualisieren und gegen eine neuere Version zu validieren, einschließlich der Überprüfung auf wichtige Änderungen und Versionskompatibilität. Sie können die MCP-Server über Kiro CLI, Claude Code oder einen anderen MCP-compatible agentischen Endpunkt verwenden.
- Verwenden Sie den `engineSettings.engineVersion` Parameter in`StartRun`, um Workflows mit einer neueren Version zu testen, ohne Ihre Workflow-Definition zu ändern. Weitere Informationen finden Sie unter Geben Sie die Nextflow-Engine-Einstellungen an.
- In den https://www.nextflow.io/docs/latest/migrations/index.html Nextflow-Migrationsleitfäden finden Sie wichtige Änderungen zwischen den Versionen.
- Testen Sie bei Workflows, die von v25.10 oder früher auf v26.04 umgestellt werden, die Kompatibilität mit dem Strict Parser (V2) oder setzen Sie explizit auf. `syntaxVersion v1`
- Folgen Sie für Workflows auf DSL 1 (v22.04) vor dem Upgrade der Anleitung zur Migration von DSL 1.

Versionen des Nextflow-Plugins

HealthOmics installiert einen kuratierten Satz von Nextflow-Plugins für jede unterstützte Engine-Version vor. Das Plugin-Set ist für jede Engine-Version spezifisch und kann nicht geändert werden.

Engine-Version	Pre-installed Plugins
v22.04	Keine Plugin-Unterstützung
v23.10	nf-schema @2 .3.0, nf-Validierung @1 .1.1
v24.10	nf-schema @2 .3.0
v25.10	nf-schema @2 .6.1, nf-core-utils @0 .4.0, nf-prov @1 .7.0, nf-fgbio @1 .0.1
v26.04	nf-schema @2 .7.2, nf-core-utils @0 .4.0, nf-prov @1 .7.0, nf-fgbio @1 .0.1

> **Note**
>
> Für Nextflow v24.10 und höher ersetzt nf-schema das veraltete nf-validierungs-Plugin. Du kannst nf-validation nur auf Version 23.10 verwenden.

Wie werden Nextflow-Plugin-Versionen verwaltet

HealthOmics verwaltet eine Plugin-Version pro Engine-Version:

- Einzelne Version pro Plugin pro Engine — Nur die in der vorherigen Tabelle aufgeführte vorinstallierte Version ist verfügbar.
- Es werden nur vorinstallierte Plugins unterstützt — HealthOmics unterstützt nur die in der vorherigen Tabelle für jede Engine-Version aufgeführten Plugins. Plugin-Deklarationen in Ihrem`nextflow.config`, die auf Plugins verweisen, die für Ihre Engine-Version nicht vorinstalliert sind, werden nicht unterstützt.
- Automatische Updates — Wenn HealthOmics die vorinstallierte Plugin-Version für eine Engine aktualisiert wird, verwenden alle Workflows auf dieser Engine die aktualisierte Version bei ihrer nächsten Ausführung.

- Kein An- oder Abmelden — Plugin-Updates werden einheitlich angewendet. Es gibt keinen Mechanismus, um eine frühere Plugin-Version beizubehalten, und es gibt keine Option zur erweiterten Versionsverfügbarkeit für Plugins.
- Versionssperre für aktive Läufe — Eine Workflow-Ausführung, die mit einer bestimmten Plugin-Version beginnt, wird mit dieser Version abgeschlossen, auch wenn das Plugin während der Ausführung aktualisiert wird. Updates gelten nur für neue Läufe, die nach dem Update initiiert werden.

Was passiert, wenn eine Plugin-Version aktualisiert wird

Wichtige Versionsänderungen (z. B. nf-schema 2.x auf 3.x):

- Vor der Installation der Hauptversion erfolgt eine Vorankündigung von mindestens 30 Tagen.
- Während der 30-tägigen Nachfrist bleibt die vorherige Hauptversion in Gebrauch. Ihre Workflows verwenden weiterhin die Vorgängerversion, bis die Kulanzfrist abläuft.
- Nach Ablauf der Übergangsfrist ersetzt die neue Hauptversion die vorherige Version für alle Workflows.
- Die Vorankündigung erfolgt per E-Mail, AWS Health Dashboard und Bannern in der Konsole.

Kleinere Updates und Patch-Updates (z. B. nf-schema 2.3.0 bis 2.7.2): Wird automatisch ohne vorherige Ankündigung angewendet. Diese Updates führen keine grundlegenden Änderungen ein.

Plugin-Änderungen testen

Wenn eine größere Änderung der Plugin-Version angekündigt wird (30-tägige Nachfrist), empfehlen wir Folgendes:

- In den Versionshinweisen zum Plugin findest du wichtige Änderungen und Hinweise zur Migration.
- Testen Sie Ihre Workflows mit der neuen Plugin-Version, bevor die Kulanzfrist abläuft.
- Stellen Sie bei größeren Änderungen am NF-Schema sicher, dass die Parametervalidierungsdefinitionen Ihrer Pipeline mit dem Schemaformat der neuen Version kompatibel sind.

Versionshinweise zur Nextflow-Version für HealthOmics

Diese Seite enthält Versionshinweise für jede unterstützte Nextflow-Version von. HealthOmics Jeder Abschnitt fasst neue Funktionen, Verbesserungen und veraltete Funktionen für eine bestimmte Nextflow-Version zusammen.

Versionshinweise zu Nextflow v26.04

Die folgenden Tabellen fassen die HealthOmics Unterstützung für neue Funktionen, Verbesserungen und veraltete Funktionen zusammen, die in Nextflow Version 26.04 veröffentlicht wurden.

Neue Features und Verbesserungen

Feature	Von Version	HealthOmics Unterstützung	Hinweise
Strikter Syntaxparser (Standard)	26,04	Ja	Ab v26.04 standardmäßig aktiviert. Legacy-Parser ist `syntaxVersion: "v1"` in den Engine-Einstellungen verfügbar.
Datensatztypen	26.04	Ja	Weitere Informationen finden Sie unter Records in der Seqera Nextflow-Dokumentation.
Zusammenfassungen der Workflow-Ausgaben	26.04	Ja	Druckt nach Abschluss des Laufs eine Zusammenfassung der Workflow-Ausgaben. Das Ausgabeformat ist über `outputFormat` die Engine-Einstellungen konfigurierbar. Weitere Informationen finden Sie unter Geben Sie die Nextflow-Engine-Einstellungen an.
Agenten-Protokollierungsmodus	26.04	Ja	Konfigurierbar über die `agentMode` Motoreinstellungen. Weitere Informati

Feature	Von Version	HealthOmics Unterstützung	Hinweise
			onen finden Sie unter Geben Sie die Nextflow-Engine-Einstellungen an.
Modulsystem (Nextflow Registry)	26,04	Nein	HealthOmics Workflows werden in einem isolierten Netzwerk ohne ausgehenden Internetzugang ausgeführt. Sie können Module direkt in Ihre Workflow-Zip-Datei aufnehmen.
Statisches Tippen (Vorschau)	26,04	Nein	HealthOmics unterstützt keine Vorschaufunktionen.
Auto-load Sammlungsparameter aus Dateien	26,04	Nein	Erfordert statisches Tippen (Vorschau), das HealthOmics nicht unterstützt wird.
Multi-revision Pipelines auschecken	26,04	N/A	Nicht zutreffend. HealthOmics verwendet keinen Git-based Pipeline-Checkout.

Veraltungen

Veralteter Artikel	Von Version	Auswirkung	Empfohlene Aktion
`listFiles()` - Methode	26.04	Warnung vor Verfall	Ersetze durch. `listDirectory()`
`nextflow.enable.strict` - Flag	26.04	Wird nicht mehr benötigt	Aus der Konfiguration entfernen. Der strikte Modus ist jetzt die Standardeinstellung.
`manifest.defaultBranch`	26.04	Wird nicht mehr benötigt	Aus der Konfiguration entfernen. HealthOmics verwendet keinen Git-

Veralteter Artikel	Von Version	Auswirkung	Empfohlene Aktion
			based Pipeline-Checkout und hat diese Option nie unterstützt.

WDL-Motoränderungsprotokoll für HealthOmics

Diese Seite enthält das Änderungsprotokoll für die WDL-Engine sowohl für den strikten WDL-Modus als auch für den milden WDL-Modus. In jedem Abschnitt werden neue Funktionen, Verbesserungen und veraltete Funktionen zusammengefasst.

Neue Features und Verbesserungen

In der folgenden Tabelle sind neue Funktionen und Verbesserungen für die WDL-Engine on aufgeführt. HealthOmics

Datum der Änderung	Namen ändern	Änderungsbeschreibung	Empfohlene Aktion
22. September 2026	Konsistente Behandlung von CRLF-Zeil enenden	HealthOmics Die Zeilenend en von trims Windows-style (`\r\n`) sind in allen Bereichen `read_lines()` `read_tsv( )` einheitlich und `read_map()` entsprechen dem Verhalten von Standard-Linux-Tools.	Keine. Es sind keine Workflow-Änderungen erforderlich.
22. September 2026	`zip()cross()`, und verarbeitet `read_tsv()` nun Eingaben mit gemischten Typen korrekt. Manuelle Type-Cast -Problemu mgehungen	Die eingebauten Funktionen `zip()` und leiten `read_tsv( )` jetzt die `cross()` Typen ihrer Rückgabewerte korrekt ab. In früheren Versionen konnten diese Funktionen beim Kombinieren gemischter Typen (z. B. `Array[String]` und `Array[Int]` in einem `zip()` Aufruf) zu unerwarteten Typfehler	Wenn Ihr Arbeitsablauf explizite Typumwand lungen beinhaltet, die hinzugefügt wurden, um Tippfehler von`zip()`, oder zu umgehen `cross()read_tsv()` , können Sie sie entfernen . HealthOmics leitet jetzt

Datum der Änderung	Namen ändern	Änderungsbeschreibung	Empfohlene Aktion
	können entfernt werden.	n führen, sodass manuelle Typumwandlungen im WDL-Code erforderlich waren.	automatisch die richtigen Typen ab. Das Beibehalten vorhandener Typumwand lungen verursacht keine Fehler und beeinträchtigt auch nicht das Workflow-Verhalten.
22. September 2026	`basename( path, "")`fix gibt den Dateiname n statt einer leeren Zeichenfolge zurück.	`basename(path, "")` gibt jetzt den Dateinamen zurück. Zum Beispiel `basename("/ path/to/file.txt", "")` wird `"file.txt"` anstelle einer leeren Zeichenfolge zurückgeg eben.	Durchsuchen Sie Ihre Workflows nach `basename(x, "")` Anrufen. Falls vorhanden, stellen Sie sicher, dass die Rückgabe des Dateiname ns dem erwarteten Verhalten entspricht.

Datum der Änderung	Namen ändern	Änderungsbeschreibung	Empfohlene Aktion
22. September 2026	Float-key Kollision serkennung in `collect_by_key()`	`collect_by_key()` zuvor wurden `Float` Schlüssel als identisch behandelt, wenn sie mit 6 Dezimalstellen übereinstimmten. Verschiedene Werte wie `0.0000001` und `0.0000002` wurden zu einer Gruppe zusammengefasst. HealthOmics Erkennt diese Kollision ab dem Änderungsdatum bei der Workflow-Erstellung und schlägt den Workflow mit einem eindeutigen Fehler im GetWorkflow StatusMessage-Feld fehl, anstatt die Ergebnisse zu verwerfen. Ein Workflow, der zuvor beim Zusammenführen verschiedener `Float` Schlüssel erfolgreich abgeschlossen wurde, schlägt jetzt mit einem Kollision serkennungsfehler fehl wie `"statusMessage": "(collect_by_key_test.wdl Ln 17 Col 47) No such function: collect_by_key\n Map[Float, Array[String]] collected = collect_by_key(pairs_with_collision)"`	Sie können eine der drei Methoden verwenden: • `String`Schlüssel verwenden: Wechseln Sie `Array[Pair[Float, String]]` zu Ihren Schlüsselwerten `Array[Pair[String, String]]` und zitieren Sie sie. Tastenkombinationen sind immer unterschiedlich und schließen das Kollisionsrisiko vollständig aus. • `Int`Schlüssel verwenden: Ändern Sie den Schlüsseltyp, sodass Ihre Gleitkommawerte mit einer Potenz von 10 `Array[Pair[Int, String]]` multipliziert werden, um die erforderliche Genauigkeit als Ganzzahl beizubehalten. • Wenn `Float` Schlüssel unvermeidlich sind, runden Sie sie zunächst auf eine konsistente Genauigkeit ab, um sicherzustellen, dass jeder Schlüssel wirklich

Datum der Änderung	Namen ändern	Änderungsbeschreibung	Empfohlene Aktion
			anders ist und keine Daten stillschweigend zusammengeführt werden. Normalisieren Sie jeden Schlüssel , indem Sie ihn `round(value * 1000000.0) / 1000000.0` vor der Übergabe an verwenden. `collect_by_key()`

Datum der Änderung	Namen ändern	Änderungsbeschreibung	Empfohlene Aktion
Demnächst (60 Tage im Voraus ab 22. September 2026)	Die Division mit Gleitkommazahlen (/) gibt Dezimalzahlen zurück, keine gekürzten Ganzzahlen.	Aktuelles Verhalten: Der `/` Operator für `Float` Operanden gibt die nächste Ganzzahl zurück. Zum Beispiel gibt er `5.0 / 2` zurück`2.0`, nicht`2.5`. Wenn beide Operanden `Int` (z. B.`5 / 2`) sind, gibt das Ergebnis 2 zurück. Kommende Änderung (30. November 2026): Die Float-Division für Float-Operanden wird mit einer Genauigkeit von 16 Dezimalstellen zurückgegeben. Wird nach diesem Datum zurückgegeben`2.5`, `5.0 / 2.0` `150.0 / 7.0` wird `21.428...` anstelle eines Integer-Werts zurückgegeben. Die Genauigkeit des Ergebnisses hängt davon ab, wie es verarbeitet wird: Wenn der Ergebnistyp ist`Float`, wird der Wert mit einer Genauigkeit von bis zu 16 Dezimalstellen berechnet; wenn das Ergebnis zu a gezwungen wird, wird `String` es stattdessen mit 6 Dezimalstellen formatiert. Zum Beispiel `5.0 / 2.0` wird zurückgegeben und `21.428571` `4285714` als a zurückgegeben`2.5`, aber `21.428571`	Um unabhängig von der Engine-Version ganzzahlige Ergebnisse sicherzustellen, schließen Sie die Float-Division in `floor()` oder `ceil()` ein: • `floor(5.0 / 2.0)`→ kehrt immer zurück `2` • `ceil(5.0 / 2.0)`→ kehrt immer zurück `3` Wenn Ihr Workflow absichtlich Dezimalergebnisse von der Float-Division erwartet, sind keine Änderungen erforderlich. Beim nächsten Divisionsverhalten ab dem 30. November 2026 werden auch Dezimalstellen zurückgegeben. Hinweis: gibt immer `5/2` noch 2 zurück, da der `/` Operator auf ganze Zahlen angewendet wird.

Datum der Änderung	Namen ändern	Änderungsbeschreibung	Empfohlene Aktion
		wenn es in a `Float` interpoliert `150.0 / 7.0` wird. `String` `Int / Int`Ausdrücke wie `5 / 2` sind nicht betroffen und geben weiterhin 2 zurück.	

Der Arbeitsablauf zeigt an HealthOmics

HealthOmics verknüpft WDL- und Nextflow DSL2-Workflows während der Erstellung automatisch. Die integrierten Linters erkennen Fehler in Ihren Workflow-Definitionsdateien, die dazu führen können, dass Läufe fehlschlagen. Die Ergebnisse von Lint erscheinen in der `statusMessage` Feld des GetWorkflow Antwort.

Bei WDL-Workflows blockieren Lint-Ergebnisse — der Workflow wechselt nicht zu ACTIVE, wenn Fehler erkannt werden. Bei Nextflow DSL2-Workflows sind Lint-Ergebnisse nicht blockierend — der Workflow geht unabhängig von den Ergebnissen auf AKTIV über.

HealthOmics bietet auch externe Linters, die Sie lokal ausführen können, bevor Sie einen Workflow erstellen. Diese Linters können zusätzliche enthalten HealthOmics-specific Regeln.

Built-in Zeiger

HealthOmics führt eingebaute Linters serverseitig aus, wenn Sie aufrufen `CreateWorkflow` oder `CreateWorkflowVersion`.

Eingebautes WDL-Fußbett

Für WDL HealthOmics führt automatisch einen Linter aus, wenn Sie den Workflow erstellen. WDL Linting blockiert — wenn der Linter Fehler erkennt, geht der Workflow nicht in ACTIVE über. Die Linter-Ausgabe finden Sie in der `statusMessage` Feld des `GetWorkflow` Antwort. Verwenden Sie den folgenden CLI-Befehl, um die Statusausgabe abzurufen (verwenden Sie die Workflow-ID des WDL-Workflows, den Sie erstellt haben):

```
aws omics get-workflow
```

```
    –id 123456
    –query 'statusMessage'
```

Integriertes Linting von Nextflow

HealthOmics führt den in Nextflow eingebauten Strict DSL2-Linter aus (nf-lang/v2) automatisch während `CreateWorkflow` und `CreateWorkflowVersion` für alle unterstützten Nextflow DSL2-Versionen:

- Nextflow v22.04 (nur DSL2)
- Nextflow v23.10
- Nextflow v24.10
- Nextflow v25.10
- Nextflow v26.04

> **Note**
>
> Der integrierte Linter gilt nur für DSL2-Workflows. DSL1-Workflows sind nicht linted.

Der Nextflow-Linter arbeitet im nicht blockierenden Modus. Die Ergebnisse verhindern nicht, dass der Workflow AKTIV wird. Sie können die Ergebnisse jederzeit überprüfen, ohne den Prozess der Workflow-Erstellung zu unterbrechen.

> **Note**
>
> Der integrierte Nextflow-Linter validiert die Syntax Ihrer Workflow-Definition bei der Erstellung. Er unterscheidet sich von der strikten Syntax Parser verfügbar für Nextflow v26.04, das gesteuert wird von `engineSettings.syntaxVersion` und beeinflusst das Laufzeitverhalten. Der Linter wird bei der Erstellung in allen DSL2-Versionen ausgeführt, unabhängig davon, welchen Parser der Workflow zur Laufzeit verwendet.

Um die Lint-Ausgabe von Nextflow abzurufen

Lint-Ergebnisse erscheinen als strukturiertes JSON im `statusMessage` Feld des `GetWorkflow` Antwort. Verwenden Sie den folgenden CLI-Befehl, um die Lint-Ausgabe abzurufen:

```
aws omics get-workflow --id 1234567 --query 'statusMessage'
```

Das folgende Beispiel zeigt die JSON-Ausgabe aus dem Nextflow-Linter:

```
{
  "advisory": "Linting findings are from the strict Nextflow DSL2 (nf-lang/v2) linter
 and reflect the syntax that will be required in Nextflow 25/26 strict mode. On
 Nextflow 23/24 (legacy grammar) findings are advisory only.",
  "summary": {
    "errors": 2,
    "filesWithErrors": 1,
    "filesWithoutErrors": 0,
    "filesFormatted": 0,
    "warnings": 0,
    "filesWithWarnings": 0,
    "filesWithoutWarnings": 2
  },
  "errors": [
    {
      "filename": "nextflow-lint-legacy.nf",
      "startLine": 24,
      "startColumn": 5,
      "message": "`for` loops are no longer supported"
    },
    {
      "filename": "nextflow-lint-legacy.nf",
      "startLine": 24,
      "startColumn": 10,
      "message": "`i` is not defined"
    }
  ],
  "warnings": []
}
```

Lint-Ausgabefelder

Die JSON-Ausgabe enthält die folgenden Felder:

- `advisory` — Erläutert die versionsspezifische Anwendbarkeit der Ergebnisse. In Nextflow v22.04, v23.10 und v24.10 (ältere Grammatik) dienen die Ergebnisse nur als Ratgeber. In Nextflow v25.10 und v26.04 spiegeln die Ergebnisse strenge Anforderungen an die Modussyntax wider.
- `summary` — Enthält eine Gesamtanzahl von Fehlern und Warnungen für alle Dateien im Workflow.

- `errors` — Eine Reihe von Fehlerobjekten. Jeder Fehler enthält `filename`, `startLine`, `startColumn`, und `message`.
- `warnings` — Eine Reihe von Warnobjekten. Jede Warnung enthält dieselben Felder wie Fehlerobjekte.
- `truncated` — Ein Bedingungs-Flag, das erscheint, wenn die Lint-Ausgabe die Feldgrößenbeschränkungen überschreitet. Wenn eine Kürzung erfolgt, werden zuerst Warnungen gelöscht. Der Workflow bleibt AKTIV mit einem gültigen gekürzten Bericht.

Empfehlungen

Wir empfehlen den folgenden Workflow, um Lint-Ergebnisse zu beheben:

1. Überprüfen Sie die Ergebnisse von Lint in der `statusMessage` Feld.
2. Korrigieren Sie die Fehler in Ihrer Workflow-Definitionsdatei.
3. Re-create der Arbeitsablauf zum Generieren eines Clean-Lint-Berichts.

Externe Linter

Die folgenden externen Linter sind verfügbar:

- WDL — Ein öffentliches Amazon ECR-Image zum Ausführen eines WDL Linter auf der Amazon ECR Public Gallery.
- Nextflow — Ein öffentliches Amazon ECR-Bild zum Ausführen Linter-Regeln für Nextflow in der öffentlichen Amazon ECR-Galerie. Sie können auf den Quellcode von zugreifen Linter-Regeln für Nextflow Repository an GitHub.
- CWL — Nicht verfügbar.

Sie können WDL-, Nextflow- und CWL-Definitionen auch lokal validieren. Weitere Informationen finden Sie unter Workflow-Definitionen lokal validieren.

Die folgende Liste beschreibt den Unterschied zwischen integrierten und externen Lintern:

- Built-in Zeiger — Wird serverseitig ausgeführt während `CreateWorkflow` und `CreateWorkflowVersion`. Für WDL sind die Ergebnisse blockierend. Für Nextflow DSL2 sind die Ergebnisse nicht blockierend. Gilt nur für WDL- und Nextflow DSL2-Workflows.

- Externe Drucker — Lokal vor der Workflow-Erstellung ausführen. Kann zusätzliche enthalten HealthOmics-specific Kompatibilitätsregeln. Verfügbar als öffentliche Amazon ECR-Bilder.

Workflow-Definitionen lokal validieren

Sie können eine Workflow-Definition mit den Tools der Workflow-Sprache validieren, bevor Sie sie aufrufen `CreateWorkflow` oder `CreateWorkflowVersion`. Es fängt Probleme wie Syntaxfehler und Verweise ab, die sich nicht lösen lassen.

Bei der lokalen Validierung wird Ihre Definition mit der Workflow-Sprache verglichen, nicht mit HealthOmics Anforderungen. Für die Linters HealthOmics wird während der Workflow-Erstellung ausgeführt, siehe Der Arbeitsablauf zeigt an HealthOmics.

Einen Validator installieren

Bei den Validatoren handelt es sich um Software von Drittanbietern, die AWS hält nicht aufrecht.

- WDL — Installieren miniwdl, ein lokales Runner- und Entwickler-Toolkit für WDL.
- Nextflow — Installieren Nextflow v25.10 oder später. Frühere Versionen haben entweder nicht `nextflow lint` keine Warnungen befehlen oder melden.
- CWL — Installieren cwltool, die Referenzimplementierung der CWL-Standards.

> **Note**
>
> Ihr Lauf kann eine ältere Nextflow-Version verwenden als die, mit der Sie linten. Die `nextflow lint` Der Befehl wird mit der strengen Syntax analysiert, die Ihr Lauf möglicherweise nicht verwendet. Bevor Sie also eine Definition umschreiben, um einer Syntaxfindung zu entsprechen, überprüfen Sie, welche Syntaxversion Ihr Lauf verwendet. Eine Beschreibung finden Sie unter Geben Sie die Nextflow-Syntaxversion an. Für wie HealthOmics wählt die Nextflow-Version für einen Lauf aus, siehe Nextflow-Versionen.

Den Validator ausführen

Führen Sie den Validator für die Hauptdefinitionsdatei oder für Nextflow für das Verzeichnis aus, das ihn enthält.

WDL

```
miniwdl check main-file
```

Nextflow

```
nextflow lint definition-directory
```

CWL

```
cwltool --validate main-file
```

Bei einem Paket mit mehreren Dateien legen Sie die Dateien auf der Festplatte an und verwenden Sie dabei dieselben relativen Pfade wie in dem ZIP-Archiv, in das Sie hochladen HealthOmics. Verweise werden relativ zu der Datei aufgelöst, die sie deklariert, nicht zu dem Verzeichnis, in dem Sie den Befehl ausführen. Die wichtigsten Dateianforderungen finden Sie unter HealthOmics Anforderungen an die Workflow-Definition.

Pass `nextflow lint` das Verzeichnis und nicht die Hauptdatei, damit es auch überprüft `nextflow.config`, das es von der Hauptdatei aus nicht erreicht.

> **Note**
>
> Wenn du bestanden hast `-format`, `nextflow lint` schreibt die Definitionsdateien um, die keine Fehler enthalten. Das Formatieren mit v25.10 oder höher kann zu Syntax führen, die frühere Nextflow-Versionen ablehnen. Formatieren Sie also nicht, es sei denn, Ihr Rechenlauf verwendet die Version, mit der Sie formatieren.

Lesen der Validierungsergebnisse

miniwdl und `nextflow lint` Beenden Sie mit einem Status ungleich Null, wenn sie einen Fehler melden, aber mit einem Status von Null, wenn sie nur Warnungen melden. Lesen Sie also die Ausgabe, anstatt sich auf den Exit-Status zu verlassen.

- WDL — miniwdl meldet entweder den ersten Fehler, den es nicht beheben kann, oder eine Übersicht aller Dokumente, die es geladen hat, mit allen Lint-Ergebnissen.

- Nextflow — Nextflow meldet jeden Fehler und jede Warnung mit ihrer Datei, Zeile und Spalte und setzt die fehlerhafte Zeile in Anführungszeichen. Es endet mit einer Zählung der Dateien, die Fehler hatten, und einer Zählung derjenigen, die keine Fehler hatten. Eine Datei mit einer Warnung, aber ohne Fehler, wird also als Datei ohne Fehler gewertet. Um das Berichtsformat zu ändern, geben Sie Folgendes ein `-o`, zum Beispiel `-o json`.
- CWL — cwltool meldet entweder, dass es sich bei der Datei um eine gültige CWL-Datei handelt, oder dass die Überprüfung mit der Datei, Zeile und Spalte jedes Problems fehlgeschlagen ist.

> **Note**
>
> Nextflow-Versionen unterscheiden sich darin, wie viel sie melden. Nextflow v26.04 zählt Warnungen in seiner Zusammenfassung und nimmt sie auf `-o json` Ausgabe. Nextflow v25.10 druckt Warnungen nur in der Standardausgabe aus und meldet weder in seiner Zusammenfassung noch in seiner `-o json` Ausgabe.

HealthOmics Workflow-Operationen

Um einen privaten Workflow zu erstellen, benötigen Sie:

- Workflow definition file:Eine Workflow-DefinitionsdateiWDL, geschrieben inNextflow, oderCWL. Die Workflow-Definition spezifiziert die Eingaben und Ausgaben für Läufe, die den Workflow verwenden. Sie enthält auch Spezifikationen für die Ausführungen und Ausführungsaufgaben für Ihren Workflow, einschließlich Rechen- und Speicheranforderungen. Die Workflow-Definitionsdatei muss `.zip` ein Format haben. Weitere Informationen finden Sie unter Workflow-Definitionsdateien in HealthOmics.
  - Sie können Kiro CLI verwenden, um Ihre Workflow-Definitionsdateien in WDL, Nextflow und CWL zu erstellen und zu validieren. Weitere Informationen finden Sie unter Beispielaufforderungen für Kiro CLI und im Agentic Tutorial zur generativen KI unter. HealthOmics GitHub
- (Optional) Parameter template file:Eine Parameter-Vorlagendatei, in die geschrieben wurde. JSON Erstellen Sie die Datei, um die Ausführungsparameter zu definieren, oder HealthOmics generiert die Parametervorlage für Sie. Weitere Informationen finden Sie unter Parametervorlagendateien für HealthOmics Workflows.
- Amazon ECR container images:Erstellen Sie private Amazon ECR-Repositorys für jeden Container, der im Workflow verwendet wird. Erstellen Sie Container-Images für den Workflow

und speichern Sie sie in einem privaten Repository, oder synchronisieren Sie den Inhalt einer unterstützten Upstream-Registrierung mit Ihrem privaten ECR-Repository.

- (Optional) Sentieon licenses:Fordern Sie eine Sentieon Lizenz an, um die Sentieon Software in privaten Workflows zu verwenden.

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

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

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

- Name
- Description
- Standardspeichertyp
- Standardspeicherkapazität (mit Workflow-ID)
- README.md datei

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

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

> **Note**
>
> Nachdem Sie einen WDL- oder Nextflow DSL2-Workflow erstellt haben, verwenden Sie ihn, um das `statusMessage` Feld `GetWorkflow` auf Lint-Ergebnisse zu überprüfen. Weitere Informationen finden Sie unter Der Arbeitsablauf zeigt an HealthOmics.

Themen

- Erstellen Sie einen privaten Workflow
- Einen privaten Workflow aktualisieren
- Löschen Sie einen privaten Workflow
- Überprüfen Sie den Workflow-Status
- Referenzierung von Genomdateien aus einer Workflow-Definition

Erstellen Sie einen privaten Workflow

Erstellen Sie einen Workflow mit dem HealthOmics Konsole, AWS CLI-Befehle oder einer der AWS SDKs.

> **Note**
>
> Fügen Sie keine persönlich identifizierbaren Informationen (PII) in Workflow-Namen ein. Diese Namen sind sichtbar in CloudWatch Protokolle.

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

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

- Hosten Sie das Container-Image in der privaten ECR-Registry. Voraussetzungen für diese Option:
  - Erstellen Sie ein privates ECR-Repository oder wählen Sie ein vorhandenes Repository aus.
  - Konfigurieren Sie die ECR-Ressourcenrichtlinie wie unter beschrieben Amazon-ECR-Berechtigungen.
  - Laden Sie Ihr Container-Image in das private Repository hoch.
- Synchronisieren Sie das Container-Image mit dem Inhalt einer unterstützten Drittanbieter-Registrierung. Voraussetzungen für diese Option:
  - Konfigurieren Sie in der privaten ECR-Registry eine Pull-Through-Cache-Regel für jede Upstream-Registrierung. Weitere Informationen finden Sie unter Bildzuordnungen.

- Konfigurieren Sie die ECR-Ressourcenrichtlinie wie unter beschrieben Amazon-ECR-Berechtigungen.
- Erstellen Sie Vorlagen zur Repository-Erstellung. Die Vorlage definiert Einstellungen dafür, wann Amazon ECR das private Repository für eine Upstream-Registrierung erstellt.
- Erstellen Sie Präfixzuordnungen, um Container-Image-Referenzen in der Workflow-Definition den ECR-Cache-Namespaces neu zuzuordnen.

Wenn Sie einen Workflow erstellen, stellen Sie eine Workflow-Definition bereit, die Informationen über den Workflow, die Ausführungen und die Aufgaben enthält. HealthOmics kann die Workflow-Definition als .zip-Archiv abrufen, das lokal oder in einem Amazon S3-Bucket gespeichert ist, oder von einem unterstützten Git-based Repository.

Themen

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

Einen Workflow mithilfe der Konsole erstellen

Schritte zum Erstellen eines Workflows

1. Öffnen Sie die HealthOmics -Konsole.
2. Öffnen Sie bei Bedarf den linken Navigationsbereich (≡). Wählen Sie Private Workflows.
3. Auf der Private Arbeitsabläufe Seite, wählen Sie Arbeitsablauf erstellen.
4. Auf der Definieren Sie den Arbeitsablauf Seite, geben Sie die folgenden Informationen an:
    1. Name des Workflows: Ein unverwechselbarer Name für diesen Workflow. Wir empfehlen, Workflow-Namen festzulegen, um Ihre Läufe in der AWS HealthOmics Konsole und CloudWatch Protokolle.
    2. Beschreibung (optional): Eine Beschreibung dieses Workflows.
5. In der Workflow-Definition Panel, geben Sie die folgenden Informationen an:
    1. Sprache des Workflows (optional): Wählen Sie die Spezifikationssprache des Workflows aus. Andernfalls HealthOmics bestimmt die Sprache aus der Workflow-Definition.

2. Für Quelle der Workflow-Definition, wählen Sie, ob Sie den Definitionsordner aus einem importieren möchten Git-based Repository, einen Amazon S3-Standort oder von einem lokalen Laufwerk.

   a. Für Aus einem Repository-Service importieren:

   > **Note**
   >
   > HealthOmics unterstützt öffentliche und private Repositorien für GitHub, GitLab, Bitbucket, GitHub self-managed, GitLab self-managed.

      i. Wähle ein Verbindung um deine zu verbinden AWS Ressourcen für das externe Repository. Informationen zum Herstellen einer Verbindung finden Sie unter Stellen Sie eine Verbindung zu externen Code-Repositorys her.

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

      ii. In Vollständige Repository-ID, gib deine Repository-ID als user- einname/repo-Name. Stellen Sie sicher, dass Sie Zugriff auf die Dateien in diesem Repository haben.

      iii. In Quellenangabe (optional), geben Sie eine Repository-Quellreferenz ein (Branch-, Tag- oder Commit-ID). HealthOmics verwendet den Standard-Branch, wenn keine Quellreferenz angegeben ist.

      iv. In Dateimuster ausschließen, geben Sie die Dateimuster ein, um bestimmte Ordner, Dateien oder Erweiterungen auszuschließen. Dies hilft bei der Verwaltung der Datengröße beim Import von Repository-Dateien. Es gibt maximal 50 Muster, und die Muster müssen den Syntax des Glob-Musters. Beispiel:

         A. `tests/`

         B. `*.jpeg`

         C. `large_data.zip`

   b. Für Wählen Sie den Definitionsordner aus S3:

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

ii. Wenn Ihr Konto nicht Eigentümer des Amazon S3-Buckets ist, geben Sie den des Bucket-Besitzers ein AWS Konto-ID im Konto-ID des S3-Bucket-Besitzers. Diese Information ist erforderlich, damit HealthOmics kann den Besitz des Buckets überprüfen.

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

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

3. Hauptpfad der Workflow-Definitionsdatei (optional): Geben Sie den Dateipfad vom komprimierten Workflow-Definitionsordner oder -repository zum `main` datei. Dieser Parameter ist nicht erforderlich, wenn der Workflow-Definitionsordner nur eine Datei enthält oder wenn die Hauptdatei den Namen „main“ trägt.

6. In der README-Datei (optionales) Bedienfeld, wählen Sie das Quelle der README-Datei und geben Sie die folgenden Informationen an:

- Für Aus einem Repository-Service importieren, in README-Dateipfad, geben Sie den Pfad zur README-Datei im Repository ein.
- Für Wählen Sie eine Datei aus S3, in README-Datei in S3, geben Sie die Amazon S3-URI für die README-Datei ein.
- Für Wählen Sie eine Datei aus einer lokalen Quelle: in README — optional, wähle Wählen Sie eine Datei um die Markdown-Datei (.md) zum Hochladen auszuwählen.

7. In der Standard-Run-Speicherkonfiguration Panel, geben Sie den Standard-Laufspeichertyp und die Kapazität für Läufe an, die diesen Workflow verwenden:

1. Speichertyp ausführen: Wählen Sie, ob Sie statischen oder dynamischen Speicher als Standard für den temporären Laufspeicher verwenden möchten. Die Standardeinstellung ist statischer Speicher.
2. Führen Sie die Speicherkapazität aus (optional): Für den Speichertyp „Static Run“ können Sie die Standardmenge an Run-Speicher eingeben, die für diesen Workflow erforderlich ist. Der Standardwert für diesen Parameter ist 1200 GiB. Sie können diese Standardwerte überschreiben, wenn Sie einen Lauf starten.

8. Schlagworte (optional): Sie können diesem Workflow bis zu 50 Tags zuordnen.

9. Wählen Sie Weiter aus.

10. Auf der Fügen Sie Workflow-Parameter hinzu (optional) Seite, wählen Sie die Quelle des Parameters:

1. Für Aus der Workflow-Definitionsdatei analysieren, HealthOmics analysiert automatisch die Workflow-Parameter aus der Workflow-Definitionsdatei.
2. Für Geben Sie eine Parametervorlage aus dem Git-Repository an, verwenden Sie den Pfad zur Parameter-Vorlagendatei aus Ihrem Repository.
3. Für Wählen Sie eine JSON-Datei aus der lokalen Quelle, lade eine hoch JSON Datei aus einer lokalen Quelle, die die Parameter angibt.
4. Für Geben Sie Workflow-Parameter manuell ein, geben Sie Parameternamen und Beschreibungen manuell ein.

11. In der Parameter-Vorschau Im Bedienfeld können Sie die Parameter für diese Workflow-Version überprüfen oder ändern. Wenn Sie das wiederherstellen JSON Datei, verlieren Sie alle lokalen Änderungen, die Sie vorgenommen haben.
12. Wählen Sie Weiter aus.
13. Auf der Neuzuordnung von Container-URIs Seite, in der Regeln für die Zuordnung Im Panel können Sie URI-Zuordnungsregeln für Ihren Workflow definieren.

    Für Quelle der Zuordnungsdatei, wählen Sie eine der folgenden Optionen:

    - Keine — Keine Zuordnungsregeln erforderlich.
    - Wählen Sie eine JSON-Datei aus S3 — Geben Sie den S3-Speicherort für die Zuordnungsdatei an.
    - Wählen Sie eine JSON-Datei aus einer lokalen Quelle — Geben Sie den Speicherort der Zuordnungsdatei auf Ihrem lokalen Gerät an.
    - Geben Sie die Zuordnungen manuell ein — geben Sie die Registrierungszuordnungen und Image-Zuordnungen in das Zuordnungen Panel.
14. Die Konsole zeigt die Zuordnungen Panel. Wenn Sie eine Mapping-Quelldatei ausgewählt haben, zeigt die Konsole die Werte aus der Datei an.

    a. In Zuordnungen in der Registrierung, Sie können die Zuordnungen bearbeiten oder Zuordnungen hinzufügen (maximal 20 Registrierungszuordnungen).

       Jede Registrierungszuordnung enthält die folgenden Felder:

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

- (Fakultativ) Präfix für das vorgelagerte — Das Präfix des Repositorys in der Upstream-Registrierung.
- (Fakultativ) ECR-Konto-ID — Konto-ID des Kontos, dem das Upstream-Container-Image gehört.
- (Fakultativ) ARN für Anmeldeinformationen — Der ARN des Secrets Manager-Geheimnisses, das Ihre Anmeldeinformationen für die Registrierung enthält. Erforderlich für Registries, die Amazon ECR Pull-Through-Cache nicht unterstützt, wie Seqera, NVIDIA NGC, Google Artifact Registry und Google Container Registry. Weitere Informationen finden Sie unter Verwendung von Drittanbieter-Registrierungen, die vom Pull-Through-Cache nicht unterstützt werden.

b. In Bildzuordnungen, Sie können die Bildzuordnungen bearbeiten oder Zuordnungen hinzufügen (maximal 100 Bildzuordnungen).

Jede Bildzuordnung enthält die folgenden Felder:

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

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

Einen Workflow mit der CLI erstellen

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

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

Die `create-workflow` Die Operation gibt die folgende Antwort zurück:

```
{
  "arn": "arn:aws:omics:us-west-2:....",
  "id": "1234567",
  "status": "CREATING",
```

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

Optionale Parameter, die beim Erstellen eines Workflows verwendet werden sollen

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

Themen

- Geben Sie die Workflow-Definition an: Amazon S3-Standort
- Verwenden Sie die Workflow-Definition aus einem Git-based Endlager
- Geben Sie eine Readme-Datei an
- Geben Sie den an Haupt Definitionsdatei
- Geben Sie den Laufspeichertyp an
- Geben Sie die GPU-Konfiguration an
- Konfigurieren Sie die Parameter für die Pull-through-Cache-Zuordnung
- Konfigurieren Sie Cache-Mapping-Parameter ohne Pull-Through-Cache

Geben Sie die Workflow-Definition an: Amazon S3-Standort

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

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

Verwenden Sie die Workflow-Definition aus einem Git-based Endlager

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

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

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

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

Wenn Sie die Workflow-Definition aus einem angeben Git-based Repository, verwenden `parameter-template-path` um die Parameter-Vorlagendatei anzugeben. Wenn Sie diesen Parameter nicht angeben, HealthOmics erstellt den Workflow ohne eine Parametervorlage.

Das folgende Beispiel zeigt die Parameter, die sich auf den Inhalt von a beziehen Git-based privates Repositorium:

```
aws omics create-workflow \
    --name custom-variant \
    --description "Custom variant calling pipeline" \
    --engine "WDL" \
    --definition-repository '{
            "connectionArn": "arn:aws:codeconnections:us-
east-1:123456789012:connection/abcd1234-5678-90ab-cdef-1234567890ab",
            "fullRepositoryId": "myorg/my-genomics-workflows",
            "sourceReference": {
                "type": "BRANCH",
                "value": "main"
            },
```

```
            "excludeFilePatterns": ["tests/**", "*.log"]
      }' \
    --main "workflows/variant-calling/main.wdl" \
    --parameter-template-path "parameters/variant-calling-params.json" \
    --readme-path "docs/variant-calling-README.md" \
    --storage-type "DYNAMIC" \
```

Weitere Beispiele finden Sie im Blogbeitrag Wie erstelle ich ein AWS HealthOmics Workflows aus Inhalten in Git.

Geben Sie eine Readme-Datei an

Sie können den Speicherort der README-Datei mithilfe eines der folgenden Parameter angeben:

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

Verwenden Sie den Readme-Pfad nur in Verbindung mit Definition-Repository. Wenn Sie keinen README-Parameter angeben, HealthOmics importiert die Stammebene README.md Datei im Projektarchiv (falls sie existiert).

Die folgenden Beispiele zeigen, wie Sie den Speicherort der README-Datei mithilfe von readme-Pfad und readme-uri angeben.

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

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

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

Geben Sie den an Haupt Definitionsdatei

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

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

Geben Sie den Laufspeichertyp an

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

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

Geben Sie die GPU-Konfiguration an

Verwenden Sie den Accelerator-Parameter, um einen Workflow zu erstellen, der auf einer Instance mit beschleunigter Berechnung ausgeführt wird. Das folgende Beispiel zeigt, wie das verwendet wird `accelerators` Parameter. Sie geben die GPU-Konfiguration in der Workflow-Definition an. Siehe Accelerated-computing Instanzen.

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

Konfigurieren Sie die Parameter für die Pull-through-Cache-Zuordnung

Wenn Sie die Pull-Through-Cache-Zuordnungsfunktion von Amazon ECR verwenden, können Sie die Standardzuordnungen überschreiben. HealthOmics unterstützt Container-Registrierungen von Drittanbietern wie Amazon ECR Public, Kubernetes Container Image Registry, Quay, Docker Hub,

Microsoft Azure Container Registry, GitHub Container Registry und GitLab Container-Registrierung. Weitere Hinweise zu den Container-Setup-Parametern finden Sie unter Container-Images für private Workflows.

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

```
{
    "registryMappings": [
        {
            "upstreamRegistryUrl": "registry-1.docker.io",
            "ecrRepositoryPrefix": "docker-hub"
        },
        {
            "upstreamRegistryUrl": "quay.io",
            "ecrRepositoryPrefix": "quay",
            "ecrAccountId": "111122223333"
        },
        {

            "upstreamRegistryUrl": "public.ecr.aws",
            "ecrRepositoryPrefix": "ecr-public"
        }
    ],
    "imageMappings": [{
            "sourceImage": "docker.io/library/ubuntu:latest",
            "destinationImage": "111122223333.dkr.ecr.us-east-1.amazonaws.com/
healthomics-docker-2/custom/ubuntu:latest"
        },
        {
            "sourceImage": "nvcr.io/nvidia/k8s/dcgm-exporter",
            "destinationImage": "111122223333.dkr.ecr.us-east-1.amazonaws.com/
healthomics-nvidia/k8s/dcgm-exporter"
        }
    ]
}
```

Geben Sie die Zuordnungsparameter im Befehl create-workflow an:

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

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

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

Konfigurieren Sie Cache-Mapping-Parameter ohne Pull-Through-Cache

Wenn Sie Container-Images aus Registries verwenden, die Pull-Through-Cache nicht unterstützt, fügen Sie `credentialArn` zu jeder Zuordnung für eine Registrierung HealthOmics kopiert von und verwendet den vollständigen geheimen ARN. Weitere Informationen finden Sie unter Verwendung von Drittanbieter-Registrierungen, die vom Pull-Through-Cache nicht unterstützt werden.

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

```
{
    "registryMappings": [
        {
            "upstreamRegistryUrl": "community.wave.seqera.io",
            "ecrRepositoryPrefix": "wave-seqera",
            "credentialArn": "arn:aws:secretsmanager:us-
east-1:111122223333:secret:omics-registry/seqera-EXAMPLE"
        },
        {
            "upstreamRegistryUrl": "nvcr.io",
            "ecrRepositoryPrefix": "nvidia-ngc",
            "credentialArn": "arn:aws:secretsmanager:us-
east-1:111122223333:secret:omics-registry/nvidia-ngc-EXAMPLE"
        }
    ]
}
```

Geben Sie die Zuordnungsparameter im Befehl create-workflow an:

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

Sie können die Registrierungsübersicht auch direkt bereitstellen. Im folgenden Beispiel wird ein Workflow mit einer Inline-Registrierungsübersicht für Seqera Containers erstellt:

```
aws omics create-workflow \
  --name my-workflow \
  --engine NEXTFLOW \
  --definition-uri s3://amzn-s3-demo-bucket1/my-definition.zip \
  --container-registry-map '{
    "registryMappings": [
      {
        "upstreamRegistryUrl": "community.wave.seqera.io",
        "ecrRepositoryPrefix": "wave-seqera",
        "credentialArn": "arn:aws:secretsmanager:us-east-1:111122223333:secret:omics-
registry/seqera-EXAMPLE"
      }
    ]
  }'
```

Einen Workflow mithilfe eines SDK erstellen

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

```
import boto3

omics = boto3.client('omics')

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

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

Einen privaten Workflow aktualisieren

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

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

Themen

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

Aktualisierung eines Workflows mithilfe der Konsole

Schritte zum Aktualisieren eines Workflows

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

Aktualisieren eines Workflows mit der CLI

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

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

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

Aktualisierung eines Workflows mithilfe eines SDK

Sie können einen Workflow mit einem der aktualisieren SDKs.

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

```
import boto3

omics = boto3.client('omics')

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

Löschen Sie einen privaten Workflow

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

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

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

Themen

- Löschen eines Workflows mithilfe der Konsole
- Löschen eines Workflows mit der CLI
- Löschen eines Workflows mithilfe eines SDK

Löschen eines Workflows mithilfe der Konsole

Um einen Workflow zu löschen

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

Löschen eines Workflows mit der CLI

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

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

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

Löschen eines Workflows mithilfe eines SDK

Sie können einen Workflow mit einem der löschen SDKs.

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

```
import boto3
```

```
omics = boto3.client('omics')

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

Überprüfen Sie den Workflow-Status

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

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

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

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

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

> **Note**
>
> HealthOmics Führt bei Nextflow DSL2-Workflows den integrierten Strict-Linter während der Erstellung aus. Wenn der Linter Fehler oder Warnungen erkennt, wechselt der Workflow trotzdem zu ACTIVE, wobei das `statusMessage` Feld strukturierte Lint-Ergebnisse enthält. Wenn Sie einen Lauf für einen Workflow mit Linter-Fehlern starten, schlägt Ihr Lauf möglicherweise aufgrund einer falschen Workflow-Definition fehl. Ein Beispiel für die `GetWorkflow` Antwort mit Lint-Ergebnissen finden Sie unterDer Arbeitsablauf zeigt an HealthOmics.

Referenzierung von Genomdateien aus einer Workflow-Definition

Auf ein HealthOmics Referenzspeicherobjekt kann mit einem URI wie dem folgenden verwiesen werden. Verwenden Sie Ihr eigenes und *account IDreference store ID*, *reference ID* sofern angegeben.

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

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

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

Bei Verwendung eines Sequenzlesesatzes würde sich ein ähnliches Muster ergeben, wie in der Abbildung gezeigt.

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

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

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

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

Das Folgende ist ein Beispiel für eine JSON-Eingabedatei, die zwei Omics Storage-URIs verwendet.

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

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

Workflow-Versionierung in HealthOmics

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

Workflow-Versionen bieten die folgenden Vorteile:

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

  `arn:aws:omics:us-west-2:123456789012:workflow/1234567/version/myUniqueVersionName`
- Wenn Sie Eigentümer eines gemeinsam genutzten Workflows sind, können Sie den Workflow aktualisieren, ohne die Abonnenten zu stören (die die vorherige Version weiterhin verwenden

können). Abonnenten können auf alle Workflow-Versionen zugreifen. Wenn Sie eine neue Version erstellen, müssen Sie den Workflow nicht erneut teilen.

- Wenn Sie eine Workflow-Ausführung starten, können Sie die Workflow-Version angeben.
  - Benutzer können sich dafür entscheiden, für Produktionsläufe eine stabile Version zu verwenden und die neueste Version für einen Testlauf auszuprobieren.
  - Benutzer können zur vorherigen Version eines Workflows zurückkehren, wenn sie auf Probleme mit der neuen Version stoßen.
  - Abonnenten eines gemeinsam genutzten Workflows können wählen, welche Version verwendet werden soll.

Themen

- Standard-Workflow-Version
- Eine Workflow-Version erstellen
- Eine Workflow-Version aktualisieren
- Löschen Sie eine Workflow-Version

Standard-Workflow-Version

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

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

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

Eine Workflow-Version erstellen

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

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

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

Verwenden Sie optional das Feld mit der Versionsbeschreibung, um Hinweise zu dieser Version hinzuzufügen. Beispiel: Fix for syntax error in workflow definition.

> **Note**
>
> Fügen Sie dem Versionsnamen keine persönlich identifizierbaren Informationen (PII) hinzu. Versionsnamen werden im ARN der Workflow-Version angezeigt.

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

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

Themen

- Erstellen Sie mithilfe der Konsole eine Workflow-Version
- Erstellen Sie eine Workflow-Version mit der CLI
- Erstellen Sie eine Workflow-Version mithilfe eines SDK
- Überprüfen Sie den Status einer Workflow-Version

Erstellen Sie mithilfe der Konsole eine Workflow-Version

Schritte zum Erstellen einer Workflow-Version

1. Öffnen Sie die HealthOmics -Konsole.
2. Öffnen Sie bei Bedarf den linken Navigationsbereich (≡). Wählen Sie Private Workflows.
3. Auf der Private Arbeitsabläufe Seite, wählen Sie den Workflow für die neue Version.
4. Auf der Einzelheiten zum Arbeitsablauf Seite, wählen Sie Neue Version erstellen.
5. Auf der Version erstellen Seite, geben Sie die folgenden Informationen an:
    1. Name der Version: Geben Sie einen Namen für die Workflow-Version ein, der im gesamten Workflow eindeutig ist.
    2. Beschreibung der Version (optional): Sie können das Beschreibungsfeld verwenden, um Hinweise zu dieser Version hinzuzufügen.
6. In der Workflow-Definition Panel, geben Sie die folgenden Informationen an:
    1. Sprache des Workflows (optional): Wählen Sie die Spezifikationssprache für die Workflow-Version aus. Andernfalls HealthOmics bestimmt die Sprache aus der Workflow-Definition.
    2. Für Quelle der Workflow-Definition, wählen Sie, ob Sie den Definitionsordner aus einem importieren möchten Git-based Repository, einen Amazon S3-Standort oder von einem lokalen Laufwerk.
        a. Für Aus einem Repository-Service importieren:

        > **Note**
        >
        > HealthOmics unterstützt öffentliche und private Repositorien für GitHub, GitLab, Bitbucket, GitHub self-managed, GitLab self-managed.

        i. Wähle ein Verbindung um deine zu verbinden AWS Ressourcen für das externe Repository. Informationen zum Herstellen einer Verbindung finden Sie unter Stellen Sie eine Verbindung zu externen Code-Repositorys her.

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

ii. In Vollständige Repository-ID, gib deine Repository-ID als user- einname/repo-Name. Stellen Sie sicher, dass Sie Zugriff auf die Dateien in diesem Repository haben.

iii. In Quellenangabe (optional), geben Sie eine Repository-Quellreferenz ein (Branch-, Tag- oder Commit-ID). HealthOmics verwendet den Standard-Branch, wenn keine Quellreferenz angegeben ist.

iv. In Dateimuster ausschließen, geben Sie die Dateimuster ein, um bestimmte Ordner, Dateien oder Erweiterungen auszuschließen. Dies hilft bei der Verwaltung der Datengröße beim Import von Repository-Dateien. Es gibt maximal 50 Muster, und die Muster müssen den Syntax des Glob-Musters. Beispiel:

A. `tests/`

B. `*.jpeg`

C. `large_data.zip`

b. Für Wählen Sie den Definitionsordner aus S3:

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

ii. Wenn Ihr Konto nicht Eigentümer des Amazon S3-Buckets ist, geben Sie den des Bucket-Besitzers ein AWS Konto-ID im Konto-ID des S3-Bucket-Besitzers. Diese Information ist erforderlich, damit HealthOmics kann den Besitz des Buckets überprüfen.

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

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

3. Hauptpfad der Workflow-Definitionsdatei (optional): Geben Sie den Dateipfad vom komprimierten Workflow-Definitionsordner oder -repository zum `main` datei. Dieser Parameter ist nicht erforderlich, wenn der Workflow-Definitionsordner nur eine Datei enthält oder wenn die Hauptdatei den Namen „main“ trägt.

7. In der README-Datei (optionales) Bedienfeld, wählen Sie das Quelle der README-Datei und geben Sie die folgenden Informationen an:

- Für Aus einem Repository-Service importieren, in README-Dateipfad, geben Sie den Pfad zur README-Datei im Repository ein.
- Für Wählen Sie eine Datei aus S3, in README-Datei in S3, geben Sie die Amazon S3-URI für die README-Datei ein.
- Für Wählen Sie eine Datei aus einer lokalen Quelle: in README — optional, wähle Wählen Sie eine Datei um die Markdown-Datei (.md) zum Hochladen auszuwählen.

8. In der Standard-Run-Speicherkonfiguration Panel, geben Sie den Standard-Laufspeichertyp und die Kapazität für Läufe an, die diesen Workflow verwenden:

    1. Speichertyp ausführen: Wählen Sie, ob Sie statischen oder dynamischen Speicher als Standard für den temporären Laufspeicher verwenden möchten. Die Standardeinstellung ist statischer Speicher.
    2. Führen Sie die Speicherkapazität aus (optional): Für den Speichertyp „Static Run“ können Sie die Standardmenge an Run-Speicher eingeben, die für diesen Workflow erforderlich ist. Der Standardwert für diesen Parameter ist 1200 GiB. Sie können diese Standardwerte überschreiben, wenn Sie einen Lauf starten.

9. Schlagworte (optional): Sie können dieser Workflow-Version bis zu 50 Tags zuordnen.
10. Wählen Sie Weiter aus.
11. Auf der Fügen Sie Workflow-Parameter hinzu (optional) Seite, wählen Sie die Quelle des Parameters:

    1. Für Aus der Workflow-Definitionsdatei analysieren, HealthOmics analysiert automatisch die Workflow-Parameter aus der Workflow-Definitionsdatei.
    2. Für Geben Sie eine Parametervorlage aus dem Git-Repository an, verwenden Sie den Pfad zur Parameter-Vorlagendatei aus Ihrem Repository.
    3. Für Wählen Sie eine JSON-Datei aus der lokalen Quelle, lade eine hoch JSON Datei aus einer lokalen Quelle, die die Parameter angibt.
    4. Für Geben Sie Workflow-Parameter manuell ein, geben Sie Parameternamen und Beschreibungen manuell ein.

12. In der Parameter-Vorschau Im Bedienfeld können Sie die Parameter für diese Workflow-Version überprüfen oder ändern. Wenn Sie das wiederherstellen JSON Datei, verlieren Sie alle lokalen Änderungen, die Sie vorgenommen haben.
13. Auf der Neuzuordnung von Container-URIs Seite, in der Regeln für die Zuordnung Im Panel können Sie URI-Zuordnungsregeln für Ihren Workflow definieren.

Für Quelle der Zuordnungsdatei, wählen Sie eine der folgenden Optionen:

- Keine — Keine Zuordnungsregeln erforderlich.
- Wählen Sie eine JSON-Datei aus S3 — Geben Sie den S3-Speicherort für die Zuordnungsdatei an.
- Wählen Sie eine JSON-Datei aus einer lokalen Quelle — Geben Sie den Speicherort der Zuordnungsdatei auf Ihrem lokalen Gerät an.
- Geben Sie die Zuordnungen manuell ein — geben Sie die Registrierungszuordnungen und Image-Zuordnungen in das Zuordnungen Panel.

14. Die Konsole zeigt die Zuordnungen Panel. Wenn Sie eine Mapping-Quelldatei ausgewählt haben, zeigt die Konsole die Werte aus der Datei an.

    a. In Zuordnungen in der Registrierung, Sie können die Zuordnungen bearbeiten oder Zuordnungen hinzufügen (maximal 20 Registrierungszuordnungen).

       Jede Registrierungszuordnung enthält die folgenden Felder:

       - URL der Upstream-Registrierung — Die URI der Upstream-Registrierung.
       - ECR-Repository-Präfix — Das Repository-Präfix, das im privaten Amazon ECR-Repository verwendet werden soll.
       - (Fakultativ) Präfix für das vorgelagerte — Das Präfix des Repositorys in der Upstream-Registrierung.
       - (Fakultativ) ECR-Konto-ID — Konto-ID des Kontos, dem das Upstream-Container-Image gehört.
       - (Fakultativ) ARN für Anmeldeinformationen — Der ARN des Secrets Manager-Geheimnisses, das Ihre Anmeldeinformationen für die Registrierung enthält. Erforderlich für Registries, die Amazon ECR Pull-Through-Cache nicht unterstützt, wie Seqera, NVIDIA NGC, Google Artifact Registry und Google Container Registry. Weitere Informationen finden Sie unter Verwendung von Drittanbieter-Registrierungen, die vom Pull-Through-Cache nicht unterstützt werden.

    b. In Bildzuordnungen, Sie können die Bildzuordnungen bearbeiten oder Zuordnungen hinzufügen (maximal 100 Bildzuordnungen).

       Jede Bildzuordnung enthält die folgenden Felder:

       - Quellbild — Gibt den URI des Quellbilds in der Upstream-Registrierung an.

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

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

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

Erstellen Sie eine Workflow-Version mit der CLI

Sie können eine Workflow-Version mit dem erstellen `CreateWorkflowVersion` API-Betrieb. Für optionale Parameter HealthOmics verwendet die folgenden Standardwerte:

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

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

```
aws omics create-workflow-version \
--workflow-id 1234567  \
--version-name "my_version" \
--engine WDL \
--definition-zip fileb://workflow-crambam.zip \
```

```
--description "my version description" \
--main file://workflow-params.json \
--parameter-template file://workflow-params.json \
--storage-type='STATIC'   \
--storage-capacity 1200   \
--tags example123=string  \
--accelerators GPU
```

Wenn sich Ihre Workflow-Definitionsdatei in einem Amazon S3-Ordner befindet, geben Sie den Speicherort mithilfe der `definition-uri` Parameter statt `definition-zip`. Weitere Informationen finden Sie unter CreateWorkflowVersion in der AWS HealthOmics API-Referenz.

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

```
{
  "workflowId": "1234567",
  "versionName": "my_version",
  "arn": "arn:aws:omics:us-west-2:123456789012:workflow/1234567/version/3",
  "status": "ACTIVE",
  "tags": {
    "environment": "production",
    "owner": "team-alpha"
  },
  "uuid": "0ac9a563-355c-fc7a-1b47-a115167af8a2"
}
```

Erstellen Sie eine Workflow-Version mithilfe eines SDK

Sie können einen Workflow mit einem der SDKs erstellen.

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

```
import boto3

omics = boto3.client('omics')

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

response = omics.create_workflow_version(
   workflowId='1234567',
   versionName='my_version',
```

```
    requestId='my_request_1'
    definitionZip=definition,
    parameterTemplate={ ... }
)
```

Überprüfen Sie den Status einer Workflow-Version

Nachdem Sie Ihre Workflow-Version erstellt haben, können Sie den Status überprüfen und weitere Details des Workflows mithilfe von get-workflow-version, wie gezeigt.

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

Die Antwort enthält Ihre Workflow-Details, einschließlich des Status, wie abgebildet.

```
{
  "workflowId": "1234567",
  "versionName": "3.0.0",
  "arn": "arn:aws:omics:us-west-2:123456789012:workflow/1234567/version/3.0.0",
  "status": "ACTIVE",
  "description": ...
  "uuid": "0ac9a563-355c-fc7a-1b47-a115167af8a2"
}
```

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

Eine Workflow-Version aktualisieren

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

Themen

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

Aktualisieren Sie eine Workflow-Version mithilfe der Konsole

Um eine Workflow-Version zu aktualisieren

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

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

Aktualisieren Sie eine Workflow-Version mit der CLI

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

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

```
--version-name "my_version"
--storage-type 'STATIC'
--storage-capacity 2400
--description "version description"
```

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

Aktualisieren Sie eine Workflow-Version mithilfe eines SDK

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

```
import boto3

omics = boto3.client('omics')

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

Löschen Sie eine Workflow-Version

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

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

Themen

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

Löschen Sie eine Workflow-Version mithilfe der Konsole

Um eine Workflow-Version zu löschen

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

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

Löschen Sie eine Workflow-Version mit der CLI

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

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

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

Löschen Sie eine Workflow-Version mithilfe eines SDK

Sie können einen Workflow mit einem der löschen SDKs.

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

```
import boto3

omics = boto3.client('omics')

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

```
)
```

Verwenden HealthOmics läuft

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

Wenn Sie einen Lauf starten, HealthOmics weist temporären Laufspeicher zu, den die Workflow-Engine während der Ausführung verwenden kann. Um Datenisolierung und Sicherheit zu gewährleisten, HealthOmics stellt den Speicher zu Beginn jedes Laufs bereit und stellt ihn am Ende des Laufs wieder frei.

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

Wenn Sie einen Lauf starten, HealthOmics weist dem Lauf eine Run-ID und eine Run-UUID zu. Läufe in einem Konto und einer Region haben eindeutige Run-IDs. Allerdings HealthOmics verwendet gelöschte Lauf-IDs wieder, sodass ein Lauf und ein gelöschter Lauf dieselbe Run-ID haben können. Außerdem ist es selten, aber möglich, dass ein gemeinsam genutzter Workflow dieselbe Run-ID wie ein Lauf in Ihrem Konto hat.

Der run uuid ist eine GUID (Globally Unique Identifier), die Sie verwenden können, um Runs kontoübergreifend zu identifizieren oder um zwischen zwei Läufen in Ihrem Konto zu unterscheiden, die dieselbe Run-ID haben.

> **Note**
>
> Aus Gründen der Datenherkunft empfehlen wir Ihnen, den run uuid um Läufe eindeutig zu identifizieren. Die run uuid ist auch die beste Kennung, die Sie mit Ihrem internen Laborinformationsmanagementsystem (LIMS) oder Ihrem Probenverfolgungssystem verknüpfen können.

Sie können verwenden Kiro CLI um deine Läufe zu optimieren und die Laufleistung zu analysieren. Weitere Informationen finden Sie unter Beispiel für Eingabeaufforderungen für Kiro CLI und der HealthOmics Tutorial zur generativen KI von Agentic auf GitHub.

Themen

- Speichertypen in HealthOmics Workflows ausführen
- Kurzlebiger Speicher für Workflow-Aufgaben HealthOmics
- Retentionsmodus für HealthOmics Läufe ausführen
- HealthOmics Eingaben ausführen
- Lebenszyklus in einem HealthOmics Workflow ausführen
- HealthOmics Ausgaben ausführen
- Gründe für Ausführungsfehler
- Aufgabenlebenszyklus in einem HealthOmics Lauf
- Debuggen eines HealthOmics Workflow-Ausführung
- Führen Sie die Optimierung für ein privates aus HealthOmics Arbeitsablauf
- Operationen ausführen in HealthOmics

Speichertypen in HealthOmics Workflows ausführen

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

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

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

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

Dynamischer Run-Speicher

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

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

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

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

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

Codebeispiele für den sicheren Umgang mit nicht-idempotenten Vorgängen

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

```
import os
import errno

def remove_file(file_path):
    try:
        os.remove(file_path)
    except OSError as e:
        # If the error is "No such file or directory", ignore it (or log it)
        if e.errno != errno.ENOENT:
            # Otherwise, raise the error
```

```
            raise

# Example usage
remove_file("myfile")
```

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

```
rm -f my_file
```

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

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

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

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

Statisch ausgeführter Speicher

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

HealthOmics weist eine feste Menge an Run-Speicher zu. Sie geben diesen Wert an, wenn Sie den Lauf starten. Der Standard-Run-Speicher ist 1200 GiB, wenn Sie keinen Wert angeben. Die unterstützten Speichergrößen sind 1.200 GiB oder ein Vielfaches von 2.400 GiB (z. B. 2.400, 4.800, 7.200 GiB). Wenn Ihre angeforderte Größe kein unterstützter Wert ist, rundet das System auf die nächstgelegene unterstützte Größe auf.

Wenn Sie beispielsweise 42.000 GiB anfordern, stellt der Lauf 43.200 GiB (18 × 2.400 GiB) bereit, da 42.000 kein Vielfaches von 2.400 sind. In ähnlicher Weise stellt eine Anfrage für 5.000 GiB 7.200 GiB (3 × 2.400 GiB) bereit, und eine Anfrage für 1.000 GiB stellt die Mindestgröße von 1.200 GiB bereit.

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

- Basisdurchsatz von 200 MB/s pro TiB bereitgestellter Speicherkapazität.

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

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

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

Berechnung des erforderlichen statischen Laufspeichers

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

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

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

Beispiel:

```
    Maximum storage measured from a dynamic run storage workflow run:  500GiB
      File system size: 1200GiB
      7% of the file system size:  84GiB
      500 + 84 = 584GiB of static run storage required for this run.
```

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

> **Note**
>
> Zusätzlich zum gemeinsam genutzten Run-Speicher HealthOmics bietet er kurzlebigen Speicher pro Aufgabe unter. `/tmp` Durch die Umleitung von Scratch I/O auf kurzlebigen Speicher kann die Auslastung des gemeinsam genutzten Run-Dateisystems reduziert

werden. Weitere Informationen finden Sie unter Kurzlebiger Speicher für Workflow-Aufgaben HealthOmics.

Kurzlebiger Speicher für Workflow-Aufgaben HealthOmics

HealthOmics bietet temporären Speicher für Workflow-Aufgaben, die das Verzeichnis verwenden. `/tmp` Dieser Speicher ist temporär und für jede Aufgabe in einem Workflow einzigartig. HealthOmics weist jeder Aufgabeninstanz standardmäßig 16 GiB an temporärem Speicher zu. Sie können die Menge des temporären Speichers, der einzelnen Aufgaben in Ihrer Workflow-Definition zugewiesen ist, auf bis zu 3.072 GiB pro Aufgabe erhöhen. Alle darin gespeicherten Daten werden im Ruhezustand `/tmp` verschlüsselt.

Themen

- Wichtigste Vorteile
- Wie funktioniert ephemerer Speicher
- Ephemeren Speicher aktivieren
- Standard-Speicherzuweisung
- Konfiguration der Größe des kurzlebigen Speichers
- Überwachung des kurzlebigen Speichers
- Wie wird kurzlebiger Speicher abgerechnet
- Überlegungen und Einschränkungen
- Problembehandlung bei kurzlebigem Speicher

Wichtigste Vorteile

- Schnellere Ausführung von Aufgaben: Kurzlebiger Speicher kann die Laufleistung verbessern, da das gemeinsam genutzte Dateisystem reduziert wird. I/O
- Vorhersagbare Leistung: Aufgaben konkurrieren nicht mit anderen gleichzeitig ausgeführten Aufgaben um die I/O Bandbreite, wodurch die Variabilität und Drosselung, die mit einem gemeinsam genutzten Netzwerkdateisystem einhergehen, reduziert werden.
- Niedrigere Kosten: Eine schnellere Ausführung von Aufgaben reduziert direkt die Rechenzeit und die Kosten für I/O gebundene Aufgaben, die diese nutzen. `/tmp` Bei Läufen, die statischen Laufspeicher verwenden, kann dies die Menge des bereitgestellten Laufspeichers reduzieren, den Sie benötigen. Dynamische Läufe erfordern keine Änderungen.

Wie funktioniert ephemerer Speicher

Wenn Sie den temporären Speicher aktivieren, wird für jede HealthOmics Workflow-Aufgabeninstanz ein dediziertes lokales Speicher-Volume bereitgestellt`/tmp`. Der temporäre Speicher ist für temporäre Dateien vorgesehen, die während der Ausführung der Aufgabe generiert werden. Temporäre Speichervolumes werden immer gelöscht, wenn die Aufgabe beendet wird. Daten, in die geschrieben `/tmp` wird, werden nicht gespeichert, exportiert oder für andere Aufgaben oder nachfolgende Ausführungen zugänglich.

Wo Workflows temporäre Dateien schreiben

Workflows profitieren von flüchtigem Speicher, wenn Aufgabenprozesse direkt darauf zugreifen. I/O `/tmp` Workflow-Sprachen der Bioinformatik verwenden das `/tmp` Verzeichnis (und die `$TMPDIR` Umgebungsvariablen) für temporäre`$TMP`, kurzlebige Zwischendateien während der Aufgabenausführung. Stellen Sie sicher, dass Ihre Aufgabenbefehle keinem anderen Speicherort `$TMPDIR` zugeordnet sind.

Workflow-Aufgabenprozesse, die schreiben, um bei Aktivierung `/tmp` automatisch temporären Speicher zu verwenden, sodass keine Workflow-Änderungen erforderlich sind. Workflows, die Scratch-Prozesse nicht explizit dazu anweisen, `/tmp` schreiben Scratch-Daten in das Arbeitsverzeichnis auf dem gemeinsam genutzten Dateisystem, das für den Run-Speicher verwendet wird, obwohl die verwendeten Tools möglicherweise den temporären `/tmp` Speicher nutzen.

Verschlüsselung im Ruhezustand

Der gesamte temporäre Speicher wird im Ruhezustand mithilfe eines vom Service verwalteten Schlüssels verschlüsselt. AWS KMS Instances mit beschleunigter Datenverarbeitung sind hardwareverschlüsselt. Für jedes Volume gibt es einen eindeutigen Schlüssel, der beim Beenden der Instanz gelöscht wird.

Berechtigungen

HealthOmics verwaltet in Ihrem Namen die Anlage und den Lebenszyklus flüchtiger Speichervolumes. Für Ihre Rolle zur Aufgabenausführung sind keine zusätzlichen IAM-Berechtigungen erforderlich.

Ephemeren Speicher aktivieren

Sie können Scratch in I/O den lokalen Speicher umleiten, indem Sie dies `scratchStorageMode` in der `StartRun` API festlegen. Die `scratchStorageMode` Einstellung gilt nur für CPU-Instanzen und gilt für alle Aufgaben in diesem Lauf.

`scratchStorageMode`bestimmt, wo Ihr Workflow Scratch-Daten schreibt. Mögliche Werte:

- `LOCAL`— Kurzlebiger Speicher wird auf der lokalen Festplatte gespeichert. Scratch I/O hat dedizierte IOPS und einen dedizierten Durchsatz.
- `SHARED`— Das gemeinsam genutzte Dateisystem wird verwendet (Standard). Scratch I/O konkurriert mit dem Arbeitsverzeichnis.

Weitere Informationen finden Sie unter Starte einen Lauf in HealthOmics.

> **Note**
>
> GPU-Aufgaben verwenden immer lokalen temporären NVMe-Speicher für Scratch-Daten, und das `scratchStorageMode` ist immer `LOCAL` für GPU-Aufgaben vorgesehen.

Entscheiden Sie sich für kurzlebigen Speicher

Um den temporären Speicher für einen Lauf zu aktivieren, setze `LOCAL` beim Start des Laufs `scratchStorageMode` die Einstellung auf.

```
aws omics start-run \
    --workflow-id workflow-id \
    --role-arn arn:aws:iam::123456789012:role/OmicsServiceRole \
    --output-uri s3://amzn-s3-demo-bucket/output-folder/ \
    --parameters file:///path/to/parameters.json \
    --scratch-storage-mode LOCAL
```

Bei Batch-Läufen geben `scratchStorageMode` Sie den Wert ein. `defaultRunSetting` Die Einstellung gilt für jeden Lauf im Batch.

```
aws omics start-run-batch \
    --batch-name "my-batch" \
    --default-run-setting '{
      "workflowId": "workflow-id",
      "roleArn": "arn:aws:iam::123456789012:role/OmicsServiceRole",
      "outputUri": "s3://amzn-s3-demo-bucket/output-folder/",
      "storageType": "DYNAMIC",
      "parameters": {"referenceUri": "s3://amzn-s3-demo-bucket/reference.fasta"},
      "scratchStorageMode": "LOCAL"
    }' \
```

```
    --batch-run-settings '{
      "inlineSettings": [
        {
          "runSettingId": "sample-A",
          "parameters": {"inputUri": "s3://amzn-s3-demo-bucket/sampleA.fastq"}
        },
        {
          "runSettingId": "sample-B",
          "parameters": {"inputUri": "s3://amzn-s3-demo-bucket/sampleB.fastq"}
        }
      ]
    }'
```

Deaktivieren Sie den kurzlebigen Speicher

Um den kurzlebigen Speicher für einen bestimmten Lauf zu deaktivieren (z. B. um einen Fehler zu isolieren), setzen Sie den Wert auf. `scratchStorageMode` `SHARED`

```
aws omics start-run \
     --workflow-id workflow-id \
     --role-arn arn:aws:iam::123456789012:role/OmicsServiceRole \
     --output-uri s3://amzn-s3-demo-bucket/output-folder/ \
     --scratch-storage-mode SHARED
```

`scratchStorageMode`Ist dies der `SHARED` Fall, werden alle `disk` und gleichwertige Anweisungen in der Workflow-Definition ignoriert und `/tmp` vom gemeinsam genutzten Dateisystem unterstützt. Diese Einstellung gilt nur für CPU-Instanzen. GPU-Aufgaben verwenden immer lokalen temporären NVMe-Speicher und können nicht deaktiviert werden.

Überprüfen Sie den effektiven Modus

Der kurzlebige Speicher ist standardmäßig deaktiviert. Wenn in einer `StartRun` Anfrage weggelassen `scratchStorageMode` wird, `scratchStorageMode` ist auf `SHARED` (Standard) gesetzt.

`scratchStorageMode`wird in der `GetRun` Antwort nur zurückgegeben, wenn sie in der `StartRun` Anfrage explizit übergeben wurde. `SHARED`Ist der Standardwert, wenn er weggelassen wird. Rufen Sie `GetRun` auf, um den effektiven Speichermodus für einen Lauf zu bestätigen.

```
aws omics get-run --id run-id
```

Standard-Speicherzuweisung

Die standardmäßige temporäre Speicherzuweisung beträgt 16 GiB pro Aufgabe für alle Standard-, Compute- und Memory-optimized Instance-Typen. Sie müssen keine `disk` Direktive angeben, um diese Standardeinstellung zu erhalten. Verwenden Sie die `disk` Direktive, um zusätzlichen Speicher zu konfigurieren. Sie können den Umfang des temporären Speichers, der einzelnen Aufgaben in Ihrer Workflow-Definition zugewiesen ist, auf bis zu 3.072 GiB pro Aufgabe erhöhen. Speicherplatz, der über den Standardwerten von 16 GiB liegt, wird Ihnen in Rechnung gestellt.

Kurzlebiger Speicher kann in Schritten von 16 GiB konfiguriert werden. Weitere Informationen finden Sie unter Unterstützte Größen.

Ephemerer Speicher für beschleunigte Recheninstanzen

Die Speicherkapazität der GPU-Instance ist pro Instance-Typ festgelegt und wird ohne zusätzliche Kosten bereitgestellt.

GPU-Aufgaben verwenden immer lokalen ephemeren NVMe-Speicher. Die `scratchStorageMode` Einstellung auf gilt `StartRun` nicht für GPU-Aufgaben. Wenn Sie diesen Wert auf setzen, hat `SHARED` dies keine Auswirkungen auf GPU-Instances. Die Kapazität ist pro Instanztyp fest und kann nicht mithilfe der `disk` Direktive angepasst werden. Die NVMe-Kapazität wird durch den Instance-Typ vorgegeben`cpu`, `memory` der für die Anforderungen der Aufgabe ausgewählt wurde. `acceleratorType`

Größe	GPUs	vCPU	Arbeitsspeicher (GiB)	G4dn NVMe (T4)	G5 NVMe (A10 G)	G6 NVMe (L4)	G6e NVMe (L40S)
xlarge	1	4	16	125 GiB	250 GiB	250 GiB	250 GiB
2xlarge	1	8	32	225 GiB	450 GiB	450 GiB	450 GiB
4xlarge	1	16	64	225 GiB	600 GiB	600 GiB	600 GiB
8xlarge	1	32	128	900 GiB	900 GiB	900 GiB	900 GiB
12xlarge	4	48	192	900 GiB	3.800 GiB	3.760 GiB	3.800 GiB

Größe	GPUs	vCPU	Arbeitsspeicher (GiB)	G4dn NVMe (T4)	G5 NVMe (A10 G)	G6 NVMe (L4)	G6e NVMe (L40S)
16xlarge	1	64	256	900 GiB	1.900 GiB	1.880 GiB	1.900 GiB
24xlarge	4	96	384	—	3.800 GiB	3.760 GiB	3.800 GiB

Konfiguration der Größe des kurzlebigen Speichers

Wenn auf gesetzt `scratchStorageMode` ist`LOCAL`, können Sie mithilfe der `disk` Anweisung (oder einer gleichwertigen Anweisung) in Ihrer Workflow-Definition eine Erhöhung des temporären Speichers pro Aufgabe anfordern. HealthOmics behandelt die `disk` Direktive als Hinweis und gibt ein auf die nächsten 16 GiB gerundetes Volumen an. Die Verwendung der `disk` Direktive hat keinen Einfluss auf die Auswahl des Instanztyps. Der Instanztyp wird ausschließlich auf der Grundlage von `cpu`memory, und ausgewähl`acceleratorType`. Weitere Informationen finden Sie unter Aufgabenressourcen in einem HealthOmics Workflow-Definition.

Wenn keine `disk` Direktive vorhanden ist, erhält die Aufgabe die Standardgröße von 16 GiB. Aufgaben können nicht weniger als den standardmäßigen temporären Speicher haben.

Sie müssen die Größe Ihres temporären Speichers nicht für abgerufene Container-Images anpassen, die separat abgerechnet werden. Weitere Informationen finden Sie unter Container-Images für private Workflows.

Wann sollte die Disk-Direktive verwendet werden

Verwenden Sie die `disk` Direktive in Ihrer Aufgabendefinition, wenn der standardmäßige temporäre Speicher für den von Ihnen ausgewählten Instance-Typ für Ihre Aufgabenanforderungen nicht ausreicht. Zum Beispiel, wenn eine Aufgabe große Datenmengen in sie schreibt. `/tmp`

Häufige Anwendungsfälle für flüchtiger Speicher

1. RNA-Seq Fusionserkennung: RNA-seq Workflows generieren große Zwischen-BAMs, und Aufgabenprozesse erfordern oft, dass sowohl die FastQS-Rohdaten als auch die abgestimmten Ausgaben gleichzeitig vorhanden sind, was große Scratch-Disketten erfordert (z. B. 512 GiB pro Aufgabe).

2. De Novo Genome Long-read Assembly: Assembly-Workflows benötigen große Scratch-Volumina, um unformatierte Lesevorgänge und temporäre Assembly-Artefakte zu verarbeiten, die vor der Ausgabe wiederholt neu geschrieben und neu organisiert werden. Diese Aufgaben sind speicher- und festplattenintensiv und erfordern manchmal mehrere TIB an temporärem Speicher.
3. Variantenaufruf/BAM-Verarbeitung: Workflows zum Aufrufen von Varianten benötigen viel Speicherplatz für Ausrichtungs- und Sortierschritte, bei denen wiederholt große BAM- oder CRAM-Dateien gelesen und neu geschrieben werden. Der temporäre Speicherbedarf liegt in der Regel bei Hunderten von GiB.

Syntax der Direktive nach Engine

Die folgende Tabelle zeigt die entsprechende Direktive für jede Workflow-Sprache.

Engine	Direktive	Beispiel
WDL 1.1	`disks`	`disks: "/tmp 700 GiB"`
Nächster Fluss	`disk`	`disk '700 GB'`
CWL	`tmpdirMin`	`tmpdirMin: 716800` (Wert in MiB)

Die folgenden Beispiele zeigen, wie eine Aufgabe konfiguriert wird, die 700 GiB kurzlebigen Speicher anfordert. HealthOmics rundet dies auf die 704-GiB-Stufe auf.

WDL

```
task sort_bam {
    runtime {
        cpu:  16
        disks: "700 GiB"
    }
    command <<<
        samtools sort -T /tmp/sort_buffer ~{input_bam} -o ~{output_bam}
    >>>
}
```

Nextflow

```
process sort_bam {
    disk '700 GB'
    script:
    """
    samtools sort -T /tmp/sort_buffer ${input} -o ${output}
    """
}
```

CWL

```
requirements:
  ResourceRequirement:
    tmpdirMin: 716800   # 700 GiB expressed in MiB
```

Weitere Hinweise zur unterstützten Direktivensyntax finden Sie unter Besonderheiten der WDL-Workflow-Definition undBesonderheiten der Nextflow-Workflow-Definition.

Beispiel: Ausdrucksbasierte Festplattengröße

Statt einer festen Größe können Sie die `disk` Direktive auf einen Ausdruck setzen, den die Workflow-Engine zur Laufzeit für jede Aufgabe auswertet. Der folgende Nextflow-Prozess verwendet einen Abschluss, der die Anforderung anhand der Anzahl der Aufgabenversuche skaliert. Wenn die Aufgabe aus irgendeinem Grund fehlschlägt und Nextflow sie erneut versucht, wird bei der Wiederholung ein größeres Volumen angefordert.

```
process sort_bam {
    disk { 200.GB * task.attempt }
    errorStrategy 'retry'
    maxRetries 2

    script:
    """
    samtools sort -T /tmp/sort_buffer ${input} -o ${output}
    """
}
// First attempt requests 200 GiB (provisioned at the 208 GiB tier)
// Second attempt requests 400 GiB (provisioned at the 400 GiB tier)
```

Die Workflow-Engine wertet den Ausdruck für jeden Aufgabenversuch aus. HealthOmics rundet dann die aufgelöste Größe auf das nächste 16-GiB-Inkrement auf.

> **Note**
>
> Da die Engine den Ausdruck zur Laufzeit auflöst, HealthOmics kann die Größe bei nicht überprüft werden. `CreateWorkflow` HealthOmics begrenzt eine berechnete Größe auf über 3.072 GiB, wenn die Aufgabe gestartet wird. Weitere Informationen finden Sie unter Unterstützte Größen.

Allgemeine Bioinformatik-Tools und kurzlebige Speicherung

Viele Bioinformatik-Tools schreiben während der Ausführung große temporäre Dateien. Wenn auf gesetzt `scratchStorageMode` ist`LOCAL`, leiten Sie diese Tools zur Verwendung um, `/tmp` sodass Scratch auf I/O das schnelle lokale Volume und nicht auf das gemeinsam genutzte Run-Dateisystem übertragen wird. Die folgenden Beispiele zeigen die relevanten Flags für häufig verwendete Tools.

WDL

```
task sort_bam {
    runtime {
        cpu:   16
        disks: "700 GiB"
    }
    command <<<
        # samtools: -T sets the temp-file prefix
        samtools sort -T /tmp/sort_buffer ~{input_bam} -o ~{sorted_bam}

        # GATK / Picard: --TMP_DIR flag (older Picard uses TMP_DIR=/tmp)
        gatk MarkDuplicates -I ~{sorted_bam} -O ~{output_bam} --TMP_DIR /tmp

        # STAR: --outTmpDir (path must not pre-exist; STAR creates it)
        STAR --runThreadN 16 --readFilesIn ~{reads} --outTmpDir /tmp/star_tmp --
outFileNamePrefix out_

        # bcftools sort: -T / --temp-dir
        bcftools sort -T /tmp ~{vcf} -o ~{output_vcf}

        # GNU sort: -T / --temporary-directory
        sort -T /tmp ~{big_table} -o ~{sorted_table}
    >>>
```

```
}
# HealthOmics rounds 700 GiB up to the 704 GiB tier
```

Nextflow

```
process sort_bam {
    disk '700 GB'
    script:
    """
    # samtools: -T sets the temp-file prefix
    samtools sort -T /tmp/sort_buffer input.bam -o sorted.bam

    # GATK / Picard: --TMP_DIR flag (older Picard uses TMP_DIR=/tmp)
    gatk MarkDuplicates -I sorted.bam -O dedup.bam --TMP_DIR /tmp

    # STAR: --outTmpDir (path must not pre-exist; STAR creates it)
    STAR --runThreadN 16 --readFilesIn reads.fastq --outTmpDir /tmp/star_tmp --
outFileNamePrefix out_

    # bcftools sort: -T / --temp-dir
    bcftools sort -T /tmp input.vcf -o sorted.vcf

    # GNU sort: -T / --temporary-directory
    sort -T /tmp big_table.tsv -o sorted_table.tsv
    """
}
// HealthOmics rounds 700 GB up to the 704 GiB tier
```

CWL

```
class: CommandLineTool
cwlVersion: v1.2
requirements:
  ResourceRequirement:
    coresMin: 16
    tmpdirMin: 716800          # 700 GiB expressed in MiB
baseCommand: [bash, -c]
arguments:
  - |
    set -euo pipefail

    # samtools: -T sets the temp-file prefix
    samtools sort -T /tmp/sort_buffer input.bam -o sorted.bam
```

```
    # GATK / Picard: --TMP_DIR flag (older Picard uses TMP_DIR=/tmp)
    gatk MarkDuplicates -I sorted.bam -O dedup.bam --TMP_DIR /tmp

    # STAR: --outTmpDir (path must not pre-exist; STAR creates it)
    STAR --runThreadN 16 --readFilesIn reads.fastq --outTmpDir /tmp/star_tmp --
outFileNamePrefix out_

    # bcftools sort: -T / --temp-dir
    bcftools sort -T /tmp input.vcf -o sorted.vcf

    # GNU sort: -T / --temporary-directory
    sort -T /tmp big_table.tsv -o sorted_table.tsv
```

Unterstützte Größen

Die angeforderten Größen werden auf das nächste 16-GiB-Inkrement aufgerundet, ausgehend von der Standardeinstellung von 16 GiB (16, 32, 48, 64,... bis zu 3.072 GiB). Die maximal unterstützte Größe beträgt 3.072 GiB pro Aufgabe.

Wenn eine angeforderte Größe 3.072 GiB überschreitet, HealthOmics werden 3.072 GiB bereitgestellt und eine Warnung in das Ausführungsprotokoll geschrieben. Die Aufgabe ist nicht automatisch fehlgeschlagen.

> **Note**
>
> Bei ausdrucksbasierten `disk` Direktiven — wie Nextflow-Closures oder WDL-Ausdrücken wie `disks: ceil(size(input_bam, "GiB") * 2.5)` — wird der Wert zur Laufzeit ausgewertet, nicht bei. `CreateWorkflow` Wenn die berechnete Größe 3.072 GiB überschreitet, schlägt die Aufgabe zur Laufzeit fehl und alle bis zu diesem Zeitpunkt angefallenen Rechenkosten werden in Rechnung gestellt. Ein Beispiel finden Sie unter Beispiel: Ausdrucksbasierte Festplattengröße.

Unterstützte WDL-Festplattenformulare

Eine vollständige Liste der akzeptierten `disks` WDL-Formulare finden Sie unter. Unterstützte WDL-Festplattenformulare

Verwenden der Nextflow-Scratch-Direktive

Für Nextflow-Workflows können Sie die `scratch` Direktive verwenden, um zu steuern, wo Prozesse temporäre Arbeitsdateien schreiben. Hinweise zu den unterstützten Werten und der empfohlenen Verwendung von temporärem Speicher finden Sie unter. Effiziente Nutzung von Scratch Storage in Nextflow

Überwachung des kurzlebigen Speichers

HealthOmics schreibt temporäre Speichermetriken pro Aufgabe in Manifestprotokolle. CloudWatch Zu den Metriken gehören temporäre Speicherdaten pro Aufgabe in Bezug auf die Größe (n) des bereitgestellten Volumes und dessen Nutzung (en`scratchStorageReservedGiB`) für jede Aufgabe. `scratchStorageUtilizedGiB` Prüfen Sie das Manifestprotokoll, um festzustellen, ob die Aufgaben zu viel oder zu wenig bereitgestellt wurden, ohne eine direkte Abfrage durchzuführen. CloudWatch Einzelheiten zu Manifestprotokollen finden Sie unter. Überwachung HealthOmics mit CloudWatch Logs

Wie wird kurzlebiger Speicher abgerechnet

Ihnen wird nur der temporäre Speicher in Rechnung gestellt, der über der Standardzuweisung bereitgestellt wird. Anfragen, die über der Standardeinstellung in Ihrer `disk` Anweisung liegen, werden auf die nächste unterstützte Stufe aufgerundet.

Bei GPU-Instances ist kurzlebiger Speicher bereits in der Preisgestaltung für Instances berücksichtigt. Für kurzlebigen Speicher bei GPU-Aufgaben fallen keine zusätzlichen Gebühren an.

Überlegungen und Einschränkungen

Überlegungen	Detail
Kurzlebiger Speicher ist nicht persistent	Ephemere Speichervolumes werden immer gelöscht, wenn die Aufgabe beendet wird. Die `/tmp` darin enthaltenen Daten werden nicht gespeichert, exportiert und sind nicht für nachfolgende Aufgaben oder Ausführungen verfügbar. Daten in `/tmp` einem temporäre n Speicher können keine Aufgaben- oder Workflow-Ausgabe sein. Dies führt zur Laufzeit zu einem Ausfall.

Überlegungen	Detail
Der temporäre Speicher wird nicht von mehreren Aufgaben gemeinsam genutzt	Jede Aufgabe erhält ihr eigenes isolierte s temporäres Speichervolumen. Aufgaben können nicht auf die Verzeichnisse der jeweils anderen zugreifen`/tmp`. Daten, die von Aufgaben gemeinsam genutzt werden müssen, müssen in das gemeinsam genutzte Run-Datei system geschrieben werden.
Die Größe des Speichers kann während einer Aufgabe nicht geändert werden	Die Speichergröße ist beim Start der Aufgabe festgelegt. Sie können den zugewiesenen Speicher nicht erhöhen oder verringern, während eine Aufgabe ausgeführt wird.
Working-directory scratch wird nicht automatis ch umgeleitet	Workflows, die Scratch in das Arbeitsve rzeichnis schreiben — zum Beispiel`input/`,`./`, oder `out/` — profitieren nicht automatisch davon. Aktualisieren Sie Ihren Workflow, um Scratch I/O nach `/tmp` oder umzuleite n`$TMPDIR`.
In Scratch-Daten wird nicht **`/tmp`** wie erwartet geschrieben	Stellen Sie sicher, dass Ihre Task-Prozesse explizit in Ihre Task-Prozesse schreiben `/tmp` und dass Ihre Task-Befehle keinem anderen Speicherort `$TMPDIR` zugeordnet wurden.
GPU-Instances verwenden immer kurzlebigen Speicher	GPU-Aufgaben werden immer `/tmp` im lokalen NVMe-Instance-Speicher bereitgestellt. Durch `scratchStorageMode` die Einstellung auf `SHARED` wird der kurzlebige Speicher für GPU-Aufgaben nicht deaktiviert.
GPU-Instanzen: Die NVMe-Kapazität ist fest	Benutzerdefinierte Festplattengrößen werden auf GPU-Instanzen nicht unterstützt. HealthOmi cs ignoriert `disk` Anweisungen und stellt die Standard-NVMe-Kapazität für den Instanztyp bereit.

Überlegungen	Detail
Maximal 3.072 GiB pro Aufgabe (CPU)	Anfragen, die 3.072 GiB überschreiten, werden bei 3.072 GiB mit einer Run-Log-Warnung bereitgestellt. Aufgaben sind nicht fehlgesch lagen.
Nur unterstützte Stufen (CPU)	Die angeforderten Größen werden auf das nächste 16-GiB-Inkrement aufgerundet (16, 32, 48, 64,... bis zu 3.072 GiB).
Expression-based Direktiven, die zur Laufzeit ausgewertet werden	`disk`Aus Ausdrücken berechnete Werte werden beim Start der Aufgabe validiert, nicht bei`CreateWorkflow` . Bis zu diesem Zeitpunkt berechnete Kosten werden berechnet , wenn die Aufgabe zur Laufzeit fehlschlägt.
`SHARED`Der Modus ignoriert alle Festplatt endirektiven (nur CPU)	Wenn `scratchStorageMode` ist`SHARED`, `disktmpdirMin` , und äquivalente Anweisung en werden für CPU-Aufgaben ignoriert. Es wird kein lokales Speichervolume bereitgestellt. GPU-Aufgaben sind davon nicht betroffen, sie verwenden immer lokales NVMe.

Problembehandlung bei kurzlebigem Speicher

Die Aufgabe schlägt fehl, da der kurzlebige Speicher erschöpft ist

Eine Aufgabe schlägt fehl, wenn der kurzlebige Speicher seine Kapazität erreicht. Prüfen Sie Ihre CloudWatch Manifestprotokolle, um festzustellen, wie viel Speicherplatz Ihre Aufgabe tatsächlich verbraucht hat, und fügen Sie dann die `disk` Direktive hinzu oder erhöhen Sie sie, um eine größere Ebene anzufordern.

```
# Before: 4-vCPU task using the 16 GiB default
runtime { cpu: 4 }

# After: explicitly request 400 GiB
runtime { cpu: 4, disks: "400 GiB" }
```

Kurzlebiger Speicher scheint nicht genutzt zu werden

Rufen Sie an `GetRun` und überprüfen Sie das Feld`scratchStorageMode`. Wenn der Wert 0 ist`SHARED`, ist der kurzlebige Speicher für diesen Lauf nicht aktiviert. Stellen Sie es `--scratch-storage-mode LOCAL` bei Ihrem nächsten `start-run` Anruf ein.

```
aws omics get-run --id run-id
```

Retentionsmodus für HealthOmics Läufe ausführen

HealthOmics Archiviert nach Abschluss eines Laufs die Metadaten des Laufs unter CloudWatch. Speichert die Laufdaten standardmäßig auf unbestimmte Zeit, sofern Sie die CloudWatch Aufbewahrungsrichtlinie nicht ändern. CloudWatch Run-Ausgaben werden ebenfalls in Amazon S3 gespeichert, bis Sie sie löschen.

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

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

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

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

Zusätzliche Überlegungen zur Verwendung des REMOVE-Aufbewahrungsmodus:

- Wenn Sie REMOVE zum ersten Mal als Aufbewahrungsmodus verwenden, sollten Sie erwägen, einen oder mehrere Läufe zu löschen, die den RETIN-Modus verwenden, um Steckplätze freizugeben. Wenn Sie weitere REMOVE-Läufe starten, wird das automatische Entfernen übernommen, sodass genügend Steckplätze für neue Läufe verfügbar sind.
- Wenn Sie einen archivierten Lauf (oder eine Reihe von Läufen) erneut ausführen möchten, verwenden Sie das HealthOmics Rerun-CLI-Tool. Weitere Informationen und Beispiele zur Verwendung dieses Tools finden Sie unter Omics rerun im Tools-Repository. HealthOmics GitHub
- Wir empfehlen, für jeden Lauf einen eindeutigen Namen zu konfigurieren. Nach dem HealthOmics Entfernen eines Laufs können Sie die Konsole oder API nicht mehr verwenden, um den Namen oder die Lauf-ID des Laufs zu finden. Sie können sie jedoch verwenden, CloudWatch um nach dem Namen der Ausführung zu suchen. Verwenden Sie daher eindeutige Namen, um die besten Suchergebnisse zu erzielen.
- Sie können den CloudWatch start-query Befehl verwenden, um Informationen zu einem archivierten Lauf abzurufen. Wenn der Runname nicht eindeutig ist, gibt die Abfrage möglicherweise mehrere Manifeste zurück. Die Parameter Startzeit und Endzeit definieren den Zeitraum für die Suche.

```
aws logs start-query  \
    --log-group-name "/aws/omics/WorkflowLog" \
    --query-string 'filter @logStream like "manifest" and @message like "myRunName"'
 \
    --end-time <END-EPOCH-TIME> --start-time <START-EPOCH-TIME>
```

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

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

HealthOmics Eingaben ausführen

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

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

Themen

- Verwaltung der Größe von Ausführungsparametern
- Formate für Amazon S3-Eingabeparameter
- Status des Amazon S3-Eingabearchivs

Verwaltung der Größe von Ausführungsparametern

Wenn Sie einen Lauf starten, geben Sie Laufeingaben im JSON-Objekt oder in der JSON-Datei für die Ausführungsparameter an. Sie können bis zu 256 KB an Ausführungsparametern für den Workflow angeben. Sie können die folgenden Techniken verwenden, um diese Größenbeschränkung einzuhalten:

- Verwenden Sie Verzeichnisimporte

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

  Ein Musterblatt ist eine CSV- oder TSV-Datei mit einer Spalte für die Adresse fastq.gz (oder zwei für gepaartes Lesen) und zusätzlichen Spalten für Metadaten wie Beispielnamen. Sie geben das Musterblatt als Recheneingabeparameter an und nicht als Parameter für jede Eingabedatei.

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

Formate für Amazon S3-Eingabeparameter

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

- Bequemlichkeit — Sie geben den Verzeichnisnamen als Parameter an. Sie listen nicht jeden Dateinamen auf.
- Kompaktheit — Die maximale Dateigröße des Eingabeparameters beträgt 256 KB. Wenn Sie eine lange Liste von Eingabedateinamen angeben, können Sie dieses Maximum überschreiten.

Amazon S3 ist ein flaches Objektspeichersystem und unterstützt daher keine Verzeichnisse. Sie gruppieren Dateien in einem „Verzeichnis“, indem Sie jeder Datei das gleiche Objektschlüsselpräfix geben. Weitere Informationen zu Amazon S3-Objektschlüsselpräfixen finden Sie unter Objekte mithilfe von Präfixen organisieren.

HealthOmics interpretiert den Eingabeparameterwert wie folgt:

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

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

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

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

Nextflow: Behandlung des Glob-Musters in Amazon S3-Eingaben

Glob-Muster	HealthOmics Verhalten abgleichen	Hinweise
s3://bucket/directory/*.txt	Entspricht allen `.txt` Objekten in beliebiger Tiefe	

Glob-Muster	HealthOmics Verhalten abgleichen	Hinweise
	unter dem Präfix s3:///. bucket/directory Entspricht beispielsweise s3://bucket/directory/abc.txt oder s3://bucket/directory/subDir/123.txt usw.	
s3://bucket/directory/**/*.txt	Entspricht allen `.txt` Objekten in beliebiger Tiefe unter dem Präfix s3:///. bucket/directory Entspricht beispielsweise s3://bucket/directory/abc.txt oder s3://bucket/directory/subDir/123.txt usw.	In S3 `**` entspricht*.
s3://bucket/directory/{a, b} .txt	s3://bucket/directory/a.txt, s3:///b.txt bucket/directory	
s3://bucket/directory/?. txt	Findet Objekte im Präfixstamm, deren Dateiname aus einem einzelnen Zeichen besteht, gefolgt von `.txt`. Es entspricht beispielsweise s3://bucket/directory/a.txt, aber nicht s3://bucket/directory/someDir/a.txt oder s3://bucket/directory//a.txt someDir/subDir	
s3://bucket/directory/[0-9] .txt	s3://bucket/directory/0.txt, s3://bucket/directory/1.txt,... , s3://bucket/directory/9.txt	
s3://bucket/directory/[0-9] .txt	s3://bucket/directory/1.txt, s3://bucket/directory/2.txt, s3:///3.txt bucket/directory	

Glob-Muster	HealthOmics Verhalten abgleichen	Hinweise
s3://bucket/directory/[0-9] .txt	s3://bucket/directory/b.txt, s3://bucket/directory/c.txt,... , s3://bucket/directory/Y.txt	

Language-specific Behandlung von doppelten Schrägstrichen in Amazon S3-Eingaben

HealthOmics behält das native Engine-Verhalten für jede Workflow-Engine bei, wenn doppelte Schrägstriche in Amazon S3-URIs verarbeitet werden, sodass Sie keine Änderungen an Ihren Workflows vornehmen müssen, wenn Sie sie zu migrieren. HealthOmics In den folgenden Abschnitten wird beschrieben, wie jede Engine verschiedene Szenarien behandelt.

WDL

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

Eingabeparameter	Erwarteter Standort	
s3://myfiles/runs/inputs//file1.fastq	s3://myfiles/runs/inputs//file1.fastq	
s3://myfiles/runs/eingaben//	s3:///Eingaben// myfiles/runs	

Nächster Fluss

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

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

Eingabeparameter	Erwarteter Standort
runs/inputss3://myfiles////*.gz	s3://myfiles////*.gz runs/inputs
s3://myfiles////runs/inputs	s3://myfiles//runs/inputs/

CWL

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

Eingabeparameter	Erwarteter Standort
s3://myfiles// runs/inputs // file1.fastq	s3://myfiles// runs/inputs // file1.fastq
runs/inputss3://myfiles////	s3://myfiles//runs/inputs//

Status des Amazon S3-Eingabearchivs

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

- Flexible Retrieval- oder Deep Archive-Speicherklassen in Amazon S3 Glacier.
- Intelligentes Tiering bietet die Stufen Archived Access oder Deep Archive Access.

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

Lebenszyklus in einem HealthOmics Workflow ausführen

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

Sie können den Ausführungsstatus mit einer der folgenden Methoden abrufen:

- Die HealthOmics Konsole zeigt den Status jedes Laufs auf der Runs Seite an.

- Der GetRun API-Vorgang gibt den aktuellen Ausführungsstatus zurück.
- Sie können den Ausführungsstatus mithilfe von EventBridge Ereignissen überwachen. Weitere Informationen finden Sie unter Verwenden EventBridge mit AWS HealthOmics.

Themen

- Werte für den Ausführungsstatus
- Die Aufgabe wird erneut versucht
- Auswirkungen des Ausführungsstatus auf die Preisgestaltung

Werte für den Ausführungsstatus

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

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

HealthOmics unterstützt die folgenden Werte für den Ausführungsstatus:

Ausstehend

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

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

Wird gestartet

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

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

In Ausführung

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

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

Wird angehalten

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

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

Completed

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

- HealthOmics hat alle ausgeführten Aufgaben abgeschlossen und die Ausgabedaten fehlerfrei exportiert.
- Die Run-Ausgaben sind im angegebenen Amazon S3 S3-URI-Ausgabespeicherort verfügbar. HealthOmicsGeneriert für WDL und CWL eine Datei mit einer Zusammenfassung der Run-Ausgabe, die Informationen über die enthält. HealthOmics Ausgaben ausführen
- Die endgültigen Ausführungsmanifestprotokolle und Engine-Protokolle (falls zutreffend) sind unter verfügbar. CloudWatch
- Bei Läufen, die Aufgabenwiederholungen unterstützen, kann ein Lauf mit dem Status Abgeschlossen eine oder mehrere fehlgeschlagene Aufgaben enthalten. Solange eine

Aufgabenwiederholung für jede fehlgeschlagene Aufgabe erfolgreich war, wird die Ausführung in den HealthOmics Status Abgeschlossen überführt. HealthOmics weist jedem Wiederholungsversuch eine neue Aufgaben-ID zu, sodass die Ausführung die Aufgaben IDs für die fehlgeschlagenen Versuche und den abgeschlossenen Versuch umfasst.

Fehlgeschlagen

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

- Ein fehlgeschlagener Lauf wechselt in den Status Stopp, während die Ressourcen HealthOmics nicht bereitgestellt werden.

Abgebrochen

Ein Benutzer hat eine Anfrage zum Abbruch des Laufs initiiert.

- HealthOmics stoppt alle laufenden Aufgaben und hebt die Bereitstellung aller Ressourcen auf.
- HealthOmics exportiert keine Laufausgabedaten, wenn ein Benutzer einen Lauf abbricht. Sie haben keinen Zugriff auf Zwischendateien für einen abgebrochenen Lauf.
- Für Ihr Konto fallen Gebühren für die Aufgaben und Ressourcen an, die der Lauf während des Status Running vor dem Abbruch verbraucht hat.
- Es fallen keine Gebühren an, wenn Sie einen Lauf mit dem Status „Ausstehend“ oder „Wird gestartet“ stornieren.

Die Aufgabe wird erneut versucht

HealthOmics unterstützt Aufgabenwiederholungen für Aufgaben, die aufgrund von Dienstfehlern fehlschlagen (5XX-HTTP-Statuscodes).

Wenn alle Aufgaben in der Ausführung irgendwann abgeschlossen werden, auch wenn sie Wiederholungen erforderten, wird die Ausführung auf HealthOmics Abgeschlossen umgestellt. HealthOmics weist jedem Wiederholungsversuch eine neue Aufgaben-ID zu, sodass die Ausführung die Aufgaben IDs für die fehlgeschlagenen Versuche und den abgeschlossenen Versuch umfasst.

Das standardmäßige Wiederholungsverhalten hängt davon ab, welche Definitionssprache der Workflow verwendet. Die Standardeinstellung für Nextflow ist keine Wiederholungen. Bei WDL und CWL werden bis zu zwei Wiederholungen einer fehlgeschlagenen Aufgabe versucht, aber Sie können die Aufgabenwiederholung für bestimmte Aufgaben oder für alle Aufgaben in einem Workflow deaktivieren. HealthOmics Die Wiederholung von Aufgaben ist nützlich, um zeitweise

auftretende Servicefehler zu beheben. Sie könnten jedoch erwägen, eine Aufgabe zu deaktivieren, die idempotent ist.

Spezifische Informationen zu den einzelnen Workflow-Definitionssprachen finden Sie in den folgenden Themen:

- WDL — Konfigurieren Sie das Verhalten beim erneuten Versuch von Aufgaben in der Workflow-Definition. Siehe Verhalten beim erneuten Versuch von WDL-Aufgaben konfigurieren.
- Nextflow — Konfigurieren Sie das Verhalten beim erneuten Versuch von Aufgaben in der Nextflow-Konfigurationsdatei oder in der Workflow-Definition. Siehe Verhalten bei Wiederholungsversuchen von Nextflow-Aufgaben konfigurieren.
- CWL — Konfigurieren Sie das Verhalten beim erneuten Versuch von Aufgaben in der Workflow-Definition. Siehe Verhalten beim erneuten Versuch von CWL-Aufgaben konfigurieren.

Auswirkungen des Ausführungsstatus auf die Preisgestaltung

Für Ihr Konto können Gebühren anfallen, solange der Run-Status „Wird ausgeführt“ lautet. Während eines anderen Laufstatus fallen für Sie keine Gebühren an. Beispielsweise fallen keine Gebühren für Ressourcen an, wenn der Lauf gestartet oder gestoppt wird.

Ein Lauf mit dem Status Wird ausgeführt hat folgende Auswirkungen auf die Abrechnung:

- Für Ihr Konto fallen Gebühren für die Nutzung des Laufspeicher-Dateisystems an, solange der Ausführungsstatus „Wird ausgeführt“ lautet. Informationen zu den Laufspeichertypen finden Sie unter. Speichertypen in HealthOmics Workflows ausführen
- Für Ihr Konto fallen Gebühren für die Ausführung von Aufgaben an, die auf den Rechen- und Speicherressourcen basieren, die Sie für jede Aufgabe in der Workflow-Definition angegeben haben, und auf der Dauer der Aufgabe. Weitere Informationen finden Sie unter Rechen- und Speicheranforderungen für HealthOmics Aufgaben.
- Für jede Aufgabe gilt ein Mindestabrechnungsschwellenwert von einer Minute. Wenn Sie eine Aufgabe für weniger als eine Minute ausführen, fällt eine Gebühr für die Nutzung von mindestens einer Minute an. Wenn möglich, gruppieren Sie kleine Aufgaben, um die Kosten zu optimieren. Durch das Gruppieren von Aufgaben wird auch die Laufzeit reduziert, da vermieden wird, dass mehrere aufeinanderfolgende Aufgaben gleichzeitig ausgeführt werden.

Weitere Informationen zur Preisgestaltung finden Sie unter HealthOmics Preisgestaltung. HealthOmics

HealthOmics Ausgaben ausführen

Wenn ein WDL- oder CWL-Lauf abgeschlossen ist, enthalten die Ausgaben eine Ausgabeübersichtsdatei (im JSON-Format), in der alle durch den Lauf erzeugten Ausgaben aufgeführt sind. Sie können die Ausgabe-Zusammenfassungsdatei für folgende Zwecke verwenden:

- Ermitteln Sie programmgesteuert die Ausgabedateien, die durch den Lauf generiert wurden.
- Stellen Sie sicher, dass der Lauf alle erwarteten Ausgaben erzeugt hat.

Themen

- Führen Sie die Ausgabezusammenfassung für WDL aus
- Führen Sie die Ausgabezusammenfassung für CWL aus

Führen Sie die Ausgabezusammenfassung für WDL aus

Wenn ein WDL-Lauf abgeschlossen ist, HealthOmics wird eine Ausgabezusammenfassungsdatei mit dem Namen erstellt. output.json

Für jede Ausgabe des Workflows gibt es ein entsprechendes key/value Paar in der Datei. Der Schlüssel enthält den Workflow-Namen und den Ausgabenamen im folgenden Format:`WorkflowName.output_name`. Bei einer Dateiausgabe ist der Wert ein S3-URI, der auf den Ausgabeort in S3 verweist, an dem die Datei gespeichert ist. Bei einer Array [File] -Ausgabe ist der Wert ein Array von S3 URIs.

Das folgende Beispiel zeigt die output.json Datei für einen Workflow mit dem NamenBWAMappingWorkflow.

```
{
  "BWAMappingWorkflow.bam_indexes": [
    "s3://omics-outputs/8886192/out/bam_indexes/0/
pbmc8k_S1_L007_R1_001.sorted.bam.bai",
    "s3://omics-outputs/8886192/out/bam_indexes/1/pbmc8k_S1_L008_R1_001.sorted.bam.bai"
  ],
  "BWAMappingWorkflow.mapping_stats": "s3://omics-outputs/8886192/out/mapping_stats/
genome_mapping_final_stats.txt",
  "BWAMappingWorkflow.merged_bam": "s3://omics-outputs/8886192/out/merged_bam/
genome_mapping.merged.bam",
  "BWAMappingWorkflow.merged_bam_index": "s3://omics-outputs/8886192/out/
merged_bam_index/genome_mapping.merged.bam.bai",
```

```
  "BWAMappingWorkflow.reference_index_tar": "s3://omics-outputs/8886192/out/
reference_index_tar/reference_index.tar",
  "BWAMappingWorkflow.sorted_bams": [
    "s3://omics-outputs/8886192/out/sorted_bams/0/pbmc8k_S1_L007_R1_001.sorted.bam",
    "s3://omics-outputs/8886192/out/sorted_bams/1/pbmc8k_S1_L008_R1_001.sorted.bam"
  ],
  "BWAMappingWorkflow.unmapped_bams": [
    "s3://omics-outputs/8886192/out/unmapped_bams/0/
pbmc8k_S1_L007_R1_001.unmapped.bam",
    "s3://omics-outputs/8886192/out/unmapped_bams/1/pbmc8k_S1_L008_R1_001.unmapped.bam"
  ]
}
```

Wenn der Workflow Ausgaben erzeugt, die keine Dateitypen haben (wie String, Int, Float oder Bool), ist der Feldwert ein JSON-Primitiv. Zum Beispiel:

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

Führen Sie die Ausgabezusammenfassung für CWL aus

Wenn ein CWL-Lauf abgeschlossen ist, HealthOmics wird eine Ausgabezusammenfassungsdatei mit dem folgenden Namen outputs.json erstellt:

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

Die Ausgabe-Zusammenfassungsdatei enthält eine Liste von Ausgaben. Jede Ausgabe ist ein key/value Paar, wobei der Schlüssel der Name der Ausgabe ist. Der Wert ist ein Objekt, das die folgenden Eigenschaften enthält:

- location — Der vollqualifizierte Pfad zur Ausgabedatei
- basename — Der Dateiname-Teil des Pfads
- class — Der Typ der Ausgabe, normalerweise Datei
- Größe — Die Größe der Datei in Byte

Im folgenden Beispiel enthält die Datei output.json eine Liste mit zwei Ausgabedateien.

```
{
  "example_output": {
    "location": "{my-S3outputpath}/{runId}/{run-uuid}/out/output.txt",
    "basename": "output.txt",
    "class": "File",
    "size": 13
  },
  "another_output": {
    "location": "{my-S3outputpath}/{runId}/{run-uuid}/out/metrics.json",
    "basename": "metrics.json",
    "class": "File",
    "size": 256
  }
}
```

Gründe für Ausführungsfehler

Wenn ein Lauf fehlschlägt, verwenden Sie den GetRun API-Vorgang zum Abrufen der Fehlerursache.

Die Reihenfolge, in der der Ausführungsdatensatz, die Datensätze für fehlgeschlagene Aufgaben und die Protokolle gelesen werden, finden Sie unter Debuggen eines HealthOmics Workflow-Ausführung.

Überprüfe die Fehlerursache, um herauszufinden, warum der Lauf fehlgeschlagen ist. In der folgenden Tabelle sind die einzelnen Fehlergründe zusammen mit einer Beschreibung des Fehlers aufgeführt.

Failure reason (Fehlerursache)	Fehlerbeschreibung
ALL_PROFILES_INSTANCE_RESERVATION_FAILED	Es gibt nicht genug Instance-Kapazität für keinen der Beschleuniger, die Sie im `omicsResourceFallbackOrder` Richtlinie. Erwägen Sie, die Wartezeit für Ressourcen zu verlängern, oder geben Sie alternative Beschleuniger oder eine CPU-Option in Ihrer Fallback-Reihenfolge an. Weitere Informationen zum Ressourcenfallback finden Sie unter Erweiterte Ressourcenkonfiguration.
ASSUME_ROLE_FAILED	HealthOmics hat keine Berechtigung, die Rolle zu übernehmen. Geben Sie den an HealthOmics Principal in der Vertrauensbeziehung für die Rolle.

Failure reason (Fehlerursache)	Fehlerbeschreibung
CONTAINER_ERROR KANN NICHT GESTARTET WERDEN	Workflow-Aufgabe konnte nicht gestartet werden: *name*, ID: *ID* Container mit Bild: *image name*. Vergewissern Sie sich, dass das Bild gültig ist, und versuchen Sie es erneut.
CANNOT_START_CONTAINER_SIZE_ERROR	Workflow-Aufgabe konnte nicht gestartet werden: *name*, ID: *ID* Container mit Bild: *image name*. Stellen Sie sicher, dass die Bildgröße weniger als 45 GiB (95 GiB für eine GPU-Instance) beträgt, und versuchen Sie es erneut.
ECR_IMAGE_NOT_FOUND	Bild wurde unter URI nicht gefunden: *uri*. Stellen Sie sicher, dass das Image-Tag oder der Digest im Amazon ECR-Repository vorhanden ist, und versuchen Sie es erneut.
ECR_PERMISSION_ERROR	HealthOmics hat keine Berechtigung, auf die Bild-URI zuzugreifen. Vergewissern Sie sich, dass das private Amazon ECR-Repository existiert und Zugriff auf das gewährt hat HealthOmics Dienstprinzipal.
ECR_REPOSITORY_NICHT_GEFUNDEN	Das Amazon ECR-Repository wurde für den Bild-URI nicht gefunden: *uri*. Vergewissern Sie sich, dass das Repository existiert und die URI korrekt ist, und versuchen Sie es erneut.
EXPORT_FAILED	Der Export ist fehlgeschlagen. Stellen Sie sicher, dass der Ausgabe-Bucket existiert und die Run-Rolle über Schreibberechtigungen für den Bucket verfügt.
FILE_SYSTEM_OUT_OF_SPACE	Das Dateisystem hat nicht genug Speicherplatz. Erhöhen Sie die Größe des Dateisystems und führen Sie es erneut aus.
IMAGE_VERIFICATION_FAILURE	Das Bild konnte nicht verifiziert werden *image name*. Um das Problem zu beheben, versuchen Sie, das Bild abzurufen und es dann erneut in Ihr ECR-Repository zu übertragen.

Failure reason (Fehlerursache)	Fehlerbeschreibung
IMPORT_FEHLGESCHLAGEN	Der Import ist fehlgeschlagen. Stellen Sie sicher, dass die Eingabedatei vorhanden ist und die Ausführungsrolle auf die Eingabe zugreifen kann.
INACTIVE_OMICS_STORAGE_RESOURCE	Die HealthOmics Die Speicher-URI befindet sich nicht im Status AKTIV. Aktivieren Sie das Leseset und versuchen Sie es erneut. Weitere Informationen zur Aktivierung von Lesesätzen finden Sie unter Lesesätze aktivieren in HealthOmics.
EINGABE_URI_NICHT_GEFUNDEN	Die angegebene URI existiert nicht: *`uri`*. Überprüfen Sie, ob der URI-Pfad existiert, und bestätigen Sie, dass die Rolle auf das Objekt zugreifen kann.
INSTANCE_RESERVATION_FAILED	Es ist nicht genug Instanzkapazität vorhanden, um den Workflow-Lauf abzuschließen. Warten Sie und versuchen Sie erneut, den Workflow auszuführen. Weitere Informationen zur Behebung fehlgeschlagener Fehler bei der Instanzreservierung finden Sie unter Behebung von INSTANCE_RESERVATION_FAILED-Fehlern.
UNGÜLTIGE_ECR_IMAGE-URI	Die Amazon ECR-Image-URI-Struktur ist nicht gültig. Geben Sie eine gültige URI ein und versuchen Sie es erneut.
UNGÜLTIGER_TASK_RESSOURCENWERT	Die angeforderte GPU, CPU oder Arbeitsspeicher ist entweder zu hoch für die verfügbare Rechenkapazität oder liegt unter dem Mindestwert von 1 für die Aufgabe *`ID`*.
UNGÜLTIGE_URI_EINGABE	Die URI-Struktur ist nicht gültig *`uri`*. Überprüfen Sie die URI-Struktur und versuchen Sie es erneut.
MODIFIED_INPUT_RESOURCE	Der angegebene URI *`uri`* wurde nach dem Start des Laufs geändert. Versuchen Sie den Lauf erneut.

Failure reason (Fehlerursache)	Fehlerbeschreibung
OUT_OF_MEMORY_ERROR	Die Workflow-Aufgabe *ID* Der Speicher ging aus. Erhöhen Sie den Speicherwert in der Workflow-Definition und versuchen Sie den Rechenlauf erneut.
RUN_TASK_FAILED	Der Lauf ist fehlgeschlagen, weil die Aufgabe fehlgeschlagen ist. Um den Fehler der Aufgabe zu debuggen, verwenden Sie den GetRunTask API-Betrieb und Amazon CloudWatch Protokolliert den Stream.
RUN_TIMED_OUT	Führen Sie Timeout danach aus *number* Minuten.
SERVICEFEHLER	Im Dienst ist ein vorübergehender Fehler aufgetreten. Versuchen Sie erneut, den Workflow auszuführen.
TASK_TIMED_OUT	Aufgabe *id* hat nachher das Zeitlimit überschritten *number* Sekunden.
UNSUPPORTED_INPUT_SIZE	Die gesamte Eingabegröße ist zu hoch. Verringern Sie die Eingabegröße und versuchen Sie es erneut.
WORKFLOW_RUN_FAILED	Workflow-Ausführung ist fehlgeschlagen. Überprüfen Sie das CloudWatch Log-Engine-Log-Stream: *ID* um den Fehler zu debuggen.
WORKFLOW_VER_VALIDATION_FAILED	HealthOmics unterstützt die angeforderte Nextflow-Version nicht: *version* --. Die neueste unterstützte Version ist *version*. Ändern Sie Ihre Nextflow-Version auf eine unterstützte Version und versuchen Sie es erneut.
UNSUPPORTED_GPU_INSTANCE_TYPE	Der angeforderte Instanztyp wird in nicht unterstützt *Region*. Versuchen Sie den Lauf mit einem GPU-Instanztyp, der in dieser Region unterstützt wird. Verfügbare Instanztypen sind *GPU instance types*.

Behebung von INSTANCE_RESERVATION_FAILED-Fehlern

GPU-Instances sind On-Demand-Ressourcen, die möglicherweise nicht immer sofort in Ihrer bevorzugten Availability Zone oder Region verfügbar sind.

Was HealthOmics macht automatisch

HealthOmics führt in Ihrem Namen mehrere Optimierungen durch, um den Erfolg der GPU-Akquisition zu maximieren:

- Multi-AZ Suche: HealthOmics prüft mehrere Availability Zones auf GPU-Kapazität, nicht nur eine einzelne AZ.
- Automatische Wiederholungsversuche: Wenn eine GPU nicht sofort verfügbar ist, HealthOmics versucht mehrmals, Kapazität abzurufen, sobald sie verfügbar ist.
- GPU-Aufbewahrung bei allen Aufgaben: Wenn Ihr Workflow sequentielle Aufgaben umfasst, für die eine GPU erforderlich ist, HealthOmics behält die GPU, die Sie bereits haben, für nachfolgende Aufgaben bei, anstatt sie freizugeben und erneut anzufordern. So wird der erfolgreiche Abschluss mehrstufiger GPU-Workflows gewährleistet.

Die folgenden zusätzlichen bewährten Methoden können deine Erfolgsquote beim Laufen weiter verbessern GPU-accelerated Workflows aktiviert HealthOmics:

1. Verwenden Sie Beschleuniger-Pakete, um die Flexibilität zu maximieren

   Anstatt einen bestimmten GPU-Instance-Typ anzufordern, verwenden Sie Accelerator-Pakete, die Folgendes ermöglichen HealthOmics um aus mehreren kompatiblen GPU-Generationen auszuwählen (z. B. g4dn, g5, g6). Je mehr GPU-Typen Ihr Workflow akzeptieren kann, desto höher ist die Wahrscheinlichkeit, verfügbare Kapazität zu finden.

   Weitere Informationen hierzu finden Sie unter Taskbeschleuniger in einem HealthOmics Workflow-Definition. Beispiel:

   - `nvidia-t4-a10g-l4` stammt aus den Instance-Familien G4, G5 und G6
   - `nvidia-l4-a10g` bezieht sich auf G5- und G6-Instanzfamilien
2. Erwägen Sie, Ihren Workflow in einem anderen auszuführen AWS Region

   Die GPU-Verfügbarkeit variiert je nach AWS Regionen. Wenn deine Anforderungen an den Datenspeicherort dies zulassen, solltest du in Erwägung ziehen, zu rennen GPU-intensive Workflows in einer anderen Region, in der Kapazität möglicherweise leichter verfügbar ist. Weitere

Informationen zu den unterstützten Regionen finden Sie unter Servicekontingente und Endpunkte für AWS HealthOmics.

3. Ressourcen-Fallback konfigurieren

   Für WDL- und Nextflow-Workflows können Sie das verwenden `omicsResourceFallbackOrder` Direktive zur Definition einer geordneten Liste von Ressourcenprofilen (GPU oder CPU) pro Aufgabe. Wenn Ihr bevorzugter Beschleunigertyp nicht verfügbar ist, HealthOmics probiert automatisch den nächsten Beschleunigertyp oder die nächste CPU aus, anstatt auszufallen. Sie können auch angeben, wie lange HealthOmics sollte nach einer GPU suchen. Zu den Anwendungsfällen zählen:

   - Fallback von GPU zu GPU — probieren Sie zuerst Ihren bevorzugten Beschleuniger aus und greifen Sie dann auf eine Alternative zurück (versuchen Sie es zum Beispiel `nvidia-l40s` zuerst, dann `nvidia-l4`).
   - Fallback von GPU zu CPU — füge ein Finale hinzu CPU-only Profil, sodass die Aufgabe abgeschlossen wird, auch wenn keine GPU-Kapazität verfügbar ist.

   Wenn keines der Profile in Ihrer Fallback-Reihenfolge innerhalb der Wartezeit über eine Kapazität verfügt, schlägt der Lauf fehl mit ALL_PROFILES_INSTANCE_RESERVATION_FAILED. Für WDL-Workflows HealthOmics versucht die Aufgabe wegen dieses Fehlers nicht erneut. Für Nextflow-Workflows gilt: `errorStrategy` und `maxRetries` stellen Sie fest, ob die Aufgabe erneut versucht wird und ein Wiederholungsversuch erneut mit dem ersten Profil beginnt. Erwägen Sie, die Wartezeit für Ressourcen zu verlängern, oder geben Sie alternative Beschleuniger oder eine CPU-Option in Ihrer Ausweichreihenfolge an.

   Weitere Informationen hierzu finden Sie unter Erweiterte Ressourcenkonfiguration.

Hinweise für Läufe, die nicht reagieren

Bei der Entwicklung neuer Workflows können Läufe oder bestimmte Aufgaben „hängen bleiben“ oder „hängen bleiben“, wenn es Probleme mit Ihrem Code gibt und Aufgaben Prozesse nicht ordnungsgemäß beenden. Es kann schwierig sein, dies zu beheben und zu erkennen, da es normal ist, dass Aufgaben über einen längeren Zeitraum ausgeführt werden. Folgen Sie den empfohlenen bewährten Methoden in den folgenden Abschnitten, um zu verhindern und zu erkennen, dass Ausführungen nicht reagieren.

Bewährte Methoden zur Vermeidung nicht reagierender Läufe

- Stellen Sie sicher, dass Sie alle in Ihrem Taskcode geöffneten Dateien schließen. Das Öffnen zu vieler Dateien kann gelegentlich zu Threading-Problemen in den Workflow-Engines führen.
- Hintergrundprozesse, die durch eine Workflow-Aufgabe erstellt wurden, sollten beendet werden, wenn die Aufgabe beendet wird. Wenn ein Hintergrundprozess jedoch nicht ordnungsgemäß beendet wird, müssen Sie diesen Prozess in Ihrem Aufgabencode explizit beenden.
- Stellen Sie sicher, dass Ihre Prozesse nicht in einer Schleife laufen, ohne sie zu beenden. Dies kann dazu führen, dass ein Lauf nicht mehr reagiert und eine Änderung am Code der Workflow-Definition erforderlich ist, um das Problem zu beheben.
- Sorgen Sie für eine angemessene Speicher- und CPU-Zuweisung für Ihre Aufgaben. Analysieren Sie die CloudWatch aufzeichnen oder benutze die Führen Sie Analyzer aus bei erfolgreich abgeschlossenen Durchläufen Ihres Workflows, um sicherzustellen, dass Sie die optimale Rechenzuweisung haben. Verwenden Sie den Run Analyzer `headroom` Parameter, der zusätzlichen Spielraum einbezieht und sicherstellt, dass die Prozesse über genügend Ressourcen verfügen, um sie abzuschließen. Rechnen Sie den zugewiesenen Arbeitsspeicher und die CPU mit mindestens 5% Spielraum aus, um die im Hintergrund ablaufenden Betriebssystemprozesse zu berücksichtigen.
  - Erhöhen Sie außerdem die Bandbreitengröße der Instance, wenn die Instance einen höheren Durchsatz benötigt. Bei Amazon EC2-Instances mit weniger als 16 vCPUs (Größe 4xl und kleiner) kann es zu Durchsatzspitzen kommen. Weitere Informationen zum Durchsatz von Amazon EC2-Instances finden Sie unter Verfügbare Amazon EC2-Instance-Bandbreite.
- Stellen Sie sicher, dass Sie die richtige Dateisystemgröße für Ihre Läufe verwenden. Bei nicht reagierenden Läufen, die statischen Laufspeicher verwenden, sollten Sie erwägen, die Speicherzuweisung für statische Rechenläufe zu erhöhen, um einen höheren I/O-Durchsatz und eine höhere Speicherkapazität im Dateisystem zu ermöglichen. Analysieren Sie das Run-Manifest, um den maximalen Speicherplatz im Dateisystem zu ermitteln. Verwenden Sie den Run Analyzer, um zu ermitteln, ob die Dateisystemzuweisung erhöht werden muss.

Bewährte Methoden zum Abfangen nicht reagierender Läufe

- Verwenden Sie bei der Entwicklung neuer Workflows eine Ausführungsgruppe, deren maximale Ausführungszeit festgelegt ist, um außer Kontrolle geratenen Code abzufangen. Wenn eine Ausführung beispielsweise eine Stunde in Anspruch nehmen sollte, ordnen Sie sie einer Ausführungsgruppe zu, die nach 2 oder 3 Stunden (oder je nach Anwendungsfall in einem anderen Zeitraum) das Zeitlimit überschreitet, um außer Kontrolle geratene Jobs abzufangen. Wenden Sie außerdem einen Puffer an, um Abweichungen bei den Verarbeitungszeiten zu berücksichtigen.

- Richten Sie eine Reihe von Ausführungsgruppen mit unterschiedlichen maximalen Laufzeitlimits ein. Beispielsweise könnten Sie kurze Läufe einer Ausführungsgruppe zuweisen, die die Läufe nach einigen Stunden beendet, und einer Gruppe mit langen Durchläufen, die Läufe nach ein paar Tagen beendet, basierend auf Ihrer erwarteten Workflow-Dauer.
- HealthOmics hat ein standardmäßiges Service-Limit für die maximale Ausführungsdauer von 604.800 Sekunden oder 7 Tagen, das durch eine Anfrage im Kontingent-Tool angepasst werden kann. Fordern Sie nur dann eine Erhöhung des Servicelimits um dieses Kontingent an, wenn Sie Läufe haben, deren Dauer sich einer Woche nähert. Wenn Sie eine Mischung aus kurzen und langen Läufen haben und keine Run-Gruppen verwenden, sollten Sie erwägen, die Läufe mit langer Laufzeit in einem separaten Konto mit einem höheren Service-Limit für die maximale Laufdauer abzulegen.
- Untersuchen Sie den CloudWatch aufzeichnen für Aufgaben, von denen Sie vermuten, dass sie nicht reagieren könnten. Wenn eine Aufgabe normalerweise reguläre Protokollanweisungen ausgibt und dies über einen längeren Zeitraum nicht getan wurde, steckt die Aufgabe wahrscheinlich fest oder ist eingefroren.

Was tun, wenn ein Lauf nicht reagiert

- Brechen Sie den Lauf ab, um zusätzliche Kosten zu vermeiden.
- Untersuchen Sie das Aufgabenprotokolle um zu überprüfen, ob Prozesse nicht korrekt beendet wurden.
- Untersuchen Sie das Motorprotokolle um abnormales Motorverhalten zu identifizieren.
- Vergleichen Sie die Task- und Engine-Protokolle des nicht reagierenden Laufs mit denen identischer, erfolgreich abgeschlossener Läufe. Dies kann helfen, alle Unterschiede zu identifizieren, die möglicherweise dazu geführt haben, dass das Verhalten nicht reagiert.
- Wenn Sie nicht in der Lage sind, die Ursache zu ermitteln, geben Sie eine Fall unterstützen und beinhalten Folgendes:
  - ARN des hängengebliebenen Laufs und ARN eines identischen Laufs, der erfolgreich abgeschlossen wurde.
  - Engine-Logs (verfügbar, sobald der Lauf abgebrochen wurde oder fehlschlägt)
  - Aufgabenprotokolle für die Aufgabe, die nicht reagiert. Zur Fehlerbehebung benötigen wir keine Aufgabenprotokolle für alle Aufgaben im Workflow.

Aufgabenlebenszyklus in einem HealthOmics Lauf

EIN Aufgabe ist ein einzelner Prozess innerhalb eines Durchlaufs. HealthOmics ordnet jede Aufgabe in Ihrem Workflow einem Omics-Computing-Instance-Typ zu, der den für die Aufgabe erforderlichen Ressourcen am besten entspricht. Sie geben die erforderlichen Ressourcen in der Workflow-Definition an. Weitere Informationen finden Sie unter Rechen- und Speicheranforderungen für HealthOmics Aufgaben.

HealthOmics stellt temporären Ausführungsspeicher für die zu verwendende Aufgabe bereit. HealthOmics kopiert die Eingabedateien der Aufgabe als schreibgeschützte Dateien in den temporären Ausführungsspeicher. HealthOmics stellt symbolische Links bereit, sodass die Task auf die Eingabedateien aus dem Arbeitsverzeichnis zugreifen kann. Die Aufgabe hat nur Zugriff auf die Dateien, die Sie in der Workflow-Definitionsdatei deklarieren.

Werte für den Aufgabenstatus

Sie können den Fortschritt einer Aufgabe verfolgen, indem Sie den Aufgabenstatus überwachen. Wenn du einen Lauf startest, HealthOmics setzt den Aufgabenstatus auf Pending für jede Aufgabe in der Ausführung. Wenn die Aufgabe beginnt und ihren Lebenszyklus durchläuft, HealthOmics aktualisiert den Statuswert, um den aktuellen Fortschritt widerzuspiegeln.

Sie können den Aufgabenstatus mit einer der folgenden Methoden abrufen:

- Die HealthOmics Die Konsole zeigt den Status jeder Aufgabe bei einem Lauf auf der Run details Seite.
- Die GetRunTask Der API-Vorgang gibt den Aufgabenstatus zurück.
- Sie können den Aufgabenstatus mit folgenden Methoden überwachen EventBridge Ereignisse. Weitere Informationen finden Sie unter Verwenden EventBridge mit AWS HealthOmics.

Den aktuellen Status einer Aufgabe können Sie mit dem GetRunTask API-Betrieb. Das HealthOmics Die Konsole zeigt den Status für jede Aufgabe in einem Lauf auf der Run details Seite.

HealthOmics unterstützt die folgenden Aufgabenstatuswerte:

Ausstehend

Ihre Aufgabe befindet sich in der Warteschlange und wartet darauf, gestartet zu werden. Aufgaben bleiben für einen kurzen Zeitraum ausstehend, bevor sie beginnen.

- Aufgaben bleiben ausstehend, nachdem Ihr Konto die maximale Anzahl gleichzeitiger Aufgaben erreicht hat.
- Aufgaben bleiben ausstehend, wenn die Ausführung Teil einer Ausführungsgruppe ist, die einen ihrer Ressourcenhöchstwerte erreicht hat.
- Sie können die Ausführungsprioritäten so anpassen, dass bestimmte Läufe in der Warteschlange und die zugehörigen Aufgaben vor anderen Läufen in der Warteschlange beginnen. Weitere Hinweise zur Ausführungspriorität finden Sie unter Priorität ausführen

Wird gestartet

HealthOmics erstellt die Aufgabe und stellt die für die Aufgabe erforderlichen Ressourcen bereit, z. B. den Workflow-Aufgabenknoten.

In Ausführung

Der Aufgabenstatus lautet Läuft währenddessen HealthOmics verarbeitet die Aufgabe.

Wird angehalten

Nach Abschluss der Aufgabenverarbeitung und dem Exportieren der Ausgabedaten wechselt die Aufgabe zu Stoppen.

- HealthOmics hebt die Bereitstellung des Workflow-Aufgabenknotens auf.

Completed

HealthOmics hat die Verarbeitung der Aufgabe abgeschlossen und die Ausgabedaten in das Run-Storage-Dateisystem übertragen.

Fehlgeschlagen

HealthOmics Bei der Verarbeitung der Aufgabe ist ein Fehler aufgetreten und die Aufgabe wurde nicht abgeschlossen.

- Die Aufgabe wechselt in den Status Stoppen (HealthOmics stellt die Ressourcen wieder bereit) und wechselt dann in den Status Fehlgeschlagen.
- Wenn es sich bei dem Fehler um einen Dienstfehler handelt (HTTP-Statuscode 5XX) und der Workflow Wiederholungen für diese Aufgabe unterstützt, HealthOmics versucht erneut, die Aufgabe zu verarbeiten. HealthOmics weist dem Wiederholungsversuch eine neue Aufgaben-ID zu.
- Wenn Ihre Aufgabe fehlschlägt mit INSTANCE_RESERVATION_FAILED oder ALL_PROFILES_INSTANCE_RESERVATION_FAILED, Sie können verschiedene

bewährte Methoden anwenden, um den Erfolg der GPU-Akquisition zu verbessern. Weitere Informationen zu Accelerator-Paketen, regionaler Flexibilität und der Reihenfolge der Ressourcenausfälle finden Sie unter Behebung von INSTANCE_RESERVATION_FAILED-Fehlern. HealthOmics versucht eine fehlgeschlagene Aufgabe nicht erneut mit ALL_PROFILES_INSTANCE_RESERVATION_FAILED für WDL-Workflows. Für Nextflow-Workflows `errorStrategy` und `maxRetries` stellen Sie fest, ob die Aufgabe erneut versucht wird und ein Wiederholungsversuch erneut mit dem ersten Profil beginnt.

Abgebrochen

HealthOmics stoppt die Aufgabe nach einer vom Benutzer initiierten Aufforderung, die Ausführung abzubrechen.

- Die Aufgabe geht in den Status Stoppen über (HealthOmics stellt die Ressourcen wieder bereit) und wechselt dann in den Status Storniert.

Problembehandlung bei Workflow-Aufgaben

Im Folgenden finden Sie bewährte Methoden und Überlegungen zur Problembehandlung Ihrer Aufgaben.

Die Reihenfolge, in der der Datensatz einer fehlgeschlagenen Aufgabe, ihre Protokolle und ihre Messwerte gelesen werden, finden Sie unter Debuggen eines HealthOmics Workflow-Ausführung.

- Aufgabenprotokolle stützen sich auf `STDOUT` und `STDERR` wird von der Aufgabe produziert. Wenn die in der Aufgabe verwendete Anwendung keines dieser Elemente erzeugt, wird kein Aufgabenprotokoll erstellt. Um beim Debuggen zu helfen, verwenden Sie Anwendungen in `verbose` Modus.
- Um die Befehle, die in einer Aufgabe ausgeführt werden, zusammen mit ihren interpolierten Werten anzuzeigen, verwenden Sie den `set -x` Bash-Befehl. Auf diese Weise können Sie feststellen, ob die Aufgabe die richtigen Eingaben verwendet, und ermitteln, an welchen Stellen Fehler die Aufgabe möglicherweise daran gehindert haben, wie beabsichtigt ausgeführt zu werden.
- Verwenden Sie das `echo` Befehl zur Ausgabe der Variablenwerte an `STDOUT` oder `STDERR`. Auf diese Weise können Sie überprüfen, ob sie wie erwartet festgelegt wurden.
- Benutze Befehle wie `ls -l <name_of_input_file>` um zu bestätigen, dass Eingaben vorhanden sind und die erwartete Größe haben. Wenn dies nicht der Fall ist, könnte dies auf ein Problem hinweisen, bei dem eine vorherige Aufgabe aufgrund eines Fehlers zu leeren Ausgaben geführt hat.

- Benutze den Befehl `df -Ph . | awk 'NR==2 {print $4}'` in einem Aufgabenskript, um den aktuell für die Aufgabe verfügbaren Speicherplatz zu ermitteln und Situationen zu identifizieren, in denen Sie den Workflow möglicherweise mit zusätzlicher Speicherzuweisung ausführen müssen.

Wenn Sie einen der vorherigen Befehle in ein Taskskript aufnehmen, wird davon ausgegangen, dass der Aufgabencontainer auch diese Befehle enthält und dass sie sich auf der `path` der Container-Umgebung.

Debuggen eines HealthOmics Workflow-Ausführung

HealthOmics meldet das Verhalten eines Laufs anhand mehrerer Quellen, von denen jede eine andere Frage beantwortet. Lesen Sie sie bei einem fehlgeschlagenen Lauf in dieser Reihenfolge, anstatt mit den Protokollen zu beginnen:

1. Lesen Sie den Ausführungsdatensatz, um zu erfahren, ob der Lauf fehlgeschlagen ist und warum.
2. Lesen Sie die Aufgabenaufzeichnungen, ob eine Aufgabe fehlgeschlagen ist, für welche Aufgaben und für welche Instanztypen sie fehlgeschlagen sind. Eine Ausführung kann auch fehlschlagen, bevor eine Aufgabe erstellt wurde oder nachdem jede Aufgabe abgeschlossen wurde.
3. Lesen Sie die Protokolle: das Aufgabenprotokoll für das, was eine Aufgabe gemeldet hat, das Ausführungsprotokoll für das, was der Lauf außerhalb einer Aufgabe gemeldet hat, und das Engine-Protokoll für das, was die Engine gemeldet hat.
4. Lesen Sie die Ausführungsmetriken und führen Sie das Manifestprotokoll aus, um zu erfahren, was die Aufgabe verbraucht hat.

Für einen Lauf, der zwar abgeschlossen wurde, aber langsamer oder teurer als erwartet war, beginnen Sie mit Vergleiche die Läufe desselben Workflows.

Die Ressourcen- und Kostenleitlinien in diesem Thema beziehen sich auf private Workflows, und zwar aus dem unter Führen Sie die Optimierung für ein privates aus HealthOmics Arbeitsablauf. Die API-Operationen und Antwortfelder, die in den folgenden Schritten verwendet werden, finden Sie unter Holen Sie sich Informationen zum Lauf.

Ersetzen Sie in den Befehlen in diesem Thema die folgenden Werte durch Ihre eigenen:

`run_id`

Die ID der Ausführung, die Sie debuggen.

duration

Das Fenster, das aws logs tail lautet: eine relative Zeit wie 2h für einen Lauf, der kürzlich beendet wurde, oder einen Zeitstempel wie 2026-03-26T22:00:00 für einen älteren.

run_uuid

Der uuid Feld des Datensatzes dieses Laufs.

task_id

Die ID einer fehlgeschlagenen Aufgabe in diesem Lauf.

workflow_id

Die ID des Workflows, dessen Ausführungen Sie auflisten möchten.

Themen

- Diagnostiziert einen fehlgeschlagenen Lauf
- Untersuchen Sie die Ressourcennutzung anhand von Run-Metriken
- Vergleiche die Läufe desselben Workflows
- Verwandeln Sie die Ergebnisse in Kosten- und Größenänderungen

Diagnostiziert einen fehlgeschlagenen Lauf

Um einen fehlgeschlagenen Lauf zu diagnostizieren

1. Rufen Sie den Laufdatensatz ab und lesen Sie `status`, `failureReason`, `statusMessage`, und `logLocation`. Bewahren Sie die Antwort auf: In späteren Schritten wird sie verwendet. `logLocation` gibt die ARNs des Ausführungsprotokolls an (`runLogStream`) und das Motorprotokoll (`engineLogStream`).

   ```
   aws omics get-run --id run_id
   ```

   Schlagen Sie den Wert von nach `failureReason` in Gründe für Ausführungsfehler. Wenn `status` ist nicht `FAILED`, siehst du Werte für den Ausführungsstatus stattdessen.

   Einige Fehler treten schon einmal auf HealthOmics erstellt eine beliebige Aufgabe oder nach Abschluss jeder Aufgabe, sodass für die nächsten drei Schritte kein Datensatz zum Lesen

übrig bleibt. Eine Rolle, ein Eingabe-, Import- oder Exportfehler sind das Gleiche wie ein Bild HealthOmics lehnt ab, bevor die Aufgabe erstellt wird (ECR_IMAGE_NOT_FOUND, ECR_PERMISSION_ERROR, ECR_REPOSITORY_NICHT_GEFUNDEN, oder INVALID_ECR_IMAGE_URI). Gehen Sie bei jedem von ihnen später in diesem Verfahren zum Ausführungsprotokoll über. Bei einem Image-Fehler `statusMessage` benennt die Bild-URI, fahren Sie also fort von Container-Images für private Workflows und Amazon-ECR-Berechtigungen. Wenn der nächste Schritt eine fehlgeschlagene Aufgabe zurückgibt, lesen Sie stattdessen diese Aufgabe.

2. Listet die fehlgeschlagenen Aufgaben auf. Jedes Element enthält die Aufgaben `taskId`, das im nächsten Schritt verwendet wird, plus seinen Namen, `creationTime`, und, sobald es sie erreicht hat, seinen Instanztyp, `startTime`, und `stopTime`.

```
aws omics list-run-tasks --id run_id --status FAILED
```

   Wenn dadurch keine Aufgaben zurückgegeben werden, ist der Lauf einer der Fälle in Schritt 1: Überspringen Sie die nächsten beiden Schritte und lesen Sie das Ausführungsprotokoll.

3. Rufen Sie für jede fehlgeschlagene Aufgabe ihren Datensatz ab. Es fügt hinzu `failureReason`, `statusMessage`, `imageDetails`, und `logStream` zu dem, was der vorherige Schritt zurückgegeben hat. `logStream` ist der ARN des Log-Streams, den der nächste Schritt liest, und der Name des Streams folgt `log-stream:` darin.

```
aws omics get-run-task --id run_id --task-id task_id
```

4. Lesen Sie das Aufgabenprotokoll. Die Ereignisse, die den Fehler erklären, befinden sich am Ende des Streams.

```
aws logs tail /aws/omics/WorkflowLog \
  --log-stream-names run/run_id/task/task_id \
  --since duration
```

   Eingestellt *`duration`* um zu starten, bevor die Aufgabe ausgeführt wurde: Der Befehl liest von damals bis heute, und sein Standardfenster sind die letzten 10 Minuten, sodass eine Ausführung, die früher abgeschlossen wurde, keine Ereignisse zurückgibt. Die Protokollgruppe und das Format jedes Streams finden Sie unter Loggt sich ein CloudWatch.

> **Note**
>
> Warum eine Aufgabe möglicherweise überhaupt kein Protokoll hat und wie Sie eine Aufgabe besser debuggen können, finden Sie unter Problembehandlung bei Workflow-Aufgaben.

5. Lesen Sie, welches der beiden anderen Protokolle des Laufs die Frage beantwortet, die Sie hierher gebracht hat. Beide verwenden den Befehl Schritt 4 mit einem anderen Stream.

```
aws logs tail /aws/omics/WorkflowLog \
  --log-stream-names run/run_id \
  --since duration
```

   Der Lauf hat keine fehlgeschlagene Aufgabe

   Lesen Sie das Laufprotokoll, `run/`*`run_id`*, wie im vorherigen Befehl. Es meldet den eigenen Fortschritt des Laufs, einschließlich der Import- und Exportschritte, die ohne Aufgabe fehlschlagen. Was es meldet, finden Sie unter Protokolltypen für HealthOmics Workflows.

   Eine Aufgabe ist fehlgeschlagen und ihr Protokoll erklärt nicht, warum

   Lesen Sie das Motorprotokoll, `run/`*`run_id`*`/engine`, in dem berichtet wird, was die Engine aus dem Ausfall gemacht hat, z. B. die Position in der Workflow-Definition. Sein ARN ist in `logLocation.engineLogStream` ab Schritt 1; wenn dieses Feld nicht in der Antwort enthalten ist, enthält der Lauf kein Motorprotokoll.

   Um die Definition an der Position zu lesen, die die Engine benennt, exportieren Sie sie. Der folgende Befehl gibt eine Download-URL für die Definition zurück, die der Lauf verwendet hat.

```
aws omics get-workflow --id workflow_id --export DEFINITION
```

Wo Sie fortfahren müssen, hängt vom Grund des Fehlers ab. Die folgende Liste behandelt die häufigsten Ressourcen- und Containerfehler. Wie viel die Aufgaben in Anspruch genommen haben und ob es sich bei dem Fehler um ein Größenproblem handelte, finden Sie unter Verwandeln Sie die Ergebnisse in Kosten- und Größenänderungen.

CONTAINER_FEHLER KANN NICHT GESTARTET WERDEN, FEHLER IN DER GRÖSSE DES CONTAINERS KANN NICHT GESTARTET WERDEN, oder IMAGE_VERIFICATION_FAILURE

HealthOmics hat die Aufgabe erstellt und konnte ihren Container nicht starten, sodass die Aufgabe kein Protokoll erstellt hat. Nehmen Sie das Bild aus den Datensätzen der Aufgabe `imageDetails`, dann sieh Container-Images für private Workflows.

FEHLER BEI UNZUREICHENDEM ARBEITSSPEICHER

Vergleichen Sie den Speicherverbrauch der Aufgabe mit ihrer Zuweisung, wie unter beschrieben Wie dimensioniere ich die Rechenleistung und den Speicher für meinen Workflow richtig?, wobei entweder die zugehörigen Ausführungsmetriken oder die Statistiken im Ausführungsmanifestprotokoll verwendet werden Verwandeln Sie die Ergebnisse in Kosten- und Größenänderungen lokalisiert. Erhöhen Sie dann den Speicher in der Workflow-Definition und überprüfen Sie, in welchem Instanztyp der neue Wert ausgewählt wird Rechen- und Speicheranforderungen für HealthOmics Aufgaben.

TASK_TIMED_OUT oder RUN_TIMED_OUT

Bestätigen Sie, ob die Aufgabe noch funktionierte, als das Timeout abgelaufen war. Wenn nicht, behandeln Sie es als nicht reagierend und nicht als langsam, und sehen Sie Hinweise für Läufe, die nicht reagieren. Wenn das Limit, das den Lauf beendete, von einer Run-Gruppe stammte, siehe HealthOmics Run-Gruppen verwenden.

Untersuchen Sie die Ressourcennutzung anhand von Run-Metriken

Die Ausführungsmetriken decken einen Lauf ab, der noch im Gange ist. Sie können sie also verwenden, bevor ein Lauf abgeschlossen ist. Welche Serien für einen Lauf existieren, hängt von seinem Laufspeichertyp, seinem Scratch-Speichermodus und davon ab, ob für seine Aufgaben Beschleuniger angefordert wurden. Prüfen Sie daher die Bedingungen unter Verfügbare Metriken bevor Sie zu dem Schluss kommen, dass eine Metrik fehlt. Wenn ein Rechenlauf überhaupt keine Datenreihen enthält, überprüfen Sie, ob die entsprechende Funktion diese auch veröffentlichen könnte, wie unter Metriken für einen Lauf aktivieren. Die Abfragesyntax und das Konsolenverfahren finden Sie unter Abfragen HealthOmics Metriken ausführen.

Beschränken Sie jede Abfrage auf einen Lauf mit der Bezeichnung „Run Resource“. Per-task Metriken tragen auch das Task-Resource-Label, das eine Reihe mit dem Aufgabendatensatz und seinen Protokollen verknüpft. Für beide Labels siehe Gängige -Attribute.

> **Note**
>
> CloudWatch begrenzt sowohl die Anzahl der Zeitreihen, die eine PromQL-Abfrage zurückgibt, als auch den Zeitraum, den sie umfasst. Ein Lauf erzeugt mindestens eine Serie pro Aufgabe für jede aufgabenbezogene Metrik und mehr, wenn die Metrik nach Richtung oder nach GPU aufgeteilt wird. Bei einer Ausführung mit vielen Aufgaben aggregieren Sie die Reihen in der Abfrage selbst oder filtern Sie nach einer Aufgabe, einer Richtung oder einer GPU. Teilen Sie einen langen Lauf in aufeinanderfolgende Fenster auf. Die aktuellen Grenzwerte finden Sie unter Limits und Einschränkungen von PromQL.

Vergleiche die Läufe desselben Workflows

Wenn eine Ausführung langsamer oder teurer ist als eine frühere Ausführung desselben Workflows, vergleichen Sie die beiden Ausführungen Aufgabe für Aufgabe. Ordnen Sie die Aufgaben nach Namen statt nach Aufgaben-ID aus: Rufen Sie die Aufgaben beider Läufe ab und gruppieren Sie sie nach `name`, und vergleichen Sie die Dauer und die Spitzenwerte für jeden Namen. Ein Lesen des Manifestprotokolls jedes Durchlaufs gibt Ihnen die Zeiten der einzelnen Aufgaben und, sofern der Eintrag sie meldet, die Spitzenwerte; ListRunTasks gibt dir die Zeiten für einen Lauf, der noch läuft, ohne `--status FAILED` Filter, den das vorhergehende Verfahren verwendet.

Vergleichen Sie zwei Zeiträume pro Aufgabe: `startTime` zu `stopTime` ist die Dauer der Aufgabe und `creationTime` zu `startTime` gibt an, wie lange es auf den Start gewartet hat. Dies umfasst die Phasen Ausstehend und Starten in Werte für den Aufgabenstatus. Ein Lauf kann viel langsamer sein als ein anderer, solange die Dauer jeder Aufgabe gleich ist, da keine Task-Metrik die Wartezeit abdeckt.

> **Note**
>
> Mehrere Aufgaben in einem Lauf können denselben Namen haben, und ein Streuindex in einem Namen deckt nicht unbedingt dieselben Eingabedaten in beiden Durchläufen ab. Vergleichen Sie das Aggregat der Aufgaben, die einen gemeinsamen Namen haben, z. B. die längste Dauer oder den höchsten Spitzenwert, anstatt sie Index für Index zu paaren.

Aggregieren Sie eine Gruppe der abgeschlossenen Läufe, um festzustellen, was für einen Workflow normal ist, anstatt zwei Läufe zu vergleichen. Run Analyzer macht das in `--batch` Modus, wie beschrieben unter Auswirkung der Dateigrößenabweichung zwischen Durchläufen. Um die Läufe

selbst zu sammeln, filtern Sie die Antwort von ListRuns auf `workflowId`. Die Operation hat also keinen eigenen Workflow-Filter ListRuns durchsucht jeden abgeschlossenen Durchlauf, bevor Ihr Filter sie einschränkt.

```
aws omics list-runs --status COMPLETED \
  --query "items[?workflowId=='workflow_id'].id"
```

Die Verteilung über eine Reihe von Durchläufen, nicht über einen einzelnen Lauf, zeigt, ob die Zuordnung einer Aufgabe der Varianz in Ihren Daten entspricht und ob ein langsam aussehender Lauf außerhalb des normalen Bereichs liegt.

Verwandeln Sie die Ergebnisse in Kosten- und Größenänderungen

Was eine Aufgabe verbraucht hat, steht im Run-Manifest-Protokoll, dessen Stream `manifest/run/run_id/run_uuid`. HealthOmics schreibt es, nachdem der Lauf beendet ist, also setze *`duration`* um die Stoppzeit des Laufs und nicht die einer Aufgabe zu erreichen. Lies es mit dem gleichen aws logs tail Befehl wie bei den anderen Streams. Die Statistiken, die es meldet, finden Sie unter Protokolltypen für HealthOmics Workflows.

Ein Manifest-Aufgabeneintrag gibt an, was die Aufgabe reserviert hat, welchen Instanztyp sie hat und wie lange sie dauert. Er gibt an, was die Aufgabe zusammen mit diesen verwendet hat, aber nicht für jede Aufgabe. Wenn ein Eintrag keine Nutzung oder weniger meldet, als die Arbeit der Aufgabe ausmachen kann, wird stattdessen die Größe mit den Ausführungsmetriken der Aufgabe verglichen, und zwar aus dem unter Callouts.

Drei HealthOmics Verhaltensweisen schränken ein, was durch eine Änderung gespeichert wird:

- Die Kosten einer Aufgabe richten sich nach dem Instanztyp HealthOmics wählt für die von Ihnen angeforderten Ressourcen aus, nicht für die Ressourcen, die die Aufgabe verwendet. Siehe Rechen- und Speicheranforderungen für HealthOmics Aufgaben.
- Aufgabengebühren haben eine abrechnungsfähige Mindestdauer, sodass es nichts bringt, eine bereits kurze Aufgabe zu verkürzen. Siehe Auswirkungen des Ausführungsstatus auf die Preisgestaltung.
- Static Run Storage ist in festen Größen erhältlich, sodass sich die Kosten bei einer Reduzierung nur ändern, wenn die Kapazität auf eine kleinere unterstützte Größe fällt. Siehe Statisch ausgeführter Speicher.

Die Optimierungsmethoden, in die diese Ergebnisse einfließen, finden Sie unter Methoden zur Optimierung von Läufen.

Führen Sie die Optimierung für ein privates aus HealthOmics Arbeitsablauf

Sie können Läufe im Hinblick auf Gesamtkosten, Gesamtlaufzeit oder eine Kombination aus beiden optimieren. HealthOmics bietet Daten und Tools, die Ihnen bei Entscheidungen zur Laufoptimierung helfen. Die Ausführungsoptimierung gilt nicht für Ready2Run-Workflows, da Sie keine Kontrolle darüber haben, wie der Service die Ressourcenbereitstellung für diese Workflows verwaltet.

Der erste Schritt besteht darin, die aktuelle Nutzung der Aufgabenressourcen und die Kosten für die Aufgaben im Lauf zu ermitteln und dann Methoden zur Optimierung der Ausführungskosten und der Leistung anzuwenden.

Informationen zum Sammeln dieser Nutzungsdaten und zum Vergleich zwischen den einzelnen Durchläufen finden Sie unter Debuggen eines HealthOmics Workflow-Ausführung.

Themen

- Führen Sie Analyzer aus
- Ermitteln Sie die laufenden Kosten
- Ermitteln Sie die Nutzung der Laufzeit
- Erstellen einer Aufgabenzeitleiste aus dem Run-Manifest
- Methoden zur Optimierung von Läufen
- Auswirkung der Dateigrößenabweichung zwischen Durchläufen
- Methoden zur Optimierung der Parallelität von Ressourcen

Führen Sie Analyzer aus

HealthOmics bietet ein Open-Source-Tool mit dem Namen Führen Sie Analyzer aus. Dieses Tool extrahiert Informationen zur Ressourcennutzung auf Aufgabenebene für einen Lauf und schlägt Optimierungsmöglichkeiten in Bezug auf Kosten und Laufleistung vor.

> **Note**
>
> Run Analyzer schätzt die Aufgabenkosten und potenziellen Kosteneinsparungen auf der Grundlage von AWS listet die Preise zum Zeitpunkt der Ausführung des Tools auf. Bewerten Sie die Optimierungsempfehlungen und implementieren Sie diejenigen, die für

Ihre Anwendungsfälle sinnvoll sind. Testen Sie die Optimierungen, die Sie anwenden, um sicherzustellen, dass sie für Ihren Lauf funktionieren.

Run Analyzer führt die folgenden Aufgaben aus:

- Wertet Speicher- und Rechenengpässe aus.
- Identifiziert Aufgaben, für die zu viel Arbeitsspeicher oder CPU zur Verfügung steht, und empfiehlt neue Instanzgrößen, mit denen die Kosten gesenkt werden können.
- Berechnet Kostenschätzungen für einzelne Aufgaben und berechnet die potenziellen Kosteneinsparungen, wenn Sie die Empfehlungen anwenden.
- Bietet Ihnen eine Zeitleistenansicht der Aufgaben, sodass Sie die Aufgabenabhängigkeiten und die Verarbeitungsreihenfolge überprüfen können. Die Zeitleiste hilft Ihnen auch dabei, Aufgaben mit langer Laufzeit zu identifizieren.
- Enthält Empfehlungen zur Dateisystemgröße für den Ausführungsspeicher.
- Zeigt Ihnen die Bereitstellungszeiten von Aufgaben an, sodass Sie Bereiche identifizieren können, in denen große Containerlasten die Bereitstellungszeit verlangsamen könnten.
- Das Tool enthält einen Eingabeparameter (Headroom), mit dem Sie die Aggressivität der Optimierungsempfehlungen kontrollieren können.

Die folgenden Abschnitte enthalten spezifische Vorschläge zur Verwendung von Run Analyzer zur Optimierung von Läufen.

Ermitteln Sie die laufenden Kosten

Sie können die folgenden Methoden und Richtlinien verwenden, um die Betriebskosten zu ermitteln:

- Gehen Sie wie folgt vor, um die Gesamtkosten für einen Abrechnungszeitraum anzuzeigen:

  1. Öffnen Sie das Abrechnung und Kostenmanagement Konsole und wähle Rechnungen.
  2. In Gebühren nach Service, erweitern Omics.
  3. Erweitern Sie die Regionund sehen Sie sich dann die Kosten all Ihrer Läufe an, aufgeschlüsselt nach Omics-Instance-Typ, Run-Speichertyp und Ready2Run-Workflow.

- Gehen Sie wie folgt vor, um einen Kostenbericht zu erstellen, der Informationen für jeden Lauf enthält:

  1. Öffnen Sie das Abrechnung und Kostenmanagement Konsole und wähle Datenexporte.

2. Wählen Erstellen um einen neuen Datenexport zu erstellen.
3. Geben Sie ein Namen exportieren für den Datenexport. Behalten Sie die Standardwerte der anderen Felder bei, um einen CUR-Bericht (Kosten und Nutzung) zu erstellen.
4. Für Zeitliche Granularität, wählen Sie stündlich oder täglich.
5. Unter Speichereinstellungen für den Datenexport, führen Sie die folgenden Konfigurationsschritte aus:

   a. Konfigurieren Sie einen Amazon S3-Bucket für den Datenexport.
   b. Für Versionierung von Dateien, wählen Sie aus, ob die vorhandene Exportdatei überschrieben oder jedes Mal eine neue Datei erstellt werden soll.

   Das System generiert den ersten Bericht innerhalb der nächsten 24 Stunden und generiert die nachfolgenden Berichte einmal täglich.
6. Weitere Informationen zur Erstellung des Datenexports finden Sie unter Datenexporte erstellen in der AWS Benutzerhandbuch für Datenexporte.

- Du kannst deine Läufe taggen, um die Kosten nach Kategorien zu überwachen und zu optimieren, z. B. nach Team oder Projekt. Wenn du Stichwörter verwendest, befolge diese Schritte, um die laufenden Kosten nach Tag-Kategorien anzuzeigen:

  1. Öffnen Sie das Abrechnung und Kostenmanagement Konsole und wähle Cost Explorer.
  2. In Berichtsparameter > Gruppieren nach, wähle Tag als Abmessung. und wählen Sie die gewünschte Tag Name.
- Um die Ressourcennutzung für Aufgaben zu sehen, melden Sie sich im Ausführungsmanifest an CloudWatch. Weitere Informationen finden Sie unter Überwachung HealthOmics mit CloudWatch Logs.
- Benutze das Führen Sie Analyzer aus Tool, um Informationen zur Nutzung der Aufgabenressourcen für einen Lauf zu extrahieren.

Ermitteln Sie die Nutzung der Laufzeit

Sie können die folgenden Methoden verwenden, um die Laufzeitnutzung zu untersuchen:

- Aus dem Läuft Auf der Seite der Konsole können Sie die Gesamtlaufzeit eines Laufs einsehen.
- Von der Details ausführen Auf dieser Seite können Sie die folgenden Elemente anzeigen:
  - Zeigt die Gesamtlaufzeit für einen Lauf an.

  - Zeigen Sie die Laufzeit für jede Aufgabe in der Ausführung an.
  - Wählen Sie einen der Links, um die Protokolle in Amazon S3 anzuzeigen oder um die Ausführungsprotokolle oder Ausführungsmanifestprotokolle in einzusehen CloudWatch.
- Aus dem Aufgaben ausführen Liste, wählen Sie Protokolle anzeigen Link für eine Aufgabe, um die Aufgabenprotokolle einzusehen CloudWatch.
- Die Antwort auf die `listRuns` Der API-Vorgang umfasst die Start- und Stoppzeit der Ausführung, sodass Sie die gesamte Laufzeit berechnen können.
- Die Führen Sie Analyzer aus Das Tool zeigt die Dauer von Aufgaben in einer Zeitleistenansicht an. Dieses Tool bietet eine visuelle Darstellung der Reihenfolge der Aufgabenverarbeitung, die Sie der erwarteten Reihenfolge zuordnen können.
- Wie Sie die Dauer der Aufgaben, die Bereitstellungszeit und die Parallelität selbst ableiten können, finden Sie unter Erstellen einer Aufgabenzeitleiste aus dem Run-Manifest.

Erstellen einer Aufgabenzeitleiste aus dem Run-Manifest

Das Ausführungsmanifestprotokoll zeichnet die Erstellungs-, Start- und Stoppzeiten für jede Workflow-Aufgabe in einem Lauf auf. Verwenden Sie sie, um einen Zeitplan für den Lauf zu rekonstruieren. Das Führen Sie Analyzer aus Das Tool meldet bereits die meisten dieser Zahlen. Erstellen Sie die Zeitleiste selbst, wenn Sie sie in Ihrem eigenen Skript oder Dashboard haben möchten oder wenn Sie eine Messung benötigen, die das Tool nicht meldet, z. B. wie viele Aufgaben gleichzeitig ausgeführt wurden.

HealthOmics schreibt das Manifest in ein einzelnes CloudWatch Log-Stream, nachdem der Lauf einen Terminalstatus erreicht hat, und ein Lauf, der mit deaktivierter Protokollierung gestartet wurde, erhält nie einen. Informationen zu den Protokollgruppen- und Stream-Formaten sowie zur Deaktivierung der Protokollierung finden Sie unter Überwachung HealthOmics mit CloudWatch Logs. Den Inhalt des Manifests und ein Beispiel finden Sie unter the section called “Führen Sie Metadaten in aus CloudWatch”.

Jedes JSON-Objekt im Manifest ist ein separates Protokollereignis. Aufgabenobjekte tragen `creationTime`, `startTime`, und `stopTime`. HealthOmics Sets `startTime` wenn die Bereitstellung der Aufgabe abgeschlossen ist, und setzt `stopTime` ob die Aufgabe erfolgreich ist, fehlschlägt oder abgebrochen wurde. Aufgabenobjekte enthalten auch beschreibende Felder, nach denen Sie die Zeitleiste gruppieren können, sowie die Statistiken zur Ressourcenauslastung, die unter beschrieben werden Protokolltypen für HealthOmics Workflows.

HealthOmics meldet alle drei Zeitstempel in UTC, also behandeln Sie einen Zeitstempel ohne UTC-Bezeichner als UTC.

Gehen Sie folgendermaßen vor, um die Bereitstellungszeit, die Aufgabendauer und die Parallelität abzuleiten:

1. Rufen Sie den Manifest-Log-Stream ab. Die Run-UUID ist die `uuid` Feld des GetRun Antwort (siehe the section called “Wichtige Felder in der Antwort”).

   ```
   aws logs get-log-events \
     --log-group-name /aws/omics/WorkflowLog \
     --log-stream-name manifest/run/run-id/run-uuid \
     --start-from-head
   ```

   Das Lesen des gesamten Streams erfordert mehrere Anfragen. Übergeben Sie das `nextForwardToken` aus jeder Antwort als beim nächsten Anruf `--next-token`, behaltend `--start-from-head`, bis eine Antwort das von Ihnen übergebene Token zurückgibt. Das Token kann über einige leere Antworten hinweg weiterlaufen, bevor es wiederholt wird. Hören Sie also nicht bei der ersten leeren Antwort auf. Weitere Informationen finden Sie unter get-log-events in der AWS CLI Befehlsreferenz.
2. Analysieren Sie die von jedem Ereignis `message` Feld als JSON. Ein Aufgabenobjekt hat eine `arn` das enthält `:task/`. Behalte das eine Objekt, dessen `arn` Feld enthält `:run/`: es ist das Run-Objekt, und Schritt 4 verwendet es `stopTime`.
3. Berechnen Sie für jede Aufgabe die beiden Intervalle, aus denen sich die verstrichene Zeit zusammensetzt:

   a. Die Bereitstellungszeit beträgt `startTime` Minus `creationTime`.

   b. Die Laufzeit ist `stopTime` Minus `startTime`.

   Eine Aufgabe, die aus dem Run-Cache bedient wird, hat keine Zeitstempel und hat keinen Platz auf der Timeline. Andernfalls wäre eine Aufgabe ohne `startTime` nie gelaufen, und eine Aufgabe ohne `stopTime` lief noch, als der Lauf gestoppt wurde, also hat er eine Bereitstellungszeit, aber keine Ausführungsdauer. Sortieren Sie die Aufgaben, die eine Ausführungsdauer haben, um die Aufgaben zu finden, die am längsten ausgeführt werden.
4. Messen Sie die Parallelität zwischen den Aufgaben, die eine `startTime`. Baue eine `+1` Übergang bei jedem `startTime` und ein `-1` bei jedem `stopTime`, unter Verwendung der Run-Objekte `stopTime` für eine Aufgabe, die keine hat `stopTime` von sich aus. Sortieren Sie die

Übergänge nach Zeitstempel und führen Sie sie unter Beibehaltung einer laufenden Summe durch. Wenden Sie jeden Übergang an, der einen gemeinsamen Zeitstempel hat, bevor Sie die Summe lesen, sodass eine Aufgabe, die endet, wenn eine andere beginnt, nicht zweimal gezählt wird. Die Gesamtsumme bei jedem Zeitstempel entspricht der Anzahl der gleichzeitig ausgeführten Aufgaben, und der höchste Wert ist die maximale Parallelität des Laufs. Ein Lauf, bei dem keine Aufgabe gestartet wurde, hat keine.

Lange Abschnitte mit geringer Parallelität führen zu einer Abhängigkeitskette in Ihrer Workflow-Definition, die Sie möglicherweise umstrukturieren können. Konstant hohe Bereitstellungszeiten deuten auf die Größe des Container-Images oder auf das Warten auf Rechenkapazität hin. Die Methoden, die beide reduzieren, finden Sie unter Methoden zur Optimierung von Läufen.

Methoden zur Optimierung von Läufen

HealthOmics stellt automatisch Ressourcen bereit, verwaltet und optimiert sie, die das Daten-Staging durchführen (z. B. Datenimporte und Datenexporte). HealthOmics startet und führt auch die Workflow-Engine für Ihren Workflow aus. Sie können jedoch die Startzeiten der Ausführung, die Startzeiten der Aufgaben und die Gesamtlaufzeit der Aufgabe beeinflussen, indem Sie verschiedene Ausführungskonfigurationen festlegen. Ihre allgemeine Herangehensweise an die Workflow-Definition und das Design wirkt sich auch auf die Aufgabenlaufzeit aus. In der folgenden Liste werden Faktoren beschrieben, die sich auf die Ausführung und die Aufgabenleistung auswirken können:

Speichertyp ausführen

Der Laufspeichertyp wirkt sich auf die Ausführungsleistung und die Ausführungsbereitstellungszeit aus. Dynamic führt die Speicherbereitstellung schneller aus, und es geht nie zu wenig Arbeitsspeicher aus, da er dynamisch an Ihre Laufspeicheranforderungen angepasst wird. Dynamic Run Storage eignet sich auch gut für Workflows in der Entwicklung, bei denen Sie häufig einen Workflow starten und beenden müssen, um Probleme zu beheben.

Statischer Laufspeicher erfordert längere Zeiten für die Bereitstellung des Dateisystems, kann aber einige Läufe schneller abschließen, typischerweise, wenn die Läufe eine hohe Anzahl an parallelen Aufgaben aufweisen oder mehr als 9,6 TiB an Dateisystemkapazität benötigen. Statischer Laufspeicher eignet sich gut für lang andauernde Workflows mit hoher I/O Anforderungen.

Um Ihnen bei der Bewertung der Kosten und der Leistung der einzelnen Laufspeichertypen für einen bestimmten Lauf zu helfen, können Sie Folgendes versuchen A/B Testen Sie, um festzustellen, welcher Laufspeichertyp eine bessere Leistung bietet. Erwägen Sie auch,

dynamischen Laufspeicher für Ihre Entwicklungszyklen zu verwenden und dann statischen Laufspeicher für Produktionsläufe in großem Maßstab zu verwenden.

Weitere Informationen zu Run-Storage-Typen finden Sie Speichertypen in HealthOmics Workflows ausführen

Over-provision statischen Speicher ausführen

Wenn die Berechnung Ihrer Workflow-Aufgabe eingeschränkt ist durch I/O, ziehen Sie in Betracht, den statischen Run-Speicher zu überlasten. Die Speicherkosten steigen mit der Größe, aber auch der maximale Durchsatz des Dateisystems steigt. Wenn eine teure Rechenaufgabe ansteht I/O Engpässe: Eine Erhöhung der Dateisystemgröße zur Verkürzung der Task-Laufzeit kann die Gesamtkosten senken.

Reduzieren Sie die Größe der Container-Images

Wenn jede Aufgabe beginnt, HealthOmics lädt den Container, den Sie für die Aufgabe angegeben haben. Das Laden größerer Container dauert länger. Optimieren Sie Ihre Container so, dass sie so klein wie möglich sind, um die Effizienz beim Starten neuer Aufgaben zu verbessern. Wenn Sie Ihren Containern große Datensätze hinzufügen, sollten Sie erwägen, die Datensätze in S3 zu speichern und Ihren Workflow die Daten aus S3 importieren zu lassen. Für die maximalen Containergrößen gilt das HealthOmics unterstützt, siehe HealthOmics feste Größenkontingente für Workflows.

Aufgabengröße

Sie können kleine, aufeinanderfolgende Aufgaben zu einer einzigen Aufgabe kombinieren, um Zeit für die Aufgabenbereitstellung zu sparen. Außerdem HealthOmics hat eine Gebühr für die Mindestdauer einer Aufgabe von einer Minute, sodass das Kombinieren von Aufgaben die Kosten senken kann. Innerhalb der kombinierten Aufgabe können Sie möglicherweise Unix-Pipes verwenden, um Folgendes zu vermeiden I/O Kosten für das Serialisieren und Deserialisieren von Dateien.

Komprimierung von Dateien

Vermeiden Sie eine übermäßige Komprimierung von Workflow-Zwischendateien. Die meisten Genomikformate verwenden die Komprimierung „gzip“ oder „block gzip“. Das Dekomprimieren der Aufgabeneingabedatei und das erneute Komprimieren der Aufgabenausgabedatei können einen großen Prozentsatz der gesamten CPU-Auslastung der Aufgabe beanspruchen. Bei einigen Genomik-Anwendungen können Sie die Komprimierungsstufe beim Serialisieren von Ausgaben festlegen. Indem Sie den Komprimierungsgrad reduzieren, können Sie die CPU-Zeit reduzieren, obwohl größere Dateien den Zeitaufwand für das Schreiben auf die Festplatte erhöhen. Je nach

Aufgabe und Anwendung können Sie die optimale Komprimierungsstufe für Zwischendateien ermitteln, die zu der kürzesten Laufzeit führen. Wir empfehlen, dass Sie sich zunächst auf die Aufgaben mit den größten Ausgabedateien konzentrieren. Eine Komprimierungsstufe von 2 eignet sich für mehrere Szenarien gut. Sie können für Ihren Anwendungsfall mit dieser Stufe beginnen und die Ergebnisse vergleichen, indem Sie andere Komprimierungsstufen ausprobieren.

Anzahl der Threads

Wenn Sie in Ihrer Aufgabendefinition Threads angeben, setzen Sie die Anzahl der Threads auf den gleichen Wert wie die Anzahl der angeforderten vCPUs.

Geben Sie Rechenleistung und Arbeitsspeicher an

Wenn Sie in Ihrer Aufgabe keine Speicher- oder Rechenressourcen angeben, HealthOmics weist den kleinsten Instanztyp zu (`omics.c.large`) als Standard. Deklarieren Sie explizit Ihre Speicher- und Rechenanforderungen, wenn Sie möchten HealthOmics um einen größeren Instanztyp zuzuweisen.

HealthOmics weist die Anzahl der vCPUs, des Speichers und der GPU-Ressourcen zu, die Sie anfordern. Wenn Sie beispielsweise nach 15 vCPUs und 33 GiB fragen, HealthOmics weist Ihrer Aufgabe eine omics.m.4xl-Instanz (16 vCPUs, 64 GB) zu, aber Ihre Aufgabe kann nur 15 vCPUs und 33 GB verwenden. Daher empfehlen wir, dass Sie vCPUs und Speicherressourcen anfordern, die einer Omics-Instanz entsprechen.

Verbessern Sie die Verfügbarkeit der GPU-Instanz

Wenn die Ausführung Ihres GPU-Workflows fehlschlägt mit INSTANCE_RESERVATION_FAILED oder ALL_PROFILES_INSTANCE_RESERVATION_FAILED, Sie können verschiedene bewährte Methoden anwenden, um den Erfolg der GPU-Akquisition zu verbessern. Weitere Informationen zu Accelerator-Paketen, regionaler Flexibilität und der Reihenfolge der Ressourcenausfälle finden Sie unter Behebung von INSTANCE_RESERVATION_FAILED-Fehlern.

Stapeln Sie mehrere Proben in einem Durchlauf zusammen

Da die Bereitstellung des Dateisystems zu Beginn des Laufs einige Zeit in Anspruch nimmt, können Sie Bereitstellungszeit sparen, indem Sie mehrere Proben in einem Lauf zusammenfassen. Berücksichtigen Sie die folgenden Faktoren, bevor Sie sich für diesen Ansatz entscheiden:

- Eine einzelne fehlerhafte Probe kann dazu führen, dass ein Arbeitsablauf fehlschlägt, sodass das Stapeln von Proben die Anzahl der fehlgeschlagenen Workflows erhöhen könnte. Wenn Sie nicht sicher sind, ob Ihr Workflow die meiste Zeit erfolgreich sein wird, könnte ein Durchlauf pro Probe der bessere Ansatz sein.

- HealthOmics weist ein Run-Speicherdateisystem für den gesamten Workflow zu. Stellen Sie bei einem Stapel von Proben sicher, dass Sie eine ausreichend große Menge an Durchlaufspeicher angeben, um alle Proben verarbeiten zu können.
- Pro Arbeitsablauf gibt es eine maximale Menge an Durchlaufspeicher, sodass die Anzahl der Proben, die Sie dem Stapel hinzufügen können, möglicherweise begrenzt ist.
- Die Mindestgröße des Laufspeichers beträgt 1,2 TiB. Durch die Stapelverarbeitung können die Kosten gesenkt werden, wenn der Workflow viel weniger Speicherplatz als das Minimum für jede Probe benötigt.
- Run-Storage kann mehrere gleichzeitige Verbindungen verarbeiten. Wenn also mehrere Tasks denselben Run-Storage verwenden, sollte das nicht zu folgenden Problemen führen I/O Engpässe.
- Jeder Lauf hat seinen eigenen Satz von Tags. Wenn Sie Workflows mit Informationen für die Budgetierung oder Nachverfolgung kennzeichnen, ist es möglicherweise besser, separate Läufe zu verwenden.
- IAM-Rollen gelten für den gesamten Lauf. Jeder Benutzer hat Zugriff auf alle Daten für einen Stapel von Proben. Durch separate Workflows haben Sie die Möglichkeit, detailliertere Berechtigungen zu verwenden.
- HealthOmics legt Kontingente auf Kontoebene für die maximale Anzahl gleichzeitiger Workflows und die maximale Anzahl gleichzeitiger Aufgaben in einem Workflow fest. Informationen darüber, wie Sie eine Erhöhung dieser Kontingente beantragen können, finden Sie unter HealthOmics Servicekontingente.

Verwenden Sie Parameter für Container-Bilder

Parametrisieren Sie Ihre Container-Images, anstatt ihre URIs in den Workflow einzubetten. Wenn es sich um Ausführungsparameter handelt, HealthOmics überprüft, ob der Lauf Zugriff auf Ihre Container hat, bevor der Lauf beginnt. Andernfalls schlägt die Aufgabe während des Laufs fehl, wenn Ihnen Gebühren für alle abgeschlossenen Aufgaben entstanden sind. Außerdem, weil es sich um parametrisierte Eingaben handelt, HealthOmics generiert eine Prüfsumme im Run-Manifest, wodurch die Herkunft der Ausführung verbessert wird.

Verwenden Sie einen Linter

Verwenden Sie einen Linter, um häufig auftretende Workflow-Fehler zu finden, bevor Sie einen neuen Workflow ausführen. Weitere Informationen finden Sie unter Der Arbeitsablauf zeigt an HealthOmics.

Verwenden EventBridge um Probleme zu melden

Benutzen EventBridge benutzerdefinierte Benachrichtigungen, um Anomalien zu erkennen, die für Ihre Geschäftslogik spezifisch sind.

Verwenden Sie Sequenzspeicher

Erwägen Sie die Verwendung eines Sequenzspeichers für Ihre Quelldaten, um Speicherkosten zu sparen. Weitere Informationen finden Sie in der Speichern Sie Omics-Daten in jeder Größenordnung kostengünstig mit HealthOmics Blogbeitrag.

Auswirkung der Dateigrößenabweichung zwischen Durchläufen

Benutzer entwerfen und testen Läufe häufig anhand einer kleinen Menge von Testdaten und stoßen dann bei Produktionsläufen auf eine Vielzahl von Daten mit erheblichen Abweichungen in der Dateigröße. Stellen Sie sicher, dass Sie diese Abweichung berücksichtigen, wenn Sie den Testlauf optimieren.

In der folgenden Liste werden Optimierungsempfehlungen beschrieben, wenn die Dateigrößen erheblich voneinander abweichen:

Variieren Sie die Dateigrößen in Ihren Testdaten

Versuchen Sie, während der Entwicklung Testdaten zu verwenden, die eine repräsentative Varianz aufweisen.

Verwenden Sie Run Analyzer

Verwenden Sie das Tool Run Analyzer für eine Vielzahl von Stichproben, um Abweichungen bei den Datengrößen zu berücksichtigen.

Sie können den Run Analyzer verwenden, um die Varianz zwischen den Durchläufen in Ihren Produktionsdatenproben zu verstehen. Verwenden `--batch` Modus in Run Analyzer, um Statistiken für einen Stapel von Durchläufen zu generieren und die maximale Rechenleistung zu analysieren, die zur Behandlung von Ausreißern in Ihren Datensätzen erforderlich ist.

Beispielsweise können Sie Run Analyzer im Batch-Modus eine vollständige Flusszelle mit Daten zuweisen, um die maximale vCPU- und Speicherauslastung für die gesamte Flusszelle zu ermitteln.

Reduzieren Sie die Größenvarianz der Eingabedatensätze

Wenn Sie eine hohe Varianz bei den Stichprobenumfängen feststellen, können Sie die Stichproben vor HealthOmics und wählen Sie für jeden Stapel unterschiedliche Dateisystemgrößen aus, um Speicherkosten zu sparen.

Verwenden Sie in WDL die `size` Funktion zur Aufteilung der Ressourcenzuweisung für einzelne Aufgaben zwischen großen und kleinen Stichproben. Wenden Sie diese Strategie auf Ihre teuersten Aufgaben an, um die größte Wirkung zu erzielen.

Verwenden Sie in Nextflow bedingte Ressourcen für die Stufenzuweisung von Ressourcen auf der Grundlage der Dateigröße oder des Dateinamens. Weitere Informationen finden Sie unter Bedingte Prozessressourcen auf dem Nextflow GitHub Seite.

Optimiere nicht zu früh

Finalisieren Sie Ihren Workflow-Code und Ihre Logik, bevor Sie in umfangreiche Maßnahmen zur Leistungsoptimierung investieren. Eine Änderung Ihres Codes kann erhebliche Auswirkungen auf die erforderlichen Ressourcen haben. Wenn Sie einen Lauf zu früh im Entwicklungsprozess optimieren, überoptimieren Sie möglicherweise zu stark oder Sie müssen möglicherweise erneut optimieren, wenn sich die Workflow-Definition später ändert.

Re-run das Tool Run Analyzer regelmäßig ausführen

Wenn Sie im Laufe der Zeit Änderungen an Ihrer Workflow-Definition vornehmen oder wenn sich Ihre Stichprobenvarianz ändert, führen Sie das Tool Run Analyzer regelmäßig aus, damit Sie weitere Optimierungen vornehmen können.

Methoden zur Optimierung der Parallelität von Ressourcen

HealthOmics bietet die folgenden Funktionen, mit denen Sie die Kosten kontrollieren und verwalten können, wenn die Verarbeitung in großem Maßstab erfolgt:

- Verwenden Sie Ausführungsgruppen, um Ihre Kosten und Ihren Ressourcenverbrauch zu kontrollieren. Sie können in der Ausführungsgruppe Höchstwerte für die Anzahl gleichzeitiger Läufe, vCPUs, GPUs und die Gesamtlaufzeit pro Aufgabe festlegen. Wenn separate Teams oder Gruppen dasselbe Konto verwenden, können Sie für jedes Team eine separate Ausführungsgruppe erstellen. Du kannst den Ressourcenverbrauch und die Kosten pro Team kontrollieren, indem du die Höchstwerte für die Laufgruppe konfigurierst. Weitere Informationen finden Sie unter HealthOmics Run-Gruppen verwenden.

- Während der Entwicklung können Sie eine separate Ausführungsgruppe mit niedrigeren Höchstwerten konfigurieren, um außer Kontrolle geratene Aufgaben abzufangen.
- Servicekontingente tragen auch dazu bei, Ihr Konto vor übermäßigen Ressourcenanforderungen zu schützen. Informationen zu Servicekontingenten und dazu, wie Sie eine Erhöhung des Kontingentwerts beantragen können, finden Sie unter HealthOmics Servicekontingente

Operationen ausführen in HealthOmics

Sie können einen Lauf starten, erneut ausführen, klonen, abbrechen, löschen oder Informationen zu einem Lauf abrufen:

- Start— HealthOmics erstellt einen neuen Lauf mit den von Ihnen angegebenen Konfigurationseinstellungen und startet dann den Lauf.
- Get run information— Sie können den Status, die Konfiguration und die Details auf Aufgabenebene jedes ausgeführten Laufs mithilfe der get-run API oder der HealthOmics Konsole abrufen. Sie können auch alle Läufe auflisten, die Aufgaben innerhalb eines Laufs auflisten und auf die Ausführungsmetadaten in CloudWatch Logs zugreifen.
- Rerun— HealthOmics erstellt einen neuen Lauf, der ein Duplikat des von Ihnen angegebenen Laufs ist. Sie können einen gelöschten Lauf mit dem HealthOmics rerun Tool erneut ausführen.
- Clone— Sie können einen vorhandenen Lauf mithilfe der Konsole klonen. Die Konsole öffnet die Seite „Clone-Run“ und füllt die Konfigurationsfelder anhand der Werte aus dem vorhandenen Lauf vorab aus. Sie können die Werte nach Bedarf ändern und den geklonten Lauf starten.
- Cancel— Sie können einen Lauf abbrechen, der noch nicht abgeschlossen ist. Wenn Sie einen Lauf abbrechen, HealthOmics werden keine der Laufausgaben gespeichert.
- Delete— Sie können abgeschlossene Läufe manuell löschen oder den Aufbewahrungsmodus so einstellen, HealthOmics dass die ältesten Läufe automatisch gelöscht werden. Weitere Informationen zum Aufbewahrungsmodus finden Sie unterthe section called “Führen Sie den Aufbewahrungsmodus aus”.

Themen

- Starte einen Lauf in HealthOmics
- Holen Sie sich Informationen zum Lauf
- Führen Sie einen Run in erneut aus HealthOmics
- Einen Run in klonen HealthOmics

- Einen Run in stornieren HealthOmics
- Lösche einen Run in HealthOmics

Starte einen Lauf in HealthOmics

Wenn Sie einen Lauf starten, geben Sie die Ressourcen an HealthOmics weist für den Lauf zu. Die folgenden Einstellungen sind verfügbar:

A. Ort der Ausgabe — Geben Sie eine Amazon S3-URI an, in der die Ausgabedateien des Laufs gespeichert werden. Wenn Sie eine große Anzahl von Workflows gleichzeitig ausführen, verwenden Sie für jeden Workflow separate Amazon S3-Ausgabe-URIs, um eine Bucket-Drosselung zu vermeiden.

   Es gelten die folgenden Regeln für `outputUri`:

   - `outputUri` (`--output-uri`) ist erforderlich und muss eine gültige Amazon S3-URI sein. Maximal 750 Zeichen.
   - Informationen zu den Region-Anforderungen für Eingabe- und Ausgabespeicherorte finden Sie unter HealthOmics Anforderungen an die Workflow-Definition.
   - Die Servicerolle in `roleArn` (`--role-arn`) muss die Erlaubnis haben, in den Bucket zu schreiben.
   - Ein abschließender Schrägstrich ist optional. HealthOmics Leckereien `s3://amzn-s3-demo-bucket/output` und `s3://amzn-s3-demo-bucket/output/` hat den gleichen Ausgabeort.

   Weitere Informationen finden Sie unter Objekte mithilfe von Präfixen organisieren in der Amazon S3-Benutzerhandbuch und Skalieren Sie Speicherverbindungen horizontal in der Optimierung der Amazon S3-Leistung Whitepaper.

B. Rolle im Dienst — Geben Sie eine IAM-Servicerolle an, die Folgendes gewährt HealthOmics Berechtigungen für den Zugriff auf die Ressourcen, die für den Lauf benötigt werden. Optional kann die Konsole die Servicerolle für Sie erstellen. Sie können auch Sitzungsrichtlinien verwenden, um die Berechtigungen für einzelne Läufe weiter einzuschränken. Weitere Informationen finden Sie unter Servicerollen für AWS HealthOmics.

C. Führen Sie den Speicher aus (optional, standardmäßig dynamisch) — Geben Sie den Laufspeichertyp und die Speichermenge an (für statischen Speicher). Um Datenisolierung und Sicherheit zu gewährleisten, HealthOmics stellt den Speicher zu Beginn jedes Laufs bereit und stellt ihn am Ende des Laufs wieder frei. Weitere Informationen finden Sie unter Speichertypen in HealthOmics Workflows ausführen.

D. Priorität ausführen (optional) — Weisen Sie dem Lauf eine Priorität zu. Wie sich die Priorität auf den Lauf auswirkt, hängt davon ab, ob der Lauf einer Ausführungsgruppe zugeordnet ist. Weitere Informationen finden Sie unter Priorität ausführen.

E. Workflow-Version (optional) — Wählen Sie eine bestimmte Workflow-Version für den Lauf aus. Wenn Sie keine Version angeben, HealthOmics startet den Standard-Workflow-Version.

F. Einstellungen der Nextflow-Engine (optional, nur Nextflow) — Geben Sie Engine-Einstellungen wie Version und Syntaxparser für Nextflow-Workflows während der Laufzeit an. Weitere Informationen finden Sie unter Geben Sie die Nextflow-Engine-Einstellungen an.

G. Eingabeparameter (optional) — Geben Sie die Workflow-Eingabeparameter als JSON-Datei oder als Inline-Werte an. Erforderliche Parameter werden in der Parametervorlage des Workflows definiert. Weitere Informationen finden Sie unter HealthOmics Eingaben ausführen.

H. Anforderungs-ID (optional, nur API und CLI) — Geben Sie für jeden Lauf eine eindeutige Anforderungs-ID an. Die Anforderungs-ID ist ein Idempotenz-Token, das HealthOmics wird verwendet, um doppelte Anfragen zu identifizieren und den Lauf nur einmal zu starten.

Themen

- Einen Lauf mit der Konsole starten
- Einen Lauf mit der API starten
- Geben Sie die Nextflow-Engine-Einstellungen an
- VPC-Netzwerk
- Laufmetriken aktivieren

Einen Lauf mit der Konsole starten

Der Start-Run-Assistent besteht aus vier Schritten:

1. Geben Sie die Ausführungsdetails an
2. Fügen Sie Parameterwerte hinzu
3. Fügen Sie Run-Gruppe, Run-Cache und Tags hinzu
4. Überprüfen und starten Sie den Lauf

Um einen Lauf zu starten

i. Öffnen Sie die HealthOmics -Konsole.

ii. Öffnen Sie bei Bedarf den linken Navigationsbereich (≡). Wählen Sie Läuft.

iii. Wählen Starte den Lauf.

Schritt 1: Geben Sie die Laufdetails an

Geben Sie die folgenden Einstellungen an:

#	Einstellung	Erforderlich	Description
1	Wählen Sie die Workflow-Quelle	Erforderlich	Wählen Sie Eigener Arbeitsablauf (private Workflows , die Sie besitzen) oder Gemeinsamer Arbeitsablauf (Workflows, die mit Ihnen geteilt wurden).
2	Workflow-ID	Erforderlich	Wählen Sie die Workflow-ID für diesen Lauf aus.
3	Name der Ausführung	Erforderlich (optional auf API und CLI)	Ein beschreibender Name für diesen Lauf. Maximal 127 Zeichen. Eine Run-ID wird automatisch generiert, sobald der Workflow ausgeführt wird.
4	Konfiguration	Optional	Wählen Sie eine Konfiguration, um VPC-Einst ellungen (Subnetze, Sicherheitsgruppen) für die Internetverbindung anzugeben und Container-Registri erungszuordnungen und Git-Repository-Verbindungen einzubeziehen. Kann nach dem Start des Laufs nicht geändert werden.
5	Priorität ausführen	Optional	Legt die Priorität eines Laufs in einer Ausführun gsgruppe fest. Eine höhere Zahl bedeutet eine höhere Priorität. Nur ganze Zahlen im Bereich 0—1.000.
6	Speichertyp ausführen	Optional	Wählen Dynamischer Speicher (Standard, empfohlen) oder Statischer Speicher. Dynamischer Speicher skaliert pro Aufgabe nach oben und unten. Statische r Speicher stellt einen festen Betrag zur Verfügung. Weitere Informationen finden Sie unter Speichertypen in HealthOmics Workflows ausführen.

#	Einstellung	Erforderlich	Description
7	Speicherkapazität ausführen	Bedingt	Nur für statischen Speicher. Geben Sie den Betrag in GiB an.
8	Wählen Sie das S3-Ausgabeziel	Erforderlich	Der Amazon S3-Standort, an dem die ausgeführten Ausgaben geliefert werden. Format: `s3://bucket/prefix/object` .
9	Konto-ID des Bucket-Besitzers ausgeben	Optional	Wenn Ihr Konto nicht Eigentümer des Ausgabe-Buckets ist, geben Sie die des Bucket-Besitzers ein AWS-Konto ID.
10	Führen Sie den Metadaten-Aufbewahrungsmodus aus	Optional	Wählen Sie Metadaten für den Lauf beibehalten (Standard) oder Löscht automatisch das älteste. `RETAIN` ist der Standardwert; in diesem Modus HealthOmics löscht die Run-Metadaten nicht. Wenn du wählst `REMOVE`, HealthOmics löscht dauerhaft die ältesten Run-Metadaten, wenn die Anzahl der Durchläufe den Höchstwert erreicht. Weitere Informationen finden Sie unter Retentionsmodus für HealthOmics Läufe ausführen.
11	Netzwerkzugriff	Optional	Wählen Sie Eingeschränkt (Standard) oder Virtuelle private Cloud (VPC). Weitere Informationen finden Sie unter VPC-Netzwerk.
12	Rolle im Dienst	Erforderlich	Wählen Sie eine vorhandene Servicerolle oder erstellen Sie eine neue. HealthOmics erfordert Berechtigungen für Amazon S3 und KMS. Weitere Informationen finden Sie unter Servicerollen für AWS HealthOmics.

Wählen Sie Weiter um mit Schritt 2 fortzufahren.

Schritt 2: Parameterwerte hinzufügen

Geben Sie auf dieser Seite die Parameterwerte für den Lauf ein, oder wählen Sie vordefinierte Werte aus, die vom Workflow-Autor bereitgestellt wurden.

Wenn Sie einen Nextflow-Workflow auswählen, um den Lauf zu starten, werden zusätzliche Abschnitte für Einstellungen der Nextflow-Engine und Nextflow-Profile erscheinen oben in diesem Schritt. Vollständige Informationen zu diesen Einstellungen (unterstützte Werte, Verhalten und Profilpriorität) finden Sie unter Geben Sie die Nextflow-Engine-Einstellungen an.

Parameterwerte ausführen

Geben Sie die Ausführungsparameter an. Sie können eine JSON-Datei hochladen oder die Werte manuell eingeben. Die JSON-Datei enthält den genauen Namen jedes Eingabeparameters und einen Wert für den Parameter.

HealthOmics unterstützt die folgenden JSON-Typen für Parameterwerte.

JSON-Typ	Beispiel für Schlüssel und Wert	Hinweise
boolesch	„b“: wahr	Der Wert steht nicht in Anführungszeichen und wird ausschließlich in Kleinbuch staben geschrieben.
Ganzzahl	„i“ :7	Der Wert steht nicht in Anführungszeichen.
number	„f“ :42,3	Der Wert steht nicht in Anführungszeichen.
Zeichenfolge	„s“: „Zeichen“	Der Wert steht in Anführung szeichen. Verwenden Sie den String-Typ für Textwerte und URIs. Das URI-Ziel muss dem erwarteten Eingabetyp entsprechen.

JSON-Typ	Beispiel für Schlüssel und Wert	Hinweise
Array	„a“: [1,2,3]	Der Wert steht nicht in Anführungszeichen. Array-Mitglieder müssen jeweils den durch den Eingabeparameter definierten Typ haben.
object	„o“: {"left“ :"a“, „right“ :1}	In WDL wird das Objekt WDL Pair, Map oder Struct zugeordnet

Weitere Informationen erhalten Sie unter HealthOmics Eingaben ausführen und Verwaltung der Größe von Ausführungsparametern.

Wählen Sie Weiter um mit Schritt 3 fortzufahren.

Schritt 3: Run-Gruppe, Run-Cache und Tags hinzufügen

Alle Einstellungen auf dieser Seite sind optional.

#	Einstellung	Erforderlich	Description
1	Gruppe ausführen	Optional	Wählen Sie eine vorhandene Ausführungsgruppe aus oder erstellen Sie eine neue. Run-Gruppen bündeln Sie Läufe nach Kategorie und Priorität und legen Sie die maximale vCPUs und die maximale Laufzeit fest. Weitere Informationen finden Sie unter HealthOmics Run-Gruppen verwenden.
2	Cache ausführen	Optional	Verwenden Sie einen Run-Cache, um die Ergebnisse abgeschlossener Aufgaben wiederzuverwenden, anstatt sie erneut zu berechnen. Weitere Informationen finden Sie unter Konfiguration eines Laufs mit Run-Cache über die Konsole.

#	Einstellung	Erforderlich	Description
3	Tags	Optional	Fügen Sie bis zu 50 Schlüsselwert-Tags für die Suche, Filterung und Kostenverfolgung hinzu.

Wählen Sie Weiter um mit Schritt 4 fortzufahren.

Schritt 4: Überprüfen und den Lauf starten

Überprüfen Sie die Ausführungskonfiguration aus allen vorherigen Schritten. Um eine Einstellung zu ändern, wählen Sie Bearbeiten neben dem entsprechenden Schritt. Wenn Sie bereit sind, wählen Sie Starte den Lauf.

Einen Lauf mit der API starten

Benutze das StartRun API-Vorgang, um einen Lauf zu erstellen und zu starten.

Starte einen einfachen Lauf

Im folgenden Beispiel werden die Workflow-ID, die Servicerolle und der Ausgabe-URI angegeben. In diesem Beispiel wird der Aufbewahrungsmodus auf festgelegt `REMOVE`. Weitere Informationen zum Aufbewahrungsmodus finden Sie unter Retentionsmodus für HealthOmics Läufe ausführen.

```
aws omics start-run \
     --workflow-id workflow id \
     --role-arn arn:aws:iam::123456789012:role/OmicsRole \
     --output-uri s3://amzn-s3-demo-bucket/output \
     --name "my-workflow-run" \
     --retention-mode REMOVE
```

Als Antwort erhalten Sie die folgende Ausgabe. Das `uuid` ist einzigartig für den Lauf, und zusammen mit `outputUri` kann verwendet werden, um zu verfolgen, wo Ausgabedaten geschrieben werden.

```
{
    "arn": "arn:aws:omics:us-west-2:123456789012:run/1234567",
    "id": "123456789",
    "uuid": "96c57683-74bf-9d6d-ae7e-f09b097db14a",
    "outputUri": "s3://bucket/folder/8405154/96c57683-74bf-9d6d-ae7e-f09b097db14a",
    "status": "PENDING"
```

```
}
```

Allgemeine API-Optionen

Fügen Sie eine Parameterdatei hinzu

Wenn die Parametervorlage für einen Workflow alle erforderlichen Parameter deklariert, können Sie eine lokale JSON-Datei der Eingaben bereitstellen, wenn Sie eine Workflow-Ausführung starten. Die JSON-Datei enthält den genauen Namen jedes Eingabeparameters und einen Wert für den Parameter.

Verweisen Sie auf die JSON-Eingabedatei in der AWS CLI durch Hinzufügen `--parameters file://<input_file.json>` zu deinem `start-run` Anfrage. Weitere Informationen finden Sie unter Unterstützte JSON-Typen für Parameterwerte in Schritt 2: Parameterwerte hinzufügen und HealthOmics Eingaben ausführen.

Geben Sie eine Anforderungs-ID an

Sie können eine eindeutige angeben `requestId` für jeden Lauf. Die Anforderungs-ID ist ein Idempotenz-Token, das HealthOmics wird verwendet, um doppelte Anfragen abzufangen. Es wird kein Lauf gestartet, wenn die Anforderungs-ID ein Duplikat einer vorherigen Ausführung ist.

Wenn Sie Infrastruktur (wie Lambda-Funktionen oder Schrittfunktionen) für die Orchestrierung von Laufstarts verwenden, empfiehlt es sich, für jeden Lauf eine eindeutige Anforderungs-ID anzugeben StartRun Anfrage. Dadurch wird sichergestellt, dass, wenn Ihre Infrastruktur versehentlich einen bereits gestarteten Lauf startet, HealthOmics startet den doppelten Lauf nicht.

```
aws omics start-run \
     --workflow-id workflow id \
     ... \
     --request-id "unique-request-id-12345"
```

Wählen Sie eine Workflow-Version

Sie können eine Workflow-Version für den Lauf angeben. Wenn Sie keine Version angeben, HealthOmics startet den Lauf mit der Standard-Workflow-Version.

```
aws omics start-run \
     --workflow-id workflow id \
```

```
    ... \
    --workflow-version-name '1.2.1'
```

Überschreibt den Laufspeichertyp

Sie können den Standard-Laufspeichertyp, der im Workflow festgelegt wurde, überschreiben.

```
aws omics start-run \
    --workflow-id workflow id \
    ... \
    --storage-type STATIC \
    --storage-capacity 2400
```

Ephemeren Speicher aktivieren

Um den temporären Speicher für einen Lauf zu aktivieren, setzen Sie `--scratch-storage-mode` zu `LOCAL` wenn du den Lauf startest. HealthOmics mountet ein dediziertes lokales Speichervolume unter `/tmp` für jede Workflow-Aufgabeninstanz.

```
aws omics start-run \
    --workflow-id workflow-id \
    --role-arn arn:aws:iam::123456789012:role/OmicsServiceRole \
    --output-uri s3://amzn-s3-demo-bucket/output-folder/ \
    --parameters file:///path/to/parameters.json \
    --scratch-storage-mode LOCAL
```

Um den temporären Speicher für einen bestimmten Lauf zu deaktivieren (z. B. um einen Fehler zu isolieren), setzen Sie `--scratch-storage-mode` zu `SHARED`.

Für die Werte, die `scratchStorageMode` akzeptiert, die Standardeinstellung, die gilt, wenn Sie den Parameter weglassen, und wie sich die Einstellung auf CPU- und GPU-Aufgaben auswirkt, finden Sie unter Kurzlebiger Speicher für Workflow-Aufgaben HealthOmics.

Informationen zu den Nextflow-Engine-Einstellungen (Engine-Version, Profile, Syntaxparser) finden Sie unter Geben Sie die Nextflow-Engine-Einstellungen an.

Geben Sie die Nextflow-Engine-Einstellungen an

Für Nextflow-Workflows können Sie eine übergeben `engineSettings` Karte in der StartRun API-Anfrage zur Steuerung des Engine-Verhaltens, ohne Ihren Workflow-Quellcode zu ändern. Auf der

Konsole sind diese Einstellungen verfügbar in Schritt 2: Parameterwerte hinzufügen als separater Abschnitt, der für Nextflow-Workflows angezeigt wird.

> **Note**
>
> Die Engine-Einstellungen gelten nur für Nextflow-Workflows. Wenn Sie Engine-Einstellungen für WDL- oder CWL-Workflows in der API oder CLI angeben, werden sie stillschweigend ignoriert und sind nicht verfügbar in GetRun Antworten.

Unterstützte Schlüssel

#	Key (Schlüssel)	Zulässige Werte	Behavior	Versionsunterstützung
1	`engineVersion`	`"22.04.0"` (oder `"22.04"`), `"23.10.0"` (oder `"23.10"`), `"24.10.8"` (oder `"24.10"`), `"25.10.0"` (oder `"25.10"`), `"26.04.0"` (oder `"26.04"`)	Pintet die Nextflow-Version für den Lauf. Überschreibt die Version, die von erkannt wurde `nextflow.config` . Sowohl die Voll- als auch die Kurzversion werden akzeptiert.	Alle Nextflow-Versionen
2	`syntaxVersion`	`"v1"` (Legacy-Parser), `"v2"` (Strenger Syntaxparser)	Wählt den Syntaxparser aus.	v26.04 und höher. Frühere Versionen unterstützen nur `"v1"`.
3	`outputFormat`	`"json"`, `"text"`, `"none"`	Stellt den Motor ein stdout/stderr Zusammenfassungsformat.	v26.04 und höher (in früheren Versionen ignoriert).
4	`agentMode`	`"true"`, `"false"`	Steuert den Nextflow-Agentenmodus.	v26.04 und höher (in früheren

#	Key (Schlüssel)	Zulässige Werte	Behavior	Versionsunterstützung
				Versionen ignoriert).
5	`profile`	Comma-separated Profilnamen, zum Beispiel `"test,docker"`	Aktiviert Nextflow-Profile. Weitere Informationen finden Sie unter Nextflow-Profile.	Alle Nextflow-Versionen

Pinning der Nextflow-Engine-Version

Sie können eine bestimmte Nextflow-Engine-Version auswählen, um Ihren Workflow auszuführen, was eine kontrollierte Migration zwischen Engine-Versionen ermöglicht. Die Überschreibung der Laufzeitversion stellt sicher, dass selbst dann, wenn eine Workflow-Definition über ihre Konfiguration oder ihr Profil eine Version spezifiziert, die Engine-Version, die Sie zur Laufzeit angeben, Vorrang hat. Auf diese Weise können Sie denselben Workflow über mehrere Engine-Versionen hinweg testen, ohne den Workflow-Quellcode zu ändern.

Unterstützte Versionen: 22.04.0 (oder 22.04), 23.10.0 (oder 23.10), 24.10.8 (oder 24.10), 25.10.0 (oder 25.10) und 26.04.0 (oder 26.04). Sowohl die Voll- als auch die Kurzversion werden akzeptiert.

API und CLI

Benutze das `engineVersion` geben Sie ein `--engine-settings`:

```
aws omics start-run \
    --workflow-id workflow id \
    --role-arn arn:aws:iam::123456789012:role/OmicsRole \
    --output-uri s3://amzn-s3-demo-bucket/output \
    --engine-settings '{"engineVersion":"26.04"}'
```

Konsole

Im Schritt 2: Parameterwerte hinzufügen, wählen Sie die Version aus Version der Engine Dropdown im Abschnitt mit den Nextflow-Engine-Einstellungen.

Nextflow-Profile

Nextflow-Profile sind benannte Sätze von Ausführungseinstellungen, die in Ihrem Workflow definiert sind `nextflow.config`. Sie können zur Laufzeit ein oder mehrere Profile aktivieren, um umgebungsspezifische Einstellungen (z. B. Entwicklung, Test oder Produktion) anzuwenden, ohne den Workflow-Quellcode zu ändern.

API und CLI

Benutze das `profile` geben Sie ein `--engine-settings`. Geben Sie mehrere Profile als kommagetrennte Liste an:

```
aws omics start-run \
    --workflow-id workflow id \
    --role-arn arn:aws:iam::123456789012:role/OmicsRole \
    --output-uri s3://amzn-s3-demo-bucket/output \
    --engine-settings '{"profile":"test,docker"}'
```

Konsole

In Schritt 2: Parameterwerte hinzufügen, der Nextflow-Profile Ein Abschnitt erscheint unter den Motoreinstellungen. Wählen Sie ein oder mehrere Profile aus dem Wählen Sie Profile aus Dropdown. In der Konsole wird die Reihenfolge der Profilanwendung angezeigt. Um ein Profil zu entfernen, wählen Sie × neben seinem Namen.

Reihenfolge der Profilanwendung

Wenn Sie mehrere Profile angeben, hängt die Reihenfolge der Anwendungen von der Nextflow-Engine-Version und dem Syntaxparser ab:

- Nextflow v26.04 und höher mit dem strikten Syntaxparser (v2) — Profile werden in der Reihenfolge angewendet, die in der `profile` Wert (von links nach rechts) oder in Ihrer Auswahlreihenfolge auf der Konsole. Spätere Profile überschreiben frühere Profile aufgrund widersprüchlicher Einstellungen.
- Nextflow-Versionen vor v26.04 und v26.04 und höher mit dem Legacy-Parser (v1) — Profile werden in der Reihenfolge angewendet, die in der `nextflow.config` Datei, unabhängig von der in der API-Anfrage oder der Konsolenauswahl angegebenen Reihenfolge.

Vorrang der Konfiguration

Nextflow löst Parameter in der folgenden Reihenfolge auf, wobei spätere Ebenen frühere überschreiben:

1. `nextflow.config` Standardwerte
2. Ausgewählte Nextflow-Profile
3. Datei mit vordefinierten Werten
4. Run-time Parameter-Überschreibungen

> **Note**
>
> Empfehlung — Um ein konsistentes Verhalten der Profilanwendung bei allen Läufen zu gewährleisten, pinnen Sie die Nextflow-Engine-Version mithilfe der `engineVersion` geben Sie die API ein oder Version der Engine auf der Konsole.

VPC-Netzwerk

Sie können Workflows in Ihrer Virtual Private Cloud (VPC) ausführen, was den Zugriff auf das öffentliche Internet, den regionsübergreifenden Datenverkehr und die Kommunikation mit Ressourcen in Ihrer VPC ermöglicht.

So aktivieren Sie das VPC-Netzwerk:

1. Erstellen Sie eine Konfiguration, die VPC-Subnetze und Sicherheitsgruppen angibt.
2. Wenn Sie einen Lauf mit der Konsole starten, legen Sie Folgendes fest Zugriff auf das Netzwerk zu Virtuelle private Cloud (VPC) und wählen Sie Ihre Konfiguration in Schritt 1 aus.
3. Wenn Sie einen Lauf mit der API starten, verwenden Sie `--networking-mode VPC` und referenzieren Sie Ihre Konfiguration mit `--configuration-name`.

Für die Werte, die `networkingMode` (`--networking-mode`) akzeptiert und ist die Standardeinstellung, siehe Netzwerkmodi. Für die `configurationName` (`--configuration-name`) Parameter, siehe Einen Lauf mit VPC-Netzwerk starten.

Folgendes StartRun Regeln gelten für `configurationName`:

- Für ein `VPC` lauf, `configurationName` ist erforderlich und die Konfiguration muss Folgendes definieren `vpcConfig`, oder StartRun scheitert.

- Wenn Sie liefern `configurationName` In jedem Netzwerkmodus muss die Konfiguration den Status haben `ACTIVE`, oder StartRun scheitert. Informationen zum Konfigurationsstatus finden Sie unter CreateConfiguration. Informationen zur Problembehandlung finden Sie unter Die Ausführung kann mit dem VPC-Netzwerk nicht gestartet werden.

Laufmetriken aktivieren

Anhand von Metriken wird die Ressourcenauslastung für einen Lauf nahezu in Echtzeit gemeldet. Um Run-Metriken zu aktivieren, muss die Servicerolle, die Sie für den Lauf verwenden, über Folgendes verfügen `cloudwatch:PutMetricData` Erlaubnis. Weitere Informationen finden Sie unter Metriken für private Workflows ausführen.

Fügen Sie der Servicerolle, die Sie für die Ausführung verwenden, die folgende Berechtigung hinzu.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "cloudwatch:PutMetricData",
      "Resource": "*"
    }
  ]
}
```

Holen Sie sich Informationen zum Lauf

Nachdem Sie einen Lauf gestartet haben, können Sie dessen Status, Konfiguration und Details auf Aufgabenebene mithilfe der HealthOmics API. Auf dieser Seite wird beschrieben, wie Sie:

Themen

- Laufstatus und Details abrufen
- Listet alle Läufe auf
- Aufgaben in einem Lauf auflisten
- Rufen Sie die Aufgabendetails ab
- Führen Sie Metadaten in aus CloudWatch

Laufstatus und Details abrufen

Benutze das GetRun API-Vorgang, um den Status zu überprüfen und Details eines bestimmten Laufs abzurufen. Sie benötigen die Run-ID, die zurückgegeben wird, wenn Sie einen Lauf starten.

```
aws omics get-run --id run_id
```

Die Lauf-ID finden

Sie können die Run-ID auf folgende Weise finden:

- Aus dem StartRun API-Antwort — das `id` Das Feld wird zurückgegeben, wenn Sie einen Lauf starten.
- Aus dem HealthOmics Konsole — auf der Läuft Seite, die Run-ID wird in der ersten Spalte der Run-Liste angezeigt.
- Aus dem ListRuns API — listet alle Läufe in deinem Konto mit ihren IDs auf.

Die Antwort umfasst den Ausführungsstatus, Workflow-Details, Beschleunigerinformationen und Zeitstempel. Mögliche Ausführungsstatus sind:

Status	Description
`PENDING`	Der Lauf wurde eingereicht und wartet darauf, gestartet zu werden.
`STARTING`	HealthOmics stellt Ressourcen für den Lauf bereit.
`RUNNING`	Der Lauf führt aktiv Aufgaben aus.
`COMPLETED`	Der Lauf wurde erfolgreich abgeschlossen. Ausgabedateien sind im Amazon S3-Ausgabespeicherort verfügbar.
`FAILED`	Bei der Ausführung ist ein Fehler aufgetreten. Siehe the section called “Gründe für Ausführungsfehler”.
`CANCELLED`	Der Lauf wurde vom Benutzer abgebrochen.

Wann ist ein Lauf `COMPLETED`, Sie finden die Ausgabedateien in Ihrem Amazon S3-Ausgabe-Bucket in einem Ordner, der nach der Run-ID benannt ist.

Beispielantwort

Das folgende Beispiel zeigt die Antwort für eine abgeschlossene Ausführung eines privaten WDL-Workflows mit einem GPU-Beschleuniger:


```
{
    "arn": "arn:aws:omics:us-west-2:123456789012:run/7830534",
    "id": "7830534",
    "uuid": "96c57683-74bf-9d6d-ae7e-f09b097db14a",
    "outputUri": "s3://amzn-s3-demo-bucket/output",
    "status": "COMPLETED",
    "workflowId": "4074992",
    "workflowType": "PRIVATE",
    "workflowVersionName": "3.0.0",
    "roleArn": "arn:aws:iam::123456789012:role/OmicsRole",
    "name": "RunGroupMaxGpuTest",
    "runGroupId": "9938959",
    "digest": "sha256:a23a6fc54040d36784206234c02147302ab8658bed89860a...",
    "accelerators": "GPU",
    "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": {}
}
```


Wichtige Felder in der Antwort

Feld	Description
`uuid`	Universell eindeutiger Bezeichner für den Lauf. Zusammen mit `outputUri` , kann verwendet werden, um zu verfolgen, wo Ausgabeda ten geschrieben werden.
`status`	Aktueller Laufstatus (siehe vorherige Tabelle).
`failureReason`	Warum der Lauf fehlgeschlagen ist. Kam für eine zurück `FAILED` lauf. Siehe Gründe für Ausführungsfehler.

Feld	Description
`statusMessage`	Detail über den aktuellen Status. Bei einigen Fehlern benennt es die betroffene Ressource, z. B. eine Container-Image-URI.
`logLocation`	Der CloudWatch Log-Stream-ARNs für den Lauf: `runLogStream` für das Laufprotokoll und `engineLogStream` für das Motorprotokoll.
`workflowType`	Ob der Workflow `PRIVATE` oder `READY2RUN` .
`accelerators`	Verwendeter Beschleunigertyp (z. B. `GPU`), falls zutreffend.
`digest`	SHA-256 Zusammenfassung der Workflow-Definition.
`creationTime`	Wann der Lauf eingereicht wurde.
`startTime`	Als der Lauf mit der Ausführung begann.
`stopTime`	Wenn der Lauf beendet oder abgebrochen wurde.
`engineSettings`	Die Motoreinstellungen wurden auf den Lauf angewendet. Dieses Feld wird nur für Nextflow-Workflows zurückgegeben. Wenn Sie Engine-Einstellungen für WDL- oder CWL-Workflows angeben, werden diese stillschweigend ignoriert und dieses Feld ist nicht im GetRun Antwort. Siehe the section called “Geben Sie die Nextflow-Engine-Einstellungen an”.

Beispielantwort für einen Nextflow-Lauf mit Engine-Einstellungen

Für Nextflow-Läufe, die mit Engine-Einstellungen gestartet wurden, ist der GetRun Die Antwort beinhaltet eine `engineSettings` Karte, die die auf den Lauf angewendeten Werte wiedergibt. Das folgende Beispiel zeigt einen Lauf, der mit einer fixierten Engine-Version und zwei Profilen gestartet wurde.


```
{
    "arn": "arn:aws:omics:us-west-2:123456789012:run/9876543",
    "id": "9876543",
    "uuid": "1a2b3c4d-5e6f-7a8b-9c0d-1e2f3a4b5c6d",
    "status": "COMPLETED",
    "workflowId": "1122334",
```

```
    "workflowType": "PRIVATE",
    "workflowVersionName": "1.2.1",
    "roleArn": "arn:aws:iam::123456789012:role/OmicsRole",
    "name": "nextflow-rnaseq-run",
    "outputUri": "s3://amzn-s3-demo-bucket/output",
    "engineSettings": {
        "engineVersion": "26.04",
        "profile": "test,docker"
    },
    "creationTime": "2024-09-10T16:44:22.262471+00:00",
    "startTime": "2024-09-10T16:56:12.504000+00:00",
    "stopTime": "2024-09-10T17:22:29.908813+00:00",
    "tags": {}
}
```

Das `engineSettings` Feld wird nur für Nextflow-Workflows zurückgegeben. Bei WDL- und CWL-Workflows ist dieses Feld in der Antwort nicht vorhanden. Weitere Informationen finden Sie unter Geben Sie die Nextflow-Engine-Einstellungen an.

Listet alle Läufe auf

Benutze das ListRuns API-Vorgang, um alle Läufe in Ihrem Konto zu sehen:

```
aws omics list-runs
```

Die Antwort gibt eine paginierte Liste von Läufen mit zusammenfassenden Informationen wie Lauf-ID, Name, Status und Zeitstempeln zurück. Verwenden Sie das `--starting-token` Parameter, um zusätzliche Seiten abzurufen.

Aufgaben in einem Lauf auflisten

Benutze das ListRunTasks API-Vorgang, um alle Aufgaben zu sehen, die innerhalb eines bestimmten Laufs abgeschlossen wurden:

```
aws omics list-run-tasks --id run_id
```

Die Antwort gibt eine Liste von Aufgaben mit ihrem Status, ihrer Ressourcenzuweisung und ihren Zeitinformationen zurück.

Rufen Sie die Aufgabendetails ab

Benutze das GetRunTask API-Vorgang, um detaillierte Informationen zu einer bestimmten Aufgabe innerhalb eines Laufs zu erhalten:

```
aws omics get-run-task --id run_id --task-id task_id
```

Die Antwort umfasst Details auf Aufgabenebene wie CPU, Speicherzuweisung, Container-Image und Aufgabenstatus.

Führen Sie Metadaten in aus CloudWatch

Nachdem ein Lauf abgeschlossen ist, HealthOmics sendet die Laufmetadaten an CloudWatch Protokolle unter dem Stream:

```
manifest/run/run_id/run_uuid
```

Das Manifest enthält die vollständige Ausführungskonfiguration, Eingabeparameter, Ressourcenübersichten und Details auf Aufgabenebene. Das Folgende ist ein Beispiel für das Manifest:

```
[
  {
    "arn": "arn:aws:omics:us-east-1:123456789012:run/1695324",
    "creationTime": "2022-08-24T19:53:55.284Z",
    "resourceDigests": {
      "s3://omics-data/broad-references/hg38/v0/Homo_sapiens_assembly38.dict":
 "etag:3884c62eb0e53fa92459ed9bff133ae6",
      "s3://omics-data/broad-references/hg38/v0/Homo_sapiens_assembly38.fasta":
 "etag:e307d81c605fb91b7720a08f00276842-388",
      "s3://omics-data/broad-references/hg38/v0/Homo_sapiens_assembly38.fasta.fai":
 "etag:f76371b113734a56cde236bc0372de0a",
      "s3://omics-data/intervals/hg38-mjs-whole-chr.500M.intervals":
 "etag:27fdd1341246896721ec49a46a575334",
      "s3://omics-data/workflow-input-lists/dragen-gvcf-list.txt":
 "etag:e22f5aeed0b350a66696d8ffae453227"
    },
    "digest":
 "sha256:a5baaff84dd54085eb03f78766b0a367e93439486bc3f67de42bb38b93304964",
    "engine": "WDL",
    "main": "gatk4-basic-joint-genotyping-v2.wdl",
```

```
    "name": "1044-gvcfs",
    "outputUri": "s3://omics-data/workflow-output",
    "parameters": {
      "callset_name": "cohort",
      "input_gvcf_uris": "s3://omics-data/workflow-input-lists/dragen-gvcf-list.txt",
      "interval_list": "s3://omics-data/intervals/hg38-mjs-whole-chr.500M.intervals",
      "ref_dict": "s3://omics-data/broad-references/hg38/v0/
Homo_sapiens_assembly38.dict",
      "ref_fasta": "s3://omics-data/broad-references/hg38/v0/
Homo_sapiens_assembly38.fasta",
      "ref_fasta_index": "s3://omics-data/broad-references/hg38/v0/
Homo_sapiens_assembly38.fasta.fai"
    },
    "roleArn": "arn:aws:iam::123456789012:role/OmicsServiceRole",
    "startedBy": "arn:aws:sts::123456789012:assumed-role/admin/ahenroid-Isengard",
    "startTime": "2022-08-24T20:08:22.582Z",
    "status": "COMPLETED",
    "stopTime": "2022-08-24T20:08:22.582Z",
    "storageCapacity": 9600,
    "uuid": "a3b0ca7e-9597-4ecc-94a4-6ed45481aeab",
    "workflow": "arn:aws:omics:us-east-1:123456789012:workflow/1558364",
    "workflowType": "PRIVATE"
  },
  {
    "arn": "arn:aws:omics:us-east-1:123456789012:task/1245938",
    "cpus": 16,
    "creationTime": "2022-08-24T20:06:32.971290",
    "image": "123456789012.dkr.ecr.us-west-2.amazonaws.com/gatk",
    "imageDigest":
 "sha256:8051adab0ff725e7e9c2af5997680346f3c3799b2df3785dd51d4abdd3da747b",
    "memory": 32,
    "name": "geno-123",
    "run": "arn:aws:omics:us-east-1:123456789012:run/1695324",
    "startTime": "2022-08-24T20:08:22.278Z",
    "status": "SUCCESS",
    "stopTime": "2022-08-24T20:08:22.278Z",
    "uuid": "44c1a30a-4eee-426d-88ea-1af403858f76"
  },
  ...
]
```


Weitere Informationen zu Protokollen finden Sie unter CloudWatch zum HealthOmics läuft, siehe the section called “CloudWatch Logs”.

> **Note**
>
> Run-Metadaten werden nicht gelöscht von CloudWatch Protokolliert auch dann, wenn Sie den Lauf von entfernen HealthOmics. Du kannst verwenden CloudWatch um auf Metadaten für Läufe zuzugreifen, die nicht mehr verfügbar sind, über den HealthOmics API.

Führen Sie einen Run in erneut aus HealthOmics

Verwenden Sie für Läufe, die Sie noch nicht gelöscht haben, die Konsole oder API, um den Lauf erneut auszuführen. Verwenden Sie für Läufe, die Sie gelöscht haben, das Tool. HealthOmics rerun

Themen

- Führen Sie einen Lauf mit der Konsole erneut aus
- Führen Sie einen Lauf mithilfe der API erneut aus
- Verwenden Sie das Rerun-Tool

Führen Sie einen Lauf mit der Konsole erneut aus

Gehen Sie von der Konsole aus wie folgt vor, um einen Lauf erneut auszuführen:

1. Öffnen Sie die HealthOmics -Konsole.
2. Öffnen Sie bei Bedarf den linken Navigationsbereich (!). Wählen Sie Läufe aus.
3. Wählen Sie auf der Seite „Läufe“ den Lauf aus, der erneut ausgeführt werden soll.
4. Wählen Sie im Aktionsmenü über der Tabelle die Option Erneut ausführen aus.

Führen Sie einen Lauf mithilfe der API erneut aus

Verwenden Sie den StartRun API-Vorgang, um einen vorhandenen Lauf erneut auszuführen. Geben Sie die folgenden erforderlichen Eingaben ein:

- Eine Dienstrolle ARN (`roleArn`).
- Die ID des zu duplizierenden Laufs (`runId`).
- Ein Amazon S3 S3-Speicherort, an dem die Ausführung die Ausführungsausgaben (`outputUri`) speichert.

```
aws omics start-run
     --run-id run id \
     --role-arn arn:aws:iam::1234567892012:role/service-role/
OmicsWorkflow-20221004T164236 \
     --output-uri s3://workflow-output-b6f2fce1
```

Verwenden Sie das Rerun-Tool

Für einen gelöschten Lauf können Sie das HealthOmics rerun Tool herunterladen und verwenden, um den Lauf erneut auszuführen. Das Tool ruft Ausführungsinformationen aus dem CloudWatch Logs-Manifest ab. Laden Sie das rerun Tool aus dem HealthOmics GitHub Tool-Repository herunter.

Das folgende Beispiel zeigt, wie das rerun Tool verwendet wird.

```
aws-healthomics-rerun 9876543
```

Wenn der Lauf in existiert CloudWatch, erhalten Sie eine Antwort, die der folgenden Beispielausgabe ähnelt. Wenn der Workflow nicht mehr existiert, erhalten Sie eine Fehlermeldung.

```
Original request:
{
  "workflowId": "9679729",
  "roleArn": "arn:aws:iam::123456789012:role/DemoRole",
  "name": "sample_rerun",
  "parameters": {
    "image": "123456789012.dkr.ecr.us-west-2.amazonaws.com/default:latest",
    "file1": "omics://123456789012.storage.us-west-2.amazonaws.com/8647780323/
readSet/6389608538"
  },
  "outputUri": "s3://workflow-output-bcf2fcb1"
}
StartRun request:
{
  "workflowId": "9679729",
  "roleArn": "arn:aws:iam::123456789012:role/DemoRole",
  "name": "new test",
  "parameters": {
    "image": "123456789012.dkr.ecr.us-west-2.amazonaws.com/default:latest",
    "file1": "omics://123456789012.storage.us-west-2.amazonaws.com/8647780323/
readSet/6389608538"
  },
```

```
  "outputUri": "s3://workflow-output-bcf2fcb1"
}
StartRun response:
{
  "arn": "arn:aws:omics:us-west-2:123456789012:run/9171779",
  "id": "9171779",
  "status": "PENDING",
  "tags": {}
}
```

Einen Run in klonen HealthOmics

Sie können einen vorhandenen Lauf mithilfe der HealthOmics Konsole klonen. Beim Klonen wird ein neuer Lauf erstellt, der die Konfigurationswerte des geklonten Laufs verwendet. Sie können diese Standardwerte ändern und weitere optionale Eingaben hinzufügen.

1. Öffnen Sie die HealthOmics -Konsole.
2. Öffnen Sie bei Bedarf den linken Navigationsbereich (•). Wählen Sie Läufe.
3. Wählen Sie auf der Seite „Runs“ den Lauf aus, den Sie klonen möchten.
4. Wählen Sie im Aktionsmenü über der Tabelle die Option Ausführung klonen aus. Die Konsole öffnet das Formular zum Ausführen von Klonen. Das Formular ist identisch mit Start run, außer dass die Konsole das Formular mit allen relevanten Werten aus dem geklonten Lauf füllt.

   Die Konsole erstellt eine neue Run-ID für den Run-Clone und fügt diese Run-ID als Suffix zum Runnamen hinzu.

   Beim Durchblättern der Formularseiten können Sie die Konfigurationswerte nach Bedarf anpassen.
5. Nachdem Sie die Ausführungskonfiguration überprüft haben, wählen Sie Start run aus.

Einen Run in stornieren HealthOmics

Sie können einen Lauf stornieren, wenn er den Status`PENDING`, `STARTINGRUNNING`, oder hat`STOPPING`.

> **Note**
>
> Wenn Sie einen Lauf abbrechen, HealthOmics werden keine der Laufausgaben gespeichert.

Gehen Sie von der Konsole aus wie folgt vor, um einen Lauf abzubrechen:

1. Öffnen Sie die HealthOmics -Konsole.
2. Öffnen Sie bei Bedarf den linken Navigationsbereich (•). Wählen Sie Läufe aus.
3. Wählen Sie auf der Seite „Läufe“ den Lauf aus, den Sie abbrechen möchten.
4. Die Konsole öffnet die Seite mit den Ausführungsdetails. Wählen Sie im Statusbanner oben auf der Seite die Option Ausführung beenden aus.
5. Geben Sie Bestätigen ein, um den Lauf zu beenden.

Verwenden Sie den API-Vorgang, um einen Lauf mithilfe der CancelRun API abzubrechen.

Das folgende Beispiel zeigt, wie Sie einen Lauf mit dem abbrechen AWS CLI . Um das Beispiel auszuführen, ersetzen Sie das *run id* durch die ID des Laufs, den Sie abbrechen möchten. Bei Erfolg erfolgt keine Antwort.

```
aws omics cancel-run --id run id
```

Lösche einen Run in HealthOmics

Wenn Sie einen Lauf nicht mehr benötigen, können Sie ihn mithilfe der AWS CLI API oder der Konsole löschen. Sie können einen Lauf löschen, wenn er den Status `COMPLETED` oder hat`CANCELED`.

Gehen Sie von der Konsole aus wie folgt vor, um einen Lauf zu löschen:

1. Öffnen Sie die HealthOmics -Konsole.
2. Öffnen Sie bei Bedarf den linken Navigationsbereich (•). Wählen Sie Läufe.
3. Wählen Sie auf der Seite „Läufe“ einen oder mehrere Läufe aus, die Sie löschen möchten.
4. Wählen Sie im Aktionsmenü über der Tabelle die Option Löschen aus.
5. Geben Sie in der modalen Form confirm ein, um das Löschen zu bestätigen.

Der folgende AWS CLI Befehl löscht einen Lauf. Um das Beispiel auszuführen, ersetzen Sie das *run id* durch die ID des Laufs, den Sie löschen möchten. Es erfolgt keine Antwort, wenn der Lauf erfolgreich gelöscht wurde.

```
aws omics delete-run --id run id
```

Batch läuft in HealthOmics

AWS HealthOmics Mit Batchläufen können Sie mehrere Läufe in einer einzigen API-Anforderung einreichen. Jeder Lauf im Batch hat eine gemeinsame Basiskonfiguration, kann jedoch unterschiedliche Eingaben und laufspezifische Parameter haben. Batchläufe reduzieren den Aufwand für die Einreichung und vereinfachen das Lebenszyklusmanagement bei umfangreichen Workflow-Verarbeitungen.

Mit Batchläufen können Sie:

- Reichen Sie bis zu 100.000 Läufe in einem einzigen StartRunBatch Aufruf ein, wobei eine gemeinsame Konfiguration einmal definiert und auf alle Läufe angewendet wird.
- Wenden Sie Parameterüberschreibungen pro Lauf für einzelne Samples an, einschließlich Name, Ausgabe-URI, Parameter, Priorität und Tags.
- Verfolgen Sie den Gesamtstatus der Charge und den Fortschritt der einzelnen Durchläufe bis GetBatch zum Ende. ListRunsInBatch
- Stornieren Sie alle Läufe in einem Stapel mitCancelRunBatch.
- Löschen Sie alle Läufe in einem Stapel mitDeleteRunBatch.
- Löschen Sie Batch-Metadaten mitDeleteBatch.

Themen

- Konzepte für die Batch-Ausführung
- Voraussetzungen
- IAM-Berechtigungen für Batch-Operationen
- Erste Schritte: Reichen Sie Ihren ersten Stapel ein
- Einen Batch-Lauf starten
- Überwachung des Batch-Fortschritts
- Behandlung fehlgeschlagener Läufe
- Einen Batch stornieren
- Löschen von Batchläufen
- Löschen von Batch-Metadaten
- EventBridge Ereignisse für Stapel
- Einschränkungen und Überlegungen

Konzepte für die Batch-Ausführung

- Batch — Eine Sammlung von Workflow-Ausführungen mit einer gemeinsamen Konfiguration, die als einzelne Ressource mit eigenem Amazon-Ressourcennamen (ARN) und Lebenszyklusstatus verwaltet wird.
- Standard-Ausführungseinstellung (`defaultRunSetting`) — Workflow-Parameter, die von allen Läufen im Batch gemeinsam genutzt werden, wie Workflow-ID, IAM-Rolle, Ausgabe-URI, Sitzungsrichtlinie und gemeinsame Parameter.
- Run-specific Einstellung (`inlineSettingsoders3UriSettings`) — Per-run Konfigurationen, die die standardmäßige Ausführungseinstellung überschreiben oder mit ihr zusammenführen. Jeder Eintrag muss ein Unikat enthalten`runSettingId`.
- Run Setting-ID (`runSettingId`) — Eine erforderliche, vom Kunden bereitgestellte eindeutige Kennung für jede Ausführungskonfiguration innerhalb eines Batches. Verwenden Sie diese Option, ListRunsInBatch um die einzelnen Befehle `runSettingId` nach der HealthOmics-generated `runId` zuzuordnen, sodass Sie nachvollziehen können, welcher Lauf mit welcher Eingabekonfiguration erstellt wurde.
- Batch-Status — Der Gesamtstatus des Batch-Vorgangs. Mögliche Werte:
  - `CREATING`— Der Stapel wird erstellt.
  - `PENDING`— Der Stapel wurde erstellt; die Ausführungskonfigurationen werden asynchron validiert.
  - `SUBMITTING`— Die Validierung ist abgeschlossen; einzelne Läufe werden eingereicht.
  - `INPROGRESS`— Es wurde versucht, alle Durchläufe einzureichen; die Läufe werden ausgeführt.
  - `STOPPING`— Eine Stornierungsanfrage ist eingegangen; die Läufe werden gestoppt.
  - `CANCELLED`— Der Stapel wurde storniert.
  - `PROCESSED`— Alle Läufe haben einen Endzustand erreicht (abgeschlossen, fehlgeschlagen oder abgebrochen).
  - `FAILED`— Der Stapel selbst ist fehlgeschlagen, bevor die Läufe erstellt werden konnten. Details dazu finden Sie unter Batch-level Ausfälle.
  - `RUNS_DELETING`— Die Läufe im Stapel werden gelöscht.
  - `RUNS_DELETE_FAILED`— Einige oder alle Läufe im Stapel konnten nicht gelöscht werden.
  - `RUNS_DELETED`— Alle Läufe im Stapel wurden gelöscht.
- Übermittlungsstatus — Das Übermittlungsergebnis für einen einzelnen Durchlauf innerhalb des Batches. Mögliche Werte:

- `SUCCESS`— Der Lauf wurde erfolgreich übermittelt.
- `FAILED`— Die Übertragung des Laufs ist fehlgeschlagen (z. B. aufgrund eines Validierungsfehlers).
- `CANCEL_SUCCESS`— Der Lauf wurde erfolgreich abgebrochen.
- `CANCEL_FAILED`— Der Abbruch des Laufs ist fehlgeschlagen.
- `DELETE_SUCCESS`— Der Lauf wurde erfolgreich gelöscht.
- `DELETE_FAILED`— Der Löschvorgang ist fehlgeschlagen.

Voraussetzungen

Bevor Sie eine Batchausführung starten, stellen Sie sicher, dass Sie über Folgendes verfügen:

- Ein aktiver HealthOmics privater Workflow. Batch-Läufe werden mit Ready2Run-Workflows nicht unterstützt.
- Eine IAM-Servicerolle mit Berechtigungen zum Ausführen von HealthOmics Workflows und zum Zugriff auf Ihre Amazon S3-Buckets. Details hierzu finden Sie unter Servicerollen für AWS HealthOmics.
- Amazon S3-Standorte für Eingabedaten und Ausgabeergebnisse.
- Run-specific Parameter für jede Probe oder experimentelle Konfiguration.

IAM-Berechtigungen für Batch-Operationen

Ihre IAM-Identität muss über Berechtigungen sowohl für den Batchvorgang als auch für den zugrunde liegenden individuellen Ausführungsvorgang verfügen. Die folgende Beispielrichtlinie gewährt Berechtigungen für alle Batchvorgänge:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowBatchOperations",
      "Effect": "Allow",
      "Action": [
        "omics:StartRunBatch",
        "omics:GetBatch",
        "omics:ListBatch",
```

```
        "omics:ListRunsInBatch",
        "omics:CancelRunBatch",
        "omics:DeleteRunBatch",
        "omics:DeleteBatch"
      ],
      "Resource": "arn:aws:omics:*:123456789012:runBatch/*"
    },
    {
      "Sid": "AllowRunCreation",
      "Effect": "Allow",
      "Action": "omics:StartRun",
      "Resource": [
        "arn:aws:omics:*:123456789012:run/*",
        "arn:aws:omics:*:123456789012:workflow/*",
        "arn:aws:omics:*:123456789012:runGroup/*"
      ]
    },
    {
      "Sid": "AllowListOperations",
      "Effect": "Allow",
      "Action": [
        "omics:ListBatch",
        "omics:ListRunsInBatch"
      ],
      "Resource": "*"
    },
    {
      "Sid": "AllowPassRole",
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::123456789012:role/OmicsRole",
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": "omics.amazonaws.com"
        }
      }
    }
  ]
}
```

> **Note**
>
> StartRunBatcherfordert eine doppelte Autorisierung: `omics:StartRunBatch` für die Batch-Ressource und `omics:StartRun` für die Run-, Workflow- und Run-Gruppenressourcen. Beide Berechtigungen müssen erteilt werden, damit der Batch erfolgreich ist.

Für die übergebene `roleArn` IAM-Servicerolle (die von HealthOmics zur Ausführung der Läufe verwendet wird) sind dieselben Berechtigungen erforderlich wie für einzelne StartRun Aufrufe. Details hierzu finden Sie unter Servicerollen für AWS HealthOmics.

Tagging und tagbasierte Zugriffskontrolle

Batch-Läufe unterstützen Tags auf zwei Ebenen:

- Batch-Tags (Tags auf der StartRunBatch Anfrage) — Auf die Batch-Ressource angewendet. Tag-based Die Zugriffskontrolle (TBAC) wird für Batch-Tags vollständig durchgesetzt.
- Run-Tags (`runTags`in `defaultRunSetting` und pro Lauf`runTags`) — Wird auf einzelne Ausführungsressourcen angewendet. Diese Tags werden unter Verwendung derselben Prioritätsregeln wie Parameter zusammengeführt.

Wenn Sie tagbasierte Zugriffskontrollrichtlinien verwenden, stellen Sie sicher, dass Ihre IAM-Richtlinie die `omics:TagResource` entsprechenden `aws:RequestTag` Randbedingungen enthält. `aws:TagKeys`

Beispiel: Beschränken Sie die Batch-Erstellung auf Umgebungen außerhalb der Produktion

Die folgende Richtlinie ermöglicht Benutzern das Erstellen von Batches und Läufen, verweigert jedoch das Taggen von Ressourcen mit: `environment=production`

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowBatchAndRunCreation",
      "Effect": "Allow",
      "Action": [
        "omics:StartRunBatch",
        "omics:StartRun"
```

```
      ],
      "Resource": "*"
    },
    {
      "Sid": "AllowTaggingNonProd",
      "Effect": "Allow",
      "Action": "omics:TagResource",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/environment": ["dev", "test"]
        }
      }
    },
    {
      "Sid": "DenyProductionTags",
      "Effect": "Deny",
      "Action": "omics:TagResource",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/environment": "production"
        }
      }
    }
  ]
}
```

Mit dieser Richtlinie:

- `tags: {"environment": "dev"}`auf dem Stapel → Erlaubt
- `tags: {"environment": "production"}`auf dem Stapel → Abgelehnt (403)
- `runTags: {"environment": "production"}`in `defaultRunSetting` → Verweigert für einzelne Läufe während der asynchronen Erstellung

Erste Schritte: Reichen Sie Ihren ersten Stapel ein

Das folgende Beispiel zeigt, wie Sie einen kleinen Batch mit zwei Durchläufen einreichen, den Fortschritt überprüfen und die Ergebnisse überprüfen.

Schritt 1: Senden Sie den Stapel

```
aws omics start-run-batch \
  --batch-name "my-first-batch" \
  --default-run-setting '{
    "workflowId": "1234567",
    "roleArn": "arn:aws:iam::123456789012:role/OmicsRole",
    "outputUri": "s3://my-bucket/output/",
    "storageType": "DYNAMIC",
    "parameters": {"referenceUri": "s3://my-bucket/reference.fasta"}
  }' \
  --batch-run-settings '{
    "inlineSettings": [
      {
        "runSettingId": "sample-A",
        "parameters": {"inputUri": "s3://my-bucket/sampleA.fastq"}
      },
      {
        "runSettingId": "sample-B",
        "parameters": {"inputUri": "s3://my-bucket/sampleB.fastq"}
      }
    ]
  }'
```

Die Antwort gibt eine Batch-ID zurück, die Sie für alle nachfolgenden Operationen verwenden:

```
{
  "id": "7123456",
  "arn": "arn:aws:omics:us-west-2:123456789012:runBatch/7123456",
  "status": "CREATING"
}
```

Schritt 2: Überprüfen Sie den Batch-Fortschritt

```
aws omics get-batch --batch-id 7123456
```

Verfolgen Sie den Fortschritt des `status` Feldes von `CREATING` `PENDING` → → `SUBMITTING` → `INPROGRESS` → aus`PROCESSED`. `submissionSummary`Zeigt an, wie viele Durchläufe erfolgreich eingereicht wurden, und `runSummary` zeigt den Ausführungsfortschritt.

Schritt 3: Einzelne Läufe überprüfen

```
aws omics list-runs-in-batch --batch-id 7123456
```

Bei jedem Eintrag werden Ihre `runSettingId` Daten (z. B.`sample-A`) dem zugeordnet HealthOmics-generated`runId`, sodass Sie die Ergebnisse auf Ihre Eingabesamples zurückverfolgen können.

Schritt 4: Suchen Sie nach fehlgeschlagenen Einreichungen

```
aws omics list-runs-in-batch --batch-id 7123456 --submission-status FAILED
```

Wenn bei einem Durchlauf ein Fehler aufgetreten ist, wird in der Antwort Folgendes `submissionFailureReason` angegeben`submissionFailureMessage`, um Ihnen bei der Problemdiagnose zu helfen.

Einen Batch-Lauf starten

Wird verwendetStartRunBatch, um mehrere Läufe mit einer einzigen Anfrage zu senden. Die Anfrage beinhaltet:

- `defaultRunSetting`— Gemeinsame Konfiguration für alle Läufe im Batch. Dazu gehören Felder wie `workflowId`,`roleArn`,`outputUri`, `parameters`,`scratchStorageMode`, und`engineSettings`. Für Nextflow-Workflows verwenden Sie diese Option, `engineSettings` um Profile anzugeben, die für alle Läufe im Batch gelten. Weitere Informationen finden Sie unter Geben Sie die Nextflow-Engine-Einstellungen an.
- `batchRunSettings`— Die einzelnen Ausführungskonfigurationen, bereitgestellt als eine der folgenden:
  - `inlineSettings`— Eine Reihe von bis zu 100 laufspezifischen Konfigurationen, die direkt im Anforderungstext bereitgestellt werden. Jede Inline-Einstellung kann das Überschreiben des Standardprofils für einzelne Läufe beinhalten`engineSettings`.
  - `s3UriSettings`— Eine Amazon S3-URI, die auf eine JSON-Datei verweist, die die Ausführungskonfigurationen enthält. Erforderlich für Batches mit mehr als 100 Läufen und unterstützt bis zu 100.000 Läufe.

Sie können auch die folgenden Felder angeben:

- `batchName`— Ein optionaler, von Menschen lesbarer Name für den Stapel.
- `requestId`— Ein Idempotenz-Token, um doppelte Batch-Einreichungen zu verhindern.
- `tags`— Tags, die der Batch-Ressource selbst zugeordnet werden sollen.

Reichen Sie kleine Batches mit Inline-Run-Konfigurationen ein

Stellen Sie eine Reihe laufspezifischer Konfigurationen direkt im Anforderungstext mithilfe von bereit. `inlineSettings` Jeder Eintrag muss ein Unikat `runSettingId` (erforderlich) enthalten. Das `runSettingId` ist Ihr Schlüssel zum Korrelieren von Ergebnissen — wenn Sie anrufenListRunsInBatch, ordnet jeder Eintrag Sie dem `runSettingId` HealthOmics-generated `runId` Ende zu. `runArn`

Sie können bis zu 100 Einträge mit Inline-Konfigurationen einbeziehen.

```
{
  "batchName": "genomics-cohort-analysis",
  "requestId": "a1b2c3d4-e5f6-7890-abcd-ef1234567890",
  "tags": {
    "project": "genomics-study",
    "environment": "production"
  },
  "defaultRunSetting": {
    "workflowId": "1234567",
    "roleArn": "arn:aws:iam::123456789012:role/OmicsRole",
    "outputUri": "s3://my-bucket/output/",
    "parameters": {
      "referenceUri": "s3://my-bucket/reference/genome.fasta"
    },
    "runTags": {
      "analysis-type": "germline"
    }
  },
  "batchRunSettings": {
    "inlineSettings": [
      {
        "runSettingId": "sample-001",
        "name": "Sample-001-Analysis",
        "parameters": {
          "inputUri": "s3://my-bucket/input/sample-001.fastq",
          "sampleName": "sample-001"
        },
        "runTags": {
          "patient-id": "patient001"
        }
      },
      {
        "runSettingId": "sample-002",
```

```
        "name": "Sample-002-Analysis",
        "outputUri": "s3://my-bucket/output/special/",
        "parameters": {
          "inputUri": "s3://my-bucket/input/sample-002.fastq",
          "sampleName": "sample-002"
        },
        "runTags": {
          "patient-id": "patient002"
        }
      }
    ]
  }
}
```

CLI

```
aws omics start-run-batch \
  --batch-name "genomics-cohort-analysis" \
  --request-id "a1b2c3d4-e5f6-7890-abcd-ef1234567890" \
  --default-run-setting '{
    "workflowId": "1234567",
    "roleArn": "arn:aws:iam::123456789012:role/OmicsRole",
    "outputUri": "s3://my-bucket/output/",
    "storageType": "DYNAMIC",
    "parameters": {"referenceUri": "s3://my-bucket/reference.fasta"}
  }' \
  --batch-run-settings '{
    "inlineSettings": [
      {
        "runSettingId": "sample-001",
        "name": "Run-001",
        "parameters": {"inputUri": "s3://my-bucket/input1.fastq"}
      },
      {
        "runSettingId": "sample-002",
        "name": "Run-002",
        "parameters": {"inputUri": "s3://my-bucket/input2.fastq"}
      }
    ]
  }'
```

Reichen Sie große Batches mit Run-Konfigurationen in S3 ein

Speichern Sie Ihre Ausführungskonfigurationen für Batches mit mehr als 100 Durchläufen in einer JSON-Datei in Amazon S3 und geben Sie den URI mithilfe von an. `s3UriSettings` Die JSON-Datei muss ein Array von laufspezifischen Einstellungsobjekten enthalten, jedes mit einem eindeutigen Objekt. `runSettingId` Die Datei kann bis zu 100.000 Einträge enthalten.

Das S3-Dateiformat ist dasselbe wie das `inlineSettings` Array:

```
[
  {
    "runSettingId": "sample-001",
    "name": "Sample-001-Analysis",
    "parameters": {
      "inputUri": "s3://my-bucket/input/sample-001.fastq",
      "sampleName": "sample-001"
    }
  },
  {
    "runSettingId": "sample-002",
    "name": "Sample-002-Analysis",
    "parameters": {
      "inputUri": "s3://my-bucket/input/sample-002.fastq",
      "sampleName": "sample-002"
    }
  }
]
```

API-Anfrage

```
{
  "defaultRunSetting": {
    "workflowId": "1234567",
    "roleArn": "arn:aws:iam::123456789012:role/OmicsRole",
    "outputUri": "s3://my-bucket/output/",
    "parameters": {
      "referenceUri": "s3://my-bucket/reference/genome.fasta"
    }
  },
  "batchRunSettings": {
    "s3UriSettings": "s3://my-bucket/configs/run-configs.json"
  }
```

```
}
```

CLI

```
aws omics start-run-batch \
  --default-run-setting '{
    "workflowId": "1234567",
    "roleArn": "arn:aws:iam::123456789012:role/OmicsRole",
    "outputUri": "s3://my-bucket/output/",
    "parameters": {"referenceUri": "s3://my-bucket/reference.fasta"}
  }' \
  --batch-run-settings '{"s3UriSettings": "s3://my-bucket/configs/run-configs.json"}'
```

> **Important**
>
> Die in angegebene IAM-Servicerolle `roleArn` muss Lesezugriff auf das unter angegebene Amazon S3-Objekt haben. `s3UriSettings` HealthOmics validiert den Zugriff auf die Amazon S3-Datei während des synchronen API-Aufrufs und zeichnet das ETag der Datei auf. Wenn die Datei nach der Übermittlung geändert wird, schlägt der Batch während der asynchronen Verarbeitung fehl.

Antwort

Eine erfolgreiche Anfrage gibt eine Batch-ID und einen Anfangsstatus zurück:

```
{
  "id": "7123456",
  "arn": "arn:aws:omics:us-west-2:123456789012:runBatch/7123456",
  "status": "CREATING",
  "uuid": "96c57683-74bf-9d6d-ae7e-f09b097db14a",
  "tags": {
    "project": "genomics-study",
    "environment": "production"
  }
}
```

Schlägt die synchrone Validierung fehl (z. B. eine ungültige Workflow-ID oder eine Amazon S3-URI, auf die nicht zugegriffen werden kann), gibt die API einen Fehler zurück, bevor irgendwelche Läufe übermittelt werden.

Parameterreferenz

Die folgende Tabelle zeigt, welche StartRun Felder pro Lauf festgelegt werden können und welche nur auf Batch-Ebene gelten. Vollständige Feldbeschreibungen finden Sie in der StartRun API-Referenz.

Feld	Batch-level (**`defaultRunSetting`**)	Per-run konfigurierbar (**`inlineSettings`**)
`workflowId`	Ja	Nein
`workflowType`	Ja	Nein
`workflowVersionName`	Ja	Nein
`workflowOwnerId`	Ja	Nein
`roleArn`	Ja	Nein
`storageCapacity`	Ja	Nein
`storageType`	Ja	Nein
`runGroupId`	Ja	Nein
`logLevel`	Ja	Nein
`cacheBehavior`	Ja	Nein
`cacheId`	Ja	Nein
`retentionMode`	Ja	Nein
`networkingMode`	Ja	Nein
`configurationName`	Ja	Nein
`name`	Ja	Ja
`outputUri`	Ja	Ja

Feld	Batch-level (**`defaultRunSetting`**)	Per-run konfigurierbar (**`inlineSettings`**)
`parameters`	Ja	Ja (zusammengeführt)
`priority`	Ja	Ja
`runTags`	Ja	Ja (zusammengeführt)
`outputBucketOwnerId`	Ja	Ja

> **Note**
>
> Das Feld `runId` (für einzelne Wiederholungen) wird nicht als Eingabe für unterstützt. StartRunBatch Bei jeder Stapelübermittlung werden immer neue Durchläufe erstellt, und jeder Durchlauf erhält eine`runId`. Informationen Führen Sie einen Run in erneut aus HealthOmics zum erneuten Versuch einzelner Läufe finden Sie unter.

Zusammenführen von Parametern

Parameter von `defaultRunSetting` werden mit den laufspezifischen Parametern zusammengeführt, die in `inlineSettings` oder über eine S3-URI bereitgestellt werden. Run-specific Werte haben Vorrang, wenn sich Schlüssel überschneiden.

Beispiel:

Quelle	Parameters
`defaultRunSetting`	`{"referenceUri": "s3://bucket/ref.fasta", "version": "v1"}`
`inlineSettings` Eintrag	`{"inputUri": "s3://bucket/sample.fastq", "version": "v2"}`
Zusammengeführtes Ergebnis	`{"referenceUri": "s3://bucket/ref.fasta", "inputUri": "s3://buc`

Quelle	Parameters
	`ket/sample.fastq", "version": "v2"}`

Das gleiche Verhalten beim Zusammenführen gilt für. `runTags` In der Konfiguration pro Ausführung angegebene Tags überschreiben Tags mit demselben Schlüssel von`defaultRunSetting.runTags`, und neue Schlüssel werden hinzugefügt.

Beispiel für das Zusammenführen von Tags:

Quelle	Tags (Markierungen)
`defaultRunSetting.runTags`	`{"project": "cancer-research", "pipeline-version": "v2.1"}`
`inlineSettings[].runTags`	`{"patient-id": "patient001", "pipeline-version": "v3.0"}`
Zusammengeführtes Ergebnis (wird auf die Ausführung angewendet)	`{"project": "cancer-research", "patient-id": "patient001", "pipeline-version": "v3.0"}`

> **Note**
>
> Batch-level Tags (auf StartRunBatch Anfrage) werden nur auf die Batch-Ressource selbst angewendet. Sie werden nicht an einzelne Läufe vererbt. Run-level Die Tags werden durch `defaultRunSetting.runTags` und pro Lauf `runTags` gesteuert.

Schrittweise Einreichung

StartRunBatchvalidiert gemeinsame Felder synchron und kehrt sofort mit einer Batch-ID und einem Status zurück. `CREATING` Läufe in einem Batch werden schrittweise und asynchron mit einer kontrollierten Rate gemäß Ihren Durchsatzquoten übermittelt.

Die vollständige Liste der HealthOmics Kontingente finden Sie unter. HealthOmics Servicekontingente

Während der normalen Verarbeitung durchläuft der Stapel die folgenden Zustände:

1. `CREATING`— Der Stapel wird gerade erstellt.
2. `PENDING`— Der Stapel wurde erstellt, die Ausführungskonfigurationen werden überprüft.
3. `SUBMITTING`— Die Validierung ist abgeschlossen, einzelne Läufe werden eingereicht.
4. `INPROGRESS`— Es wurde versucht, alle Läufe einzureichen, die Läufe werden ausgeführt.
5. `PROCESSED`— Alle Läufe haben einen Endzustand erreicht.

> **Important**
>
> Für die Übermittlung von Batchläufen gilt dasselbe StartRun Durchsatzkontingent wie für direkte StartRun API-Aufrufe. Wenn Sie StartRun direkt anrufen, während ein großer Batch verarbeitet wird, konkurrieren sowohl die Batch-Übermittlungen als auch Ihre direkten Aufrufe um dieselbe Kapazität. Dies kann dazu führen, dass Batch-Läufe fehlschlagen `ThrottlingException` (Rate überschritten) oder dass Ihre StartRun Direktanrufe gedrosselt werden. Das Gleiche gilt für CancelRun und DeleteRun — diese teilen sich die Durchsatzquoten jeweils mit CancelRunBatch undDeleteRunBatch.

Überwachung des Batch-Fortschritts

Batch-Status abrufen

Wird verwendetGetBatch, um den Gesamtstatus und den Übermittlungsfortschritt für einen Stapel abzurufen.

```
aws omics get-batch --batch-id 7123456
```

Die Antwort enthält:

- `status`— Allgemeiner Batch-Status.
- `submissionSummary`— Zählt erfolgreiche und fehlgeschlagene Übermittlungen für Start-, Abbruch- und Löschvorgänge.

- `runSummary`— Anzahl der Läufe in jedem Ausführungsstatus. Mögliche Werte:`PENDING`, `STARTINGRUNNING`,`STOPPING`,`COMPLETED`,`DELETED`,`CANCELLED`, oder `FAILED`
- Zeitstempel für Lebenszyklusereignisse —`creationTime`,`submittedTime`,`processedTime`,`failedTime`.

Beispiel für eine Antwort:

```
{
  "id": "7123456",
  "arn": "arn:aws:omics:us-west-2:123456789012:runBatch/7123456",
  "uuid": "96c57683-74bf-9d6d-ae7e-f09b097db14a",
  "name": "genomics-cohort-analysis",
  "status": "INPROGRESS",
  "totalRuns": 96,
  "defaultRunSetting": {
    "workflowId": "1234567",
    "roleArn": "arn:aws:iam::123456789012:role/OmicsRole",
    "outputUri": "s3://my-bucket/output/"
  },
  "submissionSummary": {
    "successfulStartSubmissionCount": 94,
    "failedStartSubmissionCount": 2,
    "pendingStartSubmissionCount": 0
  },
  "runSummary": {
    "pendingRunCount": 10,
    "startingRunCount": 5,
    "runningRunCount": 50,
    "stoppingRunCount": 0,
    "completedRunCount": 29,
    "failedRunCount": 0,
    "cancelledRunCount": 0,
    "deletedRunCount": 0
  },
  "creationTime": "2025-03-15T10:00:00Z",
  "submittedTime": "2025-03-15T10:05:00Z",
  "tags": {
    "project": "genomics-study"
  }
}
```

> **Note**
>
> Die Zusammenfassungen der Ausführungsausführung sind letztendlich konsistent und können hinter den tatsächlichen Ausführungszuständen zurückbleiben. Die endgültigen Zählungen werden abgeglichen, wenn der Batch den Status erreichtPROCESSED.

Die Liste wird stapelweise ausgeführt

Wird verwendetListRunsInBatch, um detaillierte Informationen für einzelne Läufe innerhalb eines Batches abzurufen. Die Ergebnisse sind paginiert.

```
aws omics list-runs-in-batch --batch-id 7123456
```

Sie können Ergebnisse mithilfe eines der folgenden Abfrageparameter filtern. Pro Aufruf wird nur ein Filter unterstützt.

Filter	Description
`submissionStatus`	Nach Übermittlungsstatus filtern: `SUCCESS` `FAILEDCANCEL_SUCCESS` ,`CANCEL_FA` `ILED` ,`DELETE_SUCCESS` , oder`DELETE_FA` `ILED` .
`runSettingId`	Ruft den Lauf für eine bestimmte Ausführun gseinstellungs-ID ab.
`runId`	Ruft den Lauf für eine bestimmte Lauf-ID ab.

Beispiel: Fehlgeschlagene Übermittlungen auflisten

```
aws omics list-runs-in-batch --batch-id 7123456 --submission-status FAILED
```

Jedes Ergebnis beinhaltet:

- `runSettingId`— Die vom Kunden angegebene ID für die Run-Konfiguration.
- `runId`— Die HealthOmics-generated Lauf-ID (leer, wenn die Übermittlung fehlgeschlagen ist).

- `runArn`— Der vollständige ARN des Laufs.
- `submissionStatus`— Das Ergebnis der Einreichung (`SUCCESS`oder`FAILED`,`CANCEL_SUCCESS`,`CANCEL_FAILED`,`DELETE_SUCCESS`, oder`DELETE_FAILED`).
- `submissionFailureReason`und `submissionFailureMessage` — Einzelheiten, falls die Einreichung fehlgeschlagen ist.

> **Note**
>
> `runSettingId`ist die vom Kunden angegebene ID, die Sie in der Ausführungskonfiguration angegeben haben. `runId`ist die HealthOmics-generated ID, die nach erfolgreicher Übermittlung zugewiesen wurde. Wenn die Übermittlung fehlgeschlagen `runId` ist, ist leer.

Stapel auflisten

Wird verwendetListBatch, um alle Batch-Ressourcen in Ihrem Konto abzurufen. Die Ergebnisse sind paginiert.

```
aws omics list-batch
```

Sie können Ergebnisse mithilfe der folgenden Abfrageparameter filtern:

Filter	Description
`status`	Filtern Sie nach Stapelstatus.
`name`	Nach Stapelnamen filtern.
`runGroupId`	Filtern Sie nach der Run-Gruppen-ID.

Aktivieren Sie Run-Metriken

Der Bericht „Ausführungsmetriken“ berichtet nahezu in Echtzeit über die Ressourcenauslastung für die Läufe in einem Batch. Um Ausführungsmetriken zu aktivieren, muss die Servicerolle, die Sie für den Batch verwenden, über die `cloudwatch:PutMetricData` entsprechende Berechtigung verfügen. Weitere Informationen finden Sie unter Metriken für private Workflows ausführen.

Fügen Sie der Servicerolle, die Sie für den Batch verwenden, die folgende Berechtigung hinzu.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "cloudwatch:PutMetricData",
      "Resource": "*"
    }
  ]
}
```

Behandlung fehlgeschlagener Läufe

Bei der Stapelverarbeitung gibt es zwei verschiedene Arten von Fehlern. Für die Problembehandlung ist es wichtig, den Unterschied zu verstehen.

Batch-level Ausfälle

Ein Ausfall auf Batchebene bedeutet, dass der Batch selbst fehlgeschlagen ist — es wurden keine Läufe erstellt (oder nur einige wurden vor dem Ausfall erstellt). Der Batch-Status lautet. `FAILED`

Rufen Sie an GetBatch und überprüfen Sie das `failureReason` Feld. Zu den häufigsten Ausfallgründen gehören:

Kategorie „Ausfall“	Beispiel für **`failureReason`** eine Nachricht	Action
Validierungsfehler	„Batch hat 100001 Läufe, aber 100000 Läufe sind das Maximum. „, „Es wurden 50 einzelne Durchläufe erwartet, SettingIds aber es waren 49", „Ungültiges JSON-Format in Ausführungskonfigurationen“	Korrigieren Sie die Konfiguration und senden Sie einen neuen Stapel. Diese Fehler können nicht wiederholt werden.
Die S3-Konfiguration wurde geändert	„Erwartete UriConfigs s3-Etag:\" abc123\“, war aber: \ "def456\“. S3 UriConfigs	Ändern Sie die S3-Datei nach dem Absenden nicht. Senden

Kategorie „Ausfall“	Beispiel für **`failureReason`** eine Nachricht	Action
	haben sich seit dem Aufruf geändert“ StartRunBatch	Sie die Datei erneut mit der aktuellen Datei.
Vorübergehender Servicefehler	„Im Service ist ein vorübergehender Fehler aufgetreten. Versuchen Sie den Stapel erneut.“	Versuchen Sie es erneut, indem Sie denselben Stapel erneut senden. Verwenden Sie dasselbe `requestId` für Idempotenz.

Beispiel: Der Stapel ist aufgrund einer Validierung fehlgeschlagen

```
aws omics get-batch --batch-id 7123456
```

```
{
  "id": "7123456",
  "arn": "arn:aws:omics:us-west-2:123456789012:runBatch/7123456",
  "status": "FAILED",
  "failureReason": "Batch has 100001 runs but 100000 runs is the max.",
  "failedTime": "2025-03-15T10:01:00Z"
}
```

Run-level Ausfälle

Ein Fehler auf Run-Level bedeutet, dass der Batch selbst erfolgreich war (Status ist `INPROGRESS` oder`PROCESSED`), einzelne Läufe jedoch nicht gesendet werden konnten. Der Stapel setzt die Verarbeitung anderer Läufe fort — er stoppt nicht beim ersten Ausfall.

1. Prüfen Sie die Zusammenfassung der Einreichung

```
aws omics get-batch --batch-id 7123456
```

Schau dir an`submissionSummary.failedStartSubmissionCount`. Wenn dieser Wert größer als Null ist, sind einige Durchläufe beim Senden fehlgeschlagen.

2. Fehlgeschlagene Beiträge auflisten

```
aws omics list-runs-in-batch --batch-id 7123456 --submission-status FAILED
```

Jeder fehlgeschlagene Eintrag beinhaltet:

- `runSettingId`— Welche Ausführungskonfiguration ist fehlgeschlagen.
- `submissionFailureReason`— Die Fehlerkategorie.
- `submissionFailureMessage`— Eine ausführliche Fehlermeldung.

3. Gründe für das Scheitern der Einreichung

In der folgenden Tabelle sind die möglichen Werte für `submissionFailureReason` einzelne Durchläufe aufgeführt:

submissionFailureReason	Bedeutung	Wiederholbar?
`ValidationException`	Ausführungsparameter sind ungültig. Beispiel: Parameter stimmen nicht mit der Workflow-Definition überein, das S3-URI-Format ist ungültig oder es gibt Verstöße gegen Einschränkungen.	Nein — korrigieren Sie die Konfiguration.
`AccessDeniedException`	Der IAM-Servicerolle fehlen die Berechtigungen für den Zugriff auf die erforderlichen Ressourcen (S3-Eingaben, KMS-Schlüssel, CloudWatch Protokolle, ECR-Bilder).	Nein — aktualisieren Sie die Rollenrichtlinie.
`ResourceNotFoundException`	Eine Ressource, auf die verwiesen wird (Workflow, Ausführungsgruppe oder Run-Cache), ist nicht vorhanden	Nein — überprüfen Sie die Ressourcen-IDs.

submissionFailureReason	Bedeutung	Wiederholbar?
	oder befindet sich nicht in einem aktiven Zustand.	
`ServiceQuotaExceededException`	Das Konto hat die maximale Anzahl aktiver Läufe oder ein anderes Dienstkontingent erreicht.	Warten Sie, bis die Läufe abgeschlossen sind, oder fordern Sie eine Kontingenterhöhung an.
`ConflictException`	Ein widersprüchlicher Vorgang ist im Gange, z. B. ein Versuch, einen doppelten Lauf zu erstellen.	In der Regel löst er sich von selbst auf.
`ThrottlingException`	Die Anfrage wurde aufgrund von API-Ratenbeschränkungen gedrosselt.	Der Dienst versucht es automatisch erneut. Wenn das Problem auch nach wiederholten Versuchen bestehen bleibt, reduzieren Sie die Anzahl gleichzeitiger Batch-Übermittlungen.
`RequestTimeoutException`	Bei der Anfrage ist während der Verarbeitung ein Timeout aufgetreten.	Der Dienst versucht es automatisch erneut. Wenn das Problem weiterhin besteht, suchen Sie nach Downstream-Problemen.
`InternalServerException`	Ein unerwarteter Servicefehler ist aufgetreten.	Der Dienst versucht es automatisch erneut. Wenn er auch nach erneuten Versuchen weiterhin besteht, wenden Sie sich an den AWS Support.

> **Note**
>
> HealthOmics wiederholt automatisch vorübergehende Fehler (`ThrottlingException`,`RequestTimeoutException`,`InternalServerException`) bei jedem Lauf. Ein Lauf wird als `FAILED` erst markiert, wenn alle Wiederholungsversuche ausgeschöpft sind. Non-retryable Fehler (`ValidationException`,`AccessDeniedException`,`ResourceNotFoundException`) schlagen sofort ohne erneute Versuche fehl.

4. Fehlgeschlagene Läufe erneut einreichen

Erstellen Sie einen neuen Stapel, der nur die korrigierten Laufkonfigurationen enthält. Verwenden Sie dasselbe `defaultRunSetting` und schließen Sie nur die `runSettingId` Einträge ein, bei denen ein Fehler aufgetreten ist. Es gibt keinen integrierten Wiederholungsmechanismus für fehlgeschlagene Läufe — Sie müssen einen neuen Stapel einreichen.

Einen Batch stornieren

Wird verwendetCancelRunBatch, um einen laufenden Stapel abzubrechen. Der Vorgang zum Abbrechen:

- Verhindert, dass noch nicht gesendete und ausstehende Ausführungen gestartet werden
- Sendet CancelRun Anfragen für Läufe, die bereits gestartet wurden.

```
aws omics cancel-run-batch --batch-id 7123456
```

> **Important**
>
> - Abbrechen ist nur für Batches im Status `PENDINGSUBMITTING`, oder `INPROGRESS` zulässig.
> - Pro Stapel ist jeweils nur ein Vorgang zum Abbrechen oder Löschen zulässig.
> - Abbruchvorgänge sind nicht atomar und können teilweise erfolgreich sein. Verwenden Sie GetBatch zur Überprüfung `successfulCancelSubmissionCount` und `failedCancelSubmissionCount` in der`submissionSummary`.

Löschen von Batchläufen

Wird verwendetDeleteRunBatch, um die einzelnen Läufe innerhalb eines Batches zu löschen.

```
aws omics delete-run-batch --batch-id 7123456
```

Important

- Das Löschen ist nur für Batches im `CANCELLED` Status `PROCESSED` oder zulässig.
- Pro Stapel ist jeweils nur ein Vorgang zum Abbrechen oder Löschen zulässig.
- Löschvorgänge sind nicht atomar und können teilweise erfolgreich sein. Verwenden Sie GetBatch zur Überprüfung `successfulDeleteSubmissionCount` und `failedDeleteSubmissionCount` in der`submissionSummary`.

Löschen von Batch-Metadaten

Wird verwendetDeleteBatch, um die Batch-Ressource und die zugehörigen Metadaten zu entfernen. Dies ist ein separater Vorgang vonDeleteRunBatch.

```
aws omics delete-batch --batch-id 7123456
```

Important

- DeleteBatcherfordert, dass sich der Stapel im Endzustand befindet (`PROCESSED`,`FAILED`,`CANCELLED`,`RUNS_DELETE_FAILED`, oder`RUNS_DELETED`).
- DeleteBatchlöscht die einzelnen Läufe nicht. Verwenden Sie DeleteRunBatch zuerst, wenn Sie auch die Läufe entfernen möchten.
- Nach DeleteBatch Abschluss des Vorgangs sind die Batch-Metadaten nicht mehr zugänglich. Sie könnenGetBatch, ListRunsInBatchDeleteRunBatch, oder CancelRunBatch für einen gelöschten Batch nicht aufrufen.

EventBridge Ereignisse für Stapel

HealthOmics sendet Ereignisse an Amazon, EventBridge wenn ein Stapel den Status ändert. Sie können diese Ereignisse verwenden, um Arbeitsabläufe zu automatisieren, z. B. um eine Benachrichtigung auszulösen, wenn ein Batch abgeschlossen ist oder fehlschlägt, oder um eine Downstream-Pipeline zu starten, wenn alle Läufe abgeschlossen sind.

Batch-Ereignisse verwenden denselben Event-Bus und dieselbe Quelle wie andere Ereignisse. HealthOmics Allgemeine Anweisungen zur Einrichtung finden Sie unterVerwenden EventBridge mit AWS HealthOmics.

Typ der Veranstaltungsdetails

Ereignisname	Wird ausgegeben wenn
RunBatch Status ändern	Der Stapel wechselt in einen neuen Status (`CREATINGPENDING`,`SUBMITTING` ,`INPROGRESS` ,`STOPPING`,`CANCELLED` ,`PROCESSED` ,`FAILED`,`RUNS_DELETING` ,`RUNS_DELETED`)

Felder mit Eventdetails

Das `detail` Objekt in einem Batch-Ereignis umfasst die folgenden Felder:

Feld	Typ	Description
`omicsVersion`	Zeichenfolge	Version des Ereignisschemas (derzeit 1.0.0).
`arn`	Zeichenfolge	Der Batch-ARN.
`batchId`	Zeichenfolge	Die Batch-ID.
`status`	Zeichenfolge	Der neue Batchstatus.
`uuid`	Zeichenfolge	Die Batch-UUID.

Feld	Typ	Description
`batchName`	Zeichenfolge	Der Batchname (falls angegeben).
`totalRuns`	Zeichenfolge	Gesamtzahl der Durchläufe im Stapel.
`failureReason`	Zeichenfolge	Fehlergrund (nur vorhanden , wenn der Status lautet`FAILED`).
`failureMessage`	Zeichenfolge	Detaillierte Fehlermeldung (nur vorhanden, wenn der Status erreicht ist`FAILED`).
`successfulStartSubmissionCount`	Zeichenfolge	Anzahl der erfolgreich übermittelten Läufe.
`failedStartSubmissionCount`	Zeichenfolge	Anzahl der Läufe, die nicht übermittelt werden konnten.
`pendingStartSubmissionCount`	Zeichenfolge	Anzahl der Läufe, die noch eingereicht werden müssen.
`pendingRunCount`	Zeichenfolge	Anzahl der Läufe im Status „Ausstehend“.
`startingRunCount`	Zeichenfolge	Anzahl der startenden Läufe.
`runningRunCount`	Zeichenfolge	Anzahl der derzeit laufenden Läufe.
`stoppingRunCount`	Zeichenfolge	Anzahl der Läufe, die gestoppt werden.
`completedRunCount`	Zeichenfolge	Anzahl der abgeschlossenen Läufe.

Feld	Typ	Description
`failedRunCount`	Zeichenfolge	Anzahl der fehlgeschlagenen Läufe.
`cancelledRunCount`	Zeichenfolge	Anzahl der stornierten Läufe.
`deletedRunCount`	Zeichenfolge	Anzahl der gelöschten Läufe.
`workflowId`	Zeichenfolge	Die Workflow-ID.
`workflowArn`	Zeichenfolge	Der Workflow-ARN.
`workflowVersionArn`	Zeichenfolge	Der ARN der Workflow-Version (falls zutreffend).
`workflowOwnerId`	Zeichenfolge	Die Konto-ID des Workflow-Besitzers (für gemeinsam genutzte Workflows).
`runCache`	Zeichenfolge	Der Run-Cache-ARN (falls zutreffend).
`runCacheBehavior`	Zeichenfolge	Das Verhalten des Run-Cache (falls zutreffend).

Beispiele für Ereignisse

Beispiel: Ereignis „Batch abgeschlossen“

```
{
  "version": "0",
  "id": "a1b2c3d4-5678-90ab-cdef-example11111",
  "detail-type": "RunBatch Status Change",
  "source": "aws.omics",
  "account": "123456789012",
  "time": "2025-03-15T12:30:00Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:omics:us-west-2:123456789012:runBatch/7123456"
```

```
  ],
  "detail": {
    "omicsVersion": "1.0.0",
    "arn": "arn:aws:omics:us-west-2:123456789012:runBatch/7123456",
    "batchId": "7123456",
    "status": "PROCESSED",
    "uuid": "96c57683-74bf-9d6d-ae7e-f09b097db14a",
    "batchName": "genomics-cohort-analysis",
    "totalRuns": "96",
    "successfulStartSubmissionCount": "94",
    "failedStartSubmissionCount": "2",
    "pendingStartSubmissionCount": "0",
    "completedRunCount": "90",
    "failedRunCount": "4",
    "cancelledRunCount": "0",
    "workflowId": "1234567"
  }
}
```

Beispiel: Ereignis „Batch ist fehlgeschlagen“

```
{
  "version": "0",
  "id": "a1b2c3d4-5678-90ab-cdef-example22222",
  "detail-type": "RunBatch Status Change",
  "source": "aws.omics",
  "account": "123456789012",
  "time": "2025-03-15T10:01:00Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:omics:us-west-2:123456789012:runBatch/7123456"
  ],
  "detail": {
    "omicsVersion": "1.0.0",
    "arn": "arn:aws:omics:us-west-2:123456789012:runBatch/7123456",
    "batchId": "7123456",
    "status": "FAILED",
    "failureReason": "VALIDATION_EXCEPTION",
    "failureMessage": "Expected 100 unique runSettingIds but there were 99"
  }
}
```

Beispiel: EventBridge Regel für die Stapelvervollständigung

Das folgende Ereignismuster stimmt überein, wenn ein Stapel einen Endzustand erreicht:

```
{
  "source": ["aws.omics"],
  "detail-type": ["RunBatch Status Change"],
  "detail": {
    "status": ["PROCESSED", "FAILED", "CANCELLED"]
  }
}
```

Einschränkungen und Überlegungen

- Kontingente für gemeinsamen Durchsatz — Für Batch-Operationen gelten dieselben Kontingente pro Konto wie für ihre jeweiligen API-Gegenstücke. StartRunBatchverbraucht StartRun Servicekontingente. CancelRunBatchverbraucht CancelRun Kontingente und DeleteRunBatch verbraucht DeleteRun Kontingente. Vermeiden Sie es, einzelne Ausführungs-APIs aufzurufen, während ein großer Batch in Bearbeitung ist, da dies zu Fehlern bei der Übermittlung führen kann.
- Schrittweise Übermittlung — Läufe in einem Batch werden schrittweise und asynchron gemäß Ihren Durchsatzquoten übermittelt.
- Non-atomic Operationen — StartRunBatchCancelRunBatch, und alle DeleteRunBatch können teilweise erfolgreich sein. Überprüfe immer die Übermittlungszusammenfassungen, um herauszufinden, welche Läufe wiederholt werden müssen.
- Eventuelle Konsistenz — Die Anzahl der Ausführungsstatuswerte GetBatch kann hinter den tatsächlichen Ausführungszuständen zurückbleiben. Die endgültigen Zählungen sind korrekt, sobald der Batch den Status erreicht hat`PROCESSED`.
- Ein Filter pro Listenaufruf — ListRunsInBatch und ListBatch unterstützt nur einen Filter pro API-Aufruf.
- Re-run nicht unterstützt — Das Feld `runId` (erneut ausführen) wird in StartRunBatch nicht unterstützt. Bei jeder Stapelübermittlung werden immer neue Durchläufe erstellt.
- Ready2Run-Workflows — Batch-Läufe werden in Ready2Run-Workflows nicht unterstützt.
- Begrenzung der Inline-Konfiguration — Inline-Konfigurationen (`inlineSettings`) unterstützen bis zu 100 Einträge. Verwenden `s3UriSettings` Sie für größere Chargen. Diese Grenze ist nicht einstellbar.
- S3-Konfigurationsdatei — Die S3-Konfigurationsdatei muss ein JSON-Array von Run-Setting-Objekten sein. Die maximale Dateigröße beträgt 6 GB und unterstützt bis zu 100.000 Ausführungskonfigurationen.

- Unveränderlichkeit der S3-Datei — Ändern Sie die S3-Konfigurationsdatei nicht, nachdem Sie den Batch gesendet haben. HealthOmics validiert das Entitäts-Tag (ETag) der Datei während der Übermittlung und schlägt beim Batch fehl, wenn sich die Datei geändert hat.

HealthOmics Run-Gruppen verwenden

Sie können optional eine Ausführungsgruppe erstellen, um die Rechenressourcen für die Läufe, die Sie der Gruppe hinzufügen, zu begrenzen. Run-Groups können Ihnen dabei helfen:

- Stellen Sie Ihre Läufe in eine Warteschlange, damit Sie die Dienstlimits nicht überschreiten.
- Halten Sie ungenutzte Aufgaben fest, indem Sie eine maximale Ausführungsdauer festlegen.
- Verwalte die Priorität jedes Laufs so, dass die wichtigsten Läufe zuerst abgeschlossen werden.

Wenn Sie die maximale Anzahl gleichzeitiger vCPU, GPU oder Läufe festlegen, werden ausgeführte Aufgaben in eine Warteschlange gestellt, wenn die maximale Anzahl erreicht ist. Wenn Sie eine maximale Ausführungsdauer festlegen, schlägt die Ausführung fehl, wenn sie die maximale Dauer überschreitet.

Verwenden Sie die Einstellung für die Ausführungspriorität, um die Priorität innerhalb einer Ausführungsgruppe festzulegen.

Dienstlimits haben Vorrang vor Grenzwerten für Ausführungsgruppen. Wenn Sie beispielsweise ein Maximum für Ausführungsgruppen auf einen höheren Wert als Ihr Dienstmaximum in einer Region festlegen, wird das Dienstmaximum HealthOmics angewendet.

Themen

- Priorität ausführen
- Erstellen Sie mit der Konsole eine Ausführungsgruppe
- Erstellen Sie eine Ausführungsgruppe mit der CLI
- Löschen Sie eine Ausführungsgruppe mithilfe der Konsole
- Löschen Sie eine Ausführungsgruppe mit der CLI

Priorität ausführen

Sie können die Ausführungspriorität verwenden, um die Priorität von Läufen in einer Ausführungsgruppe festzulegen.

Wenn mehrere Läufe dieselbe Priorität haben, hat der Lauf, der zuerst gestartet wurde, die höhere Priorität.

Sie können auch eine Priorität für einen Lauf festlegen, der sich nicht in einer Ausführungsgruppe befindet. Die Priorität wird mit den Prioritäten aller anderen Läufe verglichen, die sich nicht in einer Ausführungsgruppe befinden

Sie legen die Ausführungspriorität fest, wenn Sie den Lauf starten. Weitere Informationen finden Sie unter Starte einen Lauf in HealthOmics.

Erstellen Sie mit der Konsole eine Ausführungsgruppe

Um eine Ausführungsgruppe zu erstellen

1. Öffnen Sie die HealthOmics -Konsole.
2. Öffnen Sie bei Bedarf den linken Navigationsbereich (±1). Wählen Sie Gruppen ausführen.
3. Wählen Sie auf der Seite „Run-Gruppen“ die Option Run-Gruppe erstellen aus.
4. Geben Sie auf der Detailseite zur Ausführungsgruppe erstellen die folgenden Informationen ein

   - Name der Ausführungsgruppe — Ein eindeutiger Name für diese Ausführungsgruppe.
   - Max. vCPU für gleichzeitige Läufe — Die maximale Anzahl von vCPUs , die gleichzeitig über alle aktiven Läufe in der Ausführungsgruppe ausgeführt werden können.
   - Max GPUs — Die maximale Anzahl davon GPUs , die gleichzeitig auf allen aktiven Läufen in der Ausführungsgruppe ausgeführt werden können.
   - Max. Laufzeit (Minuten) pro Lauf — Die maximale Laufzeit für jeden Lauf (in Minuten). Wenn ein Lauf die maximale Laufzeit überschreitet, schlägt der Lauf automatisch fehl.
   - Max. Anzahl gleichzeitiger Läufe — Die maximale Anzahl von Läufen, die gleichzeitig ausgeführt werden können.
5. (optional) Sie können der Ausführungsgruppe bis zu 50 Tags hinzufügen.
6. Wählen Sie „Ausführungsgruppe erstellen“.

Erstellen Sie eine Ausführungsgruppe mit der CLI

Um eine Ausführungsgruppe zu erstellen, verwenden Sie den create-run-groupAPI-Vorgang, um eine Ausführungsgruppe mit dem Namen zu erstellen`TestRunGroup`. Im folgenden Beispiel werden maximal 20 CPUs, 10 GPUs, 5 Läufe und eine maximale Ausführungsdauer von 600 Minuten festgelegt.

```
aws omics create-run-group --name TestRunGroup \
--max-cpus 20 \
--max-gpus 10 \
--max-duration 600 \
--max-runs 5
```

Die Antwort aus diesem API-Vorgang enthält die ID der neu erstellten`RunGroup`.

```
{
    "arn": "arn:aws:omics:us-west-2:12345678901:runGroup/2839621",
    "id": "2839621",
    "tags": {}
}
```

Um zusätzliche Informationen über die Run-Gruppe zu erhalten, verwenden Sie diese ID zusammen mit dem get-run-groupAPI-Vorgang, wie im folgenden Beispiel gezeigt.

```
aws omics get-run-group --id run group id
```

Die Antwort umfasst die Grenzwerteinstellungen für die Ausführungsgruppe und die zugewiesenen Tags.

```
{
    "arn": "arn:aws:omics:us-west-2:776893852117:runGroup/2839621",
    "id": "2839621",
    "name": "TestRunGroup",
    "maxCpus": 20,
    "maxRuns": 5,
    "maxDuration": 600,
    "creationTime": "2024-06-12T15:35:39.191730+00:00",
    "tags": {},
    "maxGpus": 10
}
```

Sie können den list-run-groupAPI-Vorgang auch verwenden, um alle erstellten Ausführungsgruppen anzuzeigen.

```
aws omics list-run-groups
```

Löschen Sie eine Ausführungsgruppe mithilfe der Konsole

Sie können eine Ausführungsgruppe löschen, wenn dieser Ausführungsgruppe keine Läufe mit dem Status`PENDING`, `STARTINGRUNNING`, oder zugeordnet sind`STOPPING`.

Gehen Sie wie folgt vor, um eine Ausführungsgruppe zu löschen.

1. Öffnen Sie die HealthOmics -Konsole.
2. Öffnen Sie bei Bedarf den linken Navigationsbereich (±1). Wählen Sie Gruppen ausführen.
3. Wählen Sie auf der Seite „Gruppen ausführen“ die Ausführungsgruppe aus, die Sie löschen möchten, und klicken Sie im Feld xx auf Löschen.

Löschen Sie eine Ausführungsgruppe mit der CLI

Sie können eine Ausführungsgruppe löschen, wenn dieser Ausführungsgruppe mit dem Status`PENDING`, `STARTINGRUNNING`, oder keine Läufe zugeordnet sind`STOPPING`.

Das folgende Beispiel zeigt, wie Sie die verwenden können AWS CLI , um eine Ausführungsgruppe zu löschen. Sie werden keine Antwort erhalten. Um das Beispiel auszuführen, ersetzen Sie das *`run group id`* durch die ID der Ausführungsgruppe, die Sie löschen möchten.

```
aws omics delete-run-group --id run group id
```

Aufruf-Caching für Läufe HealthOmics

AWS HealthOmics unterstützt das Zwischenspeichern von Anrufen, auch bekannt als Resume, für private Workflows. Beim Aufruf-Caching werden die Ausgaben abgeschlossener Workflow-Aufgaben nach Abschluss eines Laufs gespeichert. Nachfolgende Läufe können die Aufgabenausgaben aus dem Cache verwenden, anstatt die Aufgabenausgaben erneut zu berechnen. Durch das Caching von Aufrufen wird die Nutzung der Rechenressourcen reduziert, was zu kürzeren Ausführungsdauern und zu Einsparungen bei den Rechenkosten führt.

Nach Abschluss der Ausführung können Sie auf die zwischengespeicherten Aufgabenausgabedateien zugreifen. Für das erweiterte Debugging und die Problembehandlung von Aufgaben können Sie zwischengespeicherte Aufgabendateien zwischenspeichern, indem Sie diese Dateien in der Workflow-Definition als Aufgabenausgaben angeben.

Sie können die Zwischenspeicherung von Aufrufen verwenden, um die abgeschlossenen Aufgabenergebnisse fehlgeschlagener Ausführungen zu speichern. Die nächste Ausführung beginnt mit der letzten erfolgreich abgeschlossenen Aufgabe, anstatt die abgeschlossenen Aufgaben erneut zu berechnen.

Wenn HealthOmics kein passender Cache-Eintrag für eine Aufgabe gefunden wird, schlägt die Ausführung nicht fehl. HealthOmics berechnet die Aufgabe und ihre abhängigen Aufgaben neu.

Informationen zur Behebung von Problemen mit der Zwischenspeicherung von Anrufen finden Sie unter. Behebung von Problemen beim Zwischenspeichern von Anrufen

Themen

- So funktioniert das Caching von Anrufen
- Einen Run-Cache erstellen
- Einen Run-Cache aktualisieren
- Löschen eines Run-Caches
- Inhalt eines Run-Caches
- Engine-specific Caching-Funktionen
- Verwenden Sie den Run-Cache

So funktioniert das Caching von Anrufen

Um Call Caching zu verwenden, erstellen Sie einen Run-Cache und konfigurieren ihn so, dass er über einen zugehörigen Amazon S3 S3-Speicherort für die zwischengespeicherten Daten verfügt. Wenn Sie einen Lauf starten, geben Sie den Run-Cache an. Ein Run-Cache ist nicht für einen einzigen Workflow vorgesehen. Läufe aus mehreren Workflows können denselben Cache verwenden.

Während der Exportphase eines Laufs exportiert das System die abgeschlossenen Aufgabenausgaben an den Amazon S3 S3-Speicherort. Um zwischengeschaltete Aufgabendateien zu exportieren, deklarieren Sie diese Dateien in der Workflow-Definition als Aufgabenausgaben. Das

Aufruf-Caching speichert auch intern Metadaten und erstellt eindeutige Hashes für jeden Cache-Eintrag.

Für jede Aufgabe in einem Lauf erkennt die Workflow-Engine, ob es für diese Aufgabe einen passenden Cache-Eintrag gibt. Wenn es keinen passenden Cache-Eintrag gibt, HealthOmics berechnet die Aufgabe. Wenn es einen passenden Cache-Eintrag gibt, ruft die Engine die zwischengespeicherten Ergebnisse ab.

HealthOmics Verwendet zum Abgleichen von Cache-Einträgen den Hashing-Mechanismus, der in den systemeigenen Workflow-Engines enthalten ist. HealthOmicserweitert diese vorhandenen Hash-Implementierungen um HealthOmics Variablen wie S3-ETags und ECR-Container-Digests.

HealthOmics unterstützt das Caching von Aufrufen für diese Workflow-Sprachversionen:

- Die WDL-Versionen 1.0, 1.1 und die Entwicklungsversion
- Nextflow Version 23.10 und höher
- Alle CWL-Versionen

> **Note**
>
> HealthOmics unterstützt kein Aufruf-Caching für Ready2Run-Workflows.

Themen

- Modell der geteilten Verantwortung
- Anforderungen an das Zwischenspeichern von Aufgaben
- Führen Sie die Cache-Leistung aus
- Ereignisse zur Aufbewahrung und Invalidierung von Daten zwischenspeichern

Modell der geteilten Verantwortung

Die Benutzer sind gemeinsam dafür verantwortlich, AWS zu bestimmen, ob Aufgaben und Läufe sich für das Call-Caching eignen. Das Aufruf-Caching erzielt die besten Ergebnisse, wenn alle Aufgaben idempotent sind (wiederholte Ausführungen einer Aufgabe mit denselben Eingaben führen zu denselben Ergebnissen).

Wenn eine Aufgabe jedoch nicht deterministische Elemente enthält (wie Zufallszahlengenerationen oder Systemzeit), können wiederholte Ausführungen der Aufgabe mit denselben Eingaben zu unterschiedlichen Ausgaben führen. Dies kann sich auf folgende Weise auf die Effektivität des Caching von Anrufen auswirken:

- Wenn ein Cache-Eintrag HealthOmics verwendet wird (der durch einen vorherigen Lauf erstellt wurde), der nicht mit der Ausgabe identisch ist, die die Ausführung der Aufgabe für den aktuellen Lauf erzeugen würde, kann der Lauf zu anderen Ergebnissen führen als derselbe Lauf ohne Zwischenspeicherung.
- HealthOmics findet möglicherweise keinen passenden Cache-Eintrag für eine Aufgabe, der entsprechen sollte, weil die Aufgabenausgabe nicht deterministisch ist. Wenn der gültige Cache-Eintrag nicht gefunden wird, wird die Aufgabe bei der Ausführung unnötig neu berechnet, wodurch die Kosteneinsparung durch die Verwendung von Aufruf-Caching beeinträchtigt wird.

Im Folgenden sind bekannte Verhaltensweisen von Aufgaben aufgeführt, die zu nicht deterministischen Ergebnissen führen können, die sich auf die Ergebnisse der Zwischenspeicherung von Aufrufen auswirken:

- Verwendung von Zufallszahlengeneratoren.
- Abhängigkeit von der Systemzeit.
- Verwendung von Parallelität (Rennbedingungen können zu Leistungsabweichungen führen).
- Abrufen von lokalen oder entfernten Dateien, die über die in den Eingabeparametern der Aufgabe angegebenen hinausgehen.

Weitere Szenarien, die zu nicht deterministischem Verhalten führen können, finden Sie auf der Nextflow-Dokumentationsseite unter Non-deterministic Prozesseingaben.

Wenn Sie vermuten, dass eine Aufgabe nicht deterministische Ergebnisse erzeugt, sollten Sie die Funktionen der Workflow-Engine verwenden, um zu vermeiden, dass bestimmte Aufgaben zwischengespeichert werden, die nicht deterministisch sind. Anweisungen, wie Sie das Caching für einzelne Aufgaben in jeder unterstützten Workflow-Sprache deaktivieren können, finden Sie unter. Engine-specific Caching-Funktionen

Wir empfehlen Ihnen, Ihre spezifischen Anforderungen an den Arbeitsablauf und die Aufgaben gründlich zu überprüfen, bevor Sie das Caching von Anrufen in Umgebungen aktivieren, in denen ineffektives Caching von Anrufen oder andere als erwartete Ergebnisse ein Risiko darstellen können.

Beispielsweise sollten die potenziellen Einschränkungen von Call Caching bei der Entscheidung, ob Call Caching für klinische Anwendungsfälle geeignet ist, sorgfältig abgewogen werden.

Anforderungen an das Zwischenspeichern von Aufgaben

HealthOmics speichert Aufgabenausgaben für Aufgaben, die die folgenden Anforderungen erfüllen:

- Die Aufgabe muss einen Container definieren. HealthOmics speichert keine Ausgaben für eine Aufgabe ohne Container im Cache.
- Die Aufgabe muss eine oder mehrere Ausgaben erzeugen. Sie geben die Ausgaben der Aufgaben in der Workflow-Definition an.
- Die Workflow-Definition darf keine dynamischen Werte verwenden. Wenn Sie beispielsweise einen Parameter mit einem Wert an eine Aufgabe übergeben, der bei jedem Lauf erhöht wird, werden die Ausgaben der Aufgabe HealthOmics nicht zwischengespeichert.

> **Note**
>
> Wenn mehrere Aufgaben in einer Ausführung dasselbe Container-Image verwenden, wird HealthOmics für alle diese Aufgaben dieselbe Image-Version bereitgestellt. Nach HealthOmics dem Abrufen des Images werden alle Aktualisierungen des Container-Images für die Dauer der Ausführung ignoriert. Dieser Ansatz bietet eine vorhersehbare und konsistente Benutzererfahrung und verhindert potenzielle Probleme, die sich aus Aktualisierungen des Container-Images ergeben könnten, die während der Ausführung bereitgestellt werden.

Führen Sie die Cache-Leistung aus

Wenn Sie das Zwischenspeichern von Aufrufen für einen Lauf aktivieren, stellen Sie möglicherweise die folgenden Auswirkungen auf die Ausführungsleistung fest:

- HealthOmics Speichert beim ersten Lauf die Cache-Daten für Aufgaben in der Ausführung. Bei diesem Lauf kann es zu längeren Exportzeiten kommen, da das Aufruf-Caching die Menge der Exportdaten erhöht.
- Wenn Sie in nachfolgenden Ausführungen einen Lauf aus dem Cache wieder aufnehmen, kann dies die Anzahl der Verarbeitungsschritte und die Laufzeit verringern.

- Wenn Sie sich auch dafür entscheiden, Zwischendateien als Ausgaben zu deklarieren, können Ihre Exportzeiten sogar noch länger sein, da diese Daten ausführlicher sein können.

Ereignisse zur Aufbewahrung und Invalidierung von Daten zwischenspeichern

Der Hauptzweck eines Run-Caches besteht darin, die Berechnung der Aufgaben während der Ausführung zu optimieren. Wenn es einen gültigen passenden Cache-Eintrag für eine Aufgabe gibt, wird der Cache-Eintrag HealthOmics verwendet, anstatt die Aufgabe neu zu berechnen. Andernfalls wird das Standardverhalten des Dienstes HealthOmics wiederhergestellt, bei dem die Aufgabe und die von ihr abhängigen Aufgaben neu berechnet werden. Bei diesem Ansatz führen Cache-Fehler nicht dazu, dass die Ausführung fehlschlägt.

Es wird empfohlen, die Größe des Run-Caches zu verwalten. Im Laufe der Zeit sind Cacheeinträge aufgrund von Workflow-Engine- oder HealthOmics Service-Updates oder aufgrund von Änderungen, die Sie an der Ausführung oder den Ausführungsaufgaben vorgenommen haben, möglicherweise nicht mehr gültig. Die folgenden Abschnitte enthalten zusätzliche Informationen.

Themen

- Manifeste Versionsupdates und Datenaktualität
- Verhalten beim Ausführen des Caches
- Steuern Sie die Größe des Run-Caches

Manifeste Versionsupdates und Datenaktualität

In regelmäßigen Abständen führt der HealthOmics Dienst möglicherweise neue Funktionen oder Workflow-Engine-Updates ein, die einige oder alle Run-Cache-Einträge ungültig machen. In diesem Fall kann es bei Ihren Läufen zu einem einmaligen Cachefehler kommen.

HealthOmics erstellt für jeden Cache-Eintrag eine JSON-Manifestdatei. Für Läufe, die nach dem 12. Februar 2025 gestartet wurden, enthält die Manifestdatei einen Versionsparameter. Wenn ein Service-Update Cache-Einträge ungültig macht, wird die Versionsnummer HealthOmics erhöht, sodass Sie die Legacy-Cache-Einträge identifizieren können, die entfernt werden müssen.

Das folgende Beispiel zeigt eine Manifestdatei, deren Version auf 2 gesetzt ist:

```
{
     "arn": "arn:aws:omics:us-west-2:12345678901:runCache/0123456/
cacheEntry/1234567-195f-3921-a1fa-ffffcef0a6a4",
```

```
    "s3uri": "s3://example/1234567-d0d1-e230-
d599-10f1539f4a32/1348677/4795326/7e8c69b1-145f-3991-a1fa-ffffcef0a6a4",
    "taskArn": "arn:aws:omics:us-west-2:12345678901:task/4567891",
    "workDir": "/mnt/workflow/1234567-d0d1-e230-d599-10f1539f4a32/workdir/call-
TxtFileCopyTask/5w6tn5feyga7noasjuecdeoqpkltrfo3/wxz2fuddlo6hc4uh5s2lreaayczduxdm",
    "files": [
        {
            "name": "output_txt_file",
            "path": "out/output_txt_file/outfile.txt",
            "etag": "ajdhyg9736b9654673b9fbb486753bc8"
        }
    ],
    "nextflowContext": {},
    "otherOutputs": {},
    "version": 2,
  }
```

Bei Läufen mit Cacheeinträgen, die nicht mehr gültig sind, erstellen Sie den Cache neu, um neue gültige Einträge zu erstellen. Führen Sie für jeden Lauf die folgenden Schritte aus:

1. Starten Sie den Lauf einmal, wobei die Cache-Aufbewahrung auf CACHE ALWAYS eingestellt ist. Bei diesem Lauf werden die neuen Cache-Einträge erstellt.
2. Setzen Sie für nachfolgende Läufe die Cache-Aufbewahrung auf die vorherige Einstellung (CACHE ALWAYS oder CACHE ON FAILURE).

Um Cache-Einträge zu bereinigen, die nicht mehr gültig sind, können Sie diese Cache-Einträge aus dem Amazon S3 S3-Cache-Bucket löschen. HealthOmics verwendet diese Cache-Einträge niemals wieder. Wenn Sie sich dafür entscheiden, ungültige Einträge beizubehalten, hat dies keine Auswirkungen auf Ihre Läufe.

> **Note**
>
> Beim Aufruf-Caching werden die Ausgabedaten der Aufgaben an dem für den Cache angegebenen Amazon S3 S3-Speicherort gespeichert, wodurch Ihnen Gebühren entstehen. AWS-Konto

Verhalten beim Ausführen des Caches

Sie können das Verhalten beim Ausführen des Caches festlegen, um die Taskausgaben für fehlgeschlagene Läufe (Cache on Failure) oder für alle Läufe (Cache immer) zu speichern. Wenn Sie einen Run-Cache erstellen, legen Sie das Standard-Cache-Verhalten für alle Läufe fest, die diesen Cache verwenden. Sie können das Standardverhalten überschreiben, wenn Sie einen Lauf starten.

Cache on failureist nützlich, wenn Sie einen Workflow debuggen, der fehlschlägt, nachdem mehrere Aufgaben erfolgreich abgeschlossen wurden. Die nachfolgende Ausführung wird mit der letzten erfolgreich abgeschlossenen Aufgabe fortgesetzt, wenn alle vom Hash berücksichtigten eindeutigen Variablen mit der vorherigen Ausführung identisch sind.

Cache alwaysist nützlich, wenn Sie eine Aufgabe in einem Workflow aktualisieren, der erfolgreich abgeschlossen wurde. Wir empfehlen, dass Sie die folgenden Schritte ausführen:

1. Erstellen Sie einen neuen Lauf. Stellen Sie das Cache-Verhalten auf Immer zwischenspeichern ein und starten Sie die Ausführung.
2. Nachdem die Ausführung abgeschlossen ist, aktualisieren Sie die Aufgabe im Workflow und starten Sie eine neue Ausführung mit dem Verhalten „Immer zwischenspeichern“. Dieser Lauf verarbeitet die aktualisierte Aufgabe und alle nachfolgenden Aufgaben, die von der aktualisierten Aufgabe abhängig sind. Alle anderen Aufgaben verwenden die zwischengespeicherten Ergebnisse.
3. Wiederholen Sie Schritt 2 nach Bedarf, bis die Entwicklung für die aktualisierte Aufgabe abgeschlossen ist.
4. Verwenden Sie die aktualisierte Aufgabe bei future Läufen nach Bedarf. Denken Sie daran, nachfolgende Läufe im Fehlerfall auf Cache umzustellen, wenn Sie beabsichtigen, für diese Läufe neue oder andere Eingaben zu verwenden.

> **Note**
>
> Wir empfehlen den Modus Immer zwischenspeichern, wenn Sie denselben Testdatensatz verwenden, jedoch nicht für mehrere Läufe. Wenn Sie diesen Modus für eine große Anzahl von Durchläufen festlegen, kann das System große Datenmengen nach Amazon S3 exportieren, was zu erhöhten Exportzeiten und Speicherkosten führt.

Steuern Sie die Größe des Run-Caches

HealthOmics löscht oder archiviert keine Run-Cache-Daten und wendet auch keine Amazon S3 S3-Bereinigungsregeln für die Verwaltung der Cache-Daten an. Wir empfehlen Ihnen, regelmäßige Cache-Bereinigungen durchzuführen, um Amazon S3 S3-Speicherkosten zu sparen und die Größe Ihres Run-Caches überschaubar zu halten. Sie können Dateien direkt löschen oder retention/replication Datenrichtlinien für den Run-Cache-Bucket festlegen.

Sie können beispielsweise eine Amazon S3 S3-Lebenszyklusrichtlinie so konfigurieren, dass Objekte nach 90 Tagen ablaufen, oder Sie können die Cache-Daten am Ende jedes Entwicklungsprojekts manuell bereinigen.

Die folgenden Informationen können Ihnen bei der Verwaltung der Cache-Datengröße helfen:

- Sie können sehen, wie viele Daten sich im Cache befinden, indem Sie Amazon S3 überprüfen. HealthOmics überwacht oder berichtet nicht über die Cachegröße.
- Wenn Sie einen gültigen Cache-Eintrag löschen, schlägt die nachfolgende Ausführung nicht fehl. HealthOmics berechnet die Aufgabe und ihre abhängigen Aufgaben neu.
- Wenn Sie Cache-Namen oder Verzeichnisstrukturen so ändern, HealthOmics dass kein passender Eintrag für eine Aufgabe gefunden wird, HealthOmics berechnet die Aufgabe neu.

Wenn Sie überprüfen müssen, ob ein Cache-Eintrag noch gültig ist, überprüfen Sie die Versionsnummer des Cache-Manifests. Weitere Informationen finden Sie unter Manifeste Versionsupdates und Datenaktualität.

Einen Run-Cache erstellen

Wenn Sie einen Run-Cache erstellen, geben Sie einen Amazon S3 S3-Speicherort für die Cache-Daten an. Auf diese Daten muss sofort zugegriffen werden können. Beim Aufruf-Caching werden keine in Glacier archivierten Objekte abgerufen (z. B. GFR- und GDA-Speicherklassen).

Wenn der Amazon S3 S3-Bucket für die Cache-Daten einem anderen gehört AWS-Konto, geben Sie diese Konto-ID an, wenn Sie den Run-Cache erstellen.

Einen Run-Cache mithilfe der Konsole erstellen

Gehen Sie von der Konsole aus wie folgt vor, um einen Run-Cache zu erstellen.

1. Öffnen Sie die HealthOmics -Konsole.

2. Öffnen Sie bei Bedarf den linken Navigationsbereich (±1). Wählen Sie „Caches ausführen“.
3. Wählen Sie auf der Seite „Caches ausführen“ die Option Run-Cache erstellen aus.
4. Konfigurieren Sie auf der Seite „Run-Cache erstellen“ im Bereich „Details zum Ausführungs-Cache“ die folgenden Felder:

   a. Geben Sie einen Namen für den Run-Cache ein.
   b. (Optional) Geben Sie eine Beschreibung ein.
   c. Geben Sie einen S3-Speicherort für die zwischengespeicherte Ausgabe ein. Wählen Sie einen Bucket in derselben Region wie Ihr Workflow aus.
   d. (Optional) Geben Sie den Namen AWS-Konto des Bucket-Besitzers ein, um die Inhaberschaft des Buckets zu überprüfen. Wenn Sie keinen Wert eingeben, ist der Standardwert Ihre Konto-ID.
   e. Konfigurieren Sie unter Cache-Verhalten das Standardverhalten (ob Ausgaben für fehlgeschlagene Läufe oder für alle Läufe zwischengespeichert werden sollen). Wenn Sie einen Lauf starten, können Sie optional das Standardverhalten überschreiben.
5. (Optional) Ordnen Sie dem Run-Cache ein oder mehrere Tags zu.
6. Wählen Sie „Run-Cache erstellen“. Die Konsole zeigt den neuen Run-Cache in der Tabelle Run-Caches an.

Einen Run-Cache mit der CLI erstellen

Verwenden Sie den CLI-Befehl create-run-cache, um einen Run-Cache zu erstellen. Das Standard-Cache-Verhalten ist. `CACHE_ON_FAILURE`

```
aws omics create-run-cache \
      --name "workflow 123 run cache" \
      --description "my run cache" \
      --cache-s3-location "s3://amzn-s3-demo-bucket" \
      --cache-behavior "CACHE_ALWAYS"                \
      --cache-bucket-owner-id  "111122223333"
```

Wenn die Erstellung erfolgreich ist, erhalten Sie eine Antwort mit den folgenden Feldern.

```
{
  "arn": "string",
  "id": "string",
```

```
  "status": "ACTIVE"
  "tags": {}
  }
```

Einen Run-Cache aktualisieren

Sie können den Cache-Namen, die Beschreibung, die Tags oder das Cache-Verhalten ändern, aber nicht den S3-Speicherort für den Cache.

Einen Run-Cache mithilfe der Konsole aktualisieren

Gehen Sie von der Konsole aus wie folgt vor, um einen Run-Cache zu aktualisieren.

1. Öffnen Sie die HealthOmics -Konsole.
2. Öffnen Sie bei Bedarf den linken Navigationsbereich (±1). Wählen Sie „Caches ausführen“.
3. Wählen Sie in der Tabelle Run Caches den Run-Cache aus, der aktualisiert werden soll, und klicken Sie dann auf Bearbeiten.
4. Im Bereich „Details zum Ausführungs-Cache“ können Sie die Felder „Name“, „Beschreibung“ und „Cache-Verhalten“ aktualisieren.
5. (Optional) Ordnen Sie dem Run-Cache ein oder mehrere neue Tags zu oder entfernen Sie vorhandene Tags.
6. Wählen Sie „Run-Cache speichern“.

Aktualisierung eines Run-Caches mit der CLI

Verwenden Sie den CLI-Befehl update-run-cache, um einen Run-Cache zu aktualisieren.

```
aws omics update-run-cache \
      --name "workflow 123 run cache" \
      --id "workflow id" \
      --description "my run cache" \
      --cache-behavior "CACHE_ALWAYS"
```

Wenn das Update erfolgreich ist, erhalten Sie eine Antwort ohne Datenfelder.

Löschen eines Run-Caches

Sie können einen Run-Cache löschen, wenn ihn keine aktiven Läufe verwenden. Wenn Läufe den Run-Cache verwenden, warten Sie, bis die Läufe abgeschlossen sind, oder Sie können die Läufe abbrechen.

Durch das Löschen eines Run-Caches werden die Ressource und ihre Metadaten entfernt, die Daten in Amazon S3 werden jedoch nicht gelöscht. Nachdem Sie den Cache gelöscht haben, können Sie ihn nicht erneut anhängen oder ihn für nachfolgende Läufe verwenden.

Die zwischengespeicherten Daten verbleiben zu Ihrer Prüfung in Amazon S3. Sie können alte Cache-Daten mithilfe von Delete Standard-S3-Vorgängen entfernen. Alternativ können Sie eine Amazon S3 S3-Lebenszyklusrichtlinie erstellen, um zwischengespeicherte Daten, die Sie nicht mehr verwenden, ablaufen zu lassen.

Löschen eines Run-Caches mithilfe der Konsole

Gehen Sie von der Konsole aus wie folgt vor, um einen Run-Cache zu löschen.

1. Öffnen Sie die HealthOmics -Konsole.
2. Öffnen Sie bei Bedarf den linken Navigationsbereich (±1). Wählen Sie „Caches ausführen“.
3. Wählen Sie in der Tabelle Run Caches den Run-Cache aus, den Sie löschen möchten.
4. Wählen Sie im Tabellenmenü „Caches ausführen“ die Option Löschen aus.
5. Speichern Sie im modalen Dialog den Amazon S3 S3-Cache-Datenlink zum future Nachschlagen und bestätigen Sie dann, dass Sie den Run-Cache löschen möchten.

   Sie können den Amazon S3 S3-Link verwenden, um die zwischengespeicherten Daten zu überprüfen, aber Sie können die Daten nicht erneut mit einem anderen Run-Cache verknüpfen. Löschen Sie die Cache-Daten, wenn Sie die Inspektion abgeschlossen haben.

Löschen eines Run-Caches mit der CLI

Verwenden Sie den CLI-Befehl delete-run-cache, um einen Run-Cache zu löschen.

```
aws omics delete-run-cache \
      --id "my cache id"
```

Wenn das Löschen erfolgreich ist, erhalten Sie eine Antwort ohne Datenfelder.

Inhalt eines Run-Caches

HealthOmics organisiert Ihren Run-Cache mit der folgenden Struktur in Ihrem S3-Bucket:

```
s3://{cache.S3location}/{cache.uuid}/runID/taskID/{cacheentry.uuid}/
```

Die cache.uuid ist die weltweit eindeutige ID für den Cache. Die cacheentry.uuid ist die weltweit eindeutige UUID für eine zwischengespeicherte Aufgabe. HealthOmics weist die UUIDs Caches und Aufgaben zu.

Für alle Workflow-Engines enthält der Cache die folgenden Dateien:

- Die {cacheentryuuid}.json Datei — HealthOmics erstellt diese Manifestdatei, die Informationen über den Cache enthält, einschließlich einer Liste aller Elemente im Cache und der Cache-Version.
- Task-Ausgabedateien — Jede Task-Ausgabe besteht aus einer oder mehreren Dateien, wie von der Aufgabe definiert.

Für einen Workflow, der Nextflow verwendet, erstellt die Nextflow-Engine diese zusätzlichen Dateien im Cache:

- Die command.out Datei — Diese Datei enthält den Standardinhalt zur Aufgabenausführung.
- Die .exitcode Datei — Diese Datei enthält den Exit-Code der Aufgabe (eine Ganzzahl).

> **Note**
>
> Wenn Sie zur erweiterten Problembehandlung auf zwischengeschaltete Aufgabendateien in Ihrem Run-Cache zugreifen möchten, deklarieren Sie diese Dateien in der Workflow-Definition als Aufgabenausgaben.

Engine-specific Caching-Funktionen

HealthOmics versucht, eine konsistente Implementierung von Call Caching in allen Workflow-Engines bereitzustellen. Je nachdem, wie jede Workflow-Engine mit bestimmten Fällen umgeht, gibt es einige Unterschiede:

- Nextflow

- Das Zwischenspeichern zwischen verschiedenen Nextflow-Versionen ist nicht garantiert. Wenn du eine Aufgabe auf einer Nextflow-Version und dann dieselbe Aufgabe auf einer anderen Nextflow-Version ausführst, HealthOmics könnte es sein, dass der zweite Lauf ein Cache-Fehler ist.
- Sie können das Caching für einzelne Aufgaben deaktivieren, indem Sie die Cache-Direktive verwenden. false Informationen zu dieser Direktive finden Sie in den Prozessen in der Nextflow-Spezifikation.
- HealthOmics verwendet den Nextflow-Lenient-Modus, unterstützt aber keinen Deep-Caching-Modus.
- Beim Caching wird jedes einzelne S3-Objekt ausgewertet, wenn Sie ein Glob-Muster im S3-Pfad zu den Eingaben für eine Aufgabe verwenden. Wenn Sie ein neues Objekt hinzufügen, werden nur die Aufgaben HealthOmics neu berechnet, die das neue Objekt verwenden.
- HealthOmics zwischenspeichert keine Wiederholungsversuche von Aufgaben. Dieses Verhalten entspricht dem Standardverhalten von Nextflow.

- WDL
  - HealthOmics unterstützt den neuen Verzeichnistyp für Eingaben, wenn Sie die Entwicklungsversion des WDL-Workflows verwenden. Bei der Zwischenspeicherung von Aufrufen werden alle Aufgaben, die das Verzeichnis eingeben, HealthOmics neu berechnet, wenn sich ein Objekt im Verzeichnis ändert.
  - HealthOmics unterstützt Caching auf Aufgabenebene, aber kein Caching auf Workflow-Ebene.
  - Sie können das Caching für einzelne Aufgaben deaktivieren, indem Sie das Volatile-Attribut verwenden. Weitere Informationen finden Sie unter Deaktivieren Sie das Caching auf Aufgabenebene mit dem Attribut volatile.
- CWL
  - Konstante Ausgaben von Aufgaben sind in den Manifesten nicht explizit sichtbar. HealthOmics speichert konstante Ausgaben als Zwischendateien im Cache.
  - Sie können das Caching für einzelne Aufgaben mithilfe der WorkReuseFunktion steuern.

Verwenden Sie den Run-Cache

Standardmäßig verwenden Läufe keinen Run-Cache. Um einen Cache für die Ausführung zu verwenden, geben Sie den Run-Cache und das Verhalten des Run-Caches an, wenn Sie die Ausführung starten.

Nach Abschluss eines Laufs können Sie die Konsole, CloudWatch Protokolle oder API-Operationen verwenden, um Cache-Treffer zu verfolgen oder Cache-Probleme zu beheben. Details dazu finden Sie unter Informationen zum Verfolgen von Anrufen und Zwischenspeichern und Behebung von Problemen beim Zwischenspeichern von Anrufen.

Wenn eine oder mehrere Aufgaben in einem Lauf nicht deterministische Ausgaben generieren, empfehlen wir dringend, für die Ausführung kein Aufruf-Caching zu verwenden oder diese speziellen Aufgaben vom Caching auszuschließen. Weitere Informationen finden Sie unter Modell der geteilten Verantwortung.

> **Note**
>
> Sie geben eine IAM-Dienstrolle an, wenn Sie einen Lauf starten. Um Call Caching verwenden zu können, benötigt die Service-Rolle die Erlaubnis, auf den Amazon S3 S3-Standort Run Cache zuzugreifen. Weitere Informationen finden Sie unter Servicerollen für AWS HealthOmics.

Sie können Kiro CLI verwenden, um Ihre Run-Cache-Daten zu analysieren und zu verwalten. Weitere Informationen finden Sie unter Beispielaufforderungen für Kiro CLI und im HealthOmics Agentic Generative AI-Tutorial unter. GitHub

Themen

- Konfiguration eines Laufs mit Run-Cache über die Konsole
- Konfiguration eines Laufs mit Run-Cache über die CLI
- Fehlerfälle bei Run-Caches
- Zwischenspeichern bei abgebrochenen Läufen
- Informationen zum Verfolgen von Anrufen und Zwischenspeichern

Konfiguration eines Laufs mit Run-Cache über die Konsole

Von der Konsole aus konfigurieren Sie den Run-Cache für einen Lauf, wenn Sie den Lauf starten.

1. Öffnen Sie die HealthOmics -Konsole.
2. Öffnen Sie bei Bedarf den linken Navigationsbereich (±1). Wählen Sie Läufe aus.
3. Wählen Sie auf der Seite „Läufe“ den Lauf aus, den Sie starten möchten.

4. Wählen Sie Start run und führen Sie die Schritte 1 und 2 von Run starten aus, wie unter beschriebenEinen Lauf mit der Konsole starten.
5. Wählen Sie in Schritt 3 von Start Run die Option Existierenden Run-Cache auswählen aus.
6. Wählen Sie den Cache aus der Dropdownliste „Cache-ID ausführen“ aus.
7. Um das Standardverhalten beim Ausführen des Cache zu überschreiben, wählen Sie das Cache-Verhalten für den Lauf aus. Weitere Informationen finden Sie unter Verhalten beim Ausführen des Caches.
8. Fahren Sie mit Schritt 4 von Start Run fort.

Konfiguration eines Laufs mit Run-Cache über die CLI

Um einen Lauf zu starten, der einen Run-Cache verwendet, fügen Sie dem CLI-Befehl start-run den Parameter cache-id hinzu. Verwenden Sie optional den `cache-behavior` Parameter, um das Standardverhalten zu überschreiben, das Sie für den Run-Cache konfiguriert haben. Das folgende Beispiel zeigt nur die Cache-Felder für den Befehl:

```
aws omics start-run \
        ...
      --cache-id "xxxxxx"    \
      --cache-behavior  CACHE_ALWAYS
```

Wenn der Vorgang erfolgreich ist, erhalten Sie eine Antwort ohne Datenfelder.

Fehlerfälle bei Run-Caches

In den folgenden Szenarien werden Taskausgaben HealthOmics möglicherweise nicht zwischengespeichert, auch nicht bei einer Ausführung, bei der das Cacheverhalten auf Immer zwischenspeichern eingestellt ist.

- Wenn bei der Ausführung ein Fehler auftritt, bevor die erste Aufgabe erfolgreich abgeschlossen wurde, können keine Cache-Ausgaben exportiert werden.
- Wenn der Exportvorgang fehlschlägt, werden die Aufgabenausgaben HealthOmics nicht im Amazon S3 S3-Cache gespeichert.
- Wenn der Lauf aufgrund eines filesystem out of space Fehlers fehlschlägt, speichert das Aufruf-Caching keine Aufgabenausgaben.

- Wenn bei der Ausführung ein Timeout auftritt, speichert das Aufruf-Caching keine Aufgabenausgaben, auch wenn Sie den Lauf so konfiguriert haben, dass bei einem Fehler der Cache verwendet wird.

Zwischenspeichern bei abgebrochenen Läufen

Wenn Sie einen laufenden Workflow abbrechen, der einen Run-Cache verwendet, werden Cache-Einträge für Aufgaben HealthOmics exportiert, die vor dem Abbruch erfolgreich abgeschlossen wurden. Diese Cache-Einträge folgen demselben Abgleichs- und Wiederverwendungsverhalten wie Cache-Einträge aus abgeschlossenen Läufen. HealthOmics exportiert keine Ausgaben auf Workflow-Ebene für abgebrochene Läufe.

Dieses Verhalten gilt sowohl für das Verhalten „Immer zwischenspeichern“ als auch für das Cache-Verhalten „Bei Ausfall zwischenspeichern“ und wird für alle Workflow-Engines (WDL, CWL und Nextflow) unterstützt.

Informationen zum Verfolgen von Anrufen und Zwischenspeichern

Sie können Call-Caching-Ereignisse (z. B. Run-Cache-Treffer) mithilfe der Konsole, der CLI oder CloudWatch Logs verfolgen.

Themen

- Cache-Treffer mithilfe der Konsole verfolgen
- Verfolgen Sie das Anruf-Caching mit der CLI
- Verfolgen Sie das Caching von Anrufen mithilfe von Protokollen CloudWatch

Cache-Treffer mithilfe der Konsole verfolgen

Auf der Seite mit den Ausführungsdetails für einen Lauf werden in der Tabelle Aufgaben ausführen Informationen zu Cache-Treffern für jede Aufgabe angezeigt. Die Tabelle enthält auch einen Link zum zugehörigen Cache-Eintrag. Gehen Sie wie folgt vor, um die Cache-Trefferinformationen für einen Lauf anzuzeigen.

1. Öffnen Sie die HealthOmics -Konsole.
2. Öffnen Sie bei Bedarf den linken Navigationsbereich (±1). Wählen Sie Läufe aus.
3. Wählen Sie auf der Seite „Läufe“ den Lauf aus, den Sie überprüfen möchten.

4. Wählen Sie auf der Seite mit den Ausführungsdetails die Registerkarte Aufgaben ausführen, um die Tabelle mit den Aufgaben anzuzeigen.
5. Wenn eine Aufgabe einen Cache-Treffer hat, enthält die Spalte Cache-Treffer einen Link zum Speicherort des Run-Cache-Eintrags in Amazon S3.
6. Wählen Sie den Link, um den Run-Cache-Eintrag zu überprüfen.

Verfolgen Sie das Anruf-Caching mit der CLI

Überprüfen Sie mit dem CLI-Befehl get-run, ob der Lauf einen Aufrufcache verwendet hat.

```
aws omics get-run --id 1234567
```

Wenn das `cacheId` Feld in der Antwort gesetzt ist, verwendet die Ausführung diesen Cache.

Verwenden Sie den CLI-Befehl list-run-tasks, um den Cache-Datenspeicherort für jede zwischengespeicherte Aufgabe in der Ausführung abzurufen.

```
aws omics list-run-tasks --id 1234567
```

Wenn das Feld CacheHit für eine Aufgabe den Wert true hat, gibt das Feld caches3URI in der Antwort den Speicherort der Cache-Daten für diese Aufgabe an.

Sie können auch den CLI-Befehl get-run-task verwenden, um den Cache-Datenspeicherort für eine bestimmte Aufgabe abzurufen:

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

Verfolgen Sie das Caching von Anrufen mithilfe von Protokollen CloudWatch

HealthOmics erstellt Cache-Aktivitätsprotokolle in der `/aws/omics/WorkflowLog` CloudWatch Protokollgruppe. <cache_id><cache_uuid>Für jeden Run-Cache gibt es einen Log-Stream: RunCache//.

HealthOmics Generiert für Läufe, die Aufruf-Caching verwenden, CloudWatch Log-Einträge für die folgenden Ereignisse:

- einen Cache-Eintrag erstellen (CACHE_ENTRY_CREATED)

- Abgleichen eines Cache-Eintrags (CACHE_HIT)
- konnte mit einem Cache-Eintrag nicht übereinstimmen (CACHE_MISS)

Weitere Hinweise zu diesen Protokollen finden Sie unter. Loggt sich ein CloudWatch

Verwenden Sie die folgende CloudWatch Insights-Abfrage für die `/aws/omics/WorkflowLog` Protokollgruppe, um die Anzahl der Cache-Treffer pro Lauf für diesen Cache zurückzugeben:

```
filter @logStream like 'runCache/<CACHE_ID>/'
 fields @timestamp, @message
 filter logMessage like 'CACHE_HIT'
 parse "run: *," as run
 stats count(*) as cacheHits by run
```

Verwenden Sie die folgende Abfrage, um die Anzahl der Cache-Einträge zurückzugeben, die bei jedem Lauf erstellt wurden:

```
filter @logStream like 'runCache/<CACHE_ID>/'
 fields @timestamp, @message
 filter logMessage like 'CACHE_ENTRY_CREATED'
 parse "run: *," as run
 stats count(*) as cacheEntries by run
```

Teilen HealthOmics Workflows

Als Besitzer eines privaten Workflows können Sie den Workflow mit einem teilen AWS-Konto in derselben Region. Um einen Workflow mit mehreren Personen zu teilen AWS-Konto, erstellen Sie mehrere Shares desselben Workflows.

Als Besitzer können Sie den Zugriff auf einen geteilten Workflow widerrufen, indem Sie den Share löschen.

> **Note**
>
> HealthOmics ermöglicht einem gemeinsamen Workflow automatisch den Zugriff auf das Amazon ECR-Repository, während der Workflow im Konto des Abonnenten läuft. Für gemeinsame Workflows müssen Sie keinen zusätzlichen Repository-Zugriff gewähren.

Wenn du einen Workflow teilst, kann der Abonnent jede der Workflow-Versionen verwenden. Wenn Sie für einen gemeinsamen Workflow eine Zugriffskontrolle auf Versionsebene benötigen, empfehlen wir, separate Workflows zu erstellen, anstatt Workflow-Versionen zu verwenden.

⚠ Vertrauen Sie dem Workflow-Besitzer

Stellen Sie sicher, dass Sie dem Workflow-Besitzer vertrauen, bevor Sie eine Freigabe akzeptieren. Wenn Sie einen gemeinsamen Workflow ausführen, werden die Workflow-Definition und die Container-Images, die der Besitzer bereitgestellt hat, in Ihrem AWS-Konto. Sie werden mit den Berechtigungen der IAM-Servicerolle ausgeführt, die Sie für die Ausführung angeben. Der Inhalt des Workflows und seiner Container-Bilder sind für Sie nicht sichtbar. Das Ausführen eines gemeinsamen Workflows entspricht der Ausführung von Drittanbietercode in Ihrem Konto. Geben Sie eine Dienstrolle mit geringsten Rechten ein und verwenden Sie eine Sitzungsrichtlinie, um die Berechtigungen für die Ausführung weiter einzuschränken. Weitere Informationen finden Sie unter Verwenden Sie Sitzungsrichtlinien, um Berechtigungen einzuschränken.

Themen

- Einen gemeinsamen Workflow abonnieren
- Überwachung des Status einer Workflow-Freigabe
- Einen privaten Workflow teilen
- Akzeptieren eines gemeinsamen Workflows mithilfe der Konsole
- Einen gemeinsamen Workflow mithilfe der Konsole ausführen
- Ausführen eines gemeinsamen Workflows mithilfe der API

Einen gemeinsamen Workflow abonnieren

Um einen gemeinsamen Workflow zu abonnieren, folgen Sie diesen allgemeinen Schritten, um den Workflow zu akzeptieren und zu verwenden:

1. Verwenden Sie die Konsole oder die API, um das Teilen anzunehmen. Stellen Sie für Ihre aktuelle Region dieselbe Region ein, in der sich die Teilungsanfrage befindet.
   - Um die Teilungsanfrage in der Konsole zu finden, navigieren Sie zur Alle Ressourcen teilen Seite, dann wählen Sie die Mit mir geteilt Tab.

2. Verwenden Sie die Konsole oder API, um einen Lauf für den gemeinsamen Workflow zu erstellen.
   - Um die Workflow-Detailseite in der Konsole zu finden, navigieren Sie zu Mit mir geteilt (siehe Schritt 1), dann wähle Link zur Ressource für den gemeinsamen Workflow.
3. Sie geben Ihre eigenen Eingabedaten für den Workflow an.
4. Der gemeinsame Workflow läuft in Ihrem AWS-Konto.

Als Abonnent eines gemeinsamen Workflows verhindert das System, dass Sie die folgenden Workflow-Aktionen ausführen:

- Exportieren eines gemeinsamen Workflows
- Re-running der gemeinsame Arbeitsablauf
  - Sie erstellen einen neuen Lauf für den gemeinsamen Workflow.
- Re-sharing der Arbeitsablauf.
- Zuweisen eines Tags zum Workflow.
- Löschen des Workflows.
  - Wenn Sie den Workflow nicht mehr benötigen, löschen Sie den Workflow-Anteil.

Siehe Cross-account gemeinsame Nutzung von Ressourcen in AWS HealthOmics für weitere Informationen zur gemeinsamen Nutzung von Ressourcen.

Überwachung des Status einer Workflow-Freigabe

HealthOmics sendet ein Ereignis an EventBridge für jede Statusänderung einer Workflow-Freigabe. Wenn Sie Benachrichtigungen über bestimmte Statusänderungen erhalten möchten, richten Sie eine ein EventBridge Regel, die überwacht werden soll Workflow-Freigabe — Statusänderung Ereignisse. Beispiel:

- Sie möchten jedes Mal eine Benachrichtigung erhalten, wenn Sie eine Workflow-Freigabeanfrage erhalten, und jedes Mal, wenn ein Benutzer eine Workflow-Freigabe widerruft.
- Nachdem Sie eine Workflow-Freigabeanfrage initiiert haben, möchten Sie eine Benachrichtigung erhalten, wenn der Benutzer die Anfrage akzeptiert oder ablehnt.

Einzelheiten zur Verwendung von Ereignissen finden Sie unter Verwenden EventBridge mit AWS HealthOmics.

Einen privaten Workflow teilen

> **Note**
>
> Non-PTC Die Container-Unterstützung wird für geteilte Workflows nicht unterstützt. Wenn also ein Abonnent einen Workflow ausführt, den Sie geteilt haben, HealthOmics ignoriert Registrierungszuordnungen, die Folgendes beinhalten `credentialArn`. Weitere Informationen finden Sie unter Verwendung von Drittanbieter-Registrierungen, die vom Pull-Through-Cache nicht unterstützt werden.

Console

Sie können einen privaten Workflow mit einem teilen AWS-Konto in derselben Region wie der Workflow.

Um einen privaten Workflow zu teilen

1. Öffnen Sie die HealthOmics -Konsole.
2. Öffnen Sie bei Bedarf den linken Navigationsbereich (≡). Wählen Sie Private Workflows.
3. Aus dem Arbeitsabläufe Tabelle auf der Private Workflows Seite, wählen Sie den Workflow aus, den Sie teilen möchten, und wählen Sie Teilen.
4. In der Einzelheiten teilen Panel der Arbeitsablauf teilen Seite, geben Sie einen beschreibenden Namen für den Share ein und geben Sie den AWS-Konto des Abonnenten.
5. Wählen Sie Ressource teilen. Die Konsole zeigt Ressourcenfreigaben in der Alle gemeinsam genutzten Ressourcen Seite.

Der Ausgangszustand der Aktie ist `PENDING`. Nachdem der Abonnent die Aktie akzeptiert hat, ändert sich der Status zu `ACTIVE`.

CLI

Benutze das `create-share` AWS CLI Befehl, um eine Workflow-Freigabe zu erstellen. Der Hauptabonnent ist der AWS-Konto des Benutzers, der Zugriff auf den Workflow erhält.

```
aws omics create-share \
  --resource-arn "arn:aws:omics:us-west-2:555555555555:workflow/123456" \
  --principal-subscriber "123456789012" \
```

```
  --name "my_Share-123"
```

Wenn die Erstellung erfolgreich ist, erhalten Sie eine Antwort mit der Share-ID und dem Status.

```
{
"shareId": "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a",
"name": "my_Share-123",
"status": "PENDING"
}
```

Die Aktie bleibt in `PENDING` geben Sie an, bis der Abonnent es akzeptiert, und zwar mit dem `accept-share` AWS CLI Befehl.

Weitere Beispiele zur API-Nutzung finden Sie unter Cross-account gemeinsame Nutzung von Ressourcen in AWS HealthOmics.

Akzeptieren eines gemeinsamen Workflows mithilfe der Konsole

Sie können die Konsole verwenden, um eine angebotene Workflow-Freigabe anzunehmen. Stellen Sie sicher, dass die Konsole auf dieselbe Region wie der Workflow eingestellt ist.

1. Öffnen Sie die HealthOmics -Konsole.
2. Öffnen Sie bei Bedarf den linken Navigationsbereich (≡). Wählen Sie Alle Ressourcen teilen, wählen Sie dann Mit mir geteilt Tab.
3. Aus dem Mit mir geteilte Ressourcen Tabelle, wählen Sie den Workflow-Teilen aus und wählen Sie dann Akzeptieren.

Nachdem Sie den Workflow akzeptiert haben, wählen Sie Link zur Ressource damit der gemeinsame Workflow seine Details einsehen kann.

Einen gemeinsamen Workflow mithilfe der Konsole ausführen

Nachdem Sie eine Workflow-Freigabe akzeptiert haben, können Sie eine Ausführung des Workflows starten.

1. Öffnen Sie die HealthOmics -Konsole.
2. Öffnen Sie bei Bedarf den linken Navigationsbereich (≡). Wählen Sie Alle Ressourcen teilen, wählen Sie dann Mit mir geteilt Tab.

3. Aus dem Mit mir geteilte Ressourcen Tabelle, wählen Sie Link zur Ressource für den gemeinsamen Workflow.
4. In der Einzelheiten zum Arbeitsablauf Seite, wählen Sie Lauf erstellen.

   Die Konsole öffnet den Lauf erstellen Seite, mit dem Workflow-Typ (gemeinsam genutzt) und Workflow-ID vorab ausgefüllt.
5. Konfigurieren Sie die verbleibenden Felder in der Lauf erstellen Formular. Weitere Informationen finden Sie unter Einen Lauf mit der Konsole starten.

Ausführen eines gemeinsamen Workflows mithilfe der API

Verwenden Sie get-workflow, um den ARN des gemeinsamen Workflows abzurufen.

```
aws omics get-workflow --id 1234567 \
--workflow-owner-id 55555555555
```

Wenn Sie den Workflow ausführen, geben Sie die des Workflow-Besitzers an AWS-Konto ID und ARN des gemeinsam genutzten Workflows.

```
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
```

HealthOmics Workflows mit einer VPC verbinden

Mit Amazon Virtual Private Cloud (Amazon VPC) können Sie AWS Ressourcen in einem privaten, virtuellen Netzwerk starten, das Sie definiert haben. Sie können Ihren HealthOmics Workflows Zugriff auf Ressourcen in Ihrer VPC gewähren, indem Sie Ihre Läufe so konfigurieren, dass sie den VPC-Netzwerkmodus verwenden. Wenn das VPC-Netzwerk aktiviert ist, können Ihre Läufe auf Ressourcen in Ihrer VPC zugreifen und über das öffentliche Internet eine Verbindung zu externen Ressourcen herstellen, sofern für Ihre VPC der Internetzugang konfiguriert ist.

> **Note**
>
> Jeder HealthOmics Workflow-Lauf wird in einer VPC ausgeführt, die dem Dienst gehört und von diesem verwaltet wird. HealthOmics Diese VPCs werden automatisch verwaltet und sind

für Kunden nicht sichtbar. Die Konfiguration Ihres Laufs für den Zugriff auf Ressourcen in Ihrer Amazon VPC hat keine Auswirkungen auf die HealthOmics-managed VPC.

Wann sollte das VPC-Netzwerk verwendet werden

Verwenden Sie VPC-Netzwerke, wenn Ihre Läufe Folgendes benötigen:

- Greifen Sie über das Internet auf öffentlich verfügbare Datensätze zu (z. B. NIH-Datensätze, akademische Repositorien)
- Stellen Sie eine Verbindung zu Lizenzservern oder externen APIs von Drittanbietern her
- Lesen oder schreiben Sie Daten aus Amazon S3-Buckets in anderen Regionen AWS
- Greifen Sie auf lokale Ressourcen in Ihrem privaten Netzwerk zu
- Stellen Sie innerhalb Ihrer VPC eine Verbindung zu AWS-Ressourcen her

> **Note**
>
> Wenn Sie einen Lauf mit einer VPC verbinden, kann er nur auf Ressourcen zugreifen, die in dieser VPC verfügbar sind. Um deinem Lauf Zugriff auf das Internet zu gewähren, musst du deine VPC auch für den Internetzugang konfigurieren. Weitere Informationen finden Sie unter Internetzugang für VPC-connected Workflows.

Themen

- Netzwerkmodi
- Erste Schritte
- VPC-Anforderungen
- Service-linked Rolle für VPC-Netzwerke
- Verbindung zu einer VPC in einem anderen Konto herstellen
- Konfigurations-APIs
- Ausführen von Workflows mit VPC-Netzwerken
- Problembehandlung bei VPC-Netzwerken
- Bewährte Methoden
- Kontingente für VPC-Netzwerke

Netzwerkmodi

HealthOmics Workflows unterstützt zwei Netzwerkmodi. Standardmäßig werden Workflow-Ausführungen im RESTRICTED-Modus ausgeführt. Sie können das VPC-Netzwerk pro Lauf aktivieren, wenn Sie die Workflow-Ausführung starten.

EINGESCHRÄNKT (Standard)

Läufe können nur auf Amazon S3- und Amazon ECR-Ressourcen innerhalb derselben AWS Region zugreifen. Runs kann nicht auf andere AWS Dienste, AWS regionsübergreifende Ressourcen oder das öffentliche Internet zugreifen.

VPC

Der Laufverkehr wird über Elastic Network Interfaces (ENIs) geleitet, die HealthOmics in Ihren VPC-Subnetzen bereitgestellt werden. Sie steuern das Netzwerk-Routing, die Sicherheitsgruppen, Netzwerk-ACLs und den Internetzugang über NAT-Gateways. Dieser Modus ermöglicht den Zugriff auf:

- Öffentliche Internetressourcen (erfordert eine NAT-Gateway-Konfiguration)
- AWS Dienste in anderen Regionen
- Private Ressourcen in Ihrer VPC
- Greifen Sie auf lokale Ressourcen in Ihrem privaten Netzwerk zu

Sie geben den Netzwerkmodus an, wenn Sie eine Workflow-Ausführung mithilfe des `networkingMode` Parameters in der `StartRun` API starten.

Erste Schritte

In diesem Abschnitt erfahren Sie, wie Sie das VPC-Netzwerk für HealthOmics Workflows zum ersten Mal einrichten.

Voraussetzungen

Bevor Sie das VPC-Netzwerk für HealthOmics Workflows konfigurieren, stellen Sie sicher, dass Sie über Folgendes verfügen:

- Eine vorhandene VPC mit geeigneten Subnetzen und Sicherheitsgruppen. Die VPC muss sich in derselben Region wie Ihre Workflows befinden.
- Mindestens ein Subnetz in einer Availability Zone, das in Ihrer HealthOmics Region betrieben wird.

- Entsprechende IAM-Berechtigungen zum Erstellen und Verwalten von HealthOmics Konfigurationen.
- Verständnis der VPC-Netzwerkkonzepte (Subnetze, Sicherheitsgruppen, Routing-Tabellen).
- Ausreichende ENI-Kapazität in Ihrem Konto AWS . HealthOmics skaliert und verwaltet ENIs in Ihrer VPC mithilfe der serviceverknüpften Rolle. Die Anzahl der erforderlichen ENIs hängt von Ihrer Arbeitslast ab. Überwachen Sie Ihre ENI-Nutzung in der Amazon EC2-Konsole, um sicherzustellen, dass Sie über ausreichende Kapazitäten verfügen.

> **Important**
>
> Ihre VPC-Konfiguration muss mindestens ein Subnetz in einer Availability Zone enthalten, die in Ihrer Region HealthOmics betrieben wird, um die Platzierung von Workflow-Aufgaben zu unterstützen. Wenn Sie den VPC-Netzwerkmodus verwenden, sind Sie dafür verantwortlich, festzustellen, ob die Übertragung oder Nutzung von Daten zwischen Regionen sicher und konform ist. AWS

Schritt 1: Erstellen oder konfigurieren Sie Ihre VPC

Erstellen Sie eine VPC mit privaten Subnetzen, Sicherheitsgruppen und NAT-Gateways (falls Internetzugang erforderlich ist). Eine ausführliche schrittweise Anleitung finden Sie unter. Internetzugang für VPC-connected Workflows

Schritt 2: Sicherheitsgruppen konfigurieren

Erstellen Sie eine Sicherheitsgruppe, die ausgehenden Datenverkehr zu den Zielen ermöglicht, auf die Ihre Läufe zugreifen müssen. Konfigurieren Sie Sicherheitsgruppen so, dass nach dem Prinzip der geringsten Zugriffsrechte nur der minimal erforderliche ausgehende Datenverkehr zugelassen wird.

Beispielkonfigurationen und ausführliche Anleitungen finden Sie im Internetzugang für VPC-connected Workflows Abschnitt zur Sicherheitsgruppe unter.

Schritt 3: Routentabellen verifizieren

Stellen Sie sicher, dass Ihre privaten Subnetze über Routen zu einem NAT-Gateway für den Internetzugang verfügen. Beispiele für Routing-Tabellen-Konfigurationen finden Sie im Internetzugang für VPC-connected Workflows Abschnitt zur Routing-Tabelle unter.

> **Note**
>
> Wenn ein Lauf mit einem öffentlichen Subnetz verbunden wird, erhält er weder Internetzugang noch eine öffentliche IP-Adresse. Verwenden Sie immer private Subnetze mit NAT-Gateway-Routen für Läufe, die eine Internetverbindung erfordern.

Schritt 4: Erstellen Sie eine Konfigurationsressource

Erstellen Sie eine HealthOmics Konfigurationsressource, die Ihre VPC-Netzwerkeinstellungen definiert:

```
aws omics create-configuration \
  --name my-vpc-config \
  --description "VPC configuration for genomics workflows" \
  --run-configurations '{
    "vpcConfig": {
      "securityGroupIds": ["sg-0123456789abcdef0"],
      "subnetIds": [
        "subnet-0a1b2c3d4e5f6g7h8",
        "subnet-1a2b3c4d5e6f7g8h9"
      ]
    }
  }' \
  --region us-west-2
```

Sobald die Netzwerkressourcen bereitgestellt sind`CREATING`, geht die Konfiguration vom `ACTIVE` Status auf den Status über. Dies dauert bis zu 15 Minuten.

Schritt 5: Starten Sie einen Workflow, der mit VPC-Netzwerken ausgeführt wird

Sobald Ihre Konfiguration abgeschlossen ist`ACTIVE`, starten Sie einen Workflow-Lauf mit aktiviertem VPC-Netzwerk:

```
aws omics start-run \
  --workflow-id 1234567 \
  --role-arn arn:aws:iam::123456789012:role/OmicsWorkflowRole \
  --output-uri s3://my-bucket/outputs/ \
  --networking-mode VPC \
  --configuration-name my-vpc-config \
```

```
  --region us-west-2
```

Schritt 6: Überprüfen Sie die Konnektivität

Überwachen Sie Ihren Workflow-Lauf, um sicherzustellen, dass er auf die erforderlichen externen Ressourcen zugreifen kann. Überprüfen Sie die Workflow-Protokolle in den CloudWatch Protokollen auf Erfolgs- oder Fehlermeldungen der Verbindung. Eine ausführliche Anleitung zum Testen der Konnektivität finden Sie unterTesten der VPC-Konnektivität.

VPC-Anforderungen

Ihre VPC muss die folgenden Anforderungen erfüllen:

Anforderungen an das Subnetz

- Minimum: Mindestens ein Subnetz in einer Availability Zone, in der es funktioniert HealthOmics
- Maximum: 16 Subnetze pro Konfiguration
- Einschränkung: Maximal ein Subnetz pro Availability Zone
- Empfehlung: Verwenden Sie private Subnetze mit NAT-Gateway-Routen für Läufe, die einen Internetzugang erfordern. Sie können zwar ein einzelnes Subnetz angeben, wir empfehlen jedoch, für eine bessere Verfügbarkeit mehrere Subnetze in verschiedenen Availability Zones zu verwenden.

Anforderungen an Sicherheitsgruppen

- Mindestens: 1 Sicherheitsgruppe
- Maximum: 5 Sicherheitsgruppen pro Konfiguration
- Anforderung: Alle Sicherheitsgruppen müssen derselben VPC angehören wie die Subnetze

Sicherheitsgruppen kontrollieren den eingehenden und ausgehenden Verkehr für Ihre Läufe.

> **Note**
>
> Alle Subnetze und Sicherheitsgruppen müssen derselben VPC angehören.

Anforderungen an die Netzwerkschnittstelle

HealthOmics stellt elastische Netzwerkschnittstellen (ENIs) in Ihrer VPC bereit, um Läufe mit Ihrem Netzwerk zu verbinden. Stellen Sie sicher, dass Ihr AWS Konto über ausreichende ENI-Kapazität verfügt (Standardlimit: 5.000 ENIs pro Region).

ENIs, die von erstellt wurden, HealthOmics sind mit den folgenden Tags gekennzeichnet:

```
"TagSet": [
  {
    "Key": "Service",
    "Value": "HealthOmics"
  },
  {
    "Key": "eniType",
    "Value": "CUSTOMER"
  }
]
```

> **Important**
>
> Ändern oder löschen Sie keine ENIs, die von erstellt wurden. HealthOmics Das Ändern dieser Netzwerkschnittstellen kann zu Serviceverzögerungen oder Unterbrechungen Ihrer Workflow-Ausführungen führen.

Service-linked Rolle für VPC-Netzwerke

HealthOmics verwendet eine serviceverknüpfte IAM-Rolle, um elastische Netzwerkschnittstellen (ENIs) in Ihrer VPC zu erstellen und zu verwalten. Diese Rolle wird automatisch erstellt, wenn Sie Ihre erste VPC-Konfiguration erstellen.

Service-linked Einzelheiten zur Rolle

- Name der Rolle: `AWSServiceRoleForHealthOmics`
- Verwaltete Richtlinie: `AWSHealthOmicsServiceLinkedRolePolicy`
- Leiter des Dienstes: `omics.amazonaws.com`

Genehmigungen erteilt

Die serviceverknüpfte Rolle gewährt HealthOmics die folgenden Amazon EC2-Berechtigungen zur Verwaltung von Netzwerkschnittstellen in Ihrer VPC:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowEC2DescribeActions",
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeSubnets",
        "ec2:DescribeTags",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeSecurityGroupRules",
        "ec2:DescribeVpcs",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DescribeAvailabilityZones"
      ],
      "Resource": "*"
    },
    {
      "Sid": "AllowVpcGetActions",
      "Effect": "Allow",
      "Action": [
        "ec2:GetSecurityGroupsForVpc"
      ],
      "Resource": "arn:aws:ec2:*:*:vpc/*"
    },
    {
      "Sid": "AllowCreateNetworkInterfaceWithTag",
      "Effect": "Allow",
      "Action": "ec2:CreateNetworkInterface",
      "Resource": "arn:aws:ec2:*:*:network-interface/*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/Service": "HealthOmics"
        }
      }
    },
    {
      "Sid": "AllowCreateNetworkInterfaceSubnetSecurityGroup",
```

```
      "Effect": "Allow",
      "Action": "ec2:CreateNetworkInterface",
      "Resource": [
        "arn:aws:ec2:*:*:subnet/*",
        "arn:aws:ec2:*:*:security-group/*"
      ]
    },
    {
      "Sid": "AllowCreateTags",
      "Effect": "Allow",
      "Action": "ec2:CreateTags",
      "Resource": "arn:aws:ec2:*:*:network-interface/*",
      "Condition": {
        "StringEquals": {
          "ec2:CreateAction": "CreateNetworkInterface"
        }
      }
    },
    {
      "Sid": "AllowDeleteNetworkInterface",
      "Effect": "Allow",
      "Action": "ec2:DeleteNetworkInterface",
      "Resource": "arn:aws:ec2:*:*:network-interface/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Service": "HealthOmics"
        }
      }
    },
    {
      "Sid": "AllowAssignUnassignPrivateIpAddresses",
      "Effect": "Allow",
      "Action": [
        "ec2:AssignPrivateIpAddresses",
        "ec2:UnassignPrivateIpAddresses"
      ],
      "Resource": "arn:aws:ec2:*:*:network-interface/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Service": "HealthOmics"
        }
      }
    }
  ]
```

```
}
```

Erstellen der serviceverknüpfte -Rolle

Die serviceverknüpfte Rolle wird automatisch erstellt, wenn Sie Ihre erste VPC-Konfiguration mithilfe der API erstellen. `CreateConfiguration` Sie müssen über die folgende IAM-Berechtigung verfügen:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iam:CreateServiceLinkedRole",
      "Resource": "arn:aws:iam::*:role/aws-service-role/omics.amazonaws.com/
AWSServiceRoleForHealthOmics",
      "Condition": {
        "StringEquals": {
          "iam:AWSServiceName": "omics.amazonaws.com"
        }
      }
    }
  ]
}
```

Löschen der serviceverknüpften -Rolle

Sie können die dienstverknüpfte Rolle erst löschen, nachdem Sie alle VPC-Konfigurationen in Ihrem Konto gelöscht haben. Um die Rolle zu löschen:

```
aws iam delete-service-linked-role \
  --role-name AWSServiceRoleForHealthOmics
```

Wenn Sie aktive VPC-Konfigurationen haben, schlägt das Löschen fehl und es wird eine Meldung angezeigt, die angibt, welche Konfigurationsressourcen zuerst gelöscht werden müssen.

Verbindung zu einer VPC in einem anderen Konto herstellen

Du kannst deine geben HealthOmics Der Workflow ermöglicht den Zugriff auf Ressourcen in einer Amazon-VPC, die zu einer anderen gehört AWS Konto. Bei der gemeinsamen Nutzung von AWS Resource Access Manager (AWS RAM) teilt sich der VPC-Besitzer Subnetze mit Ihrem Konto und

HealthOmics platziert die Elastic Network Interfaces (ENIs) für Ihre Läufe in diesen Subnetzen. Beim VPC-Peering bleiben die ENIs in Ihrer eigenen VPC, und eine Peering-Verbindung leitet den Datenverkehr an die VPC im anderen Konto weiter. Für die gemeinsame Nutzung von RAM müssen beide Konten derselben Organisation angehören AWS Für Organisationen gibt es beim Peering keine derartigen Anforderungen und es werden Runs, die nur bestimmte Ressourcen erreichen müssen, auf dem anderen Konto berücksichtigt.

> **Note**
>
> Die Amazon-VPC und andere Ressourcen, die Sie für diese Methoden erstellen, können zu Gebühren für Sie führen AWS Konto. Bevor Sie den Zugriff zwischen Konten zulassen, vergewissern Sie sich, dass Ihr Tarif die Sicherheitsanforderungen der Organisationen erfüllt, die beide Konten verwalten.

Themen

- Die wichtigsten Konzepte
- Verwenden von RAM-Sharing
- Verwenden von VPC-Peering
- Bewährte Methoden
- Weitere Ressourcen

Die wichtigsten Konzepte

Wählen Sie zwischen RAM-Sharing und VPC-Peering, je nachdem, wo Ihr Workflow ausgeführt werden muss. Bei RAM-Sharing teilt sich der VPC-Besitzer Subnetze (und optional Sicherheitsgruppen) mit einem Teilnehmer desselben AWS Organisation und HealthOmics platziert elastische Netzwerkschnittstellen (ENIs) direkt in der gemeinsam genutzten VPC. Beim VPC-Peering leiten der Anforderer und der Akzeptierende den Datenverkehr zwischen ihren VPCs weiter. Die ENIs verbleiben in der VPC des Anforderers, und die Peering-Verbindung leitet den Datenverkehr an die VPC des Empfängers weiter. Beide Ansätze verwenden dasselbe HealthOmics Konfigurationsressource. Nur RAM-Sharing legt fest `vpcOwnerAccountId` Parameter.

Die folgenden Begriffe kennzeichnen die Konten in den einzelnen Ansätzen:

- VPC-Besitzer (RAM-Sharing): Das Konto, das die VPC besitzt und Subnetze (und optional Sicherheitsgruppen) gemeinsam nutzt. Bei diesem Ansatz ist das andere Konto der VPC-Besitzer.

- Teilnehmer (RAM-Sharing): Das Konto, das läuft HealthOmics arbeitet und empfängt gemeinsam genutzte Subnetze und Sicherheitsgruppen vom VPC-Besitzer. Bei diesem Ansatz ist Ihr Konto der Teilnehmer.
- Anforderer (VPC-Peering): Das Konto, das die VPC-Peering-Verbindung initiiert; die ENIs des Workflow-Laufs bleiben hier. Bei diesem Ansatz ist Ihr Konto der Anforderer.
- Akzeptierer (VPC-Peering): Das Konto, das die Peering-Verbindungsanfrage akzeptiert und dessen VPC die Ressource enthält, die Sie erreichen möchten. Bei diesem Ansatz ist das andere Konto der Akzeptierer.

Auswahl des Ansatzes

In der folgenden Tabelle sind die Unterschiede zwischen den beiden Ansätzen zusammengefasst.

Überlegungen	Gemeinsame Nutzung von RAM	VPC-Peering
AWS Organisationen erforderlich	Ja. Beide Konten müssen sich in derselben Organisation befinden.	Nein
Anwendungsfall	Der Workflow muss in einer gemeinsamen VPC (zentralisierter Ausgang, zentrales Netzwerkkonto) ablaufen	Der Workflow muss nur Ressourcen in der VPC eines anderen Kontos erreichen
Überlegungen zur Skalierung	Vorbehaltlich der AWS-RAM-Servicekontingente (standardmäßig 25.000 Ressourcenanteile pro AWS Region mit bis zu 5.000 Ressourcen und 5.000 Auftraggebern pro Aktie)	Vorbehaltlich der Kontingente für den VPC-Peering-Verbindungsdienst (standardmäßig 50 aktive Peering-Verbindungen pro VPC). Wenn Sie erwarten, dieses Limit zu überschreiten, sollten Sie stattdessen die gemeinsame Nutzung von RAM in Betracht ziehen.

Wenn Sie Ihre Anforderungen bereits kennen, folgen Sie der folgenden Anleitung:

- Wenn sich die Konten nicht in derselben Organisation befinden, verwenden Sie VPC-Peering.
- Wenn der Workflow in der VPC des anderen Kontos ausgeführt werden muss, verwenden Sie RAM-Sharing.
- Wenn Sie mehr als 50 aktive Peering-Verbindungen auf einer VPC erwarten, verwenden Sie RAM-Sharing.

Verwenden von RAM-Sharing

Verwenden Sie dieses Verfahren, um Ihre HealthOmics Der Workflow führt den Zugriff auf Subnetze (und optional Sicherheitsgruppen) durch, die ein anderer AWS Das Konto wird mithilfe von AWS Resource Access Manager (AWS RAM) mit Ihnen geteilt. Der VPC-Besitzer teilt Subnetze mit Ihrem (Teilnehmer-) Konto und HealthOmics platziert die Elastic Network Interfaces (ENIs) Ihres Workflow-Laufs direkt in den gemeinsam genutzten Subnetzen. Verwenden Sie diesen Ansatz für zentralisierte Netzwerkmodelle, z. B. eine gemeinsam genutzte Egress-VPC, die von einem zentralen Netzwerkkonto verwaltet wird.

Weitere Informationen zur gemeinsamen Nutzung von VPC-Ressourcen finden Sie unter Teilen Sie Ihre VPC-Subnetze mit anderen Konten und Sicherheitsgruppen teilen mit AWS Organisationen im Amazon VPC-Benutzerhandbuch.

Voraussetzungen

Um einen zu verwenden RAM-shared VPC mit HealthOmics, du musst haben:

- Sowohl das VPC-Besitzerkonto als auch Ihr (Teilnehmer-) Konto befinden sich im selben AWS Organisation.
- Die `AWSServiceRoleForHealthOmics` dienstgebundene Rolle in beides Konten. HealthOmics erstellt diese Rolle automatisch in Ihrem Konto, wenn Sie Ihre erste Konfiguration erstellen. HealthOmics kann die Rolle nicht im VPC-Besitzerkonto erstellen, da dies ein kontoübergreifender Vorgang wäre. Wenden Sie sich an den VPC-Besitzer, um zu bestätigen, dass die Rolle in seinem Konto vorhanden ist, oder bitten Sie ihn, sie zu erstellen.
- Eine VPC mit Subnetzen im VPC-Besitzerkonto, die Ihren Konnektivitätsanforderungen entspricht (z. B. NAT-Gateway für den Internetzugang). Subnetze müssen sich in Availability Zones befinden HealthOmics arbeitet in der AWS Region. Sie können Subnetze in einer Standard-VPC nicht gemeinsam nutzen. Nur Subnetze in einer nicht standardmäßigen VPC können mit AWS-RAM geteilt werden.

- Eine Sicherheitsgruppe in der gemeinsam genutzten VPC. Die Sicherheitsgruppe kann entweder dem VPC-Eigentümerkonto gehören (und mit Ihrem Konto mit AWS RAM geteilt werden) oder Ihrem (Teilnehmer-) Konto.
- Erlaubnis zur Erstellung von AWS-RAM-Ressourcenanteilen im VPC-Besitzerkonto.
- Erlaubnis zum Erstellen HealthOmics Konfigurationsressourcen in Ihrem Konto.
- Erlaubnis zum Starten HealthOmics Der Arbeitsablauf wird ausgeführt.

Schritt 1: Erstellen Sie die VPC und teilen Sie Ressourcen (VPC-Besitzerkonto)

Führen Sie die folgenden Schritte im VPC-Besitzerkonto aus. Sie benötigen die Erlaubnis, AWS-RAM-Ressourcenfreigaben zu erstellen. Um das Teilen mit zu ermöglichen AWS Organisationen melden sich mit dem Verwaltungskonto der Organisation bei der AWS RAM-Konsole an. Weitere Informationen finden Sie unter Aktiviere die gemeinsame Nutzung von Ressourcen in AWS Organisationen im AWS RAM-Benutzerhandbuch.

Bevor Sie beginnen, erstellen Sie die VPC, Subnetze, Sicherheitsgruppen und alle erforderlichen Verbindungen (z. B. ein NAT-Gateway) im VPC-Besitzerkonto. Weitere Informationen zur VPC-Setup finden Sie unter HealthOmics Workflows mit einer VPC verbinden.

Um das Teilen zu aktivieren und eine gemeinsame Nutzung von Ressourcen zu erstellen

1. Wenn die Konten nicht identisch sind AWS Organisation, erstelle eine Organisation und lade das Teilnehmerkonto ein.
2. Aktiviere vom Verwaltungskonto aus die gemeinsame Nutzung von Ressourcen mit AWS Organisationen:

   ```
   aws ram enable-sharing-with-aws-organization
   ```

   Mitgliedskonten erhalten dann gemeinsam genutzte Ressourcen, ohne eine Einladung anzunehmen.
3. Öffnen Sie das AWS-RAM-Konsole.
4. Wählen Sie Create resource share (Ressourcenfreigabe erstellen) aus.
5. Für Ressourcen, fügen Sie die Subnetze hinzu, die Ihre Läufe verwenden werden. Fügen Sie optional eine Sicherheitsgruppe hinzu.
6. Für Schulleiter, geben Sie die Konto-ID des Teilnehmers oder die Organisations-ID ein.
7. Wählen Sie Create resource share (Ressourcenfreigabe erstellen) aus.

Weitere Informationen finden Sie unter Erstellen einer Ressourcenfreigabe im AWS-RAM im AWS RAM-Benutzerhandbuch.

Schritt 2: Akzeptieren Sie die Freigabe und erstellen Sie eine Konfiguration (Teilnehmerkonto)

Um Ihre Workflows in der gemeinsam genutzten VPC auszuführen, führen Sie die folgenden Schritte in dem Konto aus, auf dem Ihre ausgeführt wird HealthOmics Workflows.

So nehmen Sie eine Einladung zu geteilten Ressourcen an

1. Öffnen Sie das AWS-RAM-Konsole.
2. Wählen Sie im Navigationsbereich Mit mir geteilt, dann Anteile an Ressourcen.
3. Wenn eine Einladung zur gemeinsamen Nutzung von Ressourcen aussteht, wählen Sie die Einladung aus und wählen Sie Akzeptiere die gemeinsame Nutzung von Ressourcen.

   > **Note**
   >
   > Wenn der VPC-Besitzer das Teilen mit aktiviert hat AWS Organisationen, es wird keine Einladung angezeigt. Die gemeinsam genutzten Ressourcen sind automatisch in Ihrem Konto verfügbar.

4. Öffnen Sie das Amazon VPC-Konsole und stellen Sie sicher, dass die gemeinsam genutzten Subnetze (und Sicherheitsgruppen, falls gemeinsam genutzt) in Ihrem Konto sichtbar sind.

Erstellen Sie das HealthOmics Konfiguration. Stellen Sie ein `vpcOwnerAccountId` auf die 12-stellige Konto-ID des VPC-Besitzers und geben Sie eine oder mehrere gemeinsam genutzte Subnetz- und Sicherheitsgruppen-IDs an.

Konsole

In der HealthOmics Erstellen Sie in der Konsole eine Konfiguration mit dem folgenden JSON:

```
{
  "name": "cross-account-ram-config",
  "runConfigurations": {
    "vpcConfig": {
      "securityGroupIds": ["sg-1234567890abcdef0"],
      "subnetIds": ["subnet-1234567890abcdef0"],
      "vpcOwnerAccountId": "111122223333"
    }
```

```
  }
}
```

AWS CLI

```
aws omics create-configuration \
  --name cross-account-ram-config \
  --run-configurations '{
    "vpcConfig": {
      "securityGroupIds": ["sg-1234567890abcdef0"],
      "subnetIds": ["subnet-1234567890abcdef0"],
      "vpcOwnerAccountId": "111122223333"
    }
  }'
```

Nach dem Erstellen der Konfiguration:

1. Warten Sie, bis der Konfigurationsstatus wie folgt lautet `ACTIVE`. Dieser Vorgang kann bis zu 15 Minuten dauern.
2. Starten Sie Ihren Workflow-Lauf im Netzwerkmodus, der auf eingestellt ist `VPC` und geben Sie den Namen der Konfiguration an, die Sie erstellt haben.

Weitere Informationen zum VPC-Netzwerk finden Sie unter HealthOmics Workflows mit einer VPC verbinden.

Verifizierung

Vergewissern Sie sich, dass die RAM-Sharing korrekt konfiguriert ist:

- Der HealthOmics Der Konfigurationsstatus wird angezeigt `ACTIVE` (Dies kann nach der Erstellung bis zu 15 Minuten dauern).
- Die gemeinsam genutzten Subnetze (und Sicherheitsgruppen, falls gemeinsam genutzt) sind in der Amazon VPC-Konsole des Teilnehmerkontos sichtbar.
- Ein Workflow-Lauf begann mit `networkingMode` gesetzt auf `VPC` und der RAM-shared Die Konfiguration wird ohne Netzwerkfehler abgeschlossen, und die ENIs werden in den gemeinsam genutzten Subnetzen angezeigt.

Fehlerbehebung bei der gemeinsamen Nutzung von RAM

Allgemeine AWS-RAM-Probleme finden Sie unter Fehlerbehebung bei AWS Resource Access Manager im AWS RAM-Benutzerhandbuch.

Problem	Häufige Ursache	Auflösung
Die gemeinsam genutzten Subnetze werden nicht in Ihrem Konto angezeigt	In der Ressourcenfreigabe sind Ihr Konto oder Ihre Organisation nicht enthalten , Sie haben die Einladung nicht angenommen oder Sie befinden sich in einem anderen AWS Region als die Aktie.	1. Prüfen Sie in der AWS RAM-Konsole, ob eine Einladung zur Nutzung von Ressourcen aussteht, und akzeptieren Sie sie. 2. Bestätigen Sie, dass Sie sich in derselben befinden AWS Region wie die gemeinsam genutzten Subnetze. 3. Bitten Sie den VPC-Besitzer, zu bestätigen, dass die Ressourcenfreigabe Ihre Konto-ID oder Ihre Organisation enthält.
Die Konfiguration wird nicht AKTIV oder der Lauf kann nicht gestartet werden	Die `vpcOwnerAccountId` der Wert ist falsch, oder `AWSServiceRoleForHealthOmics` Die serviceverknüpfte Rolle ist im VPC-Besitzerkonto nicht vorhanden.	1. Bestätigen Sie das `vpcOwnerAccountId` ist die 12-stellige Konto-ID des VPC-Besitzers, nicht Ihre eigene. 2. Bitten Sie den VPC-Besitzer, zu bestätigen, dass `AWSServiceRoleForHealthOmics` Die Rolle ist in ihrem Konto vorhanden, oder um sie zu erstellen.

Problem	Häufige Ursache	Auflösung
Ein Lauf kann die Ressource in der gemeinsam genutzten VPC nicht erreichen	Eine Sicherheitsgruppe oder Route in der gemeinsam genutzten VPC blockiert den Datenverkehr.	1. Vergewissern Sie sich, dass die Sicherheitsgruppe, die Sie in der Konfiguration ausgewählt haben, den Datenverkehr auf dem Port der Ressource zulässt. 2. Bitten Sie den VPC-Besitzer, zu bestätigen, dass die Routing-Tabellen für die gemeinsam genutzten Subnetze die Konnektivität bieten, die Ihre Läufe benötigen, z. B. eine Route zu einem NAT-Gateway. 3. Aktivieren Sie VPC Flow Logs in der gemeinsam genutzten VPC, um zu sehen, ob der Datenverkehr akzeptiert oder abgelehnt wird.
Der VPC-Besitzer kann ein gemeinsam genutztes Subnetz oder eine VPC nicht löschen, nachdem er eine Freigabe beendet hat	HealthOmics ENIs bleiben mit den gemeinsam genutzten Subnetzen verbunden. HealthOmics entfernt sie, nachdem sie 48 Stunden lang nicht verwendet wurden.	Warte bis HealthOmics entfernt die ENIs und löscht dann die Subnetze oder die VPC.

Stoppen einer RAM-Freigabe

Als VPC-Besitzer können Sie die gemeinsame Nutzung Ihrer Subnetze für ein Teilnehmerkonto beenden, wenn der Teilnehmer keinen Zugriff mehr benötigt, oder um den Zugriff aus Sicherheits- oder Compliance-Gründen zu entziehen. Um eine gemeinsame Nutzung zu beenden, löschen Sie die

AWS-RAM-Ressourcenfreigabe (oder entfernen Sie den Teilnehmer als Principal) in der AWS-RAM-Konsole.

Wenn Sie eine Freigabe beenden, HealthOmics schützt Sie davor, die Kontrolle über Ihre VPC zu verlieren:

- Der Teilnehmer kann keine Konfigurationen mehr verwenden, die auf die nicht gemeinsam genutzten Ressourcen verweisen, um neue Läufe zu starten.
- Der Teilnehmer kann die nicht gemeinsam genutzten Ressourcen nicht mehr verwenden, um neue Konfigurationen zu erstellen.
- HealthOmics durchsucht die Elastic Network Interfaces (ENIs), die es in den gemeinsam genutzten Subnetzen erstellt hat, nachdem sie 48 Stunden lang nicht verwendet wurden.

> **Important**
>
> HealthOmics erstellt ENIs im Teilnehmerkonto, obwohl sie an die gemeinsam genutzten Subnetze angehängt sind. Solange diese ENIs verbunden bleiben, kann der VPC-Besitzer sie nicht löschen und auch die gemeinsam genutzte VPC oder Subnetze nicht löschen. Danach HealthOmics durchsucht die ENIs, der VPC-Besitzer kann die zugehörigen Ressourcen löschen.

Verwenden von VPC-Peering

Verwenden Sie dieses Verfahren, um Ihre HealthOmics Ein Workflow führt eine private Konnektivität zu einer Ressource in der VPC eines anderen Kontos durch, indem eine VPC-Peering-Verbindung erstellt wird. Die ENIs Ihres Workflow-Laufs verbleiben in Ihrer eigenen (anfordernden) VPC, und die Peering-Verbindung leitet den Datenverkehr an die VPC des Empfängers weiter. Sie benötigen keine HealthOmics-specific richten Sie das für VPC-Peering ein, und Sie stellen das nicht ein `vpcOwnerAccountId` Parameter.

Dieses Verfahren verbindet zwei Konten mit einer Peering-Verbindung mithilfe von IPv4. Weitere Informationen zu anderen Peering-Szenarien finden Sie in der VPC-Peering-Leitfaden.

Voraussetzungen

Um VPC-Peering zu verwenden mit HealthOmics Für Workflow-Läufe benötigen Sie:

- Ein HealthOmics Workflow, der so konfiguriert ist, dass er sich mit Ihrer Ressource authentifiziert und dann aus ihr liest.
- Eine Ressource in einem anderen Konto, z. B. ein Amazon RDS-Cluster oder ein Lizenzserver, die über Amazon VPC verfügbar ist.
- Anmeldeinformationen für das Konto Ihres Anforderers und das Konto Ihres Akzeptors. Wenn Sie nicht berechtigt sind, das Konto Ihres Empfängers zu verwenden, wenden Sie sich an einen autorisierten Benutzer, um dieses Konto einzurichten.
- Erlaubnis, eine VPC (und unterstützende Amazon-VPC-Ressourcen) zu erstellen und zu aktualisieren, um sie mit Ihrer zu verknüpfen HealthOmics Der Arbeitsablauf wird ausgeführt.
- Erlaubnis zum Erstellen HealthOmics Ressourcen zur Konfiguration.
- Erlaubnis zum Erstellen einer VPC-Peering-Verbindung im Konto Ihres Anforderers.
- Erlaubnis, eine VPC-Peering-Verbindung im Konto Ihres Empfängers zu akzeptieren.
- Erlaubnis, die Konfiguration der VPC Ihres Empfängers (und der unterstützenden Amazon VPC-Ressourcen) zu aktualisieren.
- Erlaubnis zum Starten HealthOmics Der Arbeitsablauf wird ausgeführt.

Schritt 1: Erstellen Sie eine Amazon VPC im Konto Ihres Anforderers

Erstellen Sie eine Amazon-VPC, Subnetze, Routing-Tabellen und eine Sicherheitsgruppe in Ihrem HealthOmics Konto des Anfragenden. Anweisungen dazu finden Sie unter Erstellen Sie eine VPC im Amazon VPC-Benutzerhandbuch.

Um eine VPC für Ihre zu erstellen HealthOmics Der Arbeitsablauf wird ausgeführt

1. Öffnen Sie die Amazon-VPC-Konsole unter https://console.aws.amazon.com/vpc/.
2. Wählen Sie auf dem VPC-Dashboard VPC erstellen aus.
3. Für IPv4-CIDR-Block, stellen Sie ein privates Classless bereit Inter-Domain Routing-Block (CIDR), der sich nicht mit Blöcken überschneidet, die in der VPC Ihres Empfängers verwendet werden. Wählen Sie keinen Block, den die VPC Ihres Akzeptors verwendet, um Ressourcen IPs zuzuweisen, oder einen Block, der bereits in den Routing-Tabellen in der VPC Ihres Akzeptors definiert ist. Weitere Informationen finden Sie unter VPC-CIDR-Blöcke im Amazon VPC-Benutzerhandbuch.
4. Für Verfügbarkeitszonen, wählen Sie mindestens eine Availability Zone aus HealthOmics arbeitet in Ihrem AWS Region.
5. Einstellen Anzahl der öffentlichen Subnetze zu **0**.

6. Einstellen VPC-Endpunkte zu **None** (Sie können diese später zur Kostenoptimierung hinzufügen).
7. Wählen Sie VPC erstellen aus.

Schritt 2: Erstellen Sie eine VPC-Peering-Verbindungsanfrage

Erstellen Sie eine VPC-Peering-Verbindungsanforderung von der VPC Ihres Anforderers zur VPC des Akzeptierenden. Anweisungen dazu finden Sie unter Erstellen Sie eine VPC-Peering-Verbindung im Amazon VPC Peering Guide.

Um eine VPC-Peering-Verbindung anzufordern

1. Öffnen Sie die Amazon-VPC-Konsole.
2. Wählen Sie im Navigationsbereich Peering connections (Peering-Verbindungen) aus.
3. Wählen Sie Create peering connection (Peering-Verbindung erstellen).
4. Für VPC-ID (Anforderer), wählen Sie die VPC Ihres Anforderers aus.
5. Für Konto, wähle Ein anderes Konto und geben Sie die Konto-ID des Empfängers ein.
6. Für VPC-ID (Akzeptierer), geben Sie die VPC-ID des Akzeptierenden ein.
7. Wählen Sie Create peering connection (Peering-Verbindung erstellen).

Die Peering-Verbindung bleibt in der `Pending acceptance` geben Sie an, bis das Konto des Akzeptierenden dies akzeptiert.

Schritt 3: Bereite das Konto des Empfängers vor

Um Ihre Peering-Verbindung herzustellen und die VPC des Akzeptierenden auf die Verwendung der Verbindung vorzubereiten, melden Sie sich mit einer Rolle, die über die in den Voraussetzungen aufgeführten Berechtigungen verfügt, beim Konto des Akzeptierenden an. Führen Sie im Konto des Akzeptors die folgenden Verfahren aus.

> **Important**
>
> Akzeptieren Sie Peering-Anfragen nur von AWS Konten, die Sie kennen und denen Sie vertrauen. Eine Anfrage von einem unbekannten Konto könnte ein Versuch sein, Netzwerkzugriff auf Ihre VPC zu erhalten.

So akzeptieren Sie die VPC-Peering-Verbindungsanfrage

1. Öffnen Sie die Amazon-VPC-Konsole.
2. Wählen Sie im Navigationsbereich Peering connections (Peering-Verbindungen) aus.
3. Wählen Sie die ausstehende VPC-Peering-Verbindung aus (der Status lautet `pending-acceptance`).
4. Wählen Sie Aktionen, dann Anfrage annehmen.
5. Wenn Sie zur Bestätigung aufgefordert werden, wählen Sie Akzeptieren aus.
6. Wählen Ändern Sie jetzt meine Routentabellen um eine Route zur Haupt-Routing-Tabelle für Ihre VPC hinzuzufügen.

Untersuchen Sie anschließend die Routing-Tabellen für die VPC des Empfängers. Die Route, die Amazon VPC generiert, stellt möglicherweise keine Konnektivität her, je nachdem, wie die VPC des Empfängers eingerichtet ist. Suchen Sie nach Konflikten zwischen der neuen Route und der vorhandenen Konfiguration für die VPC. Anweisungen zur Problembehandlung finden Sie unter Problembehandlung bei einer VPC-Peering-Verbindung im Amazon VPC Peering Guide.

Um die Routing-Tabelle für die VPC des Empfängers zu aktualisieren

1. Öffnen Sie die Amazon-VPC-Konsole.
2. Wählen Sie im Navigationsbereich Route Tables (Routing-Tabellen) aus.
3. Wählen Sie die Routing-Tabelle für das Subnetz aus, das Ihrer Ressource zugeordnet ist.
4. Wählen Sie Aktionen, dann Routen bearbeiten.
5. Wählen Sie Route hinzufügen aus.
6. Für Zielort, geben Sie den CIDR-Block für die VPC Ihres Anforderers ein.
7. Wählen Sie als Ziel Ihre VPC-Peering-Verbindung aus. Die Verbindung muss den Status haben `Active` um in der Zielliste zu erscheinen.
8. Wählen Sie Änderungen speichern aus.

Weitere Informationen finden Sie unter Aktualisieren Sie Ihre Routing-Tabellen für eine VPC-Peering-Verbindung im Amazon VPC Peering Guide.

Um die Sicherheitsgruppe für die Ressource des Empfängers zu aktualisieren

1. Öffnen Sie die Amazon-VPC-Konsole.

2. Wählen Sie im Navigationsbereich Sicherheitsgruppen.
3. Wählen Sie die Sicherheitsgruppe für Ihre Ressource aus.
4. Wählen Aktionen, dann Regeln für eingehenden Verkehr bearbeiten.
5. Wählen Sie Regel hinzufügen aus.
6. Für Typ, wählen Sie das Protokoll aus, das Ihre Ressource verwendet (z. B. HTTPS, MySQL/Aurora, oder Benutzerdefiniertes TCP).
7. Für Port-Bereich, geben Sie den Port ein, auf dem Ihre Ressource lauscht.
8. Für Quelle, geben Sie den VPC-CIDR-Block Ihres Anforderers ein (z. B. 10.0.0.0/16).
9. Wählen Sie Regeln speichern aus.
10. Wählen Sie Edit outbound rules (Regeln für ausgehenden Datenverkehr bearbeiten). Die VPC-Standardeinstellungen gestatten allen ausgehenden Datenverkehr. Wenn der ausgehende Datenverkehr aus der Sicherheitsgruppe Ihres Empfängers eingeschränkt ist, fügen Sie eine ausgehende Regel hinzu, die den Datenverkehr zum VPC-CIDR-Block Ihres Anforderers zulässt.

Weitere Informationen finden Sie unter Konfigurieren Sie Sicherheitsgruppenregeln im Amazon VPC-Benutzerhandbuch.

(Optional) Um die DNS-Auflösung für die Peering-Verbindung zu aktivieren

1. Öffnen Sie die Amazon-VPC-Konsole.
2. Wählen Sie im Navigationsbereich Peering connections (Peering-Verbindungen) aus.
3. Wählen Sie Ihre Peering-Verbindung aus.
4. Wählen Sie Aktionen, dann DNS-Einstellungen bearbeiten.
5. Wählen Sie unter Akzepter-DNS-Auflösung die Option Erlauben Sie dem anfragenden VPC, DNS von akzeptierenden VPC-Hosts in private IP aufzulösen aus.
6. Wählen Sie Änderungen speichern aus.

Weitere Informationen finden Sie unter Aktivieren Sie die DNS-Auflösung für eine VPC-Peering-Verbindung im Amazon VPC Peering Guide.

Schritt 4: Aktualisieren Sie die VPC-Konfiguration im Konto des Anforderers

Loggen Sie sich in das Konto Ihres Anforderers ein und aktualisieren Sie dann die VPC-Konfiguration.

Um eine Route für die VPC-Peering-Verbindung hinzuzufügen

1. Öffnen Sie die Amazon-VPC-Konsole.
2. Wählen Sie im Navigationsbereich Route Tables (Routing-Tabellen) aus.
3. Wählen Sie die Routentabelle für das Subnetz aus, das Sie Ihrem zuordnen möchten HealthOmics Konfiguration.
4. Wählen Sie Aktionen, dann Routen bearbeiten.
5. Wählen Sie Route hinzufügen aus.
6. Für Zielort, geben Sie den CIDR-Block für die VPC des Empfängers ein.
7. Wählen Sie als Ziel Ihre VPC-Peering-Verbindung aus. Die Verbindung muss den Status haben `Active` um in der Zielliste zu erscheinen.
8. Wählen Sie Änderungen speichern aus.

Weitere Informationen finden Sie unter Aktualisieren Sie Ihre Routing-Tabellen für eine VPC-Peering-Verbindung im Amazon VPC Peering Guide.

Um die Sicherheitsgruppe für Ihr zu aktualisieren HealthOmics Der Arbeitsablauf wird ausgeführt

1. Öffnen Sie die Amazon-VPC-Konsole.
2. Wählen Sie im Navigationsbereich Sicherheitsgruppen.
3. Wählen Sie die Sicherheitsgruppe aus, die Sie für Ihre verwenden möchten HealthOmics Konfiguration.
4. Wählen Sie Aktionen, dann Regeln für ausgehenden Verkehr bearbeiten.
5. Wählen Sie Regel hinzufügen aus.
6. Für Typ, wählen Sie das Protokoll aus, das Ihre Ressource verwendet (z. B. HTTPS, MySQL/Aurora, oder Benutzerdefiniertes TCP).
7. Für Port-Bereich, geben Sie den Port ein, auf dem Ihre Ressource lauscht.
8. Für Zielort, geben Sie den VPC-CIDR-Block des Empfängers ein (z. B. 10.1.0.0/16).
9. Wählen Sie Regeln speichern aus.
10. Wenn Ihre Ressource Verbindungen zurück zu Ihren Workflow-Läufen herstellen muss, wählen Sie Bearbeiten Sie die Regeln für eingehenden Verkehr, fügen Sie eine Regel hinzu, die den Datenverkehr aus dem VPC-CIDR-Block des Empfängers zulässt, und wählen Sie Regeln speichern.

Weitere Informationen finden Sie unter Konfigurieren Sie Sicherheitsgruppenregeln im Amazon VPC-Benutzerhandbuch.

(Optional) Um die DNS-Auflösung für die Peering-Verbindung zu aktivieren

1. Öffnen Sie die Amazon-VPC-Konsole.
2. Wählen Sie im Navigationsbereich Peering connections (Peering-Verbindungen) aus.
3. Wählen Sie Ihre Peering-Verbindung aus.
4. Wählen Aktionen, dann DNS-Einstellungen bearbeiten.
5. Wählen Sie unter DNS-Auflösung des Antragstellers die Option Zulassen, dass der akzeptierende VPC die DNS von VPC-Hosts des Antragstellers in eine private IP auflöst aus.
6. Wählen Sie Änderungen speichern aus.

Weitere Informationen finden Sie unter Aktivieren Sie die DNS-Auflösung für eine VPC-Peering-Verbindung im Amazon VPC Peering Guide.

Schritt 5: Führen Sie Workflows mit VPC-Peering aus

Verwenden Sie beim Starten einer Workflow-Ausführung die Subnetze und Sicherheitsgruppen aus der VPC im Konto Ihres Anforderers. Nicht einstellen `vpcOwnerAccountId`. Die VPC-Peering-Verbindung leitet den Datenverkehr von Ihrem Workflow zur VPC im anderen Konto weiter.

Für Informationen zum Erstellen HealthOmics Konfigurationsressourcen und das Starten von Workflow-Läufen mit VPC-Netzwerken finden Sie unter HealthOmics Workflows mit einer VPC verbinden.

Verifizierung

Vergewissern Sie sich, dass das VPC-Peering korrekt konfiguriert ist:

- Der VPC-Peering-Verbindungsstatus lautet `Active` sowohl im Konto des Anforderers als auch im Konto des Empfängers.
- Beide VPCs haben bidirektionale Routen in ihren Routing-Tabellen, die auf die VPC-Peering-Verbindung verweisen.
- Wenn Sie DNS-Namen verwenden, ist die DNS-Auflösung in beide Richtungen auf der Peering-Verbindung aktiviert.
- Ein Workflow-Lauf begann mit den Subnetzen und Sicherheitsgruppen des Anforderers und ohne `vpcOwnerAccountId` gesetzt, erreicht erfolgreich die Ressource in der VPC des Empfängers.

Fehlerbehebung beim VPC-Peering

Allgemeine VPC-Peering-Probleme finden Sie unter Problembehandlung bei einer VPC-Peering-Verbindung im Amazon VPC Peering Guide.

Problem	Häufige Ursache	Auflösung
Ein Lauf kann die Ressource im anderen Konto nicht erreichen	Die Peering-Verbindung ist nicht aktiv, die CIDR-Blöcke überschneiden sich, in einer der VPCs fehlt eine Route oder eine Sicherheitsgruppe blockiert den Datenverkehr. Wenn der Lauf einen privaten Hostnamen verwendet, ist die DNS-Auflösung möglicher weise deaktiviert.	1. Vergewissern Sie sich, dass der Peering-Verbindung sstatus lautet `active`. 2. Vergewissern Sie sich, dass sich die beiden VPC-CIDR-Blöcke nicht überschneiden. 3. Vergewissern Sie sich, dass jede VPC eine Route zum CIDR-Block der anderen VPC hat, der auf die Peering-Verbindung abzielt. 4. Vergewissern Sie sich, dass die Sicherheitsgruppen in beiden VPCs den Datenverkehr auf dem Port der Ressource zulassen. 5. Wenn der Lauf einen privaten DNS-Hostnamen verwendet, stellen Sie sicher, dass die DNS-Auflö sung für die von Ihnen benötigte Richtung auf der Peering-Verbindung aktiviert ist. 6. Bestätigen Sie, dass Sie es nicht eingerichtet haben `vpcOwnerAccountId` in der Konfiguration.

Bewährte Methoden

Gehen Sie wie folgt vor, um Ihre kontoübergreifende VPC-Konfiguration sicher und wartbar zu halten:

- Verwenden Sie Sicherheitsgruppen mit den geringsten Rechten. Erlauben Sie nur den spezifischen Ports und Protokollen, die für Ihren Workflow erforderlich sind, den Zugriff auf die Ressource.
- Dokumentieren Sie die kontoübergreifende Beziehung. Notieren Sie, welche VPCs gemeinsam genutzt oder per Peering betrieben werden, mit wem und zu welchem Zweck, einschließlich des Zeitpunkts der ENI-Bereinigung (HealthOmics löscht ungenutzte ENIs (48 Stunden nach Beendigung einer RAM-Freigabe), sodass der VPC-Besitzer weiß, wann gemeinsam genutzte Ressourcen sicher gelöscht werden können.
- Überwachen Sie den kontoübergreifenden Verkehr. Verwenden Sie VPC Flow Logs und CloudWatch Metriken zur Überwachung von Verkehrsmustern und zur Erkennung von Anomalien.
- Gründlich testen. Überprüfen Sie die Konnektivität mit Test-Workflows, bevor Sie Produktions-Workloads ausführen, und stellen Sie sicher, dass der Konfigurationsstatus erreicht ist `ACTIVE` (RAM-Sharing) oder der Peering-Verbindungsstatus erreicht `Active` (VPC-Peering) vor dem Start der Produktionsläufe.
- Stimmen Sie sich mit den Kontoinhabern ab. Richten Sie klare Kommunikationskanäle mit dem Team ein, das das andere Konto verwaltet, zur Problembehebung und Wartung.
- Verwenden Sie Schlagworte. Kennzeichnen Sie Ihre VPC-Ressourcen, Peering-Verbindungen und AWS-RAM-Shares, um deren Zweck und Eigentümer zu identifizieren.
- Planen Sie die ENI-Bereinigung (RAM-Sharing). Richten Sie einen Prozess zwischen den Konten des VPC-Besitzers und der Teilnehmer ein, um ENIs zu bereinigen, wenn Konfigurationen nicht mehr benötigt werden. HealthOmics entfernt ungenutzte ENIs automatisch nach 48 Stunden, aber der VPC-Besitzer kann die gemeinsam genutzte VPC oder Subnetze erst löschen, wenn der Sweep abgeschlossen ist.
- Planen Sie CIDR-Blöcke sorgfältig (VPC-Peering). Stellen Sie sicher, dass sich CIDR-Blöcke nicht überschneiden und Raum für zukünftige Erweiterungen lassen. Wenn Sie davon ausgehen, dass Sie mehr als die standardmäßigen 50 aktiven Peering-Verbindungen pro VPC benötigen, sollten Sie stattdessen die gemeinsame Nutzung von RAM in Betracht ziehen.

Weitere Ressourcen

- HealthOmics Workflows mit einer VPC verbinden
- Teilen Sie Ihre VPC mit anderen Konten im Amazon VPC-Benutzerhandbuch

- Aktualisieren Sie Ihre Routing-Tabellen für eine VPC-Peering-Verbindung in der Amazon VPC Peering-Leitfaden
- Fehlerbehebung bei einer VPC-Peering-Verbindung im Amazon VPC Peering Guide
- VPC CIDR-Blöcke im Amazon VPC-Benutzerhandbuch
- AWS RAM-Benutzerhandbuch
- VPC-Peering-Handbuch

Konfigurations-APIs

HealthOmics bietet APIs zum Erstellen, Verwalten und Löschen von VPC-Konfigurationen. Sie können Konfigurationen für mehrere Workflow-Läufe wiederverwenden.

Themen

- CreateConfiguration
- GetConfiguration
- ListConfigurations
- DeleteConfiguration

CreateConfiguration

Erstellt eine neue Konfigurationsressource mit VPC-Netzwerkeinstellungen. Ein Schritt-für-Schritt-Beispiel finden Sie unter Schritt 4: Erstellen Sie eine Konfigurationsressource.

Anforderungssyntax:

```
aws omics create-configuration \
  --name configuration-name \
  --description description \
  --run-configurations '{"vpcConfig":{"securityGroupIds":["security-group-
id"],"subnetIds":["subnet-id"]}}' \
  --tags Key=key,Value=value \
  --region region
```

Parameter:

- Name (erforderlich) — Ein eindeutiger Name für die Konfiguration (maximal 50 Zeichen).
- Beschreibung (optional) — Eine Beschreibung der Konfiguration.

- run-configurations (optional) — VPC-Konfigurationseinstellungen:
  - `vpcConfig.securityGroupIds`— Eine Liste von 1—5 Sicherheitsgruppen-IDs.
  - `vpcConfig.subnetIds`— Eine Liste von 1—16 Subnetz-IDs.
- tags (optional) — Ressourcen-Tags.

Antwort:

```
{
  "arn": "arn:aws:omics:region:account-id:configuration/configuration-name",
  "uuid": "configuration-uuid",
  "name": "configuration-name",
  "runConfigurations": {
    "vpcConfig": {
      "securityGroupIds": ["security-group-id"],
      "subnetIds": ["subnet-id"],
      "vpcId": "vpc-id"
    }
  },
  "status": "CREATING",
  "creationTime": "timestamp",
  "tags": {}
}
```

Werte für den Konfigurationsstatus:

- ERSTELLEN — Die Konfiguration wird erstellt und Netzwerkressourcen werden bereitgestellt (bis zu 15 Minuten).
- AKTIV — Die Konfiguration ist einsatzbereit.
- LÖSCHEN — Die Konfiguration wird gelöscht.
- GELÖSCHT — Die Konfiguration wurde gelöscht.

GetConfiguration

Ruft Details einer bestimmten Konfiguration ab.

Anforderungssyntax:

```
aws omics get-configuration \
```

```
  --name configuration-name \
  --region region
```

Antwort:

```
{
  "arn": "arn:aws:omics:region:account-id:configuration/configuration-name",
  "uuid": "configuration-uuid",
  "name": "configuration-name",
  "runConfigurations": {
    "vpcConfig": {
      "securityGroupIds": ["security-group-id"],
      "subnetIds": ["subnet-id"],
      "vpcId": "vpc-id"
    }
  },
  "status": "ACTIVE",
  "creationTime": "timestamp",
  "tags": {}
}
```

ListConfigurations

Listet alle Konfigurationen in Ihrem Konto auf.

Anforderungssyntax:

```
aws omics list-configurations \
  --region region
```

Antwort:

```
{
  "items": [
    {
      "arn": "arn:aws:omics:region:account-id:configuration/configuration-name",
      "name": "configuration-name",
      "description": "description",
      "status": "ACTIVE",
      "creationTime": "timestamp"
    }
  ]
```

```
}
```

DeleteConfiguration

Löscht eine Konfiguration. Sie können keine Konfiguration löschen, die derzeit von aktiven Workflow-Ausführungen verwendet wird.

Anforderungssyntax:

```
aws omics delete-configuration \
  --name configuration-name \
  --region region
```

> **Note**
>
> Der Konfigurationsstatus ändert sich in LÖSCHEN, während die Netzwerkressourcen bereinigt werden, und dann in GELÖSCHT, sobald der Vorgang abgeschlossen ist.

Ausführen von Workflows mit VPC-Netzwerken

Einen Lauf mit VPC-Netzwerk starten

Um das VPC-Netzwerk in einer Workflow-Ausführung zu verwenden, geben Sie den `networking-mode` Parameter an und: `configuration-name`

```
aws omics start-run \
  --workflow-id 1234567 \
  --role-arn arn:aws:iam::123456789012:role/OmicsWorkflowRole \
  --output-uri s3://my-bucket/outputs/ \
  --networking-mode VPC \
  --configuration-name my-vpc-config \
  --region us-west-2
```

Parameter:

- Netzwerkmodus — Auf setzen, um das VPC-Netzwerk `VPC` zu aktivieren. Der Standardwert ist `RESTRICTED`.
- Konfigurationsname (erforderlich) — Der Name der zu verwendenden Konfiguration.

Die ausgeführte Netzwerkkonfiguration anzeigen

Verwenden Sie`GetRun`, um die Netzwerkkonfiguration für einen Lauf anzuzeigen:

```
aws omics get-run \
  --id run-id \
  --region region
```

Die Antwort umfasst den Netzwerkmodus, Konfigurationsdetails und die VPC-Konfiguration. Das folgende Beispiel zeigt die VPC-related Felder aus der Antwort:

```
{
  "arn": "arn:aws:omics:region:account-id:run/run-id",
  "id": "run-id",
  "status": "status",
  "workflowId": "workflow-id",
  "networkingMode": "VPC",
  "configuration": {
    "name": "configuration-name",
    "arn": "arn:aws:omics:region:account-id:configuration/configuration-name",
    "uuid": "configuration-uuid"
  },
  "vpcConfig": {
    "subnets": ["subnet-id-1", "subnet-id-2"],
    "securityGroupIds": ["security-group-id"],
    "vpcId": "vpc-id"
  }
}
```

Unveränderlichkeit der Konfiguration

Die Konfigurationsressourcen sind unveränderlich — nachdem Sie eine Konfiguration erstellt haben, können Sie ihre Einstellungen nicht mehr ändern. Um andere Netzwerkeinstellungen zu verwenden, erstellen Sie eine neue Konfiguration. Sie können eine Konfiguration nicht löschen, solange sie von aktiven Workflowläufen verwendet wird. Die `DeleteConfiguration` Anforderung wird zurückgewiesen, bis diese Ausführungen den Endzustand erreichen.

Überlegungen zum Caching von Aufrufen

Wenn Sie VPC-Netzwerke mit Call-Caching verwenden, stellen Sie sicher, dass Ihre Workflow-Engine entsprechend konfiguriert ist. Eine ausführliche Anleitung zum Caching von Anrufen pro Engine finden Sie unter. Engine-specific Caching-Funktionen

> **Important**
>
> Wenn Sie eine Verbindung zu nicht deterministischen oder dynamischen Ressourcen herstellen (z. B. Datenbanken von Drittanbietern im öffentlichen Internet), sollten Sie in Ihren Workflows die Abmeldefunktion für Cache-Aufgaben verwenden, um zu vermeiden, dass dynamische Datensätze zwischengespeichert werden, die sich auf die Laufausgaben auswirken könnten.

Problembehandlung bei VPC-Netzwerken

Themen

- Netzwerküberwachung und Fehlerbehebung
- Fehlerbehebung bei der Konfiguration

Netzwerküberwachung und Fehlerbehebung

CloudTrail Protokollierung

Alle Konfigurations-API-Operationen und Workflow-Läufe, die VPC-Netzwerke verwenden, sind angemeldet CloudTrail. Wird verwendet CloudTrail , um Konfigurationsänderungen zu überprüfen und nachzuverfolgen, welche Läufe VPC-Netzwerke verwenden.

Fehlerbehebung mit ENI-Flow-Logs

Wenn Ihr Workflow auf externe Ressourcen über das Internet zugreift, können Sie VPC Flow Logs verwenden, um Konnektivität zu überprüfen und Probleme zu diagnostizieren. HealthOmics stellt elastische Netzwerkschnittstellen (ENIs) in Ihren VPC-Subnetzen bereit, um den Datenverkehr von Ihren Workflow-Aufgaben weiterzuleiten. Indem Sie die zugehörigen Flussprotokolle untersuchen ENIs, können Sie den Netzwerkverkehr zu und von externen Zielen verfolgen.

⚠ Kostenmanagement für VPC Flow Logs

VPC-Flow-Logs können erhebliche Kosten verursachen, insbesondere auf VPC-Ebene. Um die Kosten zu minimieren:

- Löschen Sie die Flow-Logs nach der Fehlerbehebung. Sobald Sie die Verbindungsprobleme behoben haben, löschen Sie das Flow-Protokoll, damit keine Gebühren mehr anfallen.
- Verwenden Sie Amazon S3 anstelle von CloudWatch Logs für die Langzeitspeicherung. Amazon S3 S3-Speicher ist deutlich günstiger als CloudWatch Logs. Konfigurieren Sie Flow-Protokolle für die Veröffentlichung in Amazon S3, wenn Sie Protokolle für Compliance- oder Sicherheitsanalysen aufbewahren müssen.
- Legen Sie Richtlinien zur Aufbewahrung von CloudWatch Protokollen fest. Wenn Sie CloudWatch Protokolle verwenden, konfigurieren Sie den automatischen Protokollablauf (z. B. 7 Tage), um unbestimmte Speicherkosten zu vermeiden.
- Verwenden Sie zur Fehlerbehebung Flow-Logs auf ENI-Ebene. Erstellen Sie für einmaliges Debugging Flow-Logs auf der spezifischen Kunden-ENI und nicht auf der gesamten VPC.

Einrichtung von Flow-Logs zur Fehlerbehebung

Option 1: Flow-Logs auf VPC-Ebene (zur laufenden Überwachung)

Aktivieren Sie Flow-Logs auf Ihrer VPC, um den Datenverkehr aus allen HealthOmics Workflow-Läufen automatisch zu erfassen. Dies ist am besten, wenn Sie viele Workflow-Läufe haben und umfassende Transparenz wünschen, ohne einzelne ENIs Workflow-Läufe nachverfolgen zu müssen.

1. Aktivieren Sie VPC Flow Logs. In der Amazon VPC-Konsole:
   a. Wählen Sie Ihre VPCs und wählen Sie die in Ihrer HealthOmics Konfiguration verwendete VPC
   b. Wählen Sie den Tab Flow Logs
   c. Wählen Sie Flow-Protokoll erstellen
   d. Konfigurieren Sie das Flow-Protokoll so, dass der gesamte Datenverkehr (sowohl akzeptiert als auch abgelehnt) erfasst wird
   e. Wählen Sie CloudWatch Logs als Ziel aus, um die Abfrage zu vereinfachen

2. Starten Sie eine Workflow-Ausführung. Starten Sie eine Workflow-Ausführung mit aktiviertem VPC-Netzwerk. Notieren Sie sich die Lauf-ID und die Startzeit für das spätere Filtern von Flow-Protokollen.

Fragen Sie CloudWatch Flow-Logs mithilfe von Logs Insights nach Zeitfenster, Ziel-IP oder Verkehrsmustern ab. Sie müssen kein bestimmtes ENI identifizieren IDs.

Option 2: Flow-Logs auf ENI-Ebene (für gezielte Problembehebung)

Aktivieren Sie Flow-Logs nur ENIs dann, wenn Sie nur wenige HealthOmics ENIs in Ihrem Konto haben. Dies ist der kostengünstigste Ansatz und macht es einfach, den Datenverkehr für bestimmte Workflow-Läufe zu isolieren.

1. Finden Sie den Kunden ENI. In der Amazon EC2 EC2-Konsole:
   a. Wählen Sie Netzwerkschnittstellen
   b. Filtern Sie nach Tag`Service: HealthOmics`, um nur das Objekt anzuzeigen, das von ENIs erstellt wurde HealthOmics
   c. Optional können Sie weiter nach der Subnetz-ID aus Ihrer HealthOmics Konfiguration filtern
   d. Notieren Sie sich die ENI-ID und die private IP-Adresse
2. Aktivieren Sie die Flow-Logs auf der ENI.
   a. Wählen Sie das ENI und dann die Registerkarte Flow-Logs
   b. Wählen Sie Flow-Protokoll erstellen
   c. Konfigurieren Sie das Flow-Protokoll so, dass der gesamte Datenverkehr erfasst wird
   d. Wählen Sie CloudWatch Logs als Ziel

> **Note**
>
> Flow-Logs erfassen nur den Datenverkehr ab dem Zeitpunkt, zu dem sie aktiviert wurden. Aktivieren Sie Flow-Logs auf VPC-Ebene, bevor Sie Workflows ausführen. Bei Flow-Logs auf ENI-Ebene erfasst dasselbe Flow-Protokoll nach der Aktivierung auf einer ENI den Datenverkehr für alle future Workflow-Ausführungen, die diese ENI verwenden.

Grundlegendes zum VPC Flow Log-Format

VPC Flow Logs verwenden ein durch Leerzeichen getrenntes Format mit den folgenden Feldern:

```
version account_id interface_id srcaddr dstaddr srcport dstport protocol packets bytes
 start end action log_status
```

Feldbeschreibungen:

- Version — Version im Flow-Log-Format (normalerweise 2)
- account_id — Deine AWS Konto-ID
- interface_id — Die ENI-ID (zum Beispiel eni-0e57c5476efeac402)
- srcaddr — Quell-IP-Adresse
- dstaddr — Ziel-IP-Adresse
- srcport — Quell-Port-Nummer
- dstport — Zielportnummer
- protocol — IANA-Protokollnummer (6=TCP, 17=UDP, 1=ICMP)
- packages — Anzahl der Pakete im Flow
- Bytes — Anzahl der Byte im Flow
- start — Startzeit des Flows (Unix-Zeitstempel)
- end — Endzeit des Flusses (Unix-Zeitstempel)
- Aktion — AKZEPTIEREN oder ABLEHNEN
- log_status — OK, NODATA oder SKIPDATA

Beispiele für Flow-Logeinträge:

```
2 074296239033 eni-0e57c5476efeac402 10.0.130.58 13.226.238.96 40565 443 6 13 1502
 1774338927 1774338929 ACCEPT OK
2 074296239033 eni-0e57c5476efeac402 13.226.238.96 10.0.130.58 443 40565 6 8 1024
 1774338928 1774338930 ACCEPT OK
```

Diese Einträge zeigen eine erfolgreiche bidirektionale HTTPS-Kommunikation. Schlüssel IPs: 10.0.130.58 ist der Kunde, von dem ENI HealthOmics in Ihrem Konto erstellt wurde, und 13.226.238.96 ist die externe öffentliche Domain, auf die Ihr Workflow zugreift. Der erste Eintrag ist ausgehender Verkehr und der zweite ist der Rückverkehr. Beide zeigen ACCEPT an, was darauf hinweist, dass der Verkehr von Sicherheitsgruppen zugelassen wurde.

Abfragen von Flow-Logs in CloudWatch Logs Insights

Wenn Flow-Logs in CloudWatch Logs veröffentlicht werden, verwenden Sie CloudWatch Logs Insights, um die Daten abzufragen und zu analysieren.

Finden Sie abgelehnten Traffic (beginnen Sie hier)

```
fields @timestamp, interfaceId, srcAddr, dstAddr, srcPort, dstPort, protocol, action
| filter action = "REJECT"
| sort @timestamp desc
```

Wenn dies zu Ergebnissen führt, liegt möglicherweise ein Verbindungsproblem vor. Die abgelehnten Einträge zeigen, welcher Datenverkehr von Sicherheitsgruppen oder Netzwerken blockiert wird ACLs.

Finden Sie den Verkehr zu einer bestimmten externen IP

Lösen Sie die Domain zunächst mit `nslookup` oder in eine IP-Adresse auf`dig`:

```
$ nslookup ftp.ncbi.nlm.nih.gov
Server:  127.53.53.53
Address: 127.53.53.53#53

Non-authoritative answer:
ftp.ncbi.nlm.nih.gov    canonical name = ftp.wip.ncbi.nlm.nih.gov.
Name:    ftp.wip.ncbi.nlm.nih.gov
Address: 130.14.250.10
Name:    ftp.wip.ncbi.nlm.nih.gov
Address: 130.14.250.11
```

Der „Server“ und die „Adresse“ oben sind Ihr DNS-Resolver. Die Adressen unter „Nicht autoritative Antwort“ (130.14.250.10 und 130.14.250.11) sind die tatsächlichen Adressen für die Domain. IPs

Fragen Sie Flow-Logs mit einem Präfix ab, das einer beliebigen IP in diesem Bereich entspricht:

```
fields @timestamp, interfaceId, srcAddr, dstAddr, srcPort, dstPort, protocol, action
| filter dstAddr like "130.14.250"
| sort @timestamp desc
```

Dies entspricht jeder IP, die mit 130.14.250 beginnt, und erfasst den gesamten Datenverkehr IPs in diesem Subnetz.

Finden Sie HTTPS-Verkehr zu externen Zielen

```
fields @timestamp, interfaceId, srcAddr, dstAddr, srcPort, dstPort, protocol, action
| filter dstPort = 443 and protocol = 6
| filter not (dstAddr like /^10\./ or dstAddr like /^172\./ or dstAddr like /
^192\.168\./)
| sort @timestamp desc
```

Der zweite Filter schließt private IP-Bereiche aus und zeigt nur Datenverkehr zu externen (öffentlichen) Zielen an.

> **Note**
>
> Protokollnummern: 6=TCP, 17=UDP, 1=ICMP. Bei Diensten mit Lastenausgleich (z. B. CloudFront) gibt DNS möglicherweise andere Werte zurück. Filtern Sie daher nach dem Zielport IPs statt nach der IP-Adresse.

Allgemeine Ablaufprotokollmuster und Probleme

Ausgehender Verkehr wurde zurückgewiesen

```
Outbound: 2 074296239033 eni-0e57c5476efeac402 10.0.130.58 13.226.238.96 40565 443 6
 1 60 1774338927 1774338929 REJECT OK
```

Ursache: Die Sicherheitsgruppe erlaubt keinen ausgehenden Datenverkehr zum Zielport oder IP-Bereich.

Lösung: Fügen Sie Ihrer Sicherheitsgruppe eine Regel für ausgehenden Datenverkehr hinzu:

- Für HTTPS: Erlaube den TCP-Port 443 bis 0.0.0.0/0
- Für HTTP: Erlaube den TCP-Port 80 bis 0.0.0.0/0
- Für einen breiteren Zugriff: Erlaube allen bis 0.0.0.0/0 TCP/UDP

Rückverkehr abgelehnt

```
Outbound: 2 074296239033 eni-0e57c5476efeac402 10.0.130.58 8.8.8.8 54321 53 17 1 64
 1774338927 1774338929 ACCEPT OK
Return:   2 074296239033 eni-0e57c5476efeac402 8.8.8.8 10.0.130.58 53 54321 17 1 64
 1774338928 1774338930 REJECT OK
```

Ursache: Die Netzwerk-ACL blockiert den Rückverkehr. Im Gegensatz zu Sicherheitsgruppen (statusbehaftet) ACLs sind Netzwerke zustandslos und erfordern explizite Regeln für beide Richtungen.

Lösung: Überprüfen Sie in der VPC-Konsole die Netzwerk-ACL Ihres Subnetzes und stellen Sie sicher, dass die Regeln für eingehenden Datenverkehr auf ephemeren Ports (1024-65535) von externen Quellen zulassen. Fügen Sie bei Bedarf eine Regel hinzu: Erlauben Sie die Ports 1024-65535 von 0.0.0.0/0 TCP/UDP

Fehlender Rückverkehr

```
Outbound: 2 074296239033 eni-0e57c5476efeac402 10.0.130.58 8.8.8.8 54321 53 17 1 64
 1774338927 1774338929 ACCEPT OK
```

Ursache: Gateway/Internet Das NAT-Gateway ist nicht richtig konfiguriert, oder ENI hat keine Verbindung zum Internet.

Lösung:

- Stellen Sie sicher, dass die Routentabelle eine Route zum NAT-Gateway enthält (0.0.0.0/0 → nat-xxxxx)
- Stellen Sie sicher, dass sich das NAT-Gateway mit einer Elastic IP im Status AVAILABLE befindet
- Prüfen Sie, ob sich das NAT-Gateway in einem öffentlichen Subnetz mit einer Route zum Internet Gateway befindet

Keine Flow-Log-Einträge für den erwarteten Verkehr

Ursache: Der Datenverkehr erreicht die ENI nicht, oder die Flow-Protokolle sind nicht richtig konfiguriert.

Lösung:

- Stellen Sie sicher, dass die Flow-Logs aktiviert und so konfiguriert sind, dass sie den gesamten Datenverkehr erfassen
- Überprüfen Sie die CloudWatch Workflow-Protokolle in Logs, um zu bestätigen, dass der Workflow versucht, auf die externe Ressource zuzugreifen
- Stellen Sie sicher, dass die Routentabelle eine Route zum NAT-Gateway enthält (0.0.0.0/0 → nat-xxxxx)
- Stellen Sie sicher, dass sich das NAT-Gateway mit einer Elastic IP im Status AVAILABLE befindet

Bewährte Methoden für die Fehlerbehebung bei Flow-Protokollen

1. Aktivieren Sie die Flow-Protokolle, bevor Sie mit der Fehlerbehebung beginnen. Flow-Logs erfassen nur den Datenverkehr ab dem Zeitpunkt, zu dem sie aktiviert wurden. Aktivieren Sie sie in allen Subnetzen in Ihrer HealthOmics Konfiguration, bevor Sie Workflows ausführen.
2. Verwenden Sie CloudWatch Logs Insights zur Analyse. CloudWatch Logs Insights bietet leistungsstarke Abfragefunktionen für Flow-Logs. Speichern Sie häufig verwendete Abfragen für den schnellen Zugriff.
3. Nach Zeitfenster filtern. Schränken Sie Ihre Flow-Protokollabfragen auf das Zeitfenster ein, in dem Ihr Workflow-Lauf aktiv war, um Störungen zu reduzieren und die Abfrageleistung zu verbessern.
4. Achten Sie auf beide Verkehrsrichtungen. Vergewissern Sie sich immer, dass sowohl für den ausgehenden als auch für den Rückverkehr die Option ACCEPT angezeigt wird. Eine Verbindung erfordert eine bidirektionale Kommunikation.
5. Dokumentieren Sie Ihre Ergebnisse. Dokumentieren Sie bei der Behebung von Verbindungsproblemen die ENI-ID, IP-Adressen, Ports und Flow-Protokolleinträge des Kunden. Diese Informationen sind wertvoll für Supportfälle und future Problemlösungen.
6. Testen Sie zunächst mit einem einfachen Arbeitsablauf. Bevor Sie komplexe Workflows ausführen, testen Sie die Konnektivität mit einem einfachen Workflow, der versucht, auf die externe Ressource zuzugreifen, und das Ergebnis protokolliert. Dies hilft, Netzwerkprobleme von Problemen mit der Workflow-Logik zu trennen.

Fehlerbehebung bei der Konfiguration

Die Konfiguration blieb im Status CREATING hängen

Ursache: Die Bereitstellung von Netzwerkressourcen kann mehrere Minuten dauern.

Lösung: Warten Sie bis zu 10 Minuten. Wenn der Status nicht zu AKTIV wechselt, überprüfen Sie Folgendes:

- Ihre Subnetze und Sicherheitsgruppen sind vorhanden und befinden sich in derselben VPC.
- Sie verfügen über die erforderlichen IAM-Berechtigungen.
- Die serviceverknüpfte Rolle wurde erfolgreich erstellt.

Die Ausführung kann mit dem VPC-Netzwerk nicht gestartet werden

Ursache: Die Konfiguration ist möglicherweise nicht AKTIV, oder es liegen Probleme mit der Netzwerkverbindung vor.

Lösung:

- Stellen Sie sicher, dass der Konfigurationsstatus AKTIV ist, indem Sie`GetConfiguration`.
- Vergewissern Sie sich, dass die Sicherheitsgruppenregeln den erforderlichen ausgehenden Datenverkehr zulassen.
- Stellen Sie sicher, dass sich die Subnetze in den Availability Zones befinden, in denen sie betrieben HealthOmics werden

Konfiguration kann nicht gelöscht werden

Ursache: Die Konfiguration wird von aktiven Workflow-Ausführungen verwendet.

Lösung: Warten Sie, bis alle Läufe, die die Konfiguration verwenden, abgeschlossen sind, und versuchen Sie dann erneut, den Löschvorgang durchzuführen.

Die mit dem Dienst verknüpfte Rolle kann nicht gelöscht werden

Ursache: In Ihrem Konto sind aktive VPC-Konfigurationen vorhanden.

Lösung: Löschen Sie zuerst alle VPC-Konfigurationen und dann die serviceverknüpfte Rolle.

Der Workflow kann keine Verbindung zur externen Ressource herstellen

Ursache: Fehlkonfiguration der Sicherheitsgruppe oder der Routingtabelle.

Lösung:

1. Aktivieren Sie VPC Flow Logs, um abgelehnte Pakete zu identifizieren
2. Überprüfen Sie, ob die ausgehenden Sicherheitsgruppenregeln den Datenverkehr zum Ziel zulassen
3. Stellen Sie sicher, dass die Routentabelle eine Route zum NAT-Gateway hat (0.0.0.0/0 → nat-xxxxxx)
4. Stellen Sie für den regionsübergreifenden Zugriff auf AWS Dienste sicher, dass die Zielregion erreichbar ist

5. Testen Sie die Konnektivität von einer Amazon EC2 EC2-Instance im selben Subnetz

Probleme mit der Netzwerkleistung

Symptom: Langsame Datenübertragung oder Workflow-Timeouts.

Ursache: Einschränkungen des Netzwerkdurchsatzes oder Überlastung des NAT-Gateways.

Lösung:

- Der Netzwerkdurchsatz beginnt bei 10 Gbit/s pro ENI und steigt bei anhaltendem Datenverkehr innerhalb von 60 Minuten auf bis zu 100 Gbit/s
- Für Workflows mit unmittelbaren Anforderungen an einen hohen Durchsatz wenden Sie sich bitte an AWS den Support
- Überwachen Sie die NAT-Gateway-Metriken CloudWatch , um die Sättigung
- Erwägen Sie die Bereitstellung zusätzlicher NAT-Gateways in mehreren Availability Zones, um einen höheren Durchsatz zu erzielen

Der Workflow kann das Internet nicht erreichen

Ursache: Die privaten Subnetze haben möglicherweise keine Route zu einem NAT-Gateway, oder Sicherheitsgruppenregeln blockieren möglicherweise ausgehenden Datenverkehr.

Lösung:

- Stellen Sie sicher, dass die Routing-Tabelle für Ihre privaten Subnetze eine Route zu einem NAT-Gateway enthält (0.0.0.0/0 → nat-xxxxxxxxx).
- Vergewissern Sie sich, dass die Sicherheitsgruppenregeln ausgehenden Datenverkehr an den erforderlichen Ports zulassen.
- Stellen Sie sicher, dass sich das NAT-Gateway in einem öffentlichen Subnetz mit einer Route zu einem Internet-Gateway befindet.

Die Workflow-Ausführung schlägt mit Verbindungsfehlern fehl

Ursache: Der Netzwerkverkehr ist möglicherweise blockiert oder falsch konfiguriert.

Lösung:

1. Stellen Sie sicher, dass sich die Konfiguration noch im Status AKTIV befindet, indem Sie`GetConfiguration`.
2. Erstellen Sie ENIs in Ihrer VPC ein VPC-Flow-Protokoll, um den Datenverkehr zu überprüfen. Weitere Informationen finden Sie unter VPC-Flow-Protokolle im Benutzerhandbuch für Amazon VPC.
3. Überprüfen Sie das Flow-Protokoll auf REJECT-Einträge. Wenn Sie abgelehnte Pakete sehen, aktualisieren Sie Ihre Sicherheitsgruppenregeln, um den erforderlichen ausgehenden Datenverkehr zuzulassen.
4. Wenn das Flow-Protokoll keine Grundursache aufdeckt, wenden Sie sich an den AWS Support.

Bewährte Methoden

Sicherheit

1. Verwenden Sie Sicherheitsgruppen mit den geringsten Rechten. Erlauben Sie nur den minimal erforderlichen ausgehenden Verkehr. Verwenden Sie bestimmte CIDR-Zielblöcke anstelle von 0.0.0. 0/0 wenn möglich. Dokumentieren Sie den Zweck jeder Sicherheitsgruppenregel.
2. Trennen Sie die Konfigurationen nach Umgebung. Erstellen Sie separate Konfigurationen für Entwicklung, Staging und Produktion. Verwenden Sie für jede Umgebung unterschiedliche VPCs oder Subnetze. Wenden Sie die entsprechenden Tags auf die Konfigurationen für die Organisation an.
3. Implementieren Sie die Netzwerküberwachung. Aktivieren Sie VPC Flow Logs für die Sicherheitsanalyse. Richten Sie CloudWatch Alarme für ungewöhnliche Verkehrsmuster ein. Überprüfen Sie die CloudTrail Protokolle regelmäßig auf Konfigurationsänderungen.
4. Verwenden Sie VPC-Endpunkte für AWS Dienste. Konfigurieren Sie VPC-Endpunkte für Amazon S3, Amazon ECR und andere Dienste. AWS Dies reduziert die NAT-Gateway-Kosten, verbessert die Leistung und bietet zusätzliche Sicherheit, indem der Datenverkehr innerhalb des Netzwerks gehalten wird. AWS

Leistung

1. Planen Sie die Netzwerkskalierung ein. Der Netzwerkdurchsatz beginnt bei 10 Gbit/s und wird im Laufe der Zeit auf 100 Gbit/s skaliert. Wenn Sie sofort einen hohen Durchsatz benötigen, planen Sie im Voraus und fordern Sie eine Vorwärmung an. Überwachen Sie Netzwerkkennzahlen, um Ihre Workflow-Anforderungen zu verstehen.

2. Stellen Sie NAT-Gateways pro Availability Zone bereit. Verwenden Sie ein NAT-Gateway pro AZ für Produktionsworkloads. Dies verbessert die Stabilität und den Durchsatz und reduziert die Kosten für die AZ-übergreifende Datenübertragung.
3. Konfigurationen wiederverwenden. Erstellen Sie Konfigurationen, die in mehreren Workflows gemeinsam genutzt werden können. Dies reduziert den Aufwand für das Konfigurationsmanagement und gewährleistet konsistente Netzwerkeinstellungen.
4. Testen Sie die Konfigurationen vor dem Produktionseinsatz. Validieren Sie die Netzwerkkonnektivität mit Test-Workflows. Stellen Sie sicher, dass die Regeln der Sicherheitsgruppe den erforderlichen Datenverkehr zulassen. Testen Sie Failover-Szenarien mit Multi-AZ-Konfigurationen.

Kostenoptimierung

1. Verwenden Sie VPC-Endpunkte anstelle von NAT Gateway. Verwenden Sie für den AWS Servicezugriff VPC-Endpunkte (keine Datenverarbeitungsgebühren). Für Amazon S3 Gateway-Endpunkte fallen keine zusätzlichen Kosten an. Für Schnittstellenendpunkte fallen stündliche Gebühren an, können aber kostengünstiger sein als NAT Gateway.
2. Überwachen Sie die Kosten für die Datenübertragung. Die Datenübertragung ist kostenlos. Für die ausgehende Datenübertragung ins Internet fallen AWS Standarddatenübertragungsraten an. Cross-Region Die Datenübertragung hat höhere Raten. Verwenden Sie den AWS Cost Explorer, um die VPC-related Kosten zu verfolgen.
3. Right-size Bereitstellung von NAT Gateway. Verwenden Sie für die Entwicklung ein NAT-Gateway für alle AZs. Verwenden Sie für die Produktion ein NAT-Gateway pro AZ, um die Stabilität zu gewährleisten. Überwachen Sie die NAT-Gateway-Nutzung, um eine Überversorgung zu vermeiden.
4. Löschen Sie nicht verwendete Konfigurationen. Überprüfen und löschen Sie regelmäßig Konfigurationen, die nicht mehr verwendet werden. Verwenden Sie Stichwörter, um die Eigentümerschaft und den Zweck der Konfiguration zu identifizieren.

Betriebsbereit

1. Verwenden Sie beschreibende Konfigurationsnamen. Fügen Sie Umgebung, Zweck und Team in den Namen ein (z. B.`prod-genomics-vpc`,`dev-clinical-trials-vpc`).
2. Kennzeichnen Sie alle Konfigurationen. Verwenden Sie eine konsistente Tagging-Strategie für alle Ressourcen. Fügen Sie Stichwörter für Umgebung CostCenter, Eigentümer und Zweck hinzu.

3. Dokumentieren Sie die Netzwerkanforderungen. Dokumentieren Sie, auf welche externen Dienste jede Konfiguration zugreift. Pflegen Sie eine Übersicht der Sicherheitsgruppenregeln und ihrer Zwecke. Teilen Sie Diagramme der Netzwerkarchitektur mit Ihrem Team.

Kontingente für VPC-Netzwerke

In der folgenden Tabelle sind die Kontingente für VPC-Netzwerkkonfigurationen aufgeführt:

Ressource	Standardlimit	Einstellbar
Maximale Konfigurationen pro Konto	10	Ja
Maximale Sicherheitsgruppen pro Konfiguration	5	Nein
Maximale Anzahl Subnetze pro Konfiguration	16	Nein
Maximale Subnetze pro Availability Zone	1	Nein
CreateConfiguration API-TPS	1	Ja
Elastische Netzwerkschnittstellen pro Region (Kunden-VPC)	5,000	Ja

Um eine Kontingenterhöhung anzufordern, öffnen Sie die Service-Kontingente-Konsole, wählen Sie AWS Dienste aus, suchen Sie nach dem Kontingent, das Sie erhöhen möchten AWS HealthOmics, wählen Sie das Kontingent aus, das Sie erhöhen möchten, und wählen Sie Kontingenterhöhung anfordern aus. Anfragen zur Erhöhung des Kontingents werden in der Regel innerhalb von 1—2 Werktagen bearbeitet.

Internetzugang für VPC-connected Workflows

Wenn Sie einen AWS HealthOmics Lauf mit einer VPC verbinden, kann der Lauf nur auf Ressourcen zugreifen, die in dieser VPC verfügbar sind. Um deinem Lauf Zugriff auf das öffentliche Internet

oder AWS Dienste außerhalb der VPC zu gewähren, musst du deine VPC mit den entsprechenden Netzwerkressourcen konfigurieren.

In diesem Thema wird beschrieben, wie Sie Ihre VPC einrichten, um Internetzugang und effiziente Konnektivität zu AWS Diensten für Ihre Läufe bereitzustellen. VPC-connected Informationen zum Verbinden von Läufen mit einer VPC finden Sie unter. HealthOmics Workflows mit einer VPC verbinden

> **Important**
>
> Wenn ein Lauf mit einem öffentlichen Subnetz verbunden wird, erhält er weder Internetzugang noch eine öffentliche IP-Adresse. Verwenden Sie immer private Subnetze mit NAT-Gateway-Routen für Läufe, die eine Internetverbindung erfordern.

Themen

- Einrichtung einer VPC mit Internetzugang
- VPC-Endpunkte für AWS service
- Sicherheitsgruppenkonfiguration
- Konfiguration der Routing-Tabelle
- IAM-Berechtigungen für AWS service
- Testen der VPC-Konnektivität
- Beispiele
- Bewährte Methoden

Einrichtung einer VPC mit Internetzugang

Um Ihren VPC-connected Läufen Zugriff auf das Internet zu gewähren, erstellen Sie eine VPC mit privaten Subnetzen, die den ausgehenden Verkehr über ein NAT-Gateway weiterleiten.

Diese Konfiguration bietet:

- Private Subnetze für HealthOmics Workflow-Aufgaben
- Öffentliche Subnetze mit NAT-Gateways für den ausgehenden Internetzugang

Unterstützte Regionen und Verfügbarkeitszonen

HealthOmics Workflows funktionieren in den folgenden Regionen und Verfügbarkeitszonen. Stellen Sie beim Erstellen Ihrer VPC sicher, dass sich Ihre Subnetze in einer oder mehreren dieser Availability Zones befinden.

Region	Name der Verfügbarkeitszone	Availability Zone-ID
us-west-2	us-west-2a	usw2-az2
	us-west-2b	usw2-az1
	us-west-2c	usw2-az3
	us-west-2d	usw2-az4
us-east-1	us-ost-1a	use1-az4
	us-ost-1b	use1-az6
	us-ost-1c	use1-az1
	US-Ost-1D	use1-az2
	us-east-1f	use1-az5
eu-west-1	eu-west-1a	euw1-az2
	eu-west-1b	euw1-az3
	eu-west-1c	euw1-az1
eu-central-1	eu-central-1a	euc1-az2
	eu-central-1b	euc1-az3
	eu-central-1c	euc1-az1
eu-west-2	eu-west-2a	euw2-az2
	eu-west-2b	euw2-az3

Region	Name der Verfügbarkeitszone	Availability Zone-ID
	eu-west-2c	euw2-az1
ap-southeast-1	AP-Südost-1A	apse1-az2
	AP-Soudost-1B	apse1-az1
	AP-Soudost-1C	apse1-az3
il-central-1	il-Central-1a	ilc1-az1
	il-central-1b	ilc1-az2
	il-central-1c	ilc1-az3
ap-northeast-2	ap-nordost-2a	apne2-az1
	ap-nordost-2b	apne2-az2
	ap-nordost-2c	apne2-az3
ap-northeast-1	ap-northeast-1a	apne1-az4
	ap-northeast-1c	apne1-az1
	ap-northeast-1d	apne1-az2
us-east-2	us-east-2a	use2-az1
	us-east-2b	use2-az2
	us-east-2c	use2-az3

1. Wählen Sie in der Amazon VPC-Konsole Create VPC aus.
2. Wählen Sie VPC und mehr aus, um automatisch eine VPC mit öffentlichen und privaten Subnetzen zu erstellen.
3. Konfigurieren Sie die folgenden Einstellungen:
   - Anzahl der Availability Zones: 2 oder mehr
   - Anzahl der öffentlichen Subnetze: Eines pro AZ. In diesem Beispiel 2

- Anzahl der privaten Subnetze: Eines pro AZ. In diesem Beispiel 2
- NAT-Gateways: 1 pro AZ (für die Produktion) oder 1 (für development/testing)
- VPC-Endpunkte: S3-Gateway-Endpunkt (optional — Amazon S3-Verkehr in der Region wird standardmäßig über die HealthOmics Service-VPC geleitet)

Wenn Sie Ihre HealthOmics VPC-Konfiguration erstellen, geben Sie die privaten Subnetze an. Die Läufe verwenden das NAT-Gateway im öffentlichen Subnetz, um das Internet zu erreichen.

VPC-Endpunkte für AWS service

Sie können VPC-Endpunkte so konfigurieren, dass Läufe auf AWS Dienste zugreifen können, ohne das öffentliche Internet zu durchqueren. Dies verbessert die Sicherheit und kann die Datenübertragungskosten senken.

> **Important**
>
> Wenn Ihre Workflow-Definition auf AWS Dienste zugreifen muss (wie Amazon Athena-Abfragen, Amazon DynamoDB-Operationen oder andere API-Aufrufe), müssen Sie sicherstellen, dass die erforderlichen VPC-Endpunkte in Ihrer VPC konfiguriert sind. Ohne die entsprechenden Endpunkte kann Ihr Workflow mit Authentifizierungs- oder Konnektivitätsfehlern fehlschlagen.

> **Note**
>
> In-Region Der Amazon S3-Verkehr wird standardmäßig über die HealthOmics Service-VPC geleitet. Wenn Sie Amazon S3-Schnittstellenendpunkte konfigurieren, wird der Datenverkehr stattdessen über Ihre VPC geleitet. Wir empfehlen die Verwendung von Amazon S3-Gateway-Endpunkten für beste Leistung und Kostenoptimierung. Weitere Informationen finden Sie im Handbuch unter Gateway-Endpunkte für Amazon S3. AWS PrivateLink

In der folgenden Tabelle sind häufig verwendete VPC-Endpunkte für Läufe aufgeführt: HealthOmics

Service	Endpunkttyp	Endpoint name (Endpunkt name)
Amazon S3	Gateway	com.amazonaws. *region*.s3
Amazon S3 Tables	Schnittstelle	com.amazonaws. *region*.s3-Tabellen
Amazon ECR (API)	Schnittstelle	com.amazonaws. *region*.ecr.api
Amazon ECR (Docker)	Schnittstelle	com.amazonaws. *region*.ecr.dkr
SSM	Schnittstelle	com.amazonaws. *region*.ssm
CloudWatch Protokolle	Schnittstelle	com.amazonaws. *region*.protokolle
Amazon Athena	Schnittstelle	com.amazonaws. *region*. Athene

Die vollständige Liste der Dienste, auf die Sie über AWS PrivateLink Endpunkte zugreifen können, finden Sie unter AWS Dienste, die mit integriert sind. AWS PrivateLink Eine ausführliche Anleitung zur Einrichtung von Endpunkten finden Sie AWS PrivateLink im AWS PrivateLink Handbuch unter Auf AWS Dienste zugreifen.

Anforderungen an das NAT Gateway

Für Läufe, die einen öffentlichen Internetzugang erfordern:

- NAT Gateway muss in einem öffentlichen Subnetz bereitgestellt werden
- Das öffentliche Subnetz muss eine Route zu einem Internet-Gateway haben
- Private Subnetze (in denen Läufe ausgeführt werden) müssen über Routen zum NAT-Gateway verfügen

> **Note**
>
> Für NAT-Gateways fallen stündliche Gebühren und Datenverarbeitungsgebühren an. Zur Kostenoptimierung sollten Sie erwägen, VPC-Endpunkte für den AWS Servicezugriff zu verwenden, anstatt das Routing über das NAT-Gateway durchzuführen.

Sicherheitsgruppenkonfiguration

Konfigurieren Sie Ihre Sicherheitsgruppen so, dass ausgehender Datenverkehr zu den Zielen zugelassen wird, auf die Ihre Läufe zugreifen müssen:

- Öffentlicher Internetzugang — Erlaubt ausgehenden HTTPS-Verkehr (Port 443). Fügen Sie nach Bedarf Regeln für andere Protokolle hinzu, z. B. HTTP (Port 80).
- Spezifische Dienste — Konfigurieren Sie Regeln, die Ihren Anforderungen entsprechen.
- On-premises Ressourcen — Erlauben Sie den Datenverkehr zu Ihren VPN- oder CIDR-Bereichen.

Das folgende Beispiel zeigt eine Sicherheitsgruppenregel für den öffentlichen Internetzugang:

Typ	Protocol (Protokoll)	Port-Bereich	Ziel	Description
HTTPS	TCP	443	0.0.0. 0/0	Erlaube HTTPS im Internet

Konfiguration der Routing-Tabelle

Stellen Sie sicher, dass Ihre privaten Subnetze über Routing-Tabelleneinträge verfügen, die den internetgebundenen Datenverkehr an ein NAT-Gateway weiterleiten:

Bestimmungsort	Target
10.0.0. 0/16	local
0,0.0. 0/0	nat-xxxxxxxxx

Für den Zugriff auf lokale Ressourcen konfigurieren Sie Routen zu einem Virtual Private Gateway oder Gateway.

IAM-Berechtigungen für AWS service

Wenn Ihre Workflow-Aufgaben im VPC-Netzwerkmodus auf AWS Dienste wie Amazon Athena, AWS Glue oder Amazon DynamoDB zugreifen, müssen Sie der Servicerolle, die Sie an die API übergeben, die erforderlichen Berechtigungen hinzufügen. `StartRun` Ohne diese Berechtigungen schlagen Ihre Workflow-Aufgaben mit oder ohne Fehler fehl. `AccessDeniedException` `UnauthorizedException`

> **Important**
>
> Die Berechtigungen für Servicerollen sind unabhängig von der VPC-Netzwerkkonfiguration. Selbst bei ordnungsgemäß konfigurierten VPC-Endpunkten und Sicherheitsgruppen schlägt Ihr Workflow fehl, wenn der Servicerolle die erforderlichen IAM-Berechtigungen fehlen.

Wenn Ihr Workflow aufgrund von Berechtigungsfehlern fehlschlägt, überprüfen Sie die CloudWatch Protokollprotokolle für Ihre Workflow-Ausführung. Zu den häufigsten Fehlermeldungen gehören `AccessDeniedException: You are not authorized to perform: action on the resource` (der Servicerolle fehlt die erforderliche IAM-Berechtigung) oder `UnrecognizedClientException: The security token included in the request is invalid` (die Vertrauensrichtlinie für die Servicerolle ist möglicherweise falsch konfiguriert oder der übergebene Rollen-ARN `StartRun` ist falsch).

Allgemeine Serviceberechtigungen

Die folgenden Beispiele zeigen IAM-Berechtigungen für häufig verwendete AWS Dienste in Workflows im VPC-Modus. Fügen Sie diese Berechtigungen Ihrer Servicerollenrichtlinie hinzu, je nachdem, auf welche Dienste Ihr Workflow zugreift.

Example Berechtigungen

Für Workflows, die Abfragen ausführen:

```
{
  "Effect": "Allow",
  "Action": [
    "athena:StartQueryExecution",
```

```
    "athena:GetQueryExecution",
    "athena:GetQueryResults",
    "athena:StopQueryExecution"
  ],
  "Resource": "arn:aws:athena:region:account-id:workgroup/workgroup-name"
}
```

Example AWS Glue Berechtigungen für den Datenkatalog

Für Workflows, die auf AWS Glue Datenbanken und Tabellen zugreifen (häufig mit Amazon Athena verwendet):

```
{
  "Effect": "Allow",
  "Action": [
    "glue:GetDatabase",
    "glue:GetTable",
    "glue:GetPartitions",
    "glue:CreateTable",
    "glue:UpdateTable"
  ],
  "Resource": [
    "arn:aws:glue:region:account-id:catalog",
    "arn:aws:glue:region:account-id:database/database-name",
    "arn:aws:glue:region:account-id:table/database-name/*"
  ]
}
```

> **Note**
>
> Wenn Sie die Berechtigungen für Ihren AWS Glue Datenkatalog AWS Lake Formation zur Verwaltung verwenden, müssen Sie auch die entsprechenden Lake Formation-Berechtigungen gewähren. Weitere Informationen finden Sie unter Lake Formation-Berechtigungen im AWS Lake Formation Entwicklerhandbuch.

Example DynamoDB-Berechtigungen

Für Workflows, die aus DynamoDB-Tabellen lesen oder in DynamoDB-Tabellen schreiben:

```
{
```

```
  "Effect": "Allow",
  "Action": [
    "dynamodb:GetItem",
    "dynamodb:PutItem",
    "dynamodb:Query",
    "dynamodb:Scan"
  ],
  "Resource": "arn:aws:dynamodb:region:account-id:table/table-name"
}
```

Example Berechtigungen für Amazon S3-Tabellen

Für Workflows, die aus Amazon S3-Tabellen lesen oder in Amazon S3-Tabellen schreiben:

```
{
  "Effect": "Allow",
  "Action": [
    "s3tables:GetTableData",
    "s3tables:PutTableData"
  ],
  "Resource": "arn:aws:s3tables:region:account-id:bucket/bucket-name/table/table-id"
}
```

> **Note**
>
> Amazon S3 Tables verwendet einen anderen Endpunkt als Amazon S3. Sie müssen einen VPC-Endpunkt für Amazon S3-Tabellen konfigurieren und sicherstellen, dass Ihre Sicherheitsgruppe ausgehenden HTTPS-Verkehr (Port 443) zum Amazon S3 Tables-Service zulässt.

Testen der VPC-Konnektivität

Stellen Sie vor dem Ausführen von Produktionsworkflows sicher, dass Ihre VPC-Konfiguration die Konnektivität zu den erforderlichen externen Diensten ermöglicht.

Erstellen Sie einen Test-Workflow

Erstellen Sie einen einfachen Workflow, der die Konnektivität zu Ihrem externen Dienst testet. Erstellen Sie beispielsweise einen Workflow, der versucht, eine TCP-Verbindung zu einem Zieldienstendpunkt herzustellen.

Führen Sie den Test aus

```
aws omics start-run \
  --workflow-id test-workflow-id \
  --role-arn role-arn \
  --output-uri s3://bucket-name/test-outputs/ \
  --networking-mode VPC \
  --configuration-name configuration-name \
  --parameters file://test-parameters.json
```

Überprüfen der Ergebnisse

Überprüfen Sie die Workflow-Ausgabe, um die erfolgreiche Konnektivität zu bestätigen:

```
{
  "connectivity_test.result": "Testing connection to external service...\nSUCCESS:
 Connection successful!\nTest completed"
}
```

Wenn der Test fehlschlägt, überprüfen Sie Folgendes:

- Sicherheitsgruppenregeln ermöglichen ausgehenden Datenverkehr zu den erforderlichen Ports und Zielen.
- Routing-Tabellen leiten den Datenverkehr für den Internetzugang an ein NAT-Gateway weiter.
- Auf den externen Dienst kann von Ihrem Netzwerk aus zugegriffen werden.
- In Ihrem Konto sind ausreichend ENIs verfügbar.
- Das NAT-Gateway befindet sich in einem öffentlichen Subnetz mit einer Route zu einem Internet-Gateway.

> **Note**
>
> Der Netzwerkdurchsatz beginnt bei 10 Gbit/s pro ENI und skaliert bei anhaltendem Datenverkehr über einen Zeitraum von 60 Minuten auf bis zu 100 Gbit/s. Für Workflows mit unmittelbaren Anforderungen an einen hohen Durchsatz wenden Sie sich bitte an den Support. AWS

Beispiele

Zugriff auf NCBI-Daten mit API-Authentifizierung

Dieses Beispiel zeigt, wie Sie mithilfe der NCBI-Datensatz-API mit Authentifizierung auf NCBI-Daten zugreifen.

> ⚠ Bewährte Methoden für den Zugriff auf NCBI-Ressourcen
>
> Kunden sollten nach Möglichkeit die REST-API verwenden und einen von NCBI bereitgestellten API-Schlüssel verwenden. Anfragen für den Zugriff auf NCBI-Ressourcen, wie HTTP- und FTP-Anfragen für öffentliche Daten, kommen von NCBI HealthOmics und werden zu dem von NCBI festgelegten Tarif von Drittanbietern gedrosselt. Bei Spitzenauslastung kann es aufgrund von Drosselungsfehlern zu Laufausfällen kommen. Wir empfehlen Benutzern, ihren eigenen NCBI-API-Schlüssel zu erwerben und spezielle APIs zu verwenden, um eine höhere Parallelität und ein besseres Entwicklungserlebnis zu ermöglichen.

Um Ihren NCBI-API-Schlüssel zu erhalten, besuchen Sie die Dokumentation zu den NCBI-API-Schlüsseln.

Beispiel für eine Workflow-Definition:

```
version 1.0
#WORKFLOW DEFINITION

# Meant to be used as integration test for public internet access via VPC tunnel
workflow TestFlow {
    input {
        String ncbi_api_url = "https://api.ncbi.nlm.nih.gov/datasets/v2/gene/accession/
NM_021803.4?api_key=<YOUR_API_KEY>"
    }

    call DataProcessTask{
        input:
            ncbi_api_url = ncbi_api_url,
    }

    output {
```

```
        File output_file = DataProcessTask.output_file
    }

}

#Task Definitions
task DataProcessTask {
    input {
        String ncbi_api_url
    }

    command <<<
        set -eu
        # Download file from NCBI Datasets API with API key
        curl -fsSL "~{ncbi_api_url}" -o gene_data.json

        # Add data processing task here
        cat gene_data.json > processed_data.json

        # Echo the content to output file
        cat processed_data.json > outfile.txt
    >>>

    output {
        File output_file = "outfile.txt"
    }
}
```

Die wichtigsten Punkte:

- <YOUR_API_KEY>Ersetzen Sie ihn durch Ihren tatsächlichen NCBI-API-Schlüssel
- Der Workflow verwendet HTTPS, um auf die NCBI-Datensatz-API zuzugreifen
- Der API-Schlüssel wird als URL-Parameter übergeben
- Dieser Ansatz bietet höhere Ratenbeschränkungen (10 Anfragen pro Sekunde) im Vergleich zum nicht authentifizierten Zugriff (5 Anfragen pro Sekunde)

Weitere Informationen zu NCBI-API-Schlüsseln und Ratenbeschränkungen finden Sie in der API-Dokumentation zu NCBI-Datensätzen.

Bewährte Methoden

1. Verwenden Sie VPC-Endpunkte für Dienste. AWS Konfigurieren Sie VPC-Endpunkte für Amazon S3, Amazon ECR und andere AWS Dienste, um die NAT-Gateway-Kosten zu senken und die Leistung zu verbessern. Weitere Informationen finden Sie unter VPC-Endpunkte für AWS service.
2. Überwachen Sie die Netzwerkkosten. Bei VPC-Netzwerken fallen Kosten für NAT-Gateways, Datenübertragung und ENIs an. Überwachen Sie Ihre Nutzung mit Cost Explorer. AWS
3. Planen Sie Verfügbarkeitszonen ein. Stellen Sie sicher, dass sich Ihre Subnetze über die Availability Zones erstrecken, in denen HealthOmics die Workflow-Platzierung unterstützt wird.
4. Verwenden Sie NAT-Gateways in jeder AZ. Stellen Sie für Produktionsworkloads in jeder Availability Zone ein NAT-Gateway bereit, um Redundanz zu gewährleisten.

Ready2Run-Workflows in HealthOmics

Ready2Run-Workflows sind vorkonfigurierte Workflows, die von Drittanbietern veröffentlicht wurden. Einige Herausgeber, wie Sentieon Inc., bieten Workflows auf Abonnementbasis an. Für andere Ready2Run-Workflows ist kein Abonnement erforderlich, und einige Workflows sind Open Source, wie z. B. die NF-Core-Workflows.

Ready2Run-Workflows eignen sich gut für die folgenden Szenarien:

- Sie möchten sich auf die Analyse der Pipeline-Ergebnisse und die Generierung von Ergebnissen konzentrieren, ohne die zugrunde liegende Infrastruktur einrichten zu müssen.
- Sie möchten Ihre Ergebnisse mithilfe etablierter Workflows replizieren.
- Als Softwareentwickler möchten Sie Ihre Anwendung direkt in das HealthOmics SDK integrieren.

HealthOmics unterstützt die Versionierung für Ready2Run-Workflows. Für einen Ready2Run-Workflow, der Versionen anbietet, können Sie den Versionsnamen angeben, wenn Sie einen Lauf starten.

Alle Ready2Run-Workflows bieten Protokolle, einschließlich CloudWatch Protokolle, die Sie zur Fehlerbehebung verwenden können.

> **Note**
>
> Sentieon Ready2Run-Workflows basieren auf Abonnements. Wenn Sie einen Sentieon Ready2Run-Workflow zum ersten Mal in einem Konto ausführen, erstellt Sentieon automatisch eine zweiwöchige Testlizenz für Ihren. AWS-Konto Die Lizenz ist für alle Sentieon Ready2Run-Workflows gültig. Nach Ablauf des Testzeitraums können Sie eine permanente Lizenz oder eine Verlängerung der Evaluierungslizenz beantragen. Details dazu finden Sie unter Subscribing to Sentieon Ready2Run workflows.

Themen

- Verfügbare Ready2Run-Workflows in HealthOmics
- Abonnieren Sie die Sentieon Ready2Run-Workflows
- HealthOmics Ready2Run-Workflows über die Konsole starten
- HealthOmics Ready2Run-Workflows mithilfe der API starten

Verfügbare Ready2Run-Workflows in HealthOmics

In der folgenden Tabelle sind die Ready2Run-Workflows aufgeführt, die in verfügbar sind. HealthOmics

Sie können sich bei der HealthOmicsKonsole anmelden, um detaillierte Informationen zu diesen Workflows, einschließlich Eingabeparametern und Workflow-Diagrammen, einzusehen. Preisinformationen zu Ready2Run-Workflows finden Sie unter HealthOmics Preise.

> **Note**
>
> Jeder Ready2Run-Workflow hat eine maximale Eingabedateigröße. Diese maximalen Dateigrößen sind nicht einstellbar.

Workflow-Name	Publisher	Abonnement erforderlich?	Maximale Größe der Eingabedatei (GiB)	Geschätzte Laufzeit (HH:MM)
AlphaFold für 601-1200 Reste	Google DeepMind	Nein	1	11:15
AlphaFold für bis zu 600 Reste	Google DeepMind	Nein	1	7:30
Bases2Fastq für 2x150	Element Biowissenschaften	Nein	1000	1:45
Bases2Fastq für 2x300	Element Biowissenschaften	Nein	1000	1:30
Bases2Fastq für 2x75	Element Biowissenschaften	Nein	500	0:45

Workflow-Name	Publisher	Abonnement erforderlich?	Maximale Größe der Eingabedatei (GiB)	Geschätzte Laufzeit (HH:MM)
ESMFold für bis zu 800 Rückstände	Meta-Forschung	Nein	1	0:15
GATK-BP fq2bam	Breites Institut	Nein	64	10:10
GATK-BP-Keimbahn bam2vcf für das 30-fache Genom	Breites Institut	Nein	39	2:45
GATK-BP-Keimbahn fq2vcf für das 30-fache Genom	Breites Institut	Nein	64	12:30
GATK-BP Somatisches WES bam2vcf	Breites Institut	Nein	86	1:30
NVIDIA Parabricks BAM2 FQ2 BAM WGS für bis zu 30X	NVIDIA Corporation	Nein	80	1:39
NVIDIA Parabricks BAM2 FQ2 BAM WGS für bis zu 50X	NVIDIA Corporation	Nein	120	2:45

Workflow-Name	Publisher	Abonnement erforderlich?	Maximale Größe der Eingabedatei (GiB)	Geschätzte Laufzeit (HH:MM)
NVIDIA Parabricks BAM2 FQ2 BAM WGS für bis zu 5X	NVIDIA Corporation	Nein	20	0:18
NVIDIA Parabricks FQ2 BAM WGS für bis zu 30X	NVIDIA Corporation	Nein	71	1:00
NVIDIA Parabricks FQ2 BAM WGS für bis zu 50X	NVIDIA Corporation	Nein	137	1:45
NVIDIA Parabricks FQ2 BAM WGS für bis zu 5X	NVIDIA Corporation	Nein	13	0:15
NVIDIA Parabricks Germline DeepVariant WGS für bis zu 30X	NVIDIA Corporation	Nein	71	2:00
NVIDIA Parabricks Germline DeepVariant WGS für bis zu 50X	NVIDIA Corporation	Nein	137	3:30

Workflow-Name	Publisher	Abonnement erforderlich?	Maximale Größe der Eingabedatei (GiB)	Geschätzte Laufzeit (HH:MM)
NVIDIA Parabricks Germline DeepVariant WGS für bis zu 5X	NVIDIA Corporation	Nein	12	0:30
NVIDIA Parabricks Germline HaplotypeCaller WGS für bis zu 30X	NVIDIA Corporation	Nein	71	1:15
NVIDIA Parabricks Germline HaplotypeCaller WGS für bis zu 50X	NVIDIA Corporation	Nein	137	2:00
NVIDIA Parabricks Germline HaplotypeCaller WGS für bis zu 5X	NVIDIA Corporation	Nein	13	0:15
NVIDIA Parabricks Somatic Mutect2 WGS für bis zu 50X	NVIDIA Corporation	Nein	196	0:45

Workflow-Name	Publisher	Abonnement erforderlich?	Maximale Größe der Eingabedatei (GiB)	Geschätzte Laufzeit (HH:MM)
sc RNAseq mit Kallisto BUStools	NF-Kern	Nein	119	1:30
Sc RNAseq mit Lachs in Alevin-Fry	NF-Kern	Nein	119	2:30
sc RNAseq mit STARsolo	NF-Kern	Nein	119	2:30
Sention Germline BAM WES für bis zu 300x	Sentieon, Inc.	Ja	9	1:00
Sentieon Germline BAM WGS für bis zu 32x	Sentieon, Inc.	Ja	18	1:30
Sention Germline FASTQ WES für bis zu 100x	Sentieon, Inc.	Ja	5	0:45
Sention Germline FASTQ WES für bis zu 300x	Sentieon, Inc.	Ja	26	2:00
Sentieon Germline FASTQ WGS für bis zu 32x	Sentieon, Inc.	Ja	51	3:30

Workflow-Name	Publisher	Abonnement erforderlich?	Maximale Größe der Eingabedatei (GiB)	Geschätzte Laufzeit (HH:MM)
Sention für ONT LongRead	Sentieon, Inc.	Ja	25	1:30
Sention LongRead für PacBio HiFi	Sentieon, Inc.	Ja	58	4:00
Sentieon Somatic WES	Sentieon, Inc.	Ja	50	2:30
Sention Somatic WGS	Sentieon, Inc.	Ja	113	4:30
Ultima Genomics DeepVariant für bis zu 40x	Ultima Genomics	Nein	91	1:55

Wenn Sie einen Ready2Run-Workflow verwenden, ist Ihr Workflow vorkonfiguriert und kann nicht bearbeitet werden. Im Gegensatz zu privaten Workflows unterstützen Ready2Run-Workflows Folgendes nicht:

- Erhöhung der maximalen Größe der Eingabedatei
- Änderung der Rechenressourcen oder des Laufspeichers
- Änderung der Workflow-Definition oder der Container
- Hinzufügen von Läufen zu einer Ausführungsgruppe
- Den Workflow teilen

Wenn der Herausgeber den Ready2Run-Workflow freigegeben hat GitHub, können Sie Ihren eigenen privaten Workflow auf der Grundlage des Ready2Run-Workflows erstellen. Die folgende Tabelle enthält Links zu GitHub Workflows für jeden Herausgeber.

Publisher	Workflows auf GitHub
Google DeepMind, Metarecherche	Arbeitsabläufe zur Proteinfaltung
Element-Biowissenschaften	Für Informationen wenden Sie sich bitte an Element Biosciences
Breites Institut	GATK-Workflows
NVIDIA Corporation	Arbeitsabläufe bei Parabricks
NF-Kern	NF-Core-Workflows
Empfindung	Arbeitsabläufe bei Sentieon
Ultima Genomik	Arbeitsabläufe bei Ultima Genomics

Abonnieren Sie die Sentieon Ready2Run-Workflows

Sentieon Ready2Run-Workflows basieren auf Abonnements. Wenn Sie einen Sentieon Ready2Run-Workflow zum ersten Mal in einem Konto ausführen, erstellt Sentieon automatisch eine zweiwöchige Testlizenz für Ihren. AWS-Konto Die Lizenz ist für alle Sentieon Ready2Run-Workflows gültig. Nach Ablauf des Testzeitraums können Sie eine permanente Lizenz oder eine Verlängerung der Evaluierungslizenz beantragen.

Gehen Sie wie folgt vor, um die Sentieon Ready2Run-Workflows zu abonnieren:

- Finden Sie Ihre AWS Canonical Benutzer-ID, indem Sie diesen Anweisungen folgen.
- Senden Sie eine E-Mail an die Sentieon-Supportgruppe (support@sentieon.com), um eine Softwarelizenz anzufordern. Geben Sie in der AWS E-Mail Ihre Canonical Benutzer-ID an.

HealthOmics Ready2Run-Workflows über die Konsole starten

Die Verwendung von Ready2Run-Workflows in der Konsole ähnelt der Verwendung eines privaten Workflows. Ein wesentlicher Unterschied besteht darin, dass der Workflow-Herausgeber Beispieldaten bereitstellt, sodass Sie den Workflow ausprobieren können, ohne Ihre eigenen Daten zu erstellen.

Um einen Ready2Run-Workflow in der Konsole zu verwenden

1. Öffnen Sie die HealthOmics -Konsole.
2. Öffnen Sie bei Bedarf den linken Navigationsbereich (•). Wählen Sie Ready2Run-Workflows.
3. Wählen Sie auf der Seite Ready2Run-Workflows den Workflow aus, den Sie verwenden möchten. Die Konsole öffnet die Detailseite für diesen Workflow.
4. Auf der Registerkarte „Details“ werden Informationen wie Name, Listenpreis pro Lauf, Beschreibung, Workflow-Sprachtyp, Laufspeicherkapazität, Status, Erstellungsdatum und Parameter mit Beschreibungen aufgeführt. Auf der Registerkarte „Details“ erfahren Sie auch, ob für den Workflow ein Abonnement erforderlich ist.
5. Um den Workflow zu verwenden, wählen Sie Create run
6. Geben Sie auf der Seite „Ausführungsdetails angeben“ einen Namen für den Lauf ein. Optional können Sie die Workflow-Version angeben. Sie können dem Lauf auch eine Ausführungspriorität hinzufügen.
7. Geben Sie einen Amazon S3 S3-Speicherort für die Run-Ausgabe ein oder wählen Sie ihn aus.
8. Wählen Sie für den Aufbewahrungsmodus für Metadaten ausführen aus, ob Ausführungsmetadaten beibehalten oder entfernt werden sollen.
9. Wählen Sie im Bereich „Servicerolle“ aus, ob Sie eine bestehende Servicerolle verwenden oder eine neue erstellen möchten.
10. (Optional) Fügen Sie Tags hinzu, um Ihren Lauf leichter identifizieren und verwalten zu können.
11. Wählen Sie Weiter aus.
12. Wählen Sie auf der Seite „Parameter hinzufügen“ eine der Optionen aus, um die Werte der Ausführungsparameter hinzuzufügen:

    - Wählen Sie eine Parameterdatei (im JSON-Format) von einem Amazon S3 S3-Speicherort aus.
    - Wählen Sie eine Parameterdatei (im JSON-Format) von Ihrem lokalen Laufwerk aus.
    - Geben Sie die Parameterwerte manuell ein.
    - Führen Sie den Workflow mit den vom Workflow-Herausgeber bereitgestellten Ready2Run-Beispieldaten aus.
13. Wenn Sie eine JSON-Datei hochladen, analysiert die Konsole die Datei und führt eine Inline-Validierung durch. Anschließend können Sie die Werte Ihrer Parameter nach Bedarf manuell aktualisieren.
14. Wählen Sie Weiter aus.

15. Überprüfen Sie Ihre Eingaben und wählen Sie dann Start run.

HealthOmics Ready2Run-Workflows mithilfe der API starten

Die meisten API-Operationen verhalten sich bei Ready2Run-Workflows und privaten Workflows ähnlich.

Um eine Liste der verfügbaren Ready2Run-Workflows zurückzugeben, verwenden Sie list-workflows, wobei der Parameter auf RUN gesetzt ist. `type` READY2

```
aws omics list-workflows --type READY2RUN
```

Nachdem Sie anhand der Antwort auf die Liste der Workflows den auszuführenden Workflow identifiziert haben, können Sie get-workflow mit dem Parameter verwenden, um weitere Details abzurufen. `--id`

```
aws omics get-workflow --type READY2RUN --id workflow id
```

Um einen Ready2Run-Workflow auszuführen, können Sie den API-Vorgang start-run verwenden, wobei der Parameter workflow-type auf gesetzt ist, wie im folgenden Beispiel gezeigt READY2RUN

```
aws-omics start-run \
  --workflow-type READY2RUN \
  --workflow-id workflow id \
  --output-uri &example-s3-bucket; \
  --role-arn arn:aws:iam::1234567892012:role/service-role/OmicsWorkflow-20221004T164236
 \
  --parameters file:///path/to/parameters.json
```

Um eine Workflow-Version anzugeben, verwenden Sie den Parameter workflow-version, wie in diesem Beispiel gezeigt.

```
aws-omics start-run \
  --workflow-type READY2RUN \
  ...
  --version-name '3.0.0'
```

Um Ihren Lauf zu überwachen, können Sie den API-Vorgang get-run verwenden, wie in der Abbildung gezeigt.

```
aws-omics get-run \
  --id run id
```

HealthOmics Lager

Verwenden HealthOmics Speicher zum effizienten und kostengünstigen Speichern, Abrufen, Organisieren und Teilen von Genomdaten. HealthOmics Storage versteht die Beziehungen zwischen verschiedenen Datenobjekten, sodass Sie definieren können, welche Lesesätze aus denselben Quelldaten stammen. Auf diese Weise können Sie die Herkunft der Daten ermitteln.

Daten, die gespeichert sind in `ACTIVE` Der Status ist sofort abrufbar. Daten, auf die seit 30 Tagen oder länger nicht zugegriffen wurde, werden gespeichert in `ARCHIVE` Staat. Um auf archivierte Daten zuzugreifen, können Sie sie über die API-Operationen oder die Konsole reaktivieren.

HealthOmics Sequenzspeicher sind so konzipiert, dass die Inhaltsintegrität von Dateien gewahrt bleibt. Die bitweise Äquivalenz zwischen importierten und exportierten Dateien bleibt jedoch aufgrund der Komprimierung beim aktiven und beim Archiv-Tiering nicht erhalten.

Während der Einnahme HealthOmics generiert ein Entitäts-Tag, oder HealthOmics ETag, um die Überprüfung der Inhaltsintegrität Ihrer Datendateien zu ermöglichen. Sequenzierungsteile werden identifiziert und als ETag auf der Quellebene eines Lesesatzes erfasst. Die ETag-Berechnung ändert nicht die eigentliche Datei oder die Genomdaten. Nachdem ein Lesesatz erstellt wurde, sollte sich das ETag während des gesamten Lebenszyklus der Read-Set-Quelle nicht ändern. Das bedeutet, dass der erneute Import derselben Datei dazu führt, dass derselbe ETag-Wert berechnet wird.

Themen

- HealthOmics ETags und Datenherkunft
- Erstellen einer HealthOmics Referenzspeicher
- Erstellen einer HealthOmics Sequenzspeicher
- Löschen HealthOmics Referenz- und Sequenzspeicher
- Importieren von Lesesätzen in eine HealthOmics Sequenzspeicher
- Direkter Upload auf eine HealthOmics Sequenzspeicher
- Exportieren HealthOmics Sätze in einen Amazon S3-Bucket lesen
- Zugreifen HealthOmics Sätze mit Amazon S3-URIs lesen
- Lesesätze aktivieren in HealthOmics

HealthOmics ETags und Datenherkunft

EIN HealthOmics ETag (Entitäts-Tag) ist ein Hash des aufgenommenen Inhalts in einem Sequenzspeicher. Dies vereinfacht das Abrufen und Verarbeiten von Daten und gewährleistet gleichzeitig die Inhaltsintegrität der aufgenommenen Datendateien. Das ETag spiegelt Änderungen am semantischen Inhalt des Objekts wider, nicht an seinen Metadaten. Der angegebene Lesesatztyp und der angegebene Algorithmus bestimmen, wie das ETag berechnet wird. Die ETag-Berechnung ändert nicht die eigentliche Datei oder die Genomdaten. Wenn das Dateitypschema des Lesesatzes dies zulässt, aktualisiert der Sequenzspeicher Felder, die mit der Datenherkunft verknüpft sind.

Dateien haben eine bitweise Identität und eine semantische Identität. Die bitweise Identität bedeutet, dass die Bits einer Datei identisch sind, und eine semantische Identität bedeutet, dass der Inhalt einer Datei identisch ist. Die semantische Identität ist widerstandsfähig gegenüber Änderungen an Metadaten und Komprimierung, da sie die Inhaltsintegrität der Datei erfasst.

Sätze einlesen HealthOmics Sequenzspeicher durchlaufen compression/decompression Zyklen und Verfolgung der Datenherkunft während des gesamten Lebenszyklus eines Objekts. Während dieser Verarbeitung kann sich die bitweise Identität einer aufgenommenen Datei ändern und es wird erwartet, dass sie sich jedes Mal ändert, wenn eine Datei aktiviert wird. Die semantische Identität der Datei bleibt jedoch erhalten. Die semantische Identität wird als erfasst HealthOmics Entitäts-Tag oder ETag, das beim Erfassen des Sequenzspeichers berechnet wird und als Lesesatz-Metadaten verfügbar ist.

Wenn das Dateitypschema des Read-Sets dies zulässt, werden die Felder für Sequenzspeicher-Aktualisierungen mit der Datenherkunft verknüpft. Für uBAM-, BAM- und CRAM-Dateien gibt es ein neues `@CO` oder `Comment` Tag wird zur Kopfzeile hinzugefügt. Der Kommentar enthält die Sequenzspeicher-ID und den Zeitstempel der Aufnahme.

Amazon S3-eTags

Beim Zugriff auf eine Datei mithilfe der Amazon S3-URI können Amazon S3-API-Operationen auch Amazon S3-ETag- und Prüfsummenwerte zurückgeben. Die Amazon S3-ETag- und Prüfsummenwerte unterscheiden sich von den HealthOmics ETags, weil sie die bitweise Identität der Datei darstellen. Weitere Informationen zu beschreibenden Metadaten und Objekten finden Sie im Amazon S3 Dokumentation zur Objekt-API. Die Amazon S3-ETag-Werte können sich mit jedem Aktivierungszyklus eines Lesesatzes ändern, und Sie können sie verwenden, um das Lesen einer Datei zu überprüfen. Zwischenspeichern Sie Amazon S3-ETag-Werte jedoch nicht, um sie während des Lebenszyklus der Datei für die Überprüfung der Dateiidentität zu verwenden, da sie

nicht konsistent bleiben. Im Gegensatz dazu HealthOmics ETag bleibt während des gesamten Lebenszyklus des Lesesatzes konsistent.

Wie HealthOmics berechnet ETags

Das ETag wird aus einem Hash des aufgenommenen Dateiinhalts generiert. Die ETag-Algorithmusfamilie ist standardmäßig auf MD5up gesetzt, kann aber bei der Erstellung des Sequenzspeichers anders konfiguriert werden. Wenn das ETag berechnet wird, werden der Algorithmus und die berechneten Hashes zum Lesesatz hinzugefügt. Die unterstützten MD5-Algorithmen für Dateitypen lauten wie folgt.

- FastQ_MD5up — Berechnet den MD5-Hash einer unkomprimierten, vollständigen FASTQ-Readset-Quelle.
- BAM_MD5UP — Berechnet den MD5-Hash des Alignment-Abschnitts einer unkomprimierten BAM- oder uBAM-Lesatzquelle, wie sie im SAM dargestellt wird, auf der Grundlage der verknüpften Referenz, falls eine verfügbar ist.
- CRAM_MD5UP — Berechnet den MD5-Hash des Alignment-Abschnitts der unkomprimierten CRAM-Readset-Quelle, wie er im SAM dargestellt wird, auf der Grundlage der verknüpften Referenz.

> **Note**
>
> MD5-Hashing ist bekanntermaßen anfällig für Kollisionen. Aus diesem Grund könnten zwei verschiedene Dateien dasselbe ETag haben, wenn sie so hergestellt wurden, dass sie die bekannte Kollision ausnutzen.

Die folgenden Algorithmen werden für die SHA256-Familie unterstützt. Die Algorithmen werden wie folgt berechnet:

- FastQ_SHA256UP — Berechnet den SHA-256 Hash einer unkomprimierten, vollständigen FASTQ-Readset-Quelle.
- BAM_SHA256UP — Berechnet den SHA-256 Hash des Alignment-Abschnitts einer unkomprimierten BAM- oder uBAM-Lesatzquelle, wie sie im SAM dargestellt wird, basierend auf der verknüpften Referenz, falls eine verfügbar ist.

- CRAM_SHA256UP — Berechnet den SHA-256 Hash des Alignment-Abschnitts einer unkomprimierten CRAM-Readset-Quelle, wie er im SAM dargestellt wird, basierend auf der verknüpften Referenz.

Die folgenden Algorithmen werden für die SHA512-Familie unterstützt. Die Algorithmen werden wie folgt berechnet:

- FastQ_SHA512UP — Berechnet den SHA-512 Hash einer unkomprimierten, vollständigen FASTQ-Readset-Quelle.
- BAM_SHA512UP — Berechnet den SHA-512 Hash des Alignment-Abschnitts einer unkomprimierten BAM- oder uBAM-Lesatzquelle, wie sie im SAM dargestellt wird, basierend auf der verknüpften Referenz, falls eine verfügbar ist.
- CRAM_SHA512UP — Berechnet den SHA-512 Hash des Alignment-Abschnitts einer unkomprimierten CRAM-Readset-Quelle, wie er im SAM dargestellt wird, basierend auf der verknüpften Referenz.

Erstellen einer HealthOmics Referenzspeicher

Ein Referenzspeicher in HealthOmics ist ein Datenspeicher für die Speicherung von Referenzgenomen. Sie können in jedem einen einzelnen Referenzspeicher haben AWS-Konto und Region. Sie können einen Referenzspeicher mithilfe der Konsole oder der CLI erstellen.

Themen

- Erstellen eines Referenzspeichers mithilfe der Konsole
- Erstellen eines Referenzspeichers mit der CLI

Erstellen eines Referenzspeichers mithilfe der Konsole

So erstellen Sie einen Referenzspeicher

1. Öffnen Sie die HealthOmics -Konsole.
2. Öffnen Sie bei Bedarf den linken Navigationsbereich (≡). Wählen Sie Referenzgeschäft.
3. Wählen Referenzgenome aus den Speicheroptionen für Genomics-Daten.

4. Sie können entweder ein zuvor importiertes Referenzgenom auswählen oder ein neues importieren. Wenn Sie kein Referenzgenom importiert haben, wählen Sie Referenzgenom importieren oben rechts.
5. Auf der Auftrag zum Import von Referenzgenomen erstellen Seite, wählen Sie entweder Schnell erstellen oder Manuell erstellen Option, um einen Referenzspeicher zu erstellen und dann die folgenden Informationen bereitzustellen.

   - Name des Referenzgenoms - Ein eindeutiger Name für diesen Shop.
   - Beschreibung (optional) — Eine Beschreibung dieses Referenzspeichers.
   - IAM-Rolle - Wählen Sie eine Rolle mit Zugriff auf Ihr Referenzgenom aus.
   - Referenz von Amazon S3 — Wählen Sie Ihre Referenzsequenzdatei in einem Amazon S3-Bucket aus.
   - Schlagworte (optional) — Geben Sie bis zu 50 Tags für diesen Referenzspeicher an.

Erstellen eines Referenzspeichers mit der CLI

Das folgende Beispiel zeigt Ihnen, wie Sie einen Referenzspeicher erstellen, indem Sie den AWS CLI. Sie können einen Referenzshop pro AWS Region.

Referenzspeicher unterstützen das Speichern von FASTA-Dateien mit den Erweiterungen `.fasta`, `.fa`, `.fas`, `.fsa`, `.faa`, `.fna`, `.ffn`, `.frn`, `.mpfa`, `.seq`, `.txt`. Das `bgzip` Version dieser Erweiterungen wird ebenfalls unterstützt.

Ersetzen Sie im folgenden Beispiel *`reference store name`* mit dem Namen, den Sie für Ihren Referenzshop gewählt haben.

```
aws omics create-reference-store --name "reference store name"
```

Sie erhalten eine JSON-Antwort mit der Referenzspeicher-ID und dem Namen, dem ARN und dem Zeitstempel, zu dem Ihr Referenzshop erstellt wurde.

```
{
    "id": "3242349265",
    "arn": "arn:aws:omics:us-west-2:555555555555:referenceStore/3242349265",
    "name": "MyReferenceStore",
    "creationTime": "2022-07-01T20:58:42.878Z"
}
```

Sie können die Referenzspeicher-ID zusätzlich verwenden AWS CLI Befehle. Sie können die Liste der Referenz-Store-IDs abrufen, die mit Ihrem Konto verknüpft sind, indem Sie Referenzgeschäfte auflisten Befehl, wie im folgenden Beispiel gezeigt.

```
aws omics list-reference-stores
```

Als Antwort erhalten Sie den Namen Ihres neu erstellten Referenzspeichers.

```
{
    "referenceStores": [
        {
              "id": "3242349265",
              "arn": "arn:aws:omics:us-west-2:555555555555:referenceStore/3242349265",
              "name": "MyReferenceStore",
             "creationTime": "2022-07-01T20:58:42.878Z"
         }
     ]
}
```

Nachdem Sie einen Referenzspeicher erstellt haben, können Sie Importaufträge erstellen, um genomische Referenzdateien in den Speicher zu laden. Dazu müssen Sie eine IAM-Rolle verwenden oder eine erstellen, um auf die Daten zuzugreifen. Es folgt eine Beispielrichtlinie .

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:GetBucketLocation"

            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket1",
                "arn:aws:s3:::amzn-s3-demo-bucket1/*"
            ]
        }
    ]
```

```
}
```

Sie müssen außerdem über eine Vertrauensrichtlinie verfügen, die dem folgenden Beispiel ähnelt.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": [
                   "omics.amazonaws.com"
                ]
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

Sie können jetzt ein Referenzgenom importieren. In diesem Beispiel wird das Genome Reference Consortium Human Build 38 (hg38) verwendet, das frei zugänglich ist und bei der Register offener Daten unter AWS. Der Bucket, der diese Daten hostet, hat seinen Sitz in den USA Ost (Ohio). Um Buckets in anderen Bereichen zu verwenden AWS Regionen: Sie können die Daten in einen Amazon S3-Bucket kopieren, der in Ihrer Region gehostet wird. Verwenden Sie Folgendes AWS CLI Befehl, um das Genom in Ihren Amazon S3-Bucket zu kopieren.

```
aws s3 cp s3://broad-references/hg38/v0/Homo_sapiens_assembly38.fasta s3://amzn-s3-
demo-bucket
```

Sie können dann mit Ihrem Importjob beginnen. Ersetzen *reference store ID*, *role ARN*, und *source file path* mit deinem eigenen Input.

```
aws omics start-reference-import-job --reference-store-id reference store ID --role-
arn role ARN --sources source file path
```

Nachdem die Daten importiert wurden, erhalten Sie die folgende Antwort in JSON.

```
{
        "id": "7252016478",
        "referenceStoreId": "3242349265",
        "roleArn": "arn:aws:iam::111122223333:role/OmicsReferenceImport",
        "status": "CREATED",
        "creationTime": "2022-07-01T21:15:13.727Z"
}
```

Sie können den Status eines Jobs mithilfe des folgenden Befehls überwachen. Ersetzen Sie im folgenden Beispiel *reference store ID* und *job ID* mit Ihrer Referenzshop-ID und der Job-ID, über die Sie mehr erfahren möchten.

```
aws omics get-reference-import-job --reference-store-id reference store ID --id job ID
```

Als Antwort erhalten Sie eine Antwort mit den Details zu diesem Referenzshop und seinem Status.

```
{
    "id": "7252016478",
    "referenceStoreId": "3242349265",
    "roleArn": "arn:aws:iam::555555555555:role/OmicsReferenceImport",
    "status": "RUNNING",
    "creationTime": "2022-07-01T21:15:13.727Z",
    "sources": [
        {
            "sourceFile": "s3://amzn-s3-demo-bucket/Homo_sapiens_assembly38.fasta",
            "status": "IN_PROGRESS",
            "name": "MyReference"
        }
    ]
}
```

Sie können die importierte Referenz auch finden, indem Sie Ihre Referenzen auflisten und sie anhand des Referenznamens filtern. Ersetzen *reference store ID* mit Ihrer Referenz-Shop-ID und fügen Sie einen optionalen Filter hinzu, um die Liste einzugrenzen.

```
aws omics list-references --reference-store-id reference store ID --filter
 name=MyReference
```

Als Antwort erhalten Sie die folgenden Informationen.

```
{
    "references": [
        {
            "id": "1234567890",
            "arn": "arn:aws:omics:us-west-2:555555555555:referenceStore/1234567890/
reference/1234567890",
            "referenceStoreId": "12345678",
            "md5": "7ff134953dcca8c8997453bbb80b6b5e",
            "status": "ACTIVE",
            "name": "MyReference",
            "creationTime": "2022-07-02T00:15:19.787Z",
            "updateTime": "2022-07-02T00:15:19.787Z"
        }
    ]
}
```

Um mehr über die Referenzmetadaten zu erfahren, verwenden Sie die get-reference-metadata API-Betrieb. Ersetzen Sie im folgenden Beispiel *reference store ID* mit Ihrer Referenz-Store-ID und *reference ID* mit der Referenz-ID, über die Sie mehr erfahren möchten.

```
aws omics get-reference-metadata --reference-store-id reference store ID --id reference
 ID
```

Als Antwort erhalten Sie die folgenden Informationen.

```
{
    "id": "1234567890",
    "arn": "arn:aws:omics:us-west-2:555555555555:referenceStore/referencestoreID/
reference/referenceID",
    "referenceStoreId": "1234567890",
    "md5": "7ff134953dcca8c8997453bbb80b6b5e",
    "status": "ACTIVE",
    "name": "MyReference",
    "creationTime": "2022-07-02T00:15:19.787Z",
    "updateTime": "2022-07-02T00:15:19.787Z",
    "files": {
        "source": {
            "totalParts": 31,
            "partSize": 104857600,
            "contentLength": 3249912778
        },
        "index": {
```

```
            "totalParts": 1,
            "partSize": 104857600,
            "contentLength": 160928
        }
    }
}
```

Sie können auch Teile der Referenzdatei herunterladen, indem Sie get-reference. Ersetzen Sie im folgenden Beispiel *reference store ID* mit Ihrer Referenz-Store-ID und *reference ID* mit der Referenz-ID, von der Sie herunterladen möchten.

```
aws omics get-reference --reference-store-id reference store ID --id reference ID --
part-number 1 outfile.fa
```

Erstellen einer HealthOmics Sequenzspeicher

HealthOmics Sequenzspeicher unterstützen das Speichern von Genomdateien in den nicht ausgerichteten Formaten von FASTQ (nur Gzip) und uBAM. Es unterstützt auch die abgestimmten Formate von BAM und CRAM.

Importierte Dateien werden als Lesesätze gespeichert. Sie können Tags zu Lesesätzen hinzufügen und IAM-Richtlinien verwenden, um den Zugriff auf Lesesätze zu steuern. Ausgerichtete Lesesätze benötigen ein Referenzgenom, um genomische Sequenzen auszurichten. Für Lesesätze ohne Ausrichtung ist dies jedoch optional.

Um Lesesätze zu speichern, erstellen Sie zunächst einen Sequenzspeicher. Wenn Sie einen Sequenzspeicher erstellen, können Sie einen optionalen Amazon S3-Bucket als Fallback-Speicherort und den Speicherort angeben, an dem S3-Zugriffsprotokolle gespeichert werden. Der Fallback-Speicherort wird zum Speichern aller Dateien verwendet, die während eines direkten Uploads keinen Lesesatz erstellen können. Ersatzspeicherorte sind für Sequenzspeicher verfügbar, die nach dem 15. Mai 2023 erstellt wurden. Sie geben den Fallback-Speicherort an, wenn Sie den Sequenzspeicher erstellen.

Sie können bis zu fünf Read-Set-Tag-Schlüssel angeben. Wenn Sie einen Lesesatz mit einem Tag-Schlüssel erstellen oder aktualisieren, der einem dieser Schlüssel entspricht, werden die Read-Set-Tags an das entsprechende Amazon S3-Objekt weitergegeben. System-Tags, erstellt von HealthOmics werden standardmäßig weitergegeben.

Themen

- Erstellen eines Sequenzspeichers mithilfe der Konsole
- Einen Sequenzspeicher mit der CLI erstellen
- Aktualisierung eines Sequenzspeichers
- Aktualisierung der Read-Set-Tags für einen Sequenzspeicher
- Genomdateien importieren

Erstellen eines Sequenzspeichers mithilfe der Konsole

So erstellen Sie einen Sequenzspeicher

1. Öffnen Sie die HealthOmics -Konsole.
2. Öffnen Sie bei Bedarf den linken Navigationsbereich (≡). Wählen Sie Sequenzspeicher.
3. Auf der Sequenzspeicher erstellen Seite, geben Sie die folgenden Informationen an

   - Name des Sequenzspeichers - Ein eindeutiger Name für diesen Shop.
   - Beschreibung (optional) — Eine Beschreibung dieses Sequenzspeichers.
4. Für Ausweichort in S3, geben Sie einen Amazon S3-Standort an. HealthOmics verwendet den Ersatzspeicherort zum Speichern aller Dateien, die während eines direkten Uploads keinen Lesesatz erstellen können. Sie müssen Folgendes gewähren HealthOmics Service-Schreibzugriff auf den Amazon S3-Fallback-Speicherort. Eine Beispielrichtlinie finden Sie unter Konfigurieren Sie einen Ausweichort.

   Ersatzspeicherorte sind für Sequenzspeicher, die vor dem 16. Mai 2023 erstellt wurden, nicht verfügbar.
5. (Optional) Für Lesen Sie die Set-Tag-Schlüssel für die S3-Weitergabe, Sie können bis zu fünf Read-Set-Schlüssel eingeben, um sie von einem Lesesatz an die zugrunde liegenden S3-Objekte weiterzuleiten. Indem Sie Tags von einem Lesesatz an das S3-Objekt weitergeben, können Sie S3-Zugriffsberechtigungen auf der Grundlage von Tags gewähren and/or Endbenutzer erhalten Zugriff auf die über Amazon S3 weitergeleiteten TagsObjectTagging API-Betrieb.

   a. Geben Sie einen Schlüsselwert in das Textfeld ein. Die Konsole erstellt ein neues Textfeld, um den nächsten Schlüssel hinzuzufügen.

   b. (Optional) Wählen Sie Entfernen um alle Schlüssel zu entfernen.

6. Unter Datenverschlüsselung, wählen Sie aus, ob die Datenverschlüsselung Eigentum von und unter der Verwaltung sein soll AWS oder um ein vom Kunden verwaltetes CMK zu verwenden.
7. (Optional) Unter S3 Datenzugriff, wählen Sie aus, ob Sie eine neue Rolle und Richtlinie für den Zugriff auf den Sequenzspeicher über Amazon S3 erstellen möchten.
8. (Optional) Für S3-Zugriffsprotokollierung, wählen Sie `Enabled` wenn Sie möchten, dass Amazon S3 Zugriffsprotokolldatensätze sammelt.

   Für Auf den Speicherort für die Protokollierung in S3 zugreifen, geben Sie einen Amazon S3-Standort zum Speichern der Protokolle an. Dieses Feld ist nur sichtbar, wenn Sie die S3-Zugriffsprotokollierung aktiviert haben.
9. Schlagworte (optional) — Stellen Sie bis zu 50 Tags für diesen Sequenzspeicher bereit. Diese Tags sind von den Read-Set-Tags getrennt, die während des Read-Sets gesetzt werden import/tag aktualisieren

Nachdem Sie den Shop erstellt haben, ist er bereit für Genomdateien importieren.

Einen Sequenzspeicher mit der CLI erstellen

Ersetzen Sie im folgenden Beispiel *`sequence store name`* mit dem Namen, den Sie für Ihren Sequenzspeicher gewählt haben.

```
aws omics create-sequence-store --name sequence store name --fallback-location "s3://
amzn-s3-demo-bucket"
```

Sie erhalten die folgende Antwort in JSON, die die ID-Nummer für Ihren neu erstellten Sequenzspeicher enthält.

```
{
    "id": "3936421177",
    "arn": "arn:aws:omics:us-west-2:111122223333:sequenceStore/3936421177",
    "name": "sequence_store_example_name",
    "creationTime": "2022-07-13T20:09:26.038Z"
    "fallbackLocation" : "s3://amzn-s3-demo-bucket"
}
```

Sie können auch alle mit Ihrem Konto verknüpften Sequenzspeicher anzeigen, indem Sie den list-sequence-stores Befehl, wie im Folgenden gezeigt.

```
aws omics list-sequence-stores
```

Sie erhalten die folgende Antwort.

```
{
    "sequenceStores": [
        {
            "arn": "arn:aws:omics:us-west-2:111122223333:sequenceStore/3936421177",
            "id": "3936421177",
            "name": "MySequenceStore",
            "creationTime": "2022-07-13T20:09:26.038Z",
            "updatedTime": "2024-09-13T04:11:31.242Z",
            "fallbackLocation" : "s3://amzn-s3-demo-bucket",
            "status": "Active"
        }
    ]
}
```

Sie können verwenden get-sequence-store um mehr über einen Sequenzspeicher anhand seiner ID zu erfahren, wie im folgenden Beispiel gezeigt:

```
aws omics get-sequence-store --id sequence store ID
```

Sie erhalten die folgende Antwort:

```
{
  "arn": "arn:aws:omics:us-west-2:123456789012:sequenceStore/sequencestoreID",
  "creationTime": "2024-01-12T04:45:29.857Z",
  "updatedTime": "2024-09-13T04:11:31.242Z",
  "description": null,
  "fallbackLocation": null,
  "id": "2015356892",
  "name": "MySequenceStore",
  "s3Access": {
      "s3AccessPointArn": "arn:aws:s3:us-
west-2:123456789012:accesspoint/592761533288-2015356892",
      "s3Uri": "s3://592761533288-2015356892-ajdpi90jdas90a79fh9a8ja98jdfa9jf98-
s3alias/592761533288/sequenceStore/2015356892/",
      "accessLogLocation": "s3://IAD-seq-store-log/2015356892/"
  },
  "sseConfig": {
```

```
      "keyArn": "arn:aws:kms:us-west-2:123456789012:key/eb2b30f5-635d-4b6d-b0f9-
d3889fe0e648",
      "type": "KMS"
  },
  "status": "Active",
  "statusMessage": null,
  "setTagsToSync": ["withdrawn","protocol"],
}
```

Nach der Erstellung können auch mehrere Speicherparameter aktualisiert werden. Dies kann über die Konsole oder die API erfolgen `updateSequenceStore` Betrieb.

Aktualisierung eines Sequenzspeichers

Gehen Sie folgendermaßen vor, um einen Sequenzspeicher zu aktualisieren:

1. Öffnen Sie die HealthOmics -Konsole.
2. Öffnen Sie bei Bedarf den linken Navigationsbereich (≡). Wählen Sie Sequenzspeicher.
3. Wählen Sie den Sequenzspeicher aus, der aktualisiert werden soll.
4. In der Einzelheiten Panel, wählen Bearbeiten.
5. Auf der Details bearbeiten Auf dieser Seite können Sie die folgenden Felder aktualisieren:

   - Name des Sequenzspeichers - Ein eindeutiger Name für diesen Shop.
   - Beschreibung - Eine Beschreibung dieses Sequenzspeichers.
   - Ersatzspeicherort in S3, geben Sie einen Amazon S3-Standort an. HealthOmics verwendet den Ersatzspeicherort zum Speichern aller Dateien, die während eines direkten Uploads keinen Lesesatz erstellen können.
   - Liest die Set-Tag-Schlüssel für die S3-Weitergabe Sie können bis zu fünf Read-Set-Schlüssel eingeben, um sie an Amazon S3 weiterzuleiten.
   - (Optional) Für S3-Zugriffsprotokollierung, wählen Sie `Enabled` wenn Sie möchten, dass Amazon S3 Zugriffsprotokolldatensätze sammelt.

     Für Auf den Speicherort für die Protokollierung in S3 zugreifen, geben Sie einen Amazon S3-Standort zum Speichern der Protokolle an. Dieses Feld ist nur sichtbar, wenn Sie die S3-Zugriffsprotokollierung aktiviert haben.
   - Schlagworte (optional) — Stellen Sie bis zu 50 Tags für diesen Sequenzspeicher bereit.

Aktualisierung der Read-Set-Tags für einen Sequenzspeicher

Gehen Sie folgendermaßen vor, um Read-Set-Tags oder andere Felder für einen Sequenzspeicher zu aktualisieren:

1. Öffnen Sie die HealthOmics -Konsole.
2. Öffnen Sie bei Bedarf den linken Navigationsbereich (≡). Wählen Sie Sequenzspeicher.
3. Wählen Sie den Sequenzspeicher aus, den Sie aktualisieren möchten.
4. Wählen Sie die Registerkarte Details.
5. Wählen Sie Bearbeiten aus.
6. Fügen Sie je nach Bedarf neue Read-Set-Tags hinzu oder löschen Sie vorhandene Tags.
7. Aktualisieren Sie nach Bedarf den Namen, die Beschreibung, den Fallback-Speicherort oder den S3-Datenzugriff.
8. Wählen Sie Änderungen speichern aus.

Genomdateien importieren

Gehen Sie folgendermaßen vor, um genomische Dateien in einen Sequenzspeicher zu importieren:

Um eine Genomikdatei zu importieren

1. Öffnen Sie die HealthOmics -Konsole.
2. Öffnen Sie bei Bedarf den linken Navigationsbereich (≡). Wählen Sie wählen Sequenzspeicher.
3. Auf der Sequenzspeicher Seite, wählen Sie den Sequenzspeicher aus, in den Sie Ihre Dateien importieren möchten.
4. Wählen Sie auf der Seite mit den einzelnen Sequenzspeichern Importieren Sie genomische Dateien.
5. Auf der Geben Sie die Importdetails an Seite, geben Sie die folgenden Informationen an

   - IAM-Rolle — Die IAM-Rolle, die auf die Genomdateien auf Amazon S3 zugreifen kann.
   - Referenzgenom - Das Referenzgenom für diese Genomdaten.
6. Auf der Geben Sie das Importmanifest an Seite, geben Sie die folgenden Informationen an Manifest-Datei. Die Manifestdatei ist eine JSON- oder YAML-Datei, die wichtige Informationen Ihrer Genomdaten beschreibt. Hinweise zur Manifestdatei finden Sie unter Importieren von Lesesätzen in eine HealthOmics Sequenzspeicher.

7. Klicken Sie Importauftrag erstellen.

Löschen HealthOmics Referenz- und Sequenzspeicher

Sowohl Referenz- als auch Sequenzspeicher können gelöscht werden. Sequenzspeicher können nur gelöscht werden, wenn sie keine Lesesätze enthalten, und Referenzspeicher können nur gelöscht werden, wenn sie keine Referenzen enthalten. Durch das Löschen eines Sequenz- oder Referenzspeichers werden auch alle mit diesem Speicher verknüpften Tags gelöscht.

Das folgende Beispiel zeigt, wie Sie einen Referenzspeicher löschen, indem Sie den AWS CLI. Wenn die Aktion erfolgreich ist, erhalten Sie keine Antwort. Ersetzen Sie im folgenden Beispiel *reference store ID* mit Ihrer Referenz-Store-ID.

```
aws omics delete-reference-store --id reference store ID
```

Das folgende Beispiel zeigt Ihnen, wie Sie einen Sequenzspeicher löschen. Sie erhalten keine Antwort, wenn die Aktion erfolgreich ist. Ersetzen Sie im folgenden Beispiel *sequence store ID* mit Ihrer Sequenzspeicher-ID.

```
aws omics delete-sequence-store --id sequence store ID
```

Sie können auch eine Referenz in einem Referenzspeicher löschen, wie im folgenden Beispiel gezeigt. Verweise können nur gelöscht werden, wenn sie nicht in einem Lesesatz, Variantenspeicher oder Annotationsspeicher verwendet werden. Ersetzen Sie im folgenden Beispiel *reference store ID* mit Ihrer Referenz-Shop-ID und ersetzen Sie *reference ID* mit der ID für die Referenz, die Sie löschen möchten.

```
aws omics delete-reference  --id reference ID --reference-store-id reference store ID
```

Importieren von Lesesätzen in eine HealthOmics Sequenzspeicher

Nachdem Sie Ihren Sequenzspeicher erstellt haben, erstellen Sie Importaufträge, um Lesesätze in den Datenspeicher hochzuladen. Sie können Ihre Dateien aus einem Amazon S3-Bucket hochladen, oder Sie können sie direkt hochladen, indem Sie die synchronen API-Operationen verwenden. Ihr Amazon S3-Bucket muss sich in derselben Region wie Ihr Sequenzspeicher befinden.

Sie können eine beliebige Kombination aus ausgerichteten und nicht ausgerichteten Lesesätzen in Ihren Sequenzspeicher hochladen. Wenn jedoch einer der Lesesätze in Ihrem Import ausgerichtet ist, müssen Sie ein Referenzgenom angeben.

Sie können die IAM-Zugriffsrichtlinie, mit der Sie den Referenzspeicher erstellt haben, wiederverwenden.

In den folgenden Themen werden die wichtigsten Schritte beschrieben, die Sie ausführen, um einen Lesesatz in Ihren Sequenzspeicher zu importieren und anschließend Informationen zu den importierten Daten abzurufen.

Themen

- Laden Sie Dateien auf Amazon S3 hoch
- Erstellen einer Manifestdatei
- Den Importjob starten
- Überwachen Sie den Importauftrag
- Suchen Sie die importierten Sequenzdateien
- Ruft Details zu einem Lesesatz ab
- Laden Sie die Read-Set-Datendateien herunter

Laden Sie Dateien auf Amazon S3 hoch

Das folgende Beispiel zeigt, wie Sie Dateien in Ihren Amazon S3-Bucket verschieben.

```
aws s3 cp s3://1000genomes/phase1/data/HG00100/alignment/
HG00100.chrom20.ILLUMINA.bwa.GBR.low_coverage.20101123.bam s3://your-bucket
aws s3 cp s3://1000genomes/phase3/data/HG00146/sequence_read/SRR233106_1.filt.fastq.gz
 s3://your-bucket
aws s3 cp s3://1000genomes/phase3/data/HG00146/sequence_read/SRR233106_2.filt.fastq.gz
 s3://your-bucket
aws s3 cp s3://1000genomes/data/HG00096/alignment/
HG00096.alt_bwamem_GRCh38DH.20150718.GBR.low_coverage.cram s3://your-bucket
aws s3 cp s3://gatk-test-data/wgs_ubam/NA12878_20k/NA12878_A.bam s3://your-bucket
```

Das Beispiel `BAM` und `CRAM` die in diesem Beispiel verwendeten Genomreferenzen erfordern, `Hg19` und `Hg38`. Weitere Informationen oder Zugriff auf diese Referenzen finden Sie unter Die umfassenden Genomreferenzen im Register für offene Daten am AWS.

Erstellen einer Manifestdatei

Sie müssen auch eine Manifestdatei in JSON erstellen, um den Importauftrag in zu modellieren `import.json` (siehe das folgende Beispiel). Wenn Sie einen Sequenzspeicher in der Konsole erstellen, müssen Sie den `sequenceStoreId` oder `roleARN`, also beginnt Ihre Manifestdatei mit dem `sources` Eingabe.

API manifest

Im folgenden Beispiel werden mithilfe der API drei Lesesätze importiert: eins `FASTQ`, eins `BAM`, und eins `CRAM`.

```
{
  "sequenceStoreId": "3936421177",
  "roleArn": "arn:aws:iam::555555555555:role/OmicsImport",
  "sources":
  [
      {
          "sourceFiles":
          {
              "source1": "s3://amzn-s3-demo-bucket/
HG00100.chrom20.ILLUMINA.bwa.GBR.low_coverage.20101123.bam"
          },
          "sourceFileType": "BAM",
          "subjectId": "mySubject",
          "sampleId": "mySample",
          "referenceArn": "arn:aws:omics:us-
west-2:555555555555:referenceStore/0123456789/reference/0000000001",
          "name": "HG00100",
          "description": "BAM for HG00100",
          "generatedFrom": "1000 Genomes"
      },
      {
          "sourceFiles":
          {
              "source1": "s3://amzn-s3-demo-bucket/SRR233106_1.filt.fastq.gz",
              "source2": "s3://amzn-s3-demo-bucket/SRR233106_2.filt.fastq.gz"
          },
          "sourceFileType": "FASTQ",
          "subjectId": "mySubject",
          "sampleId": "mySample",
          // NOTE: there is no reference arn required here
          "name": "HG00146",
```

```
          "description": "FASTQ for HG00146",
          "generatedFrom": "1000 Genomes"
      },
      {
          "sourceFiles":
          {
              "source1": "s3://amzn-s3-demo-bucket/
HG00096.alt_bwamem_GRCh38DH.20150718.GBR.low_coverage.cram"
          },
          "sourceFileType": "CRAM",
          "subjectId": "mySubject",
          "sampleId": "mySample",
          "referenceArn": "arn:aws:omics:us-
west-2:555555555555:referenceStore/0123456789/reference/0000000001",
          "name": "HG00096",
          "description": "CRAM for HG00096",
          "generatedFrom": "1000 Genomes"
      },
      {
          "sourceFiles":
          {
              "source1": "s3://amzn-s3-demo-bucket/NA12878_A.bam"
          },
          "sourceFileType": "UBAM",
          "subjectId": "mySubject",
          "sampleId": "mySample",
          // NOTE: there is no reference arn required here
          "name": "NA12878_A",
          "description": "uBAM for NA12878",
          "generatedFrom": "GATK Test Data"
      }
  ]
}
```

Console manifest

Dieser Beispielcode wird verwendet, um einen einzelnen Lesesatz mithilfe der Konsole zu importieren.

```
[
  {
      "sourceFiles":
      {
```

```
          "source1": "s3://amzn-s3-demo-bucket/
HG00100.chrom20.ILLUMINA.bwa.GBR.low_coverage.20101123.bam"
      },
      "sourceFileType": "BAM",
      "subjectId": "mySubject",
      "sampleId": "mySample",
      "name": "HG00100",
      "description": "BAM for HG00100",
      "generatedFrom": "1000 Genomes"
  },
  {
      "sourceFiles":
      {
          "source1": "s3://amzn-s3-demo-bucket/SRR233106_1.filt.fastq.gz",
          "source2": "s3://amzn-s3-demo-bucket/SRR233106_2.filt.fastq.gz"
      },
      "sourceFileType": "FASTQ",
      "subjectId": "mySubject",
      "sampleId": "mySample",
      "name": "HG00146",
      "description": "FASTQ for HG00146",
      "generatedFrom": "1000 Genomes"
  },
  {
      "sourceFiles":
      {
          "source1": "s3://your-bucket/
HG00096.alt_bwamem_GRCh38DH.20150718.GBR.low_coverage.cram"
      },
      "sourceFileType": "CRAM",
      "subjectId": "mySubject",
      "sampleId": "mySample",
      "name": "HG00096",
      "description": "CRAM for HG00096",
      "generatedFrom": "1000 Genomes"
  },
  {
      "sourceFiles":
      {
          "source1": "s3://amzn-s3-demo-bucket/NA12878_A.bam"
      },
      "sourceFileType": "UBAM",
      "subjectId": "mySubject",
      "sampleId": "mySample",
```

```
      "name": "NA12878_A",
      "description": "uBAM for NA12878",
      "generatedFrom": "GATK Test Data"
  }
]
```

Alternativ können Sie die Manifestdatei im YAML-Format hochladen.

Den Importjob starten

Verwenden Sie Folgendes, um den Importjob zu starten AWS CLI Befehl.

```
aws omics start-read-set-import-job --cli-input-json file://import.json
```

Sie erhalten die folgende Antwort, die auf eine erfolgreiche Erstellung des Jobs hinweist.

```
{
  "id": "3660451514",
  "sequenceStoreId": "3936421177",
  "roleArn": "arn:aws:iam::111122223333:role/OmicsImport",
  "status": "CREATED",
  "creationTime": "2022-07-13T22:14:59.309Z"
}
```

Überwachen Sie den Importauftrag

Nach dem Start des Importauftrags können Sie seinen Fortschritt mit dem folgenden Befehl überwachen. Ersetzen Sie im folgenden Beispiel *`sequence store id`* mit Ihrer Sequenzspeicher-ID und ersetzen Sie *`job import ID`* mit der Import-ID.

```
aws omics get-read-set-import-job --sequence-store-id sequence store id --id job import
 ID
```

Im Folgenden werden die Status aller Importaufträge angezeigt, die der angegebenen Sequenzspeicher-ID zugeordnet sind.

```
{
  "id": "1234567890",
  "sequenceStoreId": "1234567890",
  "roleArn": "arn:aws:iam::111122223333:role/OmicsImport",
```

```
  "status": "RUNNING",
  "statusMessage": "The job is currently in progress.",
  "creationTime": "2022-07-13T22:14:59.309Z",
  "sources": [
      {
          "sourceFiles":
          {
              "source1": "s3://amzn-s3-demo-bucket/
HG00100.chrom20.ILLUMINA.bwa.GBR.low_coverage.20101123.bam"
          },
          "sourceFileType": "BAM",
          "status": "IN_PROGRESS",
          "statusMessage": "The job is currently in progress."
          "subjectId": "mySubject",
          "sampleId": "mySample",
          "referenceArn": "arn:aws:omics:us-
west-2:111122223333:referenceStore/3242349265/reference/8625408453",
          "name": "HG00100",
          "description": "BAM for HG00100",
          "generatedFrom": "1000 Genomes",
          "readSetID": "1234567890"
      },
      {
          "sourceFiles":
          {
              "source1": "s3://amzn-s3-demo-bucket/SRR233106_1.filt.fastq.gz",
              "source2": "s3://amzn-s3-demo-bucket/SRR233106_2.filt.fastq.gz"
          },
          "sourceFileType": "FASTQ",
          "status": "IN_PROGRESS",
          "statusMessage": "The job is currently in progress."
          "subjectId": "mySubject",
          "sampleId": "mySample",
          "name": "HG00146",
          "description": "FASTQ for HG00146",
          "generatedFrom": "1000 Genomes",
          "readSetID": "1234567890"
      },
      {
          "sourceFiles":
          {
              "source1": "s3://amzn-s3-demo-bucket/
HG00096.alt_bwamem_GRCh38DH.20150718.GBR.low_coverage.cram"
          },
```

```
          "sourceFileType": "CRAM",
          "status": "IN_PROGRESS",
          "statusMessage": "The job is currently in progress."
          "subjectId": "mySubject",
          "sampleId": "mySample",
          "referenceArn": "arn:aws:omics:us-
west-2:111122223333:referenceStore/3242349265/reference/1234568870",
          "name": "HG00096",
          "description": "CRAM for HG00096",
          "generatedFrom": "1000 Genomes",
          "readSetID": "1234567890"
      },
      {
          "sourceFiles":
          {
              "source1": "s3://amzn-s3-demo-bucket/NA12878_A.bam"
          },
          "sourceFileType": "UBAM",
          "status": "IN_PROGRESS",
          "statusMessage": "The job is currently in progress."
          "subjectId": "mySubject",
          "sampleId": "mySample",
          "name": "NA12878_A",
          "description": "uBAM for NA12878",
          "generatedFrom": "GATK Test Data",
          "readSetID": "1234567890"
      }
  ]
}
```

Suchen Sie die importierten Sequenzdateien

Nach Abschluss des Jobs können Sie das verwenden Lesesätze auflisten API-Vorgang zum Auffinden der importierten Sequenzdateien. Ersetzen Sie im folgenden Beispiel *sequence store id* mit Ihrer Sequenzspeicher-ID.

```
aws omics list-read-sets --sequence-store-id sequence store id
```

Sie erhalten die folgende Antwort.

```
{
  "readSets": [
```

```
      {
          "id": "0000000001",
          "arn": "arn:aws:omics:us-west-2:111122223333:sequenceStore/01234567890/
readSet/0000000001",
          "sequenceStoreId": "1234567890",
          "subjectId": "mySubject",
          "sampleId": "mySample",
          "status": "ACTIVE",
          "name": "HG00100",
          "description": "BAM for HG00100",
          "referenceArn": "arn:aws:omics:us-
west-2:111122223333:referenceStore/01234567890/reference/0000000001",
          "fileType": "BAM",
          "sequenceInformation": {
              "totalReadCount": 9194,
              "totalBaseCount": 928594,
              "generatedFrom": "1000 Genomes",
              "alignment": "ALIGNED"
          },
          "creationTime": "2022-07-13T23:25:20Z"
          "creationType": "IMPORT",
          "etag": {
              "algorithm": "BAM_MD5up",
              "source1": "d1d65429212d61d115bb19f510d4bd02"
          }
      },
      {
          "id": "0000000002",
          "arn": "arn:aws:omics:us-west-2:111122223333:sequenceStore/0123456789/
readSet/0000000002",
          "sequenceStoreId": "0123456789",
          "subjectId": "mySubject",
          "sampleId": "mySample",
          "status": "ACTIVE",
          "name": "HG00146",
          "description": "FASTQ for HG00146",
          "fileType": "FASTQ",
          "sequenceInformation": {
              "totalReadCount": 8000000,
              "totalBaseCount": 1184000000,
              "generatedFrom": "1000 Genomes",
              "alignment": "UNALIGNED"
          },
          "creationTime": "2022-07-13T23:26:43Z"
```

```
            "creationType": "IMPORT",
            "etag": {
                "algorithm": "FASTQ_MD5up",
                "source1": "ca78f685c26e7cc2bf3e28e3ec4d49cd"
            }
        },
        {
            "id": "0000000003",
            "arn": "arn:aws:omics:us-west-2:111122223333:sequenceStore/0123456789/
readSet/0000000003",
            "sequenceStoreId": "0123456789",
            "subjectId": "mySubject",
            "sampleId": "mySample",
            "status": "ACTIVE",
            "name": "HG00096",
            "description": "CRAM for HG00096",
            "referenceArn": "arn:aws:omics:us-
west-2:111122223333:referenceStore/0123456789/reference/0000000001",
            "fileType": "CRAM",
            "sequenceInformation": {
                "totalReadCount": 85466534,
                "totalBaseCount": 24000004881,
                "generatedFrom": "1000 Genomes",
                "alignment": "ALIGNED"
            },
            "creationTime": "2022-07-13T23:30:41Z"
            "creationType": "IMPORT",
            "etag": {
                "algorithm": "CRAM_MD5up",
                "source1": "66817940f3025a760e6da4652f3e927e"
            }
        },
        {
            "id": "0000000004",
            "arn": "arn:aws:omics:us-west-2:111122223333:sequenceStore/0123456789/
readSet/0000000004",
            "sequenceStoreId": "0123456789",
            "subjectId": "mySubject",
            "sampleId": "mySample",
            "status": "ACTIVE",
            "name": "NA12878_A",
            "description": "uBAM for NA12878",
            "fileType": "UBAM",
            "sequenceInformation": {
```

```
            "totalReadCount": 20000,
            "totalBaseCount": 5000000,
            "generatedFrom": "GATK Test Data",
            "alignment": "ALIGNED"
        },
        "creationTime": "2022-07-13T23:30:41Z"
        "creationType": "IMPORT",
        "etag": {
            "algorithm": "BAM_MD5up",
            "source1": "640eb686263e9f63bcda12c35b84f5c7"
        }
    }
  ]
}
```

Ruft Details zu einem Lesesatz ab

Um weitere Details zu einem Lesesatz anzuzeigen, verwenden Sie das GetReadSetMetadata API-Betrieb. Ersetzen Sie im folgenden Beispiel *`sequence store id`* mit Ihrer Sequenzspeicher-ID und ersetzen Sie *`read set id`* mit Ihrer Lesesatz-ID.

```
aws omics get-read-set-metadata --sequence-store-id sequence store id --id read set id
```

Sie erhalten die folgende Antwort.

```
{
"arn": "arn:aws:omics:us-west-2:123456789012:sequenceStore/2015356892/
readSet/9515444019",
"creationTime": "2024-01-12T04:50:33.548Z",
"creationType": "IMPORT",
"creationJobId": "33222111",
"description": null,
"etag": {
  "algorithm": "FASTQ_MD5up",
  "source1": "00d0885ba3eeb211c8c84520d3fa26ec",
  "source2": "00d0885ba3eeb211c8c84520d3fa26ec"
},
"fileType": "FASTQ",
"files": {
  "index": null,
  "source1": {
```

```
    "contentLength": 10818,
    "partSize": 104857600,
    "s3Access": {
      "s3Uri": "s3://accountID-sequence store ID-ajdpi90jdas90a79fh9a8ja98jdfa9jf98-
s3alias/592761533288/sequenceStore/2015356892/readSet/9515444019/
import_source1.fastq.gz"
},
    "totalParts": 1
  },
  "source2": {
    "contentLength": 10818,
    "partSize": 104857600,
    "s3Access": {
      "s3Uri": "s3://accountID-sequence store ID-ajdpi90jdas90a79fh9a8ja98jdfa9jf98-
s3alias/592761533288/sequenceStore/2015356892/readSet/9515444019/
import_source1.fastq.gz"
    },
    "totalParts": 1
  }
},
"id": "9515444019",
"name": "paired-fastq-import",
"sampleId": "sampleId-paired-fastq-import",
"sequenceInformation": {
  "alignment": "UNALIGNED",
  "generatedFrom": null,
  "totalBaseCount": 30000,
  "totalReadCount": 200
},
"sequenceStoreId": "2015356892",
"status": "ACTIVE",
"statusMessage": null,
"subjectId": "subjectId-paired-fastq-import"
}
```


Laden Sie die Read-Set-Datendateien herunter

Sie können mit Amazon S3 auf die Objekte für einen aktiven Lesesatz zugreifen GetObject API-Betrieb. Der URI für das Objekt wird im zurückgegeben GetReadSetMetadata API-Antwort. Weitere Informationen finden Sie unter Zugreifen HealthOmics Sätze mit Amazon S3-URIs lesen.

Verwenden Sie alternativ die HealthOmics GetReadSet API-Betrieb. Sie können verwenden GetReadSet zum parallelen Herunterladen, indem Sie einzelne Teile herunterladen. Diese Teile

ähneln Amazon S3-Teilen. Das Folgende ist ein Beispiel dafür, wie Teil 1 aus einem Readset heruntergeladen wird. Im folgenden Beispiel ersetzen Sie *sequence store id* mit Ihrer Sequenzspeicher-ID und ersetzen Sie *read set id* mit Ihrer Lesesatz-ID.

```
aws omics get-read-set --sequence-store-id sequence store id --id read set id  --part-
number 1 outfile.bam
```

Sie können auch die verwenden HealthOmics Transfer Manager zum Herunterladen von Dateien für einen HealthOmics Referenz- oder Lesesatz. Sie können das herunterladen HealthOmics Transfer Manager hier. Weitere Informationen zur Verwendung und Einrichtung des Transfer Managers finden Sie hier GitHub Repositorium.

Direkter Upload auf eine HealthOmics Sequenzspeicher

Wir empfehlen Ihnen, das zu verwenden HealthOmics Transfer Manager, um Dateien zu Ihrem Sequenzspeicher hinzuzufügen. Weitere Informationen zur Verwendung von Transfer Manager finden Sie hier GitHub Repositorium. Sie können Ihre Lesesätze auch direkt über die API-Operationen zum direkten Hochladen in einen Sequenzspeicher hochladen.

Direkt hochgeladene Lesesätze existieren zuerst in `PROCESSING_UPLOAD` Bundesstaat. Das bedeutet, dass die Dateiteile gerade hochgeladen werden und Sie auf die Metadaten des Lesesatzes zugreifen können. Nachdem die Teile hochgeladen und die Prüfsummen validiert wurden, wird der Lesesatz `ACTIVE` und verhält sich genauso wie ein importierter Lesesatz.

Schlägt der direkte Upload fehl, wird der Status des Lesesatzes wie folgt angezeigt `UPLOAD_FAILED`. Sie können einen Amazon S3-Bucket als Ausweichort für Dateien konfigurieren, die nicht hochgeladen werden können. Ersatzspeicherorte sind für Sequenzspeicher verfügbar, die nach dem 15. Mai 2023 erstellt wurden.

Themen

- Direkter Upload in einen Sequenzspeicher mit dem AWS CLI
- Konfigurieren Sie einen Ausweichort

Direkter Upload in einen Sequenzspeicher mit dem AWS CLI

Starten Sie zunächst einen mehrteiligen Upload. Sie können dies tun, indem Sie den AWS CLI, wie im folgenden Beispiel gezeigt.

Zum direkten Upload mit AWS CLI commands

1. Erstellen Sie die Teile, indem Sie Ihre Daten trennen, wie im folgenden Beispiel gezeigt.

```
split -b 100MiB SRR233106_1.filt.fastq.gz source1_part_
```

2. Nachdem Ihre Quelldateien in Teilen vorliegen, erstellen Sie einen mehrteiligen Lesesatz-Upload, wie im folgenden Beispiel gezeigt. Ersetzen *sequence store ID* und die anderen Parameter mit Ihrer Sequenzspeicher-ID und anderen Werten.

```
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"
```

   Du bekommst die `uploadID` und andere Metadaten in der Antwort. Benutze das `uploadID` für den nächsten Schritt des Upload-Vorgangs.

```
{
"sequenceStoreId": "1504776472",
"uploadId": "7640892890",
"sourceFileType": "FASTQ",
"subjectId": "mySubject",
"sampleId": "mySample",
"generatedFrom": "1000 Genomes",
"name": "HG00146",
"description": "FASTQ for HG00146",
"creationTime": "2023-11-20T23:40:47.437522+00:00"
}
```

3. Fügen Sie Ihre Lesesätze zum Upload hinzu. Wenn Ihre Datei klein genug ist, müssen Sie diesen Schritt nur einmal ausführen. Bei größeren Dateien führen Sie diesen Schritt für jeden Teil Ihrer Datei durch. Wenn Sie ein neues Teil unter Verwendung einer zuvor verwendeten Artikelnummer hochladen, wird das zuvor hochgeladene Teil überschrieben.

   Ersetzen Sie im folgenden Beispiel *sequence store ID*, *upload ID*, und die anderen Parameter mit Ihren Werten.

```
aws omics upload-read-set-part \
--sequence-store-id sequence store ID \
--upload-id upload ID \
--part-source SOURCE1 \
--part-number part number \
--payload  source1/source1_part_aa.fastq.gz
```

Die Antwort ist eine ID, mit der Sie überprüfen können, ob die hochgeladene Datei mit der von Ihnen beabsichtigten Datei übereinstimmt.

```
{
"checksum": "984979b9928ae8d8622286c4a9cd8e99d964a22d59ed0f5722e1733eb280e635"
}
```

4. Fahren Sie bei Bedarf mit dem Hochladen der Teile Ihrer Datei fort. Um zu überprüfen, ob Ihre Lesesätze hochgeladen wurden, verwenden Sie das list-read-set-upload-parts API-Betrieb, wie im Folgenden gezeigt. Ersetzen Sie im folgenden Beispiel *sequence store ID* , *upload ID*, und die *part source* mit deinem eigenen Input.

```
aws omics list-read-set-upload-parts \
 --sequence-store-id sequence store ID \
 --upload-id upload ID \
 --part-source SOURCE1
```

Die Antwort gibt die Anzahl der Lesesätze, die Größe und den Zeitstempel der letzten Aktualisierung zurück.

```
{
"parts": [
    {
        "partNumber": 1,
        "partSize": 104857600,
        "partSource": "SOURCE1",
        "checksum": "MVMQk+vB9C3Ge8ADHkbKq752n3BCUzyl41qEkqlOD5M=",
        "creationTime": "2023-11-20T23:58:03.500823+00:00",
        "lastUpdatedTime": "2023-11-20T23:58:03.500831+00:00"
    },
    {
        "partNumber": 2,
        "partSize": 104857600,
```

```
        "partSource": "SOURCE1",
        "checksum": "keZzVzJNChAqgOdZMvOmjBwrOPM0enPj1UAfs0nvRto=",
        "creationTime": "2023-11-21T00:02:03.813013+00:00",
        "lastUpdatedTime": "2023-11-21T00:02:03.813025+00:00"
    },
    {
        "partNumber": 3,
        "partSize": 100339539,
        "partSource": "SOURCE1",
        "checksum": "TBkNfMsaeDpXzEf3ldlbi0ipFDPaohKHyZ+LF1J4CHk=",
        "creationTime": "2023-11-21T00:09:11.705198+00:00",
        "lastUpdatedTime": "2023-11-21T00:09:11.705208+00:00"
    }
]
}
```

5. Um alle aktiven mehrteiligen Leseset-Uploads anzuzeigen, verwenden Sie list-multipart-read-set-uploads, wie im Folgenden gezeigt. Ersetzen *sequence store ID* mit der ID für Ihren eigenen Sequenzspeicher.

```
aws omics list-multipart-read-set-uploads --sequence-store-id
             sequence store ID
```

Diese API gibt nur mehrteilige Leseset-Uploads zurück, die gerade ausgeführt werden. Nach der Aufnahme der Lesesätze `ACTIVE`, oder wenn der Upload fehlgeschlagen ist, wird der Upload nicht in der Antwort auf die list-multipart-read-set-uploads API. Um aktive Lesesätze anzuzeigen, verwenden Sie die Lesesätze auflisten API. Eine Beispielantwort für list-multipart-read-set-uploads wird im Folgenden gezeigt.

```
{
"uploads": [
    {
        "sequenceStoreId": "1234567890",
        "uploadId": "8749584421",
        "sourceFileType": "FASTQ",
        "subjectId": "mySubject",
        "sampleId": "mySample",
        "generatedFrom": "1000 Genomes",
        "name": "HG00146",
        "description": "FASTQ for HG00146",
        "creationTime": "2023-11-29T19:22:51.349298+00:00"
```

```
    },
    {
        "sequenceStoreId": "1234567890",
        "uploadId": "5290538638",
        "sourceFileType": "BAM",
        "subjectId": "mySubject",
        "sampleId": "mySample",
        "generatedFrom": "1000 Genomes",
        "referenceArn": "arn:aws:omics:us-
west-2:123456789012:referenceStore/8168613728/reference/2190697383",
        "name": "HG00146",
        "description": "BAM for HG00146",
        "creationTime": "2023-11-29T19:23:33.116516+00:00"
    },
    {
        "sequenceStoreId": "1234567890",
        "uploadId": "4174220862",
        "sourceFileType": "BAM",
        "subjectId": "mySubject",
        "sampleId": "mySample",
        "generatedFrom": "1000 Genomes",
        "referenceArn": "arn:aws:omics:us-
west-2:123456789012:referenceStore/8168613728/reference/2190697383",
        "name": "HG00147",
        "description": "BAM for HG00147",
        "creationTime": "2023-11-29T19:23:47.007866+00:00"
    }
]
}
```

6. Nachdem Sie alle Teile Ihrer Datei hochgeladen haben, verwenden Sie Complete-Multipart-Read-Set-Upload um den Upload-Vorgang abzuschließen, wie im folgenden Beispiel gezeigt. Ersetzen *sequence store ID*, *upload ID*, und der Parameter für Teile mit Ihren eigenen Werten.

```
aws omics complete-multipart-read-set-upload \
--sequence-store-id sequence store ID \
--upload-id upload ID \
--parts '[{"checksum":"gaCBQMe+rpCFZxLpoP6gydBoXaKKDA/
Vobh5zBDb4W4=","partNumber":1,"partSource":"SOURCE1"}]'
```

Die Antwort für Complete-Multipart-Read-Set-Upload ist die Lesesatz-IDs für Ihre importierten Lesesätze.

```
{
"readSetId": "0000000001"
}
```

7. Um den Upload zu stoppen, verwenden Sie abort-multipart-read-set-upload mit der Upload-ID, um den Upload-Vorgang zu beenden. Ersetzen *sequence store ID* und *upload ID* mit Ihren eigenen Parameterwerten.

```
aws omics abort-multipart-read-set-upload \
--sequence-store-id sequence store ID \
--upload-id upload ID
```

8. Nachdem der Upload abgeschlossen ist, rufen Sie Ihre Daten aus dem Lesesatz ab, indem Sie get-read-set, wie im Folgenden gezeigt. Wenn der Upload noch in Bearbeitung ist, get-read-set gibt begrenzte Metadaten zurück, und die generierten Indexdateien sind nicht verfügbar. Ersetzen *sequence store ID* und die anderen Parameter mit deiner eigenen Eingabe.

```
aws omics get-read-set
 --sequence-store-id sequence store ID \
 --id read set ID \
 --file SOURCE1 \
 --part-number 1 myfile.fastq.gz
```

9. Um die Metadaten, einschließlich des Status Ihres Uploads, zu überprüfen, verwenden Sie get-read-set-metadata API-Betrieb.

```
aws omics get-read-set-metadata --sequence-store-id sequence store ID --id read set
 ID
```

Die Antwort enthält Metadatendetails wie den Dateityp, den Referenz-ARN, die Anzahl der Dateien und die Länge der Sequenzen. Sie beinhaltet auch den Status. Mögliche Status sind `PROCESSING_UPLOAD`, `ACTIVE`, und `UPLOAD_FAILED`.

```
{
"id": "0000000001",
```

```
"arn": "arn:aws:omics:us-west-2:555555555555:sequenceStore/0123456789/
readSet/0000000001",
"sequenceStoreId": "0123456789",
"subjectId": "mySubject",
"sampleId": "mySample",
"status": "PROCESSING_UPLOAD",
"name": "HG00146",
"description": "FASTQ for HG00146",
"fileType": "FASTQ",
"creationTime": "2022-07-13T23:25:20Z",
"files": {
    "source1": {
        "totalParts": 5,
        "partSize": 123456789012,
        "contentLength": 6836725,

    },
    "source2": {
        "totalParts": 5,
        "partSize": 123456789056,
        "contentLength": 6836726
    }
},
'creationType": "UPLOAD"
}
```

Konfigurieren Sie einen Ausweichort

Wenn Sie einen Sequenzspeicher erstellen oder aktualisieren, können Sie einen Amazon S3-Bucket als Fallback-Speicherort für Dateien konfigurieren, die nicht hochgeladen werden können. Die Dateiteile für diese Lesesätze werden an den Fallback-Speicherort übertragen. Ersatzspeicherorte sind für Sequenzspeicher verfügbar, die nach dem 15. Mai 2023 erstellt wurden.

Erstellen Sie eine Amazon S3-Bucket-Richtlinie, die Sie gewähren möchten HealthOmics Schreibzugriff auf den Amazon S3-Fallback-Speicherort, wie im folgenden Beispiel gezeigt:

```
{
    "Effect": "Allow",
    "Principal": {
        "Service": "omics.amazonaws.com"
    },
    "Action": "s3:PutObject",
```

```
    "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
}
```

Wenn der Amazon S3-Bucket für Fallback- oder Zugriffsprotokolle einen vom Kunden verwalteten Schlüssel verwendet, fügen Sie der Schlüsselrichtlinie die folgenden Berechtigungen hinzu:

```
 {
    "Sid": "Allow use of key",
    "Effect": "Allow",
    "Principal": {
        "Service": "omics.amazonaws.com"
    },
    "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey*"
    ],
    "Resource": "*"
}
```

Exportieren HealthOmics Sätze in einen Amazon S3-Bucket lesen

Sie können Lesesätze als Batch-Exportauftrag in einen Amazon S3-Bucket exportieren. Erstellen Sie dazu zunächst eine IAM-Richtlinie, die Schreibzugriff auf den Bucket hat, ähnlich dem folgenden Beispiel für eine IAM-Richtlinie.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:GetBucketLocation"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket1",
        "arn:aws:s3:::amzn-s3-demo-bucket1/*"
      ]
    }
```

```
  ]
}
```

JSON

```
{
"Version":"2012-10-17",
"Statement": [
  {
      "Effect": "Allow",
      "Principal": {
          "Service": [
              "omics.amazonaws.com"
          ]
      },
      "Action": "sts:AssumeRole"
  }
]
}
```

Nachdem die IAM-Richtlinie in Kraft ist, beginnen Sie mit Ihrem Read-Set-Exportauftrag. Das folgende Beispiel zeigt Ihnen, wie Sie das tun können, indem Sie start-read-set-export-job API-Betrieb. Ersetzen Sie im folgenden Beispiel alle Parameter, wie *sequence store ID*, *destination* , *role ARN*, und *sources*, mit deiner Eingabe.

```
aws omics start-read-set-export-job
--sequence-store-id sequence store id \
--destination valid s3 uri \
--role-arn role ARN \
--sources readSetId=read set id_1 readSetId=read set id_2
```

Sie erhalten die folgende Antwort mit Informationen zum ursprünglichen Sequenzspeicher und zum Amazon S3-Ziel-Bucket.

```
{
"id": <job-id>,
"sequenceStoreId": <sequence-store-id>,
"destination": <destination-s3-uri>,
"status": "SUBMITTED",
```

```
"creationTime": "2022-10-22T01:33:38.079000+00:00"
}
```

Nachdem der Job gestartet wurde, können Sie seinen Status ermitteln, indem Sie das get-read-set-export-job API-Betrieb, wie im Folgenden gezeigt. Ersetzen Sie das *sequence store ID* und *job ID* mit Ihrer Sequenzspeicher-ID bzw. Job-ID.

```
aws omics get-read-set-export-job --id job-id --sequence-store-id sequence store ID
```

Sie können alle Exportaufträge anzeigen, die für einen Sequenzspeicher initialisiert wurden, indem Sie list-read-set-export-jobs API-Betrieb, wie im Folgenden gezeigt. Ersetzen Sie das *sequence store ID* mit Ihrer Sequenzspeicher-ID.

```
aws omics list-read-set-export-jobs --sequence-store-id sequence store ID.
```

```
{
"exportJobs": [
  {
      "id": <job-id>,
      "sequenceStoreId": <sequence-store-id>,
      "destination": <destination-s3-uri>,
      "status": "COMPLETED",
      "creationTime": "2022-10-22T01:33:38.079000+00:00",
      "completionTime": "2022-10-22T01:34:28.941000+00:00"
  }
]
}
```

Sie können Ihre Lesesätze nicht nur exportieren, sondern sie auch mithilfe der Amazon S3-Zugriffs-URIs teilen. Weitere Informationen hierzu finden Sie unter Zugreifen HealthOmics Sätze mit Amazon S3-URIs lesen.

Zugreifen HealthOmics Sätze mit Amazon S3-URIs lesen

Sie können Amazon S3-URI-Pfade verwenden, um auf Ihre aktiven Sequenzspeicher-Lesesätze zuzugreifen.

Mit dem Amazon S3-URI-Pfad können Sie Amazon S3-Operationen verwenden, um Ihre Lesesätze aufzulisten, zu teilen und herunterzuladen. Der Zugriff über die S3-APIs beschleunigt die Zusammenarbeit und die Toolintegration, da viele Branchen-Tools bereits für das Lesen von S3

entwickelt wurden. Darüber hinaus können Sie den Zugriff auf die S3-APIs mit anderen Konten teilen und regionsübergreifenden Lesezugriff auf Daten gewähren.

HealthOmics unterstützt den Amazon S3-URI-Zugriff auf archivierte Lesesätze nicht. Wenn Sie einen Lesesatz aktivieren, wird er jedes Mal auf denselben URI-Pfad wiederhergestellt.

Mit Daten, die geladen wurden HealthOmics speichert, da die Amazon S3-URI auf Amazon S3-Zugriffspunkten basiert, können Sie eine direkte Integration mit branchenüblichen Tools durchführen, die Amazon S3-URIs lesen, wie z. B. die folgenden:

- Visuelle Analyseanwendungen wie Integrative Genomics Viewer (IGV) oder UCSC Genome Browser.
- Allgemeine Workflows mit Amazon S3-Erweiterungen wie CWL, WDL und Nextflow.
- Jedes Tool, das Amazon S3-URIs von Access Points authentifizieren und von diesen lesen oder vorsignierte Amazon S3-URIs lesen kann.
- Amazon S3-Dienstprogramme wie Mountpoint oder CloudFront.

Amazon S3 Mountpoint ermöglicht es Ihnen, einen Amazon S3-Bucket als lokales Dateisystem zu verwenden. Weitere Informationen zu Mountpoint und zur Installation zur Verwendung finden Sie unter Mountpoint für Amazon S3.

Amazonas CloudFront ist ein Content Delivery Network (CDN) -Service, der auf hohe Leistung, Sicherheit und Komfort für Entwickler ausgelegt ist. Um mehr über die Nutzung von Amazon zu erfahren CloudFront, siehe der Amazonas CloudFront Dokumentation. Zum Einrichten CloudFront mit einem Sequenzspeicher wenden Sie sich an den AWS HealthOmics Mannschaft.

Das Root-Konto des Datenbesitzers ist für die Aktionen aktiviert S3:GetObject, S3:GetObjectTagging, und S3:List Bucket für das Sequenzspeicher-Präfix. Damit ein Benutzer im Konto auf die Daten zugreifen kann, erstellen Sie eine IAM-Richtlinie und hängen sie dem Benutzer oder der Rolle an. Eine Beispielrichtlinie finden Sie unter Berechtigungen für den Datenzugriff mit Amazon S3 URIs.

Sie können die folgenden Amazon S3-API-Operationen für die aktiven Lesesätze verwenden, um Ihre Daten aufzulisten und abzurufen. Sie können über Amazon S3-URIs auf archivierte Lesesätze zugreifen, nachdem sie aktiviert wurden.

- GetObject— Ruft ein Objekt aus Amazon S3 ab.
- HeadObject— Die HEAD-Operation ruft Metadaten von einem Objekt ab, ohne das Objekt selbst zurückzugeben. Diese Operation ist nützlich, wenn Sie nur die Metadaten eines Objekts benötigen.

- ListObjects und ListObject v2— Gibt einige oder alle (bis zu 1.000) der Objekte in einem Bucket zurück.
- CopyObject— Erstellt eine Kopie eines Objekts, das bereits in Amazon S3 gespeichert ist. HealthOmics unterstützt das Kopieren auf einen Amazon S3-Zugangspunkt, aber nicht das Schreiben auf einen Access Point.

HealthOmics Sequenzspeicher behalten die semantische Identität von Dateien mithilfe von ETags bei. Während des gesamten Lebenszyklus einer Datei kann sich das Amazon S3-ETag, das auf einer bitweisen Identität basiert, ändern, jedoch HealthOmics ETag bleibt gleich. Weitere Informationen hierzu finden Sie unter HealthOmics ETags und Datenherkunft.

Themen

- Amazon S3-URI-Struktur in HealthOmics Lager
- Verwenden von Hosted oder Local IGV für den Zugriff auf Lesesätze
- Verwenden Sie Samtools oder HTSLib in HealthOmics
- Mit Mountpoint HealthOmics
- Verwenden CloudFront mit HealthOmics

Amazon S3-URI-Struktur in HealthOmics Lager

Alle Dateien mit Amazon S3-URIs haben `omics:subjectId` und `omics:sampleId` Ressourcen-Tags. Sie können diese Tags verwenden, um den Zugriff gemeinsam zu nutzen, indem Sie IAM-Richtlinien nach einem Muster wie dem folgenden verwenden `"s3:ExistingObjectTag/omics:subjectId": "pattern desired"`.

Die Dateistruktur ist wie folgt:

`.../`*`account_id`*`/sequenceStore/`*`seq_store_id`*`/readSet/`*`read_set_id`*`/`*`files`*`.`

Bei Dateien, die aus Amazon S3 in Sequenzspeicher importiert wurden, versucht der Sequenzspeicher, den ursprünglichen Quellnamen beizubehalten. Wenn die Namen in Konflikt geraten, hängt das System Read-Set-Informationen an, um sicherzustellen, dass die Dateinamen eindeutig sind. Wenn beispielsweise bei Fastq-Lesets beide Dateinamen identisch sind, um die Namen eindeutig zu machen, `sourceX` wird vor .fastq.gz oder.fq.gz eingefügt. Für einen direkten Upload folgen die Dateinamen den folgenden Mustern:

- Für FASTQ— *`read_set_name_sourcex`*.fastq.gz

- Für uBAM/BAM/CREME — *read_set_name.file extension* mit Erweiterungen von `.bam` oder `.cram`. Ein Beispiel ist `NA193948.bam`.

Für Lesesätze, bei denen es sich um BAM- oder CRAM handelt, werden Indexdateien während des Aufnahmevorgangs automatisch generiert. Für die generierten Indexdateien wird die richtige Indexerweiterung am Ende des Dateinamens angewendet. Es hat das Muster *<name of the Source the index is on>.<file index extension>.* Die Indexerweiterungen sind `.bai` oder `.crai`.

Verwenden von Hosted oder Local IGV für den Zugriff auf Lesesätze

IGV ist ein Genombrowser, der zur Analyse von BAM- und CRAM-Dateien verwendet wird. Er benötigt sowohl die Datei als auch den Index, da nur ein Teil des Genoms gleichzeitig angezeigt wird. IGV kann heruntergeladen und lokal verwendet werden, und es gibt Anleitungen zum Erstellen eines von AWS gehosteten IGV. Die öffentliche Webversion wird nicht unterstützt, da sie CORS benötigt.

Local IGV ist auf Local angewiesen AWS Konfiguration für den Zugriff auf Dateien. Stellen Sie sicher, dass an die in dieser Konfiguration verwendete Rolle eine Richtlinie angehängt ist, die kms:Decrypt und s3: aktiviertGetObject Berechtigungen für den s3-URI der Lesesätze, auf die zugegriffen wird. Danach können Sie in IGV „Datei > Von URL laden“ verwenden und die URI für die Quelle und den Index einfügen. Alternativ können vorsignierte URLs auf dieselbe Weise generiert und verwendet werden, wodurch die AWS-Konfiguration umgangen wird. Beachten Sie, dass CORS beim Amazon S3-URI-Zugriff nicht unterstützt wird, sodass Anfragen, die auf CORS basieren, nicht unterstützt werden.

Das Beispiel AWS Hosted IGV verlässt sich auf AWS Cognito, um die richtigen Konfigurationen und Berechtigungen innerhalb der Umgebung zu erstellen. Stellen Sie sicher, dass eine Richtlinie erstellt wird, die kms:decrypt und s3: aktiviertGetObject Berechtigungen für die Amazon S3-URI der Lesesätze, auf die zugegriffen wird, und fügen Sie diese Richtlinie der Rolle hinzu, die dem Cognito-Benutzerpool zugewiesen ist. Danach können Sie in IGV „Datei > Von URL laden“ verwenden und die URI für die Quelle und den Index eingeben. Alternativ können vorsignierte URLs auf dieselbe Weise generiert und verwendet werden, wodurch die AWS-Konfiguration umgangen wird.

Beachten Sie, dass der Sequenzspeicher nicht auf der Registerkarte „Amazon“ angezeigt wird, da dort nur Buckets angezeigt werden, die Ihnen in der Region gehören, in der AWS Profil ist konfiguriert.

Verwenden Sie Samtools oder HTSLib in HealthOmics

HTSLib ist die Kernbibliothek, die von mehreren Tools wie Samtools, RSAMTools, gemeinsam genutzt wird PySam, und andere. Verwenden Sie HTSLib Version 1.20 oder höher, um eine nahtlose Unterstützung für Amazon S3 Access Points zu erhalten. Für ältere Versionen der HTSLib-Bibliothek können Sie die folgenden Problemumgehungen verwenden:

- Legen Sie die Umgebungsvariable für den HTS Amazon S3-Host fest mit: `export HTS_S3_HOST="s3.region.amazonaws.com"`.
- Generieren Sie eine vorsignierte URL für die Dateien, die Sie verwenden möchten. Wenn ein BAM oder CRAM verwendet wird, stellen Sie sicher, dass eine vorsignierte URL sowohl für die Datei als auch für den Index generiert wird. Danach können beide Dateien mit den Bibliotheken verwendet werden.
- Verwenden Sie Mountpoint, um den Sequenzspeicher oder das Read-Set-Präfix in derselben Umgebung zu mounten, in der Sie HTSlib-Bibliotheken verwenden. Von hier aus kann über lokale Dateipfade auf die Dateien zugegriffen werden.

Mit Mountpoint HealthOmics

Mountpoint für Amazon S3 ist ein einfacher Dateiclient mit hohem Durchsatz für Mounten eines Amazon S3-Buckets als lokales Dateisystem. Mit Mountpoint für Amazon S3 können Ihre Anwendungen durch Dateioperationen wie Öffnen und Lesen auf in Amazon S3 gespeicherte Objekte zugreifen. Mountpoint für Amazon S3 übersetzt diese Operationen automatisch in Amazon S3-Objekt-API-Aufrufe, sodass Ihre Anwendungen über eine Dateischnittstelle auf den elastischen Speicher und Durchsatz von Amazon S3 zugreifen können.

Mountpoint kann wie folgt installiert werden die Mountpoint-Installationsanleitung. Mountpoint verwendet das AWS-Profil, das für die Installation lokal ist und auf einer Amazon S3-Präfix-Ebene arbeitet. Stellen Sie sicher, dass das verwendete Profil über eine Richtlinie verfügt, die s3 aktiviert:GetObject, s3:ListBucket, und kms:decrypt-Berechtigungen für das Amazon S3-URI-Präfix der Lesesätze oder des Sequenzspeichers, auf die zugegriffen wird. Danach kann der Bucket über den folgenden Pfad gemountet werden:

```
mount-s3 access point arn local path to mount --prefix prefix to sequence store or read
 set --region region
```

Verwenden CloudFront mit HealthOmics

Amazon CloudFront ist ein Content Delivery Network (CDN) -Service, der auf hohe Leistung, Sicherheit und Komfort für Entwickler ausgelegt ist. Kunden, die Folgendes nutzen möchten CloudFront muss mit dem Serviceteam zusammenarbeiten, um das einzuschalten CloudFront Vertrieb. Arbeiten Sie mit Ihrem Account-Team zusammen, um die HealthOmics Serviceteam.

Lesesätze aktivieren in HealthOmics

Sie können Lesesätze aktivieren, die mit dem Start-Lese-Set-Activation-Job API-Betrieb oder über die AWS CLI, wie im folgenden Beispiel gezeigt. Ersetzen Sie den *sequence store ID* und *read set id* mit Ihrer Sequenzspeicher-ID und Ihren Lesesatz-IDs.

```
aws omics start-read-set-activation-job
     --sequence-store-id sequence store ID \
     --sources readSetId=read set ID readSetId=read set id_1 read set id_2
```

Sie erhalten eine Antwort, die die Informationen zum Aktivierungsauftrag enthält, wie im Folgenden dargestellt.

```
{
    "id": "12345678",
    "sequenceStoreId": "1234567890",
    "status": "SUBMITTED",
    "creationTime": "2022-10-22T00:50:54.670000+00:00"
}
```

Nachdem der Aktivierungsauftrag gestartet wurde, können Sie seinen Fortschritt mit dem get-read-set-activation-job API-Betrieb. Das Folgende ist ein Beispiel für die Verwendung der AWS CLI um den Status Ihres Aktivierungsauftrags zu überprüfen. Ersetzen *job ID* und *sequence store ID* mit Ihrer Sequenzspeicher-ID bzw. Ihren Job-IDs.

```
aws omics get-read-set-activation-job --id job ID --sequence-store-id sequence store ID
```

Die Antwort fasst den Aktivierungsauftrag zusammen, wie im Folgenden dargestellt.

```
{
```

```
    "id": 123567890,
    "sequenceStoreId": 123467890,
    "status": "SUBMITTED",
    "statusUpdateReason": "The job is submitted and will start soon.",
    "creationTime": "2022-10-22T00:50:54.670000+00:00",
    "sources": [
        {
            "readSetId": <reads set id_1>,
            "status": "NOT_STARTED",
            "statusUpdateReason": "The source is queued for the job."
        },
        {
            "readSetId": <read set id_2>,
            "status": "NOT_STARTED",
            "statusUpdateReason": "The source is queued for the job."
        }
    ]
}
```

Sie können den Status eines Aktivierungsauftrags mit dem get-read-set-metadata API-Betrieb. Mögliche Status sind `ACTIVE`, `ACTIVATING`, und `ARCHIVED`. Ersetzen Sie im folgenden Beispiel *`sequence store ID`* mit Ihrer Sequenzspeicher-ID und ersetzen Sie *`read set ID`* mit Ihrer Lesesatz-ID.

```
aws omics get-read-set-metadata --sequence-store-id sequence store ID --id read set ID
```

Die folgende Antwort zeigt Ihnen, dass der Lesesatz aktiv ist.

```
{
    "id": "12345678",
    "arn": "arn:aws:omics:us-west-2:555555555555:sequenceStore/1234567890/
readSet/12345678",
    "sequenceStoreId": "0123456789",
    "subjectId": "mySubject",
    "sampleId": "mySample",
    "status": "ACTIVE",
    "name": "HG00100",
    "description": "HG00100 aligned to HG38 BAM",
    "fileType": "BAM",
    "creationTime": "2022-07-13T23:25:20Z",
    "sequenceInformation": {
        "totalReadCount": 1513467,
```

```
        "totalBaseCount": 163454436,
        "generatedFrom": "Pulled from SRA",
        "alignment": "ALIGNED"
    },
    "referenceArn": "arn:aws:omics:us-west-2:555555555555:referenceStore/0123456789/
reference/0000000001",
    "files": {
        "source1": {
            "totalParts": 2,
            "partSize":  10485760,
            "contentLength": 17112283,
            "s3Access": {
        "s3Uri": "s3://accountID-sequence store ID-ajdpi90jdas90a79fh9a8ja98jdfa9jf98-
s3alias/592761533288/sequenceStore/2015356892/readSet/9515444019/
import_source1.fastq.gz"
},
         },
        "index": {
            "totalParts": 1,
            "partSize": 53216,
            "contentLength": 10485760
            "s3Access": {
        "s3Uri": "s3://accountID-sequence store ID-ajdpi90jdas90a79fh9a8ja98jdfa9jf98-
s3alias/592761533288/sequenceStore/2015356892/readSet/9515444019/
import_source1.fastq.gz"
},
        }
    },
    "creationType": "IMPORT",
    "etag": {
        "algorithm": "BAM_MD5up",
        "source1": "d1d65429212d61d115bb19f510d4bd02"
    }
}
```

Sie können alle ReadSet-Aktivierungsaufträge anzeigen, indem Sie List-Read-Set-Activation-Jobs, wie im folgenden Beispiel gezeigt. Ersetzen Sie im folgenden Beispiel *`sequence store ID`* mit Ihrer Sequenzspeicher-ID.

```
aws omics list-read-set-activation-jobs --sequence-store-id sequence store ID
```

Sie erhalten die folgende Antwort.

```
{
    "activationJobs": [
        {
            "id": 1234657890,
            "sequenceStoreId": "1234567890",
            "status": "COMPLETED",
            "creationTime": "2022-10-22T01:33:38.079000+00:00",
            "completionTime": "2022-10-22T01:34:28.941000+00:00"
        }
    ]
}
```

HealthOmics Analytik

> **Important**
>
> AWS HealthOmics Variantenspeicher und Annotationsspeicher stehen Neukunden nicht mehr zur Verfügung. Vorhandene Kunden können den Service weiterhin wie gewohnt verwenden. Weitere Informationen finden Sie unter AWS HealthOmics Änderung der Verfügbarkeit von Variantenspeicher und Annotationsspeicher.

HealthOmics analytics unterstützt die Speicherung und Analyse genomischer Varianten und Annotationen. Analytics bietet zwei Arten von Speicherressourcen: Variantenspeicher und Annotationsspeicher. Sie verwenden diese Ressourcen, um genomische Variantendaten und Annotationsdaten zu speichern, zu transformieren und abzufragen. Nachdem Sie Daten in einen Datenspeicher importiert haben, können Sie Athena verwenden, um erweiterte Analysen der Daten durchzuführen.

Sie können die HealthOmics Konsole oder API verwenden, um Geschäfte zu erstellen und zu verwalten, Daten zu importieren und analytische Speicherdaten mit Mitarbeitern zu teilen.

Variant speichert Unterstützungsdaten in VCF-Formaten, und Annotation speichert Unterstützung TSV/CSV und Formate. GFF3 Genomische Koordinaten werden als auf Null basierende, halbgeschlossene, halboffene Intervalle dargestellt. Wenn sich Ihre Daten im HealthOmics Analysedatenspeicher befinden, wird der Zugriff auf die VCF-Dateien über verwaltet. AWS Lake Formation Anschließend können Sie die VCF-Dateien mithilfe von Amazon Athena abfragen. Abfragen müssen die Athena Query Engine Version 3 verwenden. Weitere Informationen zu den Versionen der Athena-Abfrage-Engine finden Sie in der Amazon Athena Athena-Dokumentation.

Themen

- HealthOmics Variantenspeicher erstellen
- Importjobs für HealthOmics Variantenspeicher erstellen
- HealthOmics Annotationsspeicher erstellen
- Importjobs für HealthOmics Annotationsspeicher erstellen
- HealthOmics Annotation Store-Versionen erstellen
- HealthOmics Analytics-Stores löschen

- Abfragen von HealthOmics Analysedaten
- HealthOmics Analytics-Shops teilen

HealthOmics Variantenspeicher erstellen

> **Important**
>
> AWS HealthOmics Variantenspeicher und Annotationsspeicher stehen Neukunden nicht mehr zur Verfügung. Vorhandene Kunden können den Service weiterhin wie gewohnt verwenden. Weitere Informationen finden Sie unter AWS HealthOmics Änderung der Verfügbarkeit von Variantenspeicher und Annotationsspeicher.

In den folgenden Themen wird beschrieben, wie HealthOmics Variantenspeicher mithilfe der Konsole und der API erstellt werden.

Themen

- Einen Variantenspeicher mithilfe der Konsole erstellen
- Einen Variantenspeicher mithilfe der API erstellen

Einen Variantenspeicher mithilfe der Konsole erstellen

Sie können mit der HealthOmics Konsole einen Variantenspeicher erstellen.

1. Öffnen Sie die HealthOmics -Konsole.
2. Öffnen Sie bei Bedarf den linken Navigationsbereich (±1). Wählen Sie Variant Stores aus.
3. Geben Sie auf der Seite Variantenspeicher erstellen die folgenden Informationen ein

   - Name des Variantenspeichers — Ein eindeutiger Name für diesen Shop.
   - Beschreibung (optional) — Eine Beschreibung dieses Variantenspeichers.
   - Referenzgenom — Das Referenzgenom für diesen Variantenspeicher.
   - Datenverschlüsselung — Wählen Sie aus, ob die Datenverschlüsselung von Ihnen AWS oder von Ihnen selbst verwaltet werden soll.
   - Tags (optional) — Geben Sie bis zu 50 Tags für diesen Variantenspeicher an.
4. Wählen Sie Variantenspeicher erstellen aus.

Einen Variantenspeicher mithilfe der API erstellen

Verwenden Sie den HealthOmics `CreateVariantStore` API-Betrieb, um Variantenspeicher zu erstellen. Sie können diesen Vorgang auch mit dem ausführen AWS CLI.

Um einen Variantenspeicher zu erstellen, geben Sie einen Namen für den Speicher und den ARN eines Referenzspeichers an. Der Variantenspeicher ist bereit, Daten aufzunehmen, wenn sich sein Status auf READY ändert.

Im folgenden Beispiel wird der verwendet AWS CLI , um einen Variantenspeicher zu erstellen.

```
aws omics create-variant-store --name myvariantstore \
    --reference referenceArn="arn:aws:omics:us-
west-2:555555555555:referenceStore/123456789/reference/5987565360"
```

Um die Erstellung Ihres Variantenspeichers zu bestätigen, erhalten Sie die folgende Antwort.

```
{
    "creationTime": "2022-11-03T18:19:52.296368+00:00",
    "id": "45aeb91d5678",
    "name": "myvariantstore",
    "reference": {
        "referenceArn": "arn:aws:omics:us-west-2:555555555555:referenceStore/123456789/
reference/5987565360"
    },
    "status": "CREATING"
}
```

Verwenden Sie die get-variant-storeAPI, um mehr über einen Variantenshop zu erfahren.

```
aws omics get-variant-store --name myvariantstore
```

Sie erhalten die folgende Antwort.

```
{
    "id": "45aeb91d5678",
    "reference": {
        "referenceArn": "arn:aws:omics:us-west-2:555555555555:referenceStore/123456789/
reference/5987565360"
    },
    "status": "ACTIVE",
```

```
    "storeArn": "arn:aws:omics:us-west-2:555555555555:variantStore/myvariantstore",
    "name": "myvariantstore",
    "creationTime": "2022-11-03T18:19:52.296368+00:00",
    "updateTime": "2022-11-03T18:30:56.272792+00:00",
    "tags": {},
    "storeSizeBytes": 0
}
```

Verwenden Sie die list-variant-storesAPI, um alle mit einem Konto verknüpften Variantenshops anzuzeigen.

```
aws omics list-variant-stores
```

Sie erhalten eine Antwort, in der alle Variantenshops zusammen mit ihrem IDs Status und anderen Details aufgeführt sind, wie in der folgenden Beispielantwort dargestellt.

```
{
    "variantStores": [
        {
            "id": "45aeb91d5678",
            "reference": {
                "referenceArn": "arn:aws:omics:us-
west-2:55555555555:referenceStore/5506874698"
            },
            "status": "ACTIVE",
            "storeArn": "arn:aws:omics:us-west-2:55555555555:variantStore/
new_variant_store",
            "name": "variantstore",
            "creationTime": "2022-11-03T18:19:52.296368+00:00",
            "updateTime": "2022-11-03T18:30:56.272792+00:00",
            "statusMessage": "",
            "storeSizeBytes": 141526
        }
    ]
}
```

Sie können die Antworten für die list-variant-storesAPI auch nach Status oder anderen Kriterien filtern.

VCF-Dateien, die in Analysespeicher importiert wurden, die am oder nach dem 15. Mai 2023 erstellt wurden, enthalten definierte Schemas für VEP-Anmerkungen (Variant Effect Predictor).

Dies erleichtert das Abfragen und Analysieren importierter VCF-Daten. Die Änderung wirkt sich nicht auf Stores aus, die vor dem 15. Mai 2023 erstellt wurden, es sei denn, der `annotation fields` Parameter ist im API- oder CLI-Aufruf enthalten. Für diese Shops führt die Verwendung des `annotation fields` Parameters dazu, dass die Anfrage fehlschlägt.

Importjobs für HealthOmics Variantenspeicher erstellen

> **Important**
>
> AWS HealthOmics Variantenspeicher und Annotationsspeicher stehen neuen Kunden nicht mehr zur Verfügung. Vorhandene Kunden können den Service weiterhin wie gewohnt verwenden. Weitere Informationen finden Sie unter AWS HealthOmics Änderung der Verfügbarkeit von Variantenspeicher und Annotationsspeicher.

Das folgende Beispiel zeigt, wie Sie mit AWS CLI dem einen Importjob für einen Variantenspeicher erstellen können.

```
aws omics start-variant-import-job \
       --destination-name myvariantstore \
       --runLeftNormalization false \
       --role-arn  arn:aws:iam::55555555555:role/roleName \
       --items source=s3://my-omics-bucket/sample.vcf.gz source=s3://my-omics-bucket/
sample2.vcf.gz
```

```
{
    "destinationName": "store_a",
    "roleArn": "....",
    "runLeftNormalization": false,
    "items": [
        {"source": "s3://my-omics-bucket/sample.vcf.gz"},
        {"source": "s3://my-omics-bucket/sample2.vcf.gz"}
    ]
}
```

Für Geschäfte, die nach dem 15. Mai 2023 erstellt wurden, zeigt das folgende Beispiel, wie der `--annotation-fields` Parameter hinzugefügt wird. Die Annotationsfelder werden beim Import definiert.

```
aws omics start-variant-import-job \
   --destination-name annotationparsingvariantstore \
   --role-arn arn:aws:iam::123456789012:role/<role_name> \
   --items source=s3://pathToS3/sample.vcf
   --annotation-fields '{"VEP": "CSQ"}'
```

```
{
    "jobId": "981e2286-e954-4391-8a97-09aefc343861"
}
```

Wird verwendet get-variant-import-job, um den Status zu überprüfen.

```
aws omics get-variant-import-job --job-id 08279950-a9e3-4cc3-9a3c-a574f9c9e229
```

Sie erhalten eine JSON-Antwort, die den Status Ihres Importauftrags anzeigt. VEP-Anmerkungen in der VCF werden nach Informationen geparst, die in der INFO-Spalte als Paar gespeichert sind. ID/Value Die Standard-ID für die INFO-Spalte der Anmerkungen zu Ensembl Variant Effect Predictor ist CSQ, aber Sie können den `--annotation-fields` Parameter verwenden, um einen benutzerdefinierten Wert anzugeben, der in der INFO-Spalte verwendet wird. Das Parsen wird derzeit für VEP-Anmerkungen unterstützt.

Für einen Shop, der vor dem 15. Mai 2023 erstellt wurde, oder für VCF-Dateien, die keine VEP-Anmerkung enthalten, enthält die Antwort keine Kommentarfelder.

```
{
    "creationTime": "2023-04-11T17:52:37.241958+00:00",
    "destinationName": "annotationparsingvariantstore",
    "id": "7a1c67e3-b7f9-434d-817b-9c571fd63bea",
    "items": [

    {
       "jobStatus": "COMPLETED",
       "source": "s3://amzn-s3-demo-bucket/NA12878.2k.garvan.vcf"
    }
 ],
    "roleArn": "arn:aws:iam::555555555555:role/<role_name>",

    "runLeftNormalization": false,
    "status": "COMPLETED",
    "updateTime": "2023-04-11T17:58:22.676043+00:00",
```

```
}
```

Die VEP-Anmerkungen, die Teil von VCF-Dateien sind, werden als vordefiniertes Schema mit der folgenden Struktur gespeichert. Das Feld extras kann verwendet werden, um alle zusätzlichen VEP-Felder zu speichern, die nicht im Standardschema enthalten sind.

```
annotations struct<
   vep: array<struct<
      allele:string,
      consequence: array<string>,
      impact:string,
      symbol:string,
      gene:string,
      `feature_type`: string,
      feature: string,
      biotype: string,
      exon: struct<rank:string, total:string>,
      intron: struct<rank:string, total:string>,
      hgvsc: string,
      hgvsp: string,
      `cdna_position`: string,
      `cds_position`: string,
      `protein_position`: string,
      `amino_acids`: struct<reference:string, variant: string>,
      codons: struct<reference:string, variant: string>,
      `existing_variation`: array<string>,
      distance: string,
      strand: string,
      flags: array<string>,
      symbol_source: string,
      hgnc_id: string,
      `extras`: map<string, string>
    >>
>
```

Die Analyse erfolgt nach bestem Wissen und Gewissen. Wenn der VEP-Eintrag nicht den VEP-Standardspezifikationen entspricht, wird er nicht analysiert und die Zeile im Array ist leer.

Bei einem neuen Variantenspeicher get-variant-import-jobwürde die Antwort für die Annotationsfelder enthalten, wie in der Abbildung gezeigt.

```
aws omics get-variant-import-job --job-id 08279950-a9e3-4cc3-9a3c-a574f9c9e229
```

Sie erhalten eine JSON-Antwort, die den Status Ihres Importauftrags anzeigt.

```
{
    "creationTime": "2023-04-11T17:52:37.241958+00:00",
    "destinationName": "annotationparsingvariantstore",
    "id": "7a1c67e3-b7f9-434d-817b-9c571fd63bea",
    "items": [

    {
    "jobStatus": "COMPLETED",
    "source": "s3://amzn-s3-demo-bucket/NA12878.2k.garvan.vcf"
    }
 ],
    "roleArn": "arn:aws:iam::123456789012:role/<role_name>",
    "runLeftNormalization": false,
    "status": "COMPLETED",
    "updateTime": "2023-04-11T17:58:22.676043+00:00",
    "annotationFields" : {"VEP": "CSQ"}
  }
}
```

Sie können list-variant-import-jobssie verwenden, um alle Importaufträge und deren Status zu sehen.

```
aws omics list-variant-import-jobs --ids 7a1c67e3-b7f9-434d-817b-9c571fd63bea
```

Die Antwort enthält Informationen wie folgt.

```
{
    "variantImportJobs": [
    {
        "creationTime": "2023-04-11T17:52:37.241958+00:00",
        "destinationName": "annotationparsingvariantstore",
        "id": "7a1c67e3-b7f9-434d-817b-9c571fd63bea",
        "roleArn": "arn:aws:iam::55555555555:role/roleName",
        "runLeftNormalization": false,
        "status": "COMPLETED",
        "updateTime": "2023-04-11T17:58:22.676043+00:00",
        "annotationFields" : {"VEP": "CSQ"}
        }
    ]
  }
```

```
}
```

Bei Bedarf können Sie einen Importauftrag mit dem folgenden Befehl abbrechen.

```
aws omics cancel-variant-import-job
     --job-id edd7b8ce-xmpl-47e2-bc99-258cac95a508
```

HealthOmics Annotationsspeicher erstellen

> **Important**
>
> AWS HealthOmics Variantenspeicher und Annotationsspeicher stehen neuen Kunden nicht mehr zur Verfügung. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter AWS HealthOmics Änderung der Verfügbarkeit von Variantenspeicher und Annotationsspeicher.

Ein Annotationsspeicher ist ein Datenspeicher, der eine Annotationsdatenbank darstellt, z. B. eine Datenbank aus einer TSV-, VCF- oder GFF-Datei. Wenn dasselbe Referenzgenom angegeben wird, werden Annotationsspeicher während eines Imports demselben Koordinatensystem zugeordnet wie Variantenspeicher. In den folgenden Themen wird gezeigt, wie Sie die HealthOmics Konsole verwenden und AWS CLI Annotationsspeicher erstellen und verwalten.

Themen

- Erstellen eines Annotationsspeichers mithilfe der Konsole
- Einen Annotationsspeicher mithilfe der API erstellen

Erstellen eines Annotationsspeichers mithilfe der Konsole

Gehen Sie wie folgt vor, um Annotationsspeicher mit der HealthOmics Konsole zu erstellen.

Um einen Annotationsspeicher zu erstellen

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

4. Geben Sie auf der Seite Annotationsspeicher erstellen die folgenden Informationen ein

   - Name des Speichers für Anmerkungen — Ein eindeutiger Name für diesen Speicher.
   - Beschreibung (optional) — Eine Beschreibung dieses Referenzgenoms.
   - Datenformat und Schemadetails — Wählen Sie das Datendateiformat aus und laden Sie die Schemadefinition für diesen Speicher hoch.
   - Referenzgenom — Das Referenzgenom für diese Annotation.
   - Datenverschlüsselung — Wählen Sie aus, ob die Datenverschlüsselung von Ihnen AWS oder von Ihnen selbst verwaltet werden soll.
   - Tags (optional) — Stellen Sie bis zu 50 Tags für diesen Annotationsspeicher bereit.
5. Wählen Sie „Annotationsspeicher erstellen“.

Einen Annotationsspeicher mithilfe der API erstellen

Das folgende Beispiel zeigt, wie Sie einen Annotationsspeicher mit dem erstellen AWS CLI. Für alle AWS CLI API-Operationen müssen Sie das Format Ihrer Daten angeben.

```
aws omics create-annotation-store --name my_annotation_store \
           --store-format GFF \
           --reference referenceArn="arn:aws:omics:us-
west-2:555555555555:referenceStore/6505293348/reference/5987565360"
           --version-name new_version
```

Sie erhalten die folgende Antwort, um die Erstellung Ihres Annotationsspeichers zu bestätigen.

```
{
           "creationTime": "2022-08-24T20:34:19.229500Z",
           "id": "3b93cdef69d2",
           "name": "my_annotation_store",
           "reference": {
               "referenceArn": "arn:aws:omics:us-
west-2:555555555555:referenceStore/6505293348/reference/5987565360"
           },
           "status": "CREATING"
           "versionName": "my_version"
       }
```

Verwenden Sie die get-annotation-storeAPI, um mehr über einen Annotationsspeicher zu erfahren.

```
aws omics get-annotation-store --name my_annotation_store
```

Sie erhalten die folgende Antwort.

```
{
          "id": "eeb019ac79c2",
          "reference": {
              "referenceArn": "arn:aws:omics:us-
west-2:555555555555:referenceStore/5638433913/reference/5871590330“
          },
          "status": "ACTIVE",
          "storeArn": "arn:aws:omics:us-west-2:555555555555:annotationStore/gffstore",
          "name": "my_annotation_store",
          "creationTime": "2022-11-05T00:05:19.136131+00:00",
          "updateTime": "2022-11-05T00:10:36.944839+00:00",
          "tags": {},
          "storeFormat": "GFF",
          "statusMessage": "",
          "storeSizeBytes": 0,
          "numVersions": 1
      }
```

Verwenden Sie den list-annotation-storesAPI-Vorgang, um alle mit einem Konto verknüpften Annotationsspeicher anzuzeigen.

```
aws omics list-annotation-stores
```

Sie erhalten eine Antwort, in der alle Annotationsspeicher zusammen mit ihrem IDs Status und anderen Details aufgeführt sind, wie in der folgenden Beispielantwort dargestellt.

```
{
           "annotationStores": [
               {
                  "id": "4d8f3eada259",
                   "reference":
                       "referenceArn": "arn:aws:omics:us-
west-2:555555555555:referenceStore/5638433913/reference/5871590330"
                   },
                   "status": "CREATING",
                   "name": "gffstore",
                   "creationTime": "2022-09-27T17:30:52.182990+00:00",
```

```
                "updateTime": "2022-09-27T17:30:53.025362+00:00"
            }
        ]
    }
```

Sie können Antworten auch nach Status oder anderen Kriterien filtern.

Importjobs für HealthOmics Annotationsspeicher erstellen

> **Important**
>
> AWS HealthOmics Variantenspeicher und Annotationsspeicher stehen neuen Kunden nicht mehr zur Verfügung. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter AWS HealthOmics Änderung der Verfügbarkeit von Variantenspeicher und Annotationsspeicher.

Themen

- Einen Job zum Importieren von Anmerkungen mithilfe der API erstellen
- Zusätzliche Parameter für die Formate TSV und VCF
- Annotationsspeicher im TSV-Format erstellen
- Importaufträge im VCF-Format werden gestartet

Einen Job zum Importieren von Anmerkungen mithilfe der API erstellen

Das folgende Beispiel zeigt, wie Sie den verwenden AWS CLI , um einen Importjob für Anmerkungen zu starten.

```
aws omics start-annotation-import-job \
           --destination-name myannostore \
           --version-name myannostore \
           --role-arn arn:aws:iam::123456789012:role/roleName \
           --items source=s3://my-omics-bucket/sample.vcf.gz
           --annotation-fields '{"VEP": "CSQ"}'
```

Annotationsspeicher, die vor dem 15. Mai 2023 erstellt wurden, geben eine Fehlermeldung zurück, wenn die Annotationsfelder enthalten sind. Sie geben keine Ausgabe für API-Operationen zurück, die mit Importaufträgen für Annotationsspeicher verbunden sind.

Sie können dann den get-annotation-import-jobAPI-Vorgang und den `job ID` Parameter verwenden, um weitere Informationen zum Importjob für Anmerkungen zu erhalten.

```
aws omics get-annotation-import-job --job-id 9e4198fb-fa85-446c-9301-9b823a1a8ba8
```

Sie erhalten die folgende Antwort, einschließlich der Annotationsfelder.

```
{
          "creationTime": "2023-04-11T19:09:25.049767+00:00",
          "destinationName": "parsingannotationstore",
          "versionName": "parsingannotationstore",
          "id": "9e4198fb-fa85-446c-9301-9b823a1a8ba8",
          "items": [
              {
                  "jobStatus": "COMPLETED",
                  "source": "s3://my-omics-bucket/sample.vep.vcf"
              }
          ],
          "roleArn": "arn:aws:iam::55555555555:role/roleName",
          "runLeftNormalization": false,
          "status": "COMPLETED",
          "updateTime": "2023-04-11T19:13:09.110130+00:00",
          "annotationFields" : {"VEP": "CSQ"}
       }
```

Um alle Importaufträge für den Annotationsspeicher anzuzeigen, verwenden Sie list-annotation-import-jobs.

```
aws omics list-annotation-import-jobs --ids 9e4198fb-fa85-446c-9301-9b823a1a8ba8
```

Die Antwort enthält die Details und den Status Ihrer Importaufträge für den Annotationsspeicher.

```
{
          "annotationImportJobs": [
          {
              "creationTime": "2023-04-11T19:09:25.049767+00:00",
              "destinationName": "parsingannotationstore",
              "versionName": "parsingannotationstore",
```

```
            "id": "9e4198fb-fa85-446c-9301-9b823a1a8ba8",
            "roleArn": "arn:aws:iam::55555555555:role/roleName",
            "runLeftNormalization": false,
            "status": "COMPLETED",
            "updateTime": "2023-04-11T19:13:09.110130+00:00",
            "annotationFields" : {"VEP": "CSQ"}
        }
        ]
    }
```

Zusätzliche Parameter für die Formate TSV und VCF

Für die Formate TSV und VCF gibt es zusätzliche Parameter, die die API darüber informieren, wie Ihre Eingabe analysiert werden soll.

> **Important**
>
> CSV-Annotationsdaten, die mit Abfrage-Engines exportiert werden, geben direkt Informationen aus dem Datensatzimport zurück. Wenn die importierten Daten Formeln oder Befehle enthalten, wird die Datei möglicherweise einer CSV-Injektion unterzogen. Daher können mit Abfrage-Engines exportierte Dateien zu Sicherheitswarnungen führen. Um böswillige Aktivitäten zu vermeiden, deaktivieren Sie Links und Makros beim Lesen von Exportdateien.

Der TSV-Parser führt auch grundlegende bioinformatische Operationen durch, wie die Normalisierung der linken Seite und die Standardisierung von genomischen Koordinaten, die in der folgenden Tabelle aufgeführt sind.

Typ des Formats	Description
Generisch	Generische Textdatei. Keine genomischen Informationen.
`CHR_POS`	Startposition - 1, Endposition hinzufügen, die identisch `POS` ist mit.
`CHR_POS_REF_ALT`	Enthält Informationen zu den Allelen Contig, 1-Basen-Position, Ref und Alt.

Typ des Formats	Description
CHR_START_END_REF_ALT_ONE_BASE	Enthält Contig-, Start-, End-, Ref- und Alt-Allel informationen. Die Koordinaten basieren auf Eins.
CHR_START_END_ZERO_BASE	Enthält Zähl-, Start- und Endpositionen. Die Koordinaten basieren auf 0.
CHR_START_END_ONE_BASE	Enthält Längs-, Start- und Endpositionen. Die Koordinaten basieren auf 1.
CHR_START_END_REF_ALT_ZERO_BASE	Enthält Contig-, Start-, End-, Ref- und Alt-Allel informationen. Die Koordinaten basieren auf 0.

Eine Anforderung für den TSV-Import eines Annotationsspeichers sieht wie das folgende Beispiel aus.

```
aws omics start-annotation-import-job \
--destination-name tsv_anno_example \
--role-arn arn:aws:iam::555555555555:role/demoRole \
--items source=s3://demodata/genomic_data.bed.gz \
--format-options '{ "tsvOptions": {
        "readOptions": {
            "header": false,
            "sep": "\t"
        }
    }
}'
```

Annotationsspeicher im TSV-Format erstellen

Im folgenden Beispiel wird mithilfe einer tabulatorbeschränkten Datei, die eine Kopfzeile, Zeilen und Kommentare enthält, ein Annotationsspeicher erstellt. Die Koordinaten lautenCHR_START_END_ONE_BASED, und sie enthält die HG19 Genkarte aus der Synopsis of the Human Gene Map der OMIM.

```
aws omics create-annotation-store --name mimgenemap \
```

```
  --store-format TSV \
  --reference=referenceArn=arn:aws:omics:us-
west-2:555555555555:referenceStore/6505293348/reference/2310864158 \
  --store-options=tsvStoreOptions='{
    annotationType=CHR_START_END_ONE_BASE,
    formatToHeader={CHR=chromosome, START=genomic_position_start,
 END=genomic_position_end},
    schema=[
      {chromosome=STRING},
      {genomic_position_start=LONG},
      {genomic_position_end=LONG},
      {cyto_location=STRING},
      {computed_cyto_location=STRING},
      {mim_number=STRING},
      {gene_symbols=STRING},
      {gene_name=STRING},
      {approved_gene_name=STRING},
      {entrez_gene_id=STRING},
      {ensembl_gene_id=STRING},
      {comments=STRING},
      {phenotypes=STRING},
      {mouse_gene_symbol=STRING}]}'
```

Sie können Dateien mit oder ohne Header importieren. Um dies in einer CLI-Anforderung anzugeben, verwenden Sie`header=false`, wie im folgenden Beispiel für einen Importjob gezeigt.

```
aws omics start-annotation-import-job \
   --role-arn arn:aws:iam::555555555555:role/demoRole \
   --items=source=s3://amzn-s3-demo-bucket/annotation-examples/hg38_genemap2.txt \
   --destination-name output-bucket \
   --format-options=tsvOptions='{readOptions={sep="\t",header=false,comment="#"}}'
```

Im folgenden Beispiel wird ein Speicher für Anmerkungen für eine BED-Datei erstellt. Eine Bettdatei ist eine einfache tabulatorgetrennte Datei. In diesem Beispiel lauten die Spalten Chromosom, Start, Ende und Regionsname. Die Koordinaten basieren auf Null, und die Daten haben keinen Header.

```
aws omics create-annotation-store \
   --name cexbed --store-format TSV \
   --reference=referenceArn=arn:aws:omics:us-
west-2:555555555555:referenceStore/6505293348/reference/2310864158 \
   --store-options=tsvStoreOptions='{
   annotationType=CHR_START_END_ZERO_BASE,
```

```
   formatToHeader={CHR=chromosome, START=start, END=end},
   schema=[{chromosome=STRING}, {start=LONG}, {end=LONG}, {name=STRING}]}'
```

Anschließend können Sie die Bed-Datei mit dem folgenden CLI-Befehl in den Annotationsspeicher importieren.

```
aws omics start-annotation-import-job \
   --role-arn arn:aws:iam::555555555555:role/demoRole \
   --items=source=s3://amzn-s3-demo-bucket/TruSeq_Exome_TargetedRegions_v1.2.bed \
   --destination-name cexbed \
   --format-options=tsvOptions='{readOptions={sep="\t",header=false,comment="#"}}'
```

Im folgenden Beispiel wird ein Annotationsspeicher für eine tabulatorgetrennte Datei erstellt, die die ersten Spalten einer VCF-Datei enthält, gefolgt von Spalten mit Annotationsinformationen. Es enthält Genompositionen mit Informationen zu den Chromosomen-, Start-, Referenz- und alternativen Allelen sowie eine Überschrift.

```
aws omics create-annotation-store --name gnomadchrx --store-format TSV \
--reference=referenceArn=arn:aws:omics:us-
west-2:555555555555:referenceStore/6505293348/reference/2310864158 \
--store-options=tsvStoreOptions='{
    annotationType=CHR_POS_REF_ALT,
    formatToHeader={CHR=chromosome, POS=start, REF=ref, ALT=alt},
    schema=[
        {chromosome=STRING},
        {start=LONG},
        {ref=STRING},
        {alt=STRING},
        {filters=STRING},
        {ac_hom=STRING},
        {ac_het=STRING},
        {af_hom=STRING},
        {af_het=STRING},
        {an=STRING},
        {max_observed_heteroplasmy=STRING}]}'
```

Anschließend würden Sie die Datei mit dem folgenden CLI-Befehl in den Annotationsspeicher importieren.

```
aws omics start-annotation-import-job \
```

```
  --role-arn arn:aws:iam::555555555555:role/demoRole \
   --items=source=s3://amzn-s3-demo-bucket/
gnomad.genomes.v3.1.sites.chrM.reduced_annotations.tsv \
   --destination-name gnomadchrx \
   --format-options=tsvOptions='{readOptions={sep="\t",header=true,comment="#"}}'
```

Das folgende Beispiel zeigt, wie ein Kunde einen Annotationsspeicher für eine mim2gene-Datei erstellen kann. Eine mim2Gene-Datei stellt die Verbindungen zwischen den Genen in OMIM und einem anderen Genidentifikator bereit. Sie ist tabulatorgetrennt und enthält Kommentare.

```
aws omics create-annotation-store \
  --name mim2gene \
  --store-format TSV \
  --reference=referenceArn=arn:aws:omics:us-
west-2:555555555555:referenceStore/6505293348/reference/2310864158 \
  --store-options=tsvStoreOptions='
    {annotationType=GENERIC,
    formatToHeader={},
    schema=[
        {mim_gene_id=STRING},
        {mim_type=STRING},
        {entrez_id=STRING},
        {hgnc=STRING},
        {ensembl=STRING}]}'
```

Anschließend können Sie Daten wie folgt in Ihren Shop importieren.

```
aws omics start-annotation-import-job \
   --role-arn arn:aws:iam::555555555555:role/demoRole \
   --items=source=s3://xquek-dev-aws/annotation-examples/mim2gene.txt \
   --destination-name mim2gene \
   --format-options=tsvOptions='{readOptions={sep="\t",header=false,comment="#"}}'
```

Importaufträge im VCF-Format werden gestartet

Für VCF-Dateien gibt es zwei zusätzliche Eingaben, `ignoreQualField` und, die diese Parameter ignorieren oder einschließen`ignoreFilterField`, wie in der Abbildung gezeigt.

```
aws omics start-annotation-import-job --destination-name annotation_example\
```

```
  --role-arn arn:aws:iam::555555555555:role/demoRole \
  --items source=s3://demodata/example.garvan.vcf \
  --format-options '{ "vcfOptions": {
    "ignoreQualField": false,
    "ignoreFilterField": false
    }
   }'
```

Sie können einen Import eines Annotationsspeichers auch abbrechen, wie hier gezeigt. Wenn die Stornierung erfolgreich ist, erhalten Sie keine Antwort auf diesen AWS CLI Anruf. Wenn die Importauftrags-ID jedoch nicht gefunden wird oder der Importauftrag abgeschlossen ist, erhalten Sie eine Fehlermeldung.

```
aws omics cancel-annotation-import-job --job-id edd7b8ce-xmpl-47e2-bc99-258cac95a508
```

> **Note**
>
> Ihre Metadaten importieren den Jobverlauf für get-annotation-import-jobget-variant-import-job, list-annotation-import-jobs, und list-variant-import-jobswerden nach zwei Jahren automatisch gelöscht. Die importierten Varianten- und Annotationsdaten werden nicht automatisch gelöscht und verbleiben in Ihren Datenspeichern.

HealthOmics Annotation Store-Versionen erstellen

> **Important**
>
> AWS HealthOmics Variantenspeicher und Annotationsspeicher stehen neuen Kunden nicht mehr zur Verfügung. Vorhandene Kunden können den Service weiterhin wie gewohnt verwenden. Weitere Informationen finden Sie unter AWS HealthOmics Änderung der Verfügbarkeit von Variantenspeicher und Annotationsspeicher.

Sie können neue Versionen von Annotationsspeichern erstellen, um verschiedene Versionen Ihrer Annotationsdatenbanken zu sammeln. Dies hilft Ihnen bei der Organisation Ihrer Annotationsdaten, die regelmäßig aktualisiert werden.

Verwenden Sie die create-annotation-store-versionAPI, um eine neue Version eines vorhandenen Annotationsspeichers zu erstellen, wie im folgenden Beispiel gezeigt.

```
aws omics create-annotation-store-version \
     --name my_annotation_store \
     --version-name my_version
```

Sie erhalten die folgende Antwort mit der Versions-ID des Annotationsspeichers, die bestätigt, dass eine neue Version Ihrer Annotation erstellt wurde.

```
{
     "creationTime": "2023-07-21T17:15:49.251040+00:00",
     "id": "3b93cdef69d2",
     "name": "my_annotation_store",
     "reference": {
         "referenceArn": "arn:aws:omics:us-
west-2:555555555555:referenceStore/6505293348/reference/5987565360"
     },
     "status": "CREATING",
     "versionName": "my_version"
}
```

Um die Beschreibung einer Annotation Store-Version zu aktualisieren, können Sie damit Aktualisierungen update-annotation-store-versionzu einer Annotation Store-Version hinzufügen.

```
aws omics update-annotation-store-version \
    --name my_annotation_store \
    --version-name my_version \
    --description "New Description"
```

Sie erhalten die folgende Antwort, in der bestätigt wird, dass die Annotation Store-Version aktualisiert wurde.

```
{
     "storeId": "4934045d1c6d",
     "id": "2a3f4a44aa7b",
     "description":"New Description",
     "status": "ACTIVE",
     "name": "my_annotation_store",
     "versionName": "my_version",
     "creation Time": "2023-07-21T17:20:59.380043+00:00",
     "updateTime": "2023-07-21T17:26:17.892034+00:00"
}
```

Um die Details einer Annotation Store-Version anzuzeigen, verwenden Sie get-annotation-store-version.

```
aws omics get-annotation-store-version --name my_annotation_store --version-name
 my_version
```

Sie erhalten eine Antwort mit dem Versionsnamen, dem Status und anderen Details.

```
{
     "storeId": "4934045d1c6d",
     "id": "2a3f4a44aa7b",
     "status": "ACTIVE",
     "versionArn": "arn:aws:omics:us-west-2:555555555555:annotationStore/
my_annotation_store/version/my_version",
     "name": "my_annotation_store",
     "versionName": "my_version",
     "creationTime": "2023-07-21T17:15:49.251040+00:00",
     "updateTime": "2023-07-21T17:15:56.434223+00:00",
     "statusMessage": "",
     "versionSizeBytes": 0
    }
```

Um alle Versionen eines Annotationsspeichers anzuzeigen, können Sie list-annotation-store-versions, wie im folgenden Beispiel gezeigt, Folgendes verwenden.

```
aws omics list-annotation-store-versions --name my_annotation_store
```

Sie erhalten eine Antwort mit den folgenden Informationen

```
{
  "annotationStoreVersions": [
    {
     "storeId": "4934045d1c6d",
     "id": "2a3f4a44aa7b",
     "status": "CREATING",
     "versionArn": "arn:aws:omics:us-west-2:555555555555:annotationStore/
my_annotation_store/version/my_version_2",
     "name": "my_annotation_store",
     "versionName": "my_version_2",
     "creation Time": "2023-07-21T17:20:59.380043+00:00",
```

```
     "versionSizeBytes": 0
    },
    {
     "storeId": "4934045d1c6d",
     "id": "4934045d1c6d",
     "status": "ACTIVE",
     "versionArn": "arn:aws:omics:us-west-2:555555555555:annotationStore/
my_annotation_store/version/my_version_1",
     "name": "my_annotation_store",
     "versionName": "my_version_1",
     "creationTime": "2023-07-21T17:15:49.251040+00:00",
     "updateTime": "2023-07-21T17:15:56.434223+00:00",
     "statusMessage": "",
     "versionSizeBytes": 0
    }
}
```

Wenn Sie eine Version des Annotationsspeichers nicht mehr benötigen, können Sie delete-annotation-store-versionssie verwenden, um eine Version des Annotationsspeichers zu löschen, wie im folgenden Beispiel gezeigt.

```
aws omics delete-annotation-store-versions --name my_annotation_store --versions
 my_version
```

Wenn die Store-Version ohne Fehler gelöscht wird, erhalten Sie die folgende Antwort.

```
{
  "errors": []
}
```

Wenn Fehler auftreten, erhalten Sie eine Antwort mit den Einzelheiten der Fehler, wie in der Abbildung dargestellt.

```
{
  "errors": [
    {
      "versionName": "my_version",
      "message": "Version with versionName: my_version was not found."
    }
  ]
}
```

Wenn Sie versuchen, eine Annotation Store-Version zu löschen, die über einen aktiven Importauftrag verfügt, erhalten Sie eine Antwort mit einem Fehler, wie hier gezeigt.

```
{
  "errors": [
    {
      "versionName": "my_version",
      "message": "version has an inflight import running"
    }
  ]
}
```

In diesem Fall können Sie das Löschen der Annotation Store-Version erzwingen, wie im folgenden Beispiel gezeigt.

```
aws omics delete-annotation-store-versions --name my_annotation_store --versions
 my_version --force
```

HealthOmics Analytics-Stores löschen

> **Important**
>
> AWS HealthOmics Variantenspeicher und Annotationsspeicher stehen neuen Kunden nicht mehr zur Verfügung. Vorhandene Kunden können den Service weiterhin wie gewohnt verwenden. Weitere Informationen finden Sie unter AWS HealthOmics Änderung der Verfügbarkeit von Variantenspeicher und Annotationsspeicher.

Wenn Sie einen Varianten- oder Annotationsspeicher löschen, löscht das System auch alle importierten Daten in diesem Speicher und alle zugehörigen Tags.

Das folgende Beispiel zeigt, wie Sie einen Variantenspeicher mit dem AWS CLI löschen. Wenn die Aktion erfolgreich ist, wechselt der Status des Variantenspeichers zu`DELETING`.

```
aws omics delete-variant-store --id <variant-store-id>
```

Das folgende Beispiel zeigt, wie ein Annotationsspeicher gelöscht wird. Wenn die Aktion erfolgreich ist, wechselt der Status des Annotationsspeichers zu`DELETING`. Annotationsspeicher können nicht gelöscht werden, wenn mehr als eine Version vorhanden ist.

```
aws omics delete-annotation-store --id <annotation-store-id>
```

Abfragen von HealthOmics Analysedaten

> **Important**
>
> AWS HealthOmics Variantenspeicher und Annotationsspeicher stehen neuen Kunden nicht mehr zur Verfügung. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter AWS HealthOmics Änderung der Verfügbarkeit von Variantenspeicher und Annotationsspeicher.

Sie können Abfragen für Ihre Variantenshops mit AWS Lake Formation Amazon Athena oder Amazon EMR durchführen. Bevor Sie Abfragen ausführen, müssen Sie die (in den folgenden Abschnitten beschriebenen) Einrichtungsverfahren für Lake Formation und Amazon Athena abschließen.

Informationen zu Amazon EMR finden Sie unter Tutorial: Erste Schritte mit Amazon EMR

HealthOmics Partitionieren Sie bei Variantenspeichern, die nach dem 26. September 2024 erstellt wurden, den Speicher nach der Proben-ID. Diese Partitionierung bedeutet, dass die Proben-ID HealthOmics verwendet wird, um die Speicherung der Varianteninformationen zu optimieren. Abfragen, die Probeninformationen als Filter verwenden, geben schneller Ergebnisse zurück, da die Abfrage weniger Daten scannt.

HealthOmics verwendet Beispiel IDs als Partitionsdateinamen. Bevor Sie Daten aufnehmen, überprüfen Sie, ob die Proben-ID PHI-Daten enthält. Ist dies der Fall, ändern Sie die Proben-ID, bevor Sie die Daten aufnehmen. Weitere Informationen darüber, welche Inhalte in die Stichprobe aufgenommen werden sollen und welche nicht IDs, finden Sie in den Anleitungen auf der Webseite zur AWS HIPAA-Konformität.

Themen

- Konfiguration von Lake Formation zur Verwendung HealthOmics
- Athena für Abfragen konfigurieren
- Abfragen in HealthOmics Variantenspeichern ausführen

Konfiguration von Lake Formation zur Verwendung HealthOmics

> **Important**
>
> AWS HealthOmics Variantenspeicher und Annotationsspeicher stehen Neukunden nicht mehr zur Verfügung. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter AWS HealthOmics Änderung der Verfügbarkeit von Variantenspeicher und Annotationsspeicher.

Bevor Sie Lake Formation zur Verwaltung von HealthOmics Datenspeichern verwenden, führen Sie die folgenden Lake Formation-Konfigurationsverfahren durch.

Themen

- Lake Formation-Administratoren erstellen oder verifizieren
- Erstellen von Ressourcenlinks mithilfe der Lake Formation-Konsole
- Konfiguration von Berechtigungen für AWS RAM gemeinsame Nutzung von Ressourcen

Lake Formation-Administratoren erstellen oder verifizieren

Bevor Sie einen Data Lake in Lake Formation erstellen können, definieren Sie einen oder mehrere Administratoren.

Administratoren sind Benutzer und Rollen mit Berechtigungen zum Erstellen von Ressourcenlinks. Sie richten Data Lake-Administratoren pro Konto pro Region ein.

Erstellen Sie in der Lake Formation-Konsole einen Admin-Benutzer

1. Öffnen Sie die AWS Lake Formation-Konsole: Lake Formation-Konsole
2. Wenn auf der Konsole das Fenster Willkommen bei Lake Formation angezeigt wird, wählen Sie Erste Schritte.

   Lake Formation fügt Sie zur Data Lake-Administratortabelle hinzu.
3. Andernfalls wählen Sie im linken Menü die Option Administrative Rollen und Aufgaben aus.
4. Fügen Sie nach Bedarf weitere Administratoren hinzu.

Erstellen von Ressourcenlinks mithilfe der Lake Formation-Konsole

Um eine gemeinsam genutzte Ressource zu erstellen, die Benutzer abfragen können, müssen die Standardzugriffskontrollen deaktiviert sein. Weitere Informationen zum Deaktivieren der Standardzugriffskontrollen finden Sie unter Ändern der Standardsicherheitseinstellungen für Ihren Data Lake in der Lake Formation-Dokumentation. Sie können Ressourcenlinks einzeln oder als Gruppe erstellen, sodass Sie auf Daten in Amazon Athena oder anderen AWS Diensten (wie Amazon EMR) zugreifen können.

Erstellen von Ressourcenlinks in der AWS Lake Formation-Konsole und deren Weitergabe an HealthOmics Analytics-Benutzer

1. Öffnen Sie die AWS Lake Formation-Konsole: Lake Formation-Konsole
2. Wählen Sie in der primären Navigationsleiste Datenbanken aus.
3. Wählen Sie in der Tabelle Datenbanken die gewünschte Datenbank aus.
4. Wählen Sie im Menü Erstellen die Option Ressourcenlink aus.
5. Geben Sie einen Ressourcen-Linknamen ein. Wenn Sie vorhaben, von Athena aus auf die Datenbank zuzugreifen, geben Sie einen Namen ein, der nur Kleinbuchstaben verwendet (bis zu 256 Zeichen).
6. Wählen Sie Erstellen aus.
7. Der neue Ressourcenlink ist jetzt unter Datenbanken aufgeführt.

Gewähren Sie mithilfe der Lake Formation-Konsole Zugriff auf die gemeinsam genutzte Ressource

Ein Lake Formation-Datenbankadministrator kann mithilfe des folgenden Verfahrens Zugriff auf die gemeinsam genutzte Ressource gewähren.

1. Öffnen Sie die AWS Lake Formation-Konsole: https://console.aws.amazon.com/lakeformation/
2. Wählen Sie in der primären Navigationsleiste Datenbanken aus.
3. Wählen Sie auf der Seite Datenbanken den Ressourcenlink aus, den Sie zuvor erstellt haben.
4. Wählen Sie im Menü „Aktionen“ die Option „Auf Zielebene gewähren“ aus.
5. Wählen Sie auf der Seite Datenberechtigungen gewähren unter Principals die Option IAM-Benutzer oder -Rollen aus.
6. Suchen Sie im Dropdownmenü IAM-Benutzer oder -Rollen den Benutzer, dem Sie Zugriff gewähren möchten.

7. Wählen Sie als Nächstes unter der Karte LF-Tags oder Katalogressourcen die Option Ressourcen für benannte Datenkataloge aus.
8. Wählen Sie im Tables-optional Dropdownmenü Alle Tabellen oder die Tabelle aus, die Sie zuvor erstellt haben.
9. Wählen Sie auf der Karte mit den Tabellenberechtigungen unter Tabellenberechtigungen die Option Beschreiben und Auswählen aus.
10. Wählen Sie als Nächstes Grant aus.

Um die Lake Formation-Berechtigungen anzuzeigen, wählen Sie im primären Navigationsbereich die Option Data Lake-Berechtigungen aus. Die Tabelle zeigt die verfügbaren Datenbanken und Ressourcenlinks.

Konfiguration von Berechtigungen für AWS RAM gemeinsame Nutzung von Ressourcen

Sehen Sie sich die Berechtigungen in der AWS Lake Formation-Konsole an, indem Sie in der primären Navigationsleiste Data Lake-Berechtigungen auswählen. Auf der Seite mit den Datenberechtigungen können Sie unter RAM Resource Share eine Tabelle mit den Ressourcentypen und Datenbanken einsehen, **ARN** die sich auf eine gemeinsam genutzte Ressource beziehen. Wenn Sie eine AWS Resource Access Manager (AWS RAM) - Ressourcenfreigabe akzeptieren müssen, werden Sie in der Konsole AWS Lake Formation darüber informiert.

HealthOmics kann die AWS RAM Ressourcenfreigaben bei der Erstellung des Shops implizit akzeptieren. Um die AWS RAM Ressourcenfreigabe zu akzeptieren, muss der IAM-Benutzer oder die IAM-Rolle, der die `CreateVariantStore` oder `CreateAnnotationStore` API-Operationen aufruft, die folgenden Aktionen zulassen:

- `ram:GetResourceShareInvitations`- Diese Aktion ermöglicht das HealthOmics Auffinden der Einladungen.
- `ram:AcceptResourceShareInvitation`- Diese Aktion ermöglicht es HealthOmics , die Einladung mithilfe eines FAS-Tokens anzunehmen.

Ohne diese Berechtigungen wird bei der Erstellung des Shops ein Autorisierungsfehler angezeigt.

Hier ist eine Beispielrichtlinie, die diese Aktionen beinhaltet. Fügen Sie diese Richtlinie dem IAM-Benutzer oder der IAM-Rolle hinzu, der bzw. die die AWS RAM Ressourcenfreigabe akzeptiert.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "omics:*",
        "ram:AcceptResourceShareInvitation",
        "ram:GetResourceShareInvitations"
      ],
      "Resource": "*"
    }
  ]
}
```

Athena für Abfragen konfigurieren

> **Important**
>
> AWS HealthOmics Variantenspeicher und Annotationsspeicher stehen Neukunden nicht mehr zur Verfügung. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter AWS HealthOmics Änderung der Verfügbarkeit von Variantenspeicher und Annotationsspeicher.

Sie können Athena verwenden, um Varianten und Anmerkungen abzufragen. Bevor Sie Abfragen ausführen, führen Sie die folgenden Einrichtungsaufgaben aus:

Themen

- Konfigurieren Sie mithilfe der Athena-Konsole einen Speicherort für Abfrageergebnisse
- Konfigurieren Sie eine Arbeitsgruppe mit der Athena-Engine v3

Konfigurieren Sie mithilfe der Athena-Konsole einen Speicherort für Abfrageergebnisse

Gehen Sie folgendermaßen vor, um einen Speicherort für Abfrageergebnisse zu konfigurieren.

1. Öffnen Sie die Athena-Konsole: Athena-Konsole
2. Wählen Sie in der primären Navigationsleiste Abfrage-Editor aus.
3. Wählen Sie im Abfrage-Editor die Registerkarte „Einstellungen“ und dann „Verwalten“.
4. Geben Sie das S3-Präfix eines Standorts ein, um das Abfrageergebnis zu speichern.

Konfigurieren Sie eine Arbeitsgruppe mit der Athena-Engine v3

Gehen Sie wie folgt vor, um eine Arbeitsgruppe zu konfigurieren.

1. Öffnen Sie die Athena-Konsole: Athena-Konsole
2. Wählen Sie in der primären Navigationsleiste Arbeitsgruppen und dann Arbeitsgruppe erstellen aus.
3. Geben Sie einen Namen für die Arbeitsgruppe ein.
4. Wählen Sie Athena SQL als Engine-Typ aus.
5. Wählen Sie unter Abfrage-Engine aktualisieren die Option Manuell aus.
6. Wählen Sie unter Query Version Engine die Option Athena Version 3 aus.
7. Wählen Sie Create workgroup (Arbeitsgruppe erstellen) aus.

Abfragen in HealthOmics Variantenspeichern ausführen

> **Important**
>
> AWS HealthOmics Variantenspeicher und Annotationsspeicher stehen Neukunden nicht mehr zur Verfügung. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter AWS HealthOmics Änderung der Verfügbarkeit von Variantenspeicher und Annotationsspeicher.

Mit Amazon Athena können Sie Abfragen in Ihrem Variantenshop durchführen. Beachten Sie, dass genomische Koordinaten in Varianten- und Annotationsspeichern als auf Null basierende, halb geschlossene, halboffene Intervalle dargestellt werden.

Führen Sie eine einfache Abfrage mit der Athena-Konsole aus

Das folgende Beispiel zeigt, wie eine einfache Abfrage ausgeführt wird.

1. Öffnen Sie den Athena-Abfrage-Editor: Athena-Abfrage-Editor
2. Wählen Sie unter Arbeitsgruppe die Arbeitsgruppe aus, die Sie beim Setup erstellt haben.
3. Vergewissern Sie sich, dass Datenquelle AwsDataCatalog
4. Wählen Sie für Datenbank den Link zur Datenbankressource aus, den Sie während der Einrichtung von Lake Formation erstellt haben.
5. Kopieren Sie die folgende Abfrage in den Abfrage-Editor auf der Registerkarte Abfrage 1:

```
SELECT * from omicsvariants limit 10
```

6. Wählen Sie dann Ausführen aus, um die Abfrage auszuführen. Die Konsole füllt die Ergebnistabelle mit den ersten 10 Zeilen der omicsvariants Tabelle auf.

Führen Sie eine komplexe Abfrage mit der Athena-Konsole aus

Das folgende Beispiel zeigt, wie eine komplexe Abfrage ausgeführt wird. Um diese Abfrage auszuführen, importieren Sie `ClinVar` sie in den Annotationsspeicher.

Führen Sie eine komplexe Abfrage aus

1. Öffnen Sie den Athena-Abfrage-Editor: Athena-Abfrage-Editor
2. Wählen Sie unter Arbeitsgruppe die Arbeitsgruppe aus, die Sie beim Setup erstellt haben.
3. Vergewissern Sie sich, dass Datenquelle AwsDataCatalog
4. Wählen Sie für Datenbank den Link zur Datenbankressource aus, den Sie während der Einrichtung von Lake Formation erstellt haben.
5. Wählen Sie + oben rechts, um eine neue Abfrage-Registerkarte mit dem Namen Query 2 zu erstellen.
6. Kopieren Sie die folgende Abfrage in den Abfrage-Editor auf der Registerkarte Abfrage 2:

```
SELECT variants.sampleid,
  variants.contigname,
  variants.start,
  variants."end",
  variants.referenceallele,
  variants.alternatealleles,
  variants.attributes AS variant_attributes,
  clinvar.attributes AS clinvar_attributes
FROM omicsvariants as variants
```

```
INNER JOIN omicsannotations as clinvar ON
  variants.contigname=CONCAT('chr',clinvar.contigname)
  AND variants.start=clinvar.start
  AND variants."end"=clinvar."end"
  AND variants.referenceallele=clinvar.referenceallele
  AND variants.alternatealleles=clinvar.alternatealleles
WHERE clinvar.attributes['CLNSIG']='Likely_pathogenic'
```

7. Wählen Sie Ausführen, um die Ausführung der Abfrage zu starten.

HealthOmics Analytics-Shops teilen

> **Important**
>
> AWS HealthOmics Variantenspeicher und Annotationsspeicher stehen Neukunden nicht mehr zur Verfügung. Vorhandene Kunden können den Service weiterhin wie gewohnt verwenden. Weitere Informationen finden Sie unter AWS HealthOmics Änderung der Verfügbarkeit von Variantenspeicher und Annotationsspeicher.

Als Besitzer eines Variantenspeichers oder eines Annotationsspeichers können Sie den Shop mit anderen AWS-Konten teilen. Der Eigentümer kann den Zugriff auf die gemeinsam genutzte Ressource widerrufen, indem er die gemeinsame Nutzung löscht.

Als Abonnent eines gemeinsam genutzten Shops akzeptieren Sie zunächst die Freigabe. Anschließend können Sie Workflows definieren, die den gemeinsamen Speicher verwenden. Die Daten werden sowohl in Lake Formation als auch in AWS Glue Form einer Tabelle angezeigt.

Wenn Sie keinen Zugriff mehr auf den Shop benötigen, löschen Sie den Share.

Weitere Informationen Cross-account gemeinsame Nutzung von Ressourcen in AWS HealthOmics zur gemeinsamen Nutzung von Ressourcen finden Sie unter.

Eine Shop-Freigabe erstellen

Verwenden Sie den API-Vorgang create-share, um ein Store-Share zu erstellen. Der Hauptabonnent ist AWS-Konto der Benutzer, der die Aktie abonnieren wird. Im folgenden Beispiel wird eine Aktie für einen Variantenspeicher erstellt. Um einen Shop mit mehr als einem Konto zu teilen, erstellen Sie mehrere Shares desselben Shops.

```
aws omics create-share  \
        --resource-arn "arn:aws:omics:us-west-2:555555555555:variantStore/
omics_dev_var_store" \
        --principal-subscriber "123456789012" \
        --name "my_Share-123"
```

Wenn die Erstellung erfolgreich ist, erhältst du eine Antwort mit der Share-ID und dem Status.

```
{
       "shareId": "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a",
       "name": "my_Share-123",
       "status": "PENDING"
  }
```

Die Freigabe bleibt so lange im Status „Ausstehend“, bis der Abonnent sie mithilfe des API-Vorgangs „Accept-Share“ akzeptiert.

Cross-account gemeinsame Nutzung von Ressourcen in AWS HealthOmics

Verwenden Sie die kontoübergreifende gemeinsame Nutzung, um Ressourcen mit anderen zu teilen, ohne Kopien zu erstellen oder IAM-Ressourcenrichtlinien zu ändern. Die folgenden Ressourcen unterstützen das kontoübergreifende Teilen:

- HealthOmics Varianten speichern
- HealthOmics Speicher für Anmerkungen
- Private Arbeitsabläufe

> ⚠ Vertraue dem Workflow-Besitzer
>
> Stellen Sie sicher, dass Sie dem Workflow-Besitzer vertrauen, bevor Sie eine Freigabe akzeptieren. Wenn Sie einen gemeinsamen Workflow ausführen, werden die Workflow-Definition und die Container-Images, die der Besitzer bereitgestellt hat, in Ihrem ausgeführt AWS-Konto. Sie werden mit den Berechtigungen der IAM-Servicerolle ausgeführt, die Sie für die Ausführung angeben. Der Inhalt des Workflows und seiner Container-Bilder sind für Sie nicht sichtbar. Das Ausführen eines gemeinsamen Workflows entspricht der Ausführung von Drittanbietercode in Ihrem Konto. Geben Sie eine Dienstrolle mit geringsten Rechten ein und verwenden Sie eine Sitzungsrichtlinie, um die Berechtigungen für die Ausführung weiter einzuschränken. Weitere Informationen finden Sie unter Verwenden Sie Sitzungsrichtlinien, um Berechtigungen einzuschränken.

Die gemeinsame Nutzung einer Ressource umfasst die folgenden Schritte:

1. Der Ressourcenbesitzer erstellt einen Share und gibt den ARN der Ressource und den AWS-Konto des beabsichtigten Abonnenten an. Die Ressourcenfreigabe bleibt so lange im Status „Ausstehend“, bis der Abonnent die Freigabe akzeptiert.
2. Der Abonnent akzeptiert die Ressourcenfreigabe, um Zugriff auf die Ressource zu erhalten. Der Ressourcenanteil geht in den Aktivierungszustand über.
3. Der HealthOmics Dienst bietet einem Abonnentenkonto Zugriff auf die Ressource.
4. Der Ressourcenbesitzer kann den Share löschen, oder der Abonnent kann seinen Zugriff auf den Share widerrufen. Der Abonnent kann den Share oder die zugehörige Ressource nicht löschen.

Themen

- Einen Share erstellen
- Ruft Informationen über eine Aktie ab
- Sehen Sie sich die Aktien an, die Sie besitzen
- Sehen Sie sich akzeptierte Aktien von anderen Konten an
- Löschen Sie eine Aktie

Einen Share erstellen

Sie können den API-Vorgang create-share verwenden, um eine Freigabe zu erstellen. Der Hauptabonnent ist AWS-Konto der Benutzer, der die gemeinsam genutzte Ressource abonnieren wird. Im folgenden Beispiel wird ein Share für einen Variantenspeicher erstellt.

```
aws omics create-share \
    --resource-arn "arn:aws:omics:us-west-2:555555555555:variantStore/
omics_dev_var_store" \
    --principal-subscriber "123456789012"  \
    --name "my_Share-123"
```

Wenn die Erstellung erfolgreich ist, erhalten Sie eine Antwort mit der Share-ID und dem Status.

```
{
"shareId": "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a",
     "name": "my_Share-123",
     "status": "PENDING"
}
```

Die Aktie bleibt im Status „Ausstehend“, bis der Abonnent sie mithilfe der accept-share API-Operation akzeptiert.

```
 aws omics accept-share \
    --share-id "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a"
```

Nachdem der Abonnent die Aktie akzeptiert hat, geht die Aktie in den aktiven Status über.

```
{
"status": "ACTIVATING"
}
```

Ruft Informationen über eine Aktie ab

Verwenden Sie den API-Vorgang get-share, um Informationen über die Aktie abzurufen.

```
aws omics get-share --share-id "495c21bedc889d07d0ab69d710a6841e-
dd75ab7a1a9c384fa848b5bd8e5a7e0a"
```

Die API-Antwort enthält Metadateninformationen über den Share.

```
{
  "share":
    {
      "shareId": "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a",
      "name": "my_Share-123",
      "resourceArn": "arn:aws:omics:us-west-2:555555555555:variantStore/
omics_dev_var_store",
      "principalSubscriber": "123456789012",
      "ownerId": "555555555555",
      "status": "PENDING"
    }
}
```

Sehen Sie sich die Aktien an, die Sie besitzen

Verwenden Sie die List-Shares-API, um Informationen zu den einzelnen Aktien abzurufen, die Sie besitzen.

```
aws omics list-shares  --resource-owner SELF
```

Die API-Antwort enthält die Metadaten für jede Aktie, die Sie besitzen.

Sehen Sie sich akzeptierte Aktien von anderen Konten an

Verwenden Sie die List-Shares-API, um alle Aktien anzuzeigen, die Sie von anderen Konten akzeptiert haben.

```
aws omics list-shares  --resource-owner OTHER
```

Die API-Antwort enthält die Metadaten für jede Aktie, die du akzeptiert hast.

Löschen Sie eine Aktie

Verwenden Sie die Delete-Share API, um einen Share zu löschen, nachdem Sie ihn nicht mehr benötigen.

```
aws omics delete-share \
    --share-id "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a"
```

Ressourcen taggen in HealthOmics

Themen

- Wichtiger Hinweis
- Ressourcen taggen HealthOmics
- Sequenzspeicher, gelesene Satz-Tags
- Hinzufügen eines Tags zu einer HealthOmics Ressource
- Tags für eine Ressource auflisten
- Tags aus einem Datenspeicher entfernen

Wichtiger Hinweis

HealthOmics schützt Kundendaten gemäß den Richtlinien des AWS-Modells für gemeinsame Verantwortung. Das bedeutet, dass alle Kundendaten sowohl bei der Übertragung als auch bei der Speicherung verschlüsselt werden. Allerdings sind nicht alle vom Kunden eingegebenen Namen für Ressourcen wie Datenspeicher oder auftragsbasierte Vorgänge verschlüsselt. Sie sollten niemals persönlich identifizierbare Informationen oder geschützte Gesundheitsinformationen enthalten. Weitere Informationen finden Sie unter Sicherheit in AWS HealthOmics.

Ressourcen taggen HealthOmics

Sie können Ihren AWS-Ressourcen mithilfe von Tags Metadaten zuweisen. Jedes Tag ist ein Label, das aus einem benutzerdefinierten Schlüssel und Wert besteht. Mit Tags können Sie Ressourcen verwalten, identifizieren, organisieren, suchen und filtern.

In diesem Thema werden häufig verwendete Tagging-Kategorien und -Strategien beschrieben, mit denen Sie eine konsistente und effektive Tagging-Strategie implementieren können. In den folgenden Abschnitten werden Grundkenntnisse zu AWS-Ressourcen, Tagging, detaillierter Abrechnung und AWS Identity and Access Management vorausgesetzt.

Jedes -Tag besteht aus zwei Teilen:

- Ein Tag-Schlüssel (zum Beispiel CostCenter „Umgebung“ oder „Projekt“). Bei Tag-Schlüsseln wird zwischen Groß- und Kleinschreibung unterschieden.

- Ein Tag-Wert (zum Beispiel 111122223333 oder Production). Wie bei Tag-Schlüsseln wird auch bei Tag-Werten zwischen Groß- und Kleinschreibung unterschieden.

Sie können Tags verwenden, um Ressourcen nach Zweck, Eigentümer, Umgebung oder anderen Kriterien zu kategorisieren. Weitere Informationen finden Sie unter AWS Tagging-Strategien.

Sie können Tags für eine Ressource über die Servicekonsole, die Service-API oder die AWS CLI hinzufügen, ändern oder entfernen.

Um das Tagging zu aktivieren, stellen Sie sicher, dass TagResources es autorisiert ist. Sie können die Autorisierung durchführen, TagResources indem Sie eine IAM-Richtlinie wie im folgenden Beispiel anhängen.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "omics:Create*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "omics:Start*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "omics:Tag*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "omics:Untag*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "omics:List*",
```

```
            "Resource": "*"
        }
    ]
}
```

Best Practices

Beachten Sie bei der Erstellung einer Tagging-Strategie für AWS-Ressourcen die folgenden bewährten Methoden:

- Speichern Sie keine personenbezogenen Daten (PII), geschützte Gesundheitsinformationen (PHI) oder andere sensible Informationen in Tags.
- Verwenden Sie für Tags ein standardisiertes Format, bei dem die Groß-/Kleinschreibung beachtet wird, und wenden Sie es konsistent für alle Ressourcentypen an.
- Verwenden Sie Tag-Richtlinien, die mehrere Zwecke unterstützen, wie die Verwaltung der Ressourcenzugriffskontrolle, Kostenverfolgung, Automatisierung und Organisation.
- Verwenden Sie automatisierte Tools, um Ressourcen-Tags zu verwalten. AWS-Ressourcengruppen und die Resource Groups Tagging API ermöglichen die programmatische Steuerung von Tags, sodass Tags und Ressourcen automatisch verwaltet, gesucht und gefiltert werden können.
- Tagging ist effektiver, wenn Sie mehr Tags verwenden.
- Tags können bearbeitet oder geändert werden, wenn sich die Benutzeranforderungen ändern. Um Zugriffskontroll-Tags zu aktualisieren, müssen Sie jedoch auch die Richtlinien aktualisieren, die auf diese Tags verweisen, um den Zugriff auf Ihre Ressourcen zu kontrollieren.

Anforderungen zum Markieren

Für Tags gelten zwei Anforderungen:

- Schlüsseln darf nicht das Präfix aws: vorangestellt werden.
- Schlüssel müssen in einem Tag-Satz eindeutig sein.
- Schlüssel müssen zwischen 1 und 128 Zeichen lang sein.
- Ein Wert muss zwischen 0 und 256 Zeichen haben.
- Werte müssen nicht pro Tagsatz eindeutig sein.

- Zulässige Zeichen für Schlüssel und Werte sind Unicode-Buchstaben, Ziffern, Leerzeichen sowie die folgenden Sonderzeichen: _ . : / = + - @.
- Bei Schlüsseln und Werten wird die Groß-/Kleinschreibung berücksichtigt.

Sequenzspeicher, gelesene Satz-Tags

Bei Sequenzspeichern befinden sich die auf dem Lesesatz erstellten Tags auf der Ressourcenebene des Lesesatzes. Lesesätze enthalten auch Objekte unter ihnen, auf die mit S3 zugegriffen, gesucht und eingeschränkt werden kann APIs. Standardmäßig werden die Proben-ID (omics:SampleID) und die Betreff-ID (omics:SubjectID) dem Objekt hinzugefügt.

Darüber hinaus können bis zu fünf Tags zwischen dem Lesesatz und den Objekten darunter synchronisiert werden. Die Konfiguration, für die Tags synchronisiert werden sollen, ist eine Konfiguration auf Filialebene, die während der Erstellung oder Aktualisierung des Shops mithilfe des `propogatedSetLevelTags` Parameters festgelegt wurde.

Wenn sich bereits Daten im Store befinden, kann die Aktualisierung der Schlüssel einige Zeit in Anspruch nehmen. Während dieses Updates HealthOmics ändert sich der Status des Speichers aufUpdating. HealthOmics Setzt den Speicherstatus nach Abschluss aufActive. Während die Tags verbreitet werden, können Berechtigungen, die sich auf die Tags stützen, möglicherweise nicht durchgesetzt werden. Berechtigungen werden durchgesetzt, nachdem die Tag-Weitergabe abgeschlossen ist.

Wenn Tags für den Lesesatz gesetzt oder aktualisiert werden, entscheidet das System auf der Grundlage der Speicherkonfiguration, ob die Objekte für diesen Lesesatz aktualisiert werden sollen.

Hinzufügen eines Tags zu einer HealthOmics Ressource

Das Hinzufügen von Tags zu einer Ressource kann Ihnen helfen, Ihre AWS-Ressourcen zu identifizieren und zu organisieren und den Zugriff darauf zu verwalten. Zunächst fügen Sie einer Ressource ein oder mehrere Tags (Schlüssel-Wert-Paare) hinzu. Sie können bis zu 50 Tags pro Ressource verwenden. Es gibt auch Einschränkungen hinsichtlich der Zeichen, die Sie in den Schlüssel- und Wertfeldern verwenden können.

Nachdem Sie Tags hinzugefügt haben, können Sie IAM-Richtlinien erstellen, um den Zugriff auf die AWS Ressource auf der Grundlage dieser Tags zu verwalten. Sie können die HealthOmics Konsole oder die verwenden AWS CLI , um einer Ressource Tags hinzuzufügen. Das Hinzufügen von Tags zu einem Repository kann Auswirkungen auf den Zugriff auf dieses Repository haben. Bevor Sie

einem Datenspeicher ein Tag hinzufügen, überprüfen Sie alle IAM-Richtlinien, die möglicherweise Tags verwenden, um den Zugriff auf Ressourcen wie Datenspeicher zu steuern.

Service-Tags werden sowohl für einen Betreff als auch für eine Proben-ID für Sequenzspeicher automatisch generiert.

Gehen Sie wie folgt vor AWS CLI , um einer HealthOmics Ressource ein Tag hinzuzufügen. Um beispielsweise Tags zu einem Sequenzspeicher hinzuzufügen, während dieser erstellt wird, verwenden Sie den folgenden Befehl in der AWS CLI. Der Name des Sequenzspeichers lautet MySequenceStore, und die beiden hinzugefügten Tags mit Schlüsseln sind key1 und key2 mit Werten wie Wert1 bzw. Wert2 :

```
aws omics create-sequence-store --name "MySequenceStore" --tags key1=value1,key2=value2
```

Die Ausgabe listet die Tags nicht auf. Sie gibt die folgende Antwort zurück.

```
{
    "id": "6860403586",
    "referenceStoreId": "4889894479",
    "roleArn": "arn:aws:iam::555555555555:role/ImportTest",
    "status": "CREATED",
    "creationTime": "2022-07-21T01:19:07.194Z"
}
```

Um einer vorhandenen Ressource Tags hinzuzufügen, würden Sie den folgenden Beispielbefehl ausführen.

```
aws omics tag-resource --resource-arn arn:aws:omics:us-
west-2:555555555555:sequenceStore/2275234794 --tags key1=value1,key2=value2
```

Bei Erfolg gibt dieser Befehl keine Antwort zurück.

Tags für eine Ressource auflisten

Gehen Sie wie folgt vor AWS CLI , um mit dem eine Liste der AWS Tags für eine HealthOmics Ressource anzuzeigen. Wenn keine Tags hinzugefügt wurden, ist die zurückgegebene Liste leer.

Führen Sie den Befehl im Terminal oder in der list-tags-for-resource Befehlszeile aus, wie im folgenden Beispiel gezeigt.

```
aws omics list-tags-for-resource --resource-arn arn:aws:omics:us-
west-2:555555555555:sequenceStore/2275234794
```

Als Antwort erhalten Sie eine Liste von Tags im JSON-Format.

```
 {
    "tags": {
        "key1": "value1",
        "key2": "value2"
    }
}
```

Tags aus einem Datenspeicher entfernen

Sie können ein oder mehrere mit einer Ressource verknüpfte Tags entfernen. Durch das Entfernen eines Tags wird das Tag nicht aus anderen AWS-Ressourcen gelöscht, die mit diesem Tag verknüpft sind.

Führen Sie im Terminal oder in der Befehlszeile den Befehl untag-resource aus und geben Sie den Amazon-Ressourcennamen (ARN) der Ressource an, von der Sie Tags entfernen möchten, und den Tag-Schlüssel des Tags, das Sie entfernen möchten.

```
aws omics untag-resource --resource-arn arn:aws:omics:us-
west-2:555555555555:sequenceStore/2275234794 --tag-keys key1,key2
```

Bei Erfolg gibt dieser Befehl keine Antwort zurück. Um die der Ressource zugeordneten Tags zu überprüfen, führen Sie den Befehl list-tags-for-resource aus.

IAM-Berechtigungen für HealthOmics

Sie können AWS Identity and Access Management (IAM) verwenden, um den Zugriff auf die HealthOmics API und Ressourcen wie Stores und Workflows zu verwalten. Für Benutzer und Anwendungen in Ihrem Konto, die diese verwenden HealthOmics, verwalten Sie die Berechtigungen in einer Berechtigungsrichtlinie, die Sie auf IAM-Benutzer, -Gruppen oder -Rollen anwenden können.

Um die Berechtigungen für Benutzer und Anwendungen in Ihren Konten zu verwalten, verwenden Sie die Richtlinien, die HealthOmics Ihnen zur Verfügung stehen, oder schreiben Sie Ihre eigenen. Die HealthOmics Konsole verwendet mehrere Dienste, um Informationen über die Konfiguration und die Auslöser Ihrer Funktion abzurufen. Sie können die bereitgestellten Richtlinien unverändert oder als Ausgangspunkt für restriktivere Richtlinien verwenden.

HealthOmics verwendet IAM-Dienstrollen, um in Ihrem Namen auf andere Dienste zuzugreifen. Sie würden beispielsweise eine Servicerolle erstellen oder auswählen, wenn Sie einen Workflow ausführen, der Daten aus Amazon S3 liest. Für einige Funktionen müssen Sie auch Berechtigungen für Ressourcen in anderen Diensten konfigurieren. Überprüfen Sie diese Anforderungen, bevor Sie mit der Arbeit beginnen HealthOmics

Weitere Informationen zu IAM finden Sie unter Was ist IAM? im IAM-Benutzerhandbuch.

Themen

- Identitätsbasierte IAM-Richtlinien für HealthOmics
- Servicerollen für AWS HealthOmics
- Amazon-ECR-Berechtigungen
- HealthOmics Berechtigungen für Ressourcen
- Berechtigungen für den Datenzugriff mit Amazon S3 URIs

Identitätsbasierte IAM-Richtlinien für HealthOmics

Um Benutzern in Ihrem Konto Zugriff auf zu gewähren HealthOmics, verwenden Sie identitätsbasierte Richtlinien in AWS Identity and Access Management (IAM). Identitätsbasierte Richtlinien können direkt für IAM-Benutzer oder für IAM-Gruppen und -Rollen gelten, die einem Benutzer zugeordnet sind. Sie können auch Benutzern in einem anderen Konto die Berechtigung erteilen, eine Rolle in Ihrem Konto zu übernehmen und auf Ihre HealthOmics-Ressourcen zuzugreifen.

Um Benutzern die Erlaubnis zu erteilen, Aktionen an einer Workflow-Version auszuführen, müssen Sie den Workflow und die spezifische Workflow-Version zur Ressourcenliste hinzufügen.

Die folgende IAM-Richtlinie ermöglicht einem Benutzer den Zugriff auf alle HealthOmics API-Aktionen und die Übergabe von Servicerollen an HealthOmics.

Example Richtlinie für Benutzer:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "omics:*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "iam:PassRole"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": "omics.amazonaws.com"
        }
      }
    }
  ]
}
```

Wenn Sie diese Dienste verwenden HealthOmics, interagieren Sie auch mit anderen AWS Diensten. Um auf diese Dienste zuzugreifen, verwenden Sie die verwalteten Richtlinien, die von den einzelnen Diensten bereitgestellt werden. Um den Zugriff auf eine Teilmenge von Ressourcen zu beschränken, können Sie die verwalteten Richtlinien als Ausgangspunkt verwenden, um Ihre eigenen restriktiveren Richtlinien zu erstellen.

- AmazonS3 FullAccess — Zugriff auf Amazon S3-Buckets und Objekte, die von Jobs verwendet werden.
- Amazon EC2 ContainerRegistryFullAccess — Zugriff auf Amazon ECR-Registries und Repositorys für Workflow-Container-Images.
- AWSLakeFormationDataAdmin— Zugriff auf Lake Formation Formation-Datenbanken und -Tabellen, die von Analytics-Stores erstellt wurden.
- ResourceGroupsandTagEditorFullAccess— Taggen Sie HealthOmics Ressourcen mit HealthOmics Tagging-API-Vorgängen.

Die oben genannten Richtlinien erlauben es einem Benutzer nicht, IAM-Rollen zu erstellen. Damit ein Benutzer mit diesen Berechtigungen einen Job ausführen kann, muss ein Administrator die Servicerolle erstellen, die die HealthOmics Berechtigung zum Zugriff auf Datenquellen gewährt. Weitere Informationen finden Sie unter Servicerollen für AWS HealthOmics.

Definieren Sie benutzerdefinierte IAM-Berechtigungen für Läufe

Sie können jeden Workflow, jede Ausführung oder jede Ausführungsgruppe, auf die in der `StartRun` Anfrage verwiesen wird, in eine Autorisierungsanfrage aufnehmen. Führen Sie dazu die gewünschte Kombination von Workflows, Läufen oder Ausführungsgruppen in der IAM-Richtlinie auf. Sie können beispielsweise die Verwendung eines Workflows auf einen bestimmten Lauf oder eine bestimmte Ausführungsgruppe beschränken. Sie können auch angeben, dass ein Workflow nur mit einer Ausführungsgruppe verwendet werden soll.

Im Folgenden finden Sie ein Beispiel für eine IAM-Richtlinie, die einen einzelnen Workflow mit einer einzigen Ausführungsgruppe ermöglicht.

JSON

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

```
                "omics:StartRun"
            ],
            "Resource": [
                "arn:aws:omics:us-west-2:123456789012:workflow/1234567",
                "arn:aws:omics:us-west-2:123456789012:runGroup/2345678"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "omics:StartRun"
            ],
            "Resource": [
                "arn:aws:omics:us-west-2:123456789012:run/*",
                "arn:aws:omics:us-west-2:123456789012:runGroup/2345678"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "omics:GetRun",
                "omics:ListRunTasks",
                "omics:GetRunTask",
                "omics:CancelRun",
                "omics:DeleteRun"
            ],
            "Resource": [
                "arn:aws:omics:us-west-2:123456789012:run/*"
            ]
        }
    ]
}
```

Servicerollen für AWS HealthOmics

Eine Servicerolle ist eine AWS Identity and Access Management (IAM) -Rolle, die einem AWS Service Berechtigungen für den Zugriff auf Ressourcen in Ihrem Konto erteilt. Sie stellen eine Servicerolle zur Verfügung AWS HealthOmics , wenn Sie einen Importauftrag oder eine Ausführung starten.

Die HealthOmics Konsole kann die erforderliche Rolle für Sie erstellen. Wenn Sie die HealthOmics API zur Verwaltung von Ressourcen verwenden, erstellen Sie die Servicerolle mithilfe der IAM-Konsole. Weitere Informationen finden Sie unter Erstellen Sie eine Rolle, um Berechtigungen an einen zu delegieren. AWS-Service

Für Servicerollen muss die folgende Vertrauensrichtlinie gelten.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "omics.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Die Vertrauensrichtlinie ermöglicht es dem HealthOmics Dienst, die Rolle zu übernehmen.

Themen

- Beispiel für IAM-Servicerichtlinien
- Verwenden Sie Sitzungsrichtlinien, um Berechtigungen einzuschränken
- Beispiel CloudFormation Vorlage

Beispiel für IAM-Servicerichtlinien

In diesen Beispielen sind Ressourcennamen und Konto-IDs Platzhalter, die Sie durch tatsächliche Werte ersetzen können.

Das folgende Beispiel zeigt die Richtlinie für eine Servicerolle, die Sie zum Starten einer Ausführung verwenden können. Die Richtlinie gewährt Berechtigungen für den Zugriff auf den Amazon S3-Ausgabespeicherort, die Workflow-Protokollgruppe und den Amazon ECR-Container für den Lauf.

> **Note**
>
> Wenn Sie Call Caching für den Lauf verwenden, fügen Sie den Amazon S3-Standort des Run-Cache als Ressource in den S3-Berechtigungen hinzu.

Example Richtlinie zur Rolle des Dienstes für das Starten eines Laufs

JSON

```
{
"Version":"2012-10-17",
"Statement": [
      {
          "Effect": "Allow",
          "Action": [
              "s3:GetObject",
              "s3:PutObject"
          ],
          "Resource": [
              "arn:aws:s3:::amzn-s3-demo-bucket1/*"
          ]
      },
      {
          "Effect": "Allow",
          "Action": [
              "s3:ListBucket"
          ],
          "Resource": [
              "arn:aws:s3:::amzn-s3-demo-bucket1"
          ]
      },
      {
          "Effect": "Allow",
          "Action": [
              "logs:DescribeLogStreams",
              "logs:CreateLogStream",
              "logs:PutLogEvents"
          ],
          "Resource": [
              "arn:aws:logs:us-east-1:123456789012:log-group:/aws/omics/
WorkflowLog:log-stream:*"
```

```
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogGroup"
            ],
            "Resource": [
                "arn:aws:logs:us-east-1:123456789012:log-group:/aws/omics/
WorkflowLog:*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ecr:BatchGetImage",
                "ecr:GetDownloadUrlForLayer",
                "ecr:BatchCheckLayerAvailability"
            ],
            "Resource": [
                "arn:aws:ecr:us-east-1:123456789012:repository/*"
            ]
        }
    ]
}
```

Das folgende Beispiel zeigt die Richtlinie für eine Servicerolle, die Sie für einen Shop-Importauftrag verwenden können. Die Richtlinie gewährt Berechtigungen für den Zugriff auf den Amazon S3-Eingabeort.

Example Servicerolle für den Job „Reference Store“

JSON

```
{
"Version":"2012-10-17",
"Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject"
```

```
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket/*"
            ]
        },

        {
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketLocation"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket"
            ]
        }

    ]
}
```

Das folgende Beispiel zeigt die Richtlinie für eine Servicerolle, für die Laufmetriken veröffentlicht werden CloudWatch. Weitere Informationen finden Sie unter Metriken für private Workflows ausführen.

Example Richtlinie zur Servicerolle für Run-Metriken

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "cloudwatch:PutMetricData",
      "Resource": "*"
    }
  ]
}
```

Verwenden Sie Sitzungsrichtlinien, um Berechtigungen einzuschränken

Sie können Sitzungsrichtlinien verwenden, um die Berechtigungen, die durch eine Servicerolle für einzelne Workflow-Ausführungen gewährt werden, weiter einzuschränken. Sitzungsrichtlinien sind Inline-IAM-Richtlinien, die Sie beim Starten einer Ausführung angeben. Die effektiven Berechtigungen

für die Ausführung ergeben sich aus der Schnittmenge der Berechtigungen der Servicerolle und der Sitzungsrichtlinie.

Sitzungsrichtlinien sind nützlich, wenn Sie:

- Gewähren Sie temporären Zugriff auf bestimmte Amazon S3-Buckets oder -Objekte für einen einzigen Lauf
- Beschränken Sie den Zugriff auf vertrauliche Ressourcen pro Lauf
- Wenden Sie zusätzliche Sicherheitskontrollen an, ohne die Servicerolle zu ändern

> **Important**
>
> Sitzungsrichtlinien können nur Berechtigungen einschränken; sie können keine Berechtigungen gewähren, die über das hinausgehen, was die Servicerolle bereits zulässt. Die Sitzungsrichtlinie muss Berechtigungen für Amazon CloudWatch Logs (`logs:CreateLogStreamundlogs:PutLogEvents`) enthalten, da die Anmeldeinformationen des Laufs HealthOmics verwendet werden, um Protokollgruppen zu erstellen und in sie zu schreiben.

Sitzungsrichtlinien haben eine maximale Länge von 2.048 Zeichen und müssen gültige JSON-Dokumente sein. Weitere Informationen zu Sitzungsrichtlinien finden Sie unter Sitzungsrichtlinien im IAM-Benutzerhandbuch.

Example Sitzungsrichtlinie für einen Workflow-Lauf

Das folgende Beispiel zeigt eine Sitzungsrichtlinie, die einen Lauf auf den Zugriff auf ein bestimmtes Amazon S3-Präfix für die Ausgabe beschränkt und CloudWatch Logs-Berechtigungen erfordert:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:GetObject"
      ],
      "Resource": "arn:aws:s3:::my-bucket/workflow-outputs/run-123/*"
```

```
    },
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": "arn:aws:logs:*:*:log-group:/aws/omics/WorkflowLog:*"
    }
  ]
}
```

Sie können eine Sitzungsrichtlinie angeben, wenn Sie einen Lauf mithilfe der HealthOmics API starten. Die `StartRun` API enthält einen `sessionPolicy` Parameter, in dem Sie die Inline-Richtlinie als JSON-Zeichenfolge angeben. Für Batch-Ausführungen können Sie in der eine Sitzungsrichtlinie angeben`DefaultRunSetting`, die für alle Läufe im Batch gilt.

Beispiel CloudFormation Vorlage

Die folgende CloudFormation Beispielvorlage erstellt eine Servicerolle, die den HealthOmics Zugriff auf Amazon S3-Buckets mit `omics-` vorangestellten Namen und das Hochladen von Workflow-Protokollen ermöglicht.

Example Berechtigungen für Referenzspeicher, Amazon S3 und Logs CloudWatch

```
Parameters:
  bucketName:
    Description: Bucket name
    Type: String

Resources:
  serviceRole:
    Type: AWS::IAM::Role
    Properties:
      Policies:
        - PolicyName: read-reference
          PolicyDocument:
            Version: 2012-10-17
            Statement:
            - Effect: Allow
              Action:
                - omics:*
```

```
              Resource: !Sub arn:${AWS::Partition}:omics:${AWS::Region}:
${AWS::AccountId}:referenceStore/*
        - PolicyName: read-s3
          PolicyDocument:
            Version: 2012-10-17
            Statement:
            - Effect: Allow
              Action:
                - s3:ListBucket
              Resource: !Sub arn:${AWS::Partition}:s3:::${bucketName}
            - Effect: Allow
              Action:
                - s3:GetObject
                - s3:PutObject
              Resource: !Sub arn:${AWS::Partition}:s3:::${bucketName}/*
        - PolicyName: upload-logs
          PolicyDocument:
            Version: 2012-10-17
            Statement:
            - Effect: Allow
              Action:
                - logs:DescribeLogStreams
                - logs:CreateLogStream
                - logs:PutLogEvents
              Resource: !Sub arn:${AWS::Partition}:logs:${AWS::Region}:
${AWS::AccountId}:loggroup:/aws/omics/WorkflowLog:log-stream:*
            - Effect: Allow
              Action:
                - logs:CreateLogGroup
              Resource: !Sub arn:${AWS::Partition}:logs:${AWS::Region}:
${AWS::AccountId}:loggroup:/aws/omics/WorkflowLog:*
      AssumeRolePolicyDocument: |
        {
          "Version": "2012-10-17",
          "Statement": [
            {
              "Action": [
                "sts:AssumeRole"
              ],
              "Effect": "Allow",
              "Principal": {
                "Service": [
                  "omics.amazonaws.com"
                ]
```

```
            }
          }
        ]
      }
```

Amazon-ECR-Berechtigungen

Bevor Sie einen Workflow in einem Container von Ihrem privaten Amazon ECR-Repository aus ausführen können, erstellen Sie eine Ressourcenrichtlinie für das Repository. Die Richtlinie gewährt die Genehmigung für HealthOmics Dienst zur Verwendung des Containers. Sie fügen diese Ressourcenrichtlinie jedem privaten Repository hinzu, auf das der Workflow verweist.

> **Note**
>
> Das private Repository und der Workflow müssen sich in derselben Region befinden.

Falls anders AWS Konten sind Eigentümer des Workflows und des Repositorys. Sie müssen kontoübergreifende Berechtigungen konfigurieren.

Für gemeinsame Workflows müssen Sie keinen zusätzlichen Repository-Zugriff gewähren. Sie können jedoch Richtlinien erstellen, die bestimmten Workflows den Zugriff auf das Container-Image gewähren oder verweigern.

Um die Amazon ECR-Pull-Through-Cache-Funktion verwenden zu können, müssen Sie eine Registrierungsberechtigungsrichtlinie erstellen.

Um Container-Images aus Registries zu verwenden, die Amazon ECR Pull Through Cache (PTC) nicht unterstützt, fügen Sie der Servicerolle Berechtigungen hinzu, wie unter beschrieben Richtlinie zur Dienstrolle für die Unterstützung von Containern, die nicht von PTC stammen.

In den folgenden Abschnitten wird beschrieben, wie Amazon ECR-Ressourcenberechtigungen für diese Szenarien konfiguriert werden. Weitere Informationen zu Berechtigungen in Amazon ECR finden Sie unter Berechtigungen für private Registrierungen in Amazon ECR.

Themen

- Erstellen Sie eine Ressourcenrichtlinie für das Amazon ECR-Repository
- Ausführen von Workflows mit kontoübergreifenden Containern
- Amazon ECR-Richtlinien für gemeinsame Workflows

- Richtlinien für Amazon ECR ziehen den Cache durch
- Richtlinie zur Dienstrolle für die Unterstützung von Containern, die nicht von PTC stammen
- Ich verifiziere das HealthOmics kann ein Container-Image abrufen

Erstellen Sie eine Ressourcenrichtlinie für das Amazon ECR-Repository

Erstellen Sie eine Ressourcenrichtlinie, um Folgendes zu ermöglichen HealthOmics Dienst zum Ausführen eines Workflows mithilfe eines Containers im Repository. Die Richtlinie gewährt die Genehmigung für HealthOmics Service Principal für den Zugriff auf die erforderlichen Amazon ECR-Aktionen.

Folgen Sie diesen Schritten, um die Richtlinie zu erstellen:

1. Öffnen Sie das private Repositorien Seite in der Amazon ECR-Konsole und wählen Sie das Repository aus, auf das Sie Zugriff gewähren.
2. Wählen Sie in der Navigationsleiste in der Seitenleiste Berechtigungen.
3. Wählen Sie Bearbeiten aus.
4. Wählen Richtlinien-JSON bearbeiten.
5. Fügen Sie die folgende Grundsatzerklärung hinzu und wählen Sie dann Speichern.

   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": "*"
        }
```

```
    ]
}
```

Ausführen von Workflows mit kontoübergreifenden Containern

Falls anders AWS Konten sind Eigentümer des Workflows und des Containers. Sie müssen die folgenden kontoübergreifenden Berechtigungen konfigurieren:

1. Aktualisieren Sie die Amazon ECR-Richtlinie für das Repository, um dem Konto, dem der Workflow gehört, explizit die Genehmigung zu gewähren.
2. Aktualisieren Sie die Servicerolle für das Konto, dem der Workflow gehört, um ihm Zugriff auf das Container-Image zu gewähren.

Das folgende Beispiel zeigt eine Amazon ECR-Ressourcenrichtlinie, die Zugriff auf das Konto gewährt, dem der Workflow gehört.

In diesem Beispiel:

- Workflow-Konto-ID: 111122223333
- Konto-ID des Container-Repositorys: 444455556666
- Name des Containers: samtools

JSON

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

```
        },
        {
            "Sid": "AllowAccessToTheServiceRoleOfTheAccountThatOwnsTheWorkflow",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:role/DemoCustomer"
            },
            "Action": [
                "ecr:BatchCheckLayerAvailability",
                "ecr:BatchGetImage",
                "ecr:GetDownloadUrlForLayer"
            ],
            "Resource": "*"
        }
    ]
}
```

Um die Einrichtung abzuschließen, fügen Sie der Servicerolle des Kontos, dem der Workflow gehört, die folgende Richtlinienanweisung hinzu. Die Richtlinie gewährt der Servicerolle die Erlaubnis, auf das Container-Image „samtools“ zuzugreifen. Stellen Sie sicher, dass Sie die Kontonummern, den Container-Namen und die Region durch Ihre eigenen Werte ersetzen.

```
{
    "Sid": "CrossAccountEcrRepoPolicy",
    "Effect": "Allow",
    "Action": ["ecr:BatchCheckLayerAvailability", "ecr:BatchGetImage",
 "ecr:GetDownloadUrlForLayer"],
    "Resource": "arn:aws:ecr:us-west-2:444455556666:repository/samtools"
}
```

Amazon ECR-Richtlinien für gemeinsame Workflows

> **Note**
>
> HealthOmics ermöglicht einem gemeinsamen Workflow automatisch den Zugriff auf das Amazon ECR-Repository im Konto des Workflow-Besitzers, während der Workflow im Konto des Abonnenten ausgeführt wird. Für gemeinsame Workflows müssen Sie keinen zusätzlichen Repository-Zugriff gewähren. Weitere Informationen finden Sie unter Teilen HealthOmics Workflows.

Standardmäßig haben Abonnenten keinen Zugriff auf das Amazon ECR-Repository, um die zugrunde liegenden Container zu verwenden. Optional können Sie den Zugriff auf das Amazon ECR-Repository anpassen, indem Sie der Ressourcenrichtlinie des Repositorys Bedingungsschlüssel hinzufügen. Die folgenden Abschnitte enthalten Beispielrichtlinien.

Beschränken Sie den Zugriff auf bestimmte Workflows

Sie können einzelne Workflows in einer Bedingungsanweisung auflisten, sodass nur diese Workflows Container im Repository verwenden können. Der SourceArn Der Bedingungsschlüssel gibt den ARN des gemeinsamen Workflows an. Das folgende Beispiel erteilt dem angegebenen Workflow die Erlaubnis, dieses Repository zu verwenden.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
     {
       "Sid": "OmicsAccessPrincipal",
       "Effect": "Allow",
       "Principal": {
         "Service": "omics.amazonaws.com"
       },
       "Action": [
         "ecr:GetDownloadUrlForLayer",
         "ecr:BatchGetImage",
         "ecr:BatchCheckLayerAvailability"
       ],
       "Resource": "*",
       "Condition": {
          "StringEquals": {
             "aws:SourceArn": "arn:aws:omics:us-
east-1:111122223333:workflow/1234567"
          }
       }
     }
  ]
}
```

Beschränken Sie den Zugriff auf bestimmte Konten

Sie können Abonnentenkonten in einer Bedingungsanweisung auflisten, sodass nur diese Konten berechtigt sind, Container im Repository zu verwenden. Das SourceAccount Der Bedingungsschlüssel gibt an AWS-Konto des Abonnenten. Das folgende Beispiel erteilt dem angegebenen Konto die Erlaubnis, dieses Repository zu verwenden.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "OmicsAccessPrincipal",
      "Effect": "Allow",
      "Principal": {
        "Service": "omics.amazonaws.com"
      },
      "Action": [
        "ecr:GetDownloadUrlForLayer",
        "ecr:BatchGetImage",
        "ecr:BatchCheckLayerAvailability"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "111122223333"
        }
      }
    }
  ]
}
```

Sie können auch bestimmten Abonnenten Amazon ECR-Berechtigungen verweigern, wie in der folgenden Beispielrichtlinie gezeigt.

JSON

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

```
    "Statement": [
        {
            "Sid": "OmicsAccessPrincipal",
            "Effect": "Allow",
            "Principal": {
                "Service": "omics.amazonaws.com"
            },
            "Action": [
                "ecr:GetDownloadUrlForLayer",
                "ecr:BatchGetImage",
                "ecr:BatchCheckLayerAvailability"
            ],
            "Resource": "*",
            "Condition": {
              "StringNotEquals": {
                "aws:SourceAccount": "111122223333"
              }
            }
        }
    ]
}
```

Richtlinien für Amazon ECR ziehen den Cache durch

Um Amazon ECR Pull Through Cache zu verwenden, erstellen Sie eine Registrierungsberechtigungsrichtlinie. Sie erstellen auch eine Vorlage zur Repository-Erstellung, die die Berechtigungen für die Repositorys definiert, die von Amazon ECR Pull Through Cache erstellt wurden.

Die folgenden Abschnitte enthalten Beispiele für diese Richtlinien. Weitere Informationen zum Pull-Through-Cache finden Sie unter Synchronisieren Sie eine Upstream-Registrierung mit einer privaten Amazon ECR-Registrierung in der Amazon Elastic Container Registry — Benutzerhandbuch.

Richtlinien zur Genehmigung der Registrierung

Um Amazon ECR Pull Through Cache zu verwenden, erstellen Sie eine Registrierungsberechtigungsrichtlinie. Die Richtlinie für Registrierungsberechtigungen bietet die Kontrolle über Replikations- und Pull-Through-Cache-Berechtigungen.

Für die kontoübergreifende Replikation müssen Sie jede Replikation explizit zulassen AWS-Konto das seine Repositorys in Ihre Registrierung replizieren kann.

Wenn Sie eine Pull-Through-Cache-Regel erstellen, kann standardmäßig jeder IAM-Principal, der die Berechtigung hat, Bilder aus einer privaten Registrierung abzurufen, auch die Pull-Through-Cache-Regel verwenden. Sie können mit Registrierungsberechtigungen diese Berechtigungen weiter auf bestimmte Repositorys einschränken.

Fügen Sie dem Konto, dem das Container-Image gehört, eine Registrierungsberechtigungsrichtlinie hinzu.

Im folgenden Beispiel ermöglicht die Richtlinie HealthOmics Dienst, um Repositorys für jede Upstream-Registry zu erstellen und Upstream-Pull-Requests von den erstellten Repositorys zu initiieren.

JSON

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

Vorlage zur Repository-Erstellung

Um den Pull-Through-Cache zu verwenden in HealthOmics, das Amazon ECR-Repository muss über eine Vorlage zur Repository-Erstellung verfügen. Die Vorlage definiert Konfigurationseinstellungen für die privaten Repositorys, die für eine Upstream-Registrierung erstellt wurden.

Jede Vorlage enthält ein Repository-Namespace-Präfix, das Amazon ECR verwendet, um neue Repositorys einer bestimmten Vorlage zuzuordnen. In Vorlagen kann die Konfiguration für alle Repository-Einstellungen festgelegt werden, einschließlich ressourcenbasierter Zugriffsrichtlinien, Unveränderlichkeit von Tags, Verschlüsselung und Lebenszyklusrichtlinien. Weitere Informationen finden Sie unter Vorlagen zur Repository-Erstellung in der Amazon Elastic Container Registry — Benutzerhandbuch.

Im folgenden Beispiel ermöglicht die Richtlinie HealthOmics Dienst, um Upstream-Pull-Requests von den Upstream-Repositorys zu initiieren.

JSON

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

Richtlinien für den kontoübergreifenden Amazon ECR-Zugriff

Für den kontoübergreifenden Zugriff aktualisiert der Eigentümer des privaten Repositorys die Registrierungsberechtigungsrichtlinie und die Vorlage zur Repository-Erstellung, um den Zugriff für das andere Konto und die Ausführungsrolle dieses Kontos zu ermöglichen.

Fügen Sie der Registrierungsberechtigungsrichtlinie eine Grundsatzerklärung hinzu, um der Ausführungsrolle des anderen Kontos den Zugriff auf die Amazon ECR-Aktionen zu ermöglichen:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowCrossAccountPTCinRegPermissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::123456789012:role/RUN_ROLE"},
            "Action": [
                "ecr:CreateRepository",
                "ecr:BatchGetImage",
                "ecr:BatchImportUpstreamImage"
            ],
            "Resource": "arn:aws:ecr:us-east-1:123456789012:repository/path/*"
        }
    ]
}
```

Fügen Sie in der Vorlage zur Repository-Erstellung eine Richtlinienanweisung hinzu, damit die Ausführungsrolle des anderen Kontos auf die neuen Container-Images zugreifen kann. Optional können Sie Bedingungsanweisungen hinzufügen, um den Zugriff auf bestimmte Workflows einzuschränken:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowCrossAccountPTCinRepoCreationTemplate",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:role/RUN_ROLE"},
            "Action": [
                "ecr:BatchGetImage",
                "ecr:GetDownloadUrlForLayer"
            ],
            "Resource": "*",
```

```
            "Condition": {
                "StringEquals": {
                    "aws:SourceArn": "arn:aws:omics:us-
east-1:444455556666:workflow/WORKFLOW_ID",
                    "aws:SourceAccount": "111122223333"
                }
            }
        }
    ]
}
```

Fügen Sie Berechtigungen für zwei weitere Aktionen hinzu (CreateRepository und BatchImportUpstreamImage) in der Run-Rolle und geben Sie die Ressource an, auf die die Run-Rolle zugreifen kann.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "CrossAccountPTCRunRolePolicy",
            "Effect": "Allow",
            "Action": [
                "ecr:CreateRepository",
                "ecr:BatchImportUpstreamImage",
                "ecr:BatchCheckLayerAvailability",
                "ecr:BatchGetImage",
                "ecr:GetDownloadUrlForLayer",
                "ecr:BatchGetImage"
            ],
            "Resource": "arn:aws:ecr:us-east-1:123456789012::repository/{path}/*"
        }
    ]
}
```

Richtlinie zur Dienstrolle für die Unterstützung von Containern, die nicht von PTC stammen

Mit Unterstützung für Nicht-PTC-Container HealthOmics verwendet die Servicerolle, an die Sie übergeben `StartRun` um Bilder in Ihre Amazon ECR-Repositorys zu kopieren. Zusätzlich zu den Berechtigungen in Servicerollen für AWS HealthOmics, die Servicerolle benötigt folgende Berechtigungen:

- Erstellen Sie das Ziel-Repository, falls es nicht existiert (`ecr:CreateRepository`).
- Suchen Sie nach einer Repository-Richtlinie und hängen Sie eine an, falls das Repository keine hat (`ecr:GetRepositoryPolicy`, `ecr:SetRepositoryPolicy`).
- Drücken Sie das Bild (`ecr:InitiateLayerUpload`, `ecr:UploadLayerPart`, `ecr:CompleteLayerUpload`, `ecr:PutImage`).
- Lesen Sie die Registrierungsinformationen (`secretsmanager:GetSecretValue`).

Mit der folgenden Beispielrichtlinie werden diese Berechtigungen verliehen. Ersetzen Sie die Region, die Konto-ID, *`ecr-repository-prefix`*, und die geheime ARN mit Ihren Werten. Benutze den `ecrRepositoryPrefix` aus Ihrem Registrierungs-Mapping.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "EcrCopiedImages",
      "Effect": "Allow",
      "Action": [
        "ecr:BatchGetImage",
        "ecr:GetDownloadUrlForLayer",
        "ecr:BatchCheckLayerAvailability",
        "ecr:InitiateLayerUpload",
        "ecr:UploadLayerPart",
        "ecr:CompleteLayerUpload",
        "ecr:PutImage",
        "ecr:DescribeImages",
        "ecr:CreateRepository",
        "ecr:GetRepositoryPolicy",
        "ecr:SetRepositoryPolicy"
      ],
```

```
      "Resource": "arn:aws:ecr:us-east-1:111122223333:repository/ecr-repository-prefix/
*"
    },
    {
      "Sid": "EcrAuthToken",
      "Effect": "Allow",
      "Action": "ecr:GetAuthorizationToken",
      "Resource": "*"
    },
    {
      "Sid": "ReadRegistryCredential",
      "Effect": "Allow",
      "Action": "secretsmanager:GetSecretValue",
      "Resource": "arn:aws:secretsmanager:us-east-1:111122223333:secret:omics-registry/
seqera-EXAMPLE"
    }
  ]
}
```

- `ecr:GetAuthorizationToken` unterstützt keine Berechtigungen auf Ressourcenebene, daher bleibt es aktiviert `*`.
- Wenn Sie mehr als eine verwenden `ecrRepositoryPrefix`, oder mehrere Nicht-PTC-Registry, fügen Sie jeden Repository-ARN und jeden geheimen ARN hinzu.
- Wenn Sie Folgendes einstellen `ecrAccountId` Verwenden Sie in der Registrierungszuordnung die ID dieses Kontos im ARN des Amazon ECR-Repositorys.
- Wenn sich Ihr Secret in einem anderen Konto befindet, verwenden Sie den geheimen ARN dieses Kontos in der `secretsmanager:GetSecretValue` Ressource.
- Wenn Ihr Geheimnis mit einem vom Kunden verwalteten verschlüsselt ist AWS KMS Schlüssel, auch zulassen `kms:Decrypt` auf diesem Schlüssel.

Ich verifiziere das HealthOmics kann ein Container-Image abrufen

Folgen Sie diesem Verfahren, bevor Sie einen Lauf starten, um zu bestätigen, dass ein Container-Image in Amazon ECR verfügbar ist und HealthOmics kann es ziehen. HealthOmics ruft das Bild ab, während der Lauf läuft. Daher tritt ein Berechtigungsproblem normalerweise dann auf, und nicht beim Erstellen des Workflows oder Aufrufs `StartRun`.

Das Repositorium muss sich im selben befinden AWS Region, in der sich der Lauf befindet, unabhängig davon, ob er sich im Konto des Laufs oder in einem anderen Konto befindet. Führen

Sie jeden Befehl für diese Region aus und ersetzen Sie jeden Platzhalter durch Ihren eigenen Wert. Die Befehle lesen Ihre Amazon ECR-Konfiguration, daher benötigen Sie Ihre eigenen IAM-Berechtigungen `ecr:GetRepositoryPolicy` und `ecr:DescribeImages`.

1. Finden Sie Ihren Fall in der folgenden Tabelle, in der aufgeführt ist, wie ein Image einen Lauf erreicht, wo der Zugriff darauf gewährt wird und welcher Principal es abruft.

Wie erreicht das Bild den Lauf	Wo kann der Zugriff gewährt werden	Schulleiter, der zieht
Das Repository befindet sich im Konto des Laufs, und Sie pushen das Image	Die Repository-Richtlinie, wie in Erstellen Sie eine Ressourcenrichtlinie für das Amazon ECR-Repository	Die HealthOmics Dienstleiter, `omics.amazonaws.com`
Das Repository befindet sich in einem anderen Konto	Die Repository-Richtlinie dieses Kontos und die Servicerolle im Konto des Laufs, wie in Ausführen von Workflows mit kontoüber greifenden Containern	Das HealthOmics Service Principal und die Servicerolle, an die Sie übergehen `StartRun`
Der Amazon ECR-Pull-Through-Cache füllt das Repository	Die Vorlage zur Repository-Erstellung, die die Richtlinien für jedes Repositorium enthält, das der Cache erstellt, und die Registrierungsrichtlinie, die Folgendes ermöglicht HealthOmics erstellen Sie diese Repositorys und importieren Sie Bilder, beide wie folgt Richtlinien für Amazon ECR ziehen den Cache durch	Die HealthOmics Service Principal und für einen kontoübergreifenden Cache die Service-Rolle, an die Sie übergeben `StartRun`

Wie erreicht das Bild den Lauf	Wo kann der Zugriff gewährt werden	Schulleiter, der zieht
HealthOmics kopiert das Image, unterstützt aber Container, die nicht von PTC unterstützt werden	Die Servicerolle, die das Image kopiert, wie in Richtlinie zur Dienstrolle für die Unterstützung von Containern, die nicht von PTC stammenund die Ziel-Repository-Richtlinie. Wenn Sie einstellen `ecrAccountId` , auch die Registrierungs- und Repository-Richtlinien dieses Kontos	Die Servicerolle für die Kopie und HealthOmics Dienstprinzipal für den darauf folgenden Abruf

Ein Workflow, den Sie gemeinsam nutzen, ist die Ausnahme: Die Repository-Richtlinie benötigt keine Aussage für das Konto des Abonnenten, wie unter beschrieben Amazon ECR-Richtlinien für gemeinsame Workflows.

2. Bestätigen Sie, dass das Bild verfügbar ist. Der Repository-Name enthält alle Präfixe, die durch eine Pull-Through-Cache-Regel oder eine Registrierungszuordnung hinzugefügt werden, sowie alle Umbenennungen aus einer Image-Zuordnung.

```
aws ecr describe-images \
     --repository-name repository-name \
     --image-ids imageTag=tag \
     --region region
```

Um das Bild nach Digest statt nach Tag zu benennen, verwenden Sie `--image-ids imageDigest=sha256:`*`hex-digest`*, wo *`hex-digest`* ist der Teil danach `sha256:`. Benutze den Digest, wenn der Lauf eins anheftet, weil ein Tag auf ein anderes Bild verschoben werden kann.

Für ein Repository in einem anderen Konto füge hinzu `--registry-id` *`account-id`* zu diesem Befehl, sodass er die Registrierung dieses Kontos liest und nicht Ihre eigene. Die Repository-Richtlinie dieses Kontos muss Ihnen auch Folgendes gewähren `ecr:DescribeImages`.

Wenn das Bild oder das Repository nicht da ist, entscheide, ob das erwartet wird. Im Fall von Pull-Through-Cache und ohne PTC HealthOmics füllt das Repository während einer Ausführung, also vor dem ersten Lauf, der das Image benötigt:

- Beim Amazon ECR-Pull-Through-Cache erstellt Amazon ECR das Repository und importiert das Image beim ersten Durchlauf durch den Cache aus der Upstream-Registrierung. Ein Tag, das der Cache noch nicht importiert hat, wird auf die gleiche Weise importiert, auch wenn das Repository bereits existiert. Wenn das Image fehlt, überprüfen Sie die Registrierungsrichtlinie und die Vorlage zur Repository-Erstellung unter Richtlinien für Amazon ECR ziehen den Cache durch stattdessen.
- Mit Unterstützung für Nicht-PTC-Container HealthOmics kopiert das Image während der Ausführung. Prüfen Sie daher das Upstream-Image und die Berechtigungen für die Servicerolle in Richtlinie zur Dienstrolle für die Unterstützung von Containern, die nicht von PTC stammen, und bestätigen Sie dann nach der Ausführung die Kopie an ihrem Ziel-URI, wie unter beschrieben Benennung des Ziel-Repositories.

3. Lesen Sie die Richtlinien des Repositorys, das das Bild enthält.

```
aws ecr get-repository-policy \
     --repository-name repository-name \
     --region region
```

Fügen Sie für ein Repository in einem anderen Konto hinzu `--registry-id` *`account-id`*, was ebenfalls voraussetzt, dass die Repository-Richtlinie dieses Kontos Folgendes zulässt `ecr:GetRepositoryPolicy`. Ein Zugriffsverweigerungsfehler aus diesem Befehl bezieht sich auf Ihren eigenen Zugriff, nicht auf den Zugriff, der HealthOmics hat. Die Antwort und die Fehler des Befehls finden Sie unter GetRepositoryPolicy in der API-Referenz für Amazon Elastic Container Registry.

4. Bestätigen Sie, dass die Richtlinie eine `Allow` Anweisung, die das Repository abdeckt, den Principal für Ihren Fall benennt und die Aktionen gewährt, die ein Pull verwendet: `ecr:BatchGetImage`, `ecr:GetDownloadUrlForLayer`, und `ecr:BatchCheckLayerAvailability`. Gewähren Sie alle drei in einer von Ihnen verfassten Richtlinie. Eine Richtlinie, die von einem Dienst erstellt wurde, kann weniger gewähren und trotzdem funktionieren. Ein Platzhalter, der sie abdeckt, funktioniert ebenfalls.

   Die folgende JSON-Richtlinienanweisung besteht diesen Schritt. Ersetzen Sie jeden Platzhalter durch Ihren eigenen Wert, wobei *`run-account-id`* ist das Konto, in dem sich der Lauf befindet:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "HealthOmicsPull",
      "Effect": "Allow",
      "Principal": {
        "Service": "omics.amazonaws.com"
      },
      "Action": [
        "ecr:BatchGetImage",
        "ecr:GetDownloadUrlForLayer",
        "ecr:BatchCheckLayerAvailability"
      ],
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "run-account-id",
          "aws:SourceArn": "arn:aws:omics:region:run-account-id:workflow/workflow-
id"
        }
      }
    }
  ]
}
```

Wenn sich das Repository im Konto des Laufs befindet, kann eine Rolle dort stattdessen durch eine eigene Identitätsrichtlinie zugelassen werden, sodass die Repository-Richtlinie diese Rolle nicht benennen muss. Ein Repository in einem anderen Konto muss der Rolle einen Namen geben, wie in Ausführen von Workflows mit kontoübergreifenden Containern. Nur die Repository-Richtlinie kann den Service Principal zulassen.

Dann kreuzen Sie alle an `Condition` zu dieser Aussage gegen die Werte HealthOmics sendet. `aws:SourceArn` ist der ARN des Workflows, den der Lauf verwendet, und `aws:SourceAccount` ist das Konto, in dem sich der Lauf befindet. Eine Bedingung, die auf die Ausführung von ARNs statt Workflow-ARNs beschränkt ist, blockiert jeden Abruf.

Eine explizite `Deny` das entspricht dem Principal, blockiert den Pull, was auch immer `Allow` Kontoauszüge gewähren. Für wie AWS evaluiert Richtlinien, siehe Stellt fest, ob eine Anfrage zulässig oder abgelehnt ist in der IAM-Benutzerhandbuch.

Für ein Repository, das Amazon ECR per Pull-Through-Cache erstellt hat, gewährt die Richtlinie normalerweise nur `ecr:BatchGetImage` und `ecr:GetDownloadUrlForLayer`, aus der Vorlage zur Repository-Erstellung in Vorlage zur Repository-Erstellung. Amazon ECR wendet die Vorlage nur an, wenn das Repository erstellt wird, sodass eine Vorlage, die später geändert wird, die Richtlinie nicht aktualisiert. Beurteilen Sie die eigene Richtlinie des Repositorys, nicht die Vorlage. Diese beiden Aktionen reichen für einen Abruf aus, sodass eine solche Richtlinie diesen Schritt bestanden hat.

Wenn das Repository überhaupt keine Richtlinie hat, HealthOmics kann nicht daraus ableiten, es sei denn HealthOmics kopiert das Bild in: HealthOmics hängt eine Richtlinie an ein Repository an, das keine hat, und lässt eine bestehende Richtlinie unverändert, wie unter beschrieben Benennung des Ziel-Repositories.

5. Wenn in Ihrem Fall eine Servicerolle benannt ist, bestätigen Sie auch die Identitätsseite. Die eigene Richtlinie der Rolle muss dieselben Aktionen im Repository zulassen, wie in Richtlinie zur Dienstrolle für die Unterstützung von Containern, die nicht von PTC stammen für ein Bild, das HealthOmics Kopien, oder Ausführen von Workflows mit kontoübergreifenden Containern für ein Repository in einem anderen Konto. Wenn Sie eine Sitzungsrichtlinie übergeben an `StartRun`, die Berechtigungen der Ausführung sind der Schnittpunkt zwischen der Rolle und dieser Richtlinie, daher muss die Sitzungsrichtlinie auch diese Aktionen zulassen, wie unter beschrieben Verwenden Sie Sitzungsrichtlinien, um Berechtigungen einzuschränken.
6. Wenn jeder Scheck erfolgreich ist, können die Richtlinien es dem Schulleiter Ihres Falls ermöglichen, das Bild abzurufen. Sie beweisen nicht, dass das so ist, weil nur ein Lauf sie mit den Werten trainiert, die HealthOmics sendet. Startet einen Lauf, der das Bild verwendet. Bleibt ein Pull-Problem bestehen, schlägt der Lauf fehl mit ECR_PERMISSION_ERROR bevor eine Aufgabe erstellt wird, beginnen Sie also mit der Fehlerursache des Laufs. Wann HealthOmics kopiert das Bild, ein Kopierfehler schlägt auch bei der Aufgabe fehl. Prüfen Sie daher auch die Fehlerursache der Aufgabe. Gehen Sie Ihren Fall in jedem Fall erneut durch. Den Fehler finden Sie unter Gründe für Ausführungsfehler.

HealthOmics Berechtigungen für Ressourcen

AWS HealthOmics erstellt in Ihrem Namen Ressourcen in anderen Diensten und greift auf diese zu, wenn Sie einen Job ausführen oder einen Shop erstellen. In einigen Fällen müssen Sie Berechtigungen in anderen Diensten konfigurieren, um auf Ressourcen zuzugreifen oder den Zugriff darauf HealthOmics zu ermöglichen.

Informationen zu Ressourcenberechtigungen im Zusammenhang mit Amazon ECR finden Sie unterAmazon-ECR-Berechtigungen.

Lake-Formation-Berechtigungen

Bevor Sie Analysefunktionen in verwenden HealthOmics, konfigurieren Sie die Standard-Datenbankeinstellungen in Lake Formation.

So konfigurieren Sie Ressourcenberechtigungen in Lake Formation

1. Öffnen Sie die Seite mit den Datenkatalogeinstellungen in der Lake Formation Formation-Konsole.
2. Deaktivieren Sie die IAM-Zugriffskontrollanforderungen für Datenbanken und Tabellen unter Standardberechtigungen für neu erstellte Datenbanken und Tabellen.
3. Wählen Sie Speichern.

HealthOmics Analytics akzeptiert auto Daten, wenn Ihre Servicerichtlinie über die richtigen RAM-Berechtigungen verfügt, wie im folgenden Beispiel.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "omics:*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ram:AcceptResourceShareInvitation",
        "ram:GetResourceShareInvitations"
      ],
      "Resource": "*"
    }
  ]
```

```
}
```

Berechtigungen für den Datenzugriff mit Amazon S3 URIs

Sie können mithilfe von HealthOmics API-Operationen oder Amazon S3 S3-API-Vorgängen auf Sequenzspeicherdaten zugreifen.

Für den HealthOmics API-Zugriff werden die HealthOmics Berechtigungen über eine IAM-Richtlinie verwaltet. Für S3 Access sind jedoch zwei Konfigurationsebenen erforderlich: die ausdrückliche Genehmigung in der S3-Zugriffsrichtlinie des Stores und eine IAM-Richtlinie. Weitere Informationen zur Verwendung von IAM-Richtlinien mit HealthOmics finden Sie unter Servicerollen für. HealthOmics

Es gibt drei Möglichkeiten, die Fähigkeit, Objekte mit Amazon S3 zu lesen, gemeinsam zu nutzen APIs:

1. Richtlinienbasiertes Teilen — Für diese gemeinsame Nutzung muss der IAM-Prinzipal sowohl in der S3-Zugriffsrichtlinie aktiviert als auch eine IAM-Richtlinie geschrieben und an den IAM-Prinzipal angehängt werden. Weitere Informationen finden Sie im nächsten Thema.
2. Vorsigniert URLs — Sie können auch eine gemeinsam nutzbare vorsignierte URL für eine Datei im Sequenzspeicher generieren. Weitere Informationen zum Erstellen von Presigned URLs mit Amazon S3 finden Sie unter Using presigned URLs in der Amazon S3 S3-Dokumentation. Die S3-Zugriffsrichtlinie für Sequence Store unterstützt Anweisungen zur Einschränkung der Funktionen vorsignierter URLs.
3. Angenommene Rollen — Erstellen Sie eine Rolle innerhalb des Kontos des Datenbesitzers, die über eine Zugriffsrichtlinie verfügt, die es Benutzern ermöglicht, diese Rolle anzunehmen.

Themen

- Auf Richtlinien basierendes Teilen
- Beispiel für eine Einschränkung

Auf Richtlinien basierendes Teilen

Wenn Sie über eine direkte S3-URI auf Sequenzspeicherdaten zugreifen, HealthOmics werden erweiterte Sicherheitsmaßnahmen für die zugehörige S3-Bucket-Zugriffsrichtlinie bereitgestellt.

Die folgenden Regeln gelten für neue S3-Zugriffsrichtlinien. Für bestehende Richtlinien gelten die Regeln, wenn Sie die Richtlinie das nächste Mal aktualisieren:

- Die S3-Zugriffsrichtlinien unterstützen die folgenden Richtlinienelemente
  - Version, ID, Aussage, Sid, Wirkung, Prinzip, Aktion, Ressource, Zustand
- Die S3-Zugriffsrichtlinien unterstützen die folgenden Bedingungsschlüssel:
  - s3:ExistingObjectTag/<key>, s3: prefix, s3: signatureversion, s3: TlsVersion
  - Richtlinien unterstützen auch aws: PrincipalArn mit den folgenden Bedingungsoperatoren: und ArnEquals ArnLike

Wenn Sie versuchen, eine Richtlinie hinzuzufügen oder zu aktualisieren, sodass sie ein Element oder eine Bedingung enthält, die nicht unterstützt wird, lehnt das System die Anfrage ab.

Themen

- Standardmäßige S3-Zugriffsrichtlinie
- Anpassen der Zugriffsrichtlinie
- IAM-Richtlinie
- Tag-basierte Zugriffskontrolle

Standardmäßige S3-Zugriffsrichtlinie

Wenn Sie einen Sequenzspeicher erstellen, HealthOmics wird eine standardmäßige S3-Zugriffsrichtlinie erstellt, die dem Root-Konto des Datenspeicher-Besitzers die folgenden Berechtigungen für alle zugänglichen Objekte im Sequenzspeicher gewährt: S3: GetObjectGetObjectTagging, S3 und S3:ListBucket. Die erstellte Standardrichtlinie lautet:

JSON

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

```
                "AWS": "arn:aws:iam::111111111111:root"
            },
            "Action":
            [
                "s3:GetObject",
                "s3:GetObjectTagging"
            ],
            "Resource": "arn:aws:s3:us-
west-2:222222222222:accesspoint/111111111111-1234567890/object/111111111111/
sequenceStore/1234567890/*"
        },
        {
            "Effect": "Allow",
            "Principal":
            {
                "AWS": "arn:aws:iam::111111111111:root"
            },
            "Action": "s3:ListBucket",
            "Resource": "arn:aws:s3:us-
west-2:222222222222:accesspoint/111111111111-1234567890/111111111111/
sequenceStore/1234567890/*"
        }
    ]
}
```

Anpassen der Zugriffsrichtlinie

Wenn die S3-Zugriffsrichtlinie leer ist, ist kein S3-Zugriff zulässig. Wenn es eine bestehende Richtlinie gibt und Sie den S3-Zugriff entfernen müssen, verwenden Sie diese Option, `deleteS3AccessPolicy` um den gesamten Zugriff zu entfernen.

Um Einschränkungen für das Teilen hinzuzufügen oder anderen Konten Zugriff zu gewähren, können Sie die Richtlinie mithilfe der `PutS3AccessPolicy` API aktualisieren. Aktualisierungen der Richtlinie dürfen nicht über das Präfix für den Sequenzspeicher oder die angegebenen Aktionen hinausgehen.

IAM-Richtlinie

Um einem Benutzer oder IAM-Prinzipal mithilfe von Amazon S3 Zugriff zu gewähren APIs, muss zusätzlich zu den Berechtigungen in der S3-Zugriffsrichtlinie eine IAM-Richtlinie erstellt und an den Principal angehängt werden, um Zugriff zu gewähren. Eine Richtlinie, die den Zugriff auf die Amazon S3 S3-API ermöglicht, kann auf Sequenzspeicher-Ebene oder auf Leseset-Ebene angewendet

werden. Auf der Read-Set-Ebene können die Berechtigungen entweder durch das Präfix oder mithilfe von Ressourcen-Tag-Filtern für Proben- oder Betreff-ID-Muster eingeschränkt werden.

Wenn der Sequenzspeicher einen vom Kunden verwalteten Schlüssel (CMK) verwendet, muss der Principal auch über die Rechte verfügen, den KMS-Schlüssel für die Entschlüsselung zu verwenden. Weitere Informationen finden Sie unter Kontoübergreifender KMS-Zugriff im Entwicklerhandbuch. AWS Key Management Service

Das folgende Beispiel gewährt einem Benutzer Zugriff auf einen Sequenzspeicher. Sie können den Zugriff mit zusätzlichen Bedingungen oder ressourcenbasierten Filtern verfeinern.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111111111111:root"
      },
      "Action":
      [
         "s3:GetObject",
         "s3:GetObjectTagging"
      ],
      "Resource": "arn:aws:s3:us-
west-2:222222222222:accesspoint/111111111111-1234567890/object/111111111111/
sequenceStore/1234567890/*",
      "Condition": {
        "StringEquals": {
          "s3:ExistingObjectTag/omics:readSetStatus": "ACTIVE"
        }
      }
    },
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111111111111:root"
      },
      "Action": "s3:ListBucket",
```

```
      "Resource": "arn:aws:s3:us-
west-2:222222222222:accesspoint/111111111111-1234567890",
      "Condition": {
        "StringLike": {
          "s3:prefix": "111111111111/sequenceStore/1234567890/*"
        }
      }
    }
  ]
}
```

Tag-basierte Zugriffskontrolle

Um die tagbasierte Zugriffskontrolle verwenden zu können, muss der Sequenzspeicher zunächst aktualisiert werden, um die zu verwendenden Tagschlüssel weiterzugeben. Diese Konfiguration wird bei der Erstellung oder Aktualisierung des Sequenzspeichers festgelegt. Sobald die Tags weitergegeben wurden, können Tag-Bedingungen verwendet werden, um weitere Einschränkungen hinzuzufügen. Die Einschränkungen können in der S3-Zugriffsrichtlinie oder in der IAM-Richtlinie festgelegt werden. Im Folgenden finden Sie ein Beispiel für eine tabulatorbasierte S3-Zugriffsrichtlinie, die festgelegt werden würde:

```
{
    "Sid": "tagRestrictedGets",
    "Effect": "Allow",
    "Principal":
    {
        "AWS": "arn:aws:iam::<target_restricted_account_id>:root"
    },
    "Action":
    [
        "s3:GetObject",
        "s3:GetObjectTagging"
    ],
    "Resource": "arn:aws:s3:us-west-2:222222222222:accesspoint/111111111111-1234567890/
object/111111111111/sequenceStore/1234567890/*",
    "Condition":
    {
        "StringEquals":
        {
            "s3:ExistingObjectTag/tagKey1": "tagValue1",
            "s3:ExistingObjectTag/tagKey2": "tagValue2"
```

```
        }
    }
}
```

Beispiel für eine Einschränkung

Szenario: Erstellung einer Freigabe, bei der der Dateneigentümer die Möglichkeit eines Benutzers einschränken kann, „zurückgezogene“ Daten herunterzuladen.

In diesem Szenario verwaltete ein Datenbesitzer (Konto #111111111111) einen Datenspeicher. Dieser Dateneigentümer gibt die Daten an eine Vielzahl von Drittbenutzern weiter, darunter auch an einen Forscher (Konto #999999999999). Im Rahmen der Datenverwaltung erhält der Dateneigentümer regelmäßig Anfragen zum Widerruf der Daten eines Teilnehmers. Um diesen Widerruf zu verwalten, schränkt der Dateneigentümer bei Erhalt der Anfrage zunächst den direkten Download-Zugriff ein und löscht die Daten schließlich gemäß seinen Anforderungen.

Um diesem Bedarf gerecht zu werden, richtet der Dateneigentümer einen Sequenzspeicher ein und jeder Lesesatz erhält eine Markierung für „Status“, die auf „zurückgezogen“ gesetzt wird, wenn die Auszahlungsanforderung eingeht. Bei Daten, bei denen das Tag auf diesen Wert gesetzt ist, soll sichergestellt werden, dass kein Benutzer „GetObject“ für diese Datei ausführen kann. Um dieses Setup durchzuführen, muss der Dateneigentümer sicherstellen, dass zwei Schritte unternommen werden.

Schritt 1. Stellen Sie für den Sequenzspeicher sicher, dass das Status-Tag aktualisiert wird, damit es weitergegeben werden kann. Dies erfolgt durch Hinzufügen des Schlüssels „Status“ in die Befehlszeile `propogatedSetLevelTags` beim Aufrufen oder `createSequenceStore` `updateSequenceStore.`

Schritt 2. Aktualisieren Sie die S3-Zugriffsrichtlinie des Stores, um GetObject auf Objekte zu beschränken, deren Status-Tag auf „zurückgezogen“ gesetzt ist. Dies geschieht, indem die Zugriffsrichtlinie des Shops mithilfe der `PutS3AccesPolicy` API aktualisiert wird. Die folgende Richtlinie würde es Kunden ermöglichen, die zurückgezogenen Dateien weiterhin zu sehen, wenn sie Objekte anbieten, sie aber daran hindern, darauf zuzugreifen:

- Aussage 1 (restrictedGetWithdrawal): Das Konto 999999999999 kann keine zurückgezogenen Objekte abrufen.
- Aussage 2 (ownerGetAll): Konto 111111111111, der Datenbesitzer, kann alle Objekte abrufen, auch Objekte, die zurückgezogen wurden.

- Aussage 3 (everyoneListAll): Alle gemeinsam genutzten Konten, 111111111111 und 999999999999, können den Vorgang für das gesamte Präfix ausführen. ListBucket

JSON

```
{
    "Version":"2012-10-17",
    "Statement":
    [
        {
            "Sid": "restrictedGetWithdrawal",
            "Effect": "Allow",
            "Principal":
            {
                "AWS": "arn:aws:iam::999999999999:root"
            },
            "Action":
            [
                "s3:GetObject",
                "s3:GetObjectTagging"
            ],
            "Resource": "arn:aws:s3:us-
west-2:222222222222:accesspoint/111111111111-1234567890/object/111111111111/
sequenceStore/1234567890/*",
            "Condition":
            {
                "StringNotEquals":
                {
                    "s3:ExistingObjectTag/status": "withdrawn"
                }
            }
        },
        {
            "Sid": "ownerGetAll",
            "Effect": "Allow",
            "Principal":
            {
                "AWS": "arn:aws:iam::111111111111:root"
            },
            "Action":
            [
                "s3:GetObject",
```

```
                "s3:GetObjectTagging"
            ],
            "Resource": "arn:aws:s3:us-
west-2:222222222222:accesspoint/111111111111-1234567890/object/111111111111/
sequenceStore/1234567890/*",
            "Condition":
            {
                "StringEquals":
                {
                    "s3:ExistingObjectTag/omics:readSetStatus": "ACTIVE"
                }
            }
        },
        {
            "Sid": "everyoneListAll",
            "Effect": "Allow",
            "Principal":
            {
                "AWS": [
                    "arn:aws:iam::111111111111:root",
                    "arn:aws:iam::999999999999:root"
                ]
            },
            "Action": "s3:ListBucket",
            "Resource": "arn:aws:s3:us-
west-2:222222222222:accesspoint/111111111111-1234567890",
            "Condition":
            {
                "StringLike":
                {
                    "s3:prefix": "111111111111/sequenceStore/1234567890/*"
                }
            }
        }
    ]
}
```

Sicherheit in AWS HealthOmics

Cloud-Sicherheit AWS hat höchste Priorität. Als AWS Kunde profitieren Sie von Rechenzentren und Netzwerkarchitekturen, die darauf ausgelegt sind, die Anforderungen der sicherheitssensibelsten Unternehmen zu erfüllen.

Sicherheit ist eine gemeinsame AWS Verantwortung von Ihnen und Ihnen. Das Modell der geteilten Verantwortung beschreibt dies als Sicherheit der Cloud und Sicherheit in der Cloud:

- Sicherheit der Cloud — AWS ist verantwortlich für den Schutz der Infrastruktur, auf der AWS Dienste in der ausgeführt AWS Cloud werden. AWS bietet Ihnen auch Dienste, die Sie sicher nutzen können. Externe Prüfer testen und verifizieren regelmäßig die Wirksamkeit unserer Sicherheitsmaßnahmen im Rahmen der AWS . Weitere Informationen zu den Compliance-Programmen, die für AWS gelten HealthOmics, finden Sie unter AWS Services im Bereich nach Compliance-Programm AWS .
- Sicherheit in der Cloud — Ihre Verantwortung richtet sich nach dem AWS Service, den Sie nutzen. Sie sind auch für andere Faktoren verantwortlich, etwa für die Vertraulichkeit Ihrer Daten, für die Anforderungen Ihres Unternehmens und für die geltenden Gesetze und Vorschriften.

Diese Dokumentation hilft Ihnen zu verstehen, wie Sie das Modell der gemeinsamen Verantwortung bei der Nutzung von AWS anwenden können HealthOmics. In den folgenden Themen erfahren Sie, wie Sie AWS konfigurieren HealthOmics , um Ihre Sicherheits- und Compliance-Ziele zu erreichen. Sie lernen auch, wie Sie andere AWS Services nutzen können, die Sie bei der Überwachung und Sicherung Ihrer HealthOmics AWS-Ressourcen unterstützen.

Topics

- Datenschutz in AWS HealthOmics
- Identitäts- und Zugriffsmanagement in HealthOmics
- Konformitätsvalidierung für AWS HealthOmics
- Resilienz in HealthOmics
- AWS HealthOmics und Schnittstelle VPC-Endpunkte (AWS PrivateLink)

Datenschutz in AWS HealthOmics

Das AWS Modell der gilt für den Datenschutz in AWS HealthOmics. Wie in diesem Modell beschrieben, AWS ist verantwortlich für den Schutz der globalen Infrastruktur, auf der alle Systeme laufen AWS Cloud. Sie sind dafür verantwortlich, die Kontrolle über Ihre in dieser Infrastruktur gehosteten Inhalte zu behalten. Sie sind auch für die Sicherheitskonfiguration und die Verwaltungsaufgaben für die von Ihnen verwendeten AWS-Services verantwortlich. Weitere Informationen zum Datenschutz finden Sie unter Häufig gestellte Fragen zum Datenschutz . Informationen zum Datenschutz in Europa finden Sie im General Data Protection Regulation (GDPR) Center.

Aus Datenschutzgründen empfehlen wir, dass Sie Ihre AWS-Konto Anmeldeinformationen schützen und einzelne Benutzer mit AWS IAM Identity Center oder AWS Identity and Access Management (IAM) einrichten. So erhält jeder Benutzer nur die Berechtigungen, die zum Durchführen seiner Aufgaben erforderlich sind. Außerdem empfehlen wir, die Daten mit folgenden Methoden schützen:

- Verwenden Sie für jedes Konto die Multi-Faktor-Authentifizierung (MFA).
- Wird verwendet SSL/TLS , um mit AWS Ressourcen zu kommunizieren. Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Richten Sie die API und die Protokollierung von Benutzeraktivitäten mit ein AWS CloudTrail. Informationen zur Verwendung von CloudTrail Pfaden zur Erfassung von AWS Aktivitäten finden Sie unter Arbeiten mit CloudTrail Pfaden im AWS CloudTrail Benutzerhandbuch.
- Verwenden Sie AWS Verschlüsselungslösungen zusammen mit allen darin enthaltenen Standardsicherheitskontrollen AWS-Services.
- Verwenden Sie erweiterte verwaltete Sicherheitsservices wie Amazon Macie, die dabei helfen, in Amazon S3 gespeicherte persönliche Daten zu erkennen und zu schützen.
- Wenn Sie für den Zugriff AWS über eine Befehlszeilenschnittstelle oder eine API FIPS 140-3-validierte kryptografische Module benötigen, verwenden Sie einen FIPS-Endpunkt. Weitere Informationen über verfügbare FIPS-Endpunkte finden Sie unter Federal Information Processing Standard (FIPS) 140-3.

Wir empfehlen dringend, in Freitextfeldern, z. B. im Feld Name, keine vertraulichen oder sensiblen Informationen wie die E-Mail-Adressen Ihrer Kunden einzugeben. Dies gilt auch, wenn Sie mit AWS HealthOmics oder anderen zusammenarbeiten und die Konsole AWS CLI, API oder AWS SDKs AWS-Services verwenden. Alle Daten, die Sie in Tags oder Freitextfelder eingeben, die für Namen verwendet werden, können für Abrechnungs- oder Diagnoseprotokolle verwendet

werden. Wenn Sie eine URL für einen externen Server bereitstellen, empfehlen wir dringend, keine Anmeldeinformationen zur Validierung Ihrer Anforderung an den betreffenden Server in die URL einzuschließen.

Verschlüsselung im Ruhezustand

Themen

- AWS-eigene Schlüssel
- Kundenseitig verwaltete Schlüssel
- Einen vom Kunden verwalteten Schlüssel erstellen
- Erforderliche IAM-Berechtigungen für die Verwendung eines vom Kunden verwalteten Schlüssels
- Weitere Informationen

Zum Schutz vertraulicher Kundendaten im Speicher wird standardmäßig eine Verschlüsselung mit einem serviceeigenen AWS Key Management Service (AWS KMS) -Schlüssel bereitgestellt. AWS HealthOmics Vom Kunden verwaltete Schlüssel werden ebenfalls unterstützt. Weitere Informationen zu vom Kunden verwalteten Schlüsseln finden Sie unter Amazon Key Management Service.

Alle HealthOmics Datenspeicher (Storage und Analytics) unterstützen die Verwendung von kundenverwalteten Schlüsseln. Die Verschlüsselungskonfiguration kann nicht geändert werden, nachdem ein Datenspeicher erstellt wurde. Wenn ein Datenspeicher einen verwendet AWS-eigener Schlüssel, wird er als gekennzeichnet, AWS_OWNED_KMS_KEY und der spezifische Schlüssel, der für die Verschlüsselung verwendet wird, wird im Ruhezustand nicht angezeigt.

Bei HealthOmics Workflows werden vom Kunden verwaltete Schlüssel vom temporären Dateisystem nicht unterstützt. Alle Daten im Ruhezustand werden jedoch automatisch mithilfe eines XTS-AES-256 Blockchiffrierverschlüsselungsalgorithmus verschlüsselt, um das Dateisystem zu verschlüsseln. Der IAM-Benutzer und die Rolle, die zum Starten einer Workflow-Ausführung verwendet wurden, müssen auch Zugriff auf die AWS KMS Schlüssel haben, die für Workflow-Eingabe- und -Ausgabe-Buckets verwendet werden. Workflows verwenden keine Zuschüsse, und die AWS KMS Verschlüsselung ist auf Eingabe- und Ausgabe-Amazon S3-Buckets beschränkt. Die IAM-Rolle, die sowohl für Workflow-APIs verwendet wird, muss auch Zugriff auf die verwendeten AWS KMS Schlüssel sowie auf die Eingabe- und Ausgabe-Amazon S3-Buckets haben. Sie können entweder IAM-Rollen und -Berechtigungen verwenden, um den Zugriff oder Richtlinien zu kontrollieren. AWS KMS Weitere Informationen finden Sie unter Authentifizierung und Zugriffskontrolle für AWS KMS.

Wenn Sie HealthOmics Analytics verwenden AWS Lake Formation , werden alle Entschlüsselungsberechtigungen, die mit der Lake Formation verknüpft sind, auch für die Eingabe- und Ausgabe-Amazon S3-Buckets erteilt. Weitere Informationen zur AWS Lake Formation Verwaltung von Berechtigungen finden Sie in der AWS Lake Formation Dokumentation.

HealthOmics Analytics gewährt Lake Formation KMS:Decrypt Berechtigungen zum Lesen der verschlüsselten Daten in einem Amazon S3 S3-Bucket. Solange Sie berechtigt sind, die Daten über Lake Formation abzufragen, können Sie die verschlüsselten Daten lesen. Der Zugriff auf die Daten wird durch die Datenzugriffskontrolle in Lake Formation gesteuert, nicht durch eine KMS-Schlüsselrichtlinie. Weitere Informationen finden Sie in den AWS integrierten AWS-Serviceanfragen in der Lake Formation Formation-Dokumentation.

AWS-eigene Schlüssel

Standardmäßig werden Daten im Ruhezustand automatisch verschlüsselt, da diese Daten vertrauliche Informationen wie personenbezogene Daten (PII) oder geschützte Gesundheitsinformationen (PHI) enthalten können. HealthOmics AWS-eigene Schlüssel AWS-eigene Schlüssel sind nicht in Ihrem Konto gespeichert. Sie sind Teil einer Sammlung von KMS-Schlüsseln, die AWS besitzt und verwaltet, um sie in mehreren AWS-Konten zu verwenden.

AWS-Services können AWS-eigene Schlüssel zum Schutz Ihrer Daten verwendet werden. Sie können ihre Verwendung nicht einsehen, verwalten AWS-eigene Schlüssel, darauf zugreifen oder sie überprüfen. Sie müssen jedoch keine Arbeit verrichten oder Programme ändern, um die Schlüssel zu schützen, mit denen Ihre Daten verschlüsselt werden.

Ihnen wird keine monatliche Gebühr oder Nutzungsgebühr für die Nutzung berechnet AWS-eigene Schlüssel, und sie werden auch nicht auf die AWS KMS KMS-Kontingente für Ihr Konto angerechnet. Weitere Informationen finden Sie unter Von AWS verwaltete Schlüssel.

Kundenseitig verwaltete Schlüssel

HealthOmics unterstützt die Verwendung von symmetrischen, vom Kunden verwalteten Schlüsseln, die Sie selbst erstellen, besitzen und verwalten, um der vorhandenen AWS-owned Verschlüsselung eine zweite Verschlüsselungsebene hinzuzufügen. Da Sie die volle Kontrolle über diese Verschlüsselungsebene haben, können Sie beispielsweise folgende Aufgaben ausführen:

- Einrichtung und Pflege wichtiger Richtlinien, IAM-Richtlinien und Zuschüsse
- Kryptographisches Material mit rotierendem Schlüssel
- Aktivieren und Deaktivieren wichtiger Richtlinien

- Hinzufügen von -Tags
- Erstellen von Schlüsselaliasen
- Schlüssel für das Löschen von Schlüsseln planen

Sie können es auch verwenden CloudTrail , um die Anfragen nachzuverfolgen, die in Ihrem Namen HealthOmics AWS KMS an gesendet werden. Es AWS KMS fallen zusätzliche Gebühren an. Weitere Informationen finden Sie unter Vom Kunden verwaltete Schlüssel.

Einen vom Kunden verwalteten Schlüssel erstellen

Sie können einen symmetrischen, vom Kunden verwalteten Schlüssel mithilfe der AWS-Managementkonsole oder der AWS KMS APIs erstellen.

Folgen Sie den Schritten zur Erstellung symmetrischer kundenverwalteter Schlüssel im AWS Key Management Service Developer Guide.

Schlüsselrichtlinien steuern den Zugriff auf den vom Kunden verwalteten Schlüssel. Jeder vom Kunden verwaltete Schlüssel muss über genau eine Schlüsselrichtlinie verfügen, die aussagt, wer den Schlüssel wie verwenden kann. Wenn Sie einen vom Kunden verwalteten Schlüssel erstellen, können Sie eine Schlüsselrichtlinie angeben. Weitere Informationen finden Sie unter Verwaltung des Zugriffs auf vom Kunden verwaltete Schlüssel im AWS Key Management Service Developer Guide.

Um einen vom Kunden verwalteten Schlüssel mit Ihren HealthOmics Analytics-Ressourcen zu verwenden, benötigt der aufrufende Principal die CreateGrant Operationen kms: in der Schlüsselrichtlinie. Auf diese Weise kann das System mithilfe eines FAS-Tokens einen Zuschuss für einen vom Kunden verwalteten Schlüssel gewähren, der den Zugriff auf einen bestimmten KMS-Schlüssel steuert. Dieser Schlüssel gewährt einem Benutzer Zugriff auf die erforderlichen kms:grant-Operationen. HealthOmics Weitere Informationen finden Sie unter Grants verwenden.

Für HealthOmics Analysen müssen die folgenden API-Operationen für den aufrufenden Prinzipal zulässig sein:

- kms: CreateGrant fügt einem bestimmten vom Kunden verwalteten Schlüssel Zuschüsse hinzu, wodurch der Zugriff auf Grant-Operationen in HealthOmics Analytics ermöglicht wird.
- kms: DescribeKey stellt die vom Kunden verwalteten Schlüsseldetails bereit, die zur Validierung des Schlüssels erforderlich sind. Dies ist für alle Operationen erforderlich.

- kms: GenerateDataKey bietet Zugriff auf die Verschlüsselung von Ressourcen im Ruhezustand für alle Schreibvorgänge. Außerdem stellt diese Aktion vom Kunden verwaltete Schlüsselinformationen bereit, anhand derer der Dienst überprüfen kann, ob der Anrufer Zugriff auf den Schlüssel hat.
- kms:Decrypt bietet Zugriff auf Lese- oder Suchvorgänge für verschlüsselte Ressourcen.

Um einen vom Kunden verwalteten Schlüssel mit Ihren HealthOmics Speicherressourcen zu verwenden, müssen der HealthOmics Service Principal und der Calling Principal in der Schlüsselrichtlinie zugelassen sein. Auf diese Weise kann der Service überprüfen, ob der Anrufer Zugriff auf den Schlüssel hat, und den Service Principal verwenden, um die Geschäftsverwaltung mithilfe des vom Kunden verwalteten Schlüssels auszuführen. Für die HealthOmics Speicherung muss die Schlüsselrichtlinie für den Service Principal die folgenden API-Operationen zulassen:

- kms: DescribeKey stellt die vom Kunden verwalteten Schlüsseldetails bereit, die zur Validierung des Schlüssels erforderlich sind. Dies ist für alle Operationen erforderlich.
- kms: GenerateDataKey bietet Zugriff auf die Verschlüsselung von Ressourcen im Ruhezustand für alle Schreibvorgänge. Außerdem stellt diese Aktion vom Kunden verwaltete Schlüsselinformationen bereit, anhand derer der Dienst überprüfen kann, ob der Anrufer Zugriff auf den Schlüssel hat.
- kms:Decrypt bietet Zugriff auf Lese- oder Suchvorgänge für verschlüsselte Ressourcen.

Das folgende Beispiel zeigt eine Richtlinienanweisung, die es einem Dienstprinzipal ermöglicht, eine HealthOmics Sequenz oder einen Referenzspeicher zu erstellen und mit diesem zu interagieren, der mit dem vom Kunden verwalteten Schlüssel verschlüsselt ist:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
{
    "Effect": "Allow",
    "Principal": {
        "Service": "omics.amazonaws.com"
    },
    "Action": [
        "kms:Decrypt",
        "kms:DescribeKey",
        "kms:Encrypt",
```

```
        "kms:GenerateDataKey*"
    ],
    "Resource": "*"
        }
    ]
}
```

Das folgende Beispiel zeigt eine Richtlinie, die Berechtigungen für einen Datenspeicher zum Entschlüsseln von Daten aus einem Amazon S3 S3-Bucket erstellt.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "omics:GetReference",
                "omics:GetReferenceMetadata"
            ],
            "Resource": [
                "arn:aws:omics:us-east-1:123456789012:referenceStore/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::[[s3path]]/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:123456789012:key/key_id"
```

```
            ],
            "Condition": {
              "StringEquals": {
                  "kms:ViaService": [
                    "s3.us-east-1.amazonaws.com"
                  ]
              }
            }
          }
        ]
}
```

Erforderliche IAM-Berechtigungen für die Verwendung eines vom Kunden verwalteten Schlüssels

Beim Erstellen einer Ressource wie eines Datenspeichers mit AWS KMS Verschlüsselung mithilfe eines vom Kunden verwalteten Schlüssels sind Berechtigungen sowohl für die Schlüsselrichtlinie als auch für die IAM-Richtlinie für den IAM-Benutzer oder die IAM-Rolle erforderlich.

Sie können den ViaService Bedingungsschlüssel kms: verwenden, um die Verwendung des KMS-Schlüssels auf Anfragen zu beschränken, die von stammen. HealthOmics

Weitere Informationen zu wichtigen Richtlinien finden Sie unter Enabling IAM-Policies im AWS Key Management Service Developer Guide.

Themen

- Analytics-API-Berechtigungen
- Speicher-API-Berechtigungen
- Wie HealthOmics werden Zuschüsse in AWS KMS verwendet
- Überwachen Sie Ihre Verschlüsselungsschlüssel für AWS HealthOmics

Analytics-API-Berechtigungen

Für HealthOmics Analysen muss der IAM-Benutzer oder die IAM-Rolle, die die Stores erstellt, über die Berechtigungen kms:CreateGrant, kms:GenerateDataKey, kms:Decrypt und kms: sowie über die DescribeKey erforderlichen Berechtigungen verfügen. HealthOmics

Speicher-API-Berechtigungen

Für HealthOmics Speicher-APIs benötigt der IAM-Benutzer oder die IAM-Rolle, die die folgenden API-Operationen aufruft, die aufgeführten Berechtigungen:

CreateReferenceStore, CreateSequenceStore

Um einen Store zu erstellen, muss der IAM-Aufrufer über Berechtigungen und die `kms:DescribeKey` erforderlichen Berechtigungen verfügen. HealthOmics Der HealthOmics Dienstprinzipal ruft auf`kms:GenerateDataKeyWithoutPlaintext`, um Zugriffsprüfungen für das Laden und den Zugriff auf Daten durchzuführen.

StartReadSetImportJob, StartReferenceImportJob

Um Datenimportaufträge zu starten, muss `kms:Decrypt` der IAM-Aufrufer über `kms:GenerateDataKey` Berechtigungen für den KMS-Schlüssel im Store für den Import sowie über `kms:Decrypt` Berechtigungen für den Amazon S3 S3-Bucket verfügen, der die zu importierenden Objekte enthält. Darüber hinaus muss die an den Aufruf übergebene Rolle über `kms:Decrypt` Berechtigungen für den Amazon S3 S3-Bucket verfügen, der die zu importierenden Objekte enthält. Der IAM-Aufrufer muss außerdem über die Berechtigungen verfügen, um die Rolle an den Job weiterzugeben.

CreateMultipartReadSetUpload, UploadReadSetPart, CompleteMultipartReadSetUpload

Um einen mehrteiligen Upload abzuschließen, muss `kms:Decrypt` der IAM-Aufrufer den mehrteiligen Upload erstellen, hochladen und abschließen können. `kms:GenerateDataKey`

StartReadSetExportJob

Um einen Datenexportauftrag zu starten, muss der IAM-Aufrufer über die `kms:Decrypt` Berechtigung für den KMS-Schlüssel im Store zum Exportieren aus `kms:GenerateDataKey` und über `kms:Decrypt` Berechtigungen für den Amazon S3 S3-Bucket verfügen, der die Objekte empfängt. Darüber hinaus muss die an den Aufruf übergebene Rolle über `kms:Decrypt` Berechtigungen für den Amazon S3 S3-Bucket verfügen, der die Objekte empfängt. Der IAM-Aufrufer muss außerdem über Berechtigungen verfügen, um die Rolle an den Job weiterzugeben.

StartReadsetActivationJob

Um einen Read-Set-Aktivierungsjob zu starten, muss der IAM-Aufrufer über `kms:GenerateDataKey` Berechtigungen für die `kms:Decrypt` Objekte verfügen.

GetReference, GetReadSet

Um Objekte aus dem Speicher lesen zu können, muss der IAM-Aufrufer über `kms:Decrypt` Berechtigungen für die Objekte verfügen.

Lesen Sie Set S3 GetObject

Um Objekte aus dem Store mithilfe der Amazon S3 `GetObject` S3-API lesen zu können, muss der IAM-Aufrufer über `kms:Decrypt` Berechtigungen für die Objekte verfügen. Legen Sie diese Berechtigung sowohl für vom Kunden verwaltete Schlüssel als auch für Konfigurationen fest AWS-eigener Schlüssel .

Wie HealthOmics werden Zuschüsse in AWS KMS verwendet

HealthOmics Analytics erfordert eine Genehmigung zur Nutzung Ihres vom Kunden verwalteten KMS-Schlüssels. Zuschüsse sind nicht erforderlich und werden auch nicht für HealthOmics Workflows verwendet. HealthOmics Storage verwendet den vom Kunden verwalteten Schlüssel direkt vom Service Principal. Verwenden Sie daher keine Grants. Wenn Sie einen Analyseshop erstellen, der mit einem vom Kunden verwalteten Schlüssel verschlüsselt ist, erstellt HealthOmics Analytics in Ihrem Namen einen Zuschuss, indem eine CreateGrantAnfrage an AWS KMS gesendet wird. Zuschüsse in AWS KMS werden verwendet, um HealthOmics Zugriff auf einen KMS-Schlüssel in einem Kundenkonto zu gewähren.

Es wird nicht empfohlen, die Zuschüsse, die HealthOmics Analytics in Ihrem Namen gewährt, zu widerrufen oder zurückzuziehen. Wenn Sie die HealthOmics Genehmigung zur Verwendung der AWS-KMS-Schlüssel in Ihrem Konto widerrufen oder zurückziehen, können Sie HealthOmics nicht auf diese Daten zugreifen, neue Ressourcen, die in den Datenspeicher übertragen werden, verschlüsseln oder sie entschlüsseln, wenn sie abgerufen werden.

Wenn Sie einen Zuschuss für widerrufen oder zurückziehen HealthOmics, erfolgt die Änderung sofort. Um die Zugriffsrechte zu widerrufen, empfehlen wir, den Datenspeicher zu löschen, anstatt die Gewährung zu widerrufen. Wenn Sie den Datenspeicher löschen, werden die Zuschüsse HealthOmics in Ihrem Namen zurückgezogen.

Überwachen Sie Ihre Verschlüsselungsschlüssel für AWS HealthOmics

Sie können CloudTrail damit die Anfragen verfolgen, die in Ihrem Namen AWS HealthOmics AWS KMS an gesendet werden, wenn Sie einen vom Kunden verwalteten Schlüssel verwenden. Die Protokolleinträge im CloudTrail Protokoll werden im Feld UserAgent angezeigt

HealthOmics.amazonAWS.com , sodass die Anfragen von HealthOmics eindeutig unterschieden werden können.

Bei den folgenden Beispielen handelt es sich um CloudTrail Ereignisse für CreateGrant, GenerateDataKey, Decrypt und DescribeKey zur Überwachung von AWS KMS Vorgängen, die aufgerufen werden, HealthOmics um auf Daten zuzugreifen, die mit Ihrem vom Kunden verwalteten Schlüssel verschlüsselt wurden.

Im Folgenden wird auch gezeigt, wie Sie HealthOmics Analytics CreateGrant den Zugriff auf einen vom Kunden bereitgestellten KMS-Schlüssel ermöglichen, sodass HealthOmics dieser KMS-Schlüssel zur Verschlüsselung aller gespeicherten Kundendaten verwendet werden kann.

Sie müssen keine eigenen Zuschüsse erstellen. HealthOmics erstellt in Ihrem Namen einen Zuschuss, indem Sie eine CreateGrant Anfrage an AWS KMS senden. Zuschüsse AWS KMS werden verwendet, um HealthOmics Zugriff auf einen AWS KMS Schlüssel in einem Kundenkonto zu gewähren.


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "xx:test",
        "arn": "arn:AWS:sts::555555555555:assumed-role/user-admin/test",
        "accountId": "xx",
        "accessKeyId": "xxx",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "xxxx",
                "arn": "arn:AWS:iam::555555555555:role/user-admin",
                "accountId": "555555555555",
                "userName": "user-admin"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2022-11-11T01:36:17Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedBy": "apigateway.amazonAWS.com"
    },
    "eventTime": "2022-11-11T02:34:41Z",
```

```
    "eventSource": "kms.amazonAWS.com",
    "eventName": "CreateGrant",
    "AWSRegion": "us-west-2",
    "sourceIPAddress": "apigateway.amazonAWS.com",
    "userAgent": "apigateway.amazonAWS.com",
    "requestParameters": {
        "granteePrincipal": "AWS Internal",
        "keyId": "arn:AWS:kms:us-west-2:555555555555:key/a6e87d77-cc3e-4a98-a354-
e4c275d775ef",
        "operations": [
            "CreateGrant",
            "RetireGrant",
            "Decrypt",
            "GenerateDataKey"
        ]
    },
    "responseElements": {
        "grantId": "4869b81e0e1db234342842af9f5531d692a76edaff03e94f4645d493f4620ed7",
        "keyId": "arn:AWS:kms:us-west-2:245126421963:key/xx-cc3e-4a98-a354-
e4c275d775ef"
    },
    "requestID": "d31d23d6-b6ce-41b3-bbca-6e0757f7c59a",
    "eventID": "3a746636-20ef-426b-861f-e77efc56e23c",
    "readOnly": false,
    "resources": [
        {
            "accountId": "245126421963",
            "type": "AWS::KMS::Key",
            "ARN": "arn:AWS:kms:us-west-2:245126421963:key/xx-cc3e-4a98-a354-
e4c275d775ef"
        }
    ],
    "eventType": "AWSApiCall",
    "managementEvent": true,
    "recipientAccountId": "245126421963",
    "eventCategory": "Management"
}
```

Das folgende Beispiel zeigt, wie sichergestellt werden kann GenerateDataKey , dass der Benutzer vor dem Speichern über die erforderlichen Berechtigungen zum Verschlüsseln von Daten verfügt.

```
{
```

```
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "EXAMPLEUSER",
        "arn": "arn:AWS:sts::111122223333:assumed-role/Sampleuser01",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLEKEYID",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "EXAMPLEROLE",
                "arn": "arn:AWS:iam::111122223333:role/Sampleuser01",
                "accountId": "111122223333",
                "userName": "Sampleuser01"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2021-06-30T21:17:06Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedBy": "omics.amazonAWS.com"
    },
    "eventTime": "2021-06-30T21:17:37Z",
    "eventSource": "kms.amazonAWS.com",
    "eventName": "GenerateDataKey",
    "AWSRegion": "us-east-1",
    "sourceIPAddress": "omics.amazonAWS.com",
    "userAgent": "omics.amazonAWS.com",
    "requestParameters": {
        "keySpec": "AES_256",
        "keyId": "arn:AWS:kms:us-east-1:111122223333:key/EXAMPLE_KEY_ARN"
    },
    "responseElements": null,
    "requestID": "EXAMPLE_ID_01",
    "eventID": "EXAMPLE_ID_02",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:AWS:kms:us-east-1:111122223333:key/EXAMPLE_KEY_ARN"
        }
    ],
```

```
    "eventType": "AWSApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

Weitere Informationen

Die folgenden Ressourcen bieten weitere Informationen zur Verschlüsselung von Daten im Ruhezustand.

Weitere Informationen zu den Grundkonzepten von AWS Key Management Service finden Sie in der AWS KMS Dokumentation.

Weitere Informationen zu bewährten Sicherheitsmethoden finden Sie in der AWS KMS Dokumentation.

Verschlüsselung während der Übertragung

AWS HealthOmics verwendet TLS 1.2 und höher, um Daten zu verschlüsseln, die über die öffentlichen Endpunkte und über Backend-Dienste übertragen werden.

Identitäts- und Zugriffsmanagement in HealthOmics

AWS Identity and Access Management (IAM) hilft einem Administrator AWS-Service , den Zugriff auf Ressourcen sicher zu kontrollieren. AWS IAM-Administratoren kontrollieren, wer authentifiziert (angemeldet) und autorisiert werden kann (über Berechtigungen verfügt), um HealthOmics AWS-Ressourcen zu verwenden. IAM ist ein Programm AWS-Service , das Sie ohne zusätzliche Kosten nutzen können.

Themen

- Zielgruppe
- Authentifizierung mit Identitäten
- Verwalten des Zugriffs mit Richtlinien
- Wie AWS HealthOmics funktioniert mit IAM
- Beispiele für identitätsbasierte Richtlinien für AWS HealthOmics

- AWS verwaltete Richtlinien für AWS HealthOmics
- Problembehandlung bei AWS HealthOmics Identität und Zugriff
- Verwenden von serviceverknüpften Rollen für AWS HealthOmics

Zielgruppe

Wie Sie AWS Identity and Access Management (IAM) verwenden, hängt von Ihrer Rolle ab:

- Servicebenutzer – Fordern Sie von Ihrem Administrator Berechtigungen an, wenn Sie nicht auf Features zugreifen können (siehe Problembehandlung bei AWS HealthOmics Identität und Zugriff).
- Serviceadministrator – Bestimmen Sie den Benutzerzugriff und stellen Sie Berechtigungsanfragen (siehe Wie AWS HealthOmics funktioniert mit IAM).
- IAM-Administrator – Schreiben Sie Richtlinien zur Zugriffsverwaltung (siehe Beispiele für identitätsbasierte Richtlinien für AWS HealthOmics).

Authentifizierung mit Identitäten

Authentifizierung ist die Art und Weise, wie Sie sich AWS mit Ihren Identitätsdaten anmelden. Sie müssen sich als IAM-Benutzer authentifizieren oder eine IAM-Rolle annehmen. Root-Benutzer des AWS-Kontos

Sie können sich als föderierte Identität anmelden, indem Sie Anmeldeinformationen aus einer Identitätsquelle wie AWS IAM Identity Center (IAM Identity Center), Single Sign-On-Authentifizierung oder Anmeldeinformationen verwenden. Google/Facebook Weitere Informationen zum Anmelden finden Sie unter So melden Sie sich bei Ihrem AWS-Konto an im Benutzerhandbuch für AWS-Anmeldung .

AWS Bietet für den programmatischen Zugriff ein SDK und eine CLI zum kryptografischen Signieren von Anfragen. Weitere Informationen finden Sie unter AWS Signature Version 4 for API requests im IAM-Benutzerhandbuch.

AWS-Konto Root-Benutzer

Wenn Sie einen erstellen AWS-Konto, beginnen Sie mit einer Anmeldeidentität, dem sogenannten AWS-Konto Root-Benutzer, der vollständigen Zugriff auf alle AWS-Services Ressourcen hat. Wir raten ausdrücklich davon ab, den Root-Benutzer für Alltagsaufgaben zu verwenden. Eine Liste der

Aufgaben, für die Sie sich als Root-Benutzer anmelden müssen, finden Sie unter Tasks that require root user credentials im IAM-Benutzerhandbuch.

Verbundidentität

Es hat sich bewährt, dass menschliche Benutzer für den Zugriff AWS-Services mithilfe temporärer Anmeldeinformationen einen Verbund mit einem Identitätsanbieter verwenden müssen.

Eine föderierte Identität ist ein Benutzer aus Ihrem Unternehmensverzeichnis, Ihrem Directory Service Web-Identitätsanbieter oder der AWS-Services mithilfe von Anmeldeinformationen aus einer Identitätsquelle zugreift. Verbundene Identitäten übernehmen Rollen, die temporäre Anmeldeinformationen bereitstellen.

Für die zentrale Zugriffsverwaltung empfehlen wir AWS IAM Identity Center. Weitere Informationen finden Sie unter Was ist IAM Identity Center? im AWS IAM Identity Center -Benutzerhandbuch.

IAM-Benutzer und -Gruppen

Ein IAM-Benutzer ist eine Identität mit bestimmten Berechtigungen für eine einzelne Person oder Anwendung. Wir empfehlen die Verwendung temporärer Anmeldeinformationen anstelle von IAM-Benutzern mit langfristigen Anmeldeinformationen. Weitere Informationen finden Sie im IAM-Benutzerhandbuch unter Erfordern, dass menschliche Benutzer den Verbund mit einem Identitätsanbieter verwenden müssen, um AWS mithilfe temporärer Anmeldeinformationen darauf zugreifen zu können.

Eine IAM-Gruppe spezifiziert eine Sammlung von IAM-Benutzern und erleichtert die Verwaltung von Berechtigungen für große Gruppen von Benutzern. Weitere Informationen finden Sie unter Anwendungsfälle für IAM-Benutzer im IAM-Benutzerhandbuch.

IAM-Rollen

Eine IAM-Rolle ist eine Identität mit spezifischen Berechtigungen, die temporäre Anmeldeinformationen bereitstellt. Sie können eine Rolle übernehmen, indem Sie von einer Benutzer- zu einer IAM-Rolle (Konsole) wechseln AWS CLI oder einen AWS API-Vorgang aufrufen. Weitere Informationen finden Sie unter Methoden, um eine Rolle zu übernehmen im IAM-Benutzerhandbuch.

IAM-Rollen sind nützlich für den Verbundbenutzer-Zugriff, temporäre IAM-Benutzerberechtigungen, kontoübergreifenden Zugriff, serviceübergreifenden Zugriff und Anwendungen, die auf Amazon EC2 laufen. Weitere Informationen finden Sie unter Kontoübergreifender Ressourcenzugriff in IAM im IAM-Benutzerhandbuch.

Verwalten des Zugriffs mit Richtlinien

Sie kontrollieren den Zugriff, AWS indem Sie Richtlinien erstellen und diese an AWS Identitäten oder Ressourcen anhängen. Eine Richtlinie definiert Berechtigungen, wenn sie mit einer Identität oder Ressource verknüpft sind. AWS bewertet diese Richtlinien, wenn ein Principal eine Anfrage stellt. Die meisten Richtlinien werden AWS als JSON-Dokumente gespeichert. Weitere Informationen zu JSON-Richtliniendokumenten finden Sie unter Übersicht über JSON-Richtlinien im IAM-Benutzerhandbuch.

Mit Hilfe von Richtlinien legen Administratoren fest, wer Zugriff auf was hat, indem sie definieren, welches Prinzipal welche Aktionen auf welchen Ressourcenund unter welchen Bedingungendurchführen darf.

Standardmäßig haben Benutzer, Gruppen und Rollen keine Berechtigungen. Ein IAM-Administrator erstellt IAM-Richtlinien und fügt sie zu Rollen hinzu, die die Benutzer dann übernehmen können. IAM-Richtlinien definieren Berechtigungen unabhängig von der Methode, die zur Ausführung der Operation verwendet wird.

Identitätsbasierte Richtlinien

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität (Benutzer, Gruppe oder Rolle) anfügen können. Diese Richtlinien steuern, welche Aktionen Identitäten für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter Definieren benutzerdefinierter IAM-Berechtigungen mit vom Kunden verwalteten Richtlinien im IAM-Benutzerhandbuch.

Identitätsbasierte Richtlinien können Inline-Richtlinien (direkt in eine einzelne Identität eingebettet) oder verwaltete Richtlinien (eigenständige Richtlinien, die mit mehreren Identitäten verbunden sind) sein. Informationen dazu, wie Sie zwischen verwalteten und Inline-Richtlinien wählen, finden Sie unter Choose between managed policies and inline policies im IAM-Benutzerhandbuch.

Ressourcenbasierte Richtlinien

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anfügen. Beispiele hierfür sind Vertrauensrichtlinien für IAM-Rollen und Amazon S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Sie müssen in einer ressourcenbasierten Richtlinie einen Prinzipal angeben.

Ressourcenbasierte Richtlinien sind Richtlinien innerhalb dieses Diensts. Sie können AWS verwaltete Richtlinien von IAM nicht in einer ressourcenbasierten Richtlinie verwenden.

Weitere Richtlinientypen

AWS unterstützt zusätzliche Richtlinientypen, mit denen die maximalen Berechtigungen festgelegt werden können, die durch gängigere Richtlinientypen gewährt werden:

- Berechtigungsgrenzen – Eine Berechtigungsgrenze legt die maximalen Berechtigungen fest, die eine identitätsbasierte Richtlinie einer IAM-Entität erteilen kann. Weitere Informationen finden Sie unter Berechtigungsgrenzen für IAM-Entitäten im -IAM-Benutzerhandbuch.
- Richtlinien zur Dienstkontrolle (SCPs) — Geben Sie die maximalen Berechtigungen für eine Organisation oder Organisationseinheit in an AWS Organizations. Weitere Informationen finden Sie unter Service-Kontrollrichtlinien im AWS Organizations -Benutzerhandbuch.
- Richtlinien zur Ressourcenkontrolle (RCPs) — Legen Sie die maximal verfügbaren Berechtigungen für Ressourcen in Ihren Konten fest. Weitere Informationen finden Sie im AWS Organizations Benutzerhandbuch unter Richtlinien zur Ressourcenkontrolle (RCPs).
- Sitzungsrichtlinien – Sitzungsrichtlinien sind erweiterte Richtlinien, die als Parameter übergeben werden, wenn Sie eine temporäre Sitzung für eine Rolle oder einen Verbundbenutzer erstellen. Weitere Informationen finden Sie unter Sitzungsrichtlinien im IAM-Benutzerhandbuch.

Mehrere Richtlinientypen

Wenn für eine Anfrage mehrere Arten von Richtlinien gelten, sind die daraus resultierenden Berechtigungen schwieriger zu verstehen. Informationen darüber, wie AWS bestimmt wird, ob eine Anfrage zulässig ist, wenn mehrere Richtlinientypen betroffen sind, finden Sie unter Bewertungslogik für Richtlinien im IAM-Benutzerhandbuch.

Wie AWS HealthOmics funktioniert mit IAM

Bevor Sie IAM zur Verwaltung des Zugriffs auf AWS verwenden, sollten Sie sich darüber informieren HealthOmics, welche IAM-Funktionen für die Nutzung mit AWS verfügbar sind. HealthOmics

IAM-Funktionen, die Sie mit verwenden können AWS HealthOmics

IAM-Feature	HealthOmics Unterstützung
Identitätsbasierte Richtlinien	Ja
Ressourcenbasierte Richtlinien	Nein

IAM-Feature	HealthOmics Unterstützung
Richtlinienaktionen	Ja
Richtlinienressourcen	Ja
Bedingungsschlüssel für die Richtlinie	Nein
ACLs	Nein
ABAC (Tags in Richtlinien)	Ja
Temporäre Anmeldeinformationen	Ja
Prinzipalberechtigungen	Ja
Servicerollen	Ja
Service-verknüpfte Rollen	Nein

Einen allgemeinen Überblick darüber, wie HealthOmics und andere AWS Dienste mit den meisten IAM-Funktionen funktionieren, finden Sie im AWS IAM-Benutzerhandbuch unter Dienste, die mit IAM funktionieren.

Serviceübergreifende Confused-Deputy-Prävention

Das Problem des verwirrten Stellvertreters ist ein Sicherheitsproblem, bei dem eine Entität, die keine Berechtigung zur Durchführung einer Aktion hat, eine privilegiertere Entität zur Durchführung der Aktion zwingen kann. In: AWS Dienststellenübergreifender Identitätswechsel kann zu Problemen mit verwirrten Stellvertretern führen. Ein serviceübergreifender Identitätswechsel kann auftreten, wenn ein Service (der Anruf-Service) einen anderen Service anruft (den aufgerufenen Service). Der Anruf-Service kann so manipuliert werden, dass er seine Berechtigungen verwendet, um auf die Ressourcen eines anderen Kunden zu reagieren, auf die er sonst nicht zugreifen dürfte. Um dies zu verhindern, bietet AWS Tools, mit denen Sie Ihre Daten für alle Services mit Serviceprinzipalen schützen können, die Zugriff auf Ressourcen in Ihrem Konto erhalten haben.

Wir empfehlen, die Kontextschlüssel `aws:SourceArn`und die `aws:SourceAccount`globalen Bedingungsschlüssel in Ressourcenrichtlinien zu verwenden, um die Berechtigungen zu beschränken, die AWS einem anderen Service für die Ressource HealthOmics erteilt.

Um das Problem der verwirrten Stellvertreter in Rollen zu vermeiden HealthOmics, die von übernommen wurden, legen Sie `arn:aws:omics:region:accountNumber:*` in der `aws:SourceArn` Vertrauensrichtlinie der Rolle den Wert auf fest. Der Platzhalter (*) wendet die Bedingung für alle HealthOmics Ressourcen an.

Die folgende Vertrauensstellungsrichtlinie gewährt HealthOmics Zugriff auf Ihre Ressourcen und verwendet die Kontextschlüssel `aws:SourceArn` und die `aws:SourceAccount` globalen Bedingungsschlüssel, um das Problem des verwirrten Stellvertreters zu verhindern. Verwenden Sie diese Richtlinie, wenn Sie eine Rolle für erstellen HealthOmics.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "omics.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:omics:us-east-1:123456789012:*"
        }
      }
    }
  ]
}
```

Identitätsbasierte Richtlinien für HealthOmics

Unterstützt Richtlinien auf Identitätsbasis: Ja

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität anfügen können, wie z. B. IAM-Benutzern, -Benutzergruppen oder -Rollen. Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter Definieren benutzerdefinierter IAM-Berechtigungen mit vom Kunden verwalteten Richtlinien im IAM-Benutzerhandbuch.

Mit identitätsbasierten IAM-Richtlinien können Sie angeben, welche Aktionen und Ressourcen zugelassen oder abgelehnt werden. Darüber hinaus können Sie die Bedingungen festlegen, unter denen Aktionen zugelassen oder abgelehnt werden. Informationen zu sämtlichen Elementen, die Sie in einer JSON-Richtlinie verwenden, finden Sie in der IAM-Referenz für JSON-Richtlinienelemente im IAM-Benutzerhandbuch.

Beispiele für identitätsbasierte Richtlinien für HealthOmics

Beispiele für HealthOmics identitätsbasierte AWS-Richtlinien finden Sie unter. Beispiele für identitätsbasierte Richtlinien für AWS HealthOmics

Ressourcenbasierte Richtlinien innerhalb HealthOmics

Unterstützt ressourcenbasierte Richtlinien: Nein

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anfügen. Beispiele für ressourcenbasierte Richtlinien sind IAM-Rollen-Vertrauensrichtlinien und Amazon-S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Für die Ressource, an welche die Richtlinie angehängt ist, legt die Richtlinie fest, welche Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen in einer ressourcenbasierten Richtlinie einen Prinzipal angeben. Zu den Prinzipalen können Konten, Benutzer, Rollen, Verbundbenutzer oder gehören. AWS-Services

Um kontoübergreifenden Zugriff zu ermöglichen, können Sie ein gesamtes Konto oder IAM-Entitäten in einem anderen Konto als Prinzipal in einer ressourcenbasierten Richtlinie angeben. Weitere Informationen finden Sie unter Kontoübergreifender Ressourcenzugriff in IAM im IAM-Benutzerhandbuch.

Richtlinienaktionen für HealthOmics

Unterstützt Richtlinienaktionen: Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer Zugriff auf was hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

Das Element `Action` einer JSON-Richtlinie beschreibt die Aktionen, mit denen Sie den Zugriff in einer Richtlinie zulassen oder verweigern können. Nehmen Sie Aktionen in eine Richtlinie auf, um Berechtigungen zur Ausführung des zugehörigen Vorgangs zu erteilen.

Eine Liste der HealthOmics Aktionen finden Sie unter Von AWS definierte Aktionen HealthOmics in der Service Authorization Reference.

Bei Richtlinienaktionen wird vor der Aktion das folgende Präfix HealthOmics verwendet:

```
omics
```

Um mehrere Aktionen in einer einzigen Anweisung anzugeben, trennen Sie sie mit Kommata:

```
"Action": [
      "omics:action1",
      "omics:action2"
         ]
```

Beispiele für HealthOmics identitätsbasierte AWS-Richtlinien finden Sie unter. Beispiele für identitätsbasierte Richtlinien für AWS HealthOmics

Richtlinienressourcen für HealthOmics

Unterstützt Richtlinienressourcen: Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer Zugriff auf was hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

Das JSON-Richtlinienelement `Resource` gibt die Objekte an, auf welche die Aktion angewendet wird. Als Best Practice geben Sie eine Ressource mit dem zugehörigen Amazon-Ressourcennamen (ARN) an. Verwenden Sie für Aktionen, die keine Berechtigungen auf Ressourcenebene unterstützen, einen Platzhalter (*), um anzugeben, dass die Anweisung für alle Ressourcen gilt.

```
"Resource": "*"
```

Eine Liste der HealthOmics Ressourcentypen und ihrer ARNs Eigenschaften finden Sie unter Von AWS definierte Ressourcen HealthOmics in der Service Authorization Reference. Informationen darüber, mit welchen Aktionen Sie den ARN jeder Ressource angeben können, finden Sie unter Von AWS definierte Aktionen HealthOmics .

Beispiele für HealthOmics identitätsbasierte AWS-Richtlinien finden Sie unter. Beispiele für identitätsbasierte Richtlinien für AWS HealthOmics

Bedingungsschlüssel für Richtlinien für HealthOmics

Bedingungsschlüssel für Richtlinien werden in nicht unterstützt HealthOmics.

Zugriffskontrolllisten (ACLs) in HealthOmics

Unterstützt ACLs: Nein

Zugriffskontrolllisten (ACLs) steuern, welche Principals (Kontomitglieder, Benutzer oder Rollen) über Zugriffsberechtigungen für eine Ressource verfügen. ACLs ähneln ressourcenbasierten Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

Attributbasierte Zugriffskontrolle (ABAC) mit HealthOmics

Unterstützt ABAC (Tags in Richtlinien): Ja

Die attributbasierte Zugriffskontrolle (ABAC) ist eine Autorisierungsstrategie, bei der Berechtigungen basierend auf Attributen, auch als Tags bezeichnet, definiert werden. Sie können Tags an IAM-Entitäten und AWS -Ressourcen anhängen und dann ABAC-Richtlinien entwerfen, die Operationen zulassen, wenn das Tag des Prinzipals mit dem Tag auf der Ressource übereinstimmt.

Um den Zugriff auf der Grundlage von Tags zu steuern, geben Sie im Bedingungselement einer Richtlinie Tag-Informationen an, indem Sie die Schlüssel `aws:ResourceTag/`*`key-name`*, `aws:RequestTag/`*`key-name`*, oder Bedingung `aws:TagKeys` verwenden.

Wenn ein Service alle drei Bedingungsschlüssel für jeden Ressourcentyp unterstützt, lautet der Wert für den Service Ja. Wenn ein Service alle drei Bedingungsschlüssel für nur einige Ressourcentypen unterstützt, lautet der Wert Teilweise.

Weitere Informationen zu ABAC finden Sie unter Definieren von Berechtigungen mit ABAC-Autorisierung im IAM-Benutzerhandbuch. Um ein Tutorial mit Schritten zur Einstellung von ABAC anzuzeigen, siehe Attributbasierte Zugriffskontrolle (ABAC) verwenden im IAM-Benutzerhandbuch.

Weitere Informationen über das Markieren von HealthOmics-Ressourcen mit Tags finden Sie unter Ressourcen taggen in HealthOmics.

Das folgende Beispiel zeigt, wie Sie eine IAM-Richtlinie schreiben können, die den Zugriff auf eine Ressource ohne ein bestimmtes Tag verweigert.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "omics:*"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "Null": {
                  "aws:RequestTag/MyCustomTag": "true"
                }
            }
        }
    ]
}
```

Temporäre Anmeldeinformationen verwenden mit HealthOmics

Unterstützt temporäre Anmeldeinformationen: Ja

Temporäre Anmeldeinformationen ermöglichen kurzfristigen Zugriff auf AWS Ressourcen und werden automatisch erstellt, wenn Sie einen Verbund verwenden oder die Rollen wechseln. AWS empfiehlt, temporäre Anmeldeinformationen dynamisch zu generieren, anstatt langfristige Zugriffsschlüssel zu

verwenden. Weitere Informationen finden Sie unter Temporäre Anmeldeinformationen in IAM und AWS-Services , die mit IAM funktionieren im IAM-Benutzerhandbuch.

Serviceübergreifende Prinzipalberechtigungen für HealthOmics

Unterstützt Forward Access Sessions (FAS): Ja

Forward Access Sessions (FAS) verwenden die Berechtigungen des Prinzipals, der einen aufruft AWS-Service, in Kombination mit der Anforderung, Anfragen an nachgelagerte Dienste AWS-Service zu stellen. Einzelheiten zu den Richtlinien für FAS-Anforderungen finden Sie unter Zugriffssitzungen weiterleiten.

Servicerollen für HealthOmics

Unterstützt Servicerollen: Ja

Eine Servicerolle ist eine IAM-Rolle, die ein Service annimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM-Administrator kann eine Servicerolle innerhalb von IAM erstellen, ändern und löschen. Weitere Informationen finden Sie unter Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS-Service im IAM-Benutzerhandbuch.

> **Warning**
>
> Durch das Ändern der Berechtigungen für eine Servicerolle kann die HealthOmics Funktionalität beeinträchtigt werden. Bearbeiten Sie Servicerollen nur, HealthOmics wenn Sie dazu eine Anleitung erhalten.

Dienstbezogene Rollen für HealthOmics

Unterstützt serviceverknüpfte Rollen: Ja

Eine dienstbezogene Rolle ist eine Art von Servicerolle, die mit einer verknüpft ist. AWS-Service Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Dienstbezogene Rollen werden in Ihrem Dienst angezeigt AWS-Konto und gehören dem Dienst. Ein IAM-Administrator kann die Berechtigungen für Service-verknüpfte Rollen anzeigen, aber nicht bearbeiten.

Details zum Erstellen oder Verwalten von serviceverknüpften Rollen finden Sie unter AWS -Services, die mit IAM funktionieren. Suchen Sie in der Tabelle nach einem Service mit einem `Yes` in der Spalte Service-linked role (Serviceverknüpfte Rolle). Wählen Sie den Link Yes (Ja) aus, um die Dokumentation für die serviceverknüpfte Rolle für diesen Service anzuzeigen.

Beispiele für identitätsbasierte Richtlinien für AWS HealthOmics

Standardmäßig sind Benutzer und Rollen nicht berechtigt, HealthOmics AWS-Ressourcen zu erstellen oder zu ändern. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern die Berechtigung erteilen, Aktionen für die Ressourcen auszuführen, die sie benötigen.

Informationen dazu, wie Sie unter Verwendung dieser beispielhaften JSON-Richtliniendokumente eine identitätsbasierte IAM-Richtlinie erstellen, finden Sie unter Erstellen von IAM-Richtlinien (Konsole) im IAM-Benutzerhandbuch.

Einzelheiten zu den von AWS definierten Aktionen und Ressourcentypen HealthOmics, einschließlich des Formats ARNs für die einzelnen Ressourcentypen, finden Sie unter Aktionen, Ressourcen und Bedingungsschlüssel für AWS HealthOmics in der Service Authorization Reference.

Themen

- Best Practices für Richtlinien
- Verwenden der Konsole HealthOmics
- Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer

Best Practices für Richtlinien

Identitätsbasierte Richtlinien legen fest, ob jemand HealthOmics AWS-Ressourcen in Ihrem Konto erstellen, darauf zugreifen oder diese löschen kann. Dies kann zusätzliche Kosten für Ihr verursachen AWS-Konto. Wenn Sie identitätsbasierte Richtlinien erstellen oder bearbeiten, befolgen Sie diese Richtlinien und Empfehlungen:

- Erste Schritte mit AWS verwalteten Richtlinien und Umstellung auf Berechtigungen mit den geringsten Rechten — Verwenden Sie die AWS verwalteten Richtlinien, die Berechtigungen für viele gängige Anwendungsfälle gewähren, um damit zu beginnen, Ihren Benutzern und Workloads Berechtigungen zu gewähren. Sie sind in Ihrem verfügbar. AWS-Konto Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie vom AWS Kunden verwaltete Richtlinien definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind. Weitere Informationen finden Sie unter Von AWS verwaltete Richtlinien oder Von AWS verwaltete Richtlinien für Auftragsfunktionen im IAM-Benutzerhandbuch.
- Anwendung von Berechtigungen mit den geringsten Rechten – Wenn Sie mit IAM-Richtlinien Berechtigungen festlegen, gewähren Sie nur die Berechtigungen, die für die Durchführung einer Aufgabe erforderlich sind. Sie tun dies, indem Sie die Aktionen definieren, die für bestimmte

Ressourcen unter bestimmten Bedingungen durchgeführt werden können, auch bekannt als die geringsten Berechtigungen. Weitere Informationen zur Verwendung von IAM zum Anwenden von Berechtigungen finden Sie unter Richtlinien und Berechtigungen in IAM im IAM-Benutzerhandbuch.

- Verwenden von Bedingungen in IAM-Richtlinien zur weiteren Einschränkung des Zugriffs – Sie können Ihren Richtlinien eine Bedingung hinzufügen, um den Zugriff auf Aktionen und Ressourcen zu beschränken. Sie können beispielsweise eine Richtlinienbedingung schreiben, um festzulegen, dass alle Anforderungen mithilfe von SSL gesendet werden müssen. Sie können auch Bedingungen verwenden, um Zugriff auf Serviceaktionen zu gewähren, wenn diese für einen bestimmten Zweck verwendet werden AWS-Service, z. CloudFormation B. Weitere Informationen finden Sie unter IAM-JSON-Richtlinienelemente: Bedingung im IAM-Benutzerhandbuch.
- Verwenden von IAM Access Analyzer zur Validierung Ihrer IAM-Richtlinien, um sichere und funktionale Berechtigungen zu gewährleisten – IAM Access Analyzer validiert neue und vorhandene Richtlinien, damit die Richtlinien der IAM-Richtliniensprache (JSON) und den bewährten IAM-Methoden entsprechen. IAM Access Analyzer stellt mehr als 100 Richtlinienprüfungen und umsetzbare Empfehlungen zur Verfügung, damit Sie sichere und funktionale Richtlinien erstellen können. Weitere Informationen finden Sie unter Richtlinienvalidierung mit IAM Access Analyzer im IAM-Benutzerhandbuch.
- Multi-Faktor-Authentifizierung (MFA) erforderlich — Wenn Sie ein Szenario haben, das IAM-Benutzer oder einen Root-Benutzer in Ihrem System erfordert AWS-Konto, aktivieren Sie MFA für zusätzliche Sicherheit. Um MFA beim Aufrufen von API-Vorgängen anzufordern, fügen Sie Ihren Richtlinien MFA-Bedingungen hinzu. Weitere Informationen finden Sie unter Sicherer API-Zugriff mit MFA im IAM-Benutzerhandbuch.

Weitere Informationen zu bewährten Methoden in IAM finden Sie unter Best Practices für die Sicherheit in IAM im IAM-Benutzerhandbuch.

Verwenden der Konsole HealthOmics

Um auf die HealthOmics AWS-Konsole zugreifen zu können, benötigen Sie ein Mindestmaß an Berechtigungen. Diese Berechtigungen müssen es Ihnen ermöglichen, Details zu den HealthOmics AWS-Ressourcen in Ihrem aufzulisten und anzuzeigen AWS-Konto. Wenn Sie eine identitätsbasierte Richtlinie erstellen, die strenger ist als die mindestens erforderlichen Berechtigungen, funktioniert die Konsole nicht wie vorgesehen für Entitäten (Benutzer oder Rollen) mit dieser Richtlinie.

Sie müssen Benutzern, die nur die API AWS CLI oder die AWS API aufrufen, keine Mindestberechtigungen für die Konsole gewähren. Stattdessen sollten Sie nur Zugriff auf die Aktionen zulassen, die der API-Operation entsprechen, die die Benutzer ausführen möchten.

Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer

In diesem Beispiel wird gezeigt, wie Sie eine Richtlinie erstellen, die IAM-Benutzern die Berechtigung zum Anzeigen der eingebundenen Richtlinien und verwalteten Richtlinien gewährt, die ihrer Benutzeridentität angefügt sind. Diese Richtlinie umfasst Berechtigungen zum Ausführen dieser Aktion auf der Konsole oder programmgesteuert mithilfe der API AWS CLI oder AWS .

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
```

```
}
```

AWS verwaltete Richtlinien für AWS HealthOmics

Eine AWS verwaltete Richtlinie ist eine eigenständige Richtlinie, die von erstellt und verwaltet wird AWS. AWS Verwaltete Richtlinien sind so konzipiert, dass sie Berechtigungen für viele gängige Anwendungsfälle bereitstellen, sodass Sie damit beginnen können, Benutzern, Gruppen und Rollen Berechtigungen zuzuweisen.

Beachten Sie, dass AWS verwaltete Richtlinien für Ihre speziellen Anwendungsfälle möglicherweise keine Berechtigungen mit den geringsten Rechten gewähren, da sie allen AWS Kunden zur Verfügung stehen. Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie vom Kunden verwaltete Richtlinien definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind.

Sie können die in AWS verwalteten Richtlinien definierten Berechtigungen nicht ändern. Wenn die in einer AWS verwalteten Richtlinie definierten Berechtigungen AWS aktualisiert werden, wirkt sich das Update auf alle Prinzidentitäten (Benutzer, Gruppen und Rollen) aus, denen die Richtlinie zugeordnet ist. AWS aktualisiert eine AWS verwaltete Richtlinie höchstwahrscheinlich, wenn eine neue Richtlinie eingeführt AWS-Service wird oder neue API-Operationen für bestehende Dienste verfügbar werden.

Weitere Informationen finden Sie unter Von AWS verwaltete Richtlinien im IAM-Benutzerhandbuch.

AWS verwaltete Richtlinie: AWSHealth OmicsServiceLinkedRolePolicy

Diese Richtlinie ist mit einer dienstbezogenen Rolle verknüpft, mit der HealthOmics Sie Aktionen in Ihrem Namen ausführen können. Sie können sie nicht AWSHealth OmicsServiceLinkedRolePolicy an Ihre Benutzer, Gruppen oder Rollen anhängen. Weitere Informationen finden Sie unter Verwenden einer dienstbezogenen Rolle.

Weitere Informationen zu dieser Richtlinie, einschließlich des JSON-Richtliniendokuments, finden Sie AWSHealthOmicsServiceLinkedRolePolicyim AWS Managed Policy Reference Guide.

AWS verwaltete Richtlinie: AmazonOmicsFullAccess

Sie können die `AmazonOmicsFullAccess` Richtlinie an Ihre IAM-Identitäten anhängen, um ihnen vollen Zugriff zu gewähren. HealthOmics

Diese Richtlinie gewährt volle Zugriffsberechtigungen für alle HealthOmics Aktionen. Wenn Sie einen Annotations- oder Variantenspeicher erstellen, gewährt Ihnen Omics auch über eine Resource Share Invitation in der Resource Access Manager (RAM) -Konsole Zugriff auf diesen Speicher. Weitere Informationen zu Resource Share-Einladungen über Lake Formation finden Sie unter Kontoübergreifender Datenaustausch in Lake Formation. Für eine Omics-Administratorrichtlinie benötigen Sie außerdem die folgenden Berechtigungen, um auf Ihren Amazon S3 S3-Bucket zuzugreifen.

- PutObject
- GetObject
- ListBucket
- AbortMultipartUpload
- ListMultipartUploadParts

JSON

```
{
 "Version":"2012-10-17",
 "Statement": [
  {
   "Effect": "Allow",
   "Action": [
    "omics:*"
   ],
   "Resource": "*"
  },
  {
   "Effect": "Allow",
   "Action": [
    "ram:AcceptResourceShareInvitation",
    "ram:GetResourceShareInvitations"
   ],
   "Resource": "*",
   "Condition": {
```

```
    "StringEquals": {
     "aws:CalledViaLast": "omics.amazonaws.com"
    }
   }
  },
  {
   "Effect": "Allow",
   "Action": "iam:PassRole",
   "Resource": "*",
   "Condition": {
    "StringEquals": {
     "iam:PassedToService": "omics.amazonaws.com"
    }
   }
  }
 ]
}
```

AWS verwaltete Richtlinie: AmazonOmicsReadOnlyAccess

Sie können die `AWSOmicsReadOnlyAccess` Richtlinie an Ihre IAM-Identitäten anhängen, wenn Sie die Berechtigungen für diese Identität auf schreibgeschützten Zugriff beschränken möchten.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "omics:Get*",
                "omics:List*"
            ],
            "Resource": "*"
        }
    ]
}
```

HealthOmics Aktualisierungen der verwalteten Richtlinien AWS

Hier finden Sie Informationen zu Aktualisierungen AWS verwalteter Richtlinien, die HealthOmics seit Beginn der Nachverfolgung dieser Änderungen durch diesen Dienst vorgenommen wurden. Abonnieren Sie den RSS-Feed auf der Seite HealthOmics Dokumentenverlauf, um automatische Benachrichtigungen über Änderungen an dieser Seite zu erhalten.

Änderungen	Beschreibung	Date
AWSHealthOmicsServiceLinkedRolePolicy - Neue Richtlinie hinzugefügt	HealthOmics hat eine neue Richtlinie hinzugefügt, die AWS Berechtigungen HealthOmics zur Verwaltung von VPC-Netzwerkressourcen in Ihrem Konto gewährt. Weitere Informationen hierzu finden Sie unter AWSHealthOmicsServiceLinkedRolePolicy.	20. Februar 2026
AmazonOmicsFullAccess - Neue Richtlinie hinzugefügt	HealthOmics Es wurde eine neue Richtlinie hinzugefügt, um einem Benutzer vollen Zugriff auf alle Aktionen und Ressourcen zu gewähren. Weitere Informationen hierzu finden Sie unter AmazonOmicsFullAccess.	23. Februar 2023
HealthOmics hat begonnen, Änderungen zu verfolgen	HealthOmics hat begonnen, Änderungen für die AWS verwalteten Richtlinien zu verfolgen.	29. November 2022

Änderungen	Beschreibung	Date
AmazonOmicsReadOnlyAccess - Neue Richtlinie hinzugefügt	HealthOmics Es wurde eine neue Richtlinie hinzugefügt, die den Zugriff nur auf Lesezugriff beschränkt. Weitere Informationen hierzu finden Sie unter AmazonOmicsReadOnlyAccess.	29. November 2022

Problembehandlung bei AWS HealthOmics Identität und Zugriff

Verwenden Sie die folgenden Informationen, um häufig auftretende Probleme zu diagnostizieren und zu beheben, die bei der Arbeit mit AWS HealthOmics und IAM auftreten können.

Themen

- Ich bin nicht berechtigt, eine Aktion durchzuführen in HealthOmics
- Ich bin nicht berechtigt, iam auszuführen: PassRole
- Ich möchte Personen außerhalb von mir den Zugriff AWS-Konto auf meine HealthOmics Ressourcen ermöglichen

Ich bin nicht berechtigt, eine Aktion durchzuführen in HealthOmics

Wenn Sie eine Fehlermeldung erhalten, dass Sie nicht zur Durchführung einer Aktion berechtigt sind, müssen Ihre Richtlinien aktualisiert werden, damit Sie die Aktion durchführen können.

Der folgende Beispielfehler tritt auf, wenn der IAM-Benutzer `mateojackson` versucht, über die Konsole Details zu einer fiktiven *`my-example-widget`*-Ressource anzuzeigen, jedoch nicht über `omics:`*`GetWidget`*-Berechtigungen verfügt.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 omics:GetWidget on resource: my-example-widget
```

In diesem Fall muss die Richtlinie für den Benutzer `mateojackson` aktualisiert werden, damit er mit der `omics:`*`GetWidget`*-Aktion auf die *`my-example-widget`*-Ressource zugreifen kann.

Wenn Sie Hilfe benötigen, wenden Sie sich an Ihren AWS Administrator. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Ich bin nicht berechtigt, iam auszuführen: PassRole

Wenn Sie eine Fehlermeldung erhalten, dass Sie nicht berechtigt sind, die `iam:PassRole` Aktion durchzuführen, müssen Ihre Richtlinien aktualisiert werden, damit Sie eine Rolle an AWS übergeben können HealthOmics.

Einige AWS-Services ermöglichen es Ihnen, eine bestehende Rolle an diesen Service zu übergeben, anstatt eine neue Servicerolle oder eine mit einem Service verknüpfte Rolle zu erstellen. Hierzu benötigen Sie Berechtigungen für die Übergabe der Rolle an den Dienst.

Der folgende Beispielfehler tritt auf, wenn ein IAM-Benutzer mit dem Namen `marymajor` versucht, die Konsole zu verwenden, um eine Aktion in AWS HealthOmics auszuführen. Die Aktion erfordert jedoch, dass der Service über Berechtigungen verfügt, die durch eine Servicerolle gewährt werden. Mary besitzt keine Berechtigungen für die Übergabe der Rolle an den Dienst.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

In diesem Fall müssen die Richtlinien von Mary aktualisiert werden, um die Aktion `iam:PassRole` ausführen zu können.

Wenn Sie Hilfe benötigen, wenden Sie sich an Ihren AWS Administrator. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Ich möchte Personen außerhalb von mir den Zugriff AWS-Konto auf meine HealthOmics Ressourcen ermöglichen

Sie können eine Rolle erstellen, mit der Benutzer in anderen Konten oder Personen außerhalb Ihrer Organisation auf Ihre Ressourcen zugreifen können. Sie können festlegen, wem die Übernahme der Rolle anvertraut wird. Für Dienste, die ressourcenbasierte Richtlinien oder Zugriffskontrolllisten (ACLs) unterstützen, können Sie diese Richtlinien verwenden, um Personen Zugriff auf Ihre Ressourcen zu gewähren.

Weitere Informationen dazu finden Sie hier:

- Informationen darüber, ob AWS diese Funktionen HealthOmics unterstützt, finden Sie unterWie AWS HealthOmics funktioniert mit IAM.

- Informationen dazu, wie Sie Zugriff auf Ihre Ressourcen gewähren können, AWS-Konten die Ihnen gehören, finden Sie im IAM-Benutzerhandbuch unter Bereitstellen von Zugriff für einen IAM-Benutzer in einem anderen AWS-Konto , den Sie besitzen.
- Informationen dazu, wie Sie Dritten Zugriff auf Ihre Ressourcen gewähren können AWS-Konten, finden Sie AWS-Konten im IAM-Benutzerhandbuch unter Gewähren des Zugriffs für Dritte.
- Informationen dazu, wie Sie über einen Identitätsverbund Zugriff gewähren, finden Sie unter Gewähren von Zugriff für extern authentifizierte Benutzer (Identitätsverbund) im IAM-Benutzerhandbuch.
- Informationen zum Unterschied zwischen der Verwendung von Rollen und ressourcenbasierten Richtlinien für den kontoübergreifenden Zugriff finden Sie unter Kontoübergreifender Ressourcenzugriff in IAM im IAM-Benutzerhandbuch.

Verwenden von serviceverknüpften Rollen für AWS HealthOmics

AWS HealthOmics verwendet AWS Identity and Access Management (IAM) serviceverknüpfte Rollen. Eine serviceverknüpfte Rolle ist eine einzigartige Art von IAM-Rolle, die direkt mit AWS verknüpft ist. HealthOmics Servicebezogene Rollen sind von AWS vordefiniert HealthOmics und beinhalten alle Berechtigungen, die der Service benötigt, um andere AWS Services in Ihrem Namen aufzurufen.

Eine serviceverknüpfte Rolle HealthOmics erleichtert die Einrichtung von AWS, da Sie die erforderlichen Berechtigungen nicht manuell hinzufügen müssen. AWS HealthOmics definiert die Berechtigungen seiner serviceverknüpften Rollen, und sofern nicht anders definiert, HealthOmics kann nur AWS seine Rollen übernehmen. Die definierten Berechtigungen umfassen die Vertrauens- und Berechtigungsrichtlinie. Diese Berechtigungsrichtlinie kann keinen anderen IAM-Entitäten zugewiesen werden.

Sie können eine serviceverknüpfte Rolle erst löschen, nachdem ihre verwandten Ressourcen gelöscht wurden. Dadurch werden Ihre HealthOmics AWS-Ressourcen geschützt, da Sie die Zugriffsberechtigung für die Ressourcen nicht versehentlich entziehen können.

Informationen zu anderen Services, die serviceverknüpfte Rollen unterstützen, finden Sie unter AWS Services, die mit IAM funktionieren. Suchen Sie in der Spalte Service-verknüpfte Rollen nach den Services, für die Ja steht. Wählen Sie über einen Link Ja aus, um die Dokumentation zu einer serviceverknüpften Rolle für diesen Service anzuzeigen.

Servicebezogene Rollenberechtigungen für AWS HealthOmics

AWS HealthOmics verwendet eine servicebezogene Rolle mit dem Namen AwsServiceRoleForHealthOmics— eine verwaltete IAM-Richtlinie mit bereichsbezogenen Berechtigungen, die AWS Berechtigungen HealthOmics zur Verwaltung von VPC-Netzwerkressourcen gewährt, einschließlich der Erstellung, Konfiguration und Löschung von Elastic Network Interfaces () ENIs in Ihrem Namen.

Die AwsServiceRoleForHealthOmics serviceverknüpfte Rolle vertraut darauf, dass der Service-Principal die `omics.amazonaws.com` Rolle übernimmt.

Die Richtlinie für Rollenberechtigungen ist die AWSHealth OmicsServiceLinkedRolePolicy AWS verwaltete Richtlinie. Die vollständige Liste der Berechtigungen, die diese Richtlinie gewährt, einschließlich des JSON-Richtliniendokuments, finden Sie AWSHealthOmicsServiceLinkedRolePolicyim AWS Managed Policy Reference Guide.

Sie müssen Berechtigungen konfigurieren, damit eine Benutzer, Gruppen oder Rollen eine serviceverknüpfte Rolle erstellen, bearbeiten oder löschen können. Weitere Informationen finden Sie unter serviceverknüpfte Rollenberechtigung im IAM-Benutzerhandbuch.

Eine serviceverknüpfte Rolle für AWS erstellen HealthOmics

Sie müssen eine serviceverknüpfte Rolle nicht manuell erstellen. AWS HealthOmics erstellt die serviceverknüpfte Rolle für Sie zum ersten Mal, wenn Sie eine laufspezifische Konfiguration mit VPC-Konfiguration erstellen.

Wenn Sie diese serviceverknüpfte Rolle löschen und sie dann erneut erstellen müssen, können Sie dasselbe Verfahren anwenden, um die Rolle in Ihrem Konto neu anzulegen. Weitere Informationen finden Sie unter Erstellen einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Bearbeiten einer serviceverknüpften Rolle für AWS HealthOmics

In AWS können Sie die mit dem AwsServiceRoleForHealthOmics Service verknüpfte Rolle HealthOmics nicht bearbeiten. Da möglicherweise verschiedene Entitäten auf die Rolle verweisen, kann der Rollenname nach dem Erstellen einer serviceverknüpften Rolle nicht mehr geändert werden. Sie können jedoch die Beschreibung der Rolle mit IAM bearbeiten. Weitere Informationen finden Sie unter Bearbeiten einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Löschen einer serviceverknüpften Rolle für AWS HealthOmics

Wenn Sie die Funktion, für die eine serviceverknüpfte Rolle erforderlich ist, nicht mehr verwenden müssen, empfehlen wir Ihnen, diese Rolle zu löschen. Auf diese Weise haben Sie keine ungenutzte juristische Stelle, die nicht aktiv überwacht oder verwaltet wird.

> **Note**
>
> Wenn der HealthOmics AWS-Service die Rolle verwendet, wenn Sie versuchen, die Ressourcen zu löschen, schlägt das Löschen möglicherweise fehl. Wenn dies passiert, warten Sie einige Minuten und versuchen Sie es erneut.

Sie können diese dienstverknüpfte Rolle nur löschen, wenn keine aktive Ausführung mit einer der laufspezifischen Konfigurationen mit VPC-Konfiguration vorhanden ist. Dadurch wird sichergestellt, dass Sie AWS nicht versehentlich die Berechtigungen HealthOmics zur Verwaltung von VPC-Netzwerkressourcen in Ihrem Namen entziehen können.

So löschen Sie die serviceverknüpfte Rolle mit IAM

Verwenden Sie die IAM-Konsole, die oder die AWS API, um die AWS CLI serviceverknüpfte Rolle zu löschen. AwsServiceRoleForHealthOmics Weitere Informationen finden Sie unter Löschen einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Unterstützte Regionen für Rollen im HealthOmics Zusammenhang mit AWS

AWS HealthOmics unterstützt die Verwendung von serviceverknüpften Rollen in allen Regionen, in denen der Service verfügbar ist. Weitere Informationen finden Sie unter AWS -Regionen und Endpunkte.

Konformitätsvalidierung für AWS HealthOmics

Externe Prüfer bewerten die Sicherheit und Einhaltung von Vorschriften im AWS HealthOmics Rahmen mehrerer AWS Compliance-Programme. Dazu gehören HIPAA, FedRAMP und andere. Die folgende Tabelle zeigt die Compliance-Zertifizierungen für den Service. HealthOmics

Zertifizierung	Link
HIPAA	Referenz für HIPAA-fähige Dienste

Zertifizierung	Link
HiTrust-CSF	Gemeinsamer Sicherheitsrahmen der Health Information Trust Alliance
FedRAMP Moderate (Osten/West)	Risiko- und Genehmigungsmanagementprogramm des Bundes
ISO/CSA-STERN	ISO- und CSA STAR-zertifiziert
C5	Katalog zur Kontrolle der Einhaltung von Cloud-Computing-Vorschriften
DoD CC SRG IL2	Leitfaden für Cloud-Computing-Sicherheitsanforderungen des Verteidigungsministeriums
ENS High	Esquema Nacional de Seguridad
FINMA	Eidgenössische Finanzmarktaufsicht
IST MAP	Programm zur Verwaltung und Bewertung der Sicherheit von Informationssystemen
OSPAR	Prüfbericht für ausgelagerte Dienstleister
PCI	Datensicherheitsstandard der Zahlungskartenbranche
Pinakes	Bankenverband CCI — Qualifizierung durch Dritte
PiTuKri	Kriterien für die Bewertung der Informationssicherheit von Cloud-Diensten
SOC 1,2,3	System- und Organisationskontrollen

Eine Liste aller AWS Services, die für bestimmte Compliance-Programme gelten, finden Sie unter AWS-Services in Umfang nach Compliance-Programm . Allgemeine Informationen finden Sie unter AWS -Compliance-Programme.

Sie können Prüfberichte von Drittanbietern herunterladen unter AWS Artifact. Weitere Informationen finden Sie unter Berichte herunterladen unter .

HealthOmics Datenspeicher verwenden die Proben-ID für die interne Benennung von Dateien und für die Kennzeichnung von Ressourcen. Bevor Sie Daten aufnehmen, überprüfen Sie, ob die Proben-ID PHI-Daten enthält. Ist dies der Fall, ändern Sie die Proben-ID, bevor Sie die Daten aufnehmen. Weitere Informationen finden Sie in den Anleitungen auf der Webseite zur AWS HIPAA-Konformität.

Ihre Verantwortung für die Einhaltung der Vorschriften bei der Nutzung AWS HealthOmics hängt von der Vertraulichkeit Ihrer Daten, den Compliance-Zielen Ihres Unternehmens und den geltenden Gesetzen und Vorschriften ab. AWS stellt die folgenden Ressourcen zur Verfügung, die Sie bei der Einhaltung der Vorschriften unterstützen:

- Schnellstartanleitungen für Sicherheit und Compliance – In diesen Bereitstellungsleitfäden werden architektonische Überlegungen erörtert und Schritte für die Bereitstellung von sicherheits- und konformitätsorientierten Basisumgebungen auf AWS angegeben.
- Whitepaper „Architecting for HIPAA Security and Compliance“ — In diesem Whitepaper wird beschrieben, wie Unternehmen HIPAA-konforme Anwendungen erstellen können AWS .
- AWS Compliance-Ressourcen — Diese Sammlung von Arbeitsmappen und Leitfäden kann auf Ihre Branche und Ihren Standort zutreffen.
- Bewertung von Ressourcen anhand von Regeln im AWS Config Entwicklerhandbuch — AWS Config; bewertet, wie gut Ihre Ressourcenkonfigurationen den internen Praktiken, Branchenrichtlinien und Vorschriften entsprechen.
- AWS Security Hub CSPM— Dieser AWS Service bietet einen umfassenden Überblick über Ihren Sicherheitsstatus, sodass Sie überprüfen können AWS , ob Sie die Sicherheitsstandards und Best Practices der Branche einhalten.

Resilienz in HealthOmics

Die AWS globale Infrastruktur basiert auf Availability AWS-Regionen Zones. AWS-Regionen bieten mehrere physisch getrennte und isolierte Availability Zones, die über Netzwerke mit niedriger Latenz, hohem Durchsatz und hoher Redundanz miteinander verbunden sind. Mithilfe von Availability Zones können Sie Anwendungen und Datenbanken erstellen und ausführen, die automatisch Failover zwischen Zonen ausführen, ohne dass es zu Unterbrechungen kommt. Availability Zones sind besser verfügbar, fehlertoleranter und skalierbarer als herkömmliche Infrastrukturen mit einem oder mehreren Rechenzentren.

Weitere Informationen zu Availability Zones AWS-Regionen und Availability Zones finden Sie unter AWS Globale Infrastruktur.

Zusätzlich zur AWS globalen Infrastruktur HealthOmics bietet AWS mehrere Funktionen zur Unterstützung Ihrer Datenausfallsicherheit und Backup-Anforderungen.

AWS HealthOmics und Schnittstelle VPC-Endpunkte (AWS PrivateLink)

Sie können eine private Verbindung zwischen Ihrer VPC und AWS HealthOmics durch die Erstellung eines Schnittstellen-VPC-Endpunkts herstellen. Schnittstellenendpunkte werden von einer Technologie unterstützt AWS PrivateLink, mit der Sie ohne Internet-Gateway, NAT-Gerät, VPN-Verbindung oder AWS Direct Connect-Verbindung privat auf HealthOmics API-Operationen zugreifen können. Instances in Ihrer VPC benötigen keine öffentlichen IP-Adressen, um mit HealthOmics API-Vorgängen zu kommunizieren. Der Datenverkehr zwischen Ihrer VPC und HealthOmics wird nicht außerhalb des Amazon-Netzwerks übertragen.

Jeder Schnittstellenendpunkt wird durch eine oder mehrere Elastic-Network-Schnittstellen in Ihren Subnetzen dargestellt.

Weitere Informationen finden Sie unter Interface VPC Endpoints (AWS PrivateLink) im Amazon VPC-Benutzerhandbuch.

VPC-Endpunktrichtlinien werden HealthOmics für alle Regionen außer Israel (Tel Aviv) unterstützt. Standardmäßig HealthOmics ist der vollständige Zugriff auf den Endpunkt zulässig.

Überlegungen zu HealthOmics VPC-Endpunkten

Bevor Sie einen Schnittstellen-VPC-Endpunkt für einrichten HealthOmics, sollten Sie sich die Eigenschaften und Einschränkungen von Schnittstellenendpunkten im Amazon VPC-Benutzerhandbuch ansehen.

HealthOmics unterstützt Aufrufe aller HealthOmics Speicher-API-Aktionen von Ihrer VPC aus.

VPC-Endpunktrichtlinien werden HealthOmics standardmäßig nicht unterstützt, Sie können jedoch einen VPC-Endpunkt für den vollen HealthOmics Zugriff auf die HealthOmics Speichervorgänge erstellen. Weitere Informationen finden Sie unter Steuerung des Zugriffs auf Services mit VPC-Endpunkten im Amazon-VPC-Benutzerhandbuch.

Erstellen eines Schnittstellen-VPC-Endpunkts für HealthOmics

Sie können einen VPC-Endpunkt für den HealthOmics Service erstellen, indem Sie die Amazon VPC-Konsole oder die AWS Command Line Interface () verwenden.AWS CLI Weitere Informationen finden Sie unter Erstellung eines Schnittstellenendpunkts im Benutzerhandbuch für Amazon VPC.

Erstellen Sie einen VPC-Endpunkt für, HealthOmics indem Sie die folgenden Servicenamen verwenden:

- 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

Die Regionen USA Ost (Nord-Virginia), USA West (Oregon) und USA Ost (Ohio) unterstützen FIPS-Endpunkte. AWS PrivateLink Für diese Regionen können Sie auch die folgenden Dienstnamen verwenden:

- com.amazonaws. *region*.storage-omics-fips
- com.amazonaws. *region*.control-storage-omics-fips
- com.amazonaws. *region*.analytics-omics-fips
- com.amazonaws. *region*.workflows-omics-fips
- com.amazonaws. *region*.tags-omics-fips

Wenn Sie das private DNS für den Endpunkt aktivieren, können Sie API-Anfragen an diesen richten, HealthOmics indem Sie beispielsweise den Standard-DNS-Namen für die Region verwenden. `omics.us-east-1.amazonaws.com`

Weitere Informationen finden Sie unter Zugriff auf einen Service über einen Schnittstellenendpunkt im Benutzerhandbuch für Amazon VPC.

Erstellen einer VPC-Endpunktrichtlinie für HealthOmics

Sie können eine Endpunktrichtlinie an Ihren VPC-Endpunkt anhängen, der den Zugriff auf HealthOmics steuert. Die Richtlinie gibt die folgenden Informationen an:

- Der Prinzipal, der die Aktionen ausführen kann
- Aktionen, die ausgeführt werden können
- Ressourcen, für die Aktionen ausgeführt werden können

Weitere Informationen finden Sie unter Steuerung des Zugriffs auf Services mit VPC-Endpunkten im Amazon-VPC-Benutzerhandbuch.

Beispiel: VPC-Endpunktrichtlinie für HealthOmics Aktionen.

Das Folgende ist ein Beispiel für eine Endpunktrichtlinie für HealthOmics. Wenn diese Richtlinie an einen Endpunkt angehängt ist, gewährt sie allen Prinzipalen Zugriff auf HealthOmics Aktionen auf allen Ressourcen.

API

```
{
   "Statement":[
      {
         "Principal":"*",
         "Effect":"Allow",
         "Action":[
            "omics:List*"
         ],
         "Resource":"*"
      }
   ]
}
```

AWS CLI

```
aws ec2 modify-vpc-endpoint \
    --vpc-endpoint-id vpce-id \
    --region us-west-2 \
    --policy-document \
    "{\"Statement\":[{\"Principal\":\"*\",\"Effect\":\"Allow\",\"Action\":
[\"omics:List*\"],\"Resource\":\"*\"}]}"
```

Besondere Überlegungen für den Zugriff auf Lesesätze mithilfe von Amazon S3-URIs

Um über Amazon S3-URIs auf Lesesätze zuzugreifen, wenn Sie eine private Verbindung verwenden, richten Sie die PrivateLink Schnittstellenendpunkte im Sequenzspeicher ein. Nachdem Sie sie eingerichtet haben, haben die Endpunkte die folgenden Formate:

```
com.amazonaws.region.storage-omics
com.amazonaws.region.control-storage-omics
```

Um Gateway-Endpunkte zu verwenden, folgen Sie der Anleitung Gateway-Endpunkte für Amazon S3, um Ihre Gateway-Endpunkte zu konfigurieren. HealthOmics besitzt den Amazon S3-Bucket, sodass Sie die Bucket-Richtlinie nicht erstellen oder anpassen müssen. Gateway-Endpunkte basieren auf der Richtlinie, die dem Benutzer oder der Rolle zugeordnet ist, die auf die Daten zugreift. Sie können Endpunkte jedoch auch mit restriktiveren Richtlinien konfigurieren. Diese Richtlinien können Zugriffsbeschränkungen beinhalten, die auf dem Amazon S3 Access Point-ARN und den Amazon S3-Aktionen basieren.

Überwachung von AWS HealthOmics

Die Überwachung ist ein wichtiger Bestandteil der Aufrechterhaltung der Zuverlässigkeit, Verfügbarkeit und Leistung von AWS HealthOmics und Ihren anderen AWS Lösungen. AWS bietet die folgenden Überwachungstools, um AWS zu beobachten HealthOmics, zu melden, wenn etwas nicht stimmt, und gegebenenfalls automatische Maßnahmen zu ergreifen:

- Amazon CloudWatch überwacht Ihre AWS Ressourcen und die Anwendungen, auf denen Sie laufen, AWS in Echtzeit. Sie können Kennzahlen erfassen und verfolgen, benutzerdefinierte Dashboards erstellen und Alarme festlegen, die Sie benachrichtigen oder Maßnahmen ergreifen, wenn eine bestimmte Metrik einen von Ihnen festgelegten Schwellenwert erreicht. Weitere Informationen finden Sie im CloudWatch Amazon-Benutzerhandbuch.
- Anhand von Run-Metriken wird die Ressourcenauslastung für Ihre Läufe und Aufgaben nahezu in Echtzeit gemeldet, einschließlich CPU-, GPU-, Speicher- und Dateisystemauslastung. AWS verkauft HealthOmics diese Metriken CloudWatch als OpenTelemetry Metriken, und Sie fragen sie mit PromQL ab. Weitere Informationen finden Sie unter Metriken für private Workflows ausführen.
- Amazon CloudWatch Logs ermöglicht es Ihnen, Ihre Protokolldateien von Amazon EC2-Instances und anderen Quellen aus zu überwachen CloudTrail, zu speichern und darauf zuzugreifen. CloudWatch Logs können Informationen in den Protokolldateien überwachen und Sie benachrichtigen, wenn bestimmte Schwellenwerte erreicht werden. Sie können Ihre Protokolldaten auch in einem sehr robusten Speicher archivieren. Weitere Informationen finden Sie im Amazon CloudWatch Logs-Benutzerhandbuch.
- AWS CloudTrail erfasst API-Aufrufe und zugehörige Ereignisse, die von oder im Namen Ihres AWS-Konto -Kontos erfolgten, und übermittelt die Protokolldateien an einen von Ihnen angegebenen Amazon-S3-Bucket. Sie können die Benutzer und Konten, die AWS aufgerufen haben, identifizieren, sowie die Quell-IP-Adresse, von der diese Aufrufe stammen, und den Zeitpunkt der Aufrufe ermitteln. Weitere Informationen finden Sie im AWS CloudTrail - Benutzerhandbuch.
- Amazon EventBridge ist ein serverloser Event-Bus-Dienst, der es einfach macht, Ihre Anwendungen mit Daten aus einer Vielzahl von Quellen zu verbinden. EventBridge liefert einen Stream von Echtzeitdaten aus Ihren eigenen Anwendungen, Software-as-a-Service (SaaS-) Anwendungen und AWS Diensten und leitet diese Daten an Ziele wie Lambda weiter. Auf diese Weise können Sie Ereignisse überwachen, die in Services auftreten, und ereignisgesteuerte Architekturen erstellen. Weitere Informationen finden Sie im EventBridge Amazon-Benutzerhandbuch.

> **Note**
>
> Für Serviceupdates konfigurieren und überwachen Sie Ihr Personal Health Dashboard. Weitere Informationen zur Verwaltung des Dashboards finden Sie unter Erste Schritte mit Ihrem AWS Health Dashboard.

Themen

- S3-Zugriffsprotokollierung
- Überwachung HealthOmics mit CloudWatch Metriken
- Metriken für private Workflows ausführen
- Überwachung HealthOmics mit CloudWatch Logs
- Protokollieren von AWS HealthOmics API-Aufrufen mit AWS CloudTrail
- Verwenden EventBridge mit AWS HealthOmics

S3-Zugriffsprotokollierung

Sie können den Amazon S3-API-Zugriff auf HealthOmics Sequenzspeicherdaten mithilfe der vom Store erstellten Zugriffsprotokolle überwachen. Sie können es verwenden CloudWatch , um den S3-Zugriff von HealthOmics API-Vorgängen aus zu überwachen. CloudWatch bietet Einblick in den Amazon S3-Zugriff, der von Ihrem eigenen Konto aus erfolgt. Wenn Sie als Dateneigentümer den Zugriff auf ein Konto eines Drittanbieters teilen, ist die Zugriffsprotokollierung nicht verfügbar CloudWatch. Verwenden Sie stattdessen das S3-Zugriffsprotokoll des Shops, das den gesamten S3-Zugriff auf die Daten im konfigurierten Amazon S3-Bucket protokolliert.

Konfigurieren Sie S3-Zugriffsprotokolle mithilfe der `CreateSequenceStore` oder `UpdateSequenceStore` API-Operationen. Stellen Sie außerdem sicher, dass der HealthOmics Service Principal (`omics.amazonaws.com`) über `s3:PutObject` Berechtigungen für das konfigurierte S3-Präfix verfügt.

> **Note**
>
> Protokolle verwenden die Standardverschlüsselungskonfiguration des Ziel-Buckets. Wenn der Bucket einen vom Kunden verwalteten Schlüssel verwendet, muss der Serviceprinzipal Zugriff haben, um den Schlüssel zum Schreiben verwenden zu können.

Um die Zugriffsprotokollierung zu deaktivieren, verwenden Sie die Zugriffsprotokollkonfiguration `UpdateSequenceStore` und setzen Sie sie auf leer.

Überwachung HealthOmics mit CloudWatch Metriken

Sie können die HealthOmics Nutzung CloudWatch überwachen. Dabei werden Rohdaten gesammelt und zu lesbaren Metriken verarbeitet, die nahezu in Echtzeit verfügbar sind. Diese Statistiken werden 15 Monate gespeichert, damit Sie auf Verlaufsinformationen zugreifen können und einen besseren Überblick darüber erhalten, wie Ihre Webanwendung oder der Service ausgeführt werden. Sie können auch Alarme einrichten, die auf bestimmte Grenzwerte achten und Benachrichtigungen senden oder Aktivitäten auslösen, wenn diese Grenzwerte erreicht werden. Weitere Informationen finden Sie im CloudWatch Amazon-Benutzerhandbuch.

Der AWS HealthOmics Service meldet die folgenden Metriken im `AWS/Omics` Namespace.

Die Metriken zur Anzahl der API-Aufrufe werden für Folgendes AWS HealthOmics APIs gemeldet. Es wird nur die Dimension API-Operation gemeldet.

- Referenz und Referenzspeicher APIs —CreateReferenceStore, DeleteReferenceStore, StartReferenceImportJob
- Sequenzspeicher und Lesesatz APIs — CreateSequenceStore DeleteSequenceStore, StartReadSetImportJob, StartReadSetActivationJob, StartReadSetExportJob
- Variantenspeicher APIs — CreateVariantStore, DeleteVariantStore, StartVariantImportJob, CancelVariantImportJob
- Speicher für Anmerkungen APIs — CreateAnnotationStore, DeleteAnotationStore, StartAnnotationImportJob, CancelAnnotationImportJob
- Arbeitsablauf, Ausführung und Gruppe ausführen APIs — CreateWorkflow DeleteWorkflow StartRun, CancelRun, DeleteRun, CreateRunGroup, DeleteRunGroup

Anzeigen von *AWS HealthOmics*-Metriken

CloudWatch AWS HealthOmics Metriken für können in der CloudWatch Konsole angezeigt werden.

Um Metriken anzuzeigen (CloudWatch Konsole)

1. Melden Sie sich bei der AWS-Managementkonsole an und öffnen Sie die CloudWatch -Konsole.
2. Wählen Sie Metriken, Alle Metriken und dann AWS/Usage aus.

3. Service filtern nach. AWS HealthOmics
4. Wählen Sie die Dimension, den Namen einer Metrik und schließlich Add to graph (Dem Diagramm hinzufügen) aus.
5. Wählen Sie einen Wert für den Datumsbereich aus. Die Anzahl der Kennzahlen für den ausgewählten Zeitraum wird im Diagramm angezeigt.

Einen Alarm erstellen mit CloudWatch

Ein CloudWatch Alarm überwacht eine einzelne Metrik über einen bestimmten Zeitraum und führt eine oder mehrere Aktionen aus: das Senden einer Benachrichtigung an ein Amazon Simple Notification Service (Amazon SNS) -Thema oder an eine Auto Scaling Scaling-Richtlinie. Die Aktion oder Aktionen basieren auf dem Wert der Metrik im Verhältnis zu einem bestimmten Schwellenwert über eine von Ihnen angegebene Anzahl von Zeiträumen. CloudWatch kann Ihnen auch eine Amazon SNS SNS-Nachricht senden, wenn sich der Status des Alarms ändert.

CloudWatch Alarme lösen nur dann Aktionen aus, wenn sich der Status ändert und für den von Ihnen angegebenen Zeitraum andauert.

So zeigen Sie Metriken an (Konsole) CloudWatch

1. Melden Sie sich bei der AWS-Managementkonsole an und öffnen Sie die CloudWatch -Konsole.
2. Wählen Sie Alarms und dann Create Alarm.
3. Wählen Sie AWS/Usage und anschließend mithilfe der Service-Dimension eine AWS HealthOmics Metrik aus.
4. Wählen Sie für Time Range den zu überwachenden Zeitbereich und dann Next.
5. Geben Sie Name (Name) und Description (Beschreibung) ein.
6. Wählen Sie für Whenever **>=** und geben Sie einen Maximalwert ein.
7. Wenn Sie eine E-Mail senden CloudWatch möchten, wenn der Alarmstatus erreicht ist, wählen Sie im Bereich Aktionen für Wann immer dieser Alarm die Option Status ist ALARM. Wählen Sie unter Benachrichtigung senden an eine Mailingliste aus oder wählen Sie Neue Liste und erstellen Sie eine neue Mailingliste.
8. Nutzen Sie die Alarmvorschau im Bereich Alarm Preview. Wenn Sie mit dem Alarm zufrieden sind, wählen Sie Create Alarm.

Metriken für private Workflows ausführen

HealthOmics veröffentlicht Kennzahlen zur Ressourcenauslastung für Ihre Läufe und Aufgaben nahezu in Echtzeit bei Amazon CloudWatch. Diese Metriken geben dir einen Überblick darüber, wie deine Läufe voranschreiten, während sie ausgeführt werden. Du kannst diese Metriken verwenden, um:

- Identifizieren Sie Engpässe in der Central Processing Unit (CPU) oder der Graphics Processing Unit (GPU), während Aufgaben noch ausgeführt werden.
- Erkennen Sie Speicherauslastung oder die Erschöpfung des Scratch-Speichers, bevor eine Aufgabe fehlschlägt.
- Right-size die Rechen- und Speicherkonfigurationen für Ihre Workflows.
- Erstellen CloudWatch Dashboards und Alarme oder integrieren Sie sie in Observability-Tools von Drittanbietern.
- Debuggen Sie einen fehlgeschlagenen oder langsamen Lauf zusammen mit dem Ausführungsdatensatz, den Aufgabenaufzeichnungen und den Protokollen. Weitere Informationen finden Sie unter Debuggen eines HealthOmics Workflow-Ausführung.

Ausführungsmetriken sind für private und gemeinsam genutzte Workflows verfügbar.

HealthOmics verkauft diese Metriken unter der `cloudwatch.aws/omics` Umfang in Ihrem eigenen CloudWatch Konto.

Diese Metriken werden mit dem ausgegeben CloudWatch OpenTelemetry (Otel) -kompatibler Metrikstandard. Das bedeutet, dass Sie sie integrieren können mit OTel-compatible Observability-Tools zusammen mit nativen CloudWatch Dashboards und Alarme. Fragen Sie die OTel-Metriken mit Prometheus Query Language (PromQL) ab, um die Daten anzuzeigen und zu analysieren. Weitere Informationen finden Sie unter CloudWatch OpenTelemetry Metriken.

> ⚠ Verfügbarkeit in Regionen
>
> Run-Metriken sind in allen verfügbar unterstützt HealthOmics Regionen, mit Ausnahme der Region Israel (Tel Aviv) (`il-central-1`).

Metriken für einen Lauf aktivieren

Um diese Metriken zu veröffentlichen, müssen Sie AWS Identity and Access Management Die (IAM) -Rolle, die Sie für Ihren Lauf verwenden, muss die Berechtigung haben, Metriken in sie zu schreiben CloudWatch. Fügen Sie Ihrer Workflow-Ausführungsrolle die folgende Berechtigung hinzu:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "cloudwatch:PutMetricData",
      "Resource": "*"
    }
  ]
}
```

Weitere Informationen zu Berechtigungen finden Sie unter Servicerollen für AWS HealthOmics. Weitere Informationen zum Starten eines Laufs finden Sie unter Starte einen Lauf in HealthOmics.

> **CloudWatch `PutMetricData` Kontingent**
>
> CloudWatch hat eine Standardeinstellung `PutMetricData` Kontingent von 500 Anfragen pro Sekunde (Transaktionen pro Sekunde). Dieses Standardkontingent reicht aus, um Metriken für bis zu 15.000 gleichzeitig anzuzeigen HealthOmics Aufgaben. Um Kennzahlen für mehr als 15.000 gleichzeitig ablaufende Benutzer anzuzeigen HealthOmics Aufgaben, fordern Sie eine Erhöhung des Limits für CloudWatch `PutMetricData` Kontingent. Weitere Informationen finden Sie unter CloudWatch -Servicekontingente.

Verfügbare Metriken

> **Verfügbarkeit von Metriken**
>
> HealthOmics gibt Metriken basierend auf dem Workflow-Typ aus. Nicht jede Metrik wird für jeden Workflow ausgegeben.

In der folgenden Tabelle sind die Metriken aufgeführt, die HealthOmics verkauft sich bei jedem Lauf. Die Metriken sind verfügbar in CloudWatch Query Studio. Einheiten folgen dem OpenTelemetry Konvention: `By` ist Byte, `{cpu}` ist vCPUs, `{operation}` ist Betrieb und `%` ist ein Prozentsatz.

HealthOmics Metriken ausführen

Metrikname	Description	Einheit	Typ	Frequency (Frequenz)	Verfügbarkeit
`aws.omics.run.filesystem.usage`	Speicher, der auf dem gemeinsam genutzten Dateisystem des Laufs verwendet wird.	`By`	`gauge`	30 Sekunden	Für jeden Lauf.
`aws.omics.run.filesystem.limit`	Gesamtkapazität des gemeinsam genutzten Dateisystems des Laufs.	`By`	`gauge`	30 Sekunden	Nur für Läufe, die das verwenden `STATIC` Speichertyp ausführen.
`aws.omics.task.cpu.usage`	vCPUs, die von der Workflow-Aufgabe verwendet werden.	`{cpu}`	`gauge`	30 Sekunden	Für jede Aufgabe im Lauf.
`aws.omics.task.cpu.limit`	vCPUs, die für die Workflow-Aufgabe reserviert sind, basierend auf	`{cpu}`	`gauge`	30 Sekunden	Für jede Aufgabe im Lauf.

Metrikname	Description	Einheit	Typ	Frequency (Frequenz)	Verfügbarkeit
	Workflow-Definitionen oder Standardwerten.				
`aws.omics.task.memory.usage`	Speicher, der von der Workflow-Aufgabe verwendet wird.	By	`gauge`	30 Sekunden	Für jede Aufgabe in der Ausführung.
`aws.omics.task.memory.limit`	Für die Workflow-Aufgabe reservierter Speicher.	By	`gauge`	30 Sekunden	Für jede Aufgabe in der Ausführung.
`aws.omics.task.network.io`	Die Anzahl der Byte, die von der Workflow-Aufgabe übertragen und empfangen wurden. Geteilt von `network.io.direction` (`receive`, `transmit`).	By	`sum`	30 Sekunden	Für jede Aufgabe im Lauf.

Metrikname	Description	Einheit	Typ	Frequency (Frequenz)	Verfügbarkeit
`aws.omics.task.filesystem.io`	Die Anzahl der Dateisystem-Bytes, die von der Workflow-Aufgabe übertragen wurden. Geteilt von `filesystem.io.direction` (`read`, `write`).	By	`sum`	30 Sekunden	Für jede Aufgabe im Lauf.
`aws.omics.task.filesystem.operations`	Die Anzahl der Dateisystemoperationen, die von der Workflow-Aufgabe ausgeführt wurden. Geteilt von `filesystem.io.direction` (`read`, `write`).	`{operation}`	`sum`	30 Sekunden	Für jede Aufgabe im Lauf.

Metrikname	Description	Einheit	Typ	Frequency (Frequenz)	Verfügbarkeit
`aws.omics.task.filesystem.scratch.storage.usage`	Scratch-Speicher, der von der Workflow-Aufgabe verwendet wird.	By	`gauge`	30 Sekunden (`LOCAL` Modus) oder 20 Minuten (`SHARED` Modus)	Für jede Aufgabe im Lauf.
`aws.omics.task.filesystem.scratch.storage.limit`	Die gesamte Scratch-Speicherkapazität, die für die Workflow-Aufgabe verfügbar ist.	By	`gauge`	30 Sekunden	Für jede Aufgabe in einem Lauf, der Folgendes festlegt `scratchStorageMode` zu `LOCAL`.
`aws.omics.task.gpu.utilization`	GPU-Auslastung für die Workflow-Aufgabe. Ein Datenpunkt pro GPU, identifiziert durch `gpu.id`.	`%`	`gauge`	30 Sekunden	Nur für Aufgaben, die Beschleuniger verwenden.

Metrikname	Description	Einheit	Typ	Frequency (Frequenz)	Verfügbarkeit
`aws.omics.task.gpu.memory.usage`	GPU-Speicher, der von der Workflow-Aufgabe verwendet wird. Ein Datenpunkt pro GPU, identifiziert durch `gpu.id`.	By	`gauge`	30 Sekunden	Nur für Aufgaben, die Beschleuniger verwenden.
`aws.omics.task.gpu.memory.limit`	GPU-Speicher, der für die Workflow-Aufgabe verfügbar ist. Ein Datenpunkt pro GPU, identifiziert durch `gpu.id`.	By	`gauge`	30 Sekunden	Nur für Aufgaben, die Beschleuniger verwenden.

Weitere Informationen zu Beschleunigern finden Sie unter Aufgabenressourcen in einem HealthOmics Workflow-Definition.

Weitere Informationen zur kurzlebigen Speicherung finden Sie unter Kurzlebiger Speicher für Workflow-Aufgaben HealthOmics.

Gängige -Attribute

Jeder HealthOmics Eine Laufmetrik enthält einen gemeinsamen Satz von Ressourcenbeschriftungen, die die Quelle des Datenpunkts identifizieren.

Beschriftungen für Ressourcen

Label (Bezeichnung)	Description	Beispielwert
Allgemeine Ressourcenbezeichnungen		
`@resource.cloud.provider`	Der Cloud-Anbieter, der die Metrik veröffentlicht hat.	`aws`
`@resource.cloud.account.id`	Der AWS Konto, zu dem der Lauf gehört.	`123456789012`
`@resource.cloud.region`	Der AWS Region, in die der Lauf lief.	`us-west-2`
`@resource.cloud.resource_id`	Der ARN des Laufs.	`arn:aws:omics:us-west-2:123456789012:run/1234567`
`@resource.service.name`	Der Dienst, der die Metrik veröffentlicht hat.	`omics`
`@resource.aws.omics.workflow.id`	Die ID des Workflows, den der Lauf verwendet hat.	`1122334`
`@resource.aws.omics.run.id`	Die ID des Laufs.	`1234567`
`@resource.aws.omics.storage.type`	Der Laufspeichertyp.	`DYNAMIC`
Bezeichnungen der Ressourcen für Aufgaben		
`@resource.aws.omics.task.id`	Die ID der Aufgabe, für die der Datenpunkt bestimmt ist. Nur der `aws.omics.task.*` Metriken tragen dieses Label.	`1245938`

Zusätzliche Attribute

Einige Laufmetriken enthalten zusätzliche Datenpunktattribute, die die Metrik in separate Zeitreihen aufteilen. Sie können diese Attribute verwenden, um eine PromQL-Abfrage zu filtern.

Zusätzliche Metrikattribute

Zusätzliches Attribut	Metriken	Description	Werte
`gpu.id`	`aws.omics.task.gpu.utilization` , `aws.omics.task.gpu.memory.usage` , `aws.omics.task.gpu.memory.limit`	Der auf Null basierende Index der GPU auf der Instance.	`0`, `1`, `2`, oder `3`
`scratch.storage.mode`	`aws.omics.task.filesystem.scratch.storage.usage` , `aws.omics.task.filesystem.scratch.storage.limit`	Wo die Aufgabe ihre Scratch-Daten schreibt, basierend auf den effektiven `scratchStorageMode` .	`LOCAL` oder `SHARED`
`network.io.direction`	`aws.omics.task.network.io`	Die Richtung der Netzwerkübertragung.	`receive` oder `transmit`
`filesystem.io.direction`	`aws.omics.task.filesystem.io` , `aws.omics.task.fil`	Die Richtung des Dateisystembetriebs.	`read` oder `write`

Zusätzliches Attribut	Metriken	Description	Werte
	`esystem.operations`		

Abfragen HealthOmics Metriken ausführen

Sie können sich das ansehen HealthOmics führen Sie Metriken in der aus CloudWatch Konsole, indem Sie eine PromQL-Abfrage in Query Studio ausführen.

Zum Ansehen HealthOmics Metriken ausführen (CloudWatch Konsole)

1. Bestätigen Sie, dass Ihr Lauf mit dem begonnen hat `cloudwatch:PutMetricData` Erlaubnis.
2. Melden Sie sich bei der AWS-Managementkonsole an und öffnen Sie die CloudWatch -Konsole.
3. Wählen Sie im Navigationsbereich Query Studio.
4. Wählen Sie im Abfrage-Editor PromQL aus der Dropdownliste.
5. In Baumeister Modus, suchen Sie nach einem Metriknamen und seinen Beschriftungen und wählen Sie ihn aus. Oder in Herausgeber Modus, geben Sie eine PromQL-Abfrage ein.
6. Wählen Sie mit der Zeitbereichsauswahl einen Zeitraum aus.
7. Wählen Sie Lauf um die Ergebnisse als Zeitreihendiagramm anzuzeigen. Um die Darstellung des Diagramms zu ändern, wählen Sie Passen Sie an.

Die folgende Abfrage gibt beispielsweise die vCPUs zurück, die von jeder Aufgabe in einem Lauf verwendet werden. Ersetzen *runID* mit der ID des Laufs, den Sie überprüfen möchten.

```
{"aws.omics.task.cpu.usage", "@resource.aws.omics.run.id"="runID"}
```

Um jede Zeitreihe mit ihrer Lauf-ID und Task-ID zu beschriften, können Sie wählen Benutzerdefiniertes Etikett und geben Sie Folgendes ein.

```
{@resource.aws.omics.run.id="${@resource.aws.omics.run.id}",@resource.aws.omics.task.id="${@res
```

Zur Abfrage HealthOmics Metriken ausführen (API)

CloudWatch bietet Prometheus-compatible APIs und Endpunkte für die Abfrage von Metrikdaten. Weitere Informationen zum Abfragen von Metriken mit diesen APIs finden Sie unter Prometheus-compatible APIs.

Einen Alarm erstellen mit CloudWatch

Sie können eine erstellen CloudWatch Alarm aus einer PromQL-Abfrage, sodass CloudWatch benachrichtigt Sie, wenn eine Metrik einen Schwellenwert überschreitet. Der Alarm kann eine Benachrichtigung zu einem Thema von Amazon Simple Notification Service (Amazon SNS) senden oder eine weitere Aktion starten, wenn sich der Status des Alarms ändert.

Um einen Alarm aus einer PromQL-Abfrage zu erstellen (CloudWatch Konsole)

1. Melden Sie sich bei der AWS-Managementkonsole an und öffnen Sie die CloudWatch -Konsole.
2. Wählen Sie im Navigationsbereich Query Studio.
3. Wählen ProMQL, geben Sie Ihre Abfrage ein, überprüfen Sie das Diagramm und wählen Sie dann Alarm erstellen Knopf.

Hinzufügen HealthOmics Metriken zu einem ausführen CloudWatch Dashboard

Sie können eine PromQL-Abfrage zu einem hinzufügen CloudWatch Dashboard als Widget, damit Sie es überwachen können HealthOmics Ressourcennutzung zusammen mit Ihren anderen Kennzahlen.

Zum Hinzufügen HealthOmics führen Sie Metriken auf einen aus CloudWatch Dashboard (CloudWatch Konsole)

1. Melden Sie sich bei der AWS-Managementkonsole an und öffnen Sie die CloudWatch -Konsole.
2. Wählen Sie im Navigationsbereich Query Studio.
3. Wählen ProMQL, geben Sie Ihre Abfrage ein, und überprüfen Sie dann das Diagramm.
4. Wählen Sie Aktion, und dann wähle Zum Dashboard hinzufügen.
5. Wählen Sie ein vorhandenes Dashboard aus oder erstellen Sie ein neues und speichern Sie dann das Widget.

Für ein Dashboard müssen Sie nicht Widget für Widget erstellen. HealthOmics bietet ein Open-Source-Beispiel. Das HealthOmics Dashboard ausführen auf dem Sägeplatten GitHub Seitendiagramme CPU, Speicher, Dateisystem I/O, netzwerk I/O, gemeinsame Nutzung des Dateisystems und Scratch-Speicher für jeden Workflow, den Sie benennen. Sie können es mit dem Befehlszeilentool oder mit einem Klick bereitstellen AWS CloudFormation stapeln.

Analysieren von Laufmetriken mit dem HealthOmics MCP-Server

Sie können den verwenden HealthOmics MCP-Server (Model Context Protocol) zum Abrufen von Laufmetriken und zur Untersuchung von Lauffehlern in mehreren Datendimensionen mithilfe eines KI-Modells. Sie können den MCP-Server über Kiro CLI, Claude Code oder eine andere Methode verwenden MCP-compatible Ein agentischer Client. Weitere Informationen finden Sie unter AWS HealthOmics MCP-Server und unterstützte agentische Tools.

Callouts

- HealthOmics beginnt mit der Ausgabe von Run-Metriken, nachdem eine Aufgabe den `RUNNING` Status und sendet sie nicht mehr aus, nachdem die Aufgabe den `COMPLETED` Status. Weitere Informationen zum Aufgabenstatus finden Sie unter Werte für den Aufgabenstatus.
- Die ersten Datenpunkte erscheinen nach einer kurzen Verzögerung von etwa 30 Sekunden.
- Für Aufgaben, die weniger als 30 Sekunden ausgeführt werden, gibt es möglicherweise keine Metriken.
- Wenn der Speichertyp des Laufs lautet `DYNAMIC`, `aws.omics.run.filesystem.usage` kann zu einer Verzögerung von mehr als 30 Minuten kommen, sodass es für Läufe, die weniger als 30 Minuten dauern, möglicherweise nicht verfügbar ist.
- Wann `scratchStorageMode` ist `SHARED`, `aws.omics.task.filesystem.scratch.storage.usage` ist möglicherweise nicht für Aufgaben verfügbar, die eine große Anzahl temporärer Dateien erstellen. HealthOmics misst die Auslastung in diesem Modus mit einem rekursiven Scan des temporären Verzeichnisses der Aufgabe, und der Scan wird nicht immer innerhalb des Zeitlimits abgeschlossen, wenn die Anzahl der Dateien hoch ist.
- Die CPU- und Speichermesswerte können von den Ausführungsmanifestwerten abweichen, da die beiden Messungen einen anderen Bereich verwenden. Die Ausführungsmetriken geben besser wieder, was Ihre Aufgabe tatsächlich verbraucht.
- Ausführungsmetriken sind nur in der verfügbar AWS Konto, dem die Servicerolle gehört und das den Lauf startet.

Fakturierung

AWS HealthOmics berechnet Ihnen keine Run-Metriken. Metriken werden auf Amazon veröffentlicht CloudWatch in Ihrem Konto und CloudWatch stellt Ihnen die damit verbundene Aktivität direkt in Rechnung. Die Gebühren richten sich nach dem Umfang der aufgenommenen Metrikdaten. Für Preise in einer bestimmten AWS Region, siehe Amazonas CloudWatch Preisgestaltung. In der folgenden Tabelle wird beschrieben, wie CloudWatch Gebühren für Run-Metriken.

Wie CloudWatch Gebühren für Laufmetriken

Aktivität	Wie werden dir die Gebühren berechnet
Veröffentlichen OpenTelemetry Metriken	Pro GB aufgenommener Daten. Beinhaltet 15 Monate Speicherplatz, ohne dass für die Speicherung oder die Anzahl der einzelnen metrischen Reihen eine gesonderte Gebühr anfällt.
Ausführen von PromQL-Abfragen im CloudWatch Konsole, einschließlich Query Studio und Dashboards	Keine Gebühren.
Ausführen von PromQL-Abfragen mit dem CloudWatch APIs	Pro Million gescannter Proben.
Alarme, die eine PromQL-Abfrage auswerten	Eine Standardalarmgebühr zuzüglich Abfragegebühren für die bei jeder Auswertung gescannten Proben.

Abmeldung

In der Standardeinstellung HealthOmics veröffentlicht Laufmetriken immer dann, wenn die Servicerolle für einen Lauf die `cloudwatch:PutMetricData` Erlaubnis. Um sich abzumelden und zukünftige Gebühren zu stoppen, können Sie diese Berechtigung für die Servicerolle auf jeder Ausführungsebene weglassen. Um die Veröffentlichung auch dann zu beenden, wenn eine andere Richtlinie die Erlaubnis erteilt, fügen Sie der Rolle eine ausdrückliche Ablehnung hinzu:

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

```
  "Statement": [
    {
      "Effect": "Deny",
      "Action": "cloudwatch:PutMetricData",
      "Resource": "*"
    }
  ]
}
```

Eine andere Möglichkeit, sich abzumelden, besteht darin, die Protokollierung für einen Lauf durch folgende Einstellung zu deaktivieren `LogLevel = OFF` in der StartRun Anfrage. Wenn du festlegst `LogLevel` zu `OFF`, HealthOmics veröffentlicht keine Laufmetriken.

Überwachung HealthOmics mit CloudWatch Logs

HealthOmics generiert eine Vielzahl von Protokollen, die dir helfen, deine Läufe zu verstehen und Fehler zu beheben. Logs sind an zwei Stellen verfügbar: CloudWatch und Amazon S3.

Standardmäßig ist bei Läufen die Protokollierung aktiviert. Sie können die Protokollierung für einen Lauf optional deaktivieren, indem Sie `LogLevel = OFF` in der startrun Anfrage.

> **Note**
>
> Für Service-Updates konfigurieren und überwachen Sie Ihre Persönliches Gesundheits-Dashboard. Weitere Informationen zur Verwaltung des Dashboards finden Sie unter Erste Schritte mit Ihrem AWS Health Dashboard.

Themen

- Protokolltypen für HealthOmics Workflows
- Loggt sich ein CloudWatch
- Loggt sich in Amazon S3 ein
- Interaktiv CloudWatch Loggt sich in der CLI ein
- Zugreifen CloudWatch Protokolle von der Konsole

Protokolltypen für HealthOmics Workflows

HealthOmics stellt die folgenden Arten von Protokollen für Workflows bereit:

- Engine-Logs — Engine-Logs wie `.nextflow.log` und `workflow.log` werden von den zugrunde liegenden Workflow-Engines (Nextflow, WDL und CWL) produziert. Diese Protokolle können Ihnen bei der Behebung von Problemen mit der Workflow-Definition helfen.
- Ausführungsmanifestprotokolle — Diese Protokolle enthalten allgemeine Informationen zu jeder ausgeführten Aufgabe, wie z. B. Aufgabenstatus, Startzeit, Stoppzeit und Fehlergrund (falls die Aufgabe fehlgeschlagen ist).

  In den Ausführungsmanifestprotokollen werden auch Statistiken zur Ressourcenauslastung angezeigt, die für das Verständnis der Möglichkeiten zur Ressourcenoptimierung hilfreich sein können. Zu diesen Statistiken gehören:

  - Durchschnittlich CPUs
  - CPUs maximal
  - Reservierte CPUs
  - Reservierte GPUs
  - ErinnerungAverageGiB
  - ErinnerungMaximumGiB
  - ErinnerungReservedGiB
  - Laufende Sekunden

  Laufmetriken sind auch nahezu in Echtzeit verfügbar über CloudWatch OpenTelemetry Metriken. Weitere Informationen finden Sie unter Metriken für private Workflows ausführen.

  Informationen zum Erstellen eines Zeitplans für einen Lauf anhand seiner Task-Zeitstempel finden Sie unter Erstellen einer Aufgabenzeitleiste aus dem Run-Manifest.
- Ausführungsprotokolle — Ausführungsprotokolle enthalten den allgemeinen Ausführungsstatus und die Uhrzeit, zu der einzelne Aufgaben gestartet, ausgeführt, gestoppt und abgeschlossen werden. Ausführungsprotokolle geben Ihnen auch Einblick in die Schritte zum Import und Export von Dateien.
- Aufgabenprotokolle — Aufgabenprotokolle enthalten detaillierte Protokollinformationen zu einzelnen Aufgaben in Ihrem Lauf. Die Ausgaben in Ihrem Aufgabenprotokoll hängen von der Aufgabendefinition ab und davon, wo Sie Protokollanweisungen in Ihrem Code verwenden. Wenn Ihre Aufgabenprotokolle nicht den nötigen Einblick bieten, sollten Sie erwägen, Ihrer Aufgabendefinition zusätzliche Protokollanweisungen hinzuzufügen, um aussagekräftigere Aufgabenprotokolle zu erstellen.

- Cache-Protokolle ausführen — Mithilfe von Cache-Protokollen können Sie den Gesamtstatus der Ausführungs-Caches und das Zwischenspeichern der Aufgabenausgaben ermitteln. Mithilfe von Ausführungs-Cache-Protokollen erhalten Sie einen Überblick über Cache-Treffer und Fehlschläge bei jedem Lauf, der Caching verwendet.
- Outputs.json — Für WDL- und CWL-Workflows HealthOmics liefert eine von der Engine generierte Datei mit dem Namen `outputs.json`, nach Abschluss des Laufs in Ihren Amazon S3-Bucket. Diese Datei enthält eine Liste und eine Übersicht aller Ausgaben für den Lauf.

Loggt sich ein CloudWatch

CloudWatch generiert Workflow-Protokolle für fehlgeschlagene und erfolgreiche Läufe. Alle Protokolle sind für erfolgreiche und fehlgeschlagene Läufe verfügbar.

Sie finden die CloudWatch Workflow-Protokolle in der folgenden Protokollgruppe: `/aws/omics/WorkflowLog`. Außerdem die Ausgabe des Get-Run Der API-Betrieb bietet CloudWatch Logstream-ARNs für die Engine-Logs und Run-Logs.

In der Standardeinstellung AWS behält die CloudWatch Protokolliert auf unbestimmte Zeit. Sie können die Aufbewahrungsrichtlinie für die Protokollgruppe anpassen, um eine Aufbewahrungsfrist zwischen 10 Jahren und einem Tag festzulegen.

Die folgende Tabelle enthält eine Zusammenfassung der CloudWatch Loggt sich ein HealthOmics. Alle Workflow-Protokolle sind für erfolgreiche und fehlgeschlagene Läufe verfügbar.

Protokollnamen	Verfügbar in CloudWatch Logs	Wann ist das Protokoll verfügbar	Format des Log-Streams
Motorprotokolle	Ja	In Echtzeit	laufen/*runID*/motor
Führen Sie Manifestprotokolle aus	Ja	Nach Abschluss des Laufs	manifest/run/*runID*/*runUUID*
Führen Sie Protokolle aus	Ja	In Echtzeit	laufen/*runID*
Aufgaben-Protokolle	Ja	In Echtzeit	laufen/*runID*/aufgabe/*taskID*

Protokollnamen	Verfügbar in CloudWatch Logs	Wann ist das Protokoll verfügbar	Format des Log-Streams
Führen Sie Cache-Protokolle aus	Ja	In Echtzeit	Führen Sie Cache/aus*runCacheId* /*runCacheUUID*
Outputs.json (WDL und CWL)	Nein	n/a	n/a

Loggt sich in Amazon S3 ein

Unter den Protokolltypen für HealthOmics Workflows, nur die Motorprotokolle und `outputs.json` Dateien werden an Amazon S3 geliefert.

Nach Abschluss eines Laufs werden die Engine-Logs an Ihren S3-Bucket übermittelt und sind auf unbestimmte Zeit verfügbar, bis Sie sie löschen. Diese Protokolle befinden sich im Logs-Verzeichnis der S3-Ausgabe-URI, die Sie für den Workflow angegeben haben.

Der Pfad zum Protokollverzeichnis hat das folgende Format: `s3://{user_provided_path}/logs/`.

Die folgende Tabelle enthält eine Zusammenfassung der HealthOmics Protokolle, die in Ihrem Amazon S3-Bucket verfügbar sind.

Protokollnamen	Verfügbar in Amazon S3	Wann ist das Protokoll verfügbar	Stream-Pfad protokollieren
Engine-Protokolle	Ja	Nach Abschluss des Laufs	s3://*user_provided_path* /logs/engine.log
Outputs.json (WDL und CWL)	Ja	Nach Abschluss des Laufs	s3://*user_provided_path* /*runID*/*runUUID*/logs/outputs.json

Protokollnamen	Verfügbar in Amazon S3	Wann ist das Protokoll verfügbar	Stream-Pfad protokollieren
Führen Sie Manifestprotokolle, Ausführungsprotokolle und Aufgabenprotokolle aus	Nein	n/a	n/a

Interaktiv CloudWatch Loggt sich in der CLI ein

Sie können das interaktiv anzeigen CloudWatch Loggt mit dem Befehl Live Tail im interaktiven Modus. Sie können den Lauffortschritt in Echtzeit verfolgen und bis zu 5 Schlüsselwörter definieren, die in den Protokollen hervorgehoben werden sollen:

```
aws logs start-live-tail  \
  --mode interactive  \
  --log-group-identifiers arn:aws:logs:region:account-ID:log-group:/aws/omics/
WorkflowLog
```

Weitere Informationen finden Sie unter Live-Tail starten in der AWS CLI Befehlsreferenz.

Zugreifen CloudWatch Protokolle von der Konsole

Um auf die Protokolle für einen Lauf zuzugreifen, können Sie direkt von der Einzelheiten zur Ausführung Seite in HealthOmics Konsole.

1. Öffnen Sie die HealthOmics -Konsole.
2. Öffnen Sie bei Bedarf den linken Navigationsbereich (≡). Wählen Sie Läuft.
3. Wählen Sie den Lauf aus der Tabelle Runs aus.
4. Auf der Seite mit den Ausführungsdetails können Sie eine der folgenden Aktionen auswählen:
    a. Von Zusammenfassung ausführen, wähle Ausführungsprotokolle anzeigen. Die Konsole öffnet die Run-Logs im CloudWatch Konsole.
    b. Von Zusammenfassung ausführen, wähle Protokolle in Amazon S3 anzeigen. Die Konsole öffnet den Logs-Ordner in der Amazon S3-Konsole.

c. Von Aufgaben ausführen, wähle Protokolle anzeigen, Ausführungsprotokolle anzeigen oder Ausführungsmanifestprotokolle anzeigen für eine Aufgabe. Die Konsole öffnet die Protokolle im CloudWatch Konsole.

Sie können auch von der CloudWatch Konsole:

1. Öffne die CloudWatch Konsole https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im linken Menü Gruppen protokollieren.
3. Wählen Sie die `/aws/omics/WorkflowLog`-Gruppe aus.

   Wenn die Liste der Protokollgruppen lang ist, können Sie Folgendes eingeben Comics im Suchtextfeld, um die Liste einzugrenzen.
4. Wenn der Gruppendetails protokollieren Die Seite wird geöffnet. Wählen Sie den Log-Stream aus, den Sie anzeigen möchten. Die Konsole zeigt die Ereignisse für diesen Log-Stream an.

Protokollieren von AWS HealthOmics API-Aufrufen mit AWS CloudTrail

AWS HealthOmics ist in einen Dienst integriert AWS CloudTrail, der eine Aufzeichnung der Aktionen bereitstellt, die von einem Benutzer, einer Rolle oder einem AWS Dienst in ausgeführt wurden HealthOmics. CloudTrail erfasst alle API-Aufrufe HealthOmics als Ereignisse. Zu den erfassten Aufrufen gehören Aufrufe von der HealthOmics Konsole und Codeaufrufen für die HealthOmics API-Operationen. Wenn Sie einen Trail erstellen, können Sie die kontinuierliche Bereitstellung von CloudTrail Ereignissen an einen Amazon S3 S3-Bucket aktivieren, einschließlich Ereignissen für HealthOmics. Wenn Sie keinen Trail konfigurieren, können Sie die neuesten Ereignisse trotzdem in der CloudTrail Konsole im Ereignisverlauf anzeigen. Anhand der von gesammelten Informationen können Sie die Anfrage ermitteln CloudTrail, an die die Anfrage gestellt wurde HealthOmics, die IP-Adresse, von der aus die Anfrage gestellt wurde, wer die Anfrage gestellt hat, wann sie gestellt wurde, und weitere Details.

Weitere Informationen CloudTrail dazu finden Sie im AWS CloudTrail Benutzerhandbuch.

HealthOmics Informationen in CloudTrail

CloudTrail ist auf Ihrem aktiviert AWS-Konto , wenn Sie das Konto erstellen. Wenn eine Aktivität in stattfindet HealthOmics, wird diese Aktivität zusammen mit anderen AWS Serviceereignissen in der

CloudTrail Ereignishistorie in einem Ereignis aufgezeichnet. Sie können aktuelle Ereignisse in Ihrem anzeigen, suchen und herunterladen AWS-Konto. Weitere Informationen finden Sie unter Ereignisse mit dem CloudTrail Ereignisverlauf anzeigen.

Für eine fortlaufende Aufzeichnung der Ereignisse in Ihrem AWS-Konto, einschließlich der Ereignisse für HealthOmics, erstellen Sie einen Trail. Ein Trail ermöglicht CloudTrail die Übermittlung von Protokolldateien an einen Amazon S3 S3-Bucket. Wenn Sie einen Trail in der Konsole anlegen, gilt dieser für alle AWS-Regionen-Regionen. Der Trail protokolliert Ereignisse aus allen Regionen der AWS Partition und übermittelt die Protokolldateien an den von Ihnen angegebenen Amazon S3 S3-Bucket. Darüber hinaus können Sie andere AWS Dienste konfigurieren, um die in den CloudTrail Protokollen gesammelten Ereignisdaten weiter zu analysieren und darauf zu reagieren. Weitere Informationen finden Sie hier:

- Übersicht zum Erstellen eines Trails
- CloudTrail unterstützte Dienste und Integrationen
- Konfiguration von Amazon SNS SNS-Benachrichtigungen für CloudTrail
- Empfangen von CloudTrail Protokolldateien aus mehreren Regionen und Empfangen von CloudTrail Protokolldateien von mehreren Konten

Alle HealthOmics Aktionen werden von der AWS HealthOmics API-Referenz protokolliert CloudTrail und sind in dieser dokumentiert. Beispielsweise generieren Aufrufe von `StartVariantImportJob` und `CreateWorkflow` Aktionen Einträge in den CloudTrail Protokolldateien. `CreateReferenceeStore`

Jeder Ereignis- oder Protokolleintrag enthält Informationen zu dem Benutzer, der die Anforderung generiert hat. Die Identitätsinformationen unterstützen Sie bei der Ermittlung der folgenden Punkte:

- Ob die Anfrage mit IAM-Benutzeranmeldedaten gestellt wurde.
- Gibt an, ob die Anforderung mit temporären Sicherheitsanmeldeinformationen für eine Rolle oder einen Verbundbenutzer gesendet wurde.
- Ob die Anfrage von einem anderen AWS Dienst gestellt wurde.

Weitere Informationen finden Sie unter CloudTrail -Element userIdentity.

HealthOmics Logdateieinträge verstehen

Ein Trail ist eine Konfiguration, die die Übertragung von Ereignissen als Protokolldateien an einen von Ihnen angegebenen Amazon S3 S3-Bucket ermöglicht. CloudTrail Protokolldateien enthalten einen oder mehrere Protokolleinträge. Ein Ereignis stellt eine einzelne Anforderung aus einer beliebigen Quelle dar und enthält Informationen über die angeforderte Aktion, Datum und Uhrzeit der Aktion, Anforderungsparameter usw. CloudTrail Protokolldateien sind kein geordneter Stack-Trace der öffentlichen API-Aufrufe, sodass sie nicht in einer bestimmten Reihenfolge angezeigt werden.

Das folgende Beispiel zeigt einen CloudTrail Protokolleintrag, der die CreateWorkflow Aktion demonstriert.


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROAIU53LOGOMTOPXXNPG:username",
        "arn": "arn:aws:sts::account:assumed-role/admin/username",
        "accountId": "account-id",
        "accessKeyId": "accessKeyId",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAIU53LOGOMTOPXXNPG",
                "arn": "arn:aws:iam::account:role/admin",
                "accountId": "account",
                "userName": "admin"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2022-07-23T18:26:09Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2022-07-23T18:46:42Z",
    "eventSource": "omics.amazonaws.com",
    "eventName": "CreateWorkflow",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "205.251.233.176",
    "userAgent": "aws-cli/1.22.45 Python/3.9.13 Darwin/20.6.0 botocore/1.23.45",
    "requestParameters": {
```

```
        "name": "parameter_name",
        "definitionZip": "czM6Ly93b3JrZmxvd2RlZi1oZWxsby9kZWZpbml0aW9uLnppcA==",
        "requestId": "d788a73c-b81b-45fb-a8a6-d8bb4449ec8a"
    },
    "responseElements": {
        "id": "1002571",
        "arn": "arn:aws:omics:us-west-2:555555555555:instance/i-b188560f ",
        "status": "CREATING",
        "tags": {
            "resourceArn": "arn:aws:omics:us-west-2:083685709690:workflow/1002571"
        }
    },
    "requestID": "842d731d-f264-4b08-a2c9-2f7d45e1eaa3",
    "eventID": "76872ca2-f208-4193-807d-7dd7ea34e6b2",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "083685709690",
    "eventCategory": "Management"
}
```


Verwenden EventBridge mit AWS HealthOmics

HealthOmics sendet Ereignisse an Amazon EventBridge wenn Ressourcen ihren Status ändern. Zu den Ressourcen gehören Importaufträge, Exportaufträge, gemeinsam genutzte Ressourcen, Workflows, Aufgaben und Ausführungen. Für jeden Ressourcentyp gibt es eine Liste von Statusänderungen, die ein Ereignis auslösen.

Ein Event-Bus ist ein Router, der Ereignisse empfängt und sie an Ziele weiterleitet. Ihr Konto beinhaltet einen Standard-Event-Bus, der automatisch Ereignisse von empfängt AWS Dienstleistungen. Sie können zusätzliche benutzerdefinierte Event-Busse erstellen.

Sie erstellen EventBridge Regeln zur Angabe der Aktionen, die ausgeführt werden sollen, wenn der Event-Bus Ereignisse empfängt. Sie können beispielsweise eine Regel erstellen, die Sie über Statusänderungen für eine Ressource informiert.

Zu den gängigen Szenarien für die Verwendung von Ereignissen gehören:

- Um zu überwachen, wann ein Benutzer eine Ressource mit Ihnen teilt oder die gemeinsame Nutzung widerruft.
- Um zu überwachen, ob ein Lauf fehlschlägt oder erfolgreich abgeschlossen wird.

Für weitere Informationen zur Verwendung EventBridge, siehe Was ist Amazon EventBridge?

Themen

- Aufstellen EventBridge zum HealthOmics
- EventBridge Ereignisse in HealthOmics
- Struktur von Ereignismeldungen
- Beispiele für Event-Meldungen

Aufstellen EventBridge zum HealthOmics

Bevor Sie überwachen können für EventBridge Ereignisse, erstellen Sie eine EventBridge Suche und erstelle Regeln für die Ereignisse, die dich interessieren.

Konfiguriere eine EventBridge Bus

Sie können den Standard-Event-Bus für Ihren verwenden AWS-Konto oder konfigurieren Sie einen benutzerdefinierten Event-Bus. Gehen Sie folgendermaßen vor, um einen benutzerdefinierten Event-Bus zu konfigurieren:

1. Öffnen Sie den EventBridge Konsole: https://console.aws.amazon.com/events/.
2. Wählen Sie in der linken Navigationsleiste Busse für Veranstaltungen.
3. Wählen Sie Event Bus erstellen aus.
4. In der Event-Bus erstellen Formular, geben Sie einen Namen für den Bus ein.
5. Wählen Sie Erstellen um den Bus zu erstellen.

Erstellen Sie eine EventBridge Regel

Das folgende Verfahren zeigt, wie eine einfache Regel erstellt wird. Weitere Informationen zu Regeln finden Sie unter Regeln in EventBridge.

1. Öffne das EventBridge Konsole: https://console.aws.amazon.com/events/.
2. Klicken Sie im linken Navigationsbereich auf die Option Regeln.
3. Wählen Sie Regel erstellen aus. Die Konsole öffnet den Regel erstellen Formular.
4. In Regeldetails definieren, geben Sie einen Namen für die Regel ein.

    - Für Name, geben Sie einen Namen für den Bus ein.

- Für Eventbus, wählen Sie den Bus für diese Regel aus.
- Wählen Sie Weiter aus.

5. In Event-Muster erstellen, unter Quelle des Ereignisses ausgewählt AWS-Ereignisse oder EventBridge Veranstaltungen von Partnern.
6. Scrollen Sie nach unten zu Muster des Ereignisses.

   a. Für Quelle des Ereignisses, wählen AWS-Dienste.
   b. Für AWS-Service, geben Sie ein Comics im Textfilter und wähle AWS HealthOmics als Dienst.
   c. Für Art der Veranstaltung wählen Sie das Ereignis aus, das Sie interessiert (oder Alle Ereignisse).
   d. Wählen Sie Weiter aus.

7. Hinein Wählen Sie Ziel (e), wählen Sie ein Ziel für das Ereignis aus. Wählen Sie zum Beispiel AWS-Service, sie wählten CloudWatch Gruppe protokollieren, und konfigurieren Sie eine Protokollgruppe.

   Für viele Zieltypen benötigt EventBridge die Berechtigung zum Senden von Ereignissen an das Ziel. Die Konsole erstellt diese Berechtigungen für Sie.
8. (Optional) In Konfigurieren Sie die Tags, ordnen Sie der Regel Stichwörter zu.
9. In Überprüfen und aktualisieren, überprüfen Sie die Konfiguration und wählen Sie Regel erstellen.

EventBridge Ereignisse in HealthOmics

In der folgenden Tabelle sind die Ereignisse aufgeführt, die HealthOmics sendet an EventBridge, und die Liste der möglichen Statuswerte für das Ereignis.

Ereignisname	Mögliche Statuswerte
Änderung des Auftragsstatus beim Import von Anmerkungen	Übermittelt, in Bearbeitung, storniert, abgeschlossen, fehlgeschlagen oder mit Fehlern abgeschlossen
Statusänderung im Annotationsspeicher teilen	Ausstehend, aktivierend, aktiv, löschend, gelöscht, fehlgeschlagen

Ereignisname	Mögliche Statuswerte
Statusänderung des Annotationsspeichers	Das Erstellen, Erstellen, Aktualisieren, Aktualisieren, Löschen, Löschen, Löschen oder Erstellen ist fehlgeschlagen
Lesen Sie die Statusänderung des Aktivierungsauftrags festlegen	Übermittelt, in Bearbeitung, abgeschlossen, fehlgeschlagen oder mit Fehlern abgeschlossen
Lesen Sie „Statusänderung des Exportauftrags festlegen“	Übermittelt, in Bearbeitung, abgeschlossen, fehlgeschlagen oder mit Fehlern abgeschlossen
Lesen Sie „Statusänderung des Importauftrags festlegen“	Übermittelt, in Bearbeitung, abgeschlossen, fehlgeschlagen oder mit Fehlern abgeschlossen
Lesen Sie „Statusänderung festlegen“	Upload wird verarbeitet, Upload ist fehlgeschlagen, aktiv, archiviert, aktiviert oder gelöscht
Änderung des Status des Referenzimportauftrags	Übermittelt, in Bearbeitung, abgeschlossen, fehlgeschlagen oder mit Fehlern abgeschlossen
Änderung des Referenzstatus	Aktiv oder gelöscht
Änderung des Status des Referenzspeichers	Erstellt, aktualisiert, aktiv oder gelöscht
Statusänderung ausführen	Ausstehend, wird gestartet, läuft, wird gestoppt, abgeschlossen, gelöscht, fehlgeschlagen oder abgebrochen
Änderung des Sequenzspeicher-Status	Erstellt, aktualisiert, aktiv oder gelöscht
Änderung des Aufgabenstatus	Ausstehend, wird gestartet, ausgeführt, gestoppt, abgeschlossen, gelöscht, fehlgeschlagen oder storniert

Ereignisname	Mögliche Statuswerte
Änderung des Auftragsstatus für den Varianten import	Übermittelt, in Bearbeitung, storniert, abgeschlossen, fehlgeschlagen oder mit Fehlern abgeschlossen
Änderung des Status von Variant Store Share	Ausstehend, aktivierend, aktiv, löschend, gelöscht, fehlgeschlagen
Änderung des Status des Variantenspeichers	Das Erstellen, Erstellen, Aktualisieren, Aktualisieren, Löschen, Löschen, Löschen oder Erstellen ist fehlgeschlagen
Status der Workflow-Freigabe ändern	Ausstehend, aktivierend, aktiv, löschend, gelöscht, fehlgeschlagen
Änderung des Workflow-Status	Erfolgreiche Erstellung, Fehler bei der Erstellung, erfolgreicher Löschvorgang oder fehlgeschlagener Löschvorgang

Struktur von Ereignismeldungen

HealthOmics bietet nach besten Kräften die Möglichkeit, Meldungen zu Zustandsänderungen zu senden an EventBridge. Das Ereignis ist ein Objekt mit JSON-Struktur, das auch Metadatendetails enthält. Sie können die Metadaten als Eingabe verwenden, um entweder das Ereignis neu zu erstellen oder um weitere Informationen zu erhalten. Ereignisse umfassen die folgenden Felder:

- `version` — Derzeit 0 (Null) für alle Ereignisse.
- `id` — Eine UUID der Version 4, die für jedes Ereignis generiert wird.
- `detail-type` — Die Art des Ereignisses, das gesendet wird.
- `account` — Die 12-stellige Zahl AWS-Konto ID des Bucket-Besitzers.
- `source` — Identifiziert den Dienst, der das Ereignis generiert hat.
- `time` — Die Uhrzeit, zu der das Ereignis eingetreten ist.
- `region` — Identifiziert den AWS-Region des Eimers.
- `resources` — Ein JSON-Array, das den Amazon-Ressourcennamen (ARN) des Buckets enthält.
- `detail` — Ein JSON-Objekt, das Informationen über das Ereignis enthält.

Run-Ereignisse enthalten die folgenden Felder:

- `uuid` — Die allgemein eindeutige Kennung für den Lauf.
- `workflowId` — Workflow-ID des Workflows, der mit diesem Lauf verknüpft ist.
- `workflowName` — Name des Workflows, der mit diesem Lauf verknüpft ist.
- `runId` — Lauf-ID.
- `runName` — Name der Ausführung.
- `runOutputUri` — Die URI, in die der Lauf seine Ausgabedaten schreibt.

Beispiele für Event-Meldungen

Das folgende Beispiel ist ein Ereignis für eine Änderung des Ausführungsstatus und zeigt die zusätzlichen Felder.

```
{
    "version":"0",
    "id":"c0e540f4-df38-b986-86c1-3e3730f971fe",
    "detail-type":"Run Status Change",
    "source":"aws.omics",
    "account":"123456789012",
    "time":"2022-10-20T22:07:35Z",
    "region":"us-west-2",
    "resources":[
        "arn:aws:omics:us-west-2:123456789012:run/2101313"
    ],
    "detail":{
        "omicsVersion":"1.0.0",
        "arn":"arn:aws:omics:us-west-2:123456789012:run/2101313",
        "status":"COMPLETED",
        "uuid":"153893cd-097a-40ec-aec7-838a97cd2b21",
        "runId": "1234567",
        "runName": "run name",
        "runOutputUri": "s3://amzn-s3-demo-bucket/run-output/2101313",
        "workflowId": "1234567",
        "workflowName": "workflow name"
    }
}
```

Das folgende Beispiel ist ein Ereignis für eine Änderung des Aufgabenstatus.

```
{
    "version": "0",
    "id": "718d6817-c868-26d3-8ef0-0dc9b2ac73f4",
    "detail-type": "Task Status Change",
    "source": "aws.omics",
    "account": "123456789012",
    "time": "2024-10-30T09:05:44Z",
    "region": "us-west-2",
    "resources": ["arn:aws:omics:us-west-2:123456789012:task/8888888"],
    "detail": {
        "omicsVersion": "1.0.0",
        "arn": "arn:aws:omics:us-west-2:123456789012:task/8888888",
        "status": "COMPLETED",
        "runArn": "arn:aws:omics:us-west-2:123456789012:run/2101313",
        "runUuid": "153893cd-097a-40ec-aec7-838a97cd2b21",
        "runId": "1234567",
        "runName": "run name",
        "workflowId": "1234567",
        "workflowName": "workflow name",
        "taskName": "task name",
        "cpus": "4",
        "gpus": "0",
        "memory": "16",
        "instanceType": "omics.c.xlarge",
        "creationTime": "2024-10-30T09:01:12.345Z",
        "startTime": "2024-10-30T09:03:20.123Z",
        "stopTime": "2024-10-30T09:05:44.678Z"
    }
}
```


Das Folgende ist ein Beispiel für ein Ereignis bei einer Änderung des Readset-Status.


```
{
  "version": "0",
  "id": "64ca0eda-9751-dc55-c41a-1bd50b4fc9b7",
  "detail-type": "Read Set Status Change",
  "source": "aws.omics",
  "account": "123456789012",
  "time": "2023-04-04T17:53:06Z",
  "region": "us-west-2",
  "resources": ["arn:aws:omics:us-west-2:123456789012:sequenceStore/1234567890/
readSet/3456789012"],
  "detail": {
```

```
     "omicsVersion": "1.0.0",
     "arn": "arn:aws:omics:us-west-2:123456789012:sequenceStore/1234567890/
readSet/3456789012",
     "sequenceStoreId" : "1234567890",
     "id": "3456789012",
     "status": "PROCESSING_UPLOAD"
  }
}
```

Ein ähnliches Ereignis wird für einen Variantenspeicher-Importauftrag erstellt.

```
{
    "version": "0",
    "id": "6a7e8feb-b491-4cf7-a9f1-bf3703467718",
    "detail-type": "Variant Store Status Change",
    "source": "aws.omics",
    "account": "123456789012",
    "time": "2015-12-22T18:43:48Z",
    "region": "us-east-1",
    "resources": ["arn:aws:omics:us-east-1:123456789012:myvariantstore2"],
    "detail": {
       "omicsVersion": "1.0.0",
       "arn": "arn:aws:omics:us-east-1:123456789012:myvariantstore2",
       "status": "CREATED",
       "storeId": "6710c5f02610",
       "storeName": "myvariantstore2"
    }
}
```

Das Folgende ist ein Ereignis für eine Änderung des Importauftragsstatus.

```
{
    "version": "0",
    "id": "6a7e8feb-b491-4cf7-a9f1-bf3703467718",
    "detail-type": "Variant Import Job Status Change",
    "source": "aws.omics",
    "account": "123456789012",
    "time": "2015-12-22T18:43:48Z",
    "region": "us-east-1",
    "resources": ["arn:aws:omics:us-east-1:123456789012:my_variant_store/
b64ea9a3-459f-4b68-92c3-3ddb83209fe9"],
    "detail": {
       "omicsVersion": "1.0.0",
```

```
        "arn": "arn:aws:omics:us-east-1:123456789012:my_variant_store/
b64ea9a3-459f-4b68-92c3-3ddb83209fe9",
        "status": "COMPLETED",
        "jobId": "b64ea9a3-459f-4b68-92c3-3ddb83209fe9",
        "storeId": "a74869f91e20",
        "storeName": "my_variant_store"
    }
}
```

Fehlerbehebung

Die folgenden Themen können Ihnen bei der Behebung von Problemen helfen, die bei der Verwendung von HealthOmics Workflows und Datenspeicher.

Themen

- Problembehandlung bei Workflows
- Fehlerbehebung bei Laufmetriken
- Behebung von Problemen beim Zwischenspeichern von Anrufen
- Problembehandlung bei Datenspeichern
- Problembehandlung mit Kiro CLI

Problembehandlung bei Workflows

Themen

- Wie behebe ich Fehler bei einem fehlgeschlagenen Lauf?
- Wie behebe ich eine fehlgeschlagene Aufgabe?
- Wo finde ich die Engine-Logs?
- Wie kann ich die Größe der Eingabeparameter für einen Workflow reduzieren?
- Warum wird mein Lauf nicht abgeschlossen?
- Warum ist mein Lauf mit INSTANCE_RESERVATION_FAILED oder ALL_PROFILES_INSTANCE_RESERVATION_FAILED fehlgeschlagen?

Wie behebe ich Fehler bei einem fehlgeschlagenen Lauf?

Benutze das GetRun API-Vorgang, um den Fehlergrund abzurufen. Weitere Informationen finden Sie unter Gründe für Ausführungsfehler.

Den Rest der Sequenz und die Befehle für jeden Schritt finden Sie unter Debuggen eines HealthOmics Workflow-Ausführung.

Wie behebe ich eine fehlgeschlagene Aufgabe?

Überprüfen Sie den Fehlercode in der Meldung über den Ausfall der Aufgabe, um den Fehler zu verstehen. Überprüfen Sie die Task-Logins CloudWatch um detaillierte Protokollierungsmeldungen für die Aufgabe zu sehen. Wenn Sie keine detaillierten Protokollmeldungen erhalten, können Sie Ihren Workflow so ändern, dass er zusätzliche Protokollanweisungen ausgibt. Weitere Informationen finden Sie unter Überwachung HealthOmics mit CloudWatch Logs.

Wo finde ich die Engine-Logs?

HealthOmics veröffentlicht Motorprotokolle unter CloudWatch nahezu in Echtzeit für alle Läufe (erfolgreich und fehlgeschlagen). Engine-Logs werden nach Abschluss des Laufs auch an Ihren Amazon S3-Bucket übermittelt. Weitere Informationen erhalten Sie unter Überwachung HealthOmics mit CloudWatch Logs und Loggt sich in Amazon S3 ein.

Wie kann ich die Größe der Eingabeparameter für einen Workflow reduzieren?

Sie können bis zu 256 KB an Eingabeparametern für einen Workflow angeben. Sie können Verzeichnisimporte oder Musterblätter verwenden, um diese Größenbeschränkung einzuhalten. Weitere Informationen finden Sie unter Verwaltung der Größe von Ausführungsparametern.

Warum wird mein Lauf nicht abgeschlossen?

Wenn es Probleme mit Ihrem Code gibt und die Prozesse nicht ordnungsgemäß beendet wurden, reagiert Ihr Lauf möglicherweise nicht mehr oder „hängt“. Weitere Informationen darüber, wie Sie nicht reagierende Läufe verhindern und abfangen können, finden Sie unter Hinweise für Läufe, die nicht reagieren.

Warum ist mein Lauf mit INSTANCE_RESERVATION_FAILED oder ALL_PROFILES_INSTANCE_RESERVATION_FAILED fehlgeschlagen?

GPU-Instances sind Ressourcen, die möglicherweise nicht immer sofort in Ihrem bevorzugten Format verfügbar sind AWS Region. Weitere Informationen zur Verbesserung des Erfolgs der GPU-Akquisition finden Sie unter Behebung von INSTANCE_RESERVATION_FAILED-Fehlern.

Wenn du das benutzt `omicsResourceFallbackOrder` Direktive und dein Lauf schlägt fehl mit ALL_PROFILES_INSTANCE_RESERVATION_FAILED, keiner der von Ihnen angegebenen Beschleuniger hatte innerhalb der Wartezeiten eine Kapazität. Erwägen Sie, die Wartezeit für

Ressourcen zu verlängern, oder geben Sie alternative Beschleuniger oder eine CPU-Option in Ihrer Ersatzreihenfolge an. Weitere Informationen finden Sie unter Erweiterte Ressourcenkonfiguration.

Fehlerbehebung bei Laufmetriken

Themen

- Warum läuft meine Aufgabe langsamer als erwartet?
- Warum ist mein Lauf fehlgeschlagen, weil der Speicherplatz knapp wurde?
- Wie dimensioniere ich die Rechenleistung und den Speicher für meinen Workflow richtig?
- Warum sehe ich keine Metriken für meinen Lauf?
- Wie unterscheidet sich das von den Ressourcenstatistiken für das Run-Manifestprotokoll?
- Wie unterscheiden sich die Ausführungsmetriken in Query Studio vom Nextflow-Ausführungsbericht?

Warum läuft meine Aufgabe langsamer als erwartet?

Eine Aufgabe kann langsam ausgeführt werden, wenn sie an die Central Processing Unit (CPU) oder an die Graphics Processing Unit (GPU) gebunden ist oder wartet input/output (I/O). Nutzen Sie die nahezu in Echtzeit CloudWatch Metriken für den Lauf, um die Ressourcenauslastung zu sehen, während die Aufgabe noch ausgeführt wird:

- Vergleiche `aws.omics.task.cpu.usage` gegen `aws.omics.task.cpu.limit`. Die anhaltende Nutzung des Grenzwerts zeigt an, dass die Aufgabe CPU-bound und kann von mehr zugewiesener CPU profitieren.
- Informationen zu GPU-Workloads finden Sie unter `aws.omics.task.gpu.utilization`. Eine geringe Auslastung kann darauf hinweisen, dass die Aufgabe die GPU nicht effektiv nutzt.
- Prüfen `aws.omics.task.filesystem.io` und `aws.omics.task.filesystem.operations` zum I/O Engpässe.

Weitere Informationen finden Sie unter Metriken für private Workflows ausführen.

Warum ist mein Lauf fehlgeschlagen, weil der Speicherplatz knapp wurde?

Ein Lauf kann fehlschlagen, wenn der Speicherplatz auf dem gemeinsam genutzten Dateisystem erschöpft ist. Vergleiche `aws.omics.run.filesystem.usage`, wobei der verwendete

Speicher auf dem gemeinsam genutzten Dateisystem des Laufs gemeldet wird, gegen `aws.omics.run.filesystem.limit`, das seine Gesamtkapazität angibt. Sie können Dashboards oder Alarme erstellen, um die Speichernutzung zu überwachen. Weitere Informationen finden Sie unter Metriken für private Workflows ausführen.

Wie dimensioniere ich die Rechenleistung und den Speicher für meinen Workflow richtig?

Führen Sie Ihren Workflow einmal aus und überprüfen Sie die CloudWatch Kennzahlen zum Vergleich der tatsächlichen Auslastung mit den zugewiesenen Grenzwerten:

- Wenn die CPU-, GPU- oder Speicherauslastung deutlich unter dem Grenzwert bleibt, können Sie die Zuweisung reduzieren, um die Kosten zu senken.
- Wenn die Auslastung konstant am Limit liegt, erhöhen Sie die Zuweisung, um die Leistung zu verbessern oder Ausfälle zu vermeiden.
- Verwenden `aws.omics.run.filesystem.usage` um eine geeignete Dateisystemgröße für den Lauf festzulegen.

Weitere Informationen finden Sie unter Metriken für private Workflows ausführen.

Warum sehe ich keine Metriken für meinen Lauf?

Wenn die Lauf- und Aufgabenmetriken nicht in der `cloudwatch.aws/omics` Umfang in CloudWatch, überprüfen Sie Folgendes:

- Vergewissern Sie sich, dass die für den Lauf verwendete IAM-Rolle berechtigt ist, Metriken zu veröffentlichen CloudWatch.
- Die Metriken werden während der Ausführung des Laufs nahezu in Echtzeit veröffentlicht. Warten Sie mit einer kurzen Verzögerung (etwa 30 Sekunden), bis die ersten Datenpunkte angezeigt werden.
- Für Aufgaben, die weniger als 30 Sekunden ausgeführt werden, gibt es möglicherweise keine Metriken.
- Vergewissern Sie sich, dass Sie die richtige Datei anzeigen AWS Region und Konto.

Weitere Informationen finden Sie unter Metriken für private Workflows ausführen.

Wie unterscheidet sich das von den Ressourcenstatistiken für das Run-Manifestprotokoll?

In den Ausführungsmanifestprotokollen werden aggregierte Statistiken für einen abgeschlossenen Lauf gemeldet, z. B. die maximale und durchschnittliche CPU- und Speichermenge, sodass Sie einen abgeschlossenen Lauf auf Optimierungsmöglichkeiten überprüfen können. CloudWatch Bei den Ausführungsmetriken handelt es sich um eine Zeitreihe: HealthOmics veröffentlicht Datenpunkte mit der Häufigkeit, die unter angegeben ist Verfügbare Metriken, sodass Sie Auslastungsänderungen während der Ausführung überwachen und feinere Daten zur Fehlerbehebung verwenden können. Weitere Informationen zu Laufmetriken finden Sie unter Metriken für private Workflows ausführen.

Wie unterscheiden sich die Ausführungsmetriken in Query Studio vom Nextflow-Ausführungsbericht?

Beide berichten über die Ressourcenauslastung für Ihren Workflow. Sie unterscheiden sich in Bezug auf den Zeitpunkt, die Motorabdeckung und die Art und Weise, wie Sie sie verwenden. Verwenden Sie Laufmetriken, um einen laufenden Lauf zu überwachen, und verwenden Sie den Nextflow-Ausführungsbericht, um einen abgeschlossenen Lauf zu überprüfen.

- Laufmetriken sind nahezu in Echtzeit verfügbar, während der Lauf ausgeführt wird, sodass du den Fortschritt überwachen, Alarme einrichten und Fehler bei einem laufenden Lauf beheben kannst. Der Nextflow-Ausführungsbericht wird erstellt, wenn der Lauf abgeschlossen ist. Sie können ihn also nur verwenden, um einen Lauf im Nachhinein zu überprüfen.
- HealthOmics gibt Laufmetriken für jeden privaten Workflow-Lauf aus, unabhängig davon, welche Workflow-Engine der Lauf verwendet. Der Ausführungsbericht ist spezifisch für Nextflow und HealthOmics exportiert ihn nur, wenn Sie ihn so konfigurieren, dass er unter geschrieben wird `/mnt/workflow/output/`.
- Die Laufmetriken sind CloudWatch Zeitreihen, sodass Sie sie mit PromQL abfragen, sie alarmieren und sie einem Dashboard hinzufügen können. Der Ausführungsbericht ist eine statische Datei im Amazon S3-Ausgabespeicherort Ihres Laufs.

Weitere Informationen zum Ausführungsbericht finden Sie unter Generieren Sie Nextflow-Ausführungsberichte. Weitere Informationen zu Run-Metriken finden Sie unter Metriken für private Workflows ausführen.

Behebung von Problemen beim Zwischenspeichern von Anrufen

Die folgenden Themen können Ihnen bei der Behebung von Problemen helfen, die beim Zwischenspeichern von Anrufen auftreten.

Themen

- Warum wird mein Lauf nicht im Cache gespeichert?
- Warum verwendet eine Aufgabe den Cache-Eintrag nicht?
- Warum ist das Caching von Aufrufen für eine Aufgabe deaktiviert?

Warum wird mein Lauf nicht im Cache gespeichert?

1. Vergewissern Sie sich, dass der Lauf so konfiguriert ist, dass er einen Cache verwendet, indem Sie das CacheID-Feld in der GetRun Antwort auf den API-Vorgang. Führen Sie mithilfe der CLI den folgenden Befehl aus: `aws omics get-run –id <run_id>`.
2. Wenn der Lauf erfolgreich war, überprüfen Sie das Cacheverhalten, das im GetRun Die Antwort lautet CACHE_ALWAYS. Wenn das Cacheverhalten auf CACHE_ON_FAILURE gesetzt ist, werden Läufe nur dann im Cache gespeichert, wenn sie fehlschlagen.

Warum verwendet eine Aufgabe den Cache-Eintrag nicht?

In der `/aws/omics/WorkflowLog` CloudWatch Öffnen Sie den Log-Stream für den Run-Cache: <cache_id>RunCache//<cache_uuid>.

1. Stellen Sie sicher, dass bei einer vorherigen Ausführung ein Cacheeintrag für die Aufgabe erstellt wurde, von der Sie erwartet hatten, dass sie zwischengespeichert wird. Läufe, die im Cache gespeichert wurden, werden mit der Logmeldung CACHE_ENTRY_CREATED aufgezeichnet.
2. Suchen Sie das CACHE_MISS-Protokoll für die Aufgabe und führen Sie die abgeschlossene Aufgabe aus. Wenn kein Protokolleintrag vorhanden ist, überprüfen Sie, ob der Lauf für die Verwendung des Caches konfiguriert wurde.
3. Wenn ein Cache-Eintrag erstellt wurde, stellen Sie sicher, dass die CPUs, der Arbeitsspeicher, die GPUs und der Container-Digest für beide Aufgaben identisch sind. Die Task-ARN für die Aufgabe, die den Cache-Eintrag erstellt hat, befindet sich in der Protokollnachricht.
4. Wenn die Rechenanforderungen für beide Aufgaben übereinstimmen, stellen Sie sicher, dass sich die Eingaben zwischen den Aufgaben nicht geändert haben. Öffnen Sie dazu die Engine-Logs.

Motorprotokolle sind verfügbar in CloudWatch Gruppe protokollieren/aws/omics/WorkflowLog für alle Läufe. Sie sind nach Abschluss auch im Ausgabeverzeichnis des Laufs verfügbar.

Warum ist das Caching von Aufrufen für eine Aufgabe deaktiviert?

Prüfen Sie mithilfe der Workflow-Engine-Funktionen, ob die Aufgabe so konfiguriert ist, dass das Caching deaktiviert wird:

- Für WDL-Workflows: Prüfen Sie, ob für die Aufgabe die Option Volatile auf gesetzt ist `true` im Meta-Bereich
- Für Nextflow-Workflows: Prüfen Sie, ob für die Aufgabe die Cache-Direktive auf gesetzt ist `false`
- Für CWL-Workflows: Prüfen Sie, ob für die Aufgabe die EnableReuse auf gesetzt ist `false` für den WorkReuse Merkmal

Problembehandlung bei Datenspeichern

Themen

- Warum ist S3 GetObject schlägt mein Lesesatz fehl?
- Warum kann ich meinen Annotations- oder Variantenspeicher in Athena nicht sehen?
- Warum kann ich nicht auf meinen Datenspeicher in Athena zugreifen?

Warum ist S3 GetObject schlägt mein Lesesatz fehl?

In den meisten Fällen ist der Fehler auf eine fehlende Berechtigung zurückzuführen. Die Leseberechtigung für Sequenzspeicher S3 ist eine bidirektionale Konfiguration, bei der sowohl die S3-Zugriffsrichtlinie für den Sequenzspeicher den Zugriff ermöglicht, als auch dem IAM-Prinzipal eine Richtlinie zugeordnet sein muss, die den Zugriff ermöglicht. Weitere Informationen zu den Richtlinienanforderungen finden Sie unter Berechtigungen für den Datenzugriff mit Amazon S3 URIs. Vergewissern Sie sich, dass die folgenden Konfigurationen vorhanden sind:

- Die S3-Zugriffsrichtlinie für den Sequenzspeicher hat den Zugriff auf den IAM-Prinzipal oder das Stammverzeichnis des Prinzipalkontos explizit zugelassen.
- Stellen Sie sicher, dass der IAM-Prinzipal über eine Richtlinie verfügt, die ausdrücklich die Berechtigung für den Zugriff auf die Ressource erteilt. Beachten Sie, dass die IAM-Prinzipalrichtlinie bei der Definition von Berechtigungen den Access Point-ARN und nicht den auf

dem Access Point-Alias basierenden Pfad verwenden muss und dass der ARN in dem Zustand ist und nicht zur Angabe einer Ressource verwendet wird.

- Wenn Ihr Shop einen vom Kunden verwalteten Schlüssel verwendet (CMK-KMS), stellen Sie sicher, dass der IAM-Principal über kms:decrypt-Berechtigungen für den Schlüssel verfügt. Sehen Sie sich das KMS an Leitfaden für den kontoübergreifenden Zugriff zur Konfiguration der kontoübergreifenden Nutzung.

Wenn Sie eine Richtlinie haben, die tagbasierte Zugriffskontrollen verwendet, stellen Sie Folgendes sicher:

- Stellen Sie sicher, dass der Sequenzspeicher die Synchronisierung der Tags abgeschlossen hat. Dazu muss der Status des Speichers wie folgt lauten active und nicht updating.
- Stellen Sie sicher, dass der Tag-Schlüssel oder der Schlüsselwert des Lesesatzes und der Richtlinie keine Tippfehler enthalten.

Warum kann ich meinen Annotations- oder Variantenspeicher in Athena nicht sehen?

Stelle sicher, dass du in Lake Formation einen Ressourcen-Link erstellst, der auf dem Store basiert, der mit dir geteilt wurde. Sobald Sie einen Ressourcenlink erstellt haben, für den Sie eine Zugriffsberechtigung haben, sollte der Shop in Athena sichtbar sein. Weitere Informationen finden Sie unter Konfiguration von Lake Formation zur Verwendung HealthOmics.

Warum kann ich nicht auf meinen Datenspeicher in Athena zugreifen?

Wenn Ihr Annotations- oder Variantenspeicher sichtbar ist, Sie aber eine Fehlermeldung erhalten, die besagt, dass der Zugriff verweigert wurde, überprüfen Sie, welche Version der Abfrage-Engine Sie verwenden. Nur Abfragen, die mit der Engine-Version 3 ausgeführt werden, werden unterstützt. Weitere Informationen zu den Versionen der Athena-Abfrage-Engine finden Sie in der Amazon Athena-Dokumentation.

Problembehandlung mit Kiro CLI

Kiro CLI kann Ihnen helfen, Ihren Fehlerbehebungsprozess zu optimieren, indem Sie:

- Analysieren von Workflow-Ausführungen und Debuggen von Aufgabenfehlern

- Erfassung relevanter Protokolle und Fehlermeldungen
- Erstellen AWS Support-Fälle mit allen erforderlichen Debugging-Protokollen im Anhang
- Redigiert personenbezogene Daten (PII) aus Informationen, die an übermittelt wurden AWS Unterstützung

Für weitere Informationen zur Verwendung von Kiro CLI mit AWS HealthOmics Informationen zur Problembehandlung und Erstellung von Supportfällen finden Sie in der HealthOmics Tutorial zur generativen KI von Agentic auf GitHub.

> **Warning**
>
> Wenn du mit Kiro CLI arbeitest, überprüfe alle generierten Inhalte und vorgeschlagenen Aktionen, bevor du fortfährst. Geben Sie Feedback, um die Antwortqualität zu verbessern und die Anforderungen Ihres Workflows zu erfüllen. Weitere Informationen finden Sie unter Überlegungen zur Sicherheit und bewährte Methoden für Kiro.

Kontingente für AWS HealthOmics

AWS füllt Ihr Konto mit Standardwerten für die HealthOmics Kontingente. Sofern nicht anders angegeben, ist jeder Kontingentwert der Höchstwert pro Region.

> **Important**
>
> Sie können eine Erhöhung der meisten Service- und API-Kontingente beantragen. Weitere Informationen finden Sie in den folgenden Themen.

Themen

- HealthOmics Servicekontingente
- HealthOmics Kontingente mit fester Größe
- HealthOmics API-Kontingente

HealthOmics Servicekontingente

In der folgenden Tabelle sind die HealthOmics Servicekontingente zusammen mit ihren Standardwerten aufgeführt. Um die aktuellen Kontingente für jede Region anzuzeigen, öffnen Sie die Service-Kontingente-Konsole.

> **Important**
>
> Mithilfe der Service Quota-Konsole können Sie eine Erhöhung eines anpassbaren Kontingents beantragen.

Weitere Informationen zu Servicekontingenten finden Sie unter Anfordern einer Kontingenterhöhung im Service Quota-Benutzerhandbuch. Verwenden Sie für ein Kontingent, das in der Service Quota-Konsole nicht verfügbar ist, das Formular zur Kontingenterhöhung.

Name	Standard	Anpas	Description
Analytics — Maximale Anzahl an Speichern von Anmerkungen	Jede unterstützte Region: 10	Yes (Ja)	Die maximale Anzahl von Annotationsspeiche

Name	Standard	Anpas	Description
			rn in der aktuellen AWS Region
Analytics — Maximale Anzahl gleichzeitiger Importaufträge für Varianten- oder Annotationsspeicher	Jede unterstützte Region: 5	Yes (Ja)	Die maximale Anzahl gleichzeitiger Importaufträge in der aktuellen Region AWS
Analytics — Maximale Anzahl von Dateien pro Importauftrag für einen Variantenspeicher	Jede unterstützte Region: 1 000	Ja	Die maximale Anzahl von Dateien pro Variantenimportauftrag in der aktuellen AWS Region
Analytics — Maximale Anzahl von Annotationen pro Annotationsspeicher	Jede unterstützte Region: 10	Yes (Ja)	Die maximale Anzahl von Shares pro Annotationsspeicher in der aktuellen AWS Region
Analytics — Maximale Anzahl der Anteile pro Variantenspeicher	Jede unterstützte Region: 10	Yes (Ja)	Die maximale Anzahl von Aktien pro Variantenshop in der aktuellen AWS Region
Analytics — Maximale Größe jeder Datei in einem Variantenimport-Job	Jede unterstützte Region: 20 Gigabyte	Ja	Die maximale Größe einer Datei in einem Variantenimportauftrag in der aktuellen AWS Region
Analytics — Maximale Größe jeder Datei in einem Annotationsimportauftrag	Jede unterstützte Region: 20 Gigabyte	Ja	Die maximale Größe einer Datei in einem Job zum Import von Anmerkungen in der aktuellen AWS Region

Name	Standard	Anpas	Description
Analytics — Maximale Anzahl an Variantenspeichern	Jede unterstützte Region: 10	Yes (Ja)	Die maximale Anzahl von Variantenshops in der aktuellen AWS Region
Analytics — Maximale Versionen pro Annotationsspeicher	Jede unterstützte Region: 10	Yes (Ja)	Die maximale Anzahl von Versionen pro Annotationsspeicher in der aktuellen AWS Region
Konfigurationen — Maximale Konfigurationen	Jede unterstützte Region: 10	Yes (Ja)	Die maximale Anzahl von Konfigurationen in der aktuellen AWS Region.
Speicher — Maximale Anzahl gleichzeitiger Lesesatz-Aktivierungsaufträge	Jede unterstützte Region: 25	Ja	Die maximale Anzahl gleichzeitiger Leseset-Aktivierungsaufträge in der aktuellen Region AWS
Speicher — Maximale Anzahl gleichzeitiger Sequenz- und Referenzspeicher-Exportaufträge	Jede unterstützte Region: 5	Yes (Ja)	Die maximale Anzahl gleichzeitiger Exportaufträge aus einem Sequenz- oder Referenzspeicher in der aktuellen Region AWS
Speicher — Maximale Anzahl gleichzeitiger Sequenz- oder Referenzspeicher-Importaufträge	Jede unterstützte Region: 5	Yes (Ja)	Die maximale Anzahl gleichzeitiger Importaufträge für einen Sequenz- oder Referenzspeicher in der aktuellen Region AWS

Name	Standard	Anpas	Description
Speicher — Maximale Lesesätze pro Sequenzspeicher	Jede unterstützte Region: 1 000 000	Ja	Die maximale Anzahl von Lesesätzen in einem Sequenzspeicher in der aktuellen AWS Region
Speicher — Maximale Anzahl von Referenzen pro Referenzspeicher	Jede unterstützte Region: 50	Ja	Die maximale Anzahl von Referenzen in einem Referenzspeicher in der aktuellen AWS Region
Speicher — Maximale Anzahl an Sequenzspeichern	Jede unterstützte Region: 20	Ja	Die maximale Anzahl von Sequenzspeichern in der aktuellen AWS Region
Workflows — Maximale Anzahl aktiver GPUs	Jede unterstützte Region: 12	Ja	Die maximale Anzahl gleichzeitig aktiver GPUs in der aktuellen Region. AWS In us-east-1 und us-west-2 werden Anfragen zur Erhöhung des Kontingents für Werte bis zu 500 automatisch genehmigt.
Workflows — Maximale Anzahl gleichzeitiger aktiver Läufe unter Verwendung des dynamischen Laufspeichers	Jede unterstützte Region: 50	Ja	Die maximale Anzahl aktiver Läufe mit dynamischem Laufspeicher in der aktuellen AWS Region. Anfragen zur Erhöhung des Kontingents für Werte bis zu 200 werden automatisch genehmigt.

Name	Standard	Anpas	Description
Workflows — Maximale Anzahl gleichzeitiger aktiver Läufe unter Verwendung von statischem Laufspeicher	Jede unterstützte Region: 10	Yes (Ja)	Die maximale Anzahl aktiver Läufe mit statischem Laufspeicher in der aktuellen AWS Region. Anfragen zur Erhöhung des Kontingents für Werte bis zu 50 werden automatisch genehmigt.
Workflows — Maximale Anzahl gleichzeitiger Aufgaben pro Lauf	Jede unterstützte Region: 25	Ja	Die maximale Anzahl gleichzeitiger Aufgaben in jeder Ausführung in der aktuellen AWS Region. In us-east-1 und us-west-2 werden Anfragen zur Erhöhung des Kontingents für Werte bis zu 100 automatisch genehmigt.
Workflows — Maximale Ausführungsdauer	Jede unterstützte Region: 604.800 Sekunden	Ja	Die maximale Workflow-Ausführungsdauer in der aktuellen AWS Region.
Workflows — Maximale Anzahl an Durchläufen (aktiv oder inaktiv)	Jede unterstützte Region: 100 000	Ja	Die maximale Anzahl von Läufen (aktiv oder inaktiv) in der aktuellen AWS Region.
Workflows — Maximale Anzahl an Freigaben pro Workflow	Jede unterstützte Region: 100	Yes (Ja)	Die maximale Anzahl von Shares pro Workflow in der aktuellen AWS Region

Name	Standard	Anpas	Description
Workflows — Maximale statische Laufspeicherkapazität pro Lauf	Jede unterstützte Region: 9.600	Ja	Die maximale statische Laufspeicherkapazität in Gibibyte (GiB) für jeden Lauf in der aktuellen Region. AWS In us-east-1 und us-west-2 werden Anfragen zur Erhöhung des Kontingents für Werte bis zu 50.000 automatis ch genehmigt.
Workflows — Maximale Anzahl an Arbeitsabläufen	Jede unterstützte Region: 1 000	Ja	Die maximale Anzahl von Workflows in der aktuellen AWS Region.
Workflows — Transaktionen pro Sekunde (TPS) für den Vorgang StartRun	Jede unterstützte Region: 5	Yes (Ja)	Die maximale Anzahl an Transaktionen pro Sekunde (TPS) für den StartRun Vorgang in der aktuellen AWS Region. Anfragen zur Erhöhung des Kontingents für Werte bis zu 10 werden automatisch genehmigt.

HealthOmics Kontingente mit fester Größe

HealthOmics Enthält zusätzlich zu den HealthOmics Servicekontingente Kontingente mit fester Größe. Für diese Werte können Sie keine Erhöhung beantragen.

Sofern nicht anders angegeben, gibt jedes Kontingent den Höchstwert pro Region an.

Themen

- HealthOmics feste Größenkontingente für Analytics
- HealthOmics Speicherplatz: feste Größenkontingente

- HealthOmics feste Größenkontingente für Workflows
- HealthOmics Kontingente für mehrere
- HealthOmics Feste Größenkontingente für den Ready2Run-Workflow

HealthOmics feste Größenkontingente für Analytics

Die folgende Tabelle zeigt die unterstützten Höchstwerte für Analytics-Kontingente. Diese Werte sind nicht anpassbar.

Name	Description	Maximum	Einstellbar Yes/No
Analytik — Maximale Anzahl an Dateien pro Importauftrag im Annotationsspeicher	Die maximale Anzahl von Dateien pro Auftrag zum Import von Anmerkungen.	1	Nein

HealthOmics Speicherplatz: feste Größenkontingente

Die folgende Tabelle zeigt die maximal unterstützten Werte für Speicherdateien. Diese Werte sind nicht einstellbar.

Name	Description	Maximum	Einstellbar Yes/No
Speicher — Maximale Größe der S3-Zugrif fsressourcenrichtlinie	Die maximale Größe der S3-Zugriffsressour cenrichtlinie	15 KB	Nein
Speicher — Maximale Anzahl übertragener Set-Level-Tags	Die maximale Anzahl von Set-Level-Tag-Schlüsseln pro Store, die an das S3-Objekt weitergegeben werden	5	Nein

Name	Description	Maximum	Einstellbar Yes/No
Speicher — Maximale Lesesätze pro Aktivierungsauftrag	Die maximale Anzahl von Lesesätzen pro Aktivierungsjob.	20	Nein
Speicher — Maximale Lesesätze pro Exportauftrag	Die maximale Anzahl von Lesesätzen pro Exportauftrag.	100	Nein
Speicher — Maximale Lesesätze pro Importauftrag	Die maximale Anzahl von Lesesätzen pro Importauftrag.	100	Nein
Speicher — Maximale Anzahl von Referenzspeichern	Die maximale Anzahl von Referenzspeichern.	1	Nein
Speicher — Maximale Teilgröße für einen direkten Upload	Die maximale Bauteilgröße für den direkten Upload in einen Sequenzspeicher.	100 MB	Nein
Speicherplatz — Maximale Anzahl von Teilen in einer Datei für den direkten Upload	Die maximale Anzahl von Teilen in einer Datei für den direkten Upload in einen Sequenzspeicher.	10.000	Nein
Speicher — Maximale Referenzgröße	Die maximale Größe einer Referenzdatei, die in einen Referenzspeicher importiert werden kann.	15 GB	Nein

Name	Description	Maximum	Einstellbar Yes/No
Speicher — Maximale Quellgröße des Lesesatzes	Die maximale Größe einer einzelnen Quelldatei in einem Lesesatz, die in einen Sequenzspeicher importiert werden kann.	976 GB	Nein

HealthOmics feste Größenkontingente für Workflows

Die folgende Tabelle zeigt die unterstützten Höchstwerte für Workflow-Kontingente. Diese Werte sind nicht anpassbar.

Name	Description	Maximale Größe	Einstellbar Yes/No
Workflows — Maximale Anzahl an Ausführungsgruppen	Die maximale Anzahl von Ausführun gsgruppen.	1000	Nein
Workflows — Maximale Anzahl an ausgeführten Caches	Die maximale Anzahl von Run-Caches, die Sie für ein Konto erstellen können. Ein oder mehrere Läufe können denselben Run-Cache gemeinsam nutzen. Es gibt kein Kontingen t für die Anzahl der Läufe, die pro Konto zwischengespeichert werden HealthOmics können.	1000	Nein

Name	Description	Maximale Größe	Einstellbar Yes/No
Workflows — Maximale Anzahl Workflow-Versionen	Die maximale Anzahl von Workflow-Versionen pro Workflow.	1000	Nein
Workflows — Größe des CPU-Instanz-Containers	Die maximale Container-Image-Größe für eine CPU-Instanz.	45 GiB	Nein
Workflows — Größe des Containers der GPU-Instanz	Die maximale Container-Image-Größe für eine GPU-Instanz.	95 GiB	Nein
GPU-Instanz/dev/shm gemeinsam genutzter Speicher	Die maximale Menge an gemeinsam genutztem Speicher pro GPU-Instanz.	8 GB pro GPU	Nein
Workflows — Parameterdatei ausführen	Die maximale Größe einer Ausführungsparameterdatei.	256.000 Byte	Nein
Workflows — Vorlagendatei für Workflow-Parameter	Die maximale Anzahl von Einträgen und die maximale Dateigröße für eine Workflow-Parameter-Vorlagendatei. Dieses Kontingent gilt für Workflows, die Sie mit der Konsole oder API erstellen.	1.000 Einträge, 400 KB	Nein

Name	Description	Maximale Größe	Einstellbar Yes/No
Workflows — Dateigröße der Workflow-Definition — API (DefinitionZip-Direktupload)	Die maximale Größe der Workflow-Definitionsdatei, wenn Sie den Workflow mithilfe der API-Operation oder eines AWS SDK mit DefinitionZip (direkter Upload) erstellen.	4,4 MB	Nein
Workflows — Dateigröße der Workflow-Definition — API (DefinitionURI S3-Upload)	Die maximale Größe der Workflow-Definitionsdatei, wenn Sie den Workflow mithilfe der API-Operation oder eines AWS SDK mit DefinitionURI (S3-Upload) erstellen.	100 MB	Nein
Workflows — Größe der Workflow-Definitionsdatei — Konsole (direkter Upload)	Die maximale Größe der Workflow-Definitionsdatei, die Sie als direkten Upload angeben können, wenn Sie den Workflow mithilfe der Konsole erstellen.	4,4 MB	Nein
Workflows — Dateigröße der Workflow-Definition — Konsole (Upload von Amazon S3)	Die maximale Größe der Workflow-Definitionsdatei, die Sie als Upload von Amazon S3 bereitstellen können, wenn Sie den Workflow mithilfe der Konsole erstellen.	100 MB	Nein

Name	Description	Maximale Größe	Einstellbar Yes/No
Workflows — Größe des Repositorys	Die maximale Größe eines externen Code-Repositorys.	1 GiB	Nein
Workflows — Individuelle Dateigröße des Repositorys	Die maximale Größe einer einzelnen Datei aus einem externen Code-Repository.	100 MiB	Nein
Workflows — Größe der README-Datei	Die maximale Größe einer README-Datei.	500 KiB	Nein
Workflows — Maximale Konfigurationen pro Konto	Die maximale Anzahl von VPC-Konfigurationen pro Konto.	10	Nein
Workflows — Maximale Sicherheitsgruppen pro Konfiguration	Die maximale Anzahl von Sicherheitsgruppen pro VPC-Konfiguration.	5	Nein
Workflows — Maximale Anzahl Subnetze pro Konfiguration	Die maximale Anzahl von Subnetzen pro VPC-Konfiguration.	16	Nein
Workflows — Maximale Anzahl Subnetze pro Availability Zone	Die maximale Anzahl von Subnetzen pro Availability Zone in einer VPC-Konfiguration.	1	Nein
Workflows — Elastische Netzwerkschnittstellen pro VPC-Konfiguration	Die maximale Anzahl an Elastic Network Interfaces (ENIs) pro VPC-Konfiguration.	5,000	Nein

Vorschläge, wie Sie die Größe Ihrer Ausführungsparameterdatei reduzieren können, finden Sie unter. Verwaltung der Größe von Ausführungsparametern

HealthOmics Kontingente für mehrere

Die folgenden Kontingente gelten für Batch-Operationen. Diese Werte sind nicht anpassbar.

Kontingent	Wert
Maximale Anzahl an Durchläufen pro Stapel	100 000
Maximale Inline-Run-Konfigurationen pro Anfrage	100
Maximale Größe der S3-Konfigurationsdatei	6 GB
StartRunBatch API-Durchsatz	1 TPS
GetBatch API-Durchsatz	10 TPS
CancelRunBatch API-Durchsatz	10 TPS
ListRunsInBatch API-Durchsatz	10 TPS

HealthOmics Feste Größenkontingente für den Ready2Run-Workflow

Jeder Ready2Run-Workflow hat eine maximale Eingabedateigröße. In der folgenden Tabelle sind die Dateigrößeneinheiten in Gibibytes (GiB) aufgeführt. Diese maximalen Dateigrößen sind nicht anpassbar.

Name des Ready2Run-Workflows	Maximale Größe der Eingabedatei (GiB)	Einstellbar (Yes/No)
AlphaFold für 601-1200 Rückstände	1	Nein
AlphaFold für bis zu 600 Rückstände	1	Nein

Name des Ready2Run-Workflows	Maximale Größe der Eingabedatei (GiB)	Einstellbar (Yes/No)
Bases2Fastq für 2x150	1000	Nein
Bases2Fastq für 2x300	1000	Nein
Bases2Fastq für 2x75	500	Nein
SMFold für bis zu 800 Rückstände	1	Nein
GATK-BP fq2bam	64	Nein
GATK-BP Keimbahn bam2vcf für das 30x-Genom	39	Nein
GATK-BP Germline fq2vcf für 30x Genom	64	Nein
GATK-BP Somatisches WES bam2vcf	86	Nein
NVIDIA Parabricks BAM2FQ2BAM WGS für bis zu 30X	80	Nein
NVIDIA Parabricks BAM2FQ2BAM WGS für bis zu 50X	120	Nein
NVIDIA Parabricks BAM2FQ2BAM WGS für bis zu 5X	20	Nein
NVIDIA Parabricks FQ2BAM WGS für bis zu 30X	71	Nein
NVIDIA Parabricks FQ2BAM WGS für bis zu 50X	137	Nein

Name des Ready2Run-Workflows	Maximale Größe der Eingabedatei (GiB)	Einstellbar (Yes/No)
NVIDIA Parabricks FQ2BAM WGS für bis zu 5X	13	Nein
NVIDIA Parabricks Germline WGS für bis zu 30X DeepVariant	71	Nein
NVIDIA Parabricks Germline WGS für bis zu 50X DeepVariant	137	Nein
NVIDIA Parabricks Germline WGS für bis zu 5X DeepVaria nt	12	Nein
NVIDIA Parabricks Germline WGS für bis zu 30X Haplotype Caller	71	Nein
NVIDIA Parabricks Germline WGS für bis zu 50X Haplotype Caller	137	Nein
NVIDIA Parabricks Germline WGS für bis zu 5X Haplotype Caller	13	Nein
NVIDIA Parabricks Somatic Mutect2 WGS für bis zu 50X	196	Nein
SCRNAseq mit den Tools von Kallistobus	119	Nein
scRNAseq mit Lachs Alevin-fr y	119	Nein

Name des Ready2Run-Workflows	Maximale Größe der Eingabedatei (GiB)	Einstellbar (Yes/No)
scRNAseq mit StarSolo	119	Nein
Sention Germline BAM WES für bis zu 300x	9	Nein
Sentieon Germline BAM WGS für bis zu 32x	18	Nein
Sentiron Germline FASTQ WES für bis zu 100x	5	Nein
Sentiron Germline FASTQ WES für bis zu 300x	26	Nein
Sentiron Germline FASTQ WGS für bis zu 32x	51	Nein
Sentieon LongRead für ONT	25	Nein
Sentieon für LongRead PacBio HiFi	58	Nein
Sention Somatic WES	50	Nein
Sentieon Somatic WGS	113	Nein
Ultima Genomics für bis zu 40x DeepVariant	91	Nein

HealthOmics API-Kontingente

HealthOmics hat die folgenden Kontingente im Zusammenhang mit API-Vorgängen. Wo angegeben, ist das Kontingent anpassbar. Um eine Erhöhung zu beantragen, verwenden Sie das Formular zur Erhöhung der Quote.

Für jeden aufgeführten API-Vorgang entspricht das Kontingent der maximalen Anzahl von Transaktionen pro Sekunde (TPS) für diesen API-Vorgang in jeder Region.

Themen

- Allgemeine API-Kontingente
- Speicher-API-Kontingente
- Workflow-API-Kontingente
- Kontingente für Analytics-APIs

Allgemeine API-Kontingente

In der folgenden Tabelle sind allgemeine API-Operationen aufgeführt, die für mehr als eine Kategorie gelten (Speicher, Workflows und Analysen).

API-Operation	Standardmäßiger maximaler TPS	Einstellbar () Yes/No
AcceptShare, CreateShare, DeleteShare, GetShare, ListShares	1 TPS	Ja

Speicher-API-Kontingente

In der folgenden Tabelle sind die Speicher-API-Operationen aufgeführt.

Betrieb der Speicher-API	Standardmäßiger maximaler TPS	Einstellbar () Yes/No
CreateSequenceStore, UpdateSequenceStore, DeleteSequenceStore, CreateReferenceStore, DeleteReferenceStore	1 TPS	Ja
BatchDeleteReadSet, DeleteReference	1 TPS	Ja

Betrieb der Speicher-API	Standardmäßiger maximaler TPS	Einstellbar () Yes/No
CreateMultipartReadSetUpload, CompleteMultipartReadSetUpload, AbortMultipartReadSetUpload	1 TPS	Nein
GetS3AccessPolicy, PutS3AccessPolicy, DeleteS3AccessPolicy	1 TPS	Ja
GetReference	10 TPS	Ja
UploadReadSetPart	10 TPS	Ja
GetReadSet	30 TPS	Ja
GetSequenceStore, ListSequenceStores	5 TPS	Ja
GetReadSetMetadata, ListReadSets	5 TPS	Ja
StartReadSetImportJob, GetReadSetImportJob, ListReadSetImportJobs	5 TPS	Ja
StartReadSetExportJob, GetReadSetExportJob, ListReadSetExportJobs	5 TPS	Ja
ListReferenceStores	5 TPS	Ja
StartReferencetImportJob, GetReferenceImportJob, ListReferenceImportJobs	5 TPS	Ja

Betrieb der Speicher-API	Standardmäßiger maximaler TPS	Einstellbar () Yes/No
ListReferences, GetReferenceMetadata	5 TPS	Ja
StartReadsetActivationJob	5 TPS	Ja
ListReadsetActivationJobs, GetReadSetActivationJob	5 TPS	Ja
ListMultipartReadSetUploads, ListReadSetUploadParts	5 TPS	Ja
TagResource, UntagResource, ListTagsForResource	5 TPS	Ja

Workflow-API-Kontingente

In der folgenden Tabelle sind die Workflow-API-Operationen aufgeführt.

Betrieb der Workflow-API	Standardmäßiger maximaler TPS	Einstellbar () Yes/No
StartRun	1 TPS	Ja
CreateWorkflow	5 TPS	Ja
CancelRun, DeleteRun, GetRun, GetRunTask, ListRunTasks, ListRuns	10 TPS	Ja
CreateRunGroup, DeleteRunGroup, GetRunGroup, ListRunGroups, UpdateRunGroup	10 TPS	Ja

Betrieb der Workflow-API	Standardmäßiger maximaler TPS	Einstellbar () Yes/No
CreateRunCache, UpdateRunCache, DeleteRunCache, GetRunCache, ListRunCaches	10 TPS	Ja
DeleteWorkflow, GetWorkflow, ListWorkflows, UpdateWorkflow	10 TPS	Ja

Kontingente für Analytics-APIs

In der folgenden Tabelle sind die Analytics-API-Operationen aufgeführt.

Betrieb der Analytics-API	Standardmäßiger maximaler TPS	Einstellbar () Yes/No
CreateVariantStore, DeleteVariantStore, GetVariantStore, ListVariantStores, UpdateVariantStore	1 TPS	Nein
StartVariantImportJob, CancelVariantImportJob, GetVariantImportJob, ListVariantImportJobs	1 TPS	Nein
CreateAnnotationStore, DeleteAnnotationStore, GetAnnotationStore, ListAnnotationStores, UpdateAnnotationStore	1 TPS	Nein
StartAnnotationImportJob, ListAnnotationImportJobs,	1 TPS	Nein

Betrieb der Analytics-API	Standardmäßiger maximaler TPS	Einstellbar () Yes/No
GetAnnotationImportJob, CancelAnnotationImportJob		

Dokumenthistorie für den HealthOmics Benutzerleitfaden

In der folgenden Tabelle werden die Dokumentationsversionen für beschrieben HealthOmics.

Änderung	Beschreibung	Datum
Neue Inhalte	AWS HealthOmics Es wurde eine Anleitung zum Verpacken und Bereitstellen einer Workflow-Definition hinzugefügt: Wie wählt man zwischen einem ZIP-Archiv, einem Amazon S3-Speicherort und Git-based Repository; wie das ZIP-Archiv angelegt wird; wann muss das eingerichtet werden `main` Parameter; und wie der Standard-Laufspeichertyp und die Kapazität zusammenpassen. Weitere Informationen finden Sie unter Verpacken und liefern Sie eine HealthOmics Workflow-Definition.	16. Oktober 2026
Neue Funktion	Mit der erweiterten Unterstützung für Container-Registry können Sie Container-Images aus Registries wie Seqera Containers, NVIDIA NGC, und Google Artifact Registry mit AWS HealthOmics private Workflows, ohne Bilder manuell in Amazon ECR zu migrieren oder Ihre Workflow-Definitionen zu bearbeiten.	14. Oktober 2026

	Weitere Informationen finden Sie unter Die Verwendung von Registrierungen von Drittanbietern wird vom Pull-Through-Cache nicht unterstützt.	
Neue Funktion	HealthOmics fügte das hinzu `omicsResourceFallbackOrder` Direktive für WDL- und Nextflow-Workflows. Verwenden Sie sie, um Ressourcenprofile in der Reihenfolge ihrer Präferenz aufzulisten, sodass eine Aufgabe auf einen anderen Beschleunigertyp oder auf die CPU zurückfällt, wenn die bevorzugte Kapazität nicht verfügbar ist. Weitere Informationen finden Sie unter Erweiterte Ressourcenkonfiguration.	13. Oktober 2026
Neue Inhalte	HealthOmics eine Anleitung zum Laufen hinzugefügt Illumina DRAGEN Softwaremodus in privaten Workflows. Weitere Informationen finden Sie unter Illumina DRAGEN auf HealthOmics private Arbeitsabläufe.	9. Oktober 2026

Neue Funktion	HealthOmics hat ein Änderungsprotokoll für die WDL-Engine veröffentlicht, das neue Funktionen, Verbesserungen und Verhaltensänderungen für die Standard-WDL-Engine und den WDL-Milentmodus behandelt. Weitere Informationen finden Sie unter WDL-Engine-Änderungsprotokoll für HealthOmics.	22. September 2026
Neue Funktion	HealthOmics Unterstützung für Nextflow-Profile über die API hinzugefügt, sodass Sie zur Laufzeit eine umgebungsspezifische Konfiguration auswählen können, indem Sie `engineSettings` Parameter. Weitere Informationen finden Sie unter Verwenden Sie Nextflow-Profile.	1. Juni 2026
Neue Funktion	HealthOmics Unterstützung für Batch-Läufe hinzugefügt, mit denen Sie bis zu 100.000 Workflow-Läufe in einer einzigen API-Anfrage einreichen können. Weitere Informationen finden Sie unter Batch läuft in HealthOmics.	20. März 2026

AWS HealthOmics Variant Stores und Annotation Stores stehen Neukunden nicht mehr zur Verfügung.	AWS HealthOmics Variant Stores und Annotation Stores stehen Neukunden nicht mehr zur Verfügung. Weitere Informationen finden Sie unter AWS HealthOmics Die Verfügbarkeit von Varianten - und Annotationsspeichern ändert sich.	7. November 2025
AWS HealthOmics Variant Stores und Annotation Stores stehen ab dem 7. November 2025 nicht mehr für Neukunden zur Verfügung.	AWS HealthOmics Variant Stores und Annotation Stores stehen ab dem 7. November 2025 nicht mehr für Neukunden zur Verfügung. Wenn Sie Variant Stores oder Annotation Stores verwenden möchten, melden Sie sich vor diesem Datum an. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter AWS HealthOmics Die Verfügbarkeit von Varianten- und Annotationsspeichern ändert sich.	7. Oktober 2025
Neue Funktionen	HealthOmics Unterstützung für Workflows zur Synchronisation eines privaten Amazon ECR-Repositorys mit einer Upstream-Registrierung hinzugefügt. Weitere Informationen finden Sie unter Container-Images für private Workflows in HealthOmics.	28. August 2025

Neue Funktionen zur README- und Repository-Integration	Unterstützung für die Erstellung von Workflows aus hinzugefügt externe Code-Repositorys und README-Dateien.	24. Juli 2025
Neue Funktionen	HealthOmics Unterstützung für die automatische Parameterinterpolation von Nextflow hinzugefügt. Weitere Informationen finden Sie unter Parameter-Vorlagendateien für HealthOmics Workflows.	27. Juni 2025
Neue Funktionen	HealthOmics Unterstützung für Workflows hinzugefügt, um die Ausführungsparameter aus einer WDL-Workflow-Definitionsdatei zu interpolieren. Weitere Informationen finden Sie unter Parameter-Vorlagendateien für HealthOmics Workflows.	30. Mai 2025
Neue Funktionen	HealthOmics Unterstützung für Workflow-Versionierung hinzugefügt. Weitere Informationen finden Sie unter Workflow-Versionierung in HealthOmics.	18. April 2025
Neue Funktionen	HealthOmics elastischer Durchsatz für dynamischen Laufspeicher hinzugefügt. Weitere Informationen finden Sie unter Führen Sie Speichertypen in aus HealthOmics.	16. April 2025

Neue Funktionen	HealthOmics attributb asierte Zugriffskontrollen für Sequence Store S3-Standorte hinzugefügt und die Möglichke it, bis zu fünf Readset-Tags mit einem Sequence Store S3-Objekt zu synchronisieren. Weitere Informationen finden Sie unter Erstellen einer HealthOmics Sequenzsp eicher.	22. November 2024
Neue Funktionen	HealthOmics Unterstützung für das Zwischenspeichern von Anrufen, auch bekannt als Resume, für private Workflows hinzugefügt. Weitere Informati onen finden Sie unter Caching aufrufen.	20. November 2024
Neue Funktionen	HealthOmics Es wurden neue API-Felder hinzugefügt, die Ihnen bei der Zuordnung zwischen Sequenzspeicher-Eingabejobs und Lesesätzen helfen.	29. August 2024
Neue Funktionen	HealthOmics Unterstützung für die Verwaltung von Nextflow-Versionen hinzugefügt. Weitere Informationen finden Sie unter Nextflow-Versionen.	14. August 2024
Neue Funktionen	HealthOmics Unterstützung für gemeinsame Workflows und dynamischen Run-Speicher hinzugefügt.	30. April 2024

Neue Funktionen	HealthOmics Unterstützung für Amazon S3-Zugriff auf Referenz- und Sequenzsp eicher sowie Unterstützung für SHA256-ETags hinzugefügt.	15. April 2024
Neue Funktionen	HealthOmics Entitäts-Tags (ETags) für Sequenzspeicher hinzugefügt.	06. Oktober 2023
Neue Funktionen	HealthOmics Versionierung von Annotationsspeichern und gemeinsame Nutzung von Analysespeichern hinzugefügt.	15. August 2023
Neue Funktionen	HealthOmics Common Workflow Language (CWL) als unterstützte Sprache hinzugefügt für HealthOmics Arbeitsabläufe.	30. Juni 2023
Neue Funktionen	HealthOmics neue Ready2Run-Workflows hinzugefügt, GPU-Unter stützung für Workflows, Datenanalyse für Annotatio nsspeicher, direktes Hochladen in HealthOmics Speicherung und Integration mit EventBridge.	15. Mai 2023
Neue verwaltete Richtlinie	HealthOmics eine neue verwaltete Richtlinie hinzugefü gt, die vollen Zugriff bietet. Weitere Informationen finden Sie unter Von AWS verwaltete Richtlinien.	23. Februar 2023

Neue verwaltete Richtlinie	HealthOmics eine neue verwaltete Richtlinie hinzugefügt, die den Zugriff auf schreibgeschützt beschränkt. Weitere Informationen finden Sie unter Von AWS verwaltete Richtlinien.	29. November 2022
Erstversion	Erste Veröffentlichung des HealthOmics Benutzerleitfaden	29. November 2022

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.