aws

用户指南

AWS HealthOmics

版本 latest

AWS HealthOmics: 用户指南

Table of Contents

什么是 AWS HealthOmics？

AWS HealthOmics 是一项 AWS 服务，可帮助生物信息学家、研究人员和科学家等用户存储、查询、分析基因组学和其他生物学数据并从中生成见解。它简化并加快了研究和临床组织存储和分析基因组信息的过程，并加快了科学发现和见解生成速度的速度。

HealthOmics 有三个主要组成部分。 HealthOmics 存储可帮助您高效地存储和共享 PB 级基因组数据，且每 GB 数据库的成本较低。 HealthOmics Analytics 简化了为多组学和多模态分析准备基因组学数据的方式。 HealthOmics 工作流程会自动为您的生物信息学计算配置和扩展底层基础架构。

主题

重要提示

HealthOmics 不能替代专业的医疗建议、诊断或治疗，也不能用于治愈、治疗、缓解、预防或诊断任何疾病或健康状况。您有责任将人工审查作为任何使用的一部分 AWS HealthOmics，包括与任何旨在为临床决策提供依据的第三方产品相关使用。

HealthOmics 仅用于传输、存储、格式化或显示数据，以及为管理工作流程提供基础架构和配置支持。 AWS HealthOmics 不打算直接进行变异调用或基因组分析和解释。 AWS HealthOmics 不用于解释或分析临床实验室测试或其他设备数据、结果和发现，也不能替代用于基因组分析的第三方工具。

HealthOmics 概念

本主题涵盖了关键概念和特定术语的定义 HealthOmics，以帮助您理解本指南中 HealthOmics 使用的术语。

主题

- Analytics
- 工作流

存储

数据存储分为序列存储，用于存放您的基因组序列和相关信息，以及用于所有参考基因组的参考存储。以下术语描述了特定于的实现 HealthOmics。

- 序列存储 — 用于存储基因组学文件的数据存储。里面可以有一个或多个序列存储 HealthOmics。可以在序列存储上设置访问权限和 AWS KMS 加密，以控制谁有权访问数据。
- 读取集 — 读取集是基因组学读取的抽象，这些读取以 FASTQ、BAM 或 CRAM 格式存储。读取集可以导入到序列存储中，并使用元数据进行注释。您可以使用基于属性的访问控制 (ABAC) 将权限应用于读取集。
- 参考 — 基因组引用与读取一起使用，用于识别特定读取或一组读取映射到基因组中的哪个位置。它们采用 FASTA 格式并存储在参考资料库中。
- 参考存储 — 用于存储参考基因组的数据存储。您可以在每个账户和地区拥有一个参考资料库。

Analytics

您可以使用 Analytics 转换和分析您的基因组数据。 HealthOmics 创建变体存储库或注释存储库，为您的查询添加其他信息。

- 变体存储 — 按人口规模存储变体数据的数据存储。变体存储支持基因组变异调用格式 (gvCF) 和 VCF 输入。
- 注解存储 — 表示注释数据库的数据存储，例如来自 TSV/CSV、VCF 或通用要素格式 () GFF3 文件的注释数据库。导入期间，注释存储映射到与变体存储相同的坐标系。

工作流

借助 HealthOmics 工作流程，您可以处理和分析您的基因组学数据。

- 工作流程 — 端到端流程的总体定义，包括参数和对工具的引用。工作流定义可以表示为 WDL、Nextflow 或 CWL。每个创建的工作流程都有一个唯一的标识符。
- 运行-对工作流程的单次调用。单个运行使用您定义的输入数据并生成输出。每个创建的运行都有一个唯一的标识符。

- 任务-运行中的各个进程。 HealthOmics工作流程使用这些定义的计算规范来运行您的任务。每个任务都有一个唯一的标识符。
- 运行组 — 一组运行，您可以为其设置最大 vCPU、最大持续时间或最大并发运行次数，以帮助限制每次运行使用的计算资源。您可以在运行组中为运行指定和配置优先级。例如，您可以指定在优先级较低的运行之前执行高优先级的运行，从而创建优先级队列。使用运行组是可选的，并且每个运行组都有一个唯一的标识符。

HealthOmics features

HealthOmics 提供以下功能。

- HealthOmics 存储 — 帮助您以较低的每 GB 数据库成本高效存储和共享 PB 级原始基因组学数据。
- HealthOmics 分析 — 简化为多组学和多模态分析准备基因组学数据的方式。
- HealthOmics 工作流程 — 为您的生物信息学工作流程自动配置和扩展底层基础架构。

您可以单独使用每个组件，也可以将其作为集成 end-to-end解决方案的一部分使用。

HealthOmics 为您提供以下好处。

- 安全地存储和合并基因组数据 — 与其他 AWS 服务（例如 AWS Lake Formation 和 Amazon Athena）HealthOmics 集成。您可以安全地存储基因组学数据，然后将其与病史数据进行查询或合并，以获得更好的诊断和个性化的治疗计划。
- 保护患者隐私 — HealthOmics 是否符合 HIPAA 资格。它还与 IAM 和 Amazon 集成， CloudWatch 因此您可以控制和记录数据访问权限，并跟踪数据在分析中的使用情况。
- 专为扩展而构建 — 通过简化的计费和新的协作工具，支持大量人口数据分析。
- 最大限度地提高效率-使用自动化工作流程和集成工具来简化数据处理和分析。

您可以 HealthOmics 用于以下生物医学应用：

- 种群测序 — 一次查询数千个基因组，以了解基因组变异如何映射到人群中的表型。
- 临床基因组学 — 从测序仪输出到可报告数据，构建可重复的基因组学工作流程。您还可以针对高容量通量进行优化，并设置高优先级临床样本的计算要求以缩短周转时间。
- 临床试验 — 将基因组分析整合到临床试验中，以更好地了解新候选药物的功效。通过长期节省成本和数据来源来简化和加快临床试验，以满足管理机构的法规。
- 加强研究和创新 — 利用内置的行和列访问控制，简化和控制匿名基因组数据的存储、访问和分析。

相关服务

以下服务适用于 HealthOmics。

- Amazon Elastic Container Registry — 每个私有工作流程都使用 Amazon ECR 映像（在私有 Amazon ECR 存储库中）来包含运行该工作流程所需的所有可执行文件、库和脚本。
- 亚马逊简单存储服务 — Amazon S3 为商店和工作流程数据提供文件存储。
- AWS Lake Formation — Lake Formation 管理对分析数据存储的数据访问权限。
- 亚马逊 Athena — 使用 Athena 对你的 Variant 商店进行查询。
- Amazon SageMaker AI — 使用 SageMaker AI 通过 Jupyter 笔记本运行 HealthOmics 任务。

的区域和终端节点 AWS HealthOmics

有关区域和终端节点的完整列表，请参阅AWS 一般参考。

除了默认处于活动状态的 AWS 区域外，还有一些需要激活的选择加入区域。要详细了解如何激活或停用某个区域，请参阅账户管理指南中的指定您的 AWS 账户可以使用哪些 AWS 区域。

如何访问 HealthOmics

您可以使用管理控制台、CLI SDKs 或 API 访问 AWS HealthOmics 功能。

- AWS 管理控制台-提供可用于访问的 Web 界面 HealthOmics。
- AWS Command Line Interface (AWS CLI) — 为各种 AWS 服务提供命令，包括 AWS HealthOmics Windows、macOS 和 Linux，并支持这些服务。有关安装的更多信息 AWS CLI，请参阅AWS Command Line Interface。
- AWS SDKs — AWS 提供 SDKs （软件开发套件），其中包括适用于各种编程语言和平台（包括 Java、Python、Ruby、.NET、iOS 和 Android）的库和示例代码。 SDKs 提供了一种便捷的 HealthOmics编程使用方式。有关更多信息，请参阅 AWS SDK 开发人员中心。
- AWS API — 您可以使用 API 操作以 HealthOmics 编程方式进行访问和管理。有关更多信息，请参阅 HealthOmics API 参考。

了解更多

通过以下研讨会和教程了解 HealthOmics 更多信息：

- HealthOmics 研讨会 — HealthOmics 端到端研讨会
- AWS 基因组学资源 — 与基因组学相关的公共 Amazon ECR 存储库
- Python 教程 — Jupyter 笔记本教程 GitHub，内容涵盖 HealthOmics 存储、分析和工作流程

熟悉其他 HealthOmics 工具，这些工具可 AWS 提供：

- WDL linter — WDL 的 HealthOmics lin ter
- Nextflow linter — Nextf HealthOmics l ow
- HealthOmics 亚马逊 ECR 帮助工具 — 亚马逊 ECR 帮助工具 HealthOmics
- HealthOmics tools on GitHub — 用于使用的工具 HealthOmics（传输管理器、URI 解析器、Omics 重新运行、运行分析器）。

设置 HealthOmics

要进行设置 AWS HealthOmics，请注册 AWS 账户，创建管理用户，然后安全地管理其他用户的访问权限。

主题

注册获取 AWS 账户

如果您没有 AWS 账户，请完成以下步骤来创建一个。

报名参加 AWS 账户

1. 打开https://portal.aws.amazon.com/billing/注册。
2. 按照屏幕上的说明操作。

   在注册时，将接到电话或收到短信，要求使用电话键盘输入一个验证码。

   当您注册时 AWS 账户，就会创建AWS 账户根用户一个。根用户有权访问该账户中的所有 AWS 服务 和资源。作为最佳安全实践，请为用户分配管理访问权限，并且只使用根用户来执行需要根用户访问权限的任务。

AWS 注册过程完成后会向您发送一封确认电子邮件。您可以随时前往 https://aws.amazon.com/并选择“我的账户”，查看您当前的账户活动并管理您的账户。

创建具有管理访问权限的用户

注册后，请保护您的安全 AWS 账户 AWS 账户根用户 AWS IAM Identity Center，启用并创建管理用户，这样您就不会使用 root 用户执行日常任务。

保护你的 AWS 账户根用户

1. 选择 Root 用户并输入您的 AWS 账户 电子邮件地址，以账户所有者的身份登录。AWS Management Console在下一页上，输入您的密码。

   要获取使用根用户登录方面的帮助，请参阅《AWS 登录 用户指南》中的 Signing in as the root user。
2. 为您的根用户启用多重身份验证（MFA）。

   有关说明，请参阅 I A M 用户指南中的为 AWS 账户 根用户启用虚拟 MFA 设备（控制台）。

创建具有管理访问权限的用户

1. 启用 IAM Identity Center。

   有关说明，请参阅《AWS IAM Identity Center 用户指南》中的 Enabling AWS IAM Identity Center。
2. 在 IAM Identity Center 中，为用户授予管理访问权限。

   有关使用 IAM Identity Center 目录 作为身份源的教程，请参阅《用户指南》 IAM Identity Center 目录中的使用默认设置配置AWS IAM Identity Center 用户访问权限。

以具有管理访问权限的用户身份登录

- 要使用您的 IAM Identity Center 用户身份登录，请使用您在创建 IAM Identity Center 用户时发送到您的电子邮件地址的登录网址。

  有关使用 IAM Identity Center 用户登录的帮助，请参阅AWS 登录 用户指南中的登录 AWS 访问门户。

将访问权限分配给其他用户

1. 在 IAM Identity Center 中，创建一个权限集，该权限集遵循应用最低权限的最佳做法。

   有关说明，请参阅《AWS IAM Identity Center 用户指南》中的 Create a permission set。
2. 将用户分配到一个组，然后为该组分配单点登录访问权限。

   有关说明，请参阅《AWS IAM Identity Center 用户指南》中的 Add groups。

为创建 IAM 权限 HealthOmics

要使用 HealthOmics，请配置以下 IAM 权限：

- IAM 基于身份的策略供您账户中的用户访问。 HealthOmics
- 用于代表您 HealthOmics 访问资源的 IAM 服务角色。
- 您的用户在其他服务（例如 Lake Formation 和 Amazon ECR）中的权限，以及该 HealthOmics 服务访问资源的权限。

有关为配置 IAM 权限的更多信息 HealthOmics，请参阅的 IAM 权限 HealthOmics。

Connect 连接外部代码存储库

使用 AWS HealthOmics，您可以通过使用基于 Git 的存储库来管理工作流程。 AWS CodeConnections HealthOmics 使用此连接访问您的源代码存储库。

在使用外部代码存储库之前，请按照设置连接指南开始使用 AWS CodeConnections。确认您已为 AWS 账户创建了正确的 IAM 策略和权限。有关支持的 Git 提供程序列表和更多信息，请参阅我可以为哪些第三方提供商创建连接？ 。

创建连接

要创建与首选存储库提供商的连接，请按照创建连接教程进行操作。

将 Amazon Q CLI 与 HealthOmics

Amazon Q CLI 提供与的自然语言交互 AWS HealthOmics，允许您使用对话命令执行复杂的基因组工作流程和分析任务。要使用 Amazon Q CLI，请务必为 HealthOmics 其他服务（例如 Amazon ECR 或 Amazon S3）配置 IAM 权限，以便 Amazon Q 访问其资源。 CloudWatch

A HealthOmics gentic 生成式 AI 教程为配置上下文文件以及支持 Amazon Q CLI 创建、运行和优化 AWS HealthOmics 工作流程提供了 step-by-step指导。

入门 HealthOmics

首先 HealthOmics，请确保您已正确设置您的 IAM 权限和角色 HealthOmics。

在控制台中使用 Ready2Run 工作流程 HealthOmics

以下练习显示了如何使用 Ready2Run 工作流程。Ready2Run 工作流程已预先配置了运行工作流程所需的参数和工具参考。工作流程发布者提供示例数据，因此您无需创建自己的数据。

1. 打开 HealthOmics 管理控制台。
2. 选择左上角的导航窗格 ()，然后选择 Ready2Run 工作流程。
3. 在 Ready2Run 工作流程页面上，选择工作流程。ESMFold for up to 800 residues控制台将打开该工作流程的详细信息页面。
4. 详细信息选项卡提供有关工作流程的信息。要试用工作流程，请在页面的右上角选择 “开始运行”。
5. 在 “指定运行详细信息” 页面中，输入运行名称。
6. 为运行输出输入或选择一个 Amazon S3 位置。
7. 对于运行元数据保留模式，选择是保留还是删除 runmeta 数据。
8. 在服务角色面板中，选择创建并使用新的服务角色。
9. 选择下一步。
10. 在 “添加参数值” 页面上，选择 “使用 Ready2Run 测试数据运行工作流”。
11. 选择下一步。
12. 查看您的输入，然后选择 “开始运行”。

Amazon Q CLI 的示例提示

Amazon Q CLI AWS HealthOmics 可以使用自然语言命令运行基因组工作流程和分析任务。以下示例提示允许您创建工作流程、管理运行和分析基因组数据。有关更多信息和示例提示 HealthOmics，请参阅上 GitHub的 A HealthOmics gentic 生成式 AI 教程。

- “创建一个 WDL 1.1 工作流程文件`main.wdl`，因为它将在该文件上运行。 HealthOmics该工作流程将以参考基因组作为输入和成对的 fastq 文件。它将使用 BWA 对参考基因组进行索引，然后将每对 fastq 文件映射到参考文献。最后，将每个映射的 BAM 合并到一个 BAM 文件中，然后输出这个文件，它是 bai 索引。”

- “Package 将工作流程打包并在其中创建 HealthOmics”
- “更新 inputs.json 文件以使用我的 Amazon S3 存储桶中的真实文件`omics-my-bucket-with-genome-data`”（提供特定的亚马逊 S3 存储桶位置，或者让 Amazon Q 探索）
- “在我的 Amazon ECR 存储库中找到合适的容器并更新 inputs.json 以使用这些容器”
- “查找或创建合适的 IAM 角色以在运行工作流程时使用”
- “为我的工作流程创建运行缓存”
- “在中运行工作流程 HealthOmics”
- “检查运行状态”

> **Warning**
>
> 使用 Amazon Q CLI 时，请先查看所有生成的内容和建议的操作，然后再继续。提供反馈以提高响应质量并满足您的工作流程要求。有关更多信息，请参阅 Amazon Q 的安全注意事项和最佳实践。

中的私有工作流程 HealthOmics

当您想要创建自己的工作流程定义时，请使用私有工作流程。工作流定义指定有关工作流的信息并定义工作流任务。运行是对工作流程的单次调用，而任务是运行中的单个进程。

HealthOmics 支持您使用工作流描述语言 (WDL)、通用工作流语言 (CWL) 或 Nextflow 创建的工作流定义。

HealthOmics 工作流程提供以下可选功能：

- Run groups— 您可以将私有工作流程添加到运行组以控制计算使用量。运行组是共享一组资源限制的工作流程运行的集合，例如最大并发运行次数和最大运行持续时间。您可以设置这些限制来控制运行组消耗的计算资源。
- Call caching— 您可以使用呼叫缓存来保存和重用任务输出，从而缩短运行时间并节省计算成本。
- Sharing workflows— 您可以与同一地区的其他 AWS 账户 人共享您的私有工作流程。
- Workflow versions— 您可以创建私有工作流程的版本。工作流版本控制让用户能够选择何时开始使用更新的功能。工作流程版本是不可变的，并且提供的数据来源级别与工作流程相同。

有关为工作流程配置 IAM 权限的信息，请参阅的 IAM 权限 HealthOmics。

有关如何使用 HealthOmics 私有工作流程的完整示例，请参阅 HealthOmics Github 教程或 AWS 研讨会端到端教程 HealthOmics。

主题

- 在中创建私有工作流程 HealthOmics
- 使用自述文件
- 工作流程版本控制在 HealthOmics
- 开始 HealthOmics 跑步
- 删除中的跑步和跑步组 HealthOmics
- 创建 HealthOmics 跑步组
- HealthOmics 运行时调用缓存
- 共享 HealthOmics 工作流程

在中创建私有工作流程 HealthOmics

私有工作流程取决于您在创建工作流程之前创建和配置的各种资源：

- Workflow definition file：用WDL、Nextflow或写入的工作流程定义文件CWL。工作流定义为使用该工作流的运行指定输入和输出。它还包括工作流程的运行和运行任务规范，包括计算和内存要求。工作流程定义文件必须采用`.zip`格式。有关更多信息，请参阅中的工作流定义文件 HealthOmics。
- Parameter template file（可选）：写入的参数模板文件JSON。创建文件来定义运行参数，或者为您 HealthOmics 生成参数模板。有关更多信息，请参阅 HealthOmics 工作流程的参数模板文件。
- Amazon ECR container images：为工作流程创建容器映像并将其存储在私有 Amazon ECR 存储库中。
- Sentieon licenses（可选）：申请Sentieon许可证，以便在私有工作流程中使用该Sentieon软件。

或者，您可以在创建工作流程之前或之后对工作流程定义运行 linter。本linter主题描述了中可用的 linter。 HealthOmics

主题

- 中的工作流程定义文件 HealthOmics
- HealthOmics 工作流程的参数模板文件
- Amazon ECR 中用于私有工作流程的容器镜像
- 为私有工作流程申请 Sentieon 许可证
- 中的工作流程提示 HealthOmics
- 创建或更新工作流程

中的工作流程定义文件 HealthOmics

您可以使用工作流定义来指定有关工作流、运行和运行中的任务的信息。您可以使用工作流定义语言在一个或多个文件中创建工作流定义。 HealthOmics 支持用 WDL、Nextflow 或 CWL 编写的工作流程定义。有关每种语言的详细信息，请参阅 为工作流程编写工作 HealthOmics 流定义

您可以在工作流定义中指定以下类型的信息：

- Language version— 工作流程定义的语言和版本。
- Compute and memory— 工作流程中任务的计算和内存需求。
- Inputs— 工作流任务的输入位置。有关更多信息，请参阅 HealthOmics 运行输入。

- Outputs— 保存任务生成的输出的位置。
- Task resources— 每项任务的计算和内存要求。
- Accelerators— 任务所需的其他资源，例如加速器。

主题

HealthOmics 工作流程定义要求

工作 HealthOmics 流定义文件必须满足以下要求：

- 任务必须定义 input/output 参数、Amazon ECR 容器存储库和运行时规范，例如内存或 CPU 分配。
- 验证您的 IAM 角色是否具有所需的权限。
  - 您的工作流程可以访问来自 AWS 资源（例如 Amazon S3）的输入数据。
  - 您的工作流程可以在需要时访问外部存储库服务。
- 在工作流程定义中声明输出文件。要将中间运行文件复制到输出位置，请将其声明为工作流程输出。
- 输入和输出位置必须与工作流程位于同一个区域中。
- HealthOmics 存储工作流输入必须处于`ACTIVE`状态。 HealthOmics 不会导入带有`ARCHIVED`状态的输入，从而导致工作流程失败。有关 Amazon S3 对象输入的信息，请参阅HealthOmics 运行输入。
- 如果您的 ZIP 存档包含单个工作流程定义或名为 “main” 的文件，则工作流程的main位置是可选的。
  - 路径示例：`workflow-definition/main-file.wdl`
- 在通过 Amazon S3 或本地驱动器创建工作流程之前，请创建包含工作流程定义文件和任何依赖项（例如子工作流程）的 zip 存档。
- 我们建议您在工作流程中将 Amazon ECR 容器声明为验证亚马逊 ECR 权限的输入参数。

Nextflow 的其他

- /bin

  Nextflow 工作流程定义可能包括带有可执行脚本的 /bin 文件夹。此路径对任务具有只读权限和可执行访问权限。依赖这些脚本的任务应使用由相应脚本解释器构建的容器。最佳做法是直接给口译员打电话。例如：

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

- includeConfig

  基于 NextFlow 的工作流程定义可以包括 nextflow.config 文件，这些文件有助于抽象参数定义或流程资源配置文件。要支持在多个环境中开发和执行 Nextflow 管道，请使用 HealthOmics特定的配置，使用 includeConfig 指令将其添加到全局配置中。要保持可移植性，请使用以下代码将工作流程配置为仅在 HealthOmics 运行时包含文件：

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

- Reports

  HealthOmics 不支持引擎生成的 dag、trace 和执行报告。您可以使用和 GetRunTask API 调用的组合生成跟踪报告 GetRun 和执行报告的替代方案。

其他 CWL 注意事项：

- Container image uri interpolation

  HealthOmics 允许的 dockerPull 属性成为内联 javascript 表达式。 DockerRequirement 例如：

```
requirements:
  DockerRequirement:
```

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

这允许您将容器映 URIs 像指定为工作流程的输入参数。

- Javascript expressions

  Javascript 表达式必须`strict mode`兼容。

- Operation process

  HealthOmics 不支持 CWL 操作进程。

对 HealthOmics 工作流定义语言的版本支持

HealthOmics 支持用 Nextflow、WDL 或 CWL 编写的工作流程定义文件。以下各节提供有关这些语言的 HealthOmics 版本支持的信息。

主题

- WDL 版本支持
- CWL 版本支持
- Nextflow版本支持

WDL 版本支持

HealthOmics 支持 WDL 规范的 1.0、1.1 版本和 WDL 规范的开发版本。

每个 WDL 文档都必须包含一个版本声明，以指定它所遵循的规范版本（主要版本和次要版本）。有关版本的更多信息，请参阅 WDL 版本控制

WDL 规范的 1.0 和 1.1 版本不支持该`Directory`类型。要将该`Directory`类型用于输入或输出，请在文件第一行development中将版本设置为：

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

CWL 版本支持

HealthOmics 支持 CWL 语言的 1.0、1.1 和 1.2 版本。

您可以在 CWL 工作流程定义文件中指定语言版本。有关 CWL 的更多信息，请参阅 CWL 用户指南

Nextflow版本支持

HealthOmics 支持三个 Nextflow 稳定版本。Nextflow 通常每六个月发布一次稳定版本。 HealthOmics 不支持每月发布的 “边缘” 版本。

HealthOmics 支持每个版本中已发布的功能，但不支持预览功能。

支持的版本

HealthOmics 支持以下 Nextflow 版本：

- Nextflow v22.04.01 DSL 1 和 DSL 2
- Nextflow v23.10.0 DSL 2（默认）
- Nextflow v24.10.8 DSL 2

要将您的工作流程迁移到支持的最新版本 (v24.10.8)，请按照 Next flow 升级指南进行操作。

从 Nextflow v23 迁移到 v24 时有一些重大更改，如 Nextflow 迁移指南的以下部分所述：

- 24.04 中的重大变化
- 24.10 中的重大变化

检测和处理 Nextflow 版本

HealthOmics 检测您指定的 DSL 版本和 Nextflow 版本。它会根据这些输入自动确定要运行的最佳 Nextflow 版本。

DSL 版本

HealthOmics 在您的工作流程定义文件中检测请求的 DSL 版本。例如，您可以指定:`nextflow.enable.dsl=2`.

HealthOmics 默认情况下支持 DSL 2。如果在工作流程定义文件中指定，它可提供与 DSL 1 的向后兼容性。

- 如果你指定 DSL 2，则 HealthOmics 运行 Nextflow v23.10.0，除非你指定 Nextflow v22.04.0 或 v24.10.8。
- 如果你指定 DSL 1，则 HealthOmics 运行 Nextflow v22.04 DSL1（唯一支持的运行 DSL 1 的版本）。

- 如果您未指定 DSL 版本，或者由于任何原因（例如工作流程定义文件中的语法错误）而 HealthOmics 无法解析 DSL 信息，则 HealthOmics 默认为 DSL 2 并运行 Nextflow v23.10.0。
- 要将工作流程从 DSL 1 升级到 DSL 2 以利用最新的 Nextflow 版本和软件功能，请参阅从 DSL 1 迁移。

下一流版本

HealthOmics 如果你提供了 Nextflow 配置文件 (nextflow.config) 中请求的 Nextflow 版本。我们建议您在文件末尾添加`nextflowVersion`子句，以避免包含的配置中出现任何意外覆盖。有关更多信息，请参阅 Nextflow 配置。

您可以使用以下语法指定 Nextflow 版本或一系列版本：

```
// 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 按如下方式处理 Nextflow 版本信息：

- 如果您=使用指定 HealthOmics 支持的确切版本，则 HealthOmics 使用该版本。
- 如果您使用!指定不支持的确切版本或一系列版本，则 HealthOmics 会引发异常并导致运行失败。如果您想严格处理版本请求，请考虑使用此选项，如果请求包含不支持的版本，则会很快失败。
- 如果指定版本范围，则 HealthOmics 使用该范围内支持的最新版本，除非该范围包括 v24.10.8。在这种情况下， HealthOmics 优先考虑较早的版本。例如，如果该范围同时涵盖 v23.10.0 和 v24.10.8，则选择 v23.10.0。 HealthOmics
- 如果没有请求的版本，或者请求的版本无效或由于任何原因无法解析：
  - 如果你指定了 DSL 1，则 HealthOmics 运行 Nextflow v22.04。

- 否则， HealthOmics 运行 Nextflow v23.10.0。

您可以检索有关每次运行时 HealthOmics 使用的 Nextflow 版本的以下信息：

- 运行日志包含有关 HealthOmics 用于运行的实际 Nextflow 版本的信息。
- HealthOmics 如果与您请求的版本不直接匹配，或者需要使用与您指定的版本不同的版本，则会在运行日志中添加警告。
- 对 GetRun API 操作的响应包括一个字段 (`engineVersion`)，其中包含 HealthOmics 用于运行的实际 Nextflow 版本。例如：

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

HealthOmics 任务的计算和内存要求

HealthOmics 在 omics 实例中运行您的私有工作流程任务。 HealthOmics 提供了多种实例类型以适应不同类型的任务。每种实例类型都有固定的内存和 vCPU 配置（对于加速计算实例类型，还有固定的 GPU 配置）。使用 omics 实例的成本因实例类型而异。有关详细信息，请参阅定HealthOmics 价页面。

对于工作流中的任务，您可以在工作流定义文件CPUs 中指定所需的内存和 v。当工作流任务运行时，HealthOmics 分配最小的组学实例，以容纳请求的内存和 v. CPUs 例如，如果任务需要 64 GiB 内存和 8 vCPUs， HealthOmics 则选择。`omics.r.2xlarge`

我们建议您查看实例类型并设置请求的 v CPUs 和内存大小，使其与最能满足您需求的实例相匹配。即使实例类型有额外的 v CPUs 和内存，任务容器也会使用您在工作流程定义文件中指定的数量CPUs 和内存大小。

以下列表包含有关 vCPU 和内存分配的其他信息：

- 容器资源分配是硬性限制。如果任务内存不足或尝试使用其他 vCPUs ，则该任务会生成错误日志并退出。
- 如果您未指定任何计算或内存要求，请 HealthOmics 选择omics.c.large并默认为具有 1 个 vCPU 和 1 GiB 内存的配置。
- 您可以请求的最低配置为 1 个 vCPU 和 1 GiB 内存。
- 如果您指定 v CPUs、memory 或 GPUs ，则超过支持的实例类型，则 HealthOmics 会抛出一条错误消息，并且工作流程无法通过验证

- 如果指定小数单位，则向上 HealthOmics 舍入到最接近的整数。
- HealthOmics 为管理和日志代理保留少量内存 (5%)，因此任务中的应用程序可能并不总是可以使用全部内存分配。
- HealthOmics 匹配实例类型以满足您指定的计算和内存要求，并且可以混合使用几代硬件。因此，同一任务的任务运行时间可能会有一些细微的差异。

这些主题提供了有关 HealthOmics 支持的实例类型的详细信息。

主题

> **Note**
>
> 对于标准、计算和内存优化型实例，如果实例需要更高的吞吐量，请增加实例带宽大小。小于 16 vCPUs（大小为 4xl 及更小）的 Amazon EC2 实例可能会出现吞吐量激增的情况。有关 Amazon EC2 实例吞吐量的更多信息，请参阅亚马逊 EC2 可用实例带宽。

标准实例类型

对于标准实例类型，配置旨在平衡计算能力和内存。

HealthOmics 支持以下区域的 32xlarge 和 48xlarge 实例：美国西部（俄勒冈）和美国东部（弗吉尼亚北部）。

实例	v 的数量 CPUs	内存
omics.m.large	2	8 GiB
omics.m.xlarge	4	16 GiB
omics.m.2xlarge	8	32 GiB
omics.m.4xlarge	16	64 GiB

实例	v 的数量 CPUs	内存
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

计算优化型实例

对于计算优化的实例类型，配置具有更高的计算能力和更少的内存。

HealthOmics 支持以下区域的 32xlarge 和 48xlarge 实例：美国西部（俄勒冈）和美国东部（弗吉尼亚北部）。

实例	v 的数量 CPUs	内存
omics.c.large	2	4 GiB
omics.c.xlarge	4	8 GiB
omics.c.2xlarge	8	16 GiB
omics.c.4xlarge	16	32 GiB
omics.c.8xlarge	32	64 GiB
omics.c.12xlarge	48	96 GiB
omics.c.16xlarge	64	128 GiB
omics.c.24xlarge	96	192 GiB
omics.c.32xlarge	128	256 GiB

实例	v 的数量 CPUs	内存
omics.c.48xlarge	192	384 GiB

内存优化型实例

对于内存优化的实例类型，配置具有更低的计算能力和更多的内存。

HealthOmics 支持以下区域的 32xlarge 和 48xlarge 实例：美国西部（俄勒冈）和美国东部（弗吉尼亚北部）。

实例	v 的数量 CPUs	内存
omics.r.large	2	16 GiB
omics.r.xlarge	4	32 GiB
omics.r.2xlarge	8	64 GiB
omics.r.4xlarge	16	128 GiB
omics.r.8xlarge	32	256 GiB
omics.r.12xlarge	48	384 GiB
omics.r.16xlarge	64	512 GiB
omics.r.24xlarge	96	768 GiB
omics.r.32xlarge	128	1024 GiB
omics.r.48xlarge	192	1536 GiB

加速计算实例

您可以选择为工作流程中的每个任务指定 GPU 资源，以便为该任务 HealthOmics分配加速计算实例。有关如何在工作流定义文件中指定 GPU 信息的信息，请参阅工作 HealthOmics 流程定义中的任务加速器。

如果您指定支持多种实例类型的 GPU，请根据可用性 HealthOmics 选择实例类型。如果两种实例类型都可用，则 HealthOmics 优先选择成本较低的实例。

以色列（特拉维夫）地区不支持 G4 实例。亚太地区（新加坡）地区不支持 G5 实例。

主题

G6 和 G6e 实例类型

HealthOmics 支持以下 G6 加速计算实例配置。所有 omics.g6 实例都使用 Nvidia L4 或 Nvidia L4 A10G。 GPUs

HealthOmics 支持以下区域的 G6 和 G6e 实例：美国西部（俄勒冈）和美国东部（弗吉尼亚北部）。

实例	v 的数量 CPUs	内存	的数量 GPUs	GPU 内存	
omics.g6.xlarge	4	16 GiB	1	48 GiB	
omics.g6.2xlarge	8	32 GiB	1	48 GiB	
omics.g6.4xlarge	16	64 GiB	1	48 GiB	
omics.g6.8xlarge	32	128 GiB	1	48 GiB	
omics.g6.12xlarge	48	192 GiB	4	192 GiB	
omics.g6.16xlarge	64	256 GiB	1	48 GiB	
omics.g6.24xlarge	96	192 GiB	4	192 GiB	

所有 omics.g6e 实例都使用 Nvidia L40。 GPUs

实例	v 的数量 CPUs	内存	的数量 GPUs	GPU 内存	
omics.g6e .xlarge	4	32 GiB	1	24 GiB	
omics.g6e .2xlarge	8	64 GiB	1	24 GiB	
omics.g6e .4xlarge	16	128 GiB	1	24 GiB	
omics.g6e .8xlarge	32	256 GiB	1	24 GiB	
omics.g6e .12xlarge	48	384 GiB	4	96 GiB	
omics.g6e .16xlarge	64	512 GiB	1	96 GiB	
omics.g6e .24xlarge	96	768 GiB	4	96 GiB	

G4 和 G5 实例

HealthOmics 支持以下 G4 和 G5 加速计算实例配置。

所有 omics.g5 实例都使用 Nvidia L4 A10G、Nvidia Tesla A10G 或 Nvidia Tesla T4 A10G。 GPUs

实例	v 的数量 CPUs	内存	的数量 GPUs	GPU 内存	
omics.g5. xlarge	4	16 GiB	1	24 GiB	

实例	v 的数量 CPUs	内存	的数量 GPUs	GPU 内存	
omics.g5.2xlarge	8	32 GiB	1	24 GiB	
omics.g5.4xlarge	16	64 GiB	1	24 GiB	
omics.g5.8xlarge	32	128 GiB	1	24 GiB	
omics.g5.12xlarge	48	192 GiB	4	96 GiB	
omics.g5.16xlarge	64	256 GiB	1	24 GiB	
omics.g5.24xlarge	96	384 GiB	4	96 GiB	

所有 omics.g4dn 实例都使用 Nvidia Tesla T4 或 Nvidia Tesla T4 A10G。 GPUs

实例	v 的数量 CPUs	内存	的数量 GPUs	GPU 内存	
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	

实例	v 的数量 CPUs	内存	的数量 GPUs	GPU 内存	
omics.g4dn.12xlarge	48	192 GiB	4	64 GiB	
omics.g4dn.16xlarge	64	256 GiB	1	24 GiB	

工作 HealthOmics 流程定义中的任务输出

您可以在工作流定义中指定任务输出。默认情况下，当工作流程完成时，会 HealthOmics 丢弃所有中间任务文件。要导出中间文件，请将其定义为输出。

如果您使用呼叫缓存，则 HealthOmics 会将任务输出保存到缓存中，包括您定义为输出的任何中间文件。

以下主题包括每种工作流定义语言的任务定义示例。

主题

- WDL 的任务输出
- 下一流的任务输出
- CWL 的任务输出

WDL 的任务输出

对于用 WDL 编写的工作流程定义，请在顶级工作流程outputs部分中定义您的输出。

HealthOmics

主题

- STDOUT 的任务输出
- STDERR 的任务输出
- 任务输出到文件
- 任务输出到文件数组

STDOUT 的任务输出

此示例创建了一个名为的任务SayHello，该任务将 STDOUT 内容回显到任务输出文件中。WDL stdout 函数捕获 STDOUT 内容（在本例中为输入字符串 Hello World！）在文件中stdout_file。

由于 HealthOmics 会为所有 STDOUT 内容创建日志，因此输出也会与任务的其他 STDERR CloudWatch 日志信息一起显示在 “日志” 中。

```
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()
    }
}
```

STDERR 的任务输出

此示例创建了一个名为的任务SayHello，该任务将 STDERR 内容回显到任务输出文件中。WDL stderr 函数捕获 STDERR 内容（在本例中为输入字符串 Hello World！）在文件中stderr_file。

由于 HealthOmics 会为所有 STDERR 内容创建日志，因此输出将与任务的其他 STDERR CloudWatch 日志信息一起显示在日志中。

```
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()
    }
}
```

任务输出到文件

在此示例中，该 SayHello 任务创建了两个文件（message.txt 和 info.txt），并将这些文件明确声明为命名的输出（message_file 和 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"
    }
}
```

任务输出到文件数组

在此示例中，GenerateGreetings任务生成一组文件作为任务输出。该任务为输入数组的每个成员动态生成一个问候语文件names。由于文件名要等到运行时才知道，因此输出定义使用 WDL glob () 函数输出与该模式匹配的所有文件。*_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")
    }
 }
```

下一流的任务输出

对于在 Nextflow 中编写的工作流程定义，请定义 P ublishDir 指令以将任务内容导出到您的输出 Amazon S3 存储桶。将 p ublishDir 值设置为。`/mnt/workflow/pubdir`

HealthOmics 要将文件导出到 Amazon S3，文件必须位于此目录中。

如果任务生成一组输出文件作为后续任务的输入，我们建议您将这些文件分组到一个目录中，然后将该目录作为任务输出发出。枚举每个单独的文件可能会导致底层文件系统出现 I/O 瓶颈。例如：

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

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

CWL 的任务输出

对于用 CWL 编写的工作流程定义，您可以使用任务指定任务输出。CommandLineTool以下各节显示了定义不同类型输出的CommandLineTool任务示例。

主题

- STDOUT 的任务输出
- STDERR 的任务输出
- 任务输出到文件
- 任务输出到文件数组

STDOUT 的任务输出

此示例创建了一个CommandLineTool任务，该任务将 STDOUT 内容回显到名为的文本输出文件中。output.txt例如，如果您提供以下输入，则生成的任务输出为 Hello World！ 在output.txt文件中。

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

该outputs指令指定输出名称为example_out，其类型为stdout。要使下游任务消耗此任务的输出，它将输出称为example_out。

由于 HealthOmics 会为所有 STDERR 和 STDOUT 内容创建日志，因此输出也会与任务的其他 STDERR CloudWatch 日志信息一起显示在日志中。

```
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
```

STDERR 的任务输出

此示例创建了一个CommandLineTool任务，该任务将 STDERR 内容回显到名为的文本输出文件中。stderr.txt该任务会修改，baseCommand以便echo写入 STDERR（而不是 STDOUT）。

该outputs指令指定输出名称为stderr_out，其类型为stderr。

由于 HealthOmics 会为所有 STDERR 和 STDOUT 内容创建日志，因此输出将与任务的其他 STDERR CloudWatch 日志信息一起显示在日志中。

```
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
```

任务输出到文件

此示例创建了一个CommandLineTool任务，该任务根据输入文件创建压缩的 tar 存档。您可以将档案的名称作为输入参数 (archive_name) 提供。

该outputs指令指定archive_file输出类型为File，并使用对输入参数的引用绑定archive_name到输出文件。

```
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
```

任务输出到文件数组

在此示例中，CommandLineTool任务使用touch命令创建文件数组。该命令使用files-to-create输入参数中的字符串来命名文件。该命令输出文件数组。该数组包括工作目录中与glob模式匹配的所有文件。此示例使用与所有文件匹配的通配符模式 (“*”)。

```
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
```

工作 HealthOmics 流程定义中的任务资源

在工作流定义中，为每项任务定义以下内容：

- 任务的容器镜像。有关更多信息，请参阅 Amazon ECR 中用于私有工作流程的容器镜像。
- 任务所需的数量 CPUs 和内存。有关更多信息，请参阅 HealthOmics 任务的计算和内存要求。

HealthOmics 忽略每项任务的任何存储规范。 HealthOmics 提供运行中的所有任务都可以访问的运行存储空间。有关更多信息，请参阅 在 HealthOmics 工作流程中运行存储类型。

WDL

```
task my_task {
   runtime {
      container: "<aws-account-id>.dkr.ecr.<aws-region>.amazonaws.com/<image-name>"
```

```
      cpu: 2
      memory: "4 GB"
   }
   ...
}
```

对于 WDL 工作流程， HealthOmics 对于因服务错误而失败的任务（API 请求返回 5XX HTTP 状态码），最多可尝试重试两次。有关任务重试次数的更多信息，请参阅任务重试次数。

您可以通过在 WDL 定义文件中为任务指定以下配置来选择退出重试行为：

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

NextFlow

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

CWL

```
cwlVersion: v1.2
class: CommandLineTool
requirements:
    DockerRequirement:
        dockerPull: "<aws-account-id>.dkr.ecr.<aws-region>.amazonaws.com/<image-
name>"
    ResourceRequirement:
        coresMax: 2
        ramMax: 4000 # specified in mebibytes
```

工作 HealthOmics 流程定义中的任务加速器

在工作流程定义中，您可以选择为任务指定 GPU 加速器规格。 HealthOmics 支持以下加速器规格值以及支持的实例类型：

加速器规格	Healthomics 实例类型				
nvidia-tesla-t4	G4				
nvidia-tesla-t4-a10g	G4 和 G5				
nvidia-tesla-a10g	G5				
nvidia-l4-a10g	G5 和 G6				
nvidia-l4	G6				
nvidia-l40s	G6e				

如果您指定的加速器类型支持多种实例类型，请根据可用容量 HealthOmics 选择实例类型。如果两种实例类型都可用，则 HealthOmics 优先选择成本较低的实例。

有关实例类型的详细信息，请参阅加速计算实例。

在以下示例中，工作流定义指定nvidia-l4为加速器：

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"
```

为工作流程编写工作 HealthOmics 流定义

HealthOmics 支持用 WDL、Nextflow 或 CWL 编写的工作流程定义。要了解有关这些工作流程语言的更多信息，请参阅 WDL、Nextflow 或 CWL 的规范。

HealthOmics 支持三种工作流程定义语言的版本管理。有关更多信息，请参阅 对 HealthOmics 工作流定义语言的版本支持 。

主题

- 在 WDL 中编写工作流程
- 在 Nextflow 中编写
- 在 CWL 中编写工作流程
- 工作流程定义示例
- WDL 工作流程定义示例

在 WDL 中编写工作流程

下表显示了 WDL 中的输入如何映射到匹配的原始类型或复杂 JSON 类型。类型强制是有限的，只要有可能，类型就应该是显式的。

原始类型

WDL 类型	JSON 类型	示例 WDL	JSON 键和值示例	备注
Boolean	boolean	Boolean b	"b": true	该值必须为小写且不带引号。
Int	integer	Int i	"i": 7	必须不加引号。
Float	number	Float f	"f": 42.2	必须不加引号。
String	string	String s	"s": "characters"	作为 URI 的 JSON 字符串必须映射到要导入的 WDL 文件。
File	string	File f	"f": "s3://amzn-s3-demo-bucket1/path/to/file"	只要为工作流程提供 URIs 的 IAM 角色具有对这些对象的读取权限，就会导入 Amazon S3 和 HealthOmics 存储。不支持其他 URI 方案（例如file://https://、和ftp://）。URI 必须指定一个对象。它不能是目录，这意味着它不能以结尾/。
Directory	string	Directory d	"d": "s3://bucket/path/"	该Directory 类型不包含在 WDL 1.0 或 1.1 中，因此您需要将该类型添加version

WDL 类型	JSON 类型	示例 WDL	JSON 键和值示例	备注
				`developme nt` 到 WDL 文件的标题中。URI 必须是 Amazon S3 URI，且前缀必须以 “/” 结尾。该目录的所有内容将以递归方式复制到工作流程中，作为一次下载。`Directory` 应仅包含与工作流程相关的文件。

WDL 中的复杂类型是由原始类型组成的数据结构。诸如列表之类的数据结构将转换为数组。

复杂类型

WDL 类型	JSON 类型	示例 WDL	JSON 键和值示例	备注
`Array`	`array`	`Array[Int] nums`	“`nums": [1, 2, 3]`	数组的成员必须遵循 WDL 数组类型的格式。
`Pair`	`object`	`Pair[Stri ng, Int] str_to_i`	“`str_to_i ": {"left": "0", "right": 1}`	该对的每个值都必须使用其匹配的 WDL 类型的 JSON 格式。
`Map`	`object`	`Map[Int, String] int_to_st ring`	`"int_to_s tring": { 2:`	地图中的每个条目都必须使用其匹配的 WDL 类

WDL 类型	JSON 类型	示例 WDL	JSON 键和值示例	备注
			`"hello", 1: "goodbye" }`	型的 JSON 格式。
`Struct`	`object`	`struct SampleBamAndIndex { String sample_name File bam File bam_index } SampleBamAndIndex b_and_i`	`"b_and_i": { "sample_name": "NA12878", "bam": "s3://amzn-s3-demo-bucket1/NA12878.bam", "bam_index": "s3://amzn-s3-demo-bucket1/NA12878.bam.bai" }`	结构成员的名称必须与 JSON 对象键的名称完全匹配。每个值都必须使用匹配的 WDL 类型的 JSON 格式。
`Object`	不适用	不适用	不适用	WDL `Object` 类型已过时，`Struct`在所有情况下都应替换为。

工作 HealthOmics 流引擎不支持限定或命名间隔的输入参数。WDL 语言未指定对限定参数的处理及其与 WDL 参数的映射，因此可能含糊不清。出于这些原因，最佳做法是在顶级（主）工作流定义文件中声明所有输入参数，然后使用标准 WDL 机制将它们传递给子工作流调用。

在 Nextflow 中编写

HealthOmics 支持 Next DSL1 flow 和。 DSL2有关更多信息，请参阅 Nextflow版本支持。

Nextflow 基 DSL2 于 Groovy 编程语言，因此参数是动态的，并且可以使用与 Groovy 相同的规则进行类型强制。输入 JSON 提供的参数和值可在工作流程的参数 (params) 映射中找到。

> **Note**
>
> HealthOmics 支持 Nextflow 版本 v23.10（但不是 v22.04）的nf-schema和nf-validation插件。

以下信息与在 Nextflow v23.10 工作流程中使用这些插件有关：

- HealthOmics 预安装 nf 架构 @2 .3.0 和 nf-validation @1 .1.1 插件。 HealthOmics 忽略您在nextflow.config文件中指定的任何其他插件版本。
- 在工作流程运行期间，您无法检索其他插件。
- 在 Nextflow v24.04 及更高版本中，该nf-validation插件已重命名为。nf-schema有关更多信息，请参阅 Next GitHub flow 存储库中的 nf-schema。

使用 Amazon S3 或 HealthOmics URI 构建 Nextflow 文件或路径对象时，只要授予读取权限，它就会使匹配的对象可供工作流程使用。Amazon S3 URIs 允许使用前缀或目录。有关示例，请参阅 亚马逊 S3 输入参数格式。

HealthOmics 支持在 Amazon S3 URIs 或 HealthOmics 存储 URIs中使用全局模式。在工作流程定义中使用 Glob 模式来创建path或file频道。

对于用 Nextflow 编写的工作流程，请定义 PublishDir 指令以将任务内容导出到输出 Amazon S3 存储桶。如以下示例所示，将 p ublishDir 值设置为。/mnt/workflow/pubdir要将文件导出到 Amazon S3，文件必须位于此目录中。

```
 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
  """
```

```
}
```

在 CWL 中编写工作流程

用通用工作流语言 (CWL) 编写的工作流程提供的功能与用 WDL 和 Nextflow 编写的工作流程类似。您可以使用 Amazon S3 或 HealthOmics 存储 URIs 作为输入参数。

如果您在子工作流程的 SecondaryFile 中定义输入，请在主工作流程中添加相同的定义。

HealthOmics 工作流程不支持操作流程。要了解有关 CWL 工作流中操作流程的更多信息，请参阅 CWL 文档。

要转换现有的 CWL 工作流定义文件以供使用 HealthOmics，请进行以下更改：

- 将所有 Docker 容器 URIs 替换为亚马逊 EC URIs R。
- 确保在主工作流程中将所有工作流文件声明为输入，并且所有变量都已明确定义。
- 确保所有 JavaScript 代码都是严格模式投诉。

应为使用的每个容器定义 CWL 工作流程。不建议使用固定的亚马逊 ECR URI 对 DockerPull 条目进行硬编码。

以下是用 CWL 编写的工作流程示例。

```
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
```

以下文件定义了copy.cwl任务。

```
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)"
```

以下是使用 CWL 编写的、具有 GPU 要求的工作流程示例。

```
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: []
```

工作流程定义示例

以下示例显示了 WDL、Nextflow 和 CWL 中相同的工作流程定义。

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
```

WDL 工作流程定义示例

以下示例显示了在 WDL BAM 中从CRAM转换为的私有工作流程定义。t CRAM o BAM 工作流定义了两个任务并使用genomes-in-the-cloud容器中的工具，该工具如示例所示，并且已公开发布。

以下示例说明如何将 Amazon ECR 容器作为参数包括在内。这 HealthOmics 允许在容器开始运行之前验证其访问权限。

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

以下示例说明当文件位于 Amazon S3 存储桶中时，如何指定要在运行中使用哪些文件。

```
{
      "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"
```

```
  }
```

如果要指定序列存储中的文件，请使用序列存储的 URI 进行指示，如以下示例所示。

```
{
      "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"
  }
```

然后，您可以在 WDL 中定义工作流程，如下所示。

```
 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}"
     }
 }
```

HealthOmics 工作流程的参数模板文件

参数模板定义工作流的输入参数。您可以定义输入参数，使您的工作流程更加灵活和多样。例如，您可以为参考基因组文件的 Amazon S3 位置定义一个参数。参数模板可以通过基于 Git 的存储库服务或本地驱动器提供。然后，用户可以使用各种数据集运行工作流程。

您可以为工作流程创建参数模板， HealthOmics 也可以为您生成参数模板。

参数模板是一个 JSON 文件。在文件中，每个输入参数都是一个命名对象，必须与工作流程输入的名称相匹配。开始运行时，如果您没有为所有必需的参数提供值，则运行将失败。

输入参数对象包括以下属性：

- description— 此必填属性是控制台在 “开始运行” 页面中显示的字符串。此描述也作为运行元数据保留。
- optional— 此可选属性指示输入参数是否为可选参数。如果未指定optional字段，则输入参数为必填项。

以下示例参数模板显示了如何指定输入参数。

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

生成参数模板

HealthOmics 通过解析工作流定义来生成参数模板以检测输入参数。如果您为工作流提供参数模板文件，则文件中的参数将覆盖在工作流定义中检测到的参数。

如以下各节所述，CWL、WDL 和 Nextflow 引擎的解析逻辑略有不同。

主题

- CWL 的参数检测
- WDL 的参数检测
- Nextflow 参数检测

CWL 的参数检测

在 CWL 工作流引擎中，解析逻辑做出了以下假设：

- 任何支持为空的类型都被标记为可选的输入参数。
- 任何支持的非 null 类型都被标记为必填的输入参数。
- 任何具有默认值的参数都被标记为可选的输入参数。
- 描述是从`main`工作流定义的`label`部分中提取的。如果`label`未指定，则描述将为空（空字符串）。

下表显示了 CWL 插值示例。对于每个示例，参数名称均为x。如果参数为必填项，则必须为该参数提供一个值。如果参数是可选的，则无需提供值。

下表显示了原始类型的 CWL 插值示例。

输入	输入/输出示例	必需
`x:` `  type: int`	1 或 2 或...	是

`x:`
`  type: int`
`  default: 2`
	默认值为 2。有效输入为 1 或 2 或...	否

`x:` `  type: int?`	有效输入为 “无”、“1” 或 “2” 或...	否

`x:`
`  type: int?`
`  default: 2`
	默认值为 2。有效输入为 “无”、“1” 或 “2” 或...	否

下表显示了复杂类型的 CWL 插值示例。复杂类型是原始类型的集合。

输入	输入/输出示例	必需

`x:`
`  type: array`
`  items: int`
	[] 或 [1,2,3]	是

`x:`
`  type: array?`
`  items: int`
	无或 [] 或 [1,2,3]	否

`x:` `  type: array`	[] 或 [无、3、无]	是

输入	输入/输出示例	必需
`items: int?`		

`x:`
`  type: array?`
`  items: int?`
	[无] 或 “无” 或 [1,2,3] 或 [无，3] 但不是 []	否

WDL 的参数检测

在 WDL 工作流引擎中，解析逻辑做出了以下假设：

- 任何支持为空的类型都被标记为可选的输入参数。
- 对于支持不可为空的类型：
  - 任何具有字面值或表达式赋值的输入变量都被标记为可选参数。例如：

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

  - 如果没有为输入参数分配任何值或表达式，则它们将被标记为必填参数。
- 描述是从`main`工作流程定义`parameter_meta`中提取的。如果`parameter_meta`未指定，则描述将为空（空字符串）。有关更多信息，请参阅参数元数据的 WDL 规范。

下表显示了 WDL 插值示例。对于每个示例，参数名称均为`x`。如果参数为必填项，则必须为该参数提供一个值。如果参数是可选的，则无需提供值。

下表显示了基元类型的 WDL 插值示例。

输入	输入/输出示例	必需
整数 x	1 或 2 或...	是
整数 x = 2	2	否
Int x = 1+2	3	否
整数 x = y+z	y+z	否

输入	输入/输出示例	必需
整数？ x	无、1 或 2 或...	是
整数？ x = 2	无或 2	否
整数？ x = 1+2	无或 3	否
整数？ x = y+z	无或 y+z	否

下表显示了复杂类型的 WDL 插值示例。复杂类型是原始类型的集合。

输入	输入/输出示例	必需
数组 [整数] x	[1,2,3] 或 []	是
数组 [整数] + x	[1]，但不是 []	是
数组 [整数]？ x	无或 [] 或 [1,2,3]	否
数组 [整数？] x	[] 或 [无、3、无]	是
数组 [整数？] = ？ x	[无] 或 “无” 或 [1,2,3] 或 [无，3] 但不是 []	否
结构示例 {字符串 a，整数 y} 稍后在输入中：示例 mySample	`String a = mySample.a Int y = mySample.y`	是
结构示例 {字符串 a，整数 y} 稍后在输入中：样本？ 我的样本	`if (defined( mySample)) { String a = mySample.a Int y = mySample.y`	否

输入	输入/输出示例	必需		
	}			

Nextflow 参数检测

对于 Nextflow，通过解析文件来 HealthOmics 生成参数模板。`nextflow_schema.json`如果工作流定义不包括架构文件，则 HealthOmics 解析主工作流定义文件。

主题

- 解析架构文件
- 解析主文件
- 嵌套参数
- Nextflow 插值示例

解析架构文件

为了使解析正常工作，请确保架构文件满足以下要求：

- 架构文件名为`nextflow_schema.json`，与主工作流文件位于同一目录中。
- 架构文件是有效的 JSON，如以下任一架构所定义：
  - json 架构。 org/draft/2020-12/schema。
  - json 架构。 org/draft-07/schema。

HealthOmics 解析`nextflow_schema.json`文件以生成参数模板：

- 提取架构properties中定义的所有内容。
- description如果该物业可用，则包括该财产。
- 根据属性的required字段确定每个参数是可选参数还是必填参数。

以下示例显示了定义文件和生成的参数文件。

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

生成的参数模板：

```
{
    "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
    }
}
```

解析主文件

如果工作流程定义不包含`nextflow_schema.json`文件，则 HealthOmics 解析主工作流程定义文件。

HealthOmics 分析在主工作流定义文件和文件中找到的`params`和`nextflow.config`表达式。所有`params`带有默认值的都标记为可选。

为了使解析正常工作，请注意以下要求：

- HealthOmics 仅解析主工作流定义文件。为确保捕获所有参数，我们建议您将所有`params`参数连接到任何子模块和导入的工作流程。
- 配置文件是可选的。如果您定义了一个工作流定义文件，请将其命名`nextflow.config`并放置在与主工作流定义文件相同的目录中。

以下示例显示了定义文件和生成的参数模板。

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

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

生成的参数模板：

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

对于在 nextflow.config 中定义的默认值， HealthOmics 收集其中声明的`params`赋值和参数`params {}`，如以下示例所示。在赋值语句中，`params`必须出现在语句的左侧。

```
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}"
```

生成的参数模板：

```
{
    // 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
    }
}
```

嵌套参数

两者兼nextflow_schema.json而有之，并nextflow.config允许嵌套参数。但是， HealthOmics 参数模板只需要顶级参数。如果您的工作流程使用嵌套参数，则必须提供一个 JSON 对象作为该参数的输入。

架构文件中的嵌套参数

HealthOmics 解析文件params时会跳过嵌套。nextflow_schema.json例如，如果您定义了以下nextflow_schema.json文件：

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

```
            }
        },
        "input_bool": { ... }
    }
}
```

HealthOmics 忽略`input_file`，`input_num`当它生成参数模板时：

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

运行此工作流程时， HealthOmics 需要一个类似于以下内容的`input.json`文件：

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

配置文件中的嵌套参数

HealthOmics 不收集嵌套params在`nextflow.config`文件中，并在解析过程中跳过它们。例如，如果您定义了以下`nextflow.config`文件：

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

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

```
  }
```

HealthOmics 忽略`params.nested.beta`，`params.group.delta`当它生成参数模板时：

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

Nextflow 插值示例

下表显示了主文件中参数的 Nextflow 插值示例。

参数	必需
params.input_fil	是
params.input_file = “s3://bucket/data.json”	否
params.nested.input_file	不适用
params.nested.input_file = “s3://bucket/data.json”	不适用

下表显示了文件中参数的 Nextflow 插值示例。`nextflow.config`

参数	必需
`params.input_file = "s3://bucket/data.json"`	否
`params {`	否

参数	必需
`input_file = "s3://bucket/data.json" }`	
`params { nested { input_file = "s3://bucket/data.json" } }`	不适用
`input_file = params.input_file`	不适用

Amazon ECR 中用于私有工作流程的容器镜像

在创建私有工作流程之前，您需要为工作流程创建容器映像。您将图片上传到亚马逊弹性容器注册表 (Amazon ECR) Container Registry 中的私有镜像存储库。当您运行工作流程时，该 HealthOmics 服务会访问您提供的容器。

容器映像 Amazon ECR 存储库必须与调用服务的账户位于同一 AWS 区域。只要源镜像存储库提供适当的权限，其他人 AWS 账户 就可以拥有容器镜像。有关更多信息，请参阅 Amazon 弹性容器注册表共享工作流程存储库政策。

我们建议您将您的 Amazon ECR 容器映 URIs 像定义为工作流程中的参数，以便在运行开始之前可以验证访问权限。通过更改 Region 参数，还可以更轻松地在新区域中运行工作流程。

> **Note**
>
> HealthOmics 不支持 ARM 容器，也不支持访问公共仓库。

有关为访问 Amazon ECR 配置 IAM 权限的信息，请参阅资源权限。 HealthOmics

主题

- Amazon ECR 容器镜像的一般注意事项
- HealthOmics 工作流程的环境变量

- 在 Amazon ECR 容器镜像中使用 Java
- 向 ECR 容器镜像添加任务输入

Amazon ECR 容器镜像的一般注意事项

- 架构

  HealthOmics 支持 x86_64 容器。如果您的本地计算机基于 ARM（例如 Apple Mac），请使用如下命令构建 x86_64 容器镜像：

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

- 入口点和外壳

  HealthOmics 工作流引擎将 bash 脚本作为命令替换注入到工作流任务使用的容器镜像中。因此，应在没有指定的 ENTRYPOINT 的情况下构建容器映像，以便默认使用 bash shell。

- 已安装的路径

  共享文件系统挂载到位于 /tmp 的容器任务。在此位置的容器镜像中内置的任何数据或工具都将被覆盖。

  工作流定义可通过 /mnt/workflow 上的只读挂载供任务使用。

- 映像大小

  HealthOmics 工作流程固定大小配额有关容器镜像的最大尺寸，请参阅。

HealthOmics 工作流程的环境变量

HealthOmics 提供了环境变量，这些变量包含有关容器中运行的工作流程的信息。您可以在工作流程任务的逻辑中使用这些变量的值。

所有 HealthOmics 工作流程变量都以AWS_WORKFLOW_前缀开头。此前缀是受保护的环境变量前缀。请勿在工作流程容器中为自己的变量使用此前缀。

HealthOmics 提供了以下工作流环境变量：

AWS_REGION

此变量是容器运行的区域。

AWS_WORKFLOW_RUN

此变量是当前运行的名称。

AWS_WORKFLOW_RUN_ID

此变量是当前运行的运行标识符。

AWS_WORKFLOW_RUN_UUID

此变量是当前运行的运行 UUID。

AWS_WORKFLOW_TASK

此变量是当前任务的名称。

AWS_WORKFLOW_任务_ID

此变量是当前任务的任务标识符。

AWS_WORKFLOW_TASK_UUID

此变量是当前任务的任务 UUID。

以下示例显示了每个环境变量的典型值：

```
AWS Region: us-east-1
Workflow Run: arn:aws:omics:us-east-1:123456789012:run/6470304
Workflow Run ID: 6470304
Workflow Run UUID: f4d9ed47-192e-760e-f3a8-13afedbd4937
Workflow Task: arn:aws:omics:us-east-1:123456789012:task/4192063
Workflow Task ID: 4192063
Workflow Task UUID: f0c9ed49-652c-4a38-7646-60ad835e0a2e
```

在 Amazon ECR 容器镜像中使用 Java

如果工作流任务使用 Java 应用程序（例如 GATK），请考虑容器的以下内存要求：

- Java 应用程序使用堆栈内存和堆内存。默认情况下，最大堆内存是容器中总可用内存的百分比。此默认值取决于特定的 JVM 发行版和 JVM 版本，因此请查阅 JVM 的相关文档，或者使用 Java 命令行选项（例如 `-Xmx）明确设置堆内存最大值。
- 不要将最大堆内存设置为容器内存分配的 100%，因为 JVM 堆栈也需要内存。JVM 垃圾收集器和容器中运行的任何其他操作系统进程也需要内存。

- 某些 Java 应用程序（例如 GATK）可以使用本机方法调用或其他优化，例如内存映射文件。这些技术需要在 “堆外” 执行的内存分配，这些分配不受JVM最大堆参数的控制。

  如果您知道（或怀疑）您的 Java 应用程序分配了堆外内存，请确保您的任务内存分配包括堆外内存需求。

  如果这些堆外分配导致容器内存不足，则通常不会看到 Java OutOfMemory 错误，因为 JVM 无法控制此内存。

向 ECR 容器镜像添加任务输入

将运行工作流程任务所需的所有可执行文件、库和脚本添加到用于运行该任务的 Amazon ECR 映像中。

最佳做法是避免使用任务容器镜像外部的脚本、二进制文件和库。当使用使用`bin`目录作为`nf-core`工作流包一部分的工作流时，这一点尤其重要。虽然此目录可供工作流任务使用，但它是作为只读目录安装的。应将此目录中的所需资源复制到任务镜像中，并在运行时或构建用于任务的容器映像时提供。

HealthOmics 工作流程固定大小配额有关 HealthOmics 支持的容器镜像的最大大小，请参阅。

为私有工作流程申请 Sentieon 许可证

如果您的私人工作流程使用Sentieon软件，则需要Senieon许可证。请按照以下步骤申请和设置 Sentieon 软件的许可证：

- 申请 Sentieon 许可证
  - 向 Sentieon 支持小组 (support@sentieon.com) 发送电子邮件申请软件许可证。
    - 在电子邮件中提供您的 AWS 规范用户 ID。
    - 按照以下说明查找您的 AWS 规范用户 ID。
- 更新您的 HealthOmics 服务角色以授予其访问您所在地区的 Sentieon 许可服务器代理和 Sentieon Omics 存储桶的访问权限。以下示例授予中的访问权限`us-east-1`。如果需要，请将此文本替换为您所在的地区。

  JSON

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

```
            "Effect": "Allow",
            "Action": [
                "s3:GetObjectAcl",
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::omics-ap-us-east-1/*",
                "arn:aws:s3:::sentieon-omics-license-us-east-1/*"
            ]
        }
    ]
}
```

- 生成 AWS 支持案例以获取 Sentieon 许可证服务器代理的访问权限。
  - 要创建支持案例，请导航至 support.console.aws. amazon.com。
  - 在支持案例中提供您的 AWS 账户 和区域。您的帐户已添加到许可服务器代理的许可名单中。
- 使用 Sentieon 容器和 Sentieon 许可证脚本构建您的私有工作流程。
  - 有关在私有工作流程中使用 Sentieon 工具的其他说明，请参阅中的 S en tieon-Amazon-Omics。 GitHub
- Sentieon 软件版本 202112.07 及更高版本支持许可服务器代理。 HealthOmics 要使用 202112.07 之前的 Sentieon 软件版本，请联系 Sentieon 支持人员。

中的工作流程提示 HealthOmics

创建工作流程后，我们建议您在开始第一次运行之前对该工作流程运行 linter。linter 会检测可能导致运行失败的错误。

对于 WDL，在创建工作流程时 HealthOmics 会自动运行 linter。linter 输出可在get-workflow响应的`statusMessage`字段中找到。使用以下 CLI 命令检索状态输出（使用您创建的 WDL 工作流程的工作流 ID）：

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

HealthOmics 提供了可以在创建工作流程之前在工作流程定义上运行的 linter。在要迁移到的现有管道上运行这些 linter。 HealthOmics

- WDL— 用于运行 WDL l inter 的公共 Amazon ECR 镜像。

- Nextflow— 用于运行 Nextflow 的 Linter 规则的公开 Amazon ECR 镜像。您可以从GitHub中访问此 linter 的源代码。
- CWL— 不可用

创建或更新工作流程

要创建私有工作流程，您需要：

- Workflow definition file：用WDL、Nextflow或写入的工作流程定义文件CWL。工作流定义为使用该工作流的运行指定输入和输出。它还包括工作流程的运行和运行任务规范，包括计算和内存要求。工作流程定义文件必须采用`.zip`格式。有关更多信息，请参阅中的工作流定义文件 HealthOmics。
- Parameter template file（可选）：写入的参数模板文件JSON。创建文件来定义运行参数，或者为您 HealthOmics 生成参数模板。有关更多信息，请参阅 HealthOmics 工作流程的参数模板文件。
- Amazon ECR container images：为工作流程创建容器映像并将其存储在私有 Amazon ECR 存储库中。
- Sentieon licenses（可选）：申请Sentieon许可证，以便在私有工作流程中使用该Sentieon软件。

对于大于 4 MiB（已压缩）的工作流定义文件，请在创建工作流程时选择以下选项之一：

- 上传到 Amazon 简单存储服务文件夹并指定位置。
- 上传到外部存储库（最大大小 1 GiB）并指定存储库的详细信息。

创建工作流程后，您可以通过该`UpdateWorkflow`操作更新以下工作流信息：

- 名称
- 描述
- 默认存储类型
- 默认存储容量（带工作流程 ID）
- README.md 文件

要更改工作流程中的其他信息，请创建新的工作流程或工作流程版本。

使用工作流版本控制来组织和构造您的工作流程。版本还可以帮助您管理迭代工作流程更新的引入。有关版本的更多信息，请参阅创建工作流程版本。

主题

创建私有工作流程

使用 HealthOmics 控制台、 AWS CLI 命令或其中一个创建工作流程 AWS SDKs。

> **Note**
>
> 请勿在工作流程名称中包含任何个人身份信息 (PII)。这些名称在 CloudWatch 日志中可见。

创建工作流程时，会为该工作流程 HealthOmics 分配一个通用唯一标识符 (UUID)。工作流程 UUID 是一个全局唯一标识符 (guid)，在工作流程和工作流程版本中都是唯一的。出于数据来源的目的，我们建议您使用工作流程 UUID 来唯一标识工作流程。

主题

使用控制台创建工作流程

创建工作流程的步骤

1. 打开 HealthOmics 管理控制台。
2. 选择左上角的导航窗格 ()，然后选择私有工作流程。
3. 在私有工作流程页面上，选择创建工作流程。
4. 在定义工作流程页面上，提供以下信息：

    1. 工作流程名称：此工作流程的独特名称。我们建议您设置工作流程名称，以便在 AWS HealthOmics 控制台和 CloudWatch 日志中整理运行情况。

2. 描述（可选）：此工作流程的描述。

5. 在“工作流定义”面板中，提供以下信息：

1. 工作流语言（可选）：选择工作流程的规范语言。否则， HealthOmics 根据工作流程定义确定语言。
2. 对于工作流程定义源，请选择从基于 Git 的存储库、Amazon S3 位置或本地驱动器导入定义文件夹。

a. 对于从存储库服务导入：

> **Note**
>
> HealthOmics 支持、、GitHub、GitLabBitbucket、GitHub self-managed的公有和私有存储库GitLab self-managed。

i. 选择一个连接，将您的 AWS 资源连接到外部存储库。要创建连接，请参阅Connect 连接外部代码存储库。

> **Note**
>
> 该TLV地区的客户需要在 IAD (us-east-1) 区域创建连接才能创建工作流程。

ii. 在完整存储库 ID 中，输入您的存储库 ID 作为用户名/存储库名称。确认您有权访问此存储库中的文件。

iii. 在源引用（可选）中，输入存储库源引用（分支、标签或提交 ID）。 HealthOmics 如果未指定源引用，则使用默认分支。

iv. 在排除文件模式中，输入文件模式以排除特定的文件夹、文件或扩展名。这有助于在导入存储库文件时管理数据大小。最多有 50 个模式，并且模式必须遵循全局模式语法。例如：

A. `tests/`

B. `*.jpeg`

C. `large_data.zip`

b. 对于从 S3 中选择定义文件夹：

i. 输入包含压缩工作流程定义文件夹的 Amazon S3 位置。Amazon S3 存储桶必须与工作流程位于同一区域。

ii. 如果您的账户不拥有 Amazon S3 存储桶，请在 S3 存储桶拥有者的 AWS 账户 ID 中输入存储桶拥有者的账户 ID。为了验证存储桶所有权 HealthOmics ，必须提供此信息。

c. 对于从本地来源选择定义文件夹：

i. 输入压缩的工作流程定义文件夹的本地驱动器位置。

3. 工作流定义文件主路径（可选）：输入从压缩的工作流定义文件夹或存储库到该`main`文件的文件路径。如果工作流定义文件夹中只有一个文件，或者主文件名为 “main”，则不需要此参数。

6. 在自述文件（可选）面板中，提供以下信息：

1. 选择自述文件来源。

a. 对于从存储库服务导入，在自述文件路径中，输入存储库中自述文件的路径。

b. 对于从 S3 中选择文件，在 S3 的自述文件中，输入自述文件的 Amazon S3 URI。

c. 对于 “从本地来源选择文件：在来自本地源的自述文件中”，从本地来源上传自述文件。

2. 在自述文件路径中，输入源文件中自述文件的路径。

7. 在默认运行存储配置面板中，为使用此工作流程的运行提供默认的运行存储类型和容量：

1. 运行存储类型：选择使用静态存储还是动态存储作为临时运行存储的默认值。默认为静态存储。

2. 运行存储容量（可选）：对于静态运行存储类型，您可以输入此工作流程所需的默认运行存储量。此参数的默认值为 1200 GiB。开始运行时，您可以覆盖这些默认值。

8. 标签（可选）：您最多可以将 50 个标签与该工作流程相关联。

9. 选择下一步。

10. 在添加工作流程参数（可选）页面上，选择参数来源：

1. 对于从工作流定义文件解析， HealthOmics 将自动解析工作流定义文件中的工作流参数。

2. 对于从 Git 存储库提供参数模板，请使用仓库中参数模板文件的路径。

3. 对于从本地源选择 JSON JSON 文件，请从本地源上传指定参数的文件。

4. 对于手动输入工作流参数，请手动输入参数名称和描述。

11. 在参数预览面板中，您可以查看或更改此工作流程版本的参数。如果恢复该JSON文件，则您所做的任何本地更改都将丢失。

12. 选择下一步。

13. 查看工作流程配置，然后选择创建工作流程。

使用 CLI 创建工作流程

定义工作流程和参数后，可以使用 CLI 创建工作流程，如下所示。

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

如果您的工作流程定义文件位于 Amazon S3 文件夹中，请改用`definition-uri`参数输入该位置`definition-zip`。有关更多信息，请参阅 AWS HealthOmics API 参考CreateWorkflow中的。

`create-workflow`请求的响应如下：

```
{
  "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"
}
```

创建工作流程时要使用的可选参数

创建工作流程时，您可以指定任何可选参数。有关更多信息，请参阅 AWS HealthOmics API 参考CreateWorkflow中的。

如果要包括多个工作流定义文件，请使用`main`参数指定哪个文件是工作流程的主定义文件。

如果您已将工作流程定义文件上传到 Amazon S3 文件夹，请使用`definition-uri`参数指定位置，如以下示例所示。如果您的账户不拥有 Amazon S3 存储桶，请提供所有者的 AWS 账户 ID。

```
aws omics create-workflow  \
  --name Test  \
  --main multi_workflow/workflow2.wdl  \
  --definition-uri s3://omics-bucket/workflow-definition/  \
  --owner-id  123456789012  \
  --parameter-template file://params_sample_description.json
```

您可以指定默认的运行存储类型（动态或静态）和运行存储容量（静态存储所必需的）。有关运行存储类型的更多信息，请参阅在 HealthOmics 工作流程中运行存储类型。

```
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  \
```

使用加速器参数创建在加速计算实例上运行的工作流程。以下示例说明如何使用该--accelerators参数。

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

使用 SDK 创建工作流程

您可以使用其中一个来创建工作流程 SDKs。以下示例说明如何使用 Python 开发工具包创建工作流程

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

更新私有工作流程

您可以使用 HealthOmics 控制台、 AWS CLI 命令或其中一个来更新工作流程 AWS SDKs。

> **Note**
>
> 请勿在工作流程名称中包含任何个人身份信息 (PII)。这些名称在 CloudWatch 日志中可见。

主题

使用控制台更新工作流程

更新工作流程的步骤

1. 打开 HealthOmics 管理控制台。
2. 选择左上角的导航窗格 ()，然后选择私有工作流程。
3. 在私有工作流程页面上，选择要更新的工作流程。
4. 在 “工作流程” 页面上：
   - 如果工作流程有版本，请确保选择默认版本。
   - 从 “操作” 列表中选择 “编辑”。
5. 在 “编辑工作流程” 页面上，您可以更改以下任何值：
   - 工作流程名称。
   - 工作流程描述。
   - 工作流程的默认 “运行” 存储类型。
   - 默认运行存储容量（如果运行存储类型为静态存储）。有关默认运行存储配置的更多信息，请参阅使用控制台创建工作流程。
6. 选择 “保存更改” 以应用更改。

使用 CLI 更新工作流程

如以下示例所示，您可以更新工作流程名称和描述。您也可以更改默认的运行存储类型（静态或动态）和运行存储容量（对于静态存储类型）。有关运行存储类型的更多信息，请参阅在 HealthOmics 工作流程中运行存储类型。

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

```
  --storage-capacity 1200
```

您没有收到对update-workflow请求的回复。

使用 SDK 更新工作流程

您可以使用其中一个来更新工作流程 SDKs。

以下示例说明如何使用 Python 开发工具包更新工作流程

```
import boto3

omics = boto3.client('omics')

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

删除私有工作流程

当您不再需要某个工作流程时，可以使用 HealthOmics 控制台、 AWS CLI 命令或其中一个将其删除 AWS SDKs。您可以删除符合以下条件的工作流程：

- 其状态为 “活动” 或 “失败”。
- 它没有活跃股份。
- 您已经删除了所有工作流程版本。

删除工作流程不会影响正在使用该工作流程的任何正在进行的运行。

主题

- 使用控制台删除工作流程
- 使用 CLI 删除工作流程
- 使用 SDK 删除工作流程

使用控制台删除工作流程

删除工作流

1. 打开 HealthOmics 管理控制台。
2. 在左侧导航窗格中，选择私有工作流程。
3. 在私有工作流程页面上，选择要删除的工作流程。
4. 在 “工作流程” 页面上，从 “操作” 列表中选择 “删除选定内容”。
5. 在删除工作流程模式中，输入 “确认” 以确认删除。
6. 选择删除。

使用 CLI 删除工作流程

以下示例说明如何使用 AWS CLI 命令删除工作流程。要运行此示例，请将*workflow id*替换为您要删除的工作流程的 ID。

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

HealthOmics 不会发送对`delete-workflow`请求的响应。

使用 SDK 删除工作流程

您可以使用其中一个来删除工作流程 SDKs。

以下示例说明如何使用 Python 开发工具包删除工作流程。

```
import boto3

omics = boto3.client('omics')

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

验证工作流程状态

创建工作流程后，您可以使用 get-workflow 验证工作流程的状态并查看其他详细信息，如图所示。

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

响应包括工作流程的详细信息，包括状态，如图所示。

```
{
    "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"
  }
```

状态转换到后，您可以使用此工作流程开始运行ACTIVE。

从工作流程定义中引用基因组文件

可以使用如下所示的 URI 来引用 HealthOmics 参考存储对象。使用您自己的*account ID*reference store ID*、和（*reference ID*如果有指示）。

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

有些工作流程需要同时使用SOURCE和INDEX文件作为参考基因组。之前的 URI 是默认的简写形式，默认为源文件。要指定任一文件，您可以使用长 URI 格式，如下所示。

```
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
```

如图所示，使用序列读取集将具有类似的模式。

```
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
```

某些读取集（例如基于 FASTQ 的读取集）可能包含配对读取。在以下示例中，它们被称为 SOURCE1 和 SOURCE2。诸如 BAM 和 CRAM 之类的格式只能有一个文件。 SOURCE1 某些读取集将包含索引文件，例如bai或crai文件。前面的 URI 是默认的简写形式，默认为该 SOURCE1 文件。要指定确切的文件或索引，可以使用长 URI 格式，如下所示。

```
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
```

以下是使用两个 Omics 存储空间 URIs的输入 JSON 文件的示例。

```
{
   "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>"
}
```

AWS CLI 通过添加到`--inputs file://<input_file.json>`您的开始运行请求中引用输入 JSON 文件。

使用自述文件

上传一个 README.md 文件，其中包含工作流程的说明、图表和基本信息。每个工作流程版本都支持一个 README 文件，可以随时更新。

自述文件要求包括：

- 自述文件必须采用 markdown (.md) 格式
- 最大文件大小：500 KiB

主题

使用现有的自述文件

READMEs 从 Git 存储库导出的内容包含相对链接，这些链接通常在仓库之外不起作用。 HealthOmics Git 集成会自动将这些链接转换为绝对链接，以便在控制台中正确呈现，无需手动更新 URL。

对于从 Amazon S3 或本地驱动器 READMEs 导入的图像和链接，必须使用公共文件 URLs 或更新其相对路径才能正确呈现。

> **Note**
>
> 图片必须公开托管才能显示在 HealthOmics 控制台中。存储在GitHub Enterprise Server或存储GitLab Self-Managed库中的图像无法呈现。

渲染条件

HealthOmics 控制台使用绝对路径插值可公开访问的图像和链接。要 URLs 从私有仓库进行渲染，用户必须有权访问存储库。对于GitHub Enterprise Server或使用自定义域名的GitLab Self-Managed存储库， HealthOmics 无法解析相对链接，也无法呈现存储在这些私有存储库中的图像。

下表显示了 AWS 控制台自述文件视图支持的 markdown 元素。

元素	AWS 控制台
警报	是的，但没有图标
徽章	是
基本文本格式	是
代码块	是的，但没有语法突出显示和复制按钮功能
可折叠部分	是
标题	是

元素	AWS 控制台
图像格式	是
图片（可点击）	是
换行符	是
美人鱼图	只能打开图表、移动图表位置和复制代码
报价	是
下标和上标	是
表	是，但不支持文本对齐
文本对齐方式	是

工作流程版本控制在 HealthOmics

如果您需要对工作流程进行更改，可以创建新的工作流程或新的工作流程版本。版本是不可变的，但允许的配置更改除外，这些更改不会影响执行逻辑。

工作流程版本具有以下优点：

- 版本构成了相关工作流程的逻辑组。您可以为每个工作流程版本添加用户定义的名称，以便更轻松地对其进行管理（特别是对于具有大量版本的工作流程）。
- 您可以同时运行工作流程的多个版本。
- 工作流程的所有版本共享相同的工作流程 ID 和基本 ARN，这可以简化您修改工作流程后的管道管理。
- 工作流程版本提供的数据来源级别与工作流程相同。版本是不可变的，并且 HealthOmics会为每个工作流程版本创建唯一的 ARN。版本 ARN 包括工作流程 ID 和版本名称，如以下示例所示：

  ```
  arn:aws:omics:us-west-2:123456789012:workflow/1234567/version/
  myUniqueVersionName
  ```
- 如果您拥有共享的工作流程，则可以在不中断订阅者的情况下更新工作流程（订阅者可以继续使用以前的版本）。订阅者可以访问所有工作流程版本。如果您创建了新版本，则无需重新共享工作流程。
- 启动工作流程运行时，可以指定工作流程版本。

- 用户可以选择保持稳定版本进行生产运行，也可以试用最新版本进行试运行。
- 如果用户在新版本中遇到问题，他们可以恢复到工作流程的先前版本。
- 共享工作流程的订阅者可以选择要使用的版本。

主题

默认工作流程版本

创建工作流程的一个或多个版本后， HealthOmics 会将原始工作流程视为默认版本。开始运行时，您可以选择为运行指定工作流程版本。如果您在开始运行时未指定版本，则 HealthOmics 使用默认版本。

在控制台中，使用默认版本标签 HealthOmics 表示原始工作流程。只有在您创建了一个或多个工作流程版本之后，控制台才会使用此标签。原始工作流程始终保持默认版本。您不能将任何其他版本指定为默认版本。

如果有其他版本与工作流程相关联，则无法删除该工作流程的默认版本。有关更多信息，请参阅 删除私有工作流程。

创建工作流程版本

创建工作流程的新版本时，需要为新版本指定配置值。它不会从工作流程继承任何配置值。

创建版本时，请提供此工作流程中唯一的版本名称。 HealthOmics 创建版本后，您无法更改名称。

版本名称必须以字母或数字开头，可以包括大写和小写字母、数字、连字符、句点和下划线。最大长度为 64 个字符。例如，您可以使用简单的命名方案，例如版本 1、版本 2、版本 3。您还可以将工作流程版本与自己的内部版本控制约定（例如 2.7.0、2.7.1、2.7.2）进行匹配。

或者，使用版本描述字段添加有关此版本的注释。例如：Fix for syntax error in workflow definition。

> **Note**
>
> 不要在版本名称中包含任何个人身份信息 (PII)。版本名称显示在工作流程版本 ARN 中。

HealthOmics 为工作流程版本分配唯一的 ARN。根据工作流程 ID 和版本名称的组合，ARN 是唯一的。

> **Warning**
>
> 删除工作流程版本后， HealthOmics 允许您为其他工作流程版本重复使用该版本的版本名称。最佳做法是不要重复使用版本名称。如果您确实重复使用了名称，则工作流程和每个版本都有一个唯一的 UUID，您可以将其用作来源。

主题

使用控制台创建工作流程版本

创建工作流程的步骤

1. 打开 HealthOmics 管理控制台。
2. 选择左上角的导航窗格 ()，然后选择私有工作流程。
3. 在私有工作流程页面上，选择新版本的工作流程。
4. 在工作流程详细信息页面上，选择创建新版本。
5. 在创建版本页面上，提供以下信息：
    1. 版本名称：输入工作流程版本的名称，该名称在整个工作流程中是唯一的。
    2. 版本描述（可选）：您可以使用描述字段添加有关此版本的注释。
6. 在“工作流定义”面板中，提供以下信息：

1. 工作流语言（可选）：选择工作流程版本的规范语言。否则， HealthOmics 根据工作流程定义确定语言。
2. 对于工作流程定义源，请选择从基于 Git 的存储库、Amazon S3 位置或本地驱动器导入定义文件夹。

   a. 对于从存储库服务导入：

   > **Note**
   >
   > HealthOmics 支持、、GitHub、GitLabBitbucket、GitHub self-managed的公有和私有存储库GitLab self-managed。

      i. 选择一个连接，将您的 AWS 资源连接到外部存储库。要创建连接，请参阅Connect 连接外部代码存储库。

      > **Note**
      >
      > 该TLV地区的客户需要在 IAD (us-east-1) 区域创建连接才能创建工作流程。

      ii. 在完整存储库 ID 中，输入您的存储库 ID 作为用户名/存储库名称。确认您有权访问此存储库中的文件。

      iii. 在源引用（可选）中，输入存储库源引用（分支、标签或提交 ID）。 HealthOmics 如果未指定源引用，则使用默认分支。

      iv. 在排除文件模式中，输入文件模式以排除特定的文件夹、文件或扩展名。这有助于在导入存储库文件时管理数据大小。最多有 50 个模式，并且模式必须遵循全局模式语法。例如：

      A. `tests/`

      B. `*.jpeg`

      C. `large_data.zip`

   b. 对于从 S3 中选择定义文件夹：

      i. 输入包含压缩工作流程定义文件夹的 Amazon S3 位置。Amazon S3 存储桶必须与工作流程位于同一区域。

      ii. 如果您的账户不拥有 Amazon S3 存储桶，请在 S3 存储桶拥有者的 AWS 账户 ID 中输入存储桶拥有者的账户 ID。为了验证存储桶所有权 HealthOmics ，必须提供此信息。

c. 对于从本地来源选择定义文件夹：

i. 输入压缩的工作流程定义文件夹的本地驱动器位置。

3. 工作流定义文件主路径（可选）：输入从压缩的工作流定义文件夹或存储库到该`main`文件的文件路径。如果工作流定义文件夹中只有一个文件，或者主文件名为 “main”，则不需要此参数。

7. 在默认运行存储配置面板中，为使用此工作流程的运行提供默认的运行存储类型和容量：

1. 运行存储类型：选择使用静态存储还是动态存储作为临时运行存储的默认存储空间。默认为静态存储。
2. 运行存储容量（可选）：对于静态运行存储类型，您可以输入此工作流程所需的默认运行存储量。此参数的默认值为 1200 GiB。开始运行时，您可以覆盖这些默认值。

8. 标签（可选）：您最多可以将 50 个标签与该工作流程版本相关联。
9. 选择下一步。
10. 在添加工作流程参数（可选）页面上，选择参数来源：

1. 对于从工作流定义文件解析， HealthOmics 将自动解析工作流定义文件中的工作流参数。
2. 对于从 Git 存储库提供参数模板，请使用仓库中参数模板文件的路径。
3. 对于从本地源选择 JSON JSON 文件，请从本地源上传指定参数的文件。
4. 对于手动输入工作流参数，请手动输入参数名称和描述。

11. 在参数预览面板中，您可以查看或更改此工作流程版本的参数。如果恢复该JSON文件，则您所做的任何本地更改都将丢失。
12. 选择下一步。
13. 查看版本配置，然后选择创建版本。

创建版本后，控制台会返回到工作流程详细信息页面，并在工作流程和版本表中显示新版本。

使用 CLI 创建工作流程版本

您可以使用 `CreateWorkflowVersion` API 操作创建工作流程版本。对于可选参数， HealthOmics 使用以下默认值：

参数	默认值
Engine	根据工作流程定义确定
存储类型	STATIC

参数	默认值
存储容量（用于静态存储）	1200 GiB
Main	根据工作流定义文件夹的内容确定。有关更多信息，请参阅 HealthOmics 工作流程定义要求。
加速器	none
标签	none

以下 CLI 示例创建了使用静态存储作为默认运行存储的工作流程版本：

```
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
```

如果您的工作流程定义文件位于 Amazon S3 文件夹中，请改用`definition-uri`参数输入该位置`definition-zip`。有关更多信息，请参阅 AWS HealthOmics API 参考CreateWorkflowVersion中的。

您会收到以下对`create-workflow-version`请求的回复。

```
{
  "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"
}
```

使用 SDK 创建工作流程版本

您可以使用其中一个来创建工作流程 SDKs。

以下示例说明如何使用 Python SDK 创建工作流程版本

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

验证工作流程版本的状态

创建工作流程版本后，您可以使用验证状态并查看工作流程的其他详细信息 get-workflow-version，如图所示。

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

如图所示，响应会为您提供工作流程的详细信息，包括状态。

```
{
  "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"
```

```
}
```

必须先将状态转换为，然后才能使用此工作流程版本开始运行ACTIVE。

更新工作流程版本

您可以更新私有工作流程版本的描述和默认运行存储配置。要更改工作流程版本中的任何其他信息，请创建一个新版本。

主题

- 使用控制台更新工作流程版本
- 使用 CLI 更新工作流程版本
- 使用 SDK 更新工作流程版本

使用控制台更新工作流程版本

更新工作流

1. 打开 HealthOmics 管理控制台。
2. 在左侧导航窗格中，选择私有工作流程。
3. 在私有工作流程页面上，选择工作流程。
4. 在 “工作流程” 页面上，选择要更新的工作流程版本，然后从 “操作” 列表中选择 “编辑”。
   - 如果您选择默认版本，则控制台将打开 “编辑工作流程” 页面。有关更多信息，请参阅 更新私有工作流程。
   - 如果您选择用户定义的版本，则控制台将打开 “编辑版本” 页面。
5. 在编辑版本页面上，提供以下信息
   - 版本描述（可选）-此版本的描述。
6. 在默认运行存储配置面板中，为使用此工作流程版本的运行提供以下默认值。当你开始运行时，你可以覆盖默认值：
   - 对于 “运行存储类型”，选择 “静态” 或 “动态”。
   - 对于静态运行存储，请为使用此工作流程版本的运行选择默认的运行存储容量。此参数的默认值为 1200 GiB。
7. 选择保存更改。

控制台返回工作流程详细信息页面，并显示带有更新工作流程版本的页面横幅。

使用 CLI 更新工作流程版本

您可以使用以下 CLI 命令更新工作流程版本的参数。工作流程 ID 和版本名称的组合可唯一标识版本。

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

您未收到对update-workflow-version请求的回应。

使用 SDK 更新工作流程版本

您可以使用其中一个来更新工作流程版本 SDKs。以下 python SDK 示例展示了如何更新工作流程版本的存储类型和描述。

```
import boto3

omics = boto3.client('omics')

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

删除工作流程版本

您可以使用控制台、CLI 或其中一个来删除用户定义的工作流程版本 SDKs。删除工作流程版本不会影响正在使用该工作流程版本的任何正在进行的运行。

您无法删除默认工作流程版本。删除所有用户定义的版本，然后删除工作流程。

主题

- 使用控制台删除工作流程版本
- 使用 CLI 删除工作流程版本

- 使用 SDK 删除工作流程版本

使用控制台删除工作流程版本

删除工作流程版本

1. 打开 HealthOmics 管理控制台。
2. 在左侧导航窗格中，选择私有工作流程。
3. 在私有工作流程页面上，选择工作流程。
4. 在 “工作流程” 页面上，选择要删除的工作流程版本，然后从 “操作” 列表中选择 “删除”。
5. 在 “删除工作流程版本” 模式中，输入 “确认” 以确认删除。
6. 选择删除。

控制台会显示带有已删除工作流程版本的页面横幅。

使用 CLI 删除工作流程版本

您可以使用以下 CLI 命令删除用户定义的工作流程版本。工作流程 ID 和版本名称的组合可唯一标识版本。

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

您未收到对`delete-workflow-version`请求的回应。

使用 SDK 删除工作流程版本

您可以使用其中一个来删除工作流程 SDKs。

以下示例说明如何使用 Python 开发工具包删除工作流程。

```
import boto3

omics = boto3.client('omics')

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

```
)
```

开始 HealthOmics 跑步

创建工作流程后，您可以使用该工作流程开始运行。

开始运行时， HealthOmics 会分配临时运行存储空间供工作流引擎在运行期间使用。为确保数据隔离和安全 HealthOmics ，请在每次运行开始时配置存储，并在运行结束时取消配置。

HealthOmics 提供了多个与工作流程运行和任务相关的配额。默认值本质上是保守的，可帮助您避免意外的成本超支。您可以请求提高这些限额。有关更多信息，请参阅 HealthOmics 服务配额。

当你开始运行时，会为该运行 HealthOmics 分配一个运行 ID 和一个运行 uuid。账户中的跑步具有唯一的跑步次数 IDs。但是， HealthOmics 重复使用已删除的运行 IDs，因此运行和已删除的运行可以具有相同的运行 ID。此外，共享工作流程的运行 ID 与您账户中的运行具有相同的运行ID的情况很少见，但也可能如此。

run uuid是一个全局唯一标识符 (guid)，可用于识别跨账户的运行或区分账户中具有相同运行 ID 的两次运行。

> **Note**
>
> 出于数据来源的目的，我们建议您使用run uuid来唯一标识运行。run uuid也是链接到内部实验室信息管理系统 (LIMs) 或样品追踪系统的最佳标识符。

主题

在 HealthOmics 工作流程中运行存储类型

开始运行时， HealthOmics 会分配临时运行存储空间供工作流引擎在运行期间使用。 HealthOmics以文件系统的形式提供临时运行存储。

对于给定的工作流程或工作流程运行，您可以选择动态或静态运行存储。默认情况下， HealthOmics 提供静态运行存储。

> **Note**
>
> 运行存储空间使用量会对您的账户产生费用。有关静态和动态运行存储的定价信息，请参阅 HealthOmics定价。

以下各节提供了在决定使用哪种运行存储类型时需要考虑的信息。

动态运行存储

我们建议在大多数运行中使用动态运行存储，包括需要更快启动时间的运行、事先不知道存储需求的运行以及迭代开发测试周期。

您无需估计运行所需的存储空间或吞吐量。 HealthOmics 根据运行期间的文件系统利用率，动态地向上或向下扩展存储大小。 HealthOmics 还可以根据工作流程的需求动态扩展吞吐量。由于文件系统存储空间不足错误，运行永远不会失败。

动态运行存储提供的 provisioning/deprovisioning 时间比静态运行存储更快。对于大多数工作流程来说，更快的设置都是一个优势，在 development/test 周期中也是一个优势。

运行完成（成功路径或失败路径）后，GetRun API 操作会在 StorageCapacity 字段中返回运行使用的最大存储空间。您也可以在日志组中的运行清单omics日志中找到此信息。对于在 2 小时内完成的动态存储运行，最大存储值可能不可用。

对于动态运行存储，运行会提供使用 NFS 协议的文件系统。NFS 将 “创建”、“删除” 和 “重命名” 文件操作视为非等性，这有时可能会导致您的代码需要优雅处理的这些操作出现竞争条件。例如，如果您的代码尝试删除不存在的文件，则不应失败。在采用动态运行存储之前，我们建议您调整工作流程代码，使其能够适应非等性文件操作。请参阅用于安全处理非等性运算的代码示例。

用于安全处理非等性运算的代码示例

以下 python 示例显示了如果文件不存在，如何删除该文件而不会失败。

```
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")
```

以下示例使用 Bash 外壳。要安全地删除文件（即使该文件不存在），请使用：

```
rm -f my_file
```

要安全地移动（重命名）文件，请仅在该文件old_name存在于当前目录中时才运行移动命令。

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

要创建目录，请使用以下命令：

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

静态运行存储

对于静态运行存储，运行会提供使用 Lustre 协议的文件系统。默认情况下，此协议可适应非等效文件操作。您无需调整工作流程代码即可处理非等性文件操作。

HealthOmics 分配固定数量的运行存储空间。您可以在开始运行时指定此值。如果您未指定值，则默认运行存储空间为 1200 GiB。当您在 StartRun API 请求中为存储大小指定值时，系统会将该值四舍五入到最接近的 1200 GiB 的倍数。如果该存储大小不可用，则四舍五入到最接近的 2400 GiB 的倍数。

对于静态运行存储， HealthOmics 请配置以下吞吐量值：

- 预配置的 MB/s 每 TiB 存储容量的基准吞吐量为 200。
- 配置的 MB/s 每 TiB 存储容量最高可达 1300 的突发吞吐量。

如果指定的存储大小太小，则运行将失败，并显示文件系统存储空间不足错误。静态运行存储非常适合具有已知存储要求的可预测工作流程。

静态运行存储适用于具有高任务并发性的大型、突发性工作负载（例如，并行处理大量 RNASeq 样本）。与动态运行存储相比，它可提供更高的每 GiB 文件系统吞吐量和更低的每 GiB 成本。

计算所需的静态运行存储空间

工作流程在使用静态运行存储（与动态运行存储相比）时需要额外的容量，因为基础文件系统安装使用静态文件系统容量的 7%。

如果您运行动态运行存储工作流程来测量运行使用的最大存储空间，请使用以下计算来确定所需的最小静态存储量：

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

例如：

```
    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.
```

因此，1200GiB（静态运行存储的最小容量）足以满足此次运行的需求。

运行时 HealthOmics 运行保留模式

运行完成后，将运行元数据 HealthOmics 存档到 CloudWatch。默认情况下，除非您更改 CloudWatch 保留策略，否则会无限期 CloudWatch 保留运行数据。运行输出也会存储在 Amazon S3 中，直到您将其删除。

其中一个可调整项HealthOmics 服务配额是maximum number of runs (active and inactive)区域内。HealthOmics 保留运行元数据以供控制台和 API 操作使用（ListRuns 和 GetRun），最多可运行此次数。开始运行时，可以设置运行保留模式参数以指示运行的保留行为。该参数支持 “移除” 和 “保留” 值。

对于保留模式设置为 REMOVE 的新运行，如果在保存了最大运行次数后 HealthOmics 尝试添加该运行，它会自动删除设置了 REMOVE 模式的最早运行的元数据。此移除不会影响存储在 CloudWatch 或 Amazon S3 中的数据。

RETAIN 是运行保留模式的默认值。对于在此模式下运行，系统不会删除运行元数据。如果 HealthOmics 达到最大运行次数（全部设置为 RETAIN），则在删除某些试验之前，您将无法创建其他游程。

如果您计划同时运行超过最大运行次数的批次，请务必将运行保留模式设置为 REMOVE。否则，当 HealthOmics 尝试在最大值之后开始下一次运行时，批处理将失败。

使用 “移除” 保留模式的其他注意事项：

- 当你第一次开始使用 REMOVE 作为保留模式时，可以考虑删除一个或多个使用 RETAIN 模式的运行，以腾出插槽。当你开始额外的 REMOVE 运行时，自动删除会占据主导地位，因此有足够的插槽可供新的运行使用。
- 如果要重新运行已存档的运行（或一组运行），请使用 HealthOmics 重新运行 CLI 工具。有关如何使用此工具的更多信息和示例，请参阅工具存储库中的 Omics 重新运行。 HealthOmics GitHub
- 我们建议您为每次运行配置一个唯一的名称。 HealthOmics 删除运行后，您将无法使用控制台或 API 来查找运行名称或运行 ID。但是，您可以使用 CloudWatch 搜索运行名称，因此请使用唯一名称来获得最佳搜索结果。
- 您可以使用 CloudWatch start-query命令来获取有关已存档运行的信息。如果运行名称不是唯一的，则查询可能会返回多个清单。开始时间和结束时间参数定义搜索的时间范围。

```
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>
```

该start-query命令返回查询 ID。将查询 ID 传递给get-query-results命令会返回查询结果。

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

HealthOmics 运行输入

如果工作流定义为工作流或工作流任务指定了输入文件， HealthOmics 则将这些文件暂存到专用于工作流程运行的临时卷中。这些输入文件是只读的，这可以防止任务修改工作流中其他任务的潜在输入。对于目录导入，这些目录也是只读的。

许多基因组学应用程序假设索引文件与序列文件（例如文件的配套bai文件）位于同一位置。bam要包括索引文件，请在工作流定义中将其指定为任务输入。

主题

管理运行参数大小

开始运行时，您可以在运行参数 JSON 对象或文件中指定运行输入。您最多可以为工作流程指定 50 KB 的运行参数。您可以使用以下方法来保持在此大小限制范围内：

- 使用目录导入

  要指定大量输入文件，请指定一个参数作为包含所有文件的 Amazon S3 位置，而不是为每个文件位置指定一个参数。有关更多信息，请参阅下一个主题（Amazon S3 输入参数格式）。

- 使用样本表

  样本表是一个 CSV 或 TSV 文件，其中一列用于 fastq.gz 地址（或两列用于配对读取），另外一列用于元数据（例如样本名称）。您可以将样本表指定为运行输入参数，而不是每个输入文件的参数。

  您的工作流程定义了样本工作表如何映射到工作流程中的数据结构。虽然你可以用 WDL 和 CWL 为样本表编写代码，但它们更常见。 NextFlow有关示例，请参阅 nf-core GitHub 网站上的样本表。

亚马逊 S3 输入参数格式

对于接受 Amazon S3 位置的输入参数，该参数可以指定一个文件或整个文件目录的位置。使用目录具有以下优点：

- 方便-您可以将目录名指定为参数。您不会列出每个文件名。
- 紧凑性-输入参数最大文件大小为 50 KB。如果您提供的输入文件名列表很长，则可以超过此最大值。

Amazon S3 是一个扁平的对象存储系统，因此它不支持目录。通过为每个文件指定相同的对象 key 前缀，可以将文件分组到一个 “目录” 中。有关 Amazon S3 对象密钥前缀的更多信息，请参阅使用前缀组织对象。

HealthOmics 按如下方式解释输入参数值：

- 如果 Amazon S3 位置没有以正斜杠结尾或使用全局模式，则 HealthOmics 期望参数值成为一个 Amazon S3 对象的键。

  例如，你指定`s3://myfiles/runs/inputs/a/file1.fastq`要输入 file1.fastq

- 如果 Amazon S3 位置以正斜杠结尾，则会将参数值 HealthOmics 解释为 Amazon S3 前缀。它会加载所有带有该前缀的 Amazon S3 对象。

  例如，您可以指定加载所有键`s3://myfiles/runs/inputs/a/`以此前缀开头的对象。

- 对于 Nextflow，在输入参数中 HealthOmics 支持 Amazon S3 URIs 的全局模式。

  例如，您可以指定输入所有密钥`"s3://myfiles/runs/inputs/a/*.gz"`以此前缀开头的.gz 文件。

Amazon S3 输入中双斜杠的特定语言处理

HealthOmics 在 Amazon S3 中处理双斜杠时，保留每个工作流程引擎的原生引擎行为 URIs，这样当您将工作流程迁移到时，您无需对其进行任何更改。 HealthOmics以下各节描述了每个引擎如何处理各种场景。

WDL

如果输入参数在 URI 的中间或末尾包含双斜杠，则 WDL 引擎会保留该双斜杠。

输入参数	预期地点	
s3://myfiles/runs/inputs//file1.fastq	s3://myfiles/runs/inputs//file1.fastq	
s3:///myfiles/runs/inputs/	s3:///myfiles/runs/inputs/	

下一步

如果输入参数在 URI 中间包含双斜杠，则 Nextflow 引擎将保留双斜杠。对于 URI 末尾的双斜杠，Nextflow 引擎会将其解析为单个斜杠。

输入参数	预期地点	
s3://myfiles/runs/inputs//file1.fastq	s3://myfiles/runs/inputs//file1.fastq	
s3://myfiles//runs/inputs//*.gz	s3://myfiles//runs/inputs//*.gz	
s3://myfiles//runs/inputs//	s3://myfiles//runs/inputs/	

CWL

如果输入参数在 URI 的中间或末尾包含双斜杠，则 CWL 引擎会保留该双斜杠。

输入参数	预期地点	
s3://myfiles// runs/inputs//file1.fastq	s3://myfiles// runs/inputs//file1.fastq	
s3://myfiles//runs/inputs//	s3://myfiles//runs/inputs//	

亚马逊 S3 输入存档状态

HealthOmics 可以实时检索 S3 交付的 Amazon S3 对象。对于处于以下存档存储状态的对象restore，要使其可用的对象 HealthOmics：

- Amazon S3 Glacier 中的灵活检索或深度存档存储类别。
- 智能分层中的存档访问层或深度存档访问层。

有关恢复对象的信息，请参阅 Amazon S3 用户指南中的恢复存档对象。

开始磨合 HealthOmics

开始运行时，可以设置运行存储类型和存储量（用于静态存储）。有关更多信息，请参阅 在 HealthOmics 工作流程中运行存储类型。

您还可以设置运行优先级。优先级对运行的影响取决于运行是否与运行组关联。有关更多信息，请参阅运行优先级。

如果您创建了一个或多个工作流程版本，则可以在开始运行时指定版本。如果您未指定版本，则 HealthOmics 启动默认工作流程版本。

为输出文件指定 Amazon S3 的位置。如果您同时运行大量工作流程，请 URIs 为每个工作流程使用单独的 Amazon S3 输出以避免存储桶限制。有关更多信息，请参阅 Amazon S3 用户指南中的使用前缀组织对象和优化 Amazon S 3 性能白皮书中的水平扩展存储连接。

> **Note**
>
> 您可以在开始运行时指定 IAM 服务角色。或者，控制台可以为您创建服务角色。有关更多信息，请参阅 的服务角色 AWS HealthOmics。

主题

HealthOmics 运行参数

开始运行时，可以在运行参数 JSON 文件中指定运行输入，也可以内联输入参数值。有关管理运行参数 JSON 文件大小的信息，请参阅管理运行参数大小。

HealthOmics 支持以下 JSON 类型的参数值。

JSON 类型	键和值示例	备注
布尔值	“b”: true	值不在引号中，且全部为小写。
整数	“i”: 7	值不在引号中。
数字	“f”: 42.3	值不在引号中。

JSON 类型	键和值示例	备注
字符串	“s”: “字符”	值用引号表示。对文本值使用字符串类型和 URIs。URI 目标必须是预期的输入类型。
array	“a”: [1,2,3]	值不在引号中。每个数组成员都必须具有由输入参数定义的类型。
object	“o”: {“左”: “a”，“右”: 1}	在 WDL 中，对象映射到 WDL 配对、映射或结构

使用控制台开始运行

要启动工作流程，请运行

1. 打开 HealthOmics 管理控制台。
2. 在左侧导航窗格中，选择 R uns。
3. 在 “运行” 页面上，选择 “开始运行”。
4. 在运行详细信息面板中，提供以下信息

   - 工作流来源-选择 “自有的工作流程” 或 “共享工作流程”。
   - 工作流程 ID-与此运行关联的工作流程 ID。
   - 工作流程版本（可选）-选择用于此次运行的工作流程版本。如果您未选择版本，则运行将使用工作流程的默认版本。
   - 跑步名称-本次跑步的独特名称。
   - 运行优先级（可选）-此次运行的优先级。数字越大，优先级越高，优先级最高的任务首先运行。
   - 运行存储类型-在此处指定存储类型以覆盖为工作流程指定的默认运行存储类型。静态存储为运行分配固定数量的存储空间。动态存储可根据运行中的每项任务的需要向上和向下扩展。
   - 运行存储容量-对于静态运行存储，请指定运行所需的存储量。此条目将覆盖为工作流程指定的默认运行存储量。
   - 选择 S3 输出目的地-保存运行输出的 S3 位置。

- 输出存储桶所有者的账户 ID（可选）-如果您的账户不拥有输出存储桶，请输入存储桶所有者的 AWS 账户 ID。此信息是必需的，这样 HealthOmics 才能验证存储桶的所有权。
- 运行元数据保留模式-选择是保留所有运行的元数据，还是让系统在账户达到最大运行次数时删除最旧的运行元数据。有关更多信息，请参阅 运行时 HealthOmics 运行保留模式。

5. 在服务角色下，您可以使用现有的服务角色或创建新的服务角色。
6. （可选）对于标记，您最多可以为运行分配 50 个标签。
7. 选择下一步。
8. 在添加参数值页面上，提供运行参数。您可以上传指定参数的 JSON 文件，也可以手动输入值。
9. 选择下一步。
10. 在 “运行组” 面板中，您可以选择为此次运行指定运行组。有关更多信息，请参阅 创建 HealthOmics 跑步组。
11. 在 “运行缓存” 面板中，您可以选择为此次运行指定运行缓存。有关更多信息，请参阅 使用控制台配置带有运行缓存的运行。
12. 选择 Review and start run (检查并启动运行)。
13. 查看运行配置后，选择开始运行。

使用 API 开始跑步

将启动运行 API 操作与 I AM 角色和您创建的 Amazon S3 存储桶配合使用。此示例将保留模式设置为REMOVE。有关保留模式的更多信息，请参阅运行时 HealthOmics 运行保留模式。

```
aws omics start-run
     --workflow-id workflow id \
     --role-arn arn:aws:iam::1234567892012:role/service-role/
OmicsWorkflow-20221004T164236 \
     --name workflow name \
     --retention-mode REMOVE
```

作为响应，你会得到以下输出。uuid是运行所独有的，与一起outputUri可用于跟踪输出数据的写入位置。

```
{
    "arn": "arn:aws:omics:us-west-2:....:run/1234567",
    "id": "123456789",
    "uuid":"96c57683-74bf-9d6d-ae7e-f09b097db14a",
```

```
    "outputUri":"s3://bucket/folder/8405154/96c57683-74bf-9d6d-ae7e-f09b097db14a"
    "status": "PENDING"
}
```

如果工作流程的参数模板声明了任何必需的参数，则可以在启动工作流程运行时提供输入的本地 JSON 文件。JSON 文件包含每个输入参数的确切名称和参数的值。

AWS CLI 通过添加到`--parameters file://<input_file.json>`您的`start-run`请求中引用输入 JSON 文件。有关运行参数的更多信息，请参阅HealthOmics 运行输入。

您可以为运行指定工作流程版本。

```
aws omics start-run
     --workflow-id workflow id \
      ...
     --workflow-version-name '1.2.1'
```

您可以覆盖工作流程中指定的默认运行存储类型。

```
aws omics start-run
     --workflow-id workflow id \
      ...
     --storage-type STATIC
     --storage-capacity 2400
```

您也可以使用带有 GPU 工作流程 ID 的启动运行 API，如图所示。

```
aws omics start-run
     --workflow-id workflow id \
     --role-arn arn:aws:iam::1234567892012:role/service-role/
OmicsWorkflow-20221004T164236 \
     --name GPUTestRunModel \
     --output-uri s3://amzn-s3-demo-bucket1
```

获取有关工作流程运行的信息

您可以将响应中的 ID 与 get-run API 配合使用来检查运行状态，如图所示。

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

来自此 API 操作的响应会告诉您工作流程的运行状态。可能的状态是PENDING、STARTINGRUNNING、和。COMPLETED运行时COMPLETED，您可以在输出 Amazon S3 存储桶outfile.txt中找到一个名为的输出文件，该文件位于以运行 ID 命名的文件夹中。

get-run API 操作还会返回其他详细信息，例如工作流程是否为PRIVATE、工作流引擎和加速器详细信息。Ready2Run以下示例显示了私有工作流程运行时对 get-run 的响应，该工作流程在 WDL 中进行了描述，该工作流程具有 GPU 加速器且未为运行分配任何标签。

```
{
    "arn": "arn:aws:omics:us-west-2:123456789012:run/7830534",
    "id": "7830534",
    "uuid":"96c57683-74bf-9d6d-ae7e-f09b097db14a",
    "outputUri":"s3://bucket/folder/8405154/96c57683-74bf-9d6d-ae7e-f09b097db14a"
    "status": "COMPLETED",
    "workflowId": "4074992",
    "workflowType": "PRIVATE",
    "workflowVersionName": "3.0.0",
    "roleArn": "arn:aws:iam::123456789012:role/service-role/
OmicsWorkflow-20221004T164236",
    "name": "RunGroupMaxGpuTest",
    "runGroupId": "9938959",
    "digest":
 "sha256:a23a6fc54040d36784206234c02147302ab8658bed89860a86976048f6cad5ac",
    "accelerators": "GPU",
    "outputUri": "s3://amzn-s3-demo-bucket1",
    "startedBy": "arn:aws:sts::123456789012:assumed-role/Admin/<role_name>",
    "creationTime": "2023-04-07T16:44:22.262471+00:00",
    "startTime": "2023-04-07T16:56:12.504000+00:00",
    "stopTime": "2023-04-07T17:22:29.908813+00:00",
    "tags": {}
}
```

如图所示，您可以使用列表运行 API 操作查看所有运行的状态。

```
aws omics list-runs
```

要查看特定运行的所有已完成任务，请使用 list-run-tasksAPI。

```
aws omics list-run-tasks --id task ID
```

要获取任何特定任务的详细信息，请使用 get-run-task API。

```
aws omics get-run-task --id <run_id> --task-id task ID
```

运行完成后，元数据将发送到流 CloudWatch 下方**manifest/run/<run ID>/<run UUID>**。

以下是清单的示例。

```
{
    "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"
  },
  ...
```

如果 CloudWatch 日志中不存在运行元数据，则不会将其删除。您也可以使用运行 ID 通过 CLI 工具重新运行工作流程。了解更多信息并从工具 GitHub 存储库下载该HealthOmics工具。

重新运行工作流程运行

以下示例说明如何使用该rerun工具重新运行运行。您需要运行 ID，您可以从 CloudWatch 日志中检索该ID。

```
omics-rerun 9876543 --name workflow name --retention-mode REMOVE
```

如果运行存在于中 CloudWatch，则您会收到类似于以下内容的响应。

```
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": {}
}
```

如果工作流程已不存在，您会收到一条错误消息。

在 HealthOmics 工作流程中运行生命周期

您可以通过监控运行状态来跟踪运行的进度。 HealthOmics 在运行的整个生命周期中更新运行状态。

您可以使用以下任一方法检索运行状态：

- HealthOmics 控制台会在Runs页面上显示每次运行的状态。
- GetRunAPI 操作返回当前的运行状态。
- 您可以使用 EventBridge 事件监控运行状态。有关更多信息，请参阅 EventBridge 与一起使用 AWS HealthOmics。

主题

运行状态值

开始运行时， HealthOmics 将运行状态设置为Pending。随着运行的生命周期，会 HealthOmics 更新状态值以反映其当前进度。

> **Note**
>
> 除了 “正在运行” 之外，在任何运行状态下，您都不会产生任何费用。有关详细信息，请参阅下一节。

HealthOmics 支持以下运行状态值：

待处理

运行在队列中，等待开始。在开始之前，跑步通常会在短时间内处于待定状态。

- 如果您同时提交多个作业，则运行可能会在 “待定” 状态下停留更长时间。
- 在您的账户达到最大并发运行次数后，运行仍处于 “待处理” 状态。
- 如果该运行属于已达到其任何资源最大值的运行组，则该运行仍处于 “待处理” 状态。
- 您可以调整运行优先级，以便特定的排队运行在其他运行之前开始。有关运行优先级的更多信息，请参阅运行优先级。

启动

HealthOmics 创建运行并配置运行所需的资源（例如临时运行存储空间和引擎节点）。

- HealthOmics 在运行开始时配置临时运行存储空间，并在运行处于停止状态时取消配置运行存储空间。

Running

在导入过程、每个任务的处理和导出过程中，运行仍处于 “运行” 状态。

- HealthOmics 将输入文件导入到临时运行的存储文件系统。输入文件是只读的，以防止任务修改工作流中其他任务的输入。
- 在文件导出期间，将 HealthOmics 输出文件从运行存储文件系统导出到 S3 位置。
- HealthOmics 当运行状态为 “正在运行” 时 CloudWatch ，将运行日志和任务日志实时提供给。有关更多信息，请参阅 登录 CloudWatch 。

停止

导出过程完成后，运行将变为 “正在停止” 状态。

- HealthOmics 取消配置所有资源（包括运行存储文件系统和引擎节点）。

已完成

资源取消置备 HealthOmics 完成后，运行将转换为 “已完成”。

- HealthOmics 已完成所有运行任务并正确导出输出数据。
- 运行输出可在指定的 Amazon S3 URI 输出位置中找到。对于 WDL 和 CWL， HealthOmics生成运行输出摘要文件，该文件提供有关信息。HealthOmics 运行输出
- 中提供了最终运行清单日志和引擎日志（如果适用） CloudWatch。
- 对于支持任务重试的运行，处于 “已完成” 状态的运行可能包括一个或多个失败的任务。只要每个失败的任务成功重试，就会将运行 HealthOmics 转换为 “已完成”。 HealthOmics 为每次重试分配一个新的任务 ID，因此运行包括失败尝试和已完成尝试的任务 IDs 。

失败

HealthOmics 遇到了一个或多个错误，但未能完成所有运行任务。

- 在 HealthOmics 取消配置资源的同时，失败的运行会转变为 “停止” 状态。

已取消

用户发起了取消运行的请求。

- HealthOmics 停止所有正在运行的任务并取消配置所有资源。
- HealthOmics 当用户取消运行时，不会导出任何运行输出数据。对于已取消的运行，您无权访问任何中间文件。
- 取消前，您的账户会因运行状态为 “运行” 所消耗的任务和资源而产生费用。
- 如果您取消处于 “待定” 或 “正在启动” 状态的跑步，则不收取任何费用。

任务重试次数

如果任务在运行期间失败，则在以下情况下会再次 HealthOmics 重试该任务：

- 对于 WDL 工作流程， HealthOmics 支持在任务因服务错误而失败时重试任务（5XX HTTP 状态代码）。

  默认情况下，最多 HealthOmics 会尝试对失败的任务进行两次重试。您可以通过配置 WDL 定义文件来选择不重试任务。有关示例配置，请参阅 工作 HealthOmics 流程定义中的任务资源。
- 对于 Nextflow 工作流程，您可以为工作流定义中的任务配置重试条件。

- 如果运行中的每个任务最终都完成，即使它们需要重试，也会将运行 HealthOmics 转换为 “已完成”。
- HealthOmics 为每次重试分配一个新的任务 ID，因此运行包括失败尝试和已完成尝试的任务 IDs。

运行状态对定价的影响

运行状态为 “运行” 时，您的账户可能会产生费用。在任何其他运行状态下，您都不会产生任何费用。例如，当运行处于 “正在启动” 或 “停止” 状态时，不收取资源费用。

以 “运行” 状态运行会产生以下计费影响：

- 当运行状态为 “运行” 时，您的账户会因运行存储文件系统的使用量而产生费用。有关运行存储类型的信息，请参见在 HealthOmics 工作流程中运行存储类型。
- 您的账户会根据您在工作流程定义中为每个任务指定的计算和内存资源以及任务持续时间对正在运行的任务产生费用。有关更多信息，请参阅 HealthOmics 任务的计算和内存要求。
- 每项任务的最低计费阈值为一分钟。如果您运行任务的时间少于一分钟，则需要按最少一分钟的使用量付费。如果可能，将小任务组合在一起以优化成本。分组任务还可以避免多个连续任务的启动，从而缩短运行时间。

有关 HealthOmics 定价的更多信息，请参阅定HealthOmics 价。

HealthOmics 运行输出

当 WDL 或 CWL 运行完成时，输出将包括一个输出摘要文件（JSON 格式），该文件列出了运行产生的所有输出。您可以将输出摘要文件用于以下目的：

- 以编程方式确定运行生成的输出文件。
- 验证运行是否产生了所有预期的输出。

主题

运行 WDL 的输出摘要

WDL 运行完成后，将 HealthOmics 创建一个名output.json为的输出摘要文件。

对于工作流程的每个输出，文件中都有 key/value 对应的输出。密钥包含以下格式的工作流程名称和输出名称：`WorkflowName.output_name`。对于文件输出，该值是一个 S3 URI，指向 S3 中存储该文件的输出位置。对于数组 [文件] 输出，该值是 S3 的数组 URIs。

以下示例显示了名为的工作流程的output.json文件BWAMappingWorkflow。

```
{
  "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"
  ]
}
```

如果工作流程生成非文件类型（例如字符串、整数、浮点数或布尔型）的输出，则字段值为 JSON 基元。例如：

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

运行 CWL 的输出摘要

CWL 运行完成后， HealthOmics 将在以下位置创建一个名为outputs.json的输出摘要文件：

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

输出摘要文件包括输出列表。每个输出都是一 key/value 对，其中密钥是输出的名称。该值是一个包含以下属性的对象：

- location-输出文件的完全限定路径
- basename — 路径的文件名部分
- class — 输出的类型，通常是 File
- size — 文件的大小（以字节为单位）

在以下示例中，output.json 文件包含两个输出文件的列表。

```
{
  "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
  }
}
```

运行失败原因

如果运行失败，请使用 GetRunAPI 操作检索失败原因。

查看失败原因以帮助您解决运行失败的原因。下表列出了每个失败原因以及对错误的描述。

失败原因	错误描述
假设角色失败	HealthOmics 没有担任该角色的权限。为角色指定信任关系中的HealthOmics委托人。
无法启动容器错误	无法启动工作流程任务:*name*，ID：使用图像的*ID*容器：*image name*。请确保图像有效，然后重试。
无法启动容器大小错误	无法启动工作流程任务:*name*，ID：使用图像的*ID*容器：*image name*。请确保图像大小小于 25 GB，然后重试。
ECR_permission_error	HealthOmics 没有访问图片 URI 的权限。 确认 Amazon ECR 私有存储库存在并且已授予对 HealthOmics 服务主体的访问权限。
导出失败	导出失败。检查输出存储桶是否存在以及运行角色是否具有该存储桶的写入权限。
文件系统空间不足	文件系统没有足够的空间。增加文件系统的大小，然后再次运行。
图片验证失败	无法验证图片*image name*。要更正此问题，请尝试拉取镜像，然后再次将其推送到您的 ECR 存储库。
导入失败	导入失败。检查输入文件是否存在以及运行角色是否可以访问输入。
INACTIVE_OMICS_storage_RESOURCE	存 HealthOmics 储 URI 未处于活动状态。激活读取集并重试。要了解有关激活读集的更多信息，请参阅在中激活读取集 HealthOmics。
未找到输入 URI	提供的 URI 不存在:*uri*. 检查 URI 路径是否存在并确认该角色可以访问该对象。
实例预留失败	实例容量不足，无法完成工作流程运行。请稍候，再次尝试运行工作流程。
无效_ECR_IMAGE_URI	Amazon ECR 图片 URI 结构无效。请提供有效的 URI，然后重试。

失败原因	错误描述
任务资源值无效	请求的 GPU、CPU 或内存要么太高，无法提供可用的计算容量，要么小于任务的最小值 1 *ID*。
URI 输入无效	URI 结构无效*uri*。请检查 URI 结构并重试。
修改后的输入资源	运行开始后，所提供*uri*的 URI 已被修改。重试运行。
内存不足错误	工作流程任务*ID*内存不足。增加工作流程定义中的内存值，然后重试运行。
运行任务失败	由于任务失败，运行失败。要调试任务失败，请使用 GetRunTaskAPI 操作和 Amazon CloudWatch 日志流。
RUN_TIMED_OUT	*number*几分钟后运行超时。
服务错误	服务中出现暂时性错误。再次尝试运行工作流程。
不支持的输入大小	总输入大小太高。请减小输入大小，然后重试。
工作流_运行_失败	工作流程运行失败。查看 CloudWatch 日志引擎日志流：*ID*以调试故障。
工作流版本验证失败	HealthOmics 不支持请求的 Nextflow 版本:*version*--。支持的最新版本是*version*。请将您的 Nextflow 版本修改为支持的版本，然后重试。
不支持的_GPU_实例_类型	中不支持请求的实例类型*Region*。使用该区域支持的 GPU 实例类型重试运行。可用的实例类型有*GPU instance types*。

无响应跑步指南

在开发新的工作流程时，如果您的代码存在问题，并且任务无法正常退出进程，则运行或特定任务可能会变得 “卡住” 或 “挂起”。这可能很难进行故障排除和 catch，因为任务长时间运行是正常的。要防止和识别无响应的运行，请遵循以下各节中建议的最佳做法。

防止运行无响应的最佳实践

- 确保您正在关闭任务代码中打开的所有文件。打开太多文件有时会导致工作流程引擎出现线程问题。

- 工作流任务创建的后台进程应在任务退出时退出。但是，如果后台进程无法干净退出，则必须在任务代码中明确关闭该进程。
- 确保您的进程不会在不退出的情况下循环。这可能会导致运行无响应，需要更改工作流程定义代码才能解决。
- 为您的任务提供适当的内存和 CPU 分配。分析CloudWatch 日志或使用成功完成工作流运行时的计算分配来验证您的计算分配是否最佳。运行分析器使用 Run Analyzer `headroom` 参数添加额外的余量，确保流程有足够的资源来完成。在分配的内存和 CPU 中包括至少 5% 的余量，以考虑后台操作系统进程。
  - 此外，如果实例需要更高的吞吐量，请增加实例带宽大小。小于 16 vCPUs（大小为 4xl 及更小）的 Amazon EC2 实例可能会出现吞吐量激增的情况。有关 Amazon EC2 实例吞吐量的更多信息，请参阅亚马逊 EC2 可用实例带宽。
- 确保在运行时使用了正确的文件系统大小。对于使用静态运行存储的无响应运行，可以考虑增加静态运行存储分配，以便在文件系统上实现更高的 IO 吞吐量和存储容量。分析运行清单以查看最大文件系统存储空间，使用运行分析器确定是否需要增加文件系统分配。

捕捉无响应的跑步的最佳实践

- 开发新工作流程时，请使用设置了最大运行时间限制的运行组来捕获 runaway 代码。例如，如果运行需要 1 小时才能完成，则将其放入在 2 或 3 小时（或根据您的用例不同的时间段）后超时的运行组中，以捕获失控的作业。此外，还要应用缓冲区来考虑处理时间的差异。
- 设置一系列具有不同最大运行时间限制的运行组。例如，您可以根据您的预期工作流程持续时间将短期运行分配给一个在几小时后终止运行的运行组，以及一个在几天后终止运行的长跑组。
- HealthOmics 默认的最大运行持续时间服务限制为 604,800 秒或 7 天，可通过配额工具中的请求进行调整。只有当您的跑步持续时间接近一周时，才可以申请增加此配额的服务限制。如果您同时使用短跑和长跑，并且不使用运行组，请考虑将长时间运行的运行放在具有更高最大运行持续时间服务限制的单独帐户中。
- 检查CloudWatch 日志中是否有您怀疑可能没有响应的任务。如果任务通常会输出常规日志语句，但长时间没有这样做，则该任务很可能被卡住或冻结。

如果遇到无响应的跑步怎么办

- 取消运行以避免产生额外费用。
- 检查任务日志，检查是否有任何进程未能正确退出。
- 检查引擎日志以识别任何异常的发动机行为。

- 将无响应运行的任务和引擎日志与成功完成的相同运行的任务和引擎日志进行比较。这可以帮助识别可能导致无响应行为的任何差异。
- 如果您无法确定根本原因，请提出支持案例并提供以下内容：
  - 卡住运行的 ARN 和成功完成的相同运行的 ARN。
  - 引擎日志（运行被取消或失败后可用）
  - 无响应任务的任务日志。我们不需要工作流程中所有任务的任务日志即可进行故障排除。

HealthOmics 运行中的任务生命周期

任务是运行中的单个进程。 HealthOmics 将工作流程中的每项任务映射到最适合任务所需资源的 omics 计算实例类型。您可以在工作流程定义中指定所需的资源。有关更多信息，请参阅 HealthOmics 任务的计算和内存要求。

HealthOmics 提供临时运行存储空间供任务使用。 HealthOmics 将任务输入文件作为只读文件复制到临时运行存储中。 HealthOmics 提供了符号链接，以便任务可以访问工作目录中的输入文件。该任务只能访问您在工作流程定义文件中声明的文件。

任务状态值

您可以通过监控任务状态来跟踪任务的进度。开始运行时， HealthOmics 将运行中每项任务Pending的任务状态设置为。当任务开始并在其生命周期中进行时，会 HealthOmics 更新状态值以反映其当前进度。

您可以使用以下任一方法检索任务状态：

- HealthOmics 控制台在Run details页面上显示运行中每项任务的状态。
- GetRunTaskAPI 操作返回任务状态。
- 您可以使用 EventBridge 事件监控任务状态。有关更多信息，请参阅 EventBridge 与一起使用 AWS HealthOmics。

您可以使用 GetRunTask API 操作检索任务的当前状态。 HealthOmics 控制台会在Run details页面上显示运行中每项任务的状态。

HealthOmics 支持以下任务状态值：

待处理

您的任务在队列中，正在等待启动。任务在开始之前会在短时间内处于待处理状态。

- 在您的账户达到最大并发任务数后，任务仍处于待处理状态。
- 如果运行是已达到任何资源最大值的运行组的一部分，则任务仍处于待处理状态。
- 您可以调整运行优先级，以便特定的排队运行及其任务在其他排队运行之前开始。有关运行优先级的更多信息，请参见 运行优先级

启动

HealthOmics 正在创建任务并配置任务所需的资源，例如工作流任务节点。

Running

正在处理任务时，任务状态 HealthOmics 为 “正在运行”。

停止

完成任务处理并导出输出数据后，任务将转换为 “停止”。

- HealthOmics 取消配置工作流任务节点。

已完成

HealthOmics 已完成任务处理并将输出数据传输到运行的存储文件系统。

失败

HealthOmics 在处理任务时遇到了错误，但没有完成。

- 任务将转换为 “停止” 状态（HealthOmics 取消配置资源），然后转换为 “失败” 状态。
- 如果错误是服务错误（5XX HTTP 状态码），并且工作流支持重试此任务，则会 HealthOmics 尝试再次处理该任务。 HealthOmics 为重试分配一个新的任务 ID。

已取消

HealthOmics 在用户发起取消运行的请求后停止任务。

- 任务将转换为 “停止” 状态（HealthOmics 取消配置资源），然后转换为 “已取消” 状态。

工作流任务疑难解答

以下是对任务进行故障排除的最佳做法和注意事项。

- 任务日志依赖于任务STDOUT并STDERR由任务生成。如果任务中使用的应用程序没有生成其中任何一个，则不会有任务日志。要帮助调试，请在verbose模式下使用应用程序。
- 要查看任务中正在运行的命令及其插值值，请使用 `set -x` Bash 命令。这可以帮助确定任务是否使用了正确的输入，并确定哪些错误可能使任务无法按预期运行。

- 使用echo命令将变量的值输出到 STDOUT或STDERR。这可以帮助您确认它们是否按预期进行设置。
- 使用诸如ls -l <name_of_input_file>此类的命令来确认输入是否存在并且大小符合预期。如果不是，则可能会发现先前的任务由于错误而产生空输出时存在问题。
- 使用任务脚本df -Ph . | awk 'NR==2 {print $4}'中的命令来确定任务当前可用的空间，并帮助确定在哪些情况下可能需要使用额外的存储分配来运行工作流程。

在任务脚本中包含上述任何命令都假设任务容器也包含这些命令，并且它们位于容器环境中。path

为私有 HealthOmics 工作流程运行优化

您可以根据总成本、总运行时间或两者的组合来优化运行。 HealthOmics 提供数据和工具，帮助您做出运行优化决策。运行优化不适用于 Ready2Run 工作流程，因为您无法控制该服务如何管理这些工作流程的资源配置。

第一步是了解运行中任务的当前任务资源使用情况和成本，然后应用优化运行成本和性能的方法。

主题

运行分析器

HealthOmics 提供了一个名为 Run Analyzer 的开源工具。该工具可提取运行的任务级资源使用信息，并建议成本和运行性能的优化机会。

> **Note**
>
> 运行分析器根据运行该工具时的标 AWS 价估算任务成本和潜在的成本节约。评估优化建议，并实施对您的用例有意义的优化建议。测试您采用的优化，确保它们适用于您的运行。

运行 Analyzer 执行以下任务：

- 评估内存和计算瓶颈。
- 确定内存或 CPU 配置过剩的任务，并推荐可以降低成本的新实例大小。
- 计算单个任务的成本估算值，并计算应用建议后可能节省的成本。
- 为您提供任务的时间表视图，以便您可以验证任务依赖关系和处理顺序。时间轴还可以帮助您识别长时间运行的任务。
- 提供有关运行存储空间的文件系统大小的建议。
- 显示任务配置时间，以便您可以确定大型容器装载可能会减慢配置时间的区域。
- 该工具包括一个输入参数（净空），可用于控制优化建议的积极性。

以下各节包含有关使用 Run Analyzer 优化运行的具体建议。

确定运行成本

您可以使用以下方法和指南来确定运行成本：

- 要查看账单周期的总运行成本，请执行以下步骤：

  1. 打开 B illing and Cost Managem ent 控制台，然后选择账单。
  2. 在“按服务收费”中，展开 Omics。
  3. 展开区域，然后按组学实例类型、运行存储类型和 Ready2Run 工作流程逐项查看所有运行的成本。

- 要生成包含每次运行信息的成本报告，请执行以下步骤：

  1. 打开 B illing and Cost Managem ent 控制台，然后选择“数据导出”。
  2. 选择“创建”以创建新的数据导出。
  3. 输入数据导出的导出名称。将其他字段保留为默认值以创建 CUR（成本和使用情况）报告。
  4. 对于时间粒度，请选择每小时或每天。
  5. 在“数据导出存储设置”下，执行以下配置步骤：

     a. 为数据导出配置一个 Amazon S3 存储桶。

     b. 对于文件版本控制，选择是覆盖现有的导出文件还是每次都创建一个新文件。

     系统将在接下来的 24 小时内生成第一份报告，随后每天生成一次报告。

6. 有关如何创建数据导出的更多信息，请参阅《数据导出用户指南》中的创建AWS数据导出。

- 您可以按类别（例如按团队或项目）标记运行以监控和优化成本。如果您使用标签，请按照以下步骤按标签类别查看运行成本：

  1. 打开账单和成本管理控制台，然后选择 Cos t Explorer。
  2. 在报告参数 > 分组依据中，选择标签作为维度。然后选择所需的标签名称。

- 要查看任务的资源使用情况，请查看运行清单日志 CloudWatch。有关更多信息，请参阅 HealthOmics 使用 CloudWatch 日志进行监控。
- 使用该运行分析器工具提取运行的任务资源使用信息。

确定运行时间使用情况

您可以使用以下方法来帮助您调查运行时使用情况：

- 在控制台的 “运行” 页面上，您可以查看一次运行的总运行时间。
- 在运行详细信息页面上，您可以查看以下项目：
  - 查看一次运行的总运行时间。
  - 查看运行中每项任务的运行时间。
  - 选择其中一个链接来查看 Amazon S3 中的日志，或者查看运行日志或运行清单日志 CloudWatch。
- 从 “运行任务” 列表中，选择任务的 “查看日志” 链接以查看任务日志 CloudWatch。
- 对 `listRuns` API 操作的响应包括运行开始时间和停止时间，因此您可以计算总运行时间。
- 该运行分析器工具在时间轴视图上显示任务持续时间。此工具提供任务处理顺序的可视化表示，您可以将其与预期顺序进行匹配。

优化运行的方法

HealthOmics 自动配置、管理和优化执行数据暂存的资源（例如数据导入和数据导出）。 HealthOmics 还会启动并运行适用于您的工作流程的工作流引擎。但是，您可以通过设置各种运行配置来影响运行开始时间、任务开始时间和任务总体运行时间。工作流程定义和设计的总体方法也会影响任务的运行时间。以下列表描述了可能影响运行和任务性能的因素：

运行存储类型

运行存储类型会影响运行性能和运行配置时间。动态运行存储配置速度更快，并且永远不会耗尽内存，因为它可以根据您的运行存储需求进行动态扩展。动态运行存储也非常适合开发中的工作流程，在这种工作流程中，您可能经常启动和停止工作流程以解决问题。

静态运行存储需要更长的文件系统配置时间，但可以更快地完成某些运行，通常是在运行具有高任务并发性或需要大于 9.6 TiB 的文件系统容量的情况下。静态运行存储非常适合 I/O 要求很高的长时间运行的工作流程。

为了帮助您评估给定运行中每种运行存储类型的成本与性能，您可以尝试 A/B 测试，以了解哪种运行存储类型可提供更好的性能。另外，可以考虑在开发周期中使用动态运行存储，然后使用静态运行存储进行大规模生产运行。

有关运行存储类型的更多信息 在 HealthOmics 工作流程中运行存储类型

过度配置运行静态存储

如果您的工作流任务计算受到I/O, consider over-provisioning the static run storage. Storage cost increases with its size, but maximum throughput of the file system also increases. If an expensive compute task is experiencing I/O瓶颈的限制，则增加文件系统大小以缩短任务运行时间可能会降低总体成本。

减小容器镜像的大小

当每个任务启动时， HealthOmics 加载您为该任务指定的容器。较大的集装箱需要更长的时间才能装载。优化容器使其尽可能小，以提高启动新任务的效率。如果您向容器中添加大型数据集，请考虑将数据集存储在 S3 中，然后让您的工作流程从 S3 导入数据。有关 HealthOmics 支持的最大容器大小，请参阅HealthOmics 工作流程固定大小配额。

任务大小

您可以将小型的连续任务合并成单个任务，以节省任务配置时间。此外， HealthOmics 还收取一分钟的最短任务时长费用，因此合并任务可以降低成本。在组合任务中，你可以使用 Unix 管道来避免序列化和反序列化文件 I/O 的成本。

文件压缩

避免过度压缩工作流程中间文件。大多数基因组学格式使用 “gzip” 或 “block gzip” 压缩。解压缩任务输入文件和重新压缩任务输出文件可能会占用任务总体 CPU 使用率的很大一部分。某些基因组学应用程序允许您在序列化输出时设置压缩级别。通过降低压缩级别，可以缩短 CPU 时间，尽管较大的文件会增加写入磁盘所花费的时间。根据任务和应用程序，您可以找到运行时间最短的中间

文件的最佳压缩级别。我们建议您首先将输出文件最大的任务作为目标。2 的压缩级别适用于多种场景。您可以针对您的用例从这个级别开始，然后通过尝试其他压缩级别来比较结果。

线程数

如果您在任务定义中指定线程，请将线程数设置为与请求的数量 v 相同的值CPUs。

指定计算和内存

如果您未在任务中指定内存或计算资源，则会将最小的实例类型 (`omics.c.large`) HealthOmics 指定为默认值。如果您想分配更大的实例类型， HealthOmics 请明确声明您的内存和计算需求。

HealthOmics 分配您请求的 v CPUs、内存和 GPU 资源的数量。例如，如果你要求 15v CPUs 和 33GiB，则会为你的任务 HealthOmics 分配一个 omics.m.4xl 实例（16v CPUs、64GB），但你的任务只能使用 15 v 和 33GiB。CPUs 因此，我们建议您请求与 omics 实例相匹配的 v CPUs 和内存资源。

将多个样本批处理成一次运行

由于文件系统配置在运行开始时需要时间，因此您可以通过将多个样本批处理到同一个运行中来节省配置时间。在决定采用这种方法之前，请考虑以下因素：

- 单个不良样本可能导致工作流程失败，因此批处理样本可能会增加失败的工作流程数量。如果您不确定自己的工作流程在大多数情况下都能成功，那么每个样本运行一次可能是更好的方法。
- HealthOmics 为整个工作流程分配一个运行存储文件系统。对于一批样品，请确保指定足够大的运行存储空间来处理所有样本。
- 每个工作流程都有最大运行存储量，因此这可能会限制您可以添加到批次中的样本数量。
- 最小运行存储大小为 1.2 TiB，因此，如果工作流程使用的存储空间比每个样本的最小存储空间少得多，则批处理可以降低成本。
- 运行存储可以同时处理多个连接，因此使用相同的运行存储空间执行多个任务不应造成 I/O 瓶颈。
- 每次运行都有自己的一组标签。如果您使用用于预算或跟踪的信息来标记工作流程，则最好使用单独的运行。
- IAM 角色适用于整个运行。每个用户都可以访问一批样本的所有数据。将工作流程分开后，您就能够使用更精细的权限。
- HealthOmics 为工作流程中的最大并发工作流数量和最大并发任务数设置账户级别配额。有关如何申请提高这些配额的信息，请参阅HealthOmics 服务配额。

使用容器镜像的参数

对容器镜像进行参数化，而不是将其嵌入工作 URIs 流程中。当它们是运行参数时， HealthOmics 会在运行开始之前验证运行是否可以访问您的容器。否则，任务将在运行期间失败，因为任何已完成的任务都产生了费用。此外，由于这些是参数化输入，因此会在运行清单中 HealthOmics 生成校验和，从而改善运行来源。

使用 linter

在运行新工作流程之前，使用 linter 查找常见的工作流程错误。有关更多信息，请参阅 中的工作流程提示 HealthOmics。

用于 EventBridge 举报问题

使用 EventBridge 自定义警报来捕捉特定于您的业务逻辑的异常。

使用序列存储

考虑对源数据使用序列存储，以节省存储成本。如需了解更多信息，请通过 HealthOmics博客文章查看任何规模的经济高效地存储组学数据。

两次运行之间文件大小差异的影响

用户通常使用少量测试数据来设计和测试运行，然后在生产运行中遇到各种各样的数据，文件大小差异很大。在优化运行时，请务必将此差异考虑在内。

以下列表描述了文件大小存在显著差异的优化建议：

在测试数据中改变文件大小

在开发过程中，尽量使用具有代表性差异的测试数据。

使用运行分析器

使用 Run Analyzer 工具对各种样本进行分析，以考虑数据大小的差异。

您可以使用运行分析器来了解生产数据样本中运行之间的差异。使用 Run Analyzer 中的`--batch`模式为一批运行生成统计数据，并分析处理数据集中的异常值所需的最大计算资源。

例如，您可以为运行分析器提供批处理模式下的完整数据流单元，以了解整个流通池的 vCPU 和内存利用率峰值。

减少输入数据集的大小差异

如果您发现样本大小差异很大，则可以将上游的样本分开， HealthOmics 并为每个批次选择不同的文件系统大小，以节省运行存储成本。

在 WDL 中，使用`size`函数将大样本和小样本的单个任务的资源分配分开。将此策略应用于最昂贵的任务，以产生最大的影响。

在 Nextflow 中，使用条件资源根据文件大小或文件名对资源分配进行分层。有关更多信息，请参阅 Nextflow GitHub 网站上的有条件流程资源。

不要过早优化

在投入大量的性能调整工作之前，请先完成工作流程代码和逻辑。更改代码可能会对所需资源产生重大影响。如果您在开发过程中过早优化运行，则可能会过度优化，或者如果稍后工作流程定义发生变化，则可能需要再次优化。

定期重新运行运行分析器工具

如果您随着时间的推移对工作流程定义进行了更改，或者样本方差发生了变化，请定期运行 Run Analyzer 工具来帮助您进行其他优化。

优化资源并发的方法

HealthOmics 提供以下功能，可帮助您在大规模处理运行时控制和管理成本：

- 使用运行组来控制您的成本和资源使用情况。可以在运行组中设置并发运行次数 v CPUs 和每个任务的总运行时间的最大值。 GPUs如果不同的团队或小组使用相同的帐户，则可以为每个团队创建单独的跑步小组。您可以通过配置运行组的最大值来控制每个团队的资源使用量和成本。有关更多信息，请参阅 创建 HealthOmics 跑步组。
- 在开发过程中，您可以配置一个具有较低最大值的单独运行组，以捕获失控的任务。
- Service Quotas 还有助于保护您的账户免受过多资源请求的影响。有关 Service Quotas 的信息，包括如何请求增加配额值，请参阅 HealthOmics 服务配额

删除中的跑步和跑步组 HealthOmics

当您不再需要跑步或跑步组时，可以使用 AWS CLI、API 或控制台将其删除。

除了删除运行之外，您还可以取消运行。要取消运行，其状态必须为`PENDING`、`STARTING`、`RUNNING`、或`STOPPING`。

> **Note**
>
> 取消运行时， HealthOmics 不会保存任何运行输出。

以下 AWS CLI 命令显示如何取消运行。要运行此示例，*run id*请将替换为您要取消的运行的 ID。如果成功，则没有响应。

```
aws omics cancel-run --id run id
```

以下 AWS CLI 命令删除运行。只有在运行完成或取消后才能将其删除。要运行此示例，请将*run id*替换为要删除的运行的 ID。如果成功删除运行，则没有响应。

```
aws omics delete-run --id run id
```

您也可以删除运行组。只有当没有与状态为、、或的运行组关联的运行时 PENDING STARTINGRUNNING，才能删除运行组STOPPING。

以下示例说明如何使用删除运行组。 AWS CLI 您将不会收到回复。要运行此示例，请将*run group id*替换为要删除的运行组的 ID。

```
aws omics delete-run-group --id run group id
```

创建 HealthOmics 跑步组

您可以选择创建一个运行组，以限制添加到该组的运行的计算资源。跑步小组可以帮助你：

- 对跑步进行排队，以免超过服务限制。
- 通过设置最大运行持续时间来捕捉失控的任务。
- 管理每次运行的优先级，以便最重要的运行首先完成。

如果您设置了最大并发 vCPU、GPU 或运行次数，则当达到最大值时，运行任务将排队。如果您设置了最大运行持续时间，则如果超过最大运行持续时间，则运行将失败。

使用运行优先级设置来确定运行组内的优先级。

服务限制优先于运行组限制。例如，如果您将运行组的最大值设置为高于某个区域中的服务最大值，则会 HealthOmics 应用服务最大值。

主题

运行优先级

您可以使用运行优先级来确定运行组中运行的优先级。

如果多个运行具有相同的优先级，则首先开始的运行的优先级更高。

您也可以为不在跑步组中的跑步设置优先级。将优先级与不在运行组中的所有其他运行的优先级进行比较

开始运行时可以设置运行优先级。有关更多信息，请参阅 开始磨合 HealthOmics。

使用控制台创建跑步组

创建运行组

1. 打开 HealthOmics 管理控制台。
2. 在左侧导航窗格中，选择 Run groups。
3. 在 “运行组” 页面上，选择 “创建运行组”。
4. 在创建运行组详细信息页面上，提供以下信息
   - 运行组名称-此运行组的唯一名称。
   - 并发运行的最大 vCPU-在运行组中所有活动运行中CPUs 可以同时运行的最大 v 数。
   - Max GPUs-在运行组 GPUs 中所有活动运行中可以同时运行的最大数量。
   - 每次@@ 运行的最大运行时间（分钟）-每次运行的最长时间（以分钟为单位）。如果运行超过最大运行时间，则运行会自动失败。
   - 最大并发运行次数-可以同时运行的最大运行次数。
5. （可选）您最多可以向运行组添加 50 个标签。
6. 选择 “创建跑步组”。

使用 CLI 创建运行组

要创建运行组，请使用 create-run-groupAPI 操作创建名为的跑步组TestRunGroup。以下示例将最大运行时间设置为 20 CPUs、10 GPUs、5 次，最大运行持续时间设置为 600 分钟。

```
aws omics create-run-group --name TestRunGroup \
--max-cpus 20 \
--max-gpus 10 \
--max-duration 600 \
--max-runs 5
```

来自此 API 操作的响应包括新创建的 ID RunGroup。

```
{
    "arn": "arn:aws:omics:us-west-2:12345678901:runGroup/2839621",
    "id": "2839621",
    "tags": {}
}
```

要获取有关运行组的更多信息，请将此 ID 用于 get-run-groupAPI 操作，如以下示例所示。

```
aws omics get-run-group --id run group id
```

响应包括运行组的限制设置和分配的标签。

```
{
    "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
}
```

您还可以使用 list-run-groupAPI 操作查看所有创建的运行组。

```
aws omics list-run-groups
```

HealthOmics 运行时调用缓存

AWS HealthOmics 支持私有工作流程的呼叫缓存（也称为恢复）。呼叫缓存会在运行完成后保存已完成的工作流任务的输出。后续运行可以使用缓存中的任务输出，而不必再次计算任务输出。呼叫缓存可减少计算资源使用量，从而缩短运行时间并节省计算成本。

运行完成后，您可以访问缓存的任务输出文件。要执行高级任务调试和故障排除，您可以通过在工作流定义中将中间任务文件指定为任务输出来缓存这些文件。

您可以使用呼叫缓存来保存失败运行后已完成的任务结果。下一次运行从上次成功完成的任务开始，而不是再次计算已完成的任务。

如果找 HealthOmics 不到任务的匹配缓存条目，则运行不会失败。 HealthOmics 重新计算任务及其相关任务。

有关解决呼叫缓存问题的信息，请参阅解决呼叫缓存问题。

主题

呼叫缓存的工作原理

要使用呼叫缓存，您需要创建运行缓存并将其配置为与缓存数据关联的 Amazon S3 位置。开始运行时，需要指定运行缓存。运行缓存不是专用于一个工作流程的。从多个工作流程运行可以使用同一个缓存。

在运行的导出阶段，系统会将已完成的任务输出导出到 Amazon S3 位置。要导出中间任务文件，请在工作流定义中将这些文件声明为任务输出。呼叫缓存还会在内部保存元数据，并为每个缓存条目创建唯一的哈希值。

对于运行中的每个任务，工作流引擎都会检测该任务是否有匹配的缓存条目。如果没有匹配的缓存条目，则 HealthOmics 计算任务。如果有匹配的缓存条目，引擎将检索缓存的结果。

要匹配缓存条目，请 HealthOmics 使用本机工作流引擎中包含的哈希机制。 HealthOmics扩展了这些现有的哈希实现以考虑 HealthOmics 变量，例如 S3 ETag 和 ECR 容器摘要。

HealthOmics 支持以下工作流程语言版本的呼叫缓存：

- WDL 版本 1.0、1.1 和开发版本
- Nextflow 版本 23.10 和 24.10
- 所有 CWL 版本

> **Note**
>
> HealthOmics 不支持 Ready2Run 工作流程的呼叫缓存。

主题

责任共担模式

用户之间有共同的责任 AWS 来确定任务和运行是否适合呼叫缓存。当所有任务均为等性时，呼叫缓存可获得最佳结果（使用相同输入重复执行任务会产生相同的结果）。

但是，如果任务包含非确定性元素（例如随机数生成或系统时间），则使用相同的输入重复执行任务可能会导致不同的输出。这可能会通过以下方式影响呼叫缓存的有效性：

- 如果 HealthOmics 使用的缓存条目（由上一次运行创建）与任务执行为当前运行产生的输出不相同，则该运行产生的结果可能与没有缓存的同一次运行不同。
- HealthOmics 由于任务输出不确定，可能找不到本应匹配的任务的匹配缓存条目。如果找不到有效的缓存条目，则运行会不必要地重新计算任务，从而降低使用呼叫缓存节省成本的好处。

以下是已知的任务行为，这些行为可能导致影响呼叫缓存结果的不确定性结果：

- 使用随机数生成器。

- 依赖于系统时间。
- 使用并发（竞争条件可能导致输出差异）。
- 获取超出任务输入参数中指定范围的本地或远程文件。

有关可能导致非确定性行为的其他场景，请参阅 Nextflow 文档网站上的非确定性流程输入。

如果您怀疑某项任务产生的输出不确定，请考虑使用工作流引擎功能（例如 Nextflow 中的缓存选择退出），以避免缓存不确定性的特定任务。

我们建议您在呼叫缓存效率低下或输出不同于预期可能带来风险的任何环境中启用呼叫缓存之前，仔细检查您的特定工作流程和任务要求。例如，在确定呼叫缓存是否适合临床用例时，应仔细考虑呼叫缓存的潜在局限性。

任务的缓存要求

HealthOmics 缓存满足以下要求的任务的任务输出：

- 该任务必须定义一个容器。 HealthOmics 不会缓存没有容器的任务的输出。
- 该任务必须产生一个或多个输出。您可以在工作流定义中指定任务输出。
- 工作流程定义不得使用动态值。例如，如果您向任务传递一个参数，其值会随着每次运行而递增，则 HealthOmics 不会缓存任务输出。

> **Note**
>
> 如果运行中的多个任务使用相同的容器镜像，则为所有这些任务 HealthOmics 提供相同的镜像版本。 HealthOmics 拉取镜像后，它会在运行期间忽略对容器镜像的任何更新。这种方法提供了可预测且一致的体验，并防止了在运行中途部署的容器映像更新可能出现的潜在问题。

运行缓存性能

当你为跑步开启呼叫缓存时，你可能会注意到对运行性能的以下影响：

- 在第一次运行期间， HealthOmics 保存正在运行的任务的缓存数据。此次运行的导出时间可能会更长，因为呼叫缓存会增加导出数据量。
- 在随后的运行中，当从缓存中恢复运行时，它可能会缩短处理步骤的数量并缩短运行时间。

- 如果您还选择将中间文件声明为输出，则您的导出时间可能会更长，因为这些数据可能会更加冗长。

缓存数据保留和失效事件

运行缓存的主要目的是优化运行中任务的计算。如果任务有有效的匹配缓存条目，则 HealthOmics 使用缓存条目而不是重新计算任务。否则，将 HealthOmics 恢复到默认的服务行为，即重新计算任务及其相关任务。通过使用这种方法，缓存未命中不会导致运行失败。

我们建议您管理运行缓存的大小。随着时间的推移，由于工作流引擎或 HealthOmics 服务更新，或者由于您在运行或运行任务中所做的更改，缓存条目可能不再有效。以下各节提供了更多详细信息。

主题

清单版本更新和数据新鲜度

该 HealthOmics 服务可能会定期引入新功能或工作流引擎更新，从而使部分或全部运行缓存条目失效。在这种情况下，您的运行可能会遇到一次性缓存丢失。

HealthOmics 为每个缓存条目创建一个 JSON 清单文件。对于 2025 年 2 月 12 日之后开始的运行，清单文件包含版本参数。如果服务更新使任何缓存条目失效，则会 HealthOmics 增加版本号，以便您可以识别要删除的旧缓存条目。

以下示例显示了版本设置为 2 的清单文件：

```
{
     "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,
}
```

对于使用不再有效的缓存条目的运行，请重建缓存以创建新的有效条目。每次运行都要执行以下步骤：

1. 在缓存保留设置为 “始终缓存” 的情况下开始运行一次。此次运行会创建新的缓存条目。
2. 对于后续运行，请将缓存保留期设置为以前的设置（“始终缓存” 或 “失败时缓存”）。

要清理不再有效的缓存条目，您可以从 Amazon S3 缓存存储桶中删除这些缓存条目。 HealthOmics 切勿重复使用这些缓存条目。如果您选择保留无效的条目，则不会对您的跑步产生任何影响。

> **Note**
>
> 呼叫缓存将任务输出数据保存在为缓存指定的 Amazon S3 位置，这会对您 AWS 账户产生费用。

运行缓存行为

您可以设置运行缓存行为以保存失败运行（失败时缓存）或所有运行（始终缓存）的任务输出。创建运行缓存时，需要为所有使用该缓存的运行设置默认缓存行为。当你开始运行时，你可以覆盖默认行为。

Cache on failure如果您要调试的工作流程在成功完成多个任务后失败，则此功能很有用。如果哈希考虑的所有唯一变量都与前一次运行相同，则后续运行将从上次成功完成的任务中恢复。

Cache always如果您要在成功完成的工作流程中更新任务，则非常有用。我们建议您按照以下步骤操作：

1. 创建新跑步。将 “缓存” 行为设置为 “始终缓存”，然后开始运行。
2. 运行完成后，更新工作流程中的任务并开始新的运行，行为设置为 “始终缓存”。此运行将处理更新的任务以及任何依赖于更新任务的后续任务。所有其他任务都使用缓存的结果。
3. 根据需要重复步骤 2，直到更新任务的开发完成。
4. 在将来的运行中，根据需要使用更新的任务。如果您计划在这些运行中使用新的或不同的输入，请记住将后续运行切换到 “失败时缓存”。

> **Note**
>
> 我们建议在使用相同的测试数据集时使用始终缓存模式，但不适用于批量运行。如果您为大批量运行设置此模式，则系统可以将大量数据导出到 Amazon S3，从而增加导出时间和存储成本。

控制运行缓存大小

HealthOmics 不会删除或自动存档任何运行缓存数据，也不会应用 Amazon S3 清理规则来管理缓存数据。我们建议您定期清理缓存，以节省 Amazon S3 存储成本并保持运行缓存大小易于管理。您可以直接删除文件或在运行缓存存储桶上设置数据 retention/replication 策略。

例如，您可以将 Amazon S3 生命周期策略配置为在 90 天后使对象过期，也可以在每个开发项目结束时手动清理缓存数据。

以下信息可以帮助您管理缓存数据大小：

- 您可以通过查看 Amazon S3 来查看缓存中有多少数据。 HealthOmics 不监控或报告缓存大小。
- 如果删除有效的缓存条目，则后续运行不会失败。 HealthOmics 重新计算任务及其相关任务。
- 如果您修改了缓存名称或目录结构， HealthOmics 导致找不到任务的匹配条目，则会 HealthOmics 重新计算任务。

如果您需要检查缓存条目是否仍然有效，请检查缓存清单版本号。有关更多信息，请参阅 清单版本更新和数据新鲜度。

创建运行缓存

创建运行缓存时，需要为缓存数据指定一个 Amazon S3 位置。这些数据必须可以立即访问。呼叫缓存不会检索在 Glacier 中存档的对象（例如 GFR 和 GDA 存储类）。

如果缓存数据的 Amazon S3 存储桶归其他人所有 AWS 账户，请在创建运行缓存时提供该账户 ID。

使用控制台创建运行缓存

在控制台中，按照以下步骤创建运行缓存。

1. 打开 HealthOmics 管理控制台。
2. 在左侧导航窗格中，选择运行缓存。

3. 在 “运行缓存” 页面中，选择 “创建运行缓存”。
4. 在 “创建运行缓存” 页面的运行缓存详细信息面板中，配置以下字段：
   a. 输入运行缓存的名称。
   b. （可选）输入描述。
   c. 输入缓存输出的 S3 位置。选择与您的工作流程位于同一区域的存储桶。
   d. （可选）输入存储桶所有者的，以验证存储桶所有权。 AWS 账户 如果您未输入值，则默认值为您的账户 ID。
   e. 在 “缓存行为” 下，配置默认行为（是缓存失败运行的输出，还是缓存所有运行的输出）。当你开始运行时，你可以选择覆盖默认行为。
5. （可选）将一个或多个标签与运行缓存相关联。
6. 选择 “创建运行缓存”。控制台在 “运行缓存” 表中显示新的运行缓存。

使用 CLI 创建运行缓存

使用 C create-run-cacheLI 命令创建运行缓存。默认的缓存行为是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"
```

如果创建成功，您将收到包含以下字段的响应。

```
{
  "arn": "string",
  "id": "string",
  "status": "ACTIVE"
  "tags": {}
  }
```

更新运行缓存

您可以更改缓存名称、描述、标签或缓存行为，但不能更改缓存的 S3 位置。

使用控制台更新运行缓存

在控制台中，按照以下步骤更新运行缓存。

1. 打开 HealthOmics 管理控制台。
2. 在左侧导航窗格中，选择运行缓存。
3. 从 “运行缓存” 表中，选择要更新的运行缓存，然后选择 “编辑”。
4. 在运行缓存详细信息面板中，您可以更新运行缓存名称、描述和缓存行为字段。
5. （可选）将一个或多个新标签与运行缓存相关联，或移除现有标签。
6. 选择 “保存运行缓存”。

使用 CLI 更新运行缓存

使用 C update-run-cacheLI 命令更新运行缓存。

```
aws omics update-run-cache \
      --name "workflow 123 run cache" \
      --id "workflow id" \
      --description "my run cache" \
      --cache-behavior "CACHE_ALWAYS"
```

如果更新成功，您会收到一条没有数据字段的响应。

删除运行缓存

如果没有正在使用的运行缓存，则可以将其删除。如果有任何运行正在使用运行缓存，请等待运行完成，或者您可以取消运行。

删除运行缓存会移除资源及其元数据，但不会删除 Amazon S3 中的数据。删除缓存后，您将无法重新连接缓存或将其用于后续运行。

缓存的数据仍保留在 Amazon S3 中供您检查。您可以使用标准 S3 Delete 操作删除旧的缓存数据。或者，创建 Amazon S3 生命周期策略，使不再使用的缓存数据过期。

使用控制台删除运行缓存

在控制台中，按照以下步骤删除运行缓存。

1. 打开 HealthOmics 管理控制台。
2. 在左侧导航窗格中，选择运行缓存。
3. 从 “运行缓存” 表中，选择要删除的运行缓存。
4. 从 “运行缓存” 表格菜单中，选择 “删除”。
5. 在模式对话框中，保存 Amazon S3 缓存数据链接以备将来参考，然后确认要删除运行缓存。

   您可以使用 Amazon S3 链接检查缓存的数据，但不能将数据重新链接到另一个运行缓存。完成检查后删除缓存数据。

使用 CLI 删除运行缓存

使用 C delete-run-cacheLI 命令删除运行缓存。

```
aws omics delete-run-cache \
      --id "my cache id"
```

如果删除成功，您会收到一条没有数据字段的响应。

运行缓存的内容

HealthOmics 在 S3 存储桶中使用以下结构组织运行缓存：

```
s3://{cache.S3location}/{cache.uuid}/runID/taskID/{cacheentry.uuid}/
```

cache.uuid 是缓存的全局唯一 ID。cacheentry.uuid 是缓存任务的全局唯一 uuid。 HealthOmics 将 uuid 分配给缓存和任务。

对于所有工作流引擎，缓存都包含以下文件：

- {cacheentryuuid}.json文件 — HealthOmics 创建此清单文件，其中包含有关缓存的信息，包括缓存中所有项目的列表和缓存版本。
- 任务输出文件-每个任务输出由任务定义的一个或多个文件组成。

对于使用 Nextflow 的工作流程，Nextflow 引擎会在缓存中创建以下其他文件：

- command.out文件 — 此文件包含任务执行标准输出内容。
- .exitcode文件 — 此文件包含任务退出代码（整数）。

> **Note**
>
> 如果要访问运行缓存中的中间任务文件以进行高级故障排除，请在工作流程定义中将这些文件声明为任务输出。

特定于引擎的缓存功能

HealthOmics 尝试在工作流引擎之间提供一致的呼叫缓存实现。根据每个工作流引擎处理特定案例的方式，会有一些差异：

- 下一步
  - 不能保证在不同的 Nextflow 版本之间进行缓存。例如，如果您在 v23.10.0 中运行一个任务，然后在 v24.10.8 中运行相同的任务，则 HealthOmics 可能会认为第二次运行是缓存丢失。
  - 你可以使用 cache false 指令关闭单个任务的缓存。有关此指令的信息，请参阅 Nextflow 规范中的进程。
  - HealthOmics 使用 Nextflow 宽松模式，但不支持深度缓存模式。
  - 如果您在 S3 路径中使用通往任务输入的 glob 模式，则缓存会评估每个单独的 S3 对象。如果添加新对象，则仅 HealthOmics 重新计算使用新对象的任务。
  - HealthOmics 不缓存任务重试次数。此行为与 Nextflow 的默认行为一致。
- WDL
  - HealthOmics 当您使用 WDL 工作流程的开发版本时，支持新的 “目录” 输入类型。对于呼叫缓存，如果目录中的任何对象发生更改，则会 HealthOmics 重新计算所有输入该目录的任务。
  - HealthOmics 支持任务级缓存，但不支持工作流级缓存。
- CWL
  - 从清单中看不到任务的恒定输出。 HealthOmics 将常量输出缓存为中间文件。

使用运行缓存

默认情况下，运行不使用运行缓存。要使用缓存进行运行，请在开始运行时指定运行缓存和运行缓存行为。

运行完成后，您可以使用控制台、 CloudWatch 日志或 API 操作来跟踪缓存命中率或解决缓存问题。有关详细信息，请参阅 跟踪呼叫缓存信息 和 解决呼叫缓存问题。

如果运行中的一个或多个任务生成不确定的输出，我们强烈建议您不要在运行中使用调用缓存，或者选择不缓存这些特定任务。有关更多信息，请参阅 责任共担模式。

> **Note**
>
> 开始运行时，您需要提供 IAM 服务角色。要使用呼叫缓存，服务角色需要访问运行缓存 Amazon S3 位置的权限。有关更多信息，请参阅 的服务角色 AWS HealthOmics。

主题

使用控制台配置带有运行缓存的运行

在控制台中，您可以配置运行缓存，以便在开始运行时进行运行。

1. 打开 HealthOmics 管理控制台。
2. 在左侧导航窗格中，选择 R uns。
3. 在 “运行” 页面上，选择要启动的运行。
4. 选择 “开始运行”，然后按中所述完成 “开始运行” 的步骤 1 和 2 使用控制台开始运行。
5. 在 “开始运行” 的第 3 步中，选择 “选择现有的运行缓存”。
6. 从 “运行缓存 ID” 下拉列表中选择缓存。
7. 要覆盖默认的运行缓存行为，请为运行选择缓存行为。有关更多信息，请参阅 运行缓存行为。
8. 继续执行开始运行的步骤 4。

使用 CLI 配置带有运行缓存的运行

要启动使用运行缓存的运行，请在 start-run CLI 命令中添加 cache-id 参数。或者，使用`cache-behavior`参数来覆盖您为运行缓存配置的默认行为。以下示例仅显示命令的缓存字段：

```
aws omics start-run \
```

```
  ...
--cache-id "xxxxxx"    \
--cache-behavior  CACHE_ALWAYS
```

如果操作成功，您将收到不包含数据字段的响应。

运行缓存的错误案例

在以下情况下，即使在缓存行为设置为 “始终缓存” 的情况下运行，也 HealthOmics 可能无法缓存任务输出。

- 如果在第一个任务成功完成之前运行遇到错误，则没有要导出的缓存输出。
- 如果导出过程失败，则 HealthOmics 不会将任务输出保存到 Amazon S3 缓存位置。
- 如果由于filesystem out of space错误而运行失败，则调用缓存不会保存任何任务输出。
- 如果您取消运行，则呼叫缓存不会保存任何任务输出。
- 如果运行出现运行超时，即使您将运行配置为在失败时使用缓存，调用缓存也不会保存任何任务输出。

跟踪呼叫缓存信息

您可以使用控制台、CLI 或 CloudWatch 日志来跟踪呼叫缓存事件（例如运行缓存命中）。

主题

- 使用控制台跟踪缓存命中
- 使用 CLI 跟踪呼叫缓存
- 使用 CloudWatch 日志跟踪呼叫缓存

使用控制台跟踪缓存命中

在运行的运行详细信息页面中，运行任务表显示每个任务的缓存命中信息。该表还包括指向关联缓存条目的链接。使用以下步骤查看某次运行的缓存命中信息。

1. 打开 HealthOmics 管理控制台。
2. 在左侧导航窗格中，选择 R uns。
3. 在 “运行” 页面上，选择要检查的运行。

4. 在运行详细信息页面上，选择运行任务选项卡以显示任务表。
5. 如果任务有缓存命中，则缓存命中列包含指向 Amazon S3 中运行缓存条目位置的链接。
6. 选择链接以检查运行缓存条目。

使用 CLI 跟踪呼叫缓存

使用 get-run CLI 命令确认运行是否使用了呼叫缓存。

```
aws omics get-run --id 1234567
```

在响应中，如果设置了该cacheId字段，则运行将使用该缓存。

使用 list-run-tasksCLI 命令检索运行中每个缓存任务的缓存数据位置。

```
aws omics list-run-tasks --id 1234567
```

在响应中，如果任务的 cacheHit 字段为真，则 caches3uri 字段将提供该任务的缓存数据位置。

您也可以使用 get-run-taskCLI 命令检索特定任务的缓存数据位置：

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

使用 CloudWatch 日志跟踪呼叫缓存

HealthOmics 在日志组中创建缓存活动/aws/omics/WorkflowLog CloudWatch 日志。<cache_id><cache_uuid>每个运行缓存都有一个日志流：r unCache//。

对于使用呼叫缓存的运行， HealthOmics 会生成以下事件的 CloudWatch 日志条目：

- 创建缓存条目 (CACHE_ENTRY_CREATED)
- 匹配缓存条目 (CACHE_HIT)
- 无法匹配缓存条目 (CACHE_MISS)

有关这些日志的更多信息，请参阅登录 CloudWatch 。

在/aws/omics/WorkflowLog日志组上使用以下 CloudWatch Insights 查询来返回此缓存每次运行的缓存命中数：

```
filter @logStream like 'runCache/<CACHE_ID>/'
 fields @timestamp, @message
 filter logMessage like 'CACHE_HIT'
 parse "run: *," as run
 stats count(*) as cacheHits by run
```

使用以下查询返回每次运行创建的缓存条目数：

```
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
```

共享 HealthOmics 工作流程

作为私有工作流程的所有者，您可以与同一地区的人共享该工作流程。 AWS 账户 要与多个工作流程共享一个工作流程 AWS 账户，可以为同一个工作流程创建多个共享。

作为所有者，您可以通过删除共享来撤消对共享工作流程的访问权限。

> **Note**
>
> HealthOmics 在订阅者账户中运行工作流程时，自动允许共享工作流程访问 Amazon ECR 存储库。您无需为共享工作流程授予额外的存储库访问权限。

当您共享工作流程时，订阅者可以使用任何工作流程版本。如果您需要共享工作流程的版本级访问控制，我们建议您创建单独的工作流程，而不是使用工作流程版本。

主题

- 使用控制台运行共享工作流程
- 使用 API 运行共享工作流程

订阅共享工作流程

要订阅共享工作流程，请按照以下总体步骤接受和使用该工作流程：

1. 使用控制台或 API 接受共享。将您当前的区域设置为与共享请求相同的区域。
   - 要在控制台中查找共享请求，请导航至所有资源共享页面，然后选择与我共享选项卡。
2. 使用控制台或 API 为共享工作流程创建运行。
   - 要在控制台中找到工作流程详细信息页面，请导航到“与我共享”（请参阅步骤 1），然后选择共享工作流程的资源链接。
3. 您可以为工作流程提供自己的输入数据。
4. 共享工作流程在您的中运行 AWS 账户。

作为共享工作流程的订阅者，系统会阻止您执行以下工作流程操作：

- 导出共享工作流程
- 重新运行共享工作流程
  - 您可以为共享工作流程创建新的运行。
- 重新共享工作流程。
- 为工作流程分配标签。
- 删除工作流程。
  - 当您不再需要该工作流程时，可以删除工作流程共享。

有关资源共享中的跨账户资源共享 AWS HealthOmics的更多信息，请参阅。

监控工作流程共享的状态

HealthOmics 工作流共享 EventBridge 的每一次状态更改都会向发送一个事件。如果您想接收有关特定状态更改的通知，请设置一条 EventBridge 规则来监控 Workflow 共享状态更改事件。例如：

- 每次收到工作流程共享请求以及用户撤消工作流程共享时，您都希望收到通知。
- 在您发起工作流程共享请求后，您希望在用户接受或拒绝请求时收到通知。

有关使用事件的详细信息，请参阅 EventBridge 与一起使用 AWS HealthOmics。

使用控制台共享私有工作流程

在控制台中，您可以与工作流程所在区域 AWS 账户 的共享私有工作流程。

共享私有工作流程

1. 打开 HealthOmics 管理控制台。
2. 在左侧导航窗格中，选择私有工作流程。
3. 在 “私有工作流程” 页面的 “工作流程” 表格中，选择要共享的工作流程，然后选择 “共享”。
4. 在共享工作流程页面的共享详细信息面板中，输入共享的描述性名称，然后输入订阅 AWS 账户 者的名称。
5. 选择共享资源。控制台在所有资源共享页面中显示资源共享。

该份额的初始状态为待定。订阅者接受共享后，状态变为活跃。

使用 CLI 共享私有工作流程

使用创建共享 API 操作创建工作流程共享。主要订阅 AWS 账户 者是将获得工作流程访问权限的用户。

```
aws omics create-share \
  --resource-arn "arn:aws:omics:us-west-2:555555555555:workflow/123456" \
  --principal-subscriber "123456789012" \
  --name "my_Share-123"
```

如果创建成功，您将收到包含共享 ID 和状态的响应。

```
{
"shareId": "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a",
"name": "my_Share-123",
"status": "PENDING"
}
```

在订阅者使用 `accept-share` API 操作接受共享之前，共享将保持待处理状态。

中的跨账户资源共享 AWS HealthOmics有关其他 API 用法示例，请参阅。

使用控制台接受共享工作流程

您可以使用控制台接受提供的工作流程共享。确保将控制台设置为与工作流程相同的区域。

1. 打开 HealthOmics 管理控制台。
2. 在左侧导航窗格中，选择所有资源共享，然后选择与我共享选项卡。
3. 从 “与我共享的资源” 表格中，选择工作流程共享，然后选择 “接受”。

接受工作流程后，选择共享工作流的资源链接以查看其详细信息。

使用控制台运行共享工作流程

接受工作流程共享后，您可以开始运行该工作流程。

1. 打开 HealthOmics 管理控制台。
2. 在左侧导航窗格中，选择所有资源共享，然后选择与我共享选项卡。
3. 从 “与我共享的资源” 表中，选择共享工作流程的资源链接。
4. 在工作流程详细信息页面中，选择创建运行。

   控制台打开 “创建运行” 页面，其中预先填充了工作流程类型（共享）和工作流程 ID。
5. 在 “创建运行” 表单中配置其余字段。有关更多信息，请参阅 使用控制台开始运行。

使用 API 运行共享工作流程

使用 get-workflow 检索共享工作流程的 ARN。

```
aws omics get-workflow --id 1234567 \
--workflow-owner-id 55555555555
```

运行工作流程时，请提供工作流程所有者的 AWS 账户 ID 和共享工作流程的 ARN。

```
aws omics start-run --id 1234567 --workflow-owner-id 55555555555 \
--role-arn arn:aws:iam::1234567892012:role/service-role/OmicsWorkflow-20221004T164236 \
--name ArchiveTest --retention-mode REMOVE
```

Ready2Run 中的工作流程 HealthOmics

Ready2Run 工作流程是由第三方发布者发布的预配置工作流程。一些出版商，例如Sentieon Inc，提供基于订阅的工作流程。其他 Ready2Run 工作流程不需要订阅，有些工作流程是开源的，例如 NF-Core 工作流程。

Ready2Run 工作流程非常适合以下场景：

- 您想专注于分析管道输出并生成结果，而无需设置底层基础架构。
- 您想使用既定工作流程来复制结果。
- 作为一名软件开发人员，您希望将您的应用程序直接与 HealthOmics SDK 集成。

HealthOmics 支持 Ready2Run 工作流程的版本控制。对于提供版本的 Ready2Run 工作流程，您可以在开始运行时指定版本名称。

所有 Ready2Run 工作流程都提供可用于故障排除的 CloudWatch 日志，包括日志。

> **Note**
>
> Sentieon Ready2Run 工作流程是基于订阅的。当你在账户中首次运行 Sentieon Ready2Run 工作流程时，Sentieon 会自动为你创建为期两周的评估许可证。 AWS 账户该许可证适用于所有 Sentieon Ready2Run 工作流程。评估期结束后，您可以申请永久许可证或申请延长评估许可证。有关详细信息，请参阅Subscribing to Sentieon Ready2Run workflows。

主题

中可用的 Ready2Run 工作流程 HealthOmics

下表列出了中可用的 Ready2Run 工作流程。 HealthOmics

您可以登录HealthOmics控制台查看有关这些工作流程的详细信息，包括输入参数和工作流程图。有关 Ready2Run 工作流程的定价信息，请参阅定价。HealthOmics

> **Note**
>
> 每个 Ready2Run 工作流程都有最大输入文件大小。这些最大文件大小不可调整。

工作流名称	发布者	需要订阅？	最大输入文件大小 (GiB)	预计运行时间 (HH: MM)
AlphaFold 用于 601-1200 残留物	谷歌 DeepMind	否	1	11:15
AlphaFold 最多可容纳 600 个残留物	谷歌 DeepMind	否	1	7:30
适用于 2x150 的 Bases2Fastq	元素生物科学	否	1000	1:45
适用于 2x300 的 Bases2Fastq	元素生物科学	否	1000	1:30
适用于 2x75 的 Bases2Fastq	元素生物科学	否	500	0:45
ESMFold 最多可容纳 800 个残留物	元研究	否	1	0:15
GATK-BP fq2bam	布罗德研究所	否	64	10:10
GATK-BP Germline bam2vcf 用于 30 倍基因组	布罗德研究所	否	39	2:45

工作流名称	发布者	需要订阅？	最大输入文件大小 (GiB)	预计运行时间 (HH: MM)
GATK-BP Germline fq2vcf 用于 30 倍基因组	布罗德研究所	否	64	12:30
GATK-BP Somatic WES bam2vcf	布罗德研究所	否	86	1:30
NVIDIA Parabricks BAM2 FQ2 BAM WGS 最高可达 30 倍	英伟达公司	否	80	1:39
NVIDIA Parabricks BAM2 FQ2 BAM WGS 最高可达 50 倍	英伟达公司	否	120	2:45
NVIDIA Parabricks BAM2 FQ2 BAM WGS 最高可达 5 倍	英伟达公司	否	20	0:18
NVIDIA Parabricks FQ2 BAM WGS 最高可达 30 倍	英伟达公司	否	71	1:00
NVIDIA Parabricks FQ2 BAM WGS 最高可达 50 倍	英伟达公司	否	137	1:45

工作流名称	发布者	需要订阅？	最大输入文件大小 (GiB)	预计运行时间 (HH: MM)
NVIDIA Parabricks FQ2 BAM WGS 最高可达 5 倍	英伟达公司	否	13	0:15
NVIDIA Parabricks Germline DeepVariant WGS 最高可达 30 倍	英伟达公司	否	71	2:00
NVIDIA Parabricks Germline DeepVariant WGS 最高可达 50 倍	英伟达公司	否	137	3:30
NVIDIA Parabricks Germline DeepVariant WGS 售价高达 5 倍	英伟达公司	否	12	0:30
NVIDIA Parabricks Germline HaplotypeCaller WGS 最高可达 30 倍	英伟达公司	否	71	1:15

工作流名称	发布者	需要订阅？	最大输入文件大小 (GiB)	预计运行时间 (HH: MM)
NVIDIA Parabricks Germline HaplotypeCaller WGS 最高可达 50 倍	英伟达公司	否	137	2:00
NVIDIA Parabricks Germline HaplotypeCaller WGS 售价高达 5 倍	英伟达公司	否	13	0:15
NVIDIA Parabricks Somatic Mutect2 WGS 最高可达 50 倍	英伟达公司	否	196	0:45
sc wit RNAseq h Kallisto BUStools	NF-Core	否	119	1:30
sc RNAseq 配三文鱼 Alevin-fry	NF-Core	否	119	2:30
sc w RNAseq ith STARsolo	NF-Core	否	119	2:30
Sentieon Germline BAM WES 最高可达 300 倍	Sentieon, Inc.	是	9	1:00

工作流名称	发布者	需要订阅？	最大输入文件大小 (GiB)	预计运行时间 (HH: MM)
Sentieon Germline BAM WGS 最高可达 32 倍	Sentieon, Inc.	是	18	1:30
Sentieon Germline FASTQ WES 最高可达 100 倍	Sentieon, Inc.	是	5	0:45
Sentieon Germline FASTQ WES 最高可达 300 倍	Sentieon, Inc.	是	26	2:00
Sentieon Germline FASTQ WGS 最高可达 32 倍	Sentieon, Inc.	是	51	3:30
安大略省的 Sentieon LongRead	Sentieon, Inc.	是	25	1:30
Sentieon for LongRead PacBio HiFi	Sentieon, Inc.	是	58	4:00
Sentieon Somatic WES	Sentieon, Inc.	是	50	2:30
Sentieon Somatic WGS	Sentieon, Inc.	是	113	4:30

工作流名称	发布者	需要订阅？	最大输入文件大小 (GiB)	预计运行时间 (HH: MM)
Ultima Genomic DeepVariant s 最高可达 40 倍	Ultima Genomics	否	91	1:55

当您使用 Ready2Run 工作流程时，您的工作流程是预先配置的，无法编辑。与私有工作流程相比，Ready2Run 工作流程不支持以下内容：

- 增加最大输入文件大小
- 更改计算资源或运行存储
- 更改工作流程定义或容器
- 向跑步组中添加跑步
- 共享工作流程

如果发布者已共享了 Ready2Run 工作流程 GitHub，则可以基于 Ready2Run 工作流程创建自己的私有工作流程。下表提供了每个发布者 GitHub 的工作流程链接。

发布者	工作流程已开启 GitHub
谷歌 DeepMind、元研究	蛋白质折叠工作流程
元素生物科学	如需信息，请联系元素生物科学
布罗德研究所	GATK 工作流程
英伟达公司	Parabricks 工作流程
nf-core	NF 核心工作流程
Sentieon	Sentieon 工作流程
Ultima Genomics	Ultima Genomics 工作流程

订阅 Sentieon Ready2Run 工作流程

Sentieon Ready2Run 工作流程是基于订阅的。当你在账户中首次运行 Sentieon Ready2Run 工作流程时，Sentieon 会自动为你创建为期两周的评估许可证。 AWS 账户该许可证适用于所有 Sentieon Ready2Run 工作流程。评估期结束后，您可以申请永久许可证或申请延长评估许可证。

请按照以下步骤订阅 Sentieon Ready2Run 工作流程：

- 按照以下说明查找您的 AWS 规范用户 ID。
- 向 Sentieon 支持小组 (support@sentieon.com) 发送电子邮件申请软件许可证。在电子邮件中提供您的 AWS 规范用户 ID。

使用控制台启动 HealthOmics Ready2Run 工作流程

在控制台中使用 Ready2Run 工作流程与使用私有工作流程类似。一个关键的区别是，工作流程发布者提供了示例数据，因此您无需创建自己的数据即可试用工作流程。

在控制台中使用 Ready2Run 工作流程

1. 打开 HealthOmics 管理控制台。
2. 在左侧导航窗格中，选择 Ready2Run 工作流程。
3. 在 Ready2Run 工作流程页面上，选择要使用的工作流程。控制台将打开该工作流程的详细信息页面。
4. 详细信息选项卡列出了诸如名称、每次运行的标价、描述、工作流语言类型、运行存储容量、状态、创建日期和参数等信息，以及带描述的参数。详细信息选项卡还会告诉您该工作流程是否需要订阅。
5. 要使用工作流程，请选择 “创建”、“运行”
6. 在 “指定运行详细信息” 页面中，输入运行名称。或者，您可以指定工作流程版本。您也可以为运行添加运行优先级。
7. 为运行输出输入或选择一个 Amazon S3 位置。
8. 对于运行元数据保留模式，选择是保留还是移除运行元数据。
9. 在服务角色面板中，选择是使用现有服务角色还是创建新服务角色。
10. （可选）添加标签以帮助识别和管理您的跑步。
11. 选择下一步。
12. 在 “添加参数” 页面中，选择一个选项来添加运行参数值：

- 从 Amazon S3 的某个位置选择一个参数文件（JSON 格式）。
- 从本地驱动器中选择一个参数文件（JSON 格式）。
- 手动输入参数值。
- 使用工作流程发布者提供的 Ready2Run 示例数据运行工作流程。

13. 如果您上传 JSON 文件，则控制台会解析该文件并执行内联验证。然后，您可以根据需要手动更新参数的值。
14. 选择下一步。
15. 查看您的输入，然后选择 “开始运行”。

使用 API 启动 HealthOmics Ready2Run 工作流程

对于 Ready2Run 工作流程和私有工作流程，大多数 API 操作的行为方式类似。

要返回可用的 Ready2Run 工作流程列表，请使用参数设置为 RUN 的**type**列表工作流程。 READY2

```
aws omics list-workflows --type READY2RUN
```

从列表工作流响应中确定要运行的工作流程后，您可以使用带--id参数的 get-work flow 来获取更多详细信息。

```
aws omics get-workflow --type READY2RUN --id workflow id
```

要运行 Ready2Run 工作流程，您可以使用启动运行 API 操作，并将工作流类型参数设置为，如以下示例所示 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
```

要指定工作流程版本，请使用工作流版本参数，如本示例所示。

```
aws-omics start-run \
```

```
  --workflow-type READY2RUN \
  ...
  --version-name '3.0.0'
```

要监控您的运行情况，您可以使用 get-run API 操作，如图所示。

```
aws-omics get-run \
  --id run id
```

HealthOmics 存储

使用 HealthOmics 存储以低成本高效地存储、检索、组织和共享基因组学数据。 HealthOmics 存储了解不同数据对象之间的关系，因此您可以定义哪些读取集源自相同的源数据。这为您提供了数据来源。

存储在ACTIVE状态下的数据可以立即检索。30 天或更长时间未被访问的数据将以ARCHIVE状态存储。要访问存档的数据，您可以通过 API 操作或控制台将其重新激活。

HealthOmics 序列存储旨在保持文件的内容完整性。但是，由于在活动分层和存档分层期间会进行压缩，因此无法保留导入数据文件和导出文件的按位等效性。

在摄取期间， HealthOmics 生成实体标签 HealthOmics ETag，或，以便验证数据文件的内容完整性。测序部分在读取集的 ETag 源级别被识别和捕获。 ETag 计算结果不会改变实际文件或基因组数据。创建读取集后，在读取集源的整个生命周期中 ETag 不应发生变化。这意味着重新导入相同的文件会导致计算出相同的 ETag 值。

主题

HealthOmics ETags 和数据来源

HealthOmics ETag （实体标签）是序列存储中摄取内容的哈希值。这简化了数据检索和处理，同时保持了摄取的数据文件的内容完整性。 ETag 反映的是对象语义内容的变化，而不是其元数据的变化。指定的读取集类型和算法决定 ETag 如何计算。 ETag 计算结果不会改变实际文件或基因组数据。当读取集的文件类型架构允许时，序列存储会更新与数据来源相关的字段。

文件具有按位标识和语义标识。按位标识意味着文件的各个位是相同的，而语义标识意味着文件的内容是相同的。语义标识可以捕获文件的内容完整性，因此可以抵御元数据更改和压缩更改。

HealthOmics 序列存储中的读取集在对象的整个生命 compression/decompression 周期中经历周期和数据来源跟踪。在此处理过程中，载入文件的按位标识可能会发生变化，并且预计每次激活文件时都会发生变化；但是，文件的语义标识会保持不变。语义标识被捕获为 HealthOmics 实体标签 ETag ，或者在序列存储摄取期间计算出来并作为读取集元数据使用。

当读取集的文件类型架构允许时，序列存储更新字段将与数据来源相关联。对于 uBam、BAM 和 CRAM 文件，标题中会添加一个新的`@CO`或`Comment`标签。注释包含序列存储 ID 和摄取时间戳。

亚马逊 S3 ETags

使用 Amazon S3 URI 访问文件时，Amazon S3 API 操作也可能返回亚马逊 S3 ETag 和校验和值。Amazon S3 ETag 和校验和值之所以与不同， HealthOmics ETags 是因为它们代表文件的按位标识。要了解有关描述性元数据和对象的更多信息，请参阅 Amazon S3 对象 API 文档。Amazon S3 的 ETag 值可能会随着读取集的每个激活周期而变化，您可以使用它们来验证文件的读取。但是，不要缓存 Amazon S3 ETag 值以在文件生命周期中用于文件身份验证，因为它们不会保持一致。相比之下，在读取集的整个生命周期中都 HealthOmics ETag 保持一致。

如何 HealthOmics 计算 ETags

ETag 是根据提取的文件内容的哈希值生成的。 MD5up 默认情况下， ETag 算法系列设置为，但在创建序列存储期间可以对其进行不同的配置。计算 ETag 完毕后，算法和计算出的哈希值将添加到读取集合中。支持的文件类型 MD5 算法如下。

- FASTQ_ MD5up — 计算未压缩、完整 FASTQ 读取集源的 MD5哈希值。
- BAM_ MD5up — 根据链接的引用（如果有）计算 SAM 中表示的未压缩 BAM 或 uBam 读取集源的对齐部分的 MD5 哈希值。
- CRAM_ MD5up — 根据链接的 MD5 引用，计算 SAM 中表示的未压缩 CRAM 读取集源的对齐部分的哈希值。

> **Note**
>
> MD5 众所周知，哈希很容易发生冲突。因此， ETag 如果两个不同的文件是为了利用已知的碰撞而制造的，则它们可能具有相同的效果。

该 SHA256 系列支持以下算法。算法的计算方法如下：

- FASTQ_ SHA256up — 计算未压缩、完整 FASTQ 读取集源的 SHA-256 哈希值。

- BAM_ SHA256up — 根据链接的引用（如果有）计算 SAM 中表示的未压缩 BAM 或 uBam 读取集源的对齐部分的 SHA-256 哈希值。
- CRAM_ SHA256up — 根据链接的引用，计算 SAM 中表示的未压缩 CRAM 读取集源的对齐部分的 SHA-256 哈希值。

该 SHA512 系列支持以下算法。算法的计算方法如下：

- FASTQ_ SHA512up — 计算未压缩、完整 FASTQ 读取集源的 SHA-512 哈希值。
- BAM_ SHA512up — 根据链接的引用（如果有）计算 SAM 中表示的未压缩 BAM 或 uBam 读取集源的对齐部分的 SHA-512 哈希值。
- CRAM_ SHA512up — 根据链接的引用，计算 SAM 中表示的未压缩 CRAM 读取集源的对齐部分的 SHA-512 哈希值。

创建 HealthOmics 参考存储库

中的参考存储 HealthOmics 是用于存储参考基因组的数据存储。您可以在每个 AWS 账户 区域中拥有一个参考资料库。您可以使用控制台或 CLI 创建参考存储。

主题

使用控制台创建参考库

创建参考存储

1. 打开 HealthOmics 管理控制台。
2. 在左侧导航窗格中，选择 “开始使用” HealthOmics。
3. 从 “基因组学” 数据存储选项中选择 “参考基因组”。
4. 您可以选择先前导入的参考基因组，也可以导入新的参考基因组。如果您尚未导入参考基因组，请选择右上角的导入参考基因组。
5. 在 “创建参考基因组导入作业” 页面上，选择 “快速创建” 或 “手动创建” 选项来创建参考存储库，然后提供以下信息。

- 参考基因组名称-此存储的唯一名称。
- 描述（可选）-此参考库的描述。
- IAM 角色-选择有权访问您的参考基因组的角色。
- 来自 Amazon S3 的参考-在 Amazon S3 存储桶中选择您的参考序列文件。
- 标签（可选）-为此参考商店提供最多 50 个标签。

使用 CLI 创建参考存储库

以下示例向您展示了如何使用创建参考存储库 AWS CLI。每个 AWS 地区可以有一个参考库。

参考存储支持存储扩展名
为.fasta、、、、、、、、、.fa、.fas、.fsa、.faa.fna.ffn.frn.mpfa.seq、.txt的 FASTA 文件。还支持这些扩展的bgzip版本。

在以下示例中，*reference store name*使用您为参考商店选择的名称替换。

```
aws omics create-reference-store --name "reference store name"
```

您会收到一个 JSON 响应，其中包含参考存储库 ID 和名称、ARN 以及创建参考存储的时间戳。

```
{
    "id": "3242349265",
    "arn": "arn:aws:omics:us-west-2:555555555555:referenceStore/3242349265",
    "name": "MyReferenceStore",
    "creationTime": "2022-07-01T20:58:42.878Z"
}
```

您可以在其他 AWS CLI 命令中使用参考存储库 ID。您可以使用list-reference-stores命令检索与您的账户 IDs 关联的参考商店列表，如以下示例所示。

```
aws omics list-reference-stores
```

作为回应，您将收到新创建的参考商店的名称。

```
{
    "referenceStores": [
        {
              "id": "3242349265",
```

```
            "arn": "arn:aws:omics:us-west-2:555555555555:referenceStore/3242349265",
            "name": "MyReferenceStore",
           "creationTime": "2022-07-01T20:58:42.878Z"
        }
    ]
}
```

创建参考存储后，您可以创建导入任务以将基因组参考文件加载到其中。为此，您必须使用或创建 IAM 角色来访问数据。以下是示例策略。

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/*"
            ]
        }
    ]
}
```

您还必须有类似于以下示例的信任策略。

JSON

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

```
                "Service": [
                    "omics.amazonaws.com"
                ]
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

您现在可以导入参考基因组了。此示例使用基因组参考联盟 Human Build 38 (hg38)，该联盟是开放访问的，可从开放数据注册处获得。 AWS托管此数据的存储桶位于美国东部（俄亥俄州）。要在其他 AWS 区域使用存储桶，您可以将数据复制到您所在地区托管的 Amazon S3 存储桶。使用以下 AWS CLI 命令将基因组复制到您的 Amazon S3 存储桶。

```
aws s3 cp s3://broad-references/hg38/v0/Homo_sapiens_assembly38.fasta s3://amzn-s3-demo-bucket
```

然后，您就可以开始导入任务了。用您自己的输入替换*reference store ID*、*role ARN*、和*source file path*。

```
aws omics start-reference-import-job --reference-store-id reference store ID --role-arn role ARN --sources source file path
```

导入数据后，您将收到以下 JSON 格式的响应。

```
{
        "id": "7252016478",
        "referenceStoreId": "3242349265",
        "roleArn": "arn:aws:iam::111122223333:role/OmicsReferenceImport",
        "status": "CREATED",
        "creationTime": "2022-07-01T21:15:13.727Z"
}
```

您可以使用以下命令监视作业的状态。在以下示例中，将*reference store ID*和*job ID*替换为您的参考商店 ID 和您想进一步了解的任务 ID。

```
aws omics get-reference-import-job --reference-store-id reference store ID --id job ID
```

作为回应，您会收到一条回复，其中包含该参考库的详细信息及其状态。

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

您还可以通过列出您的参考文献并根据参考名称对其进行筛选来查找已导入的参考文献。*reference store ID*替换为您的参考商店 ID，然后添加可选筛选条件以缩小列表范围。

```
aws omics list-references --reference-store-id reference store ID --filter
 name=MyReference
```

作为回应，您会收到以下信息。

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

要了解有关参考元数据的更多信息，请使用 get-reference-metadataAPI 操作。在以下示例中，*reference store ID*替换为您的参考商店编号和*reference ID*您想进一步了解的参考编码。

```
aws omics get-reference-metadata --reference-store-id reference store ID --id reference
 ID
```

作为回应，您会收到以下信息。

```
{
    "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
        }
    }
}
```

您也可以使用 get- reference 下载部分参考文件。在以下示例中，*reference store ID*替换为您的参考商店编号和*reference ID*您要从中下载的参考编码。

```
aws omics get-reference --reference-store-id reference store ID --id reference ID --
part-number 1 outfile.fa
```

创建 HealthOmics 序列存储

HealthOmics 序列存储支持以FASTQ（仅限 gzip）和的未对齐格式存储基因组文件。uBAM它还支持BAM和的对齐格式CRAM。

导入的文件以读取集的形式存储。您可以为读取集添加标签，并使用 IAM 策略来控制对读取集的访问权限。对齐的读取集需要参考基因组来对齐基因组序列，但对于未对齐的读取集，它是可选的。

要存储读取集，请先创建序列存储。创建序列存储时，您可以指定一个可选的 Amazon S3 存储桶作为备用位置以及存储 S3 访问日志的位置。备用位置用于存储在直接上传期间未能创建读取集的任何文件。备用位置可用于 2023 年 5 月 15 日之后创建的序列存储。您可以在创建序列存储时指定后备位置。

您最多可以指定五个读取集标签密钥。当您使用与其中一个密钥匹配的标签密钥创建或更新读取集时，读取集标签会传播到相应的 Amazon S3 对象。默认情况下，由创建的系统标签会 HealthOmics 被传播。

主题

使用控制台创建序列存储

创建序列存储

1. 打开 HealthOmics 管理控制台。
2. 在左侧导航窗格中，选择序列存储。
3. 在 “创建序列存储” 页面上，提供以下信息
   - 序列存储名称-此存储的唯一名称。
   - 描述（可选）-此序列存储的描述。
4. 对于 S3 中的备用位置，请指定 Amazon S3 位置。 HealthOmics 使用备用位置来存储在直接上传期间未能创建读取集的所有文件。您需要向该 HealthOmics 服务授予对 Amazon S3 备用位置的写入权限。有关策略示例，请参阅 配置备用位置。

   备用位置不适用于 2023 年 5 月 16 日之前创建的序列存储库。

5. （可选）对于用于 S3 传播的 Read set 标签键，您最多可以输入五个读取集密钥，从读取集传播到底层 S3 对象。通过将标签从读取集传播到 S3 对象，您可以根据标签授予 S3 访问权限，允许 and/or 最终用户通过 Amazon S3 getObjectTagging API 操作查看传播的标签。

   a. 在文本框中输入一个键值。控制台会创建一个新的文本框来添加下一个密钥。

   b. （可选）选择 “移除” 以删除所有密钥。

6. 在 “数据加密” 下，选择是否要让数据加密由客户管理的 CMK 拥有和管理， AWS 还是要使用客户托管的 CMK。
7. （可选）在 “S3 数据访问” 下，选择是否创建新的角色和策略以通过 Amazon S3 访问序列存储。
8. （可选）对于 S3 访问日志，请选择`Enabled`是否希望 Amazon S3 收集访问日志记录。

   对于 S3 中的访问日志位置，请指定用于存储日志的 Amazon S3 位置。只有启用了 S3 访问日志记录后，此字段才可见。

9. 标签（可选）-为此序列存储提供最多 50 个标签。这些标签与读取集 import/tag 更新期间设置的读取集标签是分开的

创建商店后，就可以开始使用了导入基因组文件。

使用 CLI 创建序列存储

在以下示例中，*`sequence store name`*使用您为序列存储选择的名称替换。

```
aws omics create-sequence-store --name sequence store name --fallback-location "s3://
amzn-s3-demo-bucket"
```

您将收到以下 JSON 格式的响应，其中包括您新创建的序列存储的 ID 号。

```
{
    "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"
}
```

您还可以使用list-sequence-stores命令查看与您的账户关联的所有序列存储，如下所示。

```
aws omics list-sequence-stores
```

您会收到以下回复。

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

您可以使用序列存储的 ID get-sequence-store来了解有关序列存储的更多信息，如以下示例所示：

```
aws omics get-sequence-store --id sequence store ID
```

您会收到以下回复：

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

创建后，还可以更新多个商店参数。这可以通过控制台或 API `updateSequenceStore` 操作来完成。

更新序列存储

要更新序列存储，请执行以下步骤：

1. 打开 HealthOmics 管理控制台。
2. 在左侧导航窗格中，选择序列存储。
3. 选择要更新的序列存储。
4. 在 “详细信息” 面板中，选择 “编辑”。
5. 在编辑详细信息页面上，您可以更新以下字段：

   - 序列存储名称-此存储的唯一名称。
   - 描述-此序列存储的描述。
   - 在 S3 中的备用位置，请指定 Amazon S3 的位置。 HealthOmics 使用备用位置来存储在直接上传期间未能创建读取集的所有文件。
   - 读取 S3 传播的设置标签密钥您最多可以输入五个读取集密钥以传播到 Amazon S3。
   - （可选）对于 S3 访问日志，请选择`Enabled`是否希望 Amazon S3 收集访问日志记录。

     对于 S3 中的访问日志位置，请指定用于存储日志的 Amazon S3 位置。只有启用了 S3 访问日志记录后，此字段才可见。
   - 标签（可选）-为此序列存储提供最多 50 个标签。

更新序列存储的读取集标签

要更新序列存储的读取集标签或其他字段，请执行以下步骤：

1. 打开 HealthOmics 管理控制台。
2. 在左侧导航窗格中，选择序列存储。
3. 选择要更新的序列存储。
4. 选择详细信息选项卡。
5. 选择编辑。

6. 根据需要添加新的读取集标签或删除现有标签。
7. 根据需要更新名称、描述、备用位置或 S3 数据访问权限。
8. 选择保存更改。

导入基因组文件

要将基因组文件导入序列存储，请执行以下步骤：

导入基因组学文件

1. 打开 HealthOmics 管理控制台。
2. 在左侧导航窗格中，选择序列存储。
3. 在 Se quence 存储页面上，选择要将文件导入到的序列存储。
4. 在单个序列存储页面上，选择导入基因组文件。
5. 在 “指定导入详情” 页面上，提供以下信息
   - IAM 角色-可以访问 Amazon S3 上的基因组文件的 IAM 角色。
   - 参考基因组-该基因组学数据的参考基因组。
6. 在 “指定导入清单” 页面上，指定以下信息清单文件。清单文件是一个 JSON 或 YAML 文件，用于描述基因组学数据的基本信息。有关清单文件的信息，请参阅将读取集导入 HealthOmics 序列存储。
7. 单击 “创建导入任务”。

删除 HealthOmics 引用和序列存储

引用和序列存储均可删除。仅当序列存储不包含读取集时才能将其删除，并且只有在引用存储不包含引用时才能将其删除。删除序列或参考存储也会删除与该存储关联的所有标签。

以下示例说明如何使用删除参考资料库 AWS CLI。如果操作成功，您将不会收到回复。在以下示例中，*reference store ID*使用您的参考商店编号替换。

```
aws omics delete-reference-store --id reference store ID
```

以下示例向您展示如何删除序列存储。如果操作成功，您将不会收到回复。在以下示例中，*sequence store ID*替换为您的序列存储 ID。

```
aws omics delete-sequence-store --id sequence store ID
```

您也可以删除参考文献库中的参考文献，如以下示例所示。只有在未在读取集、变体存储或注释存储中使用引用时，才能将其删除。在以下示例中，*reference store ID*替换为您的参考商店 ID，然后*reference ID*替换为要删除的参考文献的 ID。

```
aws omics delete-reference  --id reference ID --reference-store-id reference store ID
```

将读取集导入 HealthOmics 序列存储

创建序列存储后，创建导入任务以将读取集上传到数据存储中。您可以从 Amazon S3 存储桶上传文件，也可以使用同步 API 操作直接上传。您的 Amazon S3 存储桶必须与您的序列存储位于同一区域。

您可以将对齐和未对齐读取集的任意组合上传到序列存储中，但是，如果导入中的任何读取集是对齐的，则必须包括参考基因组。

您可以重复使用用于创建参考存储的 IAM 访问策略。

以下主题描述了将读取集导入序列存储然后获取有关导入数据信息的主要步骤。

主题

将文件上传到亚马逊 S3

以下示例显示了如何将文件移动到您的 Amazon S3 存储桶中。

```
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
```

本示例BAM中CRAM使用的样本需要不同的基因组参考文献，Hg19以及Hg38。要了解更多信息或访问这些参考文献，请参阅开放数据注册表中的广泛基因组参考文献 AWS。

创建清单文件

您还必须以 JSON 格式创建清单文件来对导入任务进行建模import.json（参见以下示例）。如果您在控制台中创建序列存储，则无需指定sequenceStoreId或roleARN，因此清单文件以sources输入开头。

API manifest

以下示例使用 API 导入三个读取集：一个FASTQBAM、一个和一个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

此示例代码用于使用控制台导入单个读取集。

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

或者，您可以上传 YAML 格式的清单文件。

启动导入任务

要启动导入作业，请使用以下 AWS CLI 命令。

```
aws omics start-read-set-import-job --cli-input-json file://import.json
```

您会收到以下响应，表示成功创建了就业机会。

```
{
  "id": "3660451514",
  "sequenceStoreId": "3936421177",
  "roleArn": "arn:aws:iam::111122223333:role/OmicsImport",
  "status": "CREATED",
  "creationTime": "2022-07-13T22:14:59.309Z"
}
```

监控导入作业

导入任务启动后，您可以使用以下命令监控其进度。在以下示例中，*sequence store id*替换为您的序列存储 ID，然后*job import ID*替换为导入 ID。

```
aws omics get-read-set-import-job --sequence-store-id sequence store id --id job import
 ID
```

下图显示了与指定序列存储 ID 关联的所有导入任务的状态。

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

查找导入的序列文件

任务完成后，您可以使用 list-read-setsAPI 操作来查找导入的序列文件。在以下示例中，*sequence store id*替换为您的序列存储 ID。

```
aws omics list-read-sets --sequence-store-id sequence store id
```

您会收到以下回复。


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

获取有关阅读集的详细信息

要查看有关读取集的更多详细信息，请使用 GetReadSetMetadataAPI 操作。在以下示例中，*sequence store id*替换为您的序列存储 ID，然后*read set id*替换为您的读取集 ID。

```
aws omics get-read-set-metadata --sequence-store-id sequence store id --id read set id
```

您会收到以下回复。

```
{
"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"
}
```


下载读取集数据文件

您可以使用 Amazon S3 GetObject API 操作访问活动读取集的对象。该对象的 URI 将在 GetReadSetMetadataAPI 响应中返回。有关更多信息，请参阅 使用 Amazon S3 访问 HealthOmics 读取集 URIs。

或者，也可以使用 HealthOmics GetReadSet API 操作。您可以使用GetReadSet通过下载各个部分来并行下载。这些部分与 Amazon S3 部件类似。以下是如何从读取集下载第 1 部分的示例。在以下示例中，*sequence store id*替换为您的序列存储 ID，然后*read set id*替换为您的读取集 ID。

```
aws omics get-read-set --sequence-store-id sequence store id --id read set id  --part-
number 1 outfile.bam
```

您也可以使用 HealthOmics 传输管理器下载文件以 HealthOmics 供参考或读取。您可以在此处下载 HealthOmics 转接管理器。有关使用和设置传输管理器的更多信息，请参阅此GitHub存储库。

直接上传到 HealthOmics 序列存储

我们建议您使用 HealthOmics 传输管理器将文件添加到序列存储中。有关使用传输管理器的更多信息，请参阅此GitHub存储库。您也可以通过直接上传 API 操作将读取集直接上传到序列存储。

直接上传读取集首先处于PROCESSING_UPLOAD状态。这意味着文件段当前正在上传，您可以访问读取集元数据。上传分段并验证校验和后，读取集将变为ACTIVE和行为与导入的读取集相同。

如果直接上传失败，则读取集状态将显示为UPLOAD_FAILED。您可以将 Amazon S3 存储桶配置为上传失败的文件的备用位置。备用位置可用于 2023 年 5 月 15 日之后创建的序列存储。

主题

- 使用直接上传到序列存储库 AWS CLI
- 配置备用位置

使用直接上传到序列存储库 AWS CLI

首先，开始分段上传。您可以使用来执行此操作 AWS CLI，如以下示例所示。

使用 AWS CLI 命令直接上传

1. 通过分隔数据来创建各个部分，如以下示例所示。

   ```
   split -b 100MiB SRR233106_1.filt.fastq.gz source1_part_
   ```

2. 将源文件分段后，创建分段读取集上传，如以下示例所示。将和其他参数替换*sequence store ID*为您的序列存储 ID 和其他值。

```
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"
```

你会在响应中获得uploadID和其他元数据。使用uploadID进行上传过程的下一步操作。

```
{
"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. 将您的阅读集添加到上传内容中。如果您的文件足够小，则只需执行一次此步骤即可。对于较大的文件，您可以对文件的每个部分执行此步骤。如果您使用之前使用的分段号上传新分段，则它会覆盖之前上传的分段。

   在以下示例中，将*sequence store IDupload ID*、和其他参数替换为您的值。

   ```
   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
   ```

   响应是一个 ID，您可以使用它来验证上传的文件是否与您想要的文件匹配。

   ```
   {
   "checksum": "984979b9928ae8d8622286c4a9cd8e99d964a22d59ed0f5722e1733eb280e635"
   ```

```
}
```

4. 如有必要，请继续上传文件的各个部分。要验证您的读取集是否已上传，请使用 list-read-set-upload-par ts API 操作，如下所示。在以下示例中，将*sequence store ID* 、和*upload ID*，替换为*part source*您自己的输入。

```
aws omics list-read-set-upload-parts \
 --sequence-store-id sequence store ID \
 --upload-id upload ID \
 --part-source SOURCE1
```

响应返回读取集的数量、大小和最近更新时间的时间戳。

```
{
"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. 要查看所有活跃的分段读取集上传，请使用 up list-multipart-read-setloads，如下所示。*sequence store ID*替换为您自己的序列存储的 ID。

```
aws omics list-multipart-read-set-uploads --sequence-store-id
             sequence store ID
```

此 API 仅返回正在进行的分段读取集上传。在提取的读取集之后ACTIVE，或者如果上传失败，则不会在对 uploads API 的响应中返回上list-multipart-read-set传。要查看活跃的读取集，请使用list-read-setsAPI。list-multipart-read-set上传的响应示例如下所示。

```
{
"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. 上传文件的所有部分后，使用 complete-multipart-read-set- upload 结束上传过程，如以下示例所示。用您自己的值替换*sequence store ID*零件的*upload ID*、和参数。

```
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"}]'
```

complete-multipart-read-set-uploa d 的响应是您导入的读 IDs 取集的读取集。

```
{
"readSetId": "0000000001"
}
```

7. 要停止上传，请使用带有abort-multipart-read-set上传 ID 的- upload 来结束上传过程。*upload ID*用您自己的参数值替换*sequence store ID*和。

```
aws omics abort-multipart-read-set-upload \
--sequence-store-id sequence store ID \
--upload-id upload ID
```

8. 上传完成后，使用从读取集中检索数据 get-read-set，如下所示。如果上传仍在处理中，则get-read-set返回有限的元数据，并且生成的索引文件不可用。*sequence store ID*用您自己的输入替换和其他参数。

```
aws omics get-read-set
 --sequence-store-id sequence store ID \
 --id read set ID \
 --file SOURCE1 \
```

```
 --part-number 1 myfile.fastq.gz
```

9. 要检查元数据，包括上传状态，请使用 get-read-set-metadataAPI 操作。

```
aws omics get-read-set-metadata --sequence-store-id sequence store ID --id read set
 ID
```

响应包含元数据详细信息，例如文件类型、引用 ARN、文件数量和序列长度。它还包括状态。可能的状态是PROCESSING_UPLOADACTIVE、和。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"
}
```

配置备用位置

创建或更新序列存储时，您可以将 Amazon S3 存储桶配置为上传失败的文件的备用位置。这些读取集的文件部分被传输到备用位置。备用位置可用于 2023 年 5 月 15 日之后创建的序列存储。

创建 Amazon S3 存储桶策略以授予对 Amazon S3 备用位置的 HealthOmics 写入权限，如以下示例所示：

```
{
    "Effect": "Allow",
    "Principal": {
        "Service": "omics.amazonaws.com"
    },
    "Action": "s3:PutObject",
    "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
}
```

如果用于备用或访问日志的 Amazon S3 存储桶使用客户托管密钥，请向密钥策略添加以下权限：

```
 {
    "Sid": "Allow use of key",
    "Effect": "Allow",
    "Principal": {
        "Service": "omics.amazonaws.com"
    },
    "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey*"
    ],
    "Resource": "*"
}
```

将 HealthOmics 读取集导出到 Amazon S3 存储桶

您可以将读取集作为批量导出任务导出到 Amazon S3 存储桶。为此，请先创建一个具有存储桶写入权限的 IAM 策略，类似于以下 IAM 策略示例。

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"
  }
]
}
```

IAM 策略到位后，开始您的读取集导出任务。以下示例向您展示了如何使用 start-read-set-export-job API 操作来执行此操作。在以下示例中，将所有参数（例如、*sequence store IDdestinationrole ARNsources*、和）替换为您的输入。

```
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
```

您会收到以下响应，其中包含有关源序列存储和目标 Amazon S3 存储桶的信息。

```
{
"id": <job-id>,
"sequenceStoreId": <sequence-store-id>,
"destination": <destination-s3-uri>,
"status": "SUBMITTED",
"creationTime": "2022-10-22T01:33:38.079000+00:00"
}
```

作业启动后，您可以使用 get-read-set-export-job API 操作确定其状态，如下所示。将*sequence store ID*和*job ID*分别替换为您的序列存储 ID 和作业 ID。

```
aws omics get-read-set-export-job --id job-id --sequence-store-id sequence store ID
```

您可以使用 list-read-set-export-jobs API 操作查看为序列存储初始化的所有导出作业，如下所示。*sequence store ID*用您的序列存储 ID 替换。

```
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"
  }
]
}
```

除了导出读取集外，您还可以使用 Amazon S3 访问权限共享读取集 URIs。要了解更多信息，请参阅使用 Amazon S3 访问 HealthOmics 读取集 URIs。

使用 Amazon S3 访问 HealthOmics 读取集 URIs

您可以使用 Amazon S3 URI 路径来访问您的活动序列存储读取集。

通过 Amazon S3 URI 路径，您可以使用 Amazon S3 操作列出、共享和下载您的读取集。鉴于许多行业工具已经构建为可从 S3 读取，因此通过 S3 进行访问可以 APIs 加快协作和工具集成。此外，您可以 APIs 与其他账户共享对 S3 的访问权限，并提供对数据的跨区域读取权限。

HealthOmics 不支持 Amazon S3 URI 访问存档的读取集。当您激活读取集时，它每次都会恢复到相同的 URI 路径。

将数据加载到 HealthOmics 商店后，由于 Amazon S3 URI 基于 Amazon S3 接入点，因此您可以直接与读取 Amazon S3 的行业标准工具集成 URIs，例如：

- 视觉分析应用程序，例如综合基因组学查看器 (IGV) 或加州大学圣地亚哥分校基因组浏览器。
- 使用 Amazon S3 扩展程序（例如 CWL、WDL 和 Nextflow）的常见工作流程。
- 任何可以从接入点 Amazon S3 进行身份验证和读取 URIs 或读取预签名的 Amazon S3 URIs 的工具。
- Amazon S3 实用工具，例如 Mountpoint 或。 CloudFront

Amazon S3 Mountpoint 使您可以将 Amazon S3 存储桶用作本地文件系统。要了解有关 Mountpoint 的更多信息并安装它以供使用，请参阅适用于 Amazon S3 的 Mountpo int。

Amazon CloudFront 是一项内容分发网络 (CDN) 服务，专为高性能、安全性和开发者便利性而构建。要了解有关使用亚马逊的更多信息 CloudFront，请参阅亚马逊 CloudFront 文档。要设置 CloudFront 序列存储，请联系 AWS HealthOmics 团队。

数据所有者根账户已启用序列存储前缀上的 S3: GetObject、S3: GetObjectTagging 和 S3: List Bucket 操作。要让账户中的用户访问数据，您可以创建一个 IAM 策略并将其附加到该用户或角色。有关策略示例，请参阅 使用 Amazon S3 访问数据的权限 URIs。

您可以对活动读取集使用以下 Amazon S3 API 操作来列出和检索您的数据。激活存档读取集 URIs 后，您可以通过 Amazon S3 访问这些读取集。

- GetObject— 从 Amazon S3 检索对象。
- HeadObject— HEAD 操作从对象检索元数据，而不返回对象本身。如果您只需要对象的元数据，则此操作很有用。

- ListObjects 和 ListObject v2-返回存储桶中的部分或全部（最多 1,000 个）对象。
- CopyObject— 创建已存储在 Amazon S3 中的对象的副本。 HealthOmics支持复制到 Amazon S3 接入点，但不支持写入到接入点。

HealthOmics 序列存储通过 ETags维护文件的语义标识。在文件的整个生命周期中，基于按位身份的 Amazon S3 ETag 可能会发生变化，但 HealthOmics ETag 保持不变。要了解更多信息，请参阅HealthOmics ETags 和数据来源。

主题

- HealthOmics存储中的亚马逊 S3 URI 结构
- 使用托管或本地 IGV 访问读取集
- 使用 Samtools 或者 HTSlib 在 HealthOmics
- 使用挂载点 HealthOmics
- CloudFront 与一起使用 HealthOmics

HealthOmics存储中的亚马逊 S3 URI 结构

所有带有 Amazon S3 的文件 URIs `omics:subjectId`都有`omics:sampleId`资源标签。您可以使用这些标签通过以下模式使用 IAM 策略来共享访问权限`"s3:ExistingObjectTag/omics:subjectId": "pattern desired"`。

文件结构如下：

`.../`*`account_id`*`/sequenceStore/`*`seq_store_id`*`/readSet/`*`read_set_id`*`/`*`files`*`.`

对于从 Amazon S3 导入到序列存储中的文件，序列存储会尝试保留原始源名称。当名称冲突时，系统会附加读取集信息以确保文件名是唯一的。例如，对于 fastq 读取集，如果两个文件名相同，则为了使名称唯一，`sourceX`则将其插入到.fastq.gz 或.fq.gz 之前。对于直接上传，文件名遵循以下模式：

- 对于 FASTQ — *`read_set_name`* _ *`sourcex`* .fastq.gz
- 对于 uBAM/BAM/CRAM —*`read_set_name`*. *`file extension`*扩展名为`.bam`或`.cram`。例如，`NA193948.bam`。

对于 BAM 或 CRAM 的读取集，将在摄取过程中自动生成索引文件。对于生成的索引文件，将在文件名末尾应用正确的索引扩展名。它的模式索*`<name of the Source the index is on>.<file index extension>.`*引扩展名是`.bai`或`.crai`。

使用托管或本地 IGV 访问读取集

IGV 是一款用于分析 BAM 和 CRAM 文件的基因组浏览器。它同时需要文件和索引，因为它一次只能显示基因组的一部分。IGV 可以在本地下载和使用，还有创建 AWS 托管的 IGV 的指南。不支持公共网络版本，因为它需要 CORS。

本地 IGV 依靠本地 AWS 配置来访问文件。确保该配置中使用的角色附加了一个策略，该策略启用 kms: Decrypt 和 s3: 对正在访问的读取集的 s3 URI 的GetObject 权限。之后，在 IGV 中，您可以使用 “文件 > 从 URL 加载”，然后粘贴源和索引的 URI。或者，也可以以相同的方式生成和使用预签名 URLs ，这将绕过 AWS 配置。请注意，Amazon S3 URI 访问不支持 CORS，因此不支持依赖跨域访问的请求。

示例 AWS 托管 IGV 依靠 AWS Cognito 在环境内部创建正确的配置和权限。确保创建的策略启用 KMS: Decrypt 和 s3: 对GetObject正在访问的读取集的 Amazon S3 URI 的权限，并将此策略添加到分配给 Cognito 用户池的角色中。之后，在 IGV 中，您可以使用 “文件 > 从 URL 加载” 并输入源和索引的 URI。或者， URLs 可以按相同的方式生成和使用预签名，从而绕过 AWS 配置。

请注意，序列存储不会显示在 “Amazon” 选项卡下，因为这只会显示配置 AWS 文件所在区域中您拥有的存储桶。

使用 Samtools 或者 HTSlib 在 HealthOmics

HTSlib 是由 Samtools、RSamTools 等多种工具共享的核心库。 PySam使用 1.20 或更高 HTSlib 版本获得对 Amazon S3 接入点的无缝支持。对于 HTSlib 库的旧版本，您可以使用以下解决方法：

- 使用:设置 HTS Amazon S3 主机的环境变量`export HTS_S3_HOST="s3.`*region*`.amazonaws.com"`。
- 为您要使用的文件生成预签名 URL。如果正在使用 BAM 或 CRAM，请确保为文件和索引生成预签名 URL。之后，两个文件都可以与库一起使用。
- 使用 Mountpoint 在使用 HTSlib 库的相同环境中挂载序列存储或读取集前缀。从这里，可以使用本地文件路径访问这些文件。

使用挂载点 HealthOmics

适用于 Amazon S3 的 Mountpoint 是一款简单、高吞吐量的文件客户端，用于将 Amazon S3 存储桶作为本地文件系统进行安装。借助适用于 Amazon S3 的 Mountpoint，您的应用程序可以通过文件操作（例如打开和读取）访问存储在 Amazon S3 中的对象。适用于 Amazon S3 的 Mountpoint 会自动将这

些操作转换为 Amazon S3 对象 API 调用，从而使您的应用程序能够通过文件接口访问 Amazon S3 的弹性存储和吞吐量。

可以使用 Mountpoint 安装说明来安装 M ount point。Mountpoint 使用安装本地的 AWS 配置文件，在 Amazon S3 前缀级别上运行。确保正在使用的配置文件具有启用对正在访问的读取集或序列存储的 Amazon S3 URI 前缀的 s3: ListBucket、s3: 和 kms:解密权限的策略。GetObject之后，可以使用以下路径挂载存储桶：

```
mount-s3 access point arn local path to mount --prefix prefix to sequence store or read
 set --region region
```

CloudFront 与一起使用 HealthOmics

Amazon CloudFront 是一项内容分发网络 (CDN) 服务，专为实现高性能、安全性和开发者便利性而构建。想要使用的客户 CloudFront 必须与服务团队合作才能开启 CloudFront 分发。与您的客户团队合作，与 HealthOmics 服务团队合作。

在中激活读取集 HealthOmics

您可以激活使用 start-read-set-activation-job API 操作或通过存档的读取集 AWS CLI，如以下示例所示。将*sequence store ID*和*read set id*替换为您的序列存储 ID 和读取集 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
```

您会收到一条包含激活任务信息的响应，如下所示。

```
{
    "id": "12345678",
    "sequenceStoreId": "1234567890",
    "status": "SUBMITTED",
    "creationTime": "2022-10-22T00:50:54.670000+00:00"
}
```

激活作业启动后，您可以使用 get-read-set-activation-job API 操作监控其进度。以下是如何使用 AWS CLI 来检查激活任务状态的示例。将*job ID*和*sequence store ID*分别替换为您的序列存储 ID 和作 IDs业。

```
aws omics get-read-set-activation-job --id job ID --sequence-store-id sequence store ID
```

响应汇总了激活作业，如下所示。

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

您可以通过 get-read-set-metadataAPI 操作检查激活任务的状态。可能的状态是ACTIVEACTIVATING、和。ARCHIVED在以下示例中，*sequence store ID*替换为您的序列存储 ID，然后*read set ID*替换为您的读取集 ID。

```
aws omics get-read-set-metadata --sequence-store-id sequence store ID --id read set ID
```

以下响应显示读取集处于活动状态。

```
{
    "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"
    }
}
```

您可以使用 list-read-set-activation-jobs 查看所有读取集激活作业，如以下示例所示。在以下示例中，*sequence store ID*替换为您的序列存储 ID。

```
aws omics list-read-set-activation-jobs --sequence-store-id sequence store ID
```

您会收到以下回复。

```
{
    "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 分析

HealthOmics analytics 支持基因组变异和注释的存储和分析。Analytics 提供两种类型的存储资源——变体存储和注释存储。您可以使用这些资源来存储、转换和查询基因组变异数据和注释数据。将数据导入数据存储后，可以使用 Athena 对数据进行高级分析。

您可以使用 HealthOmics 控制台或 API 来创建和管理商店、导入数据以及与合作者共享分析商店数据。

变体存储支持 VCF 格式的数据，注释存储支持 TSV/CSV 和 GFF3格式。基因组坐标表示为从零开始、半封闭的半开区间。当您的数据存储在 HealthOmics 分析数据存储中时，将通过 AWS Lake Formation管理对 VCF 文件的访问权限。然后，您可以使用亚马逊 Athena 查询 VCF 文件。查询必须使用 Athena 查询引擎版本 3。要了解有关 Athena 查询引擎版本的更多信息，请参阅亚马逊 A thena 文档。

主题

创建 HealthOmics 多属性商店

以下主题介绍如何使用控制台和 API 创建 HealthOmics 变体存储。

主题

使用控制台创建变体商店

您可以使用 HealthOmics 控制台创建多属性商店。

1. 打开 HealthOmics 管理控制台。
2. 在左侧导航窗格中，选择变体商店。
3. 在创建多属性商店页面上，提供以下信息
   - 变体商店名称-此商店的唯一名称。
   - 描述（可选）-此变体存储的描述。
   - 参考基因组-该变异库的参考基因组。
   - 数据加密-选择数据加密由 AWS 自己所有和管理。
   - 标签（可选）-为此变体商店提供最多 50 个标签。
4. 选择创建多属性商店。

使用 API 创建变体商店

使用 HealthOmics `CreateVariantStore` API 操作创建变体存储。您也可以使用执行此操作 AWS CLI。

要创建多属性商店，您需要为商店提供名称和参考商店的 ARN。当变体存储的状态更改为 READY 时，变体存储已准备好提取数据。

以下示例使用创建 AWS CLI 变体存储。

```
aws omics create-variant-store --name myvariantstore \
    --reference referenceArn="arn:aws:omics:us-
west-2:555555555555:referenceStore/123456789/reference/5987565360"
```

为了确认您的多属性商店的创建，您会收到以下回复。

```
{
    "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"
}
```

要了解有关变体商店的更多信息，请使用 get-variant-storeAPI。

```
aws omics get-variant-store --name myvariantstore
```

您会收到以下回复。

```
{
    "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
}
```

要查看与账户关联的所有多属性商店，请使用 list-variant-storesAPI。

```
aws omics list-variant-stores
```

您会收到一条响应，其中列出了所有变体商店及其 IDs状态和其他详细信息，如以下示例响应所示。

```
{
    "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
        }
    ]
}
```

您还可以根据状态或其他条件筛选 list-variant-storesAPI 的响应。

导入到 2023 年 5 月 15 日当天或之后创建的分析存储中的 VCF 文件定义了变异效应预测变量 (VEP) 注释的架构。这样可以更轻松地查询和解析导入的 VCF 数据。此更改不会影响 2023 年 5 月 15 日之前创建的商店，除非该annotation fields参数包含在 API 或 CLI 调用中。对于这些商店，使用annotation fields参数将导致请求失败。

创建 HealthOmics 变体商店导入任务

以下示例说明如何使用 AWS CLI 为多属性商店创建导入任务。

```
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"}
    ]
}
```

对于 2023 年 5 月 15 日之后创建的商店，以下示例说明如何添加--annotation-fields参数。注释字段是在导入时定义的。

```
aws omics start-variant-import-job \
```

```
   --destination-name annotationparsingvariantstore \
   --role-arn arn:aws:iam::123456789012:role/<role_name> \
   --items source=s3://pathToS3/sample.vcf
   --annotation-fields '{"VEP": "CSQ"}'
```

```
{
    "jobId": "981e2286-e954-4391-8a97-09aefc343861"
}
```

get-variant-import-job用于检查状态。

```
aws omics get-variant-import-job --job-id 08279950-a9e3-4cc3-9a3c-a574f9c9e229
```

您将收到一个 JSON 响应，其中显示了您的导入任务的状态。VCF 中的 VEP 注释会被解析为成对存储在 INFO 列中的信息。 ID/Value E nsembl Variant Effect Predictor 注释 INFO 列的默认 ID 是 CSQ，但您可以使用该`--annotation-fields`参数来指示 INFO 列中使用的自定义值。VEP 注释目前支持解析。

对于 2023 年 5 月 15 日之前创建的商店或不包含 VEP 注释的 VCF 文件，响应中不包含任何注释字段。

```
{
    "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",
}
```

作为 VCF 文件一部分的 VEP 注释存储为预定义架构，其结构如下。extras 字段可用于存储默认架构中未包含的任何其他 VEP 字段。

```
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>
    >>
>
```

解析是以尽力而为的方法进行的。如果 VEP 条目不符合 VEP 标准规范，则不会对其进行解析，数组中的行将为空。

对于新的变体存储，的响应get-variant-import-job将包括注释字段，如图所示。

```
aws omics get-variant-import-job --job-id 08279950-a9e3-4cc3-9a3c-a574f9c9e229
```

您会收到一个 JSON 响应，其中显示了您的导入任务的状态。

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

您可以使用list-variant-import-jobs查看所有导入任务及其状态。

```
aws omics list-variant-import-jobs --ids 7a1c67e3-b7f9-434d-817b-9c571fd63bea
```

该响应包含如下信息。

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

如有必要，您可以使用以下命令取消导入任务。

```
aws omics cancel-variant-import-job
     --job-id edd7b8ce-xmpl-47e2-bc99-258cac95a508
```

创建 HealthOmics 注释存储库

注释存储是表示注释数据库的数据存储，例如来自 TSV、VCF 或 GFF 文件的注释数据库。如果指定了相同的参考基因组，则在导入过程中，注释存储将映射到与变体存储相同的坐标系。以下主题介绍如何使用 HealthOmics 控制台以及 AWS CLI 如何创建和管理注释存储库。

主题

使用控制台创建注释存储库

使用以下步骤通过 HealthOmics 控制台创建注释存储库。

创建注释存储库

1. 打开 HealthOmics 管理控制台。
2. 在左侧导航窗格中，选择注释存储。
3. 在注释存储页面上，选择创建注释存储。
4. 在创建注释存储页面上，提供以下信息
   - 注释商店名称-此商店的唯一名称。
   - 描述（可选）-此参考基因组的描述。
   - 数据格式和架构详细信息-选择数据文件格式并上传此存储的架构定义。
   - 参考基因组-此注释的参考基因组。
   - 数据加密-选择是希望数据加密由 AWS 自己拥有和管理。
   - 标签（可选）-为此注释存储库提供最多 50 个标签。
5. 选择 “创建注释存储”。

使用 API 创建注释存储库

以下示例说明如何使用创建注释存储库 AWS CLI。对于所有操作 AWS CLI 和 API 操作，您必须指定数据的格式。

```
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
```

您会收到以下回复，以确认您的注释存储库已创建。

```
{
           "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"
       }
```

要了解有关注释存储的更多信息，请使用 get-annotation-storeAPI。

```
aws omics get-annotation-store --name my_annotation_store
```

您会收到以下回复。

```
{
          "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
      }
```

要查看与账户关联的所有注释库，请使用 list-annotation-storesAPI 操作。

```
aws omics list-annotation-stores
```

您会收到一个列出所有注释库及其 IDs状态和其他详细信息的响应，如以下示例响应所示。

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

您也可以根据状态或其他条件筛选回复。

为 HealthOmics 注释存储创建导入任务

主题

- 使用 API 创建注释导入任务
- TSV 和 VCF 格式的其他参数
- 创建 TSV 格式的注释存储库
- 启动 VCF 格式化的导入作业

使用 API 创建注释导入任务

以下示例说明如何使用启动注释导入作业。 AWS CLI

```
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"}'
```

如果包含注释字段，则在 2023 年 5 月 15 日之前创建的注释存储库会返回一条错误消息。它们不会返回与注释存储导入任务相关的任何 API 操作的输出。

然后，您可以使用 get-annotation-import-jobAPI 操作和job ID参数来了解有关注释导入任务的更多详细信息。

```
aws omics get-annotation-import-job --job-id 9e4198fb-fa85-446c-9301-9b823a1a8ba8
```

您会收到以下响应，包括注释字段。

```
{
          "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"}
       }
```

要查看所有注释存储导入任务，请使用list-annotation-import-jobs。

```
aws omics list-annotation-import-jobs --ids 9e4198fb-fa85-446c-9301-9b823a1a8ba8
```

响应包括您的注释存储导入任务的详细信息和状态。

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

TSV 和 VCF 格式的其他参数

对于 TSV 和 VCF 格式，还有其他参数可以告知 API 如何解析您的输入。

> **Important**
>
> 使用查询引擎导出的 CSV 注释数据会直接返回数据集导入的信息。如果导入的数据包含公式或命令，则该文件可能会被注入 CSV。因此，使用查询引擎导出的文件可能会提示安全警告。为避免恶意活动，请在读取导出文件时关闭链接和宏。

TSV 解析器还执行基本的生物信息学操作，例如基因组学坐标的左归一化和标准化，如下表所示。

格式类型	描述
通用	通用文本文件。没有基因组信息。
CHR_POS	起始位置-1，添加结束位置，与POS。
CHR_POS_REF_ALT	包含 contig、1-base 位置、ref 和 alt 等位基因信息。
CHR_START_END_REF_ALT_ONE_BASE	包含连续、开始、结束、参考和替代等位基因信息。坐标以 1 为基准。

格式类型	描述
CHR_START_END_ZERO_BASE	包含连续位置、起始位置和结束位置。坐标以 0 为基准。
CHR_START_END_ONE_BASE	包含连续位置、起始位置和结束位置。坐标以 1 为基准。
CHR_START_END_REF_ALT_ZERO_BASE	包含连续、开始、结束、参考和替代等位基因信息。坐标以 0 为基准。

TSV 导入注解存储请求类似于以下示例。

```
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"
        }
    }
}'
```

创建 TSV 格式的注释存储库

以下示例使用包含标题、行和注释的选项卡限制文件创建注释存储。坐标是CHR_START_END_ONE_BASED，它包含 OMIM 的人类 HG19 基因图谱概要中的基因图谱。

```
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}]}'
```

您可以导入带或不带标题的文件。要在 CLI 请求中指明这一点`header=false`，请使用，如以下导入任务示例所示。

```
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="#"}}'
```

以下示例为 bed 文件创建注释存储。bed 文件是一个简单的制表符分隔文件。在此示例中，列为染色体、起点、结束和区域名称。坐标从零开始，并且数据没有标题。

```
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}]}'
```

然后，您可以使用以下 CLI 命令将 bed 文件导入注释存储区。

```
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="#"}}'
```

以下示例为以制表符分隔的文件创建注释存储，该文件包含 VCF 文件的前几列，后面是带有注释信息的列。它包含基因组位置，以及有关染色体、起点、参考和备用等位基因的信息，并包含标题。

```
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}]}'
```

然后，您可以使用以下 CLI 命令将文件导入注释存储区。

```
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="#"}}'
```

以下示例显示了客户如何为 mim2gene 文件创建注释存储库。mim2gene 文件提供了 OMIM 中的基因与其他基因标识符之间的链接。它是用制表符分隔的，包含注释。

```
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}]}'
```

然后，您可以按如下方式将数据导入您的商店。

```
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="#"}}'
```

启动 VCF 格式化的导入作业

对于 VCF 文件，还有另外两个输入`ignoreQualField`和`ignoreFilterField`，它们会忽略或包含这些参数，如图所示。

```
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
    }
   }'
```

您也可以取消注释存储库的导入，如图所示。如果取消成功，则您不会收到此 AWS CLI 呼叫的回复。但是，如果找不到导入任务 ID 或导入任务已完成，则会收到一条错误消息。

```
aws omics cancel-annotation-import-job --job-id edd7b8ce-xmpl-47e2-bc99-258cac95a508
```

> **Note**
>
> 您的元数据导入get-annotation-import-job、get-variant-import-joblist-annotation-import-jobs、和list-variant-import-jobs的任务历史记录将在两年后自动删除。导入的变体和注释数据不会自动删除，而是保留在您的数据存储中。

创建 HealthOmics 注释存储库的新版本

您可以创建新版本的注解存储库，以收集不同版本的注释数据库。这可以帮助您整理注释数据，这些数据会定期更新。

要创建现有注释库的新版本，请使用 create-annotation-store-versionAPI，如以下示例所示。

```
aws omics create-annotation-store-version \
     --name my_annotation_store \
     --version-name my_version
```

您将收到以下带有注释存储版本 ID 的响应，确认注释的新版本已创建。

```
{
     "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"
}
```

要更新注释库版本的描述，您可以使用update-annotation-store-version向注释库版本添加更新。

```
aws omics update-annotation-store-version \
    --name my_annotation_store \
    --version-name my_version \
    --description "New Description"
```

您将收到以下回复，确认注释库版本已更新。

```
{
     "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"
}
```

要查看注释库版本的详细信息，请使用get-annotation-store-version。

```
aws omics get-annotation-store-version --name my_annotation_store --version-name
 my_version
```

您将收到包含版本名称、状态和其他详细信息的回复。

```
{
     "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
    }
```

要查看注释存储库的所有版本，可以使用 list-annotation-store-versions，如以下示例所示。

```
aws omics list-annotation-store-versions --name my_annotation_store
```

您将收到包含以下信息的回复

```
{
```

```
  "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
    }
}
```

如果您不再需要注释库版本，则可以使用delete-annotation-store-versions删除注释库版本，如以下示例所示。

```
aws omics delete-annotation-store-versions --name my_annotation_store --versions
 my_version
```

如果删除商店版本时没有出现错误，您将收到以下响应。

```
{
  "errors": []
}
```

如果存在错误，您将收到包含错误详细信息的回复，如图所示。

```
{
```

```
  "errors": [
    {
      "versionName": "my_version",
      "message": "Version with versionName: my_version was not found."
    }
  ]
}
```

如果您尝试删除导入任务处于活动状态的注释库版本，则会收到一条错误响应，如图所示。

```
{
  "errors": [
    {
      "versionName": "my_version",
      "message": "version has an inflight import running"
    }
  ]
}
```

在这种情况下，您可以强制删除注释存储版本，如以下示例所示。

```
aws omics delete-annotation-store-versions --name my_annotation_store --versions
 my_version --force
```

删除 HealthOmics 分析存储

当您删除变体或注释存储时，系统还会删除该存储区中所有导入的数据以及所有关联的标签。

以下示例说明如何使用删除多属性商店 AWS CLI。如果操作成功，则变体存储状态将转换为DELETING。

```
aws omics delete-variant-store --id <variant-store-id>
```

以下示例说明如何删除注释存储库。如果操作成功，则注释存储状态将转换为DELETING。如果存在多个版本，则无法删除注释存储库。

```
aws omics delete-annotation-store --id <annotation-store-id>
```

查询 HealthOmics 分析数据

您可以使用 Amazon Athena 或 Amazon AWS Lake Formation EMR 对您的多属性商店进行查询。在运行任何查询之前，请完成 Lake Formation 和 Amazon Athena 的设置过程（如以下各节所述）。

有关 Amazon EMR 的信息，请参阅教程：亚马逊 EMR 入门

对于 2024 年 9 月 26 日之后创建的多属性商店，按样本 ID 对商店进行 HealthOmics 分区。这种分区意味着 HealthOmics 使用样本 ID 来优化变体信息的存储。使用示例信息作为筛选器的查询将更快地返回结果，因为查询扫描的数据较少。

HealthOmics 使用示例 IDs 作为分区文件名。在采集数据之前，请检查样本 ID 是否包含任何 PHI 数据。如果是，请在采集数据之前更改样本 ID。有关样本中应包含和不包含哪些内容的更多信息 IDs，请参阅 AWS HIPAA 合规性网页上的指南。

主题

配置 Lake Formation 以供使用 HealthOmics

在使用 Lake Formation 管理 HealthOmics 数据存储之前，请执行以下 Lake Formation 配置过程。

主题

创建或验证 Lake Formation 管理员

在 Lake Formation 中创建数据湖之前，需要先定义一个或多个管理员。

管理员是有权创建资源链接的用户和角色。您可以为每个区域的每个账户设置数据湖管理员。

在 Lake Formation 控制台中创建管理员用户

1. 打开 AWS Lake Formation 控制台：Lake Formation 控制台

2. 如果控制台显示 “欢迎来到 Lake Formation” 面板，请选择 “开始”。

   Lake Formation 会将您添加到数据湖管理员表中。

3. 否则，请从左侧菜单中选择 “管理角色和任务”。
4. 根据需要添加任何其他管理员。

使用 Lake Formation 控制台创建资源链接

要创建用户可以查询的共享资源，必须禁用默认访问控制。要了解有关禁用默认访问控制的更多信息，请参阅 Lake Formation 文档中的更改数据湖的默认安全设置。您可以单独创建资源链接，也可以成组创建资源链接，这样您就可以访问 Amazon Athena AWS 或其他服务（例如 Amazon EMR）中的数据。

在 AWS Lake Formation 控制台中创建资源链接并与 HealthOmics Analytics 用户共享

1. 打开 AWS Lake Formation 控制台：Lake Formation 控制台
2. 在主导航栏中，选择数据库。
3. 在 “数据库” 表中，选择所需的数据库。
4. 从 “创建” 菜单中选择 “资源链接”。
5. 输入资源链接名称。如果您计划从 Athena 访问数据库，请仅使用小写字母（最多 256 个字符）输入名称。
6. 选择创建。
7. 新的资源链接现在列在 “数据库” 下。

使用 Lake Formation 控制台授予对共享资源的访问权限

Lake Formation 数据库管理员可以使用以下步骤授予对共享资源的访问权限。

1. 打开 AWS Lake Formation 控制台：https://console.aws.amazon.com/lakeformation/
2. 在主导航栏中，选择数据库。
3. 在 “数据库” 页面上，选择您之前创建的资源链接。
4. 从 “操作” 菜单中选择 “授予目标”。
5. 在委托人下的授予数据权限页面上，选择 IAM 用户或角色。
6. 从 IAM 用户或角色下拉菜单中，找到您要向其授予访问权限的用户。

7. 接下来，在 LF-Tags 或目录资源卡下，选择命名数据目录资源选项。
8. 从 “表格可选” 下拉菜单中，选择 “所有表” 或之前创建的表。
9. 在 “表权限” 卡片中，在 “表权限” 下选择 “描述并选择”。
10. 接下来，选择授权。

要查看 Lake Formation 权限，请从主导航窗格中选择数据湖权限。该表显示了可用的数据库和资源链接。

为 AWS RAM 资源共享配置权限

在 AWS Lake Formation 控制台中，通过在主导航栏中选择数据湖权限来查看权限。在数据权限页面上，您可以查看一个表，其中显示了资源类型、数据库以及**ARN**与 RAM 资源共享下的共享资源相关的资源。如果您需要接受 AWS Resource Access Manager (AWS RAM) 资源共享，则会在控制台中 AWS Lake Formation 通知您。

HealthOmics 可以在商店创建期间隐式接受 AWS RAM 资源共享。要接受 AWS RAM 资源共享，调用或 `CreateAnnotationStore` API 操作的 IAM 用户`CreateVariantStore`或角色必须允许以下操作：

- `ram:GetResourceShareInvitations`-此操作 HealthOmics 允许查找邀请。
- `ram:AcceptResourceShareInvitation`-此操作 HealthOmics 允许使用 FAS 令牌接受邀请。

如果没有这些权限，您将在商店创建过程中看到授权错误。

以下是包含这些操作的策略示例。将此策略添加到接受 AWS RAM 资源共享的 IAM 用户或角色。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "omics:*",
        "ram:AcceptResourceShareInvitation",
        "ram:GetResourceShareInvitations"
```

```
      ],
      "Resource": "*"
    }
  ]
}
```

配置 Athena 以进行查询

您可以使用 Athena 来查询变体和注释。在运行任何查询之前，请执行以下设置任务：

主题

- 使用 Athena 控制台配置查询结果位置
- 使用 Athena 引擎 v3 配置工作组

使用 Athena 控制台配置查询结果位置

要配置查询结果位置，请按照以下步骤操作。

1. 打开 Athena 主机：Athena 主机
2. 在主导航栏中，选择查询编辑器。
3. 在查询编辑器中，选择 “设置” 选项卡，然后选择 “管理”。
4. 输入位置的 S3 前缀以保存查询结果。

使用 Athena 引擎 v3 配置工作组

要配置工作组，请执行以下步骤。

1. 打开 Athena 主机：Athena 主机
2. 在主导航栏中，选择工作组，然后选择创建工作组。
3. 输入工作组的名称。
4. 选择 Athena SQL 作为引擎类型。
5. 在 “升级查询引擎” 下，选择 “手动”。
6. 在 “查询版本引擎” 下，选择 Athena 版本 3。
7. 选择 Create workgroup (创建工作组)。

在 HealthOmics 变体商店上运行查询

您可以使用 Amazon Athena 在您的多属性商店中执行查询。请注意，变体和注释存储中的基因组坐标表示为从零开始、半封闭的半开间隔。

使用 Athena 控制台运行简单查询

以下示例说明如何运行简单查询。

1. 打开 Athena 查询编辑器：Athena 查询编辑器
2. 在“工作组”下，选择您在安装过程中创建的工作组。
3. 验证数据源是否为AwsDataCatalog。
4. 对于数据库，选择您在 Lake Formation 设置期间创建的数据库资源链接。
5. 将以下查询复制到查询编辑器的 Qu ery 1 选项卡下：

```
SELECT * from omicsvariants limit 10
```

6. 选择运行以运行查询。控制台使用表格的前 10 行填充结果omicsvariants表。

使用 Athena 控制台运行复杂查询

以下示例说明如何运行复杂查询。要运行此查询，请导ClinVar入到注释存储中。

运行复杂查询

1. 打开 Athena 查询编辑器：Athena 查询编辑器
2. 在“工作组”下，选择您在安装过程中创建的工作组。
3. 验证数据源是否为AwsDataCatalog。
4. 对于数据库，选择您在 Lake Formation 设置期间创建的数据库资源链接。
5. 选择右+上角的，创建一个名为 Query 2 的新查询选项卡。
6. 将以下查询复制到 Query 2 选项卡下的查询编辑器中：

```
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. 选择 Run 开始运行查询。

共享 HealthOmics 分析存储

作为变体存储或注释存储的所有者，您可以与其他 AWS 账户共享该商店。所有者可以通过删除共享来撤消对共享资源的访问权限。

作为共享商店的订阅者，您首先要接受共享。然后，您可以定义使用共享商店的工作流程。数据以表格形式显示在两者 AWS Glue 和 Lake Formation 中。

当您不再需要访问商店时，可以删除共享。

有关资源共享中的跨账户资源共享 AWS HealthOmics的更多信息，请参阅。

创建商店共享

要创建商店共享，请使用创建共享 API 操作。主要订阅者是 AWS 账户 将要订阅该份额的用户。以下示例为多属性商店创建共享。要将商店与多个账户共享，您需要为同一家商店创建多个共享。

```
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"
```

如果创建成功，您将收到包含共享 ID 和状态的响应。

```
{
       "shareId": "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a",
```

```
        "name": "my_Share-123",
        "status": "PENDING"
    }
```

在订阅者使用接受共享 API 操作接受共享之前，共享将保持待处理状态。

中的跨账户资源共享 AWS HealthOmics

使用跨账户共享与合作者共享资源，无需创建副本或修改 IAM 资源策略。以下资源支持跨账户共享：

- HealthOmics 变体商店
- HealthOmics 注释存储
- 私有工作流程

共享资源包括以下步骤：

1. 资源所有者创建共享，并指定资源的 ARN 和目标订阅者 AWS 账户 的 ARN。在订阅者接受共享之前，资源共享将保持待处理状态。
2. 订阅者接受资源共享以获得对资源的访问权限。资源共享将变为激活状态。
3. 该 HealthOmics 服务为订阅者账户提供对资源的访问权限。
4. 资源所有者可以删除共享，或者订阅者可以撤消他们对共享的访问权限。订阅者无法删除共享或关联的资源。

主题

创建共享

您可以使用创建共享 API 操作来创建共享。主订阅 AWS 账户 者是将订阅共享资源的用户。以下示例为多属性商店创建共享。

```
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"
```

如果创建成功，您将收到包含共享 ID 和状态的响应。

```
{
"shareId": "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a",
     "name": "my_Share-123",
     "status": "PENDING"
}
```

在订阅者使用 accept-share API 操作接受共享之前，共享将保持待处理状态。

```
 aws omics accept-share \
    --share-id "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a"
```

订阅者接受共享后，共享将变为活动状态。

```
{
"status": "ACTIVATING"
}
```

检索有关共享的信息

使用 get-share API 操作来检索有关共享的信息。

```
aws omics get-share --share-id "495c21bedc889d07d0ab69d710a6841e-
dd75ab7a1a9c384fa848b5bd8e5a7e0a"
```

API 响应包含有关共享的元数据信息。

```
{
  "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"
    }
}
```

查看您拥有的股份

使用列表共享 API 检索有关您拥有的每个共享的信息。

```
aws omics list-shares  --resource-owner SELF
```

API 响应包含您拥有的每个共享的元数据。

查看其他账户已接受的股票

使用列表共享 API 查看您从其他账户接受的所有共享。

```
aws omics list-shares  --resource-owner OTHER
```

API 响应包含您接受的每个共享的元数据。

删除共享

在不再需要共享后，使用删除共享 API 将其删除。

```
aws omics delete-share \
    --share-id "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a"
```

在中标记资源 HealthOmics

主题

重要提示

HealthOmics 根据 AWS 责任共担模型政策保护客户数据。这意味着所有客户数据在过渡和静态时都经过加密。但是，并非所有客户输入的资源（例如数据存储或基于作业的操作）的名称都经过加密。它们不应包含个人身份信息或 Protected Health 信息。有关更多信息，请参阅 AWS 中的安全 HealthOmics。

为资源添加标签 HealthOmics

您可以使用标签为您的 AWS 资源分配元数据。每个标签都是由用户定义的键和值组成的标签。标签可帮助您管理、识别、组织、搜索和筛选资源。

本主题介绍常用的标记类别和策略，以帮助您实施一致且有效的标记策略。以下各节假设对 AWS 资源、标签、详细账单和。 AWS Identity and Access Management

每个标签具有两个部分：

- 标签密钥（例如 CostCenter，“环境” 或 “项目”）。标签键区分大小写。
- 标签值（例如，111122223333 或 Production）。与标签键一样，标签值区分大小写。

您可使用标签，按用途、所有者、环境或其他标准对资源进行分类。有关更多信息，请参阅 AWS 标签策略。

您可以从资源的服务控制台、服务 API 或中为该资源添加、更改或移除标签 AWS CLI。

要启用标记，请确保 TagResources 已授权。您可以 TagResources 通过附加 IAM 策略进行授权，如下例所示。

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": "*"
        }
    ]
}
```

最佳实践

在为 AWS 资源创建标签策略时，请遵循最佳实践：

- 请勿在标签中存储个人身份信息 (PII)、Protected Health 信息 (PHI) 或其他敏感信息。

- 对标签使用标准化的区分大小写格式，并跨所有资源类型一致地应用该格式。
- 考虑支持多种用途的标签准则，如管理资源访问控制、成本跟踪、自动化和组织。
- 运用自动化工具帮助您管理资源标签。AWS Resou rce Groups 和 Resource Groups Tagging API 支持对标签进行编程控制，从而可以自动管理、搜索和筛选标签和资源。
- 使用更多标签时，标记会更有效。
- 标签可以根据用户需求的变化进行编辑或修改。但是，要更新访问控制标签，您还必须更新引用这些标签的策略以控制对资源的访问权限。

标记要求

标签具有以下要求：

- 密钥不能以 aws: 为前缀。
- 每个标签集中的各个键必须是独一无二的。
- 键的长度必须介于 1 到 128 个允许的字符之间。
- 值的长度必须介于 0 到 256 个允许的字符之间。
- 每个标签集的值不必是唯一的。
- 可以用作键和值的字符包括 Unicode 字符、数字、空格及以下符号：_ . : / = + - @。
- 键和值区分大小写。

序列存储读取集标签

对于序列存储，在读取集上创建的标签位于读取集资源级别。读取集下还包含可以使用 S3 访问、搜索和限制的对象 APIs。默认情况下，样本 ID (omics: sampleId) 和主题 ID (omics: subjectID) 会添加到对象中。

此外，读取集与其下的对象之间最多可以同步五个标签。要同步标签的配置是在商店创建或更新期间使用`propogatedSetLevelTags`参数设置的存储级别配置。

如果存储中已有数据，则更新密钥可能需要一段时间。在此更新期间，将商店状态 HealthOmics 更改为Updating。完成后， HealthOmics 将商店状态设置为Active。当标签传播时，可能无法强制执行依赖标签的权限。将在标签传播完成后强制执行权限。

在读取集上设置或更新标签时，系统将根据存储配置决定是否更新该读取集的对象。

为 HealthOmics 资源添加标签

为资源添加标签可以帮助您识别和组织您的 AWS 资源并管理对它们的访问权限。首先，向资源添加一个或多个标签（键值对）。每个资源最多可以使用 50 个标签。在键和值字段中可以使用的字符也有限制。

添加标签后，您可以根据这些标签创建 IAM 策略来管理对 AWS 资源的访问权限。您可以使用 HealthOmics 控制台或 AWS CLI 向资源添加标签。为存储库添加标签会影响对该存储库的访问。在向数据存储添加标签之前，请查看任何可能使用标签来控制对资源（例如数据存储）的访问的 IAM 策略。

系统会自动生成序列存储的主题和样本 ID 的服务标签。

按照以下步骤使用 AWS CLI 向 HealthOmics 资源添加标签。例如，要在创建序列存储时向其添加标签，可以在中使用以下命令 AWS CLI。序列存储的名称为 MySequenceStore，添加的两个带键的标签是 key1 和 key2，其值分别为 value1 和 value2 :

```
aws omics create-sequence-store --name "MySequenceStore" --tags key1=value1,key2=value2
```

输出未列出标签。它返回以下响应。

```
{
    "id": "6860403586",
    "referenceStoreId": "4889894479",
    "roleArn": "arn:aws:iam::555555555555:role/ImportTest",
    "status": "CREATED",
    "creationTime": "2022-07-21T01:19:07.194Z"
}
```

要向现有资源添加标签，您需要运行以下示例命令。

```
aws omics tag-resource --resource-arn arn:aws:omics:us-
west-2:555555555555:sequenceStore/2275234794 --tags key1=value1,key2=value2
```

如果成功，该命令不返回任何响应。

列出资源的标签

按照以下步骤 AWS CLI 使用查看 HealthOmics资源的 AWS 标签列表。如果尚未添加标签，则返回的列表为空。

在终端或命令行中，运行 list-tags-for-resource命令，如以下示例所示。

```
aws omics list-tags-for-resource --resource-arn arn:aws:omics:us-
west-2:555555555555:sequenceStore/2275234794
```

您将收到一个 JSON 格式的标签列表作为响应。

```
 {
    "tags": {
        "key1": "value1",
        "key2": "value2"
    }
}
```

从数据存储中移除标签

您可以移除一个或多个与资源关联的标签。移除标签不会从与该标签关联的其他 AWS 资源中删除该标签。

在终端或命令行中，运行 untag-resource 命令，指定要移除标签的资源的亚马逊资源名称 (ARN) 和要删除的标签的标签密钥。

```
aws omics untag-resource --resource-arn arn:aws:omics:us-
west-2:555555555555:sequenceStore/2275234794 --tag-keys key1,key2
```

如果成功，则此命令不会返回响应。要验证与资源关联的标签，请运行 list-tags-for-resource 命令。

的 IAM 权限 HealthOmics

您可以使用 AWS Identity and Access Management (IAM) 来管理 HealthOmics API 以及商店和工作流程等资源的访问权限。对于您账户中使用的用户和应用程序 HealthOmics，您可以在权限策略中管理权限，该策略可以应用于 IAM 用户、群组或角色。

要管理账户中用户和应用程序的权限，请使用 HealthOmics 提供的策略，或自行编写策略。HealthOmics 控制台使用多种服务来获取有关您的函数配置和触发器的信息。您可以按原样使用所提供的策略，也可以将其作为限制性更强的策略的起点。

HealthOmics 使用 IAM 服务角色代表您访问其他服务。例如，在运行从 Amazon S3 读取数据的工作流程时，您需要创建或选择服务角色。对于某些功能，您还需要配置对其他服务中资源的权限。在开始使用之前，请查看这些要求 HealthOmics

有关 IAM 的更多信息，请参阅 IAM 用户指南中的什么是 IAM？。

主题

- 基于身份的 IAM 策略 HealthOmics
- 的服务角色 AWS HealthOmics
- 资源权限
- 使用 Amazon S3 访问数据的权限 URIs

基于身份的 IAM 策略 HealthOmics

要向账户中的用户授予访问权限 HealthOmics，您可以在 AWS Identity and Access Management (IAM) 中使用基于身份的策略。基于身份的策略可以直接应用于 IAM 用户，也可以应用于与用户关联的 IAM 群组和角色。您也可以授予另一个账户中的用户在您的账户中代入角色和访问您的 HealthOmics 资源的权限。

要授予用户对工作流程版本执行操作的权限，必须将工作流和特定工作流程版本添加到资源列表中。

以下 IAM 策略允许用户访问所有 HealthOmics API 操作并将服务角色传递给 HealthOmics。

Example 用户策略

JSON

```
{
```

```
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "omics:*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "iam:PassRole"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": "omics.amazonaws.com"
        }
      }
    }
  ]
}
```

使用时 HealthOmics，您还会与其他 AWS 服务进行交互。要访问这些服务，请使用每项服务提供的托管策略。要限制对资源子集的访问，您可以使用托管策略作为起点来创建自己的限制性更强的策略。

- AmazonS3 FullAccess — 访问任务使用的亚马逊 S3 存储桶和对象。
- 亚马逊 EC2 ContainerRegistryFullAccess — 访问 Amazon ECR 注册表和存储库以获取工作流程容器映像。
- AWSLakeFormationDataAdmin— 访问分析商店创建的 Lake Formation 数据库和表。
- ResourceGroupsandTagEditorFullAccess— 使用标记 API 操作 HealthOmics 来标记 HealthOmics 资源。

上述策略不允许用户创建 IAM 角色。要使具有这些权限的用户运行作业，管理员必须创建服务角色来授予访问数据源的 HealthOmics 权限。有关更多信息，请参阅 的服务角色 AWS HealthOmics。

为运行定义自定义 IAM 权限

您可以在授权请求中包含StartRun请求引用的任何工作流程、运行或运行组。为此，请在 IAM 策略中列出所需的工作流程、运行或运行组组合。例如，您可以将工作流程的使用限制在特定的运行或运行组中。您也可以指定工作流程仅与运行组一起使用。

以下是一个 IAM 策略示例，该策略允许使用单个运行组使用单个工作流程。

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/*"
            ]
        }
    ]
}
```

的服务角色 AWS HealthOmics

服务角色是一个 AWS Identity and Access Management (IAM) 角色，它向 AWS 服务授予访问您账户中资源的权限。当您启动导入任务或开始运行 AWS HealthOmics 时，您可以为其提供服务角色。

HealthOmics 控制台可以为您创建所需的角色。如果您使用 HealthOmics API 管理资源，请使用 IAM 控制台创建服务角色。有关更多信息，请参阅创建角色以向委派权限 AWS 服务。

服务角色必须具有以下信任策略。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "omics.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

信任策略允许 HealthOmics 服务担任该角色。

主题

- IAM 服务策略示例
- 示例 AWS CloudFormation 模板

IAM 服务策略示例

在这些示例中，资源名称和帐户 IDs 是您可以用实际值替换的占位符。

以下示例显示了可用于启动运行的服务角色的策略。该策略授予访问用于运行的 Amazon S3 输出位置、工作流程日志组和 Amazon ECR 容器的权限。

> **Note**
>
> 如果您使用呼叫缓存进行运行，请在 s3 权限中添加运行缓存 Amazon S3 位置作为资源。

Example 用于启动运行的服务角色策略

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/*"
            ]
        }
    ]
}
```

以下示例显示了可用于商店导入任务的服务角色的策略。该策略授予访问 Amazon S3 输入位置的权限。

Example 参考商店作业的服务角色

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"
            ]
        }

    ]
}
```

示例 AWS CloudFormation 模板

以下示例 AWS CloudFormation 模板创建了一个服务角色，该角色授予访问名称前缀为前缀的 Amazon S3 存储桶和上传工作流程日志的 HealthOmics 权限。`omics-`

Example 参考存储、Amazon S3 和 CloudWatch 日志权限

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

资源权限

AWS HealthOmics 当您运行任务或创建商店时，代表您创建和访问其他服务中的资源。在某些情况下，您需要在其他服务中配置访问资源或 HealthOmics 允许访问资源的权限。

Sections

- Amazon ECR 权限
- Lake Formation 权限

Amazon ECR 权限

在该 HealthOmics 服务可以在您的私有 Amazon ECR 存储库的容器中运行工作流程之前，您需要为该容器创建资源策略。该策略授予 HealthOmics 服务使用容器的权限。您可以将此资源策略添加到工作流程引用的每个私有存储库中。

> **Note**
>
> 私有存储库和工作流程必须位于同一区域。

以下各节描述了所需的策略配置。

主题

- 为 Amazon ECR 存储库创建资源策略
- 使用跨账户容器运行工作流程
- 适用于共享工作流程的 Amazon ECR 存储库策略

为 Amazon ECR 存储库创建资源策略

创建资源策略以允许 HealthOmics 服务使用存储库中的容器运行工作流程。该政策允许 HealthOmics 服务主体访问所需的 Amazon ECR 操作。

按照以下步骤创建策略：

1. 在 Amazon ECR 控制台中打开私有存储库页面，然后选择您要授予访问权限的存储库。
2. 在侧栏导航中，选择 “权限”。
3. 选择编辑 JSON。
4. 选择添加声明。
5. 添加以下策略声明，然后选择保存策略。

   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": "*"
        }
    ]
}
```

使用跨账户容器运行工作流程

如果不同的 AWS 账户拥有工作流程和容器，则需要配置以下跨账户权限：

1. 更新存储库的 Amazon ECR 政策，以明确向拥有该工作流程的账户授予权限。

2. 更新拥有该工作流程的账户的服务角色，以授予其访问容器镜像的权限。

以下示例演示了 Amazon ECR 资源策略，该策略向拥有该工作流程的账户授予访问权限。

在本示例中：

- 工作流程账户 ID：111122223333
- 容器存储库账户 ID：444455556666
- 容器名称：samtools

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "omics.amazonaws.com"
            },
            "Action": [
                "ecr:BatchCheckLayerAvailability",
                "ecr:BatchGetImage",
                "ecr:GetDownloadUrlForLayer"
            ]
        },
        {
            "Sid": "allow access to the service role of the account that owns the
 workflow",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:role/DemoCustomer"
            },
            "Action": [
                "ecr:BatchCheckLayerAvailability",
                "ecr:BatchGetImage",
                "ecr:GetDownloadUrlForLayer"
            ]
        }
    ]
```

```
}
```

要完成设置，请向拥有该工作流程的账户的服务角色添加以下策略声明。该策略向服务角色授予访问“samtools” 容器镜像的权限。请务必用您自己的值替换账号、集装箱名称和区域。

```
{
    "Sid": "CrossAccountEcrRepoPolicy",
    "Effect": "Allow",
    "Action": ["ecr:BatchCheckLayerAvailability", "ecr:BatchGetImage",
 "ecr:GetDownloadUrlForLayer"],
    "Resource": "arn:aws:ecr:us-west-2:444455556666:repository/samtools"
}
```

适用于共享工作流程的 Amazon ECR 存储库策略

> **Note**
>
> HealthOmics 当工作流程在订阅者的账户中运行时，自动允许共享工作流程访问工作流程所有者账户中的 Amazon ECR 存储库。您无需为共享工作流程授予额外的存储库访问权限。有关更多信息，请参阅共享 HealthOmics 工作流程。

默认情况下，订阅者无权访问 Amazon ECR 存储库来使用底层容器。或者，您可以通过向存储库的资源策略添加条件密钥来自定义对 Amazon ECR 存储库的访问权限。以下各节提供了策略示例。

限制对特定工作流程的访问权限

您可以在条件语句中列出各个工作流程，因此只有这些工作流程才能使用存储库中的容器。SourceArn条件键指定共享工作流程的 ARN。以下示例授予指定工作流程使用此存储库的权限。

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"
           }
        }
      }
  ]
}
```

限制对特定账户的访问权限

您可以在条件语句中列出订阅者账户，这样只有这些账户才有权使用存储库中的容器。SourceAccount条件键指定订阅 AWS 账户 者的。以下示例授予指定账户使用此存储库的权限。

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"
        }
      }
    }
  ]
}
```

您也可以拒绝向特定订阅者授予 Amazon ECR 权限，如以下示例策略所示。

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"
                }
            }
         }
      ]
  }
```

Lake Formation 权限

在中使用分析功能之前 HealthOmics，请在 Lake Formation 中配置默认数据库设置。

在 Lake Formation 中配置资源权限

1. 在 Lake Formation 控制台中打开数据目录设置页面。
2. 在新创建的数据库和表的默认权限下，取消选中数据库和表的 IAM 访问控制要求。
3. 选择保存。

HealthOmics 如果您的服务策略具有正确的 RAM 权限，Analytics auto 会自动接受数据，例如以下示例。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "omics:*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ram:AcceptResourceShareInvitation",
        "ram:GetResourceShareInvitations"
      ],
      "Resource": "*"
    }
  ]
}
```

使用 Amazon S3 访问数据的权限 URIs

您可以使用 HealthOmics API 操作或 Amazon S3 API 操作访问序列存储数据。

对于 HealthOmics API 访问， HealthOmics 权限通过 IAM 策略进行管理。但是，S3 访问需要两个级别的配置：商店的 S3 访问策略中的明确允许和 IAM 策略。要详细了解如何将 IAM 策略与配合使用 HealthOmics，请参阅的服务角色 HealthOmics。

有三种方法可以共享使用 Amazon S3 读取对象的功能 APIs：

1. 基于策略的共享 — 这种共享需要在 S3 访问策略中启用 IAM 委托人，并编写 IAM 策略并将其附加到 IAM 委托人。有关更多详细信息，请参阅下一个主题。
2. 预签名 URLs — 您还可以为序列存储中的文件生成可共享的预签名 URL。要了解有关 URLs 使用 Amazon S3 创建预签名的更多信息，请参阅 Amazon S3 文档 URLs中的使用预签名。序列存储 S3 访问策略支持限制预签名 URL 功能的语句。
3. 代入的角色-在数据所有者的账户中创建一个角色，该角色的访问策略允许用户担任该角色。

主题

基于策略的共享

如果您使用直接 S3 URI 访问序列存储数据，则会为关联的 S3 存储桶访问策略 HealthOmics 提供增强的安全措施。

以下规则适用于新的 S3 访问策略。对于现有策略，下次更新策略时将应用以下规则：

- S3 访问策略支持以下策略元素
  - 版本、编号、陈述、Sid、效果、主体、操作、资源、条件
- S3 访问策略支持以下条件键：
  - s3:ExistingObjectTag/<key>、s3: 前缀、s3: signatureVersion、s3: TlsVersion
  - 策略还支持 aws:PrincipalArn ，使用以下条件运算符： ArnEquals 和 ArnLike

如果您尝试添加或更新策略以包含不支持的元素或条件，则系统会拒绝该请求。

主题

默认 S3 访问策略

创建序列存储时， HealthOmics 会创建一个默认 S3 访问策略，授予数据存储所有者的根账户对序列存储中所有可访问对象的以下权限：S3: GetObject GetObjectTagging、S3 和 S3: ListBucket。默认创建的策略是：

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/*"
        }
    ]
}
```

自定义访问策略

如果 S3 访问策略为空，则不允许访问 S3。如果存在现有策略并且您需要删除 s3 访问权限，请使用`deleteS3AccessPolicy`删除所有访问权限。

要添加共享限制或向其他账户授予访问权限，您可以使用 `PutS3AccessPolicy` API 更新政策。对策略的更新不能超出序列存储的前缀或指定的操作。

IAM 策略

要允许用户或 IAM 委托人使用 Amazon S3 进行访问 APIs，除了 S3 访问策略中的权限外，还需要创建一个 IAM 策略并将其附加到委托人以授予访问权限。允许 Amazon S3 API 访问权限的策略可以在序列存储级别或读取集级别应用。在读取集级别，可以通过前缀或使用资源标签过滤器来限制样本或主题 ID 模式。

如果序列存储使用客户托管密钥 (CMK)，则委托人还必须有权使用 KMS 密钥进行解密。有关更多信息，请参阅 AWS Key Management Service 开发者指南中的跨账户 KMS 访问。

以下示例为用户提供了对序列存储的访问权限。您可以使用其他条件或基于资源的过滤器来微调访问权限。

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/*"
        }
      }
    }
  ]
}
```

基于标签的访问控制

要使用基于标签的访问控制，必须先更新序列存储以传播将要使用的标签密钥。此配置是在序列存储创建或更新期间设置的。一旦标签被传播，就可以使用标签条件来进一步添加限制。可以在 S3 访问策略或 IAM 策略中设置限制。以下是将要设置的基于选项卡的 S3 访问策略的示例：

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

限制示例

场景：创建一个共享，数据所有者可以在其中限制用户下载 “已撤回” 的数据的能力。

在这种情况下，数据所有者（账户 #111111111111）管理数据存储。该数据所有者与包括研究人员在内的广泛第三方用户共享数据（账户 #999999999999）。作为管理数据的一部分，数据所有者会定期收到撤回参与者数据的请求。为了管理这种撤回，数据所有者首先在收到请求时限制直接下载权限，并最终根据其要求删除数据。

为了满足这一需求，数据所有者设置了一个序列存储，每个读取集都会收到一个 “状态” 标签，如果提款请求通过，该标签将设置为 “已撤回”。对于标签设置为该值的数据，他们希望确保没有用户可以对此文件运行 “getObject”。要进行此设置，数据所有者需要确保采取两个步骤。

第 1 步：对于序列存储，请确保更新状态标签以进行传播。这是通过在呼叫`propogatedSetLevelTags`时将 “状态” 键添加到 “`createSequenceStore`或” 来完成的`updateSequenceStore.`

第 2 步：更新商店的 s3 访问策略，将状态标签设置为已撤回的对象限制 getObject。这是通过使用`PutS3AccesPolicy` API 更新商店访问策略来完成的。以下政策允许客户在列出对象时仍能看到已撤回的文件，但禁止他们访问这些文件：

- 声明 1 (restrictedGetWithdrawal)：账户 999999999999 无法检索已提取的对象。
- 声明 2 (ownerGetAll)：账户 111111111111（数据所有者）可以检索所有对象，包括已撤回的对象。
- 声明 3 (everyoneListAll)：所有共享账户，111111111111和999999999999，都可以在整个前缀上运行该操作。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/*"
                }
            }
        }
    ]
}
```

AWS 中的安全 HealthOmics

云安全 AWS 是重中之重。作为 AWS 客户，您可以受益于专为满足大多数安全敏感型组织的要求而构建的数据中心和网络架构。

安全是双方 AWS 的共同责任。责任共担模式将其描述为云的安全性和云中的安全性：

- 云安全 — AWS 负责保护在云中运行 AWS 服务的基础架构 AWS Cloud。 AWS 还为您提供可以安全使用的服务。作为AWS 合规计划合规计划合规计划合的一部分，第三方审计师定期测试和验证我们安全的有效性。要了解适用于 AWS 的合规计划 HealthOmics，请参阅按合规计划提供的范围内的AWSAWS 服务按合规计划。
- 云端安全-您的责任由您使用的 AWS 服务决定。您还需要对其他因素负责，包括您的数据的敏感性、您公司的要求以及适用的法律法规。

本文档可帮助您了解在使用 AWS 时如何应用分担责任模型 HealthOmics。以下主题向您展示如何配置 AWS HealthOmics 以满足您的安全与合规目标。您还将学习如何使用其他 AWS 服务来帮助您监控和保护您的 AWS HealthOmics 资源。

主题

- 中的数据保护 AWS HealthOmics
- 中的身份和访问管理 HealthOmics
- 合规性验证 AWS HealthOmics
- 韧性在 HealthOmics
- AWS HealthOmics 和接口 VPC 终端节点 (AWS PrivateLink)

中的数据保护 AWS HealthOmics

责任 AWS 共担模式适用于 AWS 中的数据保护 HealthOmics。如本模型所述 AWS ，负责保护运行所有内容的全球基础架构 AWS Cloud。您负责维护对托管在此基础结构上的内容的控制。您还负责您所使用的 AWS 服务 的安全配置和管理任务。有关数据隐私的更多信息，请参阅数据隐私常见问题。有关欧洲数据保护的信息，请参阅 AWS Security Blog 上的 AWS Shared Responsibility Model and GDPR 博客文章。

出于数据保护目的，我们建议您保护 AWS 账户 凭证并使用 AWS IAM Identity Center 或 AWS Identity and Access Management (IAM) 设置个人用户。这样，每个用户只获得履行其工作职责所需的权限。还建议您通过以下方式保护数据：

- 对每个账户使用多重身份验证（MFA）。
- 用于 SSL/TLS 与 AWS 资源通信。我们要求使用 TLS 1.2，建议使用 TLS 1.3。
- 使用设置 API 和用户活动日志 AWS CloudTrail。有关使用 CloudTrail 跟踪捕获 AWS 活动的信息，请参阅《AWS CloudTrail 用户指南》中的使用跟 CloudTrail 踪。
- 使用 AWS 加密解决方案以及其中的所有默认安全控件 AWS 服务。
- 使用高级托管安全服务（例如 Amazon Macie），它有助于发现和保护存储在 Amazon S3 中的敏感数据。
- 如果您在 AWS 通过命令行界面或 API 进行访问时需要经过 FIPS 140-3 验证的加密模块，请使用 FIPS 端点。有关可用的 FIPS 端点的更多信息，请参阅《美国联邦信息处理标准（FIPS）第 140-3 版》。

强烈建议您切勿将机密信息或敏感信息（如您客户的电子邮件地址）放入标签或自由格式文本字段（如名称字段）。这包括您 AWS 服务 使用控制台、API HealthOmics 或与 AWS 或其他机构 AWS CLI合作时 AWS SDKs。在用于名称的标签或自由格式文本字段中输入的任何数据都可能会用于计费或诊断日志。如果您向外部服务器提供网址，强烈建议您不要在网址中包含凭证信息来验证对该服务器的请求。

静态加密

主题

- AWS 拥有的密钥
- 客户托管密钥
- 创建客户托管的密钥
- 使用客户托管密钥所需的 IAM 权限
- 了解更多

为了保护敏感的静态客户数据，默认使用服务自有的 AWS Key Management Service (AWS KMS) 密钥 AWS HealthOmics 提供加密。还支持客户管理的密钥。要了解有关客户托管密钥的更多信息，请参阅 Amazon 密钥管理服务。

所有 HealthOmics 数据存储（存储和分析）都支持使用客户托管密钥。创建数据存储后，无法更改加密配置。如果数据存储使用的是 AWS 拥有的密钥，则会将其表示为， AWS_OWNED_KMS_KEY 并且您将看不到用于静态加密的特定密钥。

对于 HealthOmics Workflows，临时文件系统不支持客户管理的密钥；但是，使用 XTS-AES-256 分组密码加密算法对所有数据进行静态加密，以加密文件系统。用于启动工作流程运行的 IAM 用户和角色还必须有权访问用于工作流程输入和输出存储桶的 AWS KMS 密钥。工作流程不使用授权， AWS KMS 加密仅限于输入和输出 Amazon S3 存储桶。同时用于工作流程的 IAM 角色还 APIs 必须有权访问所使用的 AWS KMS 密钥以及输入和输出 Amazon S3 存储桶。您可以使用 IAM 角色和权限来控制访问权限或 AWS KMS 策略。要了解更多信息，请参阅的身份验证和访问控制 AWS KMS。

当你 AWS Lake Formation 与 HealthOmics Analytics 一起使用时，与 Lake Formation 关联的任何解密权限也会被授予输入和输出 Amazon S3 存储桶。有关如何 AWS Lake Formation 管理权限的更多信息可以在AWS Lake Formation 文档中找到。

HealthOmics Analytics 授予 Lake Formation kms: Decrypt 读取亚马逊 S3 存储桶中加密数据的权限。只要您有权通过 Lake Formation 查询数据，您就可以读取加密的数据。对数据的访问是通过 Lake Formation 中的数据访问控制来控制的，而不是通过 KMS 密钥策略进行的。要了解更多信息，请参阅 Lake Formation 文档中的AWS 集成 AWS 服务请求。

AWS 拥有的密钥

默认情况下， HealthOmics 用于 AWS 拥有的密钥 自动加密静态数据，因为这些数据可能包含敏感信息，例如个人身份信息 (PII) 或 Protected Health 信息 (PHI)。 AWS 拥有的密钥 未存储在您的账户中。它们是 AWS 拥有和管理的 KMS 密钥集合的一部分，可在多个 AWS 账户中使用。

AWS 服务可以 AWS 拥有的密钥 用来保护您的数据。您无法查看、管理 AWS 拥有的密钥、访问或审核其使用情况。但是无需执行任何工作或更改任何计划即可保护用于加密数据的密钥。

您无需支付月费或使用费 AWS 拥有的密钥，也不计入您账户的 AWS KMS 配额。有关更多信息，请参阅 AWS 托管式密钥。

客户托管密钥

HealthOmics 支持使用您创建、拥有和管理的对称客户托管密钥，在现有 AWS 拥有的加密基础上添加第二层加密。由于您可以完全控制这层加密，因此可以执行以下任务：

- 建立和维护密钥政策、IAM Policy 和授权
- 轮换密钥加密材料
- 启用和禁用密钥政策
- 添加标签
- 创建密钥别名

- 安排密钥删除

您还可以使用 CloudTrail 来跟踪代表您 HealthOmics 发送 AWS KMS 的请求。将收取额外的 AWS KMS 费用。有关更多信息，请参阅客户托管密钥。

创建客户托管的密钥

您可以使用 AWS 管理控制台创建对称客户托管密钥，或者。 AWS KMS APIs

按照 AWS Key Management Service 开发人员指南中创建对称客户托管密钥的步骤进行操作。

密钥政策控制对客户托管密钥的访问。每个客户托管式密钥必须只有一个密钥策略，其中包含确定谁可以使用密钥以及如何使用密钥的声明。创建客户托管密钥时，可以指定密钥策略。有关更多信息，请参阅 AWS Key Management Service 开发人员指南中的管理客户托管密钥的访问权限。

要将客户托管密钥与您的 HealthOmics Analytics 资源一起使用，调用主体需要密钥策略中的 kms: CreateGrant 操作。这允许系统使用 FAS 令牌创建对客户托管密钥的授权，以控制对指定 KMS 密钥的访问权限。此密钥允许用户访问所需的 kms: grant 操作。 HealthOmics 有关更多信息，请参阅使用授权。

要进行 HealthOmics 分析，必须允许调用方委托人执行以下 API 操作：

- kms：向特定的客户托管密钥CreateGrant 添加授权，从而允许在 HealthOmics Analytics 中授予操作权限。
- km DescribeKey s：提供验证密钥所需的客户托管密钥详细信息。这是所有操作所必需的。
- kms：GenerateDataKey 为所有写入操作提供对静态加密资源的访问权限。此外，此操作还提供客户托管密钥的详细信息，服务可以使用这些详细信息来验证呼叫者是否有权使用密钥。
- KMS: Decrypt 提供对加密资源的读取或搜索操作的访问权限。

要在 HealthOmics 存储资源中使用客户托管密钥，必须在密钥策略中允许使用 HealthOmics 服务主体和调用主体。这允许服务验证呼叫者是否有权访问密钥，并使用服务主体使用客户托管密钥执行商店管理。对于 HealthOmics 存储，服务主体的密钥策略必须允许以下 API 操作：

- km DescribeKey s：提供验证密钥所需的客户托管密钥详细信息。这是所有操作所必需的。
- kms：GenerateDataKey 为所有写入操作提供对静态加密资源的访问权限。此外，此操作还提供客户托管密钥的详细信息，服务可以使用这些详细信息来验证呼叫者是否有权使用密钥。

- KMS: Decrypt 提供对加密资源的读取或搜索操作的访问权限。

以下示例显示了一个策略声明，该声明允许服务主体创建使用客户托管密钥加密的 HealthOmics 序列或参考存储并与之交互：

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
{
    "Effect": "Allow",
    "Principal": {
        "Service": "omics.amazonaws.com"
    },
    "Action": [
        "kms:Decrypt",
        "kms:DescribeKey",
        "kms:Encrypt",
        "kms:GenerateDataKey*"
    ],
    "Resource": "*"
        }
    ]
}
```

以下示例显示了一个策略，该策略为数据存储创建了解密来自 Amazon S3 存储桶的数据的权限。

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "omics:GetReference",
                "omics:GetReferenceMetadata"
            ],
            "Resource": [
```

```
                "arn:AWS:omics:{{region}}:{{accountId}}:referenceStore/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:AWS:s3:::[[s3path]]/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:AWS:kms:{{region}}:{111122223333}:key/{{key_id}}"
            ],
            "Condition": {
              "StringEquals": {
                  "kms:ViaService": [
                    "s3.{{region}}.amazonAWS.com"
                  ]
              }
           }
        }
    ]
}
```

使用客户托管密钥所需的 IAM 权限

使用客户托管密钥创建诸如 AWS KMS 加密的数据存储之类的资源时，IAM 用户或角色需要密钥策略和 IAM 策略的权限。

您可以使用 k ms: ViaService 条件密钥将 KMS 密钥的使用限制为仅限来自 HealthOmics的请求。

有关密钥策略的更多信息，请参阅 AWS Key Management Service 开发人员指南中的启用 IAM 策略。

主题

- 分析 API 权限

- 存储 API 权限
- 如何在 AWS KMS 中 HealthOmics 使用授权
- 监控您的加密密钥 AWS HealthOmics

分析 API 权限

要进行 HealthOmics 分析，创建商店的 IAM 用户或角色必须具有 kms:CreateGrant kms:GenerateDataKey、、 kms:解密和 kms:DescribeKey 权限以及必要的 HealthOmics 权限。

存储 API 权限

对于 HealthOmics 存储 APIs，调用以下 API 操作的 IAM 用户或角色需要列出的权限：

CreateReferenceStore, CreateSequenceStore

要创建商店，IAM 调用者必须拥有`kms:DescribeKey`权限和必要的 HealthOmics 权限。HealthOmics 服务主体调用`kms:GenerateDataKeyWithoutPlaintext`对数据加载和访问进行访问验证检查。

StartReadSetImportJob, StartReferenceImportJob

要启动数据导入任务，IAM 调用者必须拥有`kms:Decrypt`存储上用于导入的 KMS 密钥的`kms:Decrypt`权限，以及包含要导入的对象的 Amazon S3 存储桶的权限。`kms:GenerateDataKey`此外，传入调用的角色必须对包含要导入的对象的 Amazon S3 存储桶拥有`kms:Decrypt`权限。IAM 调用者还必须拥有将角色传递给任务的权限。

CreateMultipartReadSetUpload, UploadReadSetPart, CompleteMultipartReadSetUpload

要完成分段上传，IAM 调用者必须拥有`kms:Decrypt`和`kms:GenerateDataKey`才能创建、上传和完成分段上传。

StartReadSetExportJob

要启动数据导出任务，IAM 调用者必须拥有`kms:Decrypt`存储上的 KMS 密钥从中导出的`kms:Decrypt`权限`kms:GenerateDataKey`和接收对象的 Amazon S3 存储桶的权限。此外，传入调用的角色必须拥有接收对象的 Amazon S3 存储桶的`kms:Decrypt`权限。IAM 调用者还必须拥有将角色传递给任务的权限。

StartReadsetActivationJob

要启动读取集激活作业，IAM 调用者必须拥有`kms:Decrypt`对象的`kms:GenerateDataKey`权限。

GetReference, GetReadSet

要从存储中读取对象，IAM 调用者必须拥有对象的`kms:Decrypt`权限。

读取集 S3 GetObject

要使用 Amazon S3 `GetObject` API 从商店读取对象，IAM 调用者必须拥有对象的`kms:Decrypt`权限。为客户托管密钥和 AWS 拥有的密钥 配置设置此权限。

如何在 AWS KMS 中 HealthOmics 使用授权

HealthOmics Analytics 需要获得授权才能使用您的客户托管的 KMS 密钥。 HealthOmics 工作流程不需要或不使用赠款。 HealthOmics 存储使用直接来自服务主体的客户托管密钥，因此请勿使用授权。当您创建使用客户托管密钥加密的分析存储时， HealthOmics Analytics 会通过向 AWS KMS 发送CreateGrant请求来代表您创建授权。AWS KMS 中的赠款用于授予对客户账户中的 KMS 密钥的 HealthOmics 访问权限。

不建议撤销或撤销 HealthOmics 分析代表您创建的资助。如果您撤销或取消授予在您的账户中使用 AWS KMS 密钥的 HealthOmics 权限，则 HealthOmics 无法访问这些数据、加密推送到数据存储的新资源或在提取时对其进行解密。

当您撤销或撤销的授予时 HealthOmics，更改会立即生效。要撤消访问权限，我们建议您删除数据存储而不是撤消授权。当您删除数据存储时， HealthOmics会代表您停用授权。

监控您的加密密钥 AWS HealthOmics

使用客户托管密钥时，您可以使用 CloudTrail 来跟踪 AWS KMS 代表您 AWS HealthOmics 发送的请求。日志中的日志条目在 UserA CloudTrail gent 字段中显示 HealthOmics .Amazonaws.com，以明确区分由发出的请求。 HealthOmics

以下示例是 CreateGrant、 GenerateDataKey、Decrypt 和 DescribeKey 监视 AWS KMS 操作 CloudTrail 的事件，这些操作被调用 HealthOmics 以访问由您的客户托管密钥加密的数据。

下文还展示了 CreateGrant 如何使用允许 HealthOmics 分析访问客户提供的 KMS 密钥，从而 HealthOmics 能够使用该 KMS 密钥加密所有静态客户数据。

您无需创建自己的赠款。 HealthOmics 通过向 AWS KMS 发送 CreateGrant 请求来代表您创建资助。中的授权 AWS KMS 用于授予对客户账户中 AWS KMS 密钥的 HealthOmics 访问权限。

```
{
    "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"
}
```

以下示例说明如何使用 GenerateDataKey 来确保用户在存储数据之前拥有加密数据的必要权限。

```
{
    "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"
}
```


了解更多

以下资源提供了有关静态数据加密的更多信息。

有关 AWS Key Management Service 基本概念的更多信息，请参阅 AWS KMS 文档。

有关安全最佳实践的更多信息， AWS KMS 请参阅文档。

传输中加密

AWS HealthOmics 使用 TLS 1.2+ 对通过公共端点和后端服务传输的数据进行加密。

中的身份和访问管理 HealthOmics

AWS Identity and Access Management (IAM) AWS 服务 可帮助管理员安全地控制对 AWS 资源的访问权限。IAM 管理员控制谁可以通过身份验证（登录）和授权（拥有权限）使用 AWS HealthOmics 资源。您可以使用 IAM AWS 服务 ，无需支付额外费用。

主题

受众

您的使用方式 AWS Identity and Access Management (IAM) 会有所不同，具体取决于您在 AWS 中所做的工作 HealthOmics。

服务用户 — 如果您使用 AWS HealthOmics 服务完成工作，则您的管理员会为您提供所需的证书和权限。当您使用更多 AWS HealthOmics 功能来完成工作时，您可能需要额外的权限。了解如何管理访问权限有助于您向管理员请求适合的权限。如果您无法访问 AWS 中的某项功能 HealthOmics，请参阅对 AWS HealthOmics 身份和访问进行故障排除。

服务管理员 — 如果您负责公司的 AWS HealthOmics 资源，则可能拥有对 AWS 的完全访问权限 HealthOmics。您的工作是确定您的服务用户应访问哪些 AWS HealthOmics 功能和资源。然后，您必须向 IAM 管理员提交请求以更改服务用户的权限。请查看该页面上的信息以了解 IAM 的基本概念。要详细了解贵公司如何在 AWS 中使用 IAM HealthOmics，请参阅如何 AWS HealthOmics 与 IAM 配合使用。

IAM 管理员 — 如果您是 IAM 管理员，则可能需要详细了解如何编写策略来管理 AWS 的访问权限 HealthOmics。要查看您可以在 IAM 中使用的 HealthOmics 基于身份的 AWS 策略示例，请参阅。基于身份的策略示例 AWS HealthOmics

使用身份进行身份验证

身份验证是您 AWS 使用身份凭证登录的方式。您必须以 IAM 用户身份或通过担 AWS 账户根用户任 IAM 角色进行身份验证（登录 AWS）。

您可以使用通过身份源提供的凭据以 AWS 联合身份登录。 AWS IAM Identity Center （IAM Identity Center）用户、贵公司的单点登录身份验证以及您的 Google 或 Facebook 凭据就是联合身份的示例。当您以联合身份登录时，您的管理员以前使用 IAM 角色设置了身份联合验证。当你使用联合访问 AWS 时，你就是在间接扮演一个角色。

根据您的用户类型，您可以登录 AWS Management Console 或 AWS 访问门户。有关登录的更多信息 AWS，请参阅《AWS 登录 用户指南》中的如何登录到您 AWS 账户的。

如果您 AWS 以编程方式访问，则会 AWS 提供软件开发套件 (SDK) 和命令行接口 (CLI)，以便使用您的凭据对请求进行加密签名。如果您不使用 AWS 工具，则必须自己签署请求。有关使用推荐的方法自行签署请求的更多信息，请参阅《IAM 用户指南》中的用于签署 API 请求的AWS 签名版本 4。

无论使用何种身份验证方法，您都可能需要提供其他安全信息。例如， AWS 建议您使用多重身份验证 (MFA) 来提高账户的安全性。要了解更多信息，请参阅《AWS IAM Identity Center 用户指南》中的多重身份验证和《IAM 用户指南》中的 IAM 中的AWS 多重身份验证。

AWS 账户 root 用户

创建时 AWS 账户，首先要有一个登录身份，该身份可以完全访问账户中的所有资源 AWS 服务 和资源。此身份被称为 AWS 账户 root 用户，使用您创建账户时使用的电子邮件地址和密码登录即可访问该身份。强烈建议您不要使用根用户执行日常任务。保护好根用户凭证，并使用这些凭证来执行仅根用户可以执行的任务。有关要求您以根用户身份登录的任务的完整列表，请参阅 IAM 用户指南中的需要根用户凭证的任务。

联合身份

作为最佳实践，要求人类用户（包括需要管理员访问权限的用户）使用与身份提供商的联合身份验证 AWS 服务 通过临时证书进行访问。

联合身份是指您的企业用户目录、Web 身份提供商、Identity C enter 目录中的用户，或者任何使用 AWS 服务 通过身份源提供的凭据进行访问的用户。 AWS Directory Service当联合身份访问时 AWS 账户，他们将扮演角色，角色提供临时证书。

要集中管理访问权限，建议您使用 AWS IAM Identity Center。您可以在 IAM Identity Center 中创建用户和群组，也可以连接并同步到您自己的身份源中的一组用户和群组，以便在您的所有 AWS 账户 和

应用程序中使用。有关 IAM Identity Center 的信息，请参阅 AWS IAM Identity Center 用户指南中的什么是 IAM Identity Center？。

IAM 用户和群组

I AM 用户是您 AWS 账户 内部对个人或应用程序具有特定权限的身份。在可能的情况下，我们建议使用临时凭证，而不是创建具有长期凭证（如密码和访问密钥）的 IAM 用户。但是，如果您有一些特定的使用场景需要长期凭证以及 IAM 用户，建议您轮换访问密钥。有关更多信息，请参阅《IAM 用户指南》中的对于需要长期凭证的用例，应在需要时更新访问密钥。

IAM 组是一个指定一组 IAM 用户的身份。您不能使用组的身份登录。您可以使用组来一次性为多个用户指定权限。如果有大量用户，使用组可以更轻松地管理用户权限。例如，您可以拥有一个名为的群组，IAMAdmins并向该群组授予管理 IAM 资源的权限。

用户与角色不同。用户唯一地与某个人员或应用程序关联，而角色旨在让需要它的任何人代入。用户具有永久的长期凭证，而角色提供临时凭证。要了解更多信息，请参阅《IAM 用户指南》中的 IAM 用户的使用案例。

IAM 角色

I AM 角色是您内部具有特定权限 AWS 账户 的身份。它类似于 IAM 用户，但与特定人员不关联。要在中临时担任 IAM 角色 AWS Management Console，您可以从用户切换到 IAM 角色（控制台）。您可以通过调用 AWS CLI 或 AWS API 操作或使用自定义 URL 来代入角色。有关使用角色的方法的更多信息，请参阅《IAM 用户指南》中的代入角色的方法。

具有临时凭证的 IAM 角色在以下情况下很有用：

- 联合用户访问：要向联合身份分配权限，请创建角色并为角色定义权限。当联合身份进行身份验证时，该身份将与角色相关联并被授予由此角色定义的权限。有关用于联合身份验证的角色的信息，请参阅《IAM 用户指南》中的针对第三方身份提供商创建角色（联合身份验证）。如果您使用 IAM Identity Center，则需要配置权限集。为控制您的身份在进行身份验证后可以访问的内容，IAM Identity Center 将权限集与 IAM 中的角色相关联。有关权限集的信息，请参阅《AWS IAM Identity Center 用户指南》中的权限集。
- 临时 IAM 用户权限：IAM 用户可代入 IAM 用户或角色，以暂时获得针对特定任务的不同权限。
- 跨账户存取：您可以使用 IAM 角色以允许不同账户中的某个人（可信主体）访问您的账户中的资源。角色是授予跨账户访问权限的主要方式。但是，对于某些资源 AWS 服务，您可以将策略直接附加到资源（而不是使用角色作为代理）。要了解用于跨账户访问的角色和基于资源的策略之间的差别，请参阅 IAM 用户指南中的 IAM 中的跨账户资源访问。

- 跨服务访问 — 有些 AWS 服务 使用其他 AWS 服务服务中的功能。例如，当您在服务中拨打电话时，该服务通常会在 Amazon 中运行应用程序 EC2 或在 Amazon S3 中存储对象。服务可能会使用发出调用的主体的权限、使用服务角色或使用服务相关角色来执行此操作。
  - 转发访问会话 (FAS) — 当您使用 IAM 用户或角色在中执行操作时 AWS，您被视为委托人。使用某些服务时，您可能会执行一个操作，然后此操作在其他服务中启动另一个操作。FAS 使用调用委托人的权限以及 AWS 服务 向下游服务发出请求的请求。 AWS 服务只有当服务收到需要与其他 AWS 服务 或资源交互才能完成的请求时，才会发出 FAS 请求。在这种情况下，您必须具有执行这两项操作的权限。有关发出 FAS 请求时的策略详情，请参阅转发访问会话。
  - 服务角色 - 服务角色是服务代表您在您的账户中执行操作而分派的 IAM 角色。IAM 管理员可以在 IAM 中创建、修改和删除服务角色。有关更多信息，请参阅《IAM 用户指南》中的创建向 AWS 服务委派权限的角色。
  - 服务相关角色-服务相关角色是一种与服务相关联的服务角色。 AWS 服务服务可以代入代表您执行操作的角色。服务相关角色出现在您的中 AWS 账户 ，并且归服务所有。IAM 管理员可以查看但不能编辑服务相关角色的权限。
- 在 A@@ mazon 上运行的应用程序 EC2 — 您可以使用 IAM 角色管理在 EC2 实例上运行并发出 AWS CLI 或 AWS API 请求的应用程序的临时证书。这比在 EC2 实例中存储访问密钥更可取。要为 EC2 实例分配 AWS 角色并使其可供其所有应用程序使用，您需要创建一个附加到该实例的实例配置文件。实例配置文件包含角色并允许在 EC2 实例上运行的程序获得临时证书。有关更多信息，请参阅 IAM 用户指南中的使用 IAM 角色向在 A mazon EC2 实例上运行的应用程序授予权限。

使用策略管理访问

您可以 AWS 通过创建策略并将其附加到 AWS 身份或资源来控制中的访问权限。策略是其中的一个对象 AWS ，当与身份或资源关联时，它会定义其权限。 AWS 在委托人（用户、root 用户或角色会话）发出请求时评估这些策略。策略中的权限确定是允许还是拒绝请求。大多数策略都以 JSON 文档的 AWS 形式存储在中。有关 JSON 策略文档的结构和内容的更多信息，请参阅 IAM 用户指南中的 JSON 策略概览。

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

默认情况下，用户和角色没有权限。要授予用户对所需资源执行操作的权限，IAM 管理员可以创建 IAM 策略。管理员随后可以向角色添加 IAM 策略，用户可以代入角色。

IAM 策略定义操作的权限，无关乎您使用哪种方法执行操作。例如，假设您有一个允许 `iam:GetRole` 操作的策略。拥有该策略的用户可以从 AWS Management Console AWS CLI、或 AWS API 获取角色信息。

基于身份的策略

基于身份的策略是可附加到身份（如 IAM 用户、用户组或角色）的 JSON 权限策略文档。这些策略控制用户和角色可在何种条件下对哪些资源执行哪些操作。要了解如何创建基于身份的策略，请参阅《IAM 用户指南》中的使用客户托管策略定义自定义 IAM 权限。

基于身份的策略可以进一步归类为内联策略或托管式策略。内联策略直接嵌入单个用户、组或角色中。托管策略是独立的策略，您可以将其附加到中的多个用户、群组和角色 AWS 账户。托管策略包括 AWS 托管策略和客户托管策略。要了解如何在托管策略和内联策略之间进行选择，请参阅《IAM 用户指南》中的在托管策略与内联策略之间进行选择。

基于资源的策略

基于资源的策略是附加到资源的 JSON 策略文档。基于资源的策略的示例包括 IAM 角色信任策略和 Amazon S3 存储桶策略。在支持基于资源的策略的服务中，服务管理员可以使用它们来控制对特定资源的访问。对于在其中附加策略的资源，策略定义指定主体可以对该资源执行哪些操作以及在什么条件下执行。您必须在基于资源的策略中指定主体。委托人可以包括账户、用户、角色、联合用户或 AWS 服务。

基于资源的策略是位于该服务中的内联策略。您不能在基于资源的策略中使用 IAM 中的 AWS 托管策略。

访问控制列表 (ACLs)

访问控制列表 (ACLs) 控制哪些委托人（账户成员、用户或角色）有权访问资源。 ACLs 与基于资源的策略类似，尽管它们不使用 JSON 策略文档格式。

Amazon S3 和 Amazon VPC 就是支持的服务示例 ACLs。 AWS WAF要了解更多信息 ACLs，请参阅《亚马逊简单存储服务开发者指南》中的访问控制列表 (ACL) 概述。

其他策略类型

AWS 支持其他不太常见的策略类型。这些策略类型可以设置更常用的策略类型向您授予的最大权限。

- 权限边界：权限边界是一个高级特征，用于设置基于身份的策略可以为 IAM 实体（IAM 用户或角色）授予的最大权限。您可为实体设置权限边界。这些结果权限是实体基于身份的策略及其权限边界的交集。在 `Principal` 中指定用户或角色的基于资源的策略不受权限边界限制。任一项策略中的显式拒绝将覆盖允许。有关权限边界的更多信息，请参阅IAM 用户指南中的 IAM 实体的权限边界。
- 服务控制策略 (SCPs) — SCPs 是 JSON 策略，用于指定中组织或组织单位 (OU) 的最大权限 AWS Organizations。 AWS Organizations 是一项用于对您的企业拥有的多 AWS 账户 项进行分组和集中

管理的服务。如果您启用组织中的所有功能，则可以将服务控制策略 (SCPs) 应用于您的任何或所有帐户。SCP 限制成员账户中的实体（包括每个 AWS 账户根用户实体）的权限。有关 Organization SCPs s 和的更多信息，请参阅《AWS Organizations 用户指南》中的服务控制策略。

- 资源控制策略 (RCPs) — RCPs 是 JSON 策略，您可以使用它来设置账户中资源的最大可用权限，而无需更新附加到您拥有的每个资源的 IAM 策略。RCP 限制成员账户中资源的权限，并可能影响身份（包括身份）的有效权限 AWS 账户根用户，无论这些身份是否属于您的组织。有关 Organizations 的更多信息 RCPs，包括 AWS 服务 该支持的列表 RCPs，请参阅《AWS Organizations 用户指南》中的资源控制策略 (RCPs)。
- 会话策略：会话策略是当您以编程方式为角色或联合用户创建临时会话时作为参数传递的高级策略。结果会话的权限是用户或角色的基于身份的策略和会话策略的交集。权限也可以来自基于资源的策略。任一项策略中的显式拒绝将覆盖允许。有关更多信息，请参阅 IAM 用户指南中的会话策略。

多个策略类型

当多个类型的策略应用于一个请求时，生成的权限更加复杂和难以理解。要了解在涉及多种策略类型时如何 AWS 确定是否允许请求，请参阅 IAM 用户指南中的策略评估逻辑。

如何 AWS HealthOmics 与 IAM 配合使用

在使用 IAM 管理对 AWS 的访问权限之前 HealthOmics，请先了解有哪些 IAM 功能可用于 AWS HealthOmics。

您可以搭配使用的 IAM 功能 AWS HealthOmics

IAM 功能	HealthOmics 支持
基于身份的策略	是
基于资源的策略	否
策略操作	是
策略资源	是
策略条件密钥	否
ACLs	否

IAM 功能	HealthOmics 支持
ABAC（策略中的标签）	是
临时凭证	是
主体权限	是
服务角色	是
服务相关角色	否

要全面了解 HealthOmics 以及其他 AWS 服务如何与大多数 IAM 功能配合使用，请参阅 IAM 用户指南中的与 IAM 配合使用的AWS 服务。

防止跨服务混淆代理

混淆代理问题是一个安全性问题，即不具有某操作执行权限的实体可能会迫使具有更高权限的实体执行该操作。在中 AWS，跨服务模仿可能会导致混乱的副手问题。一个服务（呼叫服务）调用另一项服务（所谓的服务）时，可能会发生跨服务模拟。可以操纵调用服务以使用其权限对另一个客户的资源进行操作，否则该服务不应有访问权限。为了防止这种情况， AWS 提供可帮助您保护所有服务的服务委托人数据的工具，这些服务委托人有权限访问账户中的资源。

我们建议在资源策略中使用aws:SourceArn和aws:SourceAccount全局条件上下文密钥来限制 AWS HealthOmics 向该资源提供的其他服务的权限。

为防止所担任的角色出现混淆副手问题 HealthOmics，请在角色的信任策略arn:aws:omics:*region*:*accountNumber*:*中aws:SourceArn将的值设置为。通配符 (*) 将条件应用于所有 HealthOmics 资源。

以下信任关系策略授予对您的资源的 HealthOmics 访问权限，并使用aws:SourceArn和aws:SourceAccount全局条件上下文密钥来防止出现混乱的副手问题。在为创建角色时使用此策略 HealthOmics。

JSON

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

```
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "omics.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "accountNumber"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:omics:region:accountNumber:*"
        }
      }
    }
  ]
}
```

基于身份的策略 HealthOmics

支持基于身份的策略：是

基于身份的策略是可附加到身份（如 IAM 用户、用户组或角色）的 JSON 权限策略文档。这些策略控制用户和角色可在何种条件下对哪些资源执行哪些操作。要了解如何创建基于身份的策略，请参阅《IAM 用户指南》中的使用客户管理型策略定义自定义 IAM 权限。

通过使用 IAM 基于身份的策略，您可以指定允许或拒绝的操作和资源以及允许或拒绝操作的条件。您无法在基于身份的策略中指定主体，因为它适用于其附加的用户或角色。要了解可在 JSON 策略中使用的所有元素，请参阅《IAM 用户指南》中的 IAM JSON 策略元素引用。

基于身份的策略示例 HealthOmics

要查看 AWS HealthOmics 基于身份的策略示例，请参阅。基于身份的策略示例 AWS HealthOmics

内部基于资源的政策 HealthOmics

支持基于资源的策略：否

基于资源的策略是附加到资源的 JSON 策略文档。基于资源的策略的示例包括 IAM 角色信任策略和 Amazon S3 存储桶策略。在支持基于资源的策略的服务中，服务管理员可以使用它们来控制对特定资源的访问。对于在其中附加策略的资源，策略定义指定主体可以对该资源执行哪些操作以及在什么条件下执行。您必须在基于资源的策略中指定主体。委托人可以包括账户、用户、角色、联合用户或 AWS 服务。

要启用跨账户访问，您可以将整个账户或其他账户中的 IAM 实体指定为基于资源的策略中的主体。将跨账户主体添加到基于资源的策略只是建立信任关系工作的一半而已。当委托人和资源处于不同位置时 AWS 账户，可信账户中的 IAM 管理员还必须向委托人实体（用户或角色）授予访问资源的权限。他们通过将基于身份的策略附加到实体以授予权限。但是，如果基于资源的策略向同一个账户中的主体授予访问权限，则不需要额外的基于身份的策略。有关更多信息，请参阅《IAM 用户指南》中的 IAM 中的跨账户资源访问。

的政策行动 HealthOmics

支持策略操作：是

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

JSON 策略的 `Action` 元素描述可用于在策略中允许或拒绝访问的操作。策略操作通常与关联的 AWS API 操作同名。有一些例外情况，例如没有匹配 API 操作的仅限权限 操作。还有一些操作需要在策略中执行多个操作。这些附加操作称为相关操作。

在策略中包含操作以授予执行关联操作的权限。

要查看 HealthOmics 操作列表，请参阅《服务授权参考》 HealthOmics 中的 AWS 定义的操作。

正在执行的策略操作在操作前 HealthOmics 使用以下前缀：

```
omics
```

要在单个语句中指定多项操作，请使用逗号将它们隔开。

```
"Action": [
      "omics:action1",
      "omics:action2"
         ]
```

要查看 AWS HealthOmics 基于身份的策略示例，请参阅。基于身份的策略示例 AWS HealthOmics

的政策资源 HealthOmics

支持策略资源：是

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

`Resource` JSON 策略元素指定要向其应用操作的一个或多个对象。语句必须包含 `Resource` 或 `NotResource` 元素。作为最佳实践，请使用其 Amazon 资源名称（ARN）指定资源。对于支持特定资源类型（称为资源级权限）的操作，您可以执行此操作。

对于不支持资源级权限的操作（如列出操作），请使用通配符（*）指示语句应用于所有资源。

```
"Resource": "*"
```

要查看 HealthOmics 资源类型及其列表 ARNs，请参阅《服务授权参考》 HealthOmics 中的 AWS 定义的资源。要了解您可以使用哪些操作来指定每种资源的 ARN，请参阅 AWS 定义的操作。
HealthOmics

要查看 AWS HealthOmics 基于身份的策略示例，请参阅。基于身份的策略示例 AWS HealthOmics

的策略条件密钥 HealthOmics

中不支持策略条件密钥 HealthOmics。

中的访问控制列表 (ACLs) HealthOmics

支持 ACLs：否

访问控制列表 (ACLs) 控制哪些委托人（账户成员、用户或角色）有权访问资源。 ACLs 与基于资源的策略类似，尽管它们不使用 JSON 策略文档格式。

基于属性的访问控制 (ABAC) HealthOmics

支持 ABAC（策略中的标签）：是

基于属性的访问控制（ABAC）是一种授权策略，该策略基于属性来定义权限。在中 AWS，这些属性称为标签。您可以向 IAM 实体（用户或角色）和许多 AWS 资源附加标签。标记实体和资源是 ABAC 的第一步。然后设计 ABAC 策略，以在主体的标签与他们尝试访问的资源标签匹配时允许操作。

ABAC 在快速增长的环境中非常有用，并在策略管理变得繁琐的情况下可以提供帮助。

要基于标签控制访问，您需要使用 `aws:ResourceTag/key-name`、`aws:RequestTag/key-name` 或 `aws:TagKeys` 条件键在策略的条件元素中提供标签信息。

如果某个服务对于每种资源类型都支持所有这三个条件键，则对于该服务，该值为是。如果某个服务仅对于部分资源类型支持所有这三个条件键，则该值为部分。

有关 ABAC 的更多信息，请参阅《IAM 用户指南》中的使用 ABAC 授权定义权限。要查看设置 ABAC 步骤的教程，请参阅《IAM 用户指南》中的使用基于属性的访问权限控制（ABAC）。

有关标记 HealthOmics 资源的更多信息，请参阅在中标记资源 HealthOmics。

以下示例说明如何编写 IAM 策略，拒绝访问没有特定标签的资源。

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "omics:*"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "Null": {
                  "aws:RequestTag/MyCustomTag": "true"
                }
            }
        }
    ]
}
```

将临时证书与 HealthOmics

支持临时凭证：是

当你使用临时证书登录时，有些 AWS 服务 不起作用。有关更多信息，包括哪些 AWS 服务 适用于临时证书，请参阅 IAM 用户指南中的AWS 服务 与 IA M 配合使用的信息。

如果您使用除用户名和密码之外的任何方法登录，则 AWS Management Console 使用的是临时证书。例如，当您 AWS 使用公司的单点登录 (SSO) 链接进行访问时，该过程会自动创建临时证书。当您以用户身份登录控制台，然后切换角色时，您还会自动创建临时凭证。有关切换角色的更多信息，请参阅《IAM 用户指南》中的从用户切换到 IAM 角色（控制台）。

您可以使用 AWS CLI 或 AWS API 手动创建临时证书。然后，您可以使用这些临时证书进行访问 AWS。 AWS 建议您动态生成临时证书，而不是使用长期访问密钥。有关更多信息，请参阅 IAM 中的临时安全凭证。

的跨服务主体权限 HealthOmics

支持转发访问会话（FAS）：是

当您使用 IAM 用户或角色在中执行操作时 AWS，您被视为委托人。使用某些服务时，您可能会执行一个操作，然后此操作在其他服务中启动另一个操作。FAS 使用调用委托人的权限以及 AWS 服务 向下游服务发出请求的请求。 AWS 服务只有当服务收到需要与其他 AWS 服务 或资源交互才能完成的请求时，才会发出 FAS 请求。在这种情况下，您必须具有执行这两项操作的权限。有关发出 FAS 请求时的策略详细信息，请参阅转发访问会话。

HealthOmics 的服务角色

支持服务角色：是

服务角色是由一项服务担任、代表您执行操作的 IAM 角色。IAM 管理员可以在 IAM 中创建、修改和删除服务角色。有关更多信息，请参阅《IAM 用户指南》中的创建向 AWS 服务委派权限的角色。

> **Warning**
>
> 更改服务角色的权限可能会中断 HealthOmics 功能。只有在 HealthOmics 提供操作指导时才编辑服务角色。

的服务相关角色 HealthOmics

支持服务相关角色：否

服务相关角色是一种链接到的服务角色。 AWS 服务服务可以代入代表您执行操作的角色。服务相关角色出现在您的中 AWS 账户 ，并且归服务所有。IAM 管理员可以查看但不能编辑服务相关角色的权限。

有关创建或管理服务相关角色的详细信息，请参阅能够与 IAM 搭配使用的AWS 服务。在表中查找服务相关角色列中包含 Yes 的表。选择是链接以查看该服务的服务相关角色文档。

基于身份的策略示例 AWS HealthOmics

默认情况下，用户和角色无权创建或修改 AWS HealthOmics 资源。他们也无法使用 AWS Management Console、 AWS Command Line Interface (AWS CLI) 或 AWS API 执行任务。要授予用户对所需资源执行操作的权限，IAM 管理员可以创建 IAM 策略。管理员随后可以向角色添加 IAM 策略，用户可以代入角色。

要了解如何使用这些示例 JSON 策略文档创建基于 IAM 身份的策略，请参阅《IAM 用户指南》中的创建 IAM 策略（控制台）。

有关 AWS HealthOmics 定义的操作和资源类型（包括每种资源类型的格式）的详细信息，请参阅《服务授权参考》 HealthOmics 中的 AWS 操作、资源和条件密钥。 ARNs

主题

- 策略最佳实践
- 使用 HealthOmics 控制台
- 允许用户查看他们自己的权限

策略最佳实践

基于身份的策略决定是否有人可以在您的账户中创建、访问或删除 AWS HealthOmics 资源。这些操作可能会使 AWS 账户产生成本。创建或编辑基于身份的策略时，请遵循以下指南和建议：

- 开始使用 AWS 托管策略并转向最低权限权限 — 要开始向用户和工作负载授予权限，请使用为许多常见用例授予权限的AWS 托管策略。它们在你的版本中可用 AWS 账户。我们建议您通过定义针对您的用例的 AWS 客户托管策略来进一步减少权限。有关更多信息，请参阅《IAM 用户指南》中的AWS 托管式策略或工作职能的AWS 托管式策略。
- 应用最低权限：在使用 IAM 策略设置权限时，请仅授予执行任务所需的权限。为此，您可以定义在特定条件下可以对特定资源执行的操作，也称为最低权限许可。有关使用 IAM 应用权限的更多信息，请参阅《IAM 用户指南》中的 IAM 中的策略和权限。

- 使用 IAM 策略中的条件进一步限制访问权限：您可以向策略添加条件来限制对操作和资源的访问。例如，您可以编写策略条件来指定必须使用 SSL 发送所有请求。如果服务操作是通过特定的方式使用的，则也可以使用条件来授予对服务操作的访问权限 AWS 服务，例如 AWS CloudFormation。有关更多信息，请参阅《IAM 用户指南》中的 IAM JSON 策略元素：条件。
- 使用 IAM Access Analyzer 验证您的 IAM 策略，以确保权限的安全性和功能性 – IAM Access Analyzer 会验证新策略和现有策略，以确保策略符合 IAM 策略语言（JSON）和 IAM 最佳实践。IAM Access Analyzer 提供 100 多项策略检查和可操作的建议，以帮助您制定安全且功能性强的策略。有关更多信息，请参阅《IAM 用户指南》中的使用 IAM Access Analyzer 验证策略。
- 需要多重身份验证 (MFA)-如果 AWS 账户您的场景需要 IAM 用户或根用户，请启用 MFA 以提高安全性。若要在调用 API 操作时需要 MFA，请将 MFA 条件添加到您的策略中。有关更多信息，请参阅《IAM 用户指南》中的使用 MFA 保护 API 访问。

有关 IAM 中的最佳实操的更多信息，请参阅 IAM 用户指南中的 IAM 中的安全最佳实操。

使用 HealthOmics 控制台

要访问 AWS HealthOmics 控制台，您必须拥有一组最低权限。这些权限必须允许您在中列出和查看有关 AWS HealthOmics 资源的详细信息 AWS 账户。如果创建比必需的最低权限更为严格的基于身份的策略，对于附加了该策略的实体（用户或角色），控制台将无法按预期正常运行。

对于仅调用 AWS CLI 或 AWS API 的用户，您无需为其设置最低控制台权限。相反，只允许访问与其尝试执行的 API 操作相匹配的操作。

允许用户查看他们自己的权限

该示例说明了您如何创建策略，以允许 IAM 用户查看附加到其用户身份的内联和托管式策略。此策略包括在控制台上或使用 AWS CLI 或 AWS API 以编程方式完成此操作的权限。

```
{
    "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 的托管策略 AWS HealthOmics

AWS 托管策略是由创建和管理的独立策略 AWS。 AWS 托管策略旨在为许多常见用例提供权限，以便您可以开始为用户、组和角色分配权限。

请记住， AWS 托管策略可能不会为您的特定用例授予最低权限权限，因为它们可供所有 AWS 客户使用。我们建议通过定义特定于您的使用场景的客户管理型策略来进一步减少权限。

您无法更改 AWS 托管策略中定义的权限。如果 AWS 更新 AWS 托管策略中定义的权限，则更新会影响该策略所关联的所有委托人身份（用户、组和角色）。 AWS 最有可能在启动新的 API 或现有服务可以使用新 AWS 服务 的 API 操作时更新 AWS 托管策略。

有关更多信息，请参阅《IAM 用户指南》中的 AWS 托管策略。

AWS 托管策略： AmazonOmicsFullAccess

您可以将该AmazonOmicsFullAccess策略附加到您的 IAM 身份，以授予其完全访问权限 HealthOmics。

此策略授予对所有 HealthOmics 操作的完全访问权限。当您创建注释或变体存储时，Omics 还将通过资源访问管理器 (RAM) 控制台中的资源共享邀请向您提供访问该存储的权限。有关通过 Lake Formation 邀请资源共享的更多信息，请参阅 Lake Formati on 中的跨账户数据共享。对于 Omics 管理策略，您还需要以下权限才能访问您的 Amazon S3 存储桶。

- 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 托管策略： AmazonOmicsReadOnlyAccess

如果您希望将该身份的权限限制为只读访问权限，则可以将该AWSOmicsReadOnlyAccess策略附加到您的 IAM 身份。

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "omics:Get*",
                "omics:List*"
            ],
            "Resource": "*"
        }
    ]
}
```

HealthOmics AWS 托管策略的更新

查看 HealthOmics 自该服务开始跟踪这些更改以来 AWS 托管策略更新的详细信息。要获得有关此页面变更的自动提醒，请订阅 “ HealthOmics 文档历史记录” 页面上的 RSS feed。

更改	描述	日期
AmazonOmicsFullAccess -添加了新政策	HealthOmics 添加了一项新策略，以授予用户对所有操作和资源的完全访问权限。要了解更多信息，请参阅AmazonOmicsFullAccess。	2023 年 2 月 23 日
HealthOmics 开始跟踪更改	HealthOmics 开始跟踪其 AWS 托管策略的更改。	2022 年 11 月 29 日
AmazonOmicsReadOnlyAccess -添加了新政策	HealthOmics 添加了将访问权限限制为只读的新策略。要了解更多信息，AmazonOmicsReadOnlyAccess。	2022 年 11 月 29 日

对 AWS HealthOmics 身份和访问进行故障排除

使用以下信息来帮助您诊断和修复在使用 AWS HealthOmics 和 IAM 时可能遇到的常见问题。

主题

- 我无权在以下位置执行操作 HealthOmics
- 我无权执行 iam：PassRole
- 我想允许我以外的人 AWS 账户 访问我的 HealthOmics 资源

我无权在以下位置执行操作 HealthOmics

如果您收到错误提示，指明您无权执行某个操作，则必须更新策略以允许执行该操作。

当 mateojackson IAM 用户尝试使用控制台查看有关虚构 *my-example-widget* 资源的详细信息，但不拥有虚构 omics:*GetWidget* 权限时，会发生以下示例错误。

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 omics:GetWidget on resource: my-example-widget
```

在此情况下，必须更新 mateojackson 用户的策略，以允许使用 omics:*GetWidget* 操作访问 *my-example-widget* 资源。

如果您需要帮助，请联系您的 AWS 管理员。您的管理员是提供登录凭证的人。

我无权执行 iam：PassRole

如果您收到错误消息，提示您无权执行该iam:PassRole操作，则必须更新您的策略以允许您将角色传递给 AWS HealthOmics。

有些 AWS 服务 允许您将现有角色传递给该服务，而不是创建新的服务角色或服务相关角色。为此，您必须具有将角色传递到服务的权限。

当名为的 IAM 用户marymajor尝试使用控制台在 AWS 中执行操作时，会出现以下示例错误 HealthOmics。但是，服务必须具有服务角色所授予的权限才可执行此操作。Mary 不具有将角色传递到服务的权限。

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

在这种情况下，必须更新 Mary 的策略以允许她执行 iam:PassRole 操作。

如果您需要帮助，请联系您的 AWS 管理员。您的管理员是提供登录凭证的人。

我想允许我以外的人 AWS 账户 访问我的 HealthOmics 资源

您可以创建一个角色，以便其他账户中的用户或您组织外的人员可以使用该角色来访问您的资源。您可以指定谁值得信赖，可以代入角色。对于支持基于资源的策略或访问控制列表 (ACLs) 的服务，您可以使用这些策略向人们授予访问您的资源的权限。

要了解更多信息，请参阅以下内容：

- 要了解 AWS 是否 HealthOmics 支持这些功能，请参阅如何 AWS HealthOmics 与 IAM 配合使用。

- 要了解如何提供对您拥有的资源的访问权限 AWS 账户 ，请参阅 IAM 用户指南中的向您拥有 AWS 账户 的另一个 IAM 用户提供访问权限。
- 要了解如何向第三方提供对您的资源的访问权限 AWS 账户，请参阅 IAM 用户指南中的向第三方提供访问权限。 AWS 账户
- 要了解如何通过身份联合验证提供访问权限，请参阅《IAM 用户指南》中的为经过外部身份验证的用户（身份联合验证）提供访问权限。
- 要了解使用角色和基于资源的策略进行跨账户访问之间的差别，请参阅《IAM 用户指南》中的 IAM 中的跨账户资源访问。

合规性验证 AWS HealthOmics

AWS HealthOmics 作为多个合规计划的一部分，第三方审计师对安全性和 AWS 合规性进行评估。这包括 HIPAA、FedRAMP 等。下表显示了该 HealthOmics 服务的合规性认证。

认证	链接
HIPAA	符合 HIPAA 条件的服务参考
HiTrust-脑脊液	健康信息信托联盟共同安全框架
FedRAMP 中等（东部/西部）	联邦风险和授权管理计划
ISO/CSA STAR	ISO 和 CSA STAR 认证
C5	云计算合规性控制目录
国防部 CC SRG IL2	国防部云计算安全要求指南
ENS High	Nacional de Seguridad
FINMA	瑞士金融市场监管局
ISMAP	信息系统安全管理和评估计划
OSPAR	外包服务提供商的审计报告
PCI	支付卡行业数据安全标准
Pinakes	银行协会 CCI-第三方资格

认证	链接
PiTuKri	评估云服务信息安全的标准
SOC 1、2、3	系统和组织控制

有关特定合规计划范围内的所有 AWS 服务的列表，请参阅按合规计划划分的 AWS 范围内服务 AWS 按合规计划。有关常规信息，请参阅 AWS 合规性计划、、。

您可以使用下载第三方审计报告 AWS Artifact。有关更多信息，请参阅中的 “下载报告” 中的 “ AWS Artifact。

HealthOmics 数据存储使用示例 ID 进行内部文件命名和标记资源。在采集数据之前，请检查样本 ID 是否包含任何 PHI 数据。如果是，请在采集数据之前更改样本 ID。有关更多信息，请参阅 AWS HIPAA 合规性网页上的指南。

您在使用 AWS HealthOmics 时的合规责任取决于您的数据的敏感性、贵公司的合规目标以及适用的法律和法规。 AWS 提供了以下资源来帮助实现合规性：

- 安全性与合规性快速入门指南 - 这些部署指南讨论了架构注意事项，并提供了在 AWS上部署基于安全性和合规性的基准环境的步骤。
- HIPAA 安全与合规架构白皮书 — 本白皮书描述了公司如何使用来 AWS 创建符合 HIPAA 标准的应用程序。
- AWS 合AWS 规资源 — 此工作簿和指南集可能适用于您的行业和所在地区。
- 使用AWS Config 开发人员指南中的规则评估资源 — AWS Config; 评估您的资源配置在多大程度上符合内部实践、行业准则和法规。
- AWS Security Hub— 此 AWS 服务可全面了解您的安全状态 AWS ，帮助您检查是否符合安全行业标准和最佳实践。

韧性在 HealthOmics

AWS 全球基础设施是围绕 AWS 区域 可用区构建的。 AWS 区域 提供多个物理隔离和隔离的可用区，这些可用区通过低延迟、高吞吐量和高度冗余的网络连接。利用可用区，您可以设计和操作在可用区之间无中断地自动实现失效转移的应用程序和数据库。与传统的单个或多个数据中心基础结构相比，可用区具有更高的可用性、容错性和可扩展性。

有关 AWS 区域 和可用区的更多信息，请参阅AWS 全球基础设施。

除了 AWS 全球基础设施外，AWS 还 HealthOmics 提供多项功能来帮助支持您的数据弹性和备份需求。

AWS HealthOmics 和接口 VPC 终端节点 (AWS PrivateLink)

您可以通过创建接口 VPC 终端节点在您 AWS HealthOmics 的 VPC 和之间建立私有连接。接口终端节点由一项技术提供支持，无需互联网网关AWS PrivateLink、NAT 设备、VPN 连接或 AWS Direct Connect 连接，即可使用该技术私密访问 HealthOmics API 操作。您的 VPC 中的实例不需要公有 IP 地址即可与 HealthOmics API 操作通信。您的 VPC 和 VPC 之间的流量 HealthOmics 不会流出 Amazon 网络之外。

每个接口端点均由子网中的一个或多个弹性网络接口表示。

有关更多信息，请参阅 A mazon VPC 用户指南中的接口 VPC 终端节点 (AWS PrivateLink)。

除以色列（特拉维夫） HealthOmics 以外的所有地区均支持 VPC 终端节点策略。默认情况下，允许通过终端节点进行完全访问。 HealthOmics

HealthOmics VPC 终端节点的注意事项

在为设置接口 VPC 终端节点之前 HealthOmics，请务必查看 Amazon VPC 用户指南中的接口终端节点属性和限制。

HealthOmics 支持从您的 VPC 调用所有 HealthOmics 存储 API 操作。

HealthOmics 默认情况下不支持 VPC 终端节点策略，但您可以创建 VPC 终端节点以实现 HealthOmics 存储操作的完全 HealthOmics 访问权限。有关更多信息，请参阅《Amazon VPC User Guide》中的 Controlling access to services with VPC endpoints。

为 HealthOmics 创建接口 VPC 端点

您可以使用 Amazon VPC 控制台或 AWS Command Line Interface (AWS CLI) 为 HealthOmics 服务创建 VPC 终端节点。有关更多信息，请参阅《Amazon VPC User Guide》中的 Creating an interface endpoint。

使用以下服务名称为 HealthOmics 创建 VPC 终端节点：

- 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

美国东部（弗吉尼亚北部）和美国西部（俄勒冈）区域支持 AWS PrivateLink FIPS 终端节点。对于这些区域，您还可以使用以下服务名称：

- 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

如果您为终端节点开启私有 DNS，则可以使用该终端节点的默认 DNS 名称向发 HealthOmics 出 API 请求，例如，`omics.us-east-1.amazonaws.com`。

有关更多信息，请参阅《Amazon VPC 用户指南》中的通过接口端点访问服务。

为 HealthOmics 创建 VPC 端点策略

您可以为 VPC 端点附加控制对 HealthOmics 的访问的端点策略。该策略指定以下信息：

- 可执行操作的主体
- 可执行的操作
- 可对其执行操作的资源

有关更多信息，请参阅《Amazon VPC 用户指南》中的使用 VPC 端点控制对服务的访问权限。

示例：用于 HealthOmics 操作的 VPC 终端节点策略。

以下是的终端节点策略示例 HealthOmics。当连接到终端节点时，此策略授予所有委托人对所有资源 HealthOmics 执行操作的访问权限。

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\":\"*\"}]}"
```

使用 Amazon S3 访问读取集的特殊注意事项 URIs

要在使用私有连接 URIs 时通过 Amazon S3 访问读取集，请在序列存储上设置 PrivateLink接口终端节点。设置完毕后，端点将采用以下格式：

```
com.amazonaws.region.storage-omics
com.amazonaws.region.control-storage-omics
```

要使用网关终端节点，请按照 Amazon S3 网关终端节点指南配置您的网关终端节点。 HealthOmics 拥有 Amazon S3 存储桶，因此您无需创建或调整存储桶策略。网关终端节点依赖于附加到访问数据的用户或角色的策略，但您也可以使用更严格的策略配置终端节点。这些策略可能包括基于 Amazon S3 接入点 ARN 和 Amazon S3 操作的访问限制。

监控 AWS HealthOmics

监控是维护 AWS HealthOmics 和其他 AWS 解决方案的可靠性、可用性和性能的重要组成部分。AWS 提供以下监控工具，用于监视 AWS HealthOmics，在出现问题时进行报告，并在适当时自动采取措施：

- Amazon 会实时 CloudWatch监控您的 AWS 资源和您运行 AWS 的应用程序。您可以收集和跟踪指标，创建自定义的控制平面，以及设置警报以在指定的指标达到您指定的阈值时通知您或采取措施。例如，您可以 CloudWatch 跟踪您的 Amazon EC2 实例的 CPU 使用率或其他指标，并在需要时自动启动新实例。有关更多信息，请参阅 Amazon CloudWatch 用户指南。
- Amazon Lo CloudWatch gs 使您能够监控、存储和访问来自亚马逊 EC2 实例和其他来源的日志文件。 CloudTrail CloudWatch 日志可以监视日志文件中的信息，并在达到特定阈值时通知您。您还可以在高持久性存储中检索您的日志数据。有关更多信息，请参阅 Amazon CloudWatch 日志用户指南。
- AWS CloudTrail 捕获由您的 AWS 账户 或代表该账户发出的 API 调用和相关事件，并将日志文件传输到您指定的 Amazon S3 桶。您可以标识哪些用户和账户调用了 AWS、发出调用的源 IP 地址以及调用的发生时间。有关更多信息，请参阅 AWS CloudTrail 《用户指南》。
- Amazon EventBridge 是一项无服务器事件总线服务，可以轻松地将您的应用程序与来自各种来源的数据连接起来。 EventBridge 提供来自您自己的应用程序、 Software-as-a-Service (SaaS) 应用程序和 AWS 服务的实时数据流，并将这些数据路由到 Lambda 等目标。这使您能够监控服务中发生的事件，并构建事件驱动的架构。有关更多信息，请参阅 Amazon EventBridge 用户指南。

> **Note**
>
> 要获取服务更新，请配置和监控您的 Person al Health Dashboar d。有关如何管理控制面板的更多信息，请参阅 AWS Health 控制面板入门。

主题

S3 访问日志

您可以使用商店创建的访问日志监控 Amazon S3 API 对 HealthOmics 序列存储数据的访问权限。您可以使用 CloudWatch 来监控 HealthOmics API 操作中的 S3 访问权限。 CloudWatch 提供对来自您自己账户的 Amazon S3 访问权限的可见性。如果您作为数据所有者共享对第三方帐户的访问权限，则访问记录不可用 CloudWatch。取而代之的是使用商店的 S3 访问日志。该日志记录了 S3 对已配置的 Amazon S3 存储桶中数据的所有 S3 访问权限。

使用`CreateSequenceStore`或 `UpdateSequenceStore` API 操作配置 S3 访问日志。此外，请确保 HealthOmics 服务主体 (`omics.amazonaws.com`) 拥有对配置的 S3 前缀的`s3:PutObject`权限。

> **Note**
>
> 日志使用目标存储桶的默认加密配置。如果存储桶使用客户托管密钥，则服务委托人必须有权使用该密钥进行写入。

要关闭访问日志记录，请使用访问日志配置`UpdateSequenceStore`并将其设置为空白。

HealthOmics 使用 CloudWatch 指标进行监控

您可以使用 HealthOmics 进行监控 CloudWatch，它收集原始数据并将其处理为可读的近乎实时的指标。这些统计数据会保存 15 个月，从而使您能够访问历史信息，并能够更好地了解您的 Web 应用程序或服务的执行情况。此外，可以设置用于监测特定阈值的警报，并在达到相应阈值时发送通知或执行操作。有关更多信息，请参阅 Amazon CloudWatch 用户指南。

该 AWS HealthOmics 服务报告`AWS/Omics`命名空间中的以下指标。

报告了以下各项的 API 调用次数指标 AWS HealthOmics APIs。仅报告 API 操作维度。

- 参考和参考资料库 APIs — CreateReferenceStore、 DeleteReferenceStore、StartReferenceImportJob
- 序列存储和读取集 APIs — CreateSequenceStore DeleteSequenceStore、StartReadSetImportJob、 StartReadSetActivationJob、 StartReadSetExportJob
- 变体商店 APIs — CreateVariantStore、 DeleteVariantStore、 StartVariantImportJob、CancelVariantImportJob
- 注释存储 APIs — CreateAnnotationStore、 DeleteAnotationStore、 StartAnnotationImportJob、CancelAnnotationImportJob

- 工作流程、运行和运行组 APIs — CreateWorkflow DeleteWorkflow、 StartRun、 CancelRun、 DeleteRun、 CreateRunGroup、 DeleteRunGroup

查看 *AWS HealthOmics* 指标

CloudWatch 的 AWS HealthOmics 指标可在 CloudWatch 控制台中查看。

查看指标（CloudWatch 控制台）

1. 登录 AWS 管理控制台并打开CloudWatch 控制台。
2. 选择 “指标”，选择 “所有指标”，然后选择 “AWS/ 使用情况”。
3. 的筛选服务 AWS HealthOmics.
4. 选择维度、指标名称，然后选择 添加到图表。
5. 选择日期范围的值。所选日期范围的指标计数将显示在该图表中。

使用创建警报 CloudWatch

CloudWatch 警报在指定时间段内监视单个指标，并执行一项或多项操作：向亚马逊简单通知服务 (Amazon SNS) Simple Notification Service 主题或 Auto Scaling 策略发送通知。一个或多个操作基于指标在您指定的多个时间段内相对于给定阈值的值。 CloudWatch 还可以在警报状态发生变化时向您发送 Amazon SNS 消息。

CloudWatch 警报仅在状态发生变化并且持续到您指定的时间段内时才会调用操作。

查看指标（CloudWatch 控制台）

1. 登录 AWS 管理控制台并打开CloudWatch 控制台。
2. 依次选择 Alarms 和 Create Alarm。
3. 选择 AWS/ Usage，然后使用服务维度选择一个 AWS HealthOmics 指标。
4. 对于 Time Range，请选择要监控的时间范围，然后选择 Next。
5. 输入名称和描述。
6. 对于 Whenever，选择 >=，然后键入一个最大值。
7. 如果 CloudWatch 要在达到警报状态时发送电子邮件，请在 “操作” 部分的 “每当此警报” 中选择 “状态为警报”。在 “发送通知至” 中，选择一个邮件列表或选择 “新建列表” 并创建新的邮件列表。
8. 预览警报预览部分中的警报。如果对警报满意，请选择 Create Alarm (创建警报)。

HealthOmics 使用 CloudWatch 日志进行监控

HealthOmics 生成各种日志，以帮助您了解运行情况并对其进行故障排除。日志可在两个地方找到：CloudWatch 和 Amazon S3。

默认情况下，运行会开启日志记录。您可以选择通过在startrun请求`LogLevel = OFF`中设置来关闭运行的日志记录。

> **Note**
>
> 要获取服务更新，请配置和监控您的 Person al Health Dashboar d。有关如何管理控制面板的更多信息，请参阅 AWS Health 控制面板入门。

主题

HealthOmics 工作流程的日志类型

HealthOmics 为工作流提供以下类型的日志：

- 引擎日志 — 底层工作流引擎（Nextflow、WDL 和 CWL）生成运行的引擎日志。这些日志可以帮助您解决工作流程定义问题。
- 运行清单日志 — 这些日志提供有关每个运行任务的高级信息，例如任务状态、开始时间、停止时间和失败原因（如果任务失败）。

  运行清单日志还会报告资源利用率统计信息，这有助于了解资源优化机会。这些统计数据包括：

  - CPU 平均值
  - CPU 最大值
  - CPUSRerved
  - gpusReserv
  - memoryAverageGiB，

  - memoryMaximumGiB，
  - memoryReservedGiB，
  - 运行秒
- 运行日志-运行日志提供总体运行状态以及各个任务的启动、运行、停止和完成时间。运行日志还使您可以查看文件导入和导出步骤。
- 任务日志-任务日志提供有关运行中各个任务的详细日志信息。任务日志中的输出取决于任务定义以及您在代码中使用日志语句的位置。如果您的任务日志无法提供所需的洞察级别，请考虑在任务定义中添加其他日志语句，以生成更具洞察力的任务日志。
- 运行缓存日志-运行缓存日志提供运行缓存的总体状态和任务输出的缓存。通过运行缓存日志，您可以查看每次使用缓存的运行的缓存命中和未命中。
- outputs.json — 对于 WDL 和 CWL 工作流程，在运行完成后将引擎生成的名为的文件 HealthOmics 传送到`outputs.json`您的 Amazon S3 存储桶。该文件包括运行的所有输出的列表和地图。

登录 CloudWatch

您可以在以下日志组中找到 HealthOmics CloudWatch 工作流程日志：`/aws/omics/WorkflowLog`。此外，get-run API 操作的输出还提供了引擎 CloudWatch 日志和运行日志 ARNs 的日志流。

默认情况下，无限期 AWS 保留 CloudWatch 日志。您可以调整日志组的保留策略，将保留期设置为 10 年到 1 天之间。

下表提供了 CloudWatch 登录的摘要 HealthOmics。

日志名称	在 CloudWatch 日志中可用	日志何时可用	日志流格式
引擎日志	是，对于失败的运行	运行完成后	run/ /engin *runID* e
运行清单日志	是	运行完成后	manifest/run/ *runID* *runUUID*
运行日志	是	实时	运行/ *runID*
任务日志	是	实时	run/ /task/ *runID* *taskID*

日志名称	在 CloudWatch 日志中可用	日志何时可用	日志流格式
运行缓存日志	是	实时	runCache/ /*runCacheId runCacheUUID*
outputs.json（WDL 和 CWL）	否	不适用	不适用

登录亚马逊 S3

运行完成后，引擎日志将传送到您的 S3 存储桶，并且可以无限期使用，直到您将其删除。这些日志位于您为工作流程指定的 S3 输出 URI 的日志目录中。

日志目录的路径采用以下格式：`s3://{user_provided_path}/logs/`。

下表提供了您的 Amazon S3 存储桶中可用 HealthOmics 日志的摘要。

日志名称	在亚马逊 S3 中可用	日志何时可用	日志流路径
引擎日志	是	运行完成后	s3://*user_provided_path* /logs/engine.log
outputs.json（WDL 和 CWL）	是	运行完成后	s3: /////logs *user_provided_path runID runUUID* /outputs.json
运行清单日志、运行日志和任务日志	否	不适用	不适用

CLI 中的交互式 CloudWatch 日志

您可以在交互模式下使用 Live Tail 命令以交互方式查看 CloudWatch 日志。您可以实时跟踪运行进度，并定义最多 5 个关键字以在日志中突出显示：

```
aws logs start-live-tail  \
  --mode interactive  \
  --log-group-identifiers arn:aws:logs:region:account-ID:log-group:/aws/omics/
WorkflowLog
```

有关更多信息，请参阅《 AWS CLI 命令参考》中的 Start live tail。

从控制台访问 CloudWatch 日志

要访问运行日志，您可以直接从 HealthOmics 控制台的运行详细信息页面链接到这些日志。

1. 打开 HealthOmics 管理控制台。
2. 在左侧导航窗格中，选择 R uns。
3. 从 “运行” 表中选择运行。
4. 在运行详细信息页面中，您可以选择以下任一操作：

   a. 在运行摘要中，选择查看运行日志。控制台在控制台中打开运行日志。 CloudWatch

   b. 从运行摘要中，选择在 Amazon S3 中查看日志。控制台在 Amazon S3 控制台中打开日志文件夹。

   c. 在运行任务中，选择查看日志、查看运行日志或查看任务的运行清单日志。控制台在控制台中打开日志。 CloudWatch

您也可以从 CloudWatch 控制台导航到日志：

1. 打开控制 CloudWatch 台https://console.aws.amazon.com/cloudwatch/。
2. 从左侧菜单中选择 “日志组”。
3. 选择 `/aws/omics/WorkflowLog` 组。

   如果日志组列表很长，则可以在搜索文本框中输入 omics 来缩小列表范围。
4. 当日志组详细信息页面打开时，选择要查看的日志流。控制台显示此日志流的事件。

使用记录 AWS HealthOmics API 调用 AWS CloudTrail

AWS HealthOmics 与 AWS CloudTrail一项服务集成，该服务提供用户、角色或 AWS 服务在中执行的操作的记录 HealthOmics。 CloudTrail 将所有 API 调用捕获 HealthOmics 为事件。捕获的调用包括来自 HealthOmics 控制台的调用和对 HealthOmics API 操作的代码调用。如果您创建了跟踪，则可以允许将 CloudTrail 事件持续传输到 Amazon S3 存储桶，包括的事件 HealthOmics。如果您未配置跟踪，您仍然可以在 CloudTrail 控制台的 “事件历史记录” 中查看最新的事件。使用收集的信息 CloudTrail，您可以确定向哪个请求发出 HealthOmics、发出请求的 IP 地址、谁发出了请求、何时发出请求以及其他详细信息。

要了解更多信息 CloudTrail，请参阅《AWS CloudTrail 用户指南》。

HealthOmics 信息在 CloudTrail

CloudTrail 在您创建账户 AWS 账户 时已在您的账户上启用。当活动发生在中时 HealthOmics，该活动会与其他 AWS 服务 CloudTrail 事件一起记录在事件历史记录中。您可以在中查看、搜索和下载最近发生的事件 AWS 账户。有关更多信息，请参阅使用事件历史记录查看 CloudTrail 事件。

要持续记录您的 AWS 账户事件（包括的事件） HealthOmics，请创建跟踪。跟踪允许 CloudTrail 将日志文件传输到 Amazon S3 存储桶。预设情况下，在控制台中创建跟踪记录时，此跟踪记录应用于所有 AWS 区域。跟踪记录 AWS 分区中所有区域的事件，并将日志文件传送到您指定的 Amazon S3 存储桶。此外，您可以配置其他 AWS 服务，以进一步分析和处理 CloudTrail 日志中收集的事件数据。有关更多信息，请参阅下列内容：

- 创建跟踪记录概述
- CloudTrail 支持的服务和集成
- 配置 Amazon SNS 通知 CloudTrail
- 接收来自多个区域的 CloudTrail 日志文件和接收来自多个账户的 CloudTrail 日志文件

所有 HealthOmics 操作均由《API 参考》记录 CloudTrail 并记录在《AWS HealthOmics API 参考》中。例如，调用`StartVariantImportJob`和`CreateWorkflow`操作会在 CloudTrail 日志文件中生成条目。`CreateReferenceeStore`

每个事件或日志条目都包含有关生成请求的人员信息。身份信息有助于您确定以下内容：

- 请求是否使用 IAM 用户证书发出。
- 请求是使用角色还是联合用户的临时安全凭证发出的。
- 请求是否由其他 AWS 服务发出。

有关更多信息，请参阅 CloudTrail userIdentity 元素。

了解 HealthOmics 日志文件条目

跟踪是一种配置，允许将事件作为日志文件传输到您指定的 Amazon S3 存储桶。 CloudTrail 日志文件包含一个或多个日志条目。事件代表来自任何来源的单个请求，包括有关请求的操作、操作的日期和时间、请求参数等的信息。 CloudTrail 日志文件不是公共 API 调用的有序堆栈跟踪，因此它们不会按任何特定的顺序出现。

以下示例显示了演示该 CreateWorkflow 操作的 CloudTrail 日志条目。


```
{
    "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"
}
```


EventBridge 与一起使用 AWS HealthOmics

HealthOmics EventBridge 当资源状态发生变化时，向 Amazon 发送事件。资源包括导入任务、导出任务、资源共享、工作流程、任务和运行。对于每种类型的资源，都有一个生成事件的状态更改列表。

事件总线是接收事件并将其传送到目的地的路由器。您的账户包含一个默认事件总线，该总线可自动接收来自 AWS 服务的事件。您可以创建其他自定义事件总线。

您可以创建 EventBridge 规则来指定事件总线接收事件时要采取的操作。例如，您可以创建一条规则，通知您有关资源状态变化的信息。

使用事件的常见场景包括：

- 监控用户何时与您共享资源或撤消共享。
- 监视运行是失败还是成功完成。

有关使用的更多信息 EventBridge，请参阅 Amazon 是什么 EventBridge？

主题

- 设置 EventBridge 为 HealthOmics
- EventBridge 中的事件 HealthOmics

- 事件消息结构
- 事件消息示例

设置 EventBridge 为 HealthOmics

在监控 EventBridge 事件之前，请先创建 EventBridge 总线并为感兴趣的事件创建规则。

配置总 EventBridge 线

您可以为自己使用默认事件总线， AWS 账户 也可以配置自定义事件总线。要配置自定义事件总线，请执行以下步骤：

1. 打开 EventBridge 控制台:https://console.aws.amazon.com/events/.
2. 在左侧导航栏中，选择事件总线。
3. 选择 Create event bus (创建事件总线)。
4. 在创建事件总线窗体中，输入总线的名称。
5. 选择 “创建” 以创建总线。

创建 EventBridge 规则

以下过程说明如何创建简单规则。有关规则的更多信息，请参阅中的规则 EventBridge。

1. 打开 EventBridge 控制台:https://console.aws.amazon.com/events/.
2. 在左侧导航窗格中，选择 Rules。
3. 选择创建规则。控制台打开 “创建规则” 表单。
4. 在定义规则详细信息中，提供规则的名称。
   - 在名称中，输入总线的名称。
   - 对于事件总线，请为该规则选择总线。
   - 选择下一步。
5. 在构建事件模式中，在事件源下选择 AWS 事件或 EventBridge 合作伙伴事件。
6. 向下滚动到事件模式。
   a. 对于事件源，请选择 AWS 服务。
   b. 对于 AWS 服务，请在文本筛选器中输入 omics，然后选择AWS HealthOmics作为服务。
   c. 对于事件类型，选择感兴趣的事件（或所有事件）。

d. 选择下一步。

7. 在选择目标中，为事件选择一个目标。例如，选择 AWS 服务，选择CloudWatch 日志组，然后配置日志组。

   对于许多目标类型， EventBridge 需要权限将事件发送到目标。控制台会为您创建这些权限。

8. （可选）在配置标签中，将标签与规则关联。
9. 在查看和更新中，查看配置并选择创建规则。

EventBridge 中的事件 HealthOmics

下表列出了 HealthOmics 发送到 EventBridge的事件以及该事件的可能状态值列表。

事件名称	可能的状态值
注释导入 Job 状态更改	已提交、进行中、已取消、已完成、失败或已完成但失败
注释存储共享状态变更	待处理、激活、激活、删除、已删除、失败
注释存储状态变更	创建、创建、更新、更新、删除、删除或创建失败
已阅读 “设置激活 Job 状态更改”	已提交、进行中、已完成、失败或已完成但失败
已读取设置导出 Job 状态更改	已提交、进行中、已完成、失败或已完成但失败
读取设置导入 Job 状态更改	已提交、进行中、已完成、失败或已完成但失败
已阅读 “设置状态更改”	正在处理上传、上传失败、激活、已存档、激活或已删除
参考导入 Job 状态更改	已提交、进行中、已完成、失败或已完成但失败
参考状态变更	已激活或已删除
参考商店状态变更	已创建、更新、活动或已删除
运行状态更改	待处理、正在启动、正在运行、正在停止、已完成、已删除、失败或已取消

事件名称	可能的状态值
序列存储状态更改	已创建、更新、活动或已删除
任务状态更改	待处理、正在启动、正在运行、正在停止、已完成、已删除、失败或已取消
变体导入 Job 状态更改	已提交、进行中、已取消、已完成、失败或已完成但失败
变体商店共享状态变更	待处理、激活、激活、删除、已删除、失败
变体商店状态变更	创建、创建、更新、更新、删除、删除或创建失败
工作流程共享状态更改	待处理、激活、激活、删除、已删除、失败
工作流程状态变更	创建成功、创建失败、删除成功或删除失败

事件消息结构

HealthOmics 提供尽力交付以向其发送状态更改事件消息 EventBridge。该事件是一个具有 JSON 结构的对象，其中还包含元数据详细信息。您可以使用元数据作为输入来重新创建事件或了解更多信息。事件包括以下字段：

- `version`— 目前所有事件均为 0（零）。
- `id`— 为每个事件生成的版本 4 UUID。
- `detail-type`— 正在发送的事件类型。
- `account`— 存储桶所有者的 12 位 AWS 账户 ID。
- `source`— 标识生成事件的服务。
- `time`— 事件发生的时间。
- `region`— 标 AWS 区域 识存储桶的。
- `resources`— 包含存储桶的亚马逊资源名称 (ARN) 的 JSON 数组。
- `detail`— 包含事件相关信息的 JSON 对象。

运行事件包括以下字段：

- uuid— 运行的通用唯一标识符。
- workflowId— 与此运行关联的工作流程的工作流程标识符。
- workflowName— 与此运行关联的工作流程的名称。
- runId— 运行标识符。
- runName— 运行名称。
- runOutputUri— 运行将写入其输出数据的 URI。

事件消息示例

以下示例是运行状态更改的事件，显示了其他字段。

```
{
    "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"
    }
}
```

以下示例是任务状态更改的事件。

```
{
    "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"
    }
}
```

以下是读取集状态更改事件的示例。

```
{
  "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"
  }
}
```

将为变体商店导入任务创建类似的事件。

```
{
    "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"
    }
}
```

以下是导入任务状态变更的事件。

```
{
    "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"
    }
}
```

故障排除

以下主题可以帮助您解决在使用 HealthOmics 工作流程和数据存储时遇到的问题。

主题

排查工作流

主题

如何对失败的运行进行故障排除？

使用 GetRunAPI 操作检索失败原因。有关更多信息，请参阅 运行失败原因。

如何对失败的任务进行故障排除？

查看任务失败消息中的错误代码以了解失败原因。查看任务登录 CloudWatch 以查看该任务的详细日志消息。如果您没有收到详细的日志消息，则可以修改工作流程以输出其他日志语句。有关更多信息，请参阅 HealthOmics 使用 CloudWatch 日志进行监控。

在哪里可以找到成功完成运行的引擎日志？

HealthOmics 仅向发布失败运行 CloudWatch 的日志。如果运行成功完成，则会将引擎日志 HealthOmics 传送到您的 Amazon S3 存储桶。有关更多信息，请参阅 登录亚马逊 S3。

如何减小工作流程的输入参数大小？

您最多可以为工作流程指定 50 KB 的输入参数。您可以使用目录导入或样本表来保持在此大小限制范围内。有关更多信息，请参阅 管理运行参数大小。

为什么我的跑步没有完成？

如果您的代码存在问题并且进程未正确退出，则您的运行可能会变得无响应或 “卡住”。有关如何防止和 catch 无响应运行的更多信息，请参阅无响应跑步指南。

解决呼叫缓存问题

以下主题可以帮助您解决在呼叫缓存中遇到的问题。

主题

为什么我的跑步没有保存到缓存中？

1. 通过检查 GetRun API 操作响应中的 cacheId 字段，验证运行是否配置为使用缓存。使用 CLI 运行以下命令：`aws omics get-run –id <run_id>`。
2. 如果运行成功，请验证 GetRun 响应中返回的缓存行为是否为 CACHE_ALWAYS。如果将缓存行为设置为 CACHE_ON_FAILURE，则只有在运行失败时才会保存到缓存中。

为什么任务不使用缓存条目？

<cache_id><cache_uuid>在`/aws/omics/WorkflowLog` CloudWatch 日志组中，打开运行缓存的日志流：r unCache//。

1. 验证之前的运行是否为预期要缓存的任务创建了缓存条目。已保存到缓存中的运行将记录在 CACHE_ENTRY_CREATED 的日志消息中。
2. 找到任务的 CACHE_MISS 日志，然后运行已完成的任务。如果没有日志条目，请检查运行是否已配置为使用缓存。
3. 如果创建了缓存条目，请验证这两个任务的 CPUs、内存 GPUs 和容器摘要是否相同。创建缓存条目的任务的任务 ARN 在日志消息中。

4. 如果两个任务的计算要求匹配，请验证两个任务之间的输入是否没有变化。为此，请打开引擎日志。如果运行的状态为 “失败”，则日志将位于 Cloudwatch 日志组/ aws/omics/WorkflowLog 中。否则，可以在运行的输出目录中找到引擎日志。

对数据存储进行故障排除

主题

为什么 S3 在我的读取集上 GetObject 失败？

最常见的是，失败是由于缺少权限造成的。序列存储 S3 读取权限是一种双向配置，要求序列存储 S3 访问策略允许访问，并要求 IAM 委托人附加允许访问的策略。有关政策要求的更多详细信息，请参阅使用 Amazon S3 访问数据的权限 URIs。检查以下配置是否已准备就绪：

- 序列存储 S3 访问策略已明确允许访问 IAM 委托人或委托人账户的根目录。
- 检查 IAM 委托人是否有明确为正在访问的资源提供权限的策略。请注意，在定义权限时，IAM 委托人策略必须使用接入点 ARN，而不是基于接入点别名的路径，并且 ARN 处于条件中，不用于指定资源。
- 如果您的商店使用客户托管密钥 (CMK-KMS)，请确保 IAM 委托人对该密钥具有 kms:解密权限。有关配置跨账户使用情况的信息，请参阅 KMS 跨账户访问指南。

如果您的策略使用基于标签的访问控制，请确保以下几点：

- 确保序列存储已完成对标签的同步。为此，商店的状态必须是active，而不是updating。
- 确保读取集和策略上的标签键或键值中没有拼写错误。

为什么我在 Athena 中看不到我的注释库或变体存储库？

在 Lake Formation 中，请务必根据与你共享的商店创建资源链接。创建您有权访问的资源链接后，该商店应在 Athena 中可见。有关更多信息，请参阅 配置 Lake Formation 以供使用 HealthOmics。

为什么我无法访问我在 Athena 中的数据存储？

如果您的注释或变体存储可见，但您收到一条错误消息，提示访问被拒绝，请检查您使用的查询引擎版本。仅支持使用引擎版本 3 运行的查询。要了解有关 Athena 查询引擎版本的更多信息，请参阅亚马逊 A thena 文档。

的配额 AWS HealthOmics

AWS 使用 HealthOmics 配额的默认值填充您的账户。除非另有说明，否则每个配额值均为每个地区的最大值。

> **Important**
>
> 您可以申请增加大部分服务配额和 API 配额。有关详细信息，请参阅以下主题。

主题

HealthOmics 服务配额

下表列出了 HealthOmics 服务配额及其默认值。要查看每个区域的当前配额，请打开 S ervice Quotas 控制台。

> **Important**
>
> 您可以使用 Service Quotas 控制台申请增加可调整配额。

有关服务配额的更多信息，请参阅 Servic e Quotas 用户指南中的申请增加配额。如需在 Service Quotas 控制台中不可用的配额，请使用配额增加表单。

名称	默认值	可调整	描述
分析-最大注释存储量	每个受支持的区域：10 个	是	当前 AWS 区域中注释存储的最大数量

名称	默认值	可调整	描述
Analytics-最大并发变体或注释存储导入任务数	每个受支持的区域：5 个	是	当前 AWS 区域的最大并发导入任务数
Analytics-每个变体存储导入任务的最大文件数	每个受支持的区域：1,000 个	是	当前 AWS 区域中每个变体导入任务的最大文件数
分析-每个注释存储的最大份额	每个受支持的区域：10 个	是	当前 AWS 区域中每个注释存储区的最大共享数
分析-每个变体商店的最大份额	每个受支持的区域：10 个	是	当前 AWS 区域内每个变体商店的最大股票数量
Analytics-变体导入任务中每个文件的最大大小	每个受支持的区域：20 GB	是	当前 AWS 区域变体导入任务中一个文件的最大大小
Analytics-注释导入任务中每个文件的最大大小	每个受支持的区域：20 GB	是	当前 AWS 区域注释导入任务中一个文件的最大大小
分析-变体商店数量上限	每个受支持的区域：10 个	是	当前 AWS 区域中变体商店的最大数量
分析-每个注释存储库的最大版本数	每个受支持的区域：10 个	是	当前 AWS 区域中每个注释存储区的最大版本数
存储-最大并发读集激活作业数	每个受支持的区域：25 个	是	当前 AWS 区域中并发读集激活任务的最大数量
存储-最大并发序列和参考存储导出任务数	每个受支持的区域：5 个	是	当前 AWS 区域中序列或参考存储中并发导出任务的最大数量

名称	默认值	可调整	描述
存储-最大并发序列或参考存储导入任务数	每个受支持的区域：5 个	是	当前 AWS 区域中序列或参考存储的最大并发导入任务数
存储-每个序列存储的最大读取集数	每个受支持的区域：100 万个	是	当前 AWS 区域中序列存储的最大读取集数
存储-每个参考文献库的最大参考文献数量	每个受支持的区域：50 个	是	当前 AWS 区域参考文献库中的最大参考文献数量
存储-最大序列存储量	每个受支持的区域：20 个	是	当前 AWS 区域中序列存储的最大数量
工作流程-最大活动工作量 GPUs	每个受支持的区域：12 个	是	当前 AWS 区域 GPUs 中并发活动的最大数量。在 us-east-1 和 us-west-2 中，金额不超过 500 的配额增加请求会自动获得批准。
工作流程-使用动态运行存储空间的最大并发活动运行次数	每个受支持的区域：50 个	是	当前 AWS 区域中使用动态运行存储的最大活动运行次数。金额不超过 200 的配额增加请求将自动获得批准。
工作流程-使用静态运行存储空间的最大并发活动运行次数	每个受支持的区域：10 个	是	当前 AWS 区域中使用静态运行存储的最大活动运行次数。值不超过 50 的配额增加请求将自动获得批准。

名称	默认值	可调整	描述
工作流程-每次运行的最大并发任务数	每个受支持的区域：25 个	是	当前 AWS 区域中每次运行的最大并发任务数。在 us-east-1 和 us-west-2 中，金额不超过 100 的配额增加请求会自动获得批准。
工作流程-最长运行时长	每个支持的区域：604,800 秒	是	当前 AWS 区域的最大工作流程运行时长。
工作流程-最大运行次数（活动或非活动）	每个受支持的区域：5000 个	是	当前 AWS 区域的最大运行次数（活动或非活动）。
工作流程-每个工作流程的最大份额	每个受支持的区域：100 个	是	当前 AWS 区域中每个工作流程的最大共享数
工作流程-每次运行的最大静态运行存储容量	每个支持的区域：9,600	是	当前区域中每次运行的最大静态运行存储容量（以 GiB 为单位）。 AWS 在 us-east-1 和 us-west-2 中，金额不超过 50,000 的配额增加申请将自动获得批准。
工作流程-最大工作流程	每个受支持的区域：1,000 个	是	当前 AWS 区域的最大工作流数量。
工作流程-操作的每秒事务数 (TPS) StartRun	每个支持的区域：0.1	是	当前 AWS 区域中该 StartRun 操作的最大每秒事务数 (TPS)。值不超过 1 的配额增加请求将自动获得批准。

HealthOmics 固定大小配额

除了HealthOmics 服务配额，还 HealthOmics 包括具有固定大小的配额。您不能请求增加这些值。

除非另有说明，否则每个配额都列出了每个地区的最大值。

主题

HealthOmics 分析固定大小配额

下表显示了分析配额支持的最大值。这些值不可调整。

名称	描述	最大值	可调是/否
Analytics-每个注释存储导入任务的最大文件数	每个注释导入任务的最大文件数。	1	否

HealthOmics 存储固定大小配额

下表显示了存储文件支持的最大值。这些值不可调整。

名称	描述	最大值	可调是/否
存储-S3 访问资源策略的最大大小	S3 访问资源策略的最大大小	15 KB	否
存储-最大传播的集合级别标签	每个存储区传播到 S3 对象的最大设置级别标签密钥数量	5	否

名称	描述	最大值	可调是/否
存储-每个激活任务的最大读取集数	每个激活任务的最大读取集数。	20	否
存储-每个导出任务的最大读取集数	每个导出任务的最大读取集数。	100	否
存储-每个导入任务的最大读取集数	每个导入任务的最大读取集数。	100	否
存储-最大参考存储量	参考存储库的最大数量。	1	否
存储-直接上传的最大分段大小	直接上传到序列存储的最大分段大小。	100 MB	否
存储-文件中用于直接上传的最大段数	文件中可直接上传到序列存储的最大分段数。	10000	否
存储-最大参考大小	可以导入到参考存储库的参考文件的最大大小。	15 GB	否
存储-最大读取集源大小	读取集中可导入序列存储的单个源文件的最大大小。	976 GB	否

HealthOmics 工作流程固定大小配额

下表显示了工作流配额支持的最大值。这些值不可调整。

名称	描述	最大大小	可调是/否
工作流程-最大运行组数	运行组的最大数量。	1000	否

名称	描述	最大大小	可调是/否
工作流程-最大运行缓存	您可以为一个账户创建的最大运行缓存数。 一个或多个运行可以共享同一个运行缓存。每个账户 HealthOmics 可以缓存的运行次数没有配额。	1000	否
工作流程-最大工作流程版本	每个工作流程的最大工作流程版本数。	1000	否
工作流程-CPU 实例容器大小	CPU 实例的最大容器镜像大小。	45 GiB	否
工作流程-GPU 实例容器大小	GPU 实例的最大容器镜像大小。	95 GiB	否
GPU 实例 /dev/shm 共享内存	每个 GPU 实例的最大共享内存量。	每个 GPU 8 GB	否
工作流程-运行参数文件	运行参数文件的最大大小。	5 万字节	否
工作流程-工作流参数模板文件	工作流参数模板文件的最大条目数和最大文件大小。此配额适用于您使用控制台或 API 创建的工作流程。	1,000 个条目，400 KB	否
工作流程-工作流程定义文件大小-API	使用 API 操作或 AWS SDK 创建工作流程时工作流定义文件的最大大小。	100 MB	否

名称	描述	最大大小	可调是/否
工作流程-工作流程定义文件大小-控制台（直接上传）	使用控制台创建工作流程时，您可以直接上传的工作流程定义文件的最大大小。	4.4 MB	否
工作流程-工作流程定义文件大小-控制台（从 Amazon S3 上传）	使用控制台创建工作流程时，您可以作为从 Amazon S3 上传的工作流程定义文件的最大大小。	100 MB	否
工作流程-存储库大小	外部代码存储库的最大大小。	1 GiB	否
工作流程-存储库单个文件大小	来自外部代码存储库的单个文件的最大大小。	100 MiB	否
工作流程-自述文件大小	自述文件的最大大小。	500 KiB	否

有关如何减小运行参数文件大小的建议，请参阅管理运行参数大小。

HealthOmics Ready2Run 工作流程固定大小配额

每个 Ready2Run 工作流程都有最大输入文件大小。在下表中，文件大小单位以吉字节 (GiB) 为单位列出。这些最大文件大小不可调整。

Ready2Run 工作流程名称	最大输入文件大小 (GiB)	可调节（是/否）
AlphaFold 用于 601-1200 残留物	1	否
AlphaFold 最多可容纳 600 个残留物	1	否

Ready2Run 工作流程名称	最大输入文件大小 (GiB)	可调节（是/否）
适用于 2x150 的 Bases2Fastq	1000	否
适用于 2x300 的 Bases2Fastq	1000	否
适用于 2x75 的 Bases2Fastq	500	否
ESMFold 最多可容纳 800 个残留物	1	否
GATK-BP fq2bam	64	否
GATK-BP Germline bam2vcf 用于 30 倍基因组	39	否
GATK-BP Germline fq2vcf 用于 30 倍基因组	64	否
GATK-BP Somatic WES bam2vcf	86	否
NVIDIA Parabricks BAM2 FQ2 BAM WGS 最高可达 30 倍	80	否
NVIDIA Parabricks BAM2 FQ2 BAM WGS 最高可达 50 倍	120	否
NVIDIA Parabricks BAM2 FQ2 BAM WGS 最高可达 5 倍	20	否
NVIDIA Parabricks FQ2 BAM WGS 最高可达 30 倍	71	否
NVIDIA Parabricks FQ2 BAM WGS 最高可达 50 倍	137	否
NVIDIA Parabricks FQ2 BAM WGS 最高可达 5 倍	13	否

Ready2Run 工作流程名称	最大输入文件大小 (GiB)	可调节（是/否）
NVIDIA Parabricks Germline DeepVariant WGS 最高可达 30 倍	71	否
NVIDIA Parabricks Germline DeepVariant WGS 最高可达 50 倍	137	否
NVIDIA Parabricks Germline DeepVariant WGS 售价高达 5 倍	12	否
NVIDIA Parabricks Germline HaplotypeCaller WGS 最高可达 30 倍	71	否
NVIDIA Parabricks Germline HaplotypeCaller WGS 最高可达 50 倍	137	否
NVIDIA Parabricks Germline HaplotypeCaller WGS 售价高达 5 倍	13	否
NVIDIA Parabricks Somatic Mutect2 WGS 最高可达 50 倍	196	否
sc wit RNAseq h Kallisto BUStools	119	否
sc RNAseq 配三文鱼 Alevin-fry	119	否
sc w RNAseq ith STARsolo	119	否
Sentieon Germline BAM WES 最高可达 300 倍	9	否

Ready2Run 工作流程名称	最大输入文件大小 (GiB)	可调节（是/否）
Sentieon Germline BAM WGS 最高可达 32 倍	18	否
Sentieon Germline FASTQ WES 最高可达 100 倍	5	否
Sentieon Germline FASTQ WES 最高可达 300 倍	26	否
Sentieon Germline FASTQ WGS 最高可达 32 倍	51	否
安大略省的 Sentieon LongRead	25	否
Sentieon for LongRead PacBio HiFi	58	否
Sentieon Somatic WES	50	不可以
Sentieon Somatic WGS	113	否
Ultima Genomic DeepVariant s 最高可达 40 倍	91	否

HealthOmics API 配额

HealthOmics 具有以下与 API 操作相关的配额。如有说明，配额是可调整的。要申请增加配额，请使用增加配额表格。

对于列出的每个 API 操作，配额为每个区域中该 API 操作的最大每秒交易量 (TPS)。

主题

- 一般 API 配额
- 存储 API 配额
- 工作流程 API 配额

- 分析 API 配额

一般 API 配额

下表列出了适用于多个类别（存储、工作流和分析）的常规 API 操作。

API 操作	默认最大 TPS	可调节（是/否）
AcceptShare, CreateShare, DeleteShare, GetShare, ListShares	1 TPS	是

存储 API 配额

下表列出了存储 API 操作。

存储 API 操作	默认最大 TPS	可调节（是/否）
CreateSequenceStore, UpdateSequenceStore, DeleteSequenceStore, CreateReferenceStore, DeleteReferenceStore	1 TPS	是
BatchDeleteReadSet, DeleteReference	1 TPS	是
CreateMultipartReadSetUpload, CompleteMultipartReadSetUpload, AbortMultipartReadSetUpload	1 TPS	否
gets3 AccessPolicy、put AccessPolicy s3、DeleteS3 AccessPolicy	1 TPS	是
GetReference	10 TPS	是

存储 API 操作	默认最大 TPS	可调节（是/否）
UploadReadSetPart	10 TPS	是
GetReadSet	30 TPS	是
GetSequenceStore, ListSequenceStores	5 TPS	是
GetReadSetMetadata, ListReadSets	5 TPS	是
StartReadSetImportJob, GetReadSetImportJob, ListReadSetImportJobs	5 TPS	是
StartReadSetExportJob, GetReadSetExportJob, ListReadSetExportJobs	5 TPS	是
ListReferenceStores	5 TPS	是
StartReferencetImportJob, GetReferenceImportJob, ListReferenceImportJobs	5 TPS	是
ListReferences, GetReferenceMetadata	5 TPS	是
StartReadsetActivationJob	5 TPS	是
ListReadsetActivationJobs, GetReadSetActivationJob	5 TPS	是
ListMultipartReadSetUploads, ListReadSetUploadParts	5 TPS	是
TagResource, UntagResource, ListTagsForResource	5 TPS	是

工作流程 API 配额

下表列出了工作流程 API 操作。

工作流程 API 操作	默认最大 TPS	可调节（是/否）
StartRun	0.1 TPS	是
CreateWorkflow	5 TPS	是
CancelRun, DeleteRun , GetRun, GetRunTask, ListRunTasks, ListRuns	10 TPS	是
CreateRunGroup, DeleteRun Group, GetRunGroup, ListRunGroups, UpdateRun Group	10 TPS	是
CreateRunCache, UpdateRun Cache, DeleteRunCache, GetRunCache, ListRunCaches	10 TPS	是
DeleteWorkflow, GetWorkfl ow, ListWorkflows, UpdateWor kflow	10 TPS	是

分析 API 配额

下表列出了分析 API 的操作。

分析 API 操作	默认最大 TPS	可调节（是/否）
CreateVariantStore, DeleteVar iantStore, GetVariantStore, ListVariantStores, UpdateVar iantStore	1 TPS	否

分析 API 操作	默认最大 TPS	可调节（是/否）
StartVariantImportJob, CancelVariantImportJob, GetVariantImportJob, ListVariantImportJobs	1 TPS	否
CreateAnnotationStore, DeleteAnnotationStore, GetAnnotationStore, ListAnnotationStores, UpdateAnnotationStore	1 TPS	否
StartAnnotationImportJob, ListAnnotationImportJobs, GetAnnotationImportJob, CancelAnnotationImportJob	1 TPS	否

《 HealthOmics 用户指南》的文档历史记录

下表描述了的文档版本 HealthOmics。

变更	说明	日期
新的自述文件和存储库集成功能	增加了对从外部代码存储库和 README 文件创建工作流程的支持。	2025 年 7 月 24 日
新功能	HealthOmics 添加了对 Nextflow 自动参数插值的支持。要了解更多信息，请参阅 HealthOmics 工作流程的参数模板文件。	2025年6月27日
新功能	HealthOmics 添加了对工作流程版本控制的支持。要了解更多信息，请参阅中的 HealthOmics工作流程版本控制。	2025 年 4 月 18 日
新功能	HealthOmics 为动态运行存储增加了弹性吞吐量。要了解更多信息，请参阅中的运行存储类型 HealthOmics。	2025 年 4 月 16 日
新功能	HealthOmics 为序列存储 S3 位置添加了基于属性的访问控制，并且能够将多达五个读取集标签同步到 Sequence Store S3 对象。要了解更多信息，请参阅创建 HealthOmics 序列存储。	2024 年 11 月 22 日
新功能	HealthOmics 为私有工作流程添加了对呼叫缓存（也称为	2024 年 11 月 20 日

	恢复）的支持。要了解更多信息，请参阅呼叫缓存。	
新功能	HealthOmics 添加了新的 API 字段，以帮助您在序列存储输入作业和读取集之间进行映射。	2024 年 8 月 29 日
新功能	HealthOmics 增加了对管理 Nextflow 版本的支持。要了解更多信息，请参阅 Nextflow 版本。	2024 年 8 月 14 日
新功能	HealthOmics 增加了对共享工作流程和动态运行存储的支持。	2024 年 4 月 30 日
新功能	HealthOmics 增加了对 Amazon S3 访问参考和序列存储的支持，并支持 SHA256 ETags。	2024 年 4 月 15 日
新功能	HealthOmics 为序列存储添加了实体标签 (ETags)。	2023 年 10 月 6 日
新功能	HealthOmics 添加了注释存储版本控制和分析存储共享。	2023 年 8 月 15 日
新功能	HealthOmics 添加了通用工作流语言 (CWL) 作为 HealthOmics 工作流支持的语言。	2023 年 6 月 30 日
新功能	HealthOmics 添加了新的 Ready2Run 工作流程、对工作流程的 GPU 支持、注释存储的数据解析、直接上传到 HealthOmics 存储以及与的集成。 EventBridge	2023 年 5 月 15 日

新托管式策略	HealthOmics 添加了提供完全访问权限的新托管策略。要了解更多信息，请参阅 AWS 托管策略。	2023 年 2 月 23 日
新托管式策略	HealthOmics 添加了一个新的托管策略，该策略将访问权限限制为只读。要了解更多信息，请参阅 AWS 托管策略。	2022 年 11 月 29 日
初始版本	《 HealthOmics 用户指南》的初始版本	2022 年 11 月 29 日

本文属于机器翻译版本。若本译文内容与英语原文存在差异，则一律以英文原文为准。