使用Swift REST API (已弃用)

StorageGRID

NetApp
November 04, 2025

This PDF was generated from https://docs.netapp.com/zh-cn/storagegrid-117/swift/history-of-swift-api-support-in-storagegrid.html on November 04, 2025. Always check docs.netapp.com for the latest.

目录

使用Swift REST API (已弃用) 1
- 使用Swift REST API：概述 1
  - StorageGRID 中的 Swift API 支持历史记录 1
  - StorageGRID 如何实施 Swift REST API 2
  - 实施 Swift REST API 的建议 2
- 配置租户帐户和连接 3
  - 创建和配置 Swift 租户帐户 4
  - 如何配置客户端连接 4
  - 在 Swift API 配置中测试连接 6
- Swift REST API 支持的操作 7
  - StorageGRID 中支持的操作 7
  - 所有操作的通用响应标头 8
  - 支持的 Swift API 端点 8
  - 帐户操作 10
  - 容器操作 11
  - 对象操作 13
  - 选项请求 17
  - 对 Swift API 操作的错误响应 18
- StorageGRID Swift REST API 操作 19
  - 获取容器一致性请求 19
  - 提交容器一致性请求 20
- 配置 REST API 的安全性 22
  - StorageGRID 如何为REST API提供安全性 22
  - 支持 TLS 库的哈希和加密算法 23
- 监控和审核操作 24
  - 监控对象载入和检索速率 24
  - 访问和查看审核日志 26

使用Swift REST API (已弃用)

使用Swift REST API：概述

客户端应用程序可以使用 OpenStack Swift API 与 StorageGRID 系统相连接。

对Swift客户端应用程序的支持已弃用、将在未来版本中删除。

StorageGRID 支持以下特定版本的 Swift 和 HTTP 。

项目	**version**
Swift 规格	截至 2015 年 11 月的 OpenStack Swift 对象存储 API v1
HTTP	1.1 有关 HTTP 的详细信息，请参见 HTTP/1.1 （ RFC 7230-35 ）。 • 注 * ： StorageGRID 不支持 HTTP/1.1 管道化。

相关信息

"OpenStack ： 对象存储 API"

StorageGRID 中的 Swift API 支持历史记录

您应了解 StorageGRID 系统对 Swift REST API 的支持发生了哪些变化。

版本。	注释
11.7	对Swift客户端应用程序的支持已弃用、将在未来版本中删除。
11.6.	少量编辑更改。
11.5	消除了弱一致性控制。此时将改用可用的一致性级别。
11.4	增加了对TLS 1.3的支持。增加了 ILM 与一致性设置之间的关系问题描述 。
11.3	更新了 PUT 对象操作，以描述在载入时使用同步放置的 ILM 规则的影响（适用于载入行为的平衡而严格的选项）。添加了使用负载平衡器端点或高可用性组的客户端连接的问题描述 。不再支持 TLS 1.1 密码。
11.2.	对文档进行了少量编辑更改。

版本。	注释
11.1	增加了对使用 HTTP 与网格节点建立 Swift 客户端连接的支持。更新了一致性控制的定义。
11.0	增加了对每个租户帐户 1 ， 000 个容器的支持。
10.3	对文档进行管理更新和更正。删除了用于配置自定义服务器证书的部分。
10.2	StorageGRID 系统对 Swift API 的初始支持。当前支持的版本为 OpenStack Swift 对象存储 API v1 。

StorageGRID 如何实施 Swift REST API

客户端应用程序可以使用 Swift REST API 调用连接到存储节点和网关节点以创建容器以及存储和检索对象。这样，为 OpenStack Swift 开发的面向服务的应用程序便可与 StorageGRID 系统提供的内部对象存储相连接。

Swift 对象管理

将 Swift 对象载入 StorageGRID 系统后，这些对象将通过系统活动 ILM 策略中的信息生命周期管理（ ILM ）规则进行管理。。 "ILM 规则" 和 "ILM 策略" 确定StorageGRID 如何创建和分发对象数据的副本、以及如何在一段时间内管理这些副本。例如， ILM 规则可能适用于特定 Swift 容器中的对象，并可能指定在一定年数内将多个对象副本保存到多个数据中心。

如果您需要了解网格的ILM规则和策略如何影响Swift租户帐户中的对象、请联系您的NetApp专业服务顾问或StorageGRID 管理员。

客户端请求冲突

冲突的客户端请求（例如，两个客户端写入同一密钥）将以 " 最新成功 " 为基础进行解决。" 最新赢单 " 评估的时间取决于 StorageGRID 系统何时完成给定请求，而不是 Swift 客户端何时开始操作。

一致性保证和控制

默认情况下， StorageGRID 为新创建的对象提供读写一致性，并为对象更新和机头操作提供最终一致性。任意 "获取" 成功完成后执行 "PUT" 将能够读取新写入的数据。对现有对象的覆盖，元数据更新和删除最终保持一致。覆盖通常需要几秒钟或几分钟才能传播，但可能需要长达 15 天的时间。

此外，您还可以通过 StorageGRID 控制每个容器的一致性。一致性控制可以根据应用程序的要求、在对象的可用性与这些对象在不同存储节点和站点之间的一致性之间实现平衡。

实施 Swift REST API 的建议

在实施用于 StorageGRID 的 Swift REST API 时，应遵循以下建议。

针对不存在的对象的建议

如果您的应用程序定期检查某个对象是否位于您不希望该对象实际存在的路径上、则应使用"`available`"一致性控制。例如，如果应用程序在对某个位置执行 PUT 操作之前对该位置执行 HEAD 操作，则应使用 "`Available` " 一致性控制。

否则，如果 head 操作未找到对象，则在一个或多个存储节点不可用时，可能会收到大量 500 个内部服务器错误。

您可以使用为每个容器设置"`Available`"一致性控制 "提交容器一致性请求"。您可以查看使用为每个容器设置"`Available`"一致性控制 "获取容器一致性请求"。

对象名称建议

对于在 StorageGRID 11.4 或更高版本中创建的容器，不再需要限制对象名称以满足性能最佳实践。例如，现在可以对对象名称的前四个字符使用随机值。

对于在 StorageGRID 11.4 之前的版本中创建的容器，请继续对对象名称遵循以下建议：

- 不应使用随机值作为对象名称的前四个字符。这与 AWS 以前针对名称前缀的建议不同。而应使用非随机、非唯一前缀、例如 `image`。
- 如果您按照以前的 AWS 建议在名称前缀中使用随机和唯一字符，则应在对象名称前添加目录名称。也就是说，请使用以下格式：

  ```
  mycontainer/mydir/f8e3-image3132.jpg
  ```

  而不是以下格式：

  ```
  mycontainer/f8e3-image3132.jpg
  ```

关于"`范围读取`"的建议

如果 "用于压缩存储对象的全局选项" 已启用、则Swift客户端应用程序应避免执行指定要返回的字节数范围的GET对象操作。这些 "`range read` " 操作效率低下，因为 StorageGRID 必须有效解压缩对象以访问请求的字节。从非常大的对象请求少量字节的 GET 对象操作效率尤其低下；例如，从 50 GB 压缩对象读取 10 MB 范围的操作效率非常低。

如果从压缩对象读取范围，则客户端请求可能会超时。

> 如果需要压缩对象，并且客户端应用程序必须使用范围读取，请增加应用程序的读取超时时间。

配置租户帐户和连接

要将 StorageGRID 配置为接受来自客户端应用程序的连接，需要创建一个或多个租户帐户并设置连接。

创建和配置 **Swift** 租户帐户

要使 Swift API 客户端能够在 StorageGRID 上存储和检索对象，需要使用 Swift 租户帐户。每个租户帐户都有自己的帐户 ID ，组和用户以及容器和对象。

Swift 租户帐户由 StorageGRID 网格管理员使用网格管理器或网格管理 API 创建。

时间 "创建Swift租户帐户"，网格管理员指定以下信息：

- "租户的显示名称" (租户的帐户ID会自动分配、无法更改)
- (可选) "租户帐户的存储配额" --可供租户对象使用的最大GB、TB或PB数。租户的存储配额表示逻辑容量（对象大小），而不是物理容量（磁盘大小）。
- 条件 "单点登录(SSO)" 未用于StorageGRID 系统、租户帐户是使用自己的身份源还是共享网格的身份源、以及租户本地root用户的初始密码。
- 如果启用了SSO、则哪个联盟组具有配置租户帐户的root访问权限。

创建Swift租户帐户后、具有root访问权限的用户可以访问租户管理器来执行如下任务：

- 设置身份联合（除非身份源与网格共享），并创建本地组和用户
- 监控存储使用情况

> Swift用户必须具有的root访问权限 "访问租户管理器"。但是， " 根 " 访问权限不允许用户通过 Swift REST API 的身份验证来创建容器和载入对象。用户必须具有 Swift 管理员权限才能向 Swift REST API 进行身份验证。

如何配置客户端连接

网格管理员可以选择影响 Swift 客户端连接到 StorageGRID 以存储和检索数据的配置。建立连接所需的具体信息取决于所选的配置。

客户端应用程序可以通过连接到管理节点或网关节点上的负载平衡器服务来存储或检索对象、也可以连接到管理节点或网关节点的高可用性(HA)组的虚拟IP地址。

> 所有依赖StorageGRID 提供负载平衡的应用程序都应使用负载平衡器服务进行连接。

- 存储节点，具有或不具有外部负载平衡器

配置 StorageGRID 时，网格管理员可以使用网格管理器或网格管理 API 执行以下步骤，所有这些步骤均为可选步骤：

1. 为负载平衡器服务配置端点。

   您必须配置端点才能使用负载平衡器服务。管理节点或网关节点上的负载平衡器服务会将传入的网络连接从客户端应用程序分发到存储节点。创建负载平衡器端点时， StorageGRID 管理员会指定端口号，端点是否接受 HTTP 或 HTTPS 连接，将使用此端点的客户端类型（ S3 或 Swift ）以及用于 HTTPS 连接的证书（如果适用）。Swift支持这些功能 "端点类型"。

2. 配置不可信客户端网络。

如果 StorageGRID 管理员将节点的客户端网络配置为不可信，则节点仅接受客户端网络上显式配置为负载平衡器端点的端口上的入站连接。

3. 配置高可用性组。

   如果管理员创建了一个 HA 组，则多个管理节点或网关节点的网络接口将置于主动备份配置中。客户端连接使用 HA 组的虚拟 IP 地址进行。

请参见 "HA 组的配置选项" 有关详细信息 …

摘要：客户端连接的 **IP** 地址和端口

客户端应用程序使用网格节点的 IP 地址以及该节点上服务的端口号连接到 StorageGRID 。如果配置了高可用性（ HA ）组，则客户端应用程序可以使用 HA 组的虚拟 IP 地址进行连接。

建立客户端连接所需的信息

下表总结了客户端连接到 StorageGRID 的不同方式以及每种连接类型所使用的 IP 地址和端口。请参见 "客户端连接的IP地址和端口" 或联系您的StorageGRID 管理员以了解详细信息。

建立连接的位置	客户端连接到的服务	**IP** 地址	**Port**
HA 组	负载平衡器	HA 组的虚拟 IP 地址	• 负载平衡器端点端口
管理节点	负载平衡器	管理节点的 IP 地址	• 负载平衡器端点端口
网关节点	负载平衡器	网关节点的 IP 地址	• 负载平衡器端点端口
存储节点	LDR	存储节点的 IP 地址	默认 Swift 端口： • HTTPS： 18083 • HTTP： 18085

示例

要将 Swift 客户端连接到网关节点 HA 组的负载平衡器端点，请使用以下结构化 URL ：

- `https://VIP-of-HA-group:LB-endpoint-port`

例如，如果 HA 组的虚拟 IP 地址为 192.0.2.6 ， Swift 负载平衡器端点的端口号为 10444 ，则 Swift 客户端可以使用以下 URL 连接到 StorageGRID ：

- `https://192.0.2.6:10444`

可以为客户端用于连接到 StorageGRID 的 IP 地址配置 DNS 名称。请与本地网络管理员联系。

决定使用 **HTTPS** 或 **HTTP** 连接

使用负载平衡器端点建立客户端连接时，必须使用为此端点指定的协议（ HTTP 或 HTTPS ）进行连接。要使用HTTP进行客户端到存储节点的连接、必须启用HTTP。

默认情况下、当客户端应用程序连接到存储节点时、它们必须对所有连接使用加密的HTTPS。或者、您也可以通过选择来启用不太安全的HTTP连接 "为存储节点连接启用HTTP" 选项。例如，在非生产环境中测试与存储节点的连接时，客户端应用程序可能会使用 HTTP 。

> 为生产网格启用HTTP时请小心、因为请求和响应将以未加密的方式发送。

如果选择了*为存储节点连接启用HTTP *选项，则客户端对HTTP使用的端口必须与对HTTPS使用的端口不同。

在 Swift API 配置中测试连接

您可以使用 Swift 命令行界面测试与 StorageGRID 系统的连接，并验证是否可以向系统读取和写入对象。

开始之前

- 您必须已下载并安装 python swiftclient ，即 Swift 命令行客户端。

  "SwiftStack： Python-swiftclient"

- 您必须在 StorageGRID 系统中具有 Swift 租户帐户。

关于此任务

如果尚未配置安全性、则必须添加 `--insecure` 标记这些命令中的每个命令。

步骤

1. 查询 StorageGRID Swift 部署的信息 URL：

```
swift
-U <Tenant_Account_ID:Account_User_Name>
-K <User_Password>
-A https://<FQDN | IP>:<Port>/info
capabilities
```

   这足以测试您的 Swift 部署是否正常运行。要通过存储对象进一步测试帐户配置，请继续执行其他步骤。

2. 将对象放入容器中：

```
touch test_object
swift
-U <Tenant_Account_ID:Account_User_Name>
-K <User_Password>
-A https://<FQDN | IP>:<Port>/auth/v1.0
upload test_container test_object
--object-name test_object
```

3. 获取用于验证对象的容器：

```
swift
-U <Tenant_Account_ID:Account_User_Name>
-K <User_Password>
-A https://<FQDN | IP>:<Port>/auth/v1.0
list test_container
```

4. 删除对象：

```
swift
-U <Tenant_Account_ID:Account_User_Name>
-K <User_Password>
-A https://<FQDN | IP>:<Port>/auth/v1.0
delete test_container test_object
```

5. 删除容器：

```
swift
-U `<_Tenant_Account_ID:Account_User_Name_>`
-K `<_User_Password_>`
-A `\https://<_FQDN_ | _IP_>:<_Port_>/auth/v1.0'
delete test_container
```

相关信息

"创建和配置 Swift 租户帐户"

"配置 REST API 的安全性"

Swift REST API 支持的操作

StorageGRID 系统在 OpenStack Swift API 中支持大多数操作。在将 Swift REST API 客户端与 StorageGRID 集成之前，请查看帐户，容器和对象操作的实施详细信息。

StorageGRID 中支持的操作

支持以下 Swift API 操作：

- "帐户操作"
- "容器操作"
- "对象操作"

所有操作的通用响应标头

StorageGRID 系统可为 OpenStack Swift 对象存储 API v1 定义的受支持操作实施所有通用标头。

相关信息

"OpenStack： 对象存储 API"

支持的 **Swift API** 端点

StorageGRID 支持以下 Swift API 端点：信息 URL ，身份验证 URL 和存储 URL 。

信息 **URL**

您可以通过向带有 /info 路径的 Swift 基础 URL 发出 GET 请求来确定 StorageGRID Swift 实施的功能和限制。

`https://FQDN | Node IP:Swift Port/info/`

在请求中：

- *FQDN* 是完全限定域名。
- *Node IP* 是StorageGRID 网络上存储节点或网关节点的IP地址。
- *Swift Port* 是用于存储节点或网关节点上的Swift API连接的端口号。

例如，以下信息 URL 将从 IP 地址为 10.99.106.103 且使用端口 18083 的存储节点请求信息。

`https://10.99.106.103:18083/info/`

此响应包括 Swift 实施的功能，可用作 JSON 词典。客户端工具可以解析 JSON 响应以确定实施的功能，并将其用作后续存储操作的约束。

通过实施 StorageGRID ， Swift 可以对信息 URL 进行未经身份验证的访问。

身份验证 **URL**

客户端可以使用 Swift 身份验证 URL 作为租户帐户用户进行身份验证。

`https://FQDN | Node IP:Swift Port/auth/v1.0/`

您必须在中提供租户帐户ID、用户名和密码作为参数 `X-Auth-User` 和 `X-Auth-Key` 请求标题、如下所示：

`X-Auth-User: Tenant_Account_ID:Username`

`X-Auth-Key: Password`

在请求标题中：

- *Tenant_Account_ID* 是StorageGRID 在创建Swift租户时分配的帐户ID。这与租户管理器登录页面上使用的租户帐户 ID 相同。
- *Username* 是已在租户管理器中创建的租户用户的名称。此用户必须属于具有 Swift 管理员权限的组。无法将租户的root用户配置为使用Swift REST API。

如果为租户帐户启用了身份联合，请从 LDAP 服务器提供联合用户的用户名和密码。或者，也可以提供 LDAP 用户的域名。例如：

`X-Auth-User: Tenant_Account_ID:Username@Domain_Name`

- *Password* 是租户用户的密码。用户密码在租户管理器中创建和管理。

对成功的身份验证请求的响应将返回存储 URL 和身份验证令牌，如下所示：

```
X-Storage-Url: https://FQDN | Node_IP:Swift_Port/v1/Tenant_Account_ID

X-Auth-Token: token

X-Storage-Token: token
```

默认情况下，令牌在生成后 24 小时内有效。

为特定租户帐户生成令牌。一个帐户的有效令牌不会授权用户访问另一个帐户。

存储 **URL**

客户端应用程序可以通过问题描述 Swift REST API 调用对网关节点或存储节点执行支持的帐户，容器和对象操作。存储请求会发送到身份验证响应中返回的存储 URL 。此请求还必须包含从身份验证请求返回的 X-Auth-Token 标头和值。

```
https://FQDN | IP:Swift_Port/v1/Tenant_Account_ID

[/container][/object]

X-Auth-Token: token
```

某些包含使用情况统计信息的存储响应标头可能无法反映最近修改的对象的准确数字。准确的数字可能需要几分钟才能显示在这些标题中。

帐户和容器操作的以下响应标头是包含使用情况统计信息的示例：

- `X-Account-Bytes-Used`
- `X-Account-Object-Count`
- `X-Container-Bytes-Used`
- `X-Container-Object-Count`

相关信息

"配置租户帐户和连接"

"帐户操作"

"容器操作"

"对象操作"

帐户操作

对帐户执行以下 Swift API 操作。

获取帐户

此操作将检索与帐户和帐户使用情况统计信息关联的容器列表。

以下请求参数为必填项：

- `Account`

以下请求标头为必填项：

- `X-Auth-Token`

以下支持的请求查询参数是可选的：

- `Delimiter`
- `End_marker`
- `Format`
- `Limit`
- `Marker`
- `Prefix`

如果成功执行，则会返回以下标头，并在该帐户未找到容器或容器列表为空时返回 "`HTTP/1.1 204 No Content` " 响应；如果找到该帐户且容器列表不为空，则返回 "`HTTP/1.1 200 OK` " 响应：

- `Accept-Ranges`
- `Content-Length`
- `Content-Type`
- `Date`
- `X-Account-Bytes-Used`
- `X-Account-Container-Count`
- `X-Account-Object-Count`
- `X-Timestamp`
- `X-Trans-Id`

主帐户

此操作将从 Swift 帐户检索帐户信息和统计信息。

以下请求参数为必填项：

- `Account`

以下请求标头为必填项：

- `X-Auth-Token`

成功执行将返回以下标头，并显示 "`HTTP/1.1 204 No Content` " 响应：

- `Accept-Ranges`
- `Content-Length`
- `Date`
- `X-Account-Bytes-Used`
- `X-Account-Container-Count`
- `X-Account-Object-Count`
- `X-Timestamp`
- `X-Trans-Id`

相关信息

"监控和审核操作"

容器操作

每个 Swift 帐户最多支持 1 ， 000 个容器。 StorageGRID对容器执行以下 Swift API 操作。

删除容器

此操作会从 StorageGRID 系统中的 Swift 帐户中删除一个空容器。

需要以下请求参数：

- `Account`
- `Container`

以下请求标头为必填项：

- `X-Auth-Token`

成功执行将返回以下标头并显示 "HTTP/1.1 204 No Content" 响应：

- `Content-Length`
- `Content-Type`
- `Date`
- `X-Trans-Id`

获取容器

此操作将检索与此容器关联的对象列表以及 StorageGRID 系统中的容器统计信息和元数据。

需要以下请求参数：

- `Account`
- `Container`

以下请求标头为必填项：

- `X-Auth-Token`

以下支持的请求查询参数是可选的：

- `Delimiter`
- `End_marker`
- `Format`
- `Limit`
- `Marker`
- `Path`
- `Prefix`

成功执行将返回以下标头，并显示 "HTTP/1.1 200 successful" 或 "HTTP/1.1 204 No Content" 响应：

- `Accept-Ranges`
- `Content-Length`
- `Content-Type`
- `Date`
- `X-Container-Bytes-Used`
- `X-Container-Object-Count`
- `X-Timestamp`
- `X-Trans-Id`

机头容器

此操作将从 StorageGRID 系统检索容器统计信息和元数据。

需要以下请求参数：

- `Account`
- `Container`

以下请求标头为必填项：

- `X-Auth-Token`

成功执行将返回以下标头并显示 "HTTP/1.1 204 No Content" 响应：

- `Accept-Ranges`
- `Content-Length`
- `Date`
- `X-Container-Bytes-Used`
- `X-Container-Object-Count`
- `X-Timestamp`
- `X-Trans-Id`

放入容器

此操作将为 StorageGRID 系统中的帐户创建一个容器。

需要以下请求参数：

- `Account`
- `Container`

以下请求标头为必填项：

- `X-Auth-Token`

成功执行将返回以下标头，并显示 "HTTP/1.1 201 Created " 或 "HTTP/1.1 202 Accept" （如果此帐户下已存在此容器）响应：

- `Content-Length`
- `Date`
- `X-Timestamp`
- `X-Trans-Id`

容器名称在 StorageGRID 命名空间中必须是唯一的。如果容器位于其他帐户下，则返回以下标头： "HTTP/1.1 409 conflict 。 "

相关信息

"监控和审核操作"

对象操作

对对象执行以下 Swift API 操作。可以在中跟踪这些操作 "StorageGRID 审核日志"。

删除对象

此操作将从 StorageGRID 系统中删除对象的内容和元数据。

需要以下请求参数：

- `Account`
- `Container`
- `Object`

以下请求标头为必填项：

- `X-Auth-Token`

成功执行将返回以下响应标头和 `HTTP/1.1 204 No Content` 响应：

- `Content-Length`
- `Content-Type`
- `Date`
- `X-Trans-Id`

在处理删除对象请求时， StorageGRID 会尝试立即从所有存储位置删除此对象的所有副本。如果成功， StorageGRID 会立即向客户端返回响应。如果无法在30秒内删除所有副本(例如、由于某个位置暂时不可用)、则StorageGRID 会将这些副本排队等待删除、然后向客户端指示删除成功。

有关详细信息，请参见 "如何删除对象"。

获取对象

此操作将检索对象内容并从 StorageGRID 系统获取对象元数据。

需要以下请求参数：

- `Account`
- `Container`
- `Object`

以下请求标头为必填项：

- `X-Auth-Token`

以下请求标头是可选的：

- `Accept-Encoding`
- `If-Match`
- `If-Modified-Since`
- `If-None-Match`

- If-Unmodified-Since
- Range

成功执行将返回以下标头和 HTTP/1.1 200 OK 响应：

- Accept-Ranges
- Content-Disposition、只有在出现此情况时才返回 Content-Disposition 已设置元数据
- Content-Encoding、只有在出现此情况时才返回 Content-Encoding 已设置元数据
- Content-Length
- Content-Type
- Date
- ETag
- Last-Modified
- X-Timestamp
- X-Trans-Id

head 对象

此操作将从 StorageGRID 系统检索所载入对象的元数据和属性。

需要以下请求参数：

- Account
- Container
- Object

以下请求标头为必填项：

- X-Auth-Token

成功执行将返回以下标头并显示 "HTTP/1.1 200 OK" 响应：

- Accept-Ranges
- Content-Disposition、只有在出现此情况时才返回 Content-Disposition 已设置元数据
- Content-Encoding、只有在出现此情况时才返回 Content-Encoding 已设置元数据
- Content-Length
- Content-Type
- Date
- ETag
- Last-Modified
- X-Timestamp

- `X-Trans-Id`

PUT 对象

此操作将使用数据和元数据创建新对象，或者将现有对象替换为 StorageGRID 系统中的数据和元数据。

StorageGRID 支持大小高达 5 TiB （ 5 ， 497 ， 556 ， 138 ， 880 字节）的对象。

> 冲突的客户端请求（例如，两个客户端写入同一密钥）将以 " 最新成功 " 为基础进行解决。" 最新赢单 " 评估的时间取决于 StorageGRID 系统何时完成给定请求，而不是 Swift 客户端何时开始操作。

需要以下请求参数：

- `Account`
- `Container`
- `Object`

以下请求标头为必填项：

- `X-Auth-Token`

以下请求标头是可选的：

- `Content-Disposition`
- `Content-Encoding`

  请勿使用chunked `Content-Encoding` 如果ILM规则要求对对象进行适用场景 筛选、并在载入时使用同步放置(用于载入行为的平衡或严格选项)。

- `Transfer-Encoding`

  请勿使用压缩或分块 `Transfer-Encoding` 如果ILM规则要求对对象进行适用场景 筛选、并在载入时使用同步放置(用于载入行为的平衡或严格选项)。

- `Content-Length`

  如果ILM规则按大小筛选对象并在载入时使用同步放置、则必须指定 `Content-Length`。

  > 如果您未遵循这些准则、请执行以下操作 `Content-Encoding`， `Transfer-Encoding`，和 `Content-Length`、StorageGRID 必须先保存该对象、然后才能确定对象大小并应用ILM规则。换言之， StorageGRID 必须默认为在载入时创建对象的临时副本。也就是说，StorageGRID 必须对载入行为使用双提交选项。

  有关同步放置和ILM规则的详细信息、请参见 "用于载入的数据保护选项"。

- `Content-Type`
- `ETag`

- `X-Object-Meta-<name\>` (与对象相关的元数据)

  如果要使用*用户定义的创建时间*选项作为ILM规则的参考时间，则必须将该值存储在名为的用户定义标头中 `X-Object-Meta-Creation-Time`。例如：

  ```
  X-Object-Meta-Creation-Time: 1443399726
  ```

  自 1970 年 1 月 1 日以来，此字段的评估值为秒。

- `X-Storage-Class: reduced_redundancy`

  如果与所载入对象匹配的 ILM 规则指定了双重提交或平衡的载入行为，则此标头会影响 StorageGRID 创建的对象副本数。

  - * 双提交 * ： 如果 ILM 规则为载入行为指定了双提交选项，则 StorageGRID 会在载入对象时创建一个临时副本（单个提交）。
  - 均衡：如果ILM规则指定了均衡选项，则只有当系统无法立即创建规则中指定的所有副本时，StorageGRID 才会创建一个临时副本。如果 StorageGRID 可以执行同步放置，则此标头不起作用。

    。 `reduced_redundancy` 如果与对象匹配的ILM规则创建一个复制副本、则最好使用标题。在这种情况下、使用 `reduced_redundancy` 无需在每次载入操作中创建和删除额外的对象副本。

    使用 `reduced_redundancy` 在其他情况下不建议使用标头、因为它会增加载入期间丢失对象数据的风险。例如，如果最初将单个副本存储在发生故障的存储节点上，而此存储节点未能进行 ILM 评估，则可能会丢失数据。

    > 在任何一段时间内只复制一个副本会使数据面临永久丢失的风险。如果某个对象只存在一个复制副本，则在存储节点出现故障或出现严重错误时，该对象将丢失。在升级等维护过程中，您还会暂时失去对对象的访问权限。

  请注意、请指定 `reduced_redundancy` 仅影响首次载入对象时创建的副本数。它不会影响在活动 ILM 策略评估对象时创建的对象副本数，也不会导致数据在 StorageGRID 系统中以较低的冗余级别存储。

成功执行将返回以下标头，并显示 "HTTP/1.1 201 Created " 响应：

- `Content-Length`
- `Content-Type`
- `Date`
- `ETag`
- `Last-Modified`
- `X-Trans-Id`

选项请求

选项请求会检查单个 Swift 服务的可用性。选项请求由 URL 中指定的存储节点或网关节点处理。

options 方法

例如，客户端应用程序可以在不提供 Swift 身份验证凭据的情况下将选项请求问题描述 到存储节点上的 Swift 端口，以确定存储节点是否可用。您可以使用此请求进行监控，也可以允许外部负载平衡器确定存储节点何时关闭。

与信息 URL 或存储 URL 结合使用时， options 方法将返回给定 URL 支持的动词列表（例如 head ， get ， options 和 put ）。选项方法不能与身份验证URL结合使用。

以下请求参数为必填项：

- `Account`

以下请求参数是可选的：

- `Container`
- `Object`

成功执行将返回以下标头，并显示 "`HTTP/1.1 204 No Content` " 响应。对存储 URL 的选项请求不要求目标存在。

- `Allow` (给定URL支持的动词列表、例如head、get、options、和PUT)
- `Content-Length`
- `Content-Type`
- `Date`
- `X-Trans-Id`

相关信息

"支持的 Swift API 端点"

对 **Swift API** 操作的错误响应

了解可能的错误响应有助于您对操作进行故障排除。

如果操作期间发生错误，可能会返回以下 HTTP 状态代码：

Swift 错误名称	**HTTP** 状态
AccountNameTooLong ， ContainerNameTooLong ， HeaderTooBig ， InContainvaliderName ， InvalidRequest ， InvalidURI ， MetadataNameTooLong ， MetadataValueTooBig ， MissingSecurityHeader ， ObjectNameTooLong ， TooManyContainers ， TooManyMetadataItems ， TotalMetadataTooLarge	400 个错误请求
ACCESSDENIED	403 已禁用

Swift 错误名称	HTTP 状态
ContainerNotEmpty ， ContainerAlreadyExists	409 冲突
内部错误	500 内部服务器错误
InvalidRange	416 无法满足请求的范围
方法未使用	不允许使用 405 方法
MissingContent长度	411 需要长度
未找到	未找到 404
未实施	501 未实施
预条件已启用	412- 前提条件失败
ResourceNotFound	未找到 404
未授权	401 未授权
UnprocessableEntity	422 不可处理实体

StorageGRID Swift REST API 操作

在 Swift REST API 中添加了特定于 StorageGRID 系统的操作。

获取容器一致性请求

"一致性控制" 在不同存储节点和站点之间的对象可用性与这些对象的一致性之间取得平衡。使用获取容器一致性请求，您可以确定应用于特定容器的一致性级别。

请求

请求 HTTP 标头	Description
X-Auth-Token	指定要用于请求的帐户的 Swift 身份验证令牌。
X-nta-sg-sg-sg-一致性	指定请求的类型、其中 `true` =获取容器一致性、和 `false` =获取容器。
主机	请求所定向到的主机名。

请求示例

```
GET /v1/28544923908243208806/Swift container
X-Auth-Token: SGRD_3a877009a2d24cb1801587bfa9050f29
x-ntap-sg-consistency: true
Host: test.com
```

响应

响应 **HTTP** 标头	**Description**
Date	响应的日期和时间。
连接	与服务器的连接是打开还是关闭。
X-Trans-ID	请求的唯一事务标识符。
内容长度	响应正文的长度。
X-nta-sg-sg-sg-一致性	应用于容器的一致性控制级别。支持以下值： • 全部 * ： 所有节点均立即接收数据，否则请求将失败。 • 强 - 全局 * ： 保证所有站点中所有客户端请求的写入后读一致性。 • 强站点 * ： 保证站点内所有客户端请求的写入后读一致性。 读后新写入：(默认)为新对象提供读后写入一致性、并最终为对象更新提供一致性。提供高可用性和数据保护保证。建议用于大多数情况。 可用：为新对象和对象更新提供最终一致性。对于S3存储分段、请仅在需要时使用(例如、对于包含很少读取的日志值的存储分段、或者对于不存在的密钥执行HEAD或GET操作)。S3 FabricPool 存储分段不支持。

响应示例

```
HTTP/1.1 204 No Content
Date: Sat, 29 Nov 2015 01:02:18 GMT
Connection: CLOSE
X-Trans-Id: 1936575373
Content-Length: 0
x-ntap-sg-consistency: strong-site
```

提交容器一致性请求

使用 PUT 容器一致性请求可以指定要应用于对容器执行的操作的一致性级别。默认情况下，使用 `read-`

`after-new-write` 一致性级别创建新容器。

请求

请求 **HTTP** 标头	**Description**
X-Auth-Token	要用于请求的帐户的 Swift 身份验证令牌。
X-nta-sg-sg-sg-一致性	要应用于容器上的操作的一致性控制级别。支持以下值： • 全部 * ： 所有节点均立即接收数据，否则请求将失败。 • 强 - 全局 * ： 保证所有站点中所有客户端请求的写入后读一致性。 • 强站点 * ： 保证站点内所有客户端请求的写入后读一致性。 读后新写入：(默认)为新对象提供读后写入一致性、并最终为对象更新提供一致性。提供高可用性和数据保护保证。建议用于大多数情况。 可用：为新对象和对象更新提供最终一致性。对于S3存储分段、请仅在需要时使用(例如、对于包含很少读取的日志值的存储分段、或者对于不存在的密钥执行HEAD或GET操作)。S3 FabricPool 存储分段不支持。
`Host`	请求所定向到的主机名。

一致性控制和 **ILM** 规则如何交互以影响数据保护

您可以自由选择 "一致性控制" ILM规则会影响对象的保护方式。这些设置可以进行交互。

例如、存储对象时使用的一致性控制会影响对象元数据的初始放置、而会影响 "加热行为" 为ILM规则选择此选项会影响对象副本的初始放置。由于 StorageGRID 需要访问对象的元数据及其数据来满足客户端请求，因此为一致性级别和载入行为选择匹配的保护级别可以提供更好的初始数据保护和更可预测的系统响应。

一致性控制和 **ILM** 规则如何交互的示例

假设您有一个双站点网格，其中包含以下 ILM 规则和以下一致性级别设置：

- * ILM 规则 * ： 创建两个对象副本，一个在本地站点，一个在远程站点。此时将选择严格的载入行为。
- * 一致性级别 * ： `strong-global` （对象元数据会立即分发到所有站点。）

当客户端将对象存储到网格时， StorageGRID 会创建两个对象副本并将元数据分发到两个站点，然后再向客户端返回成功。

在载入成功消息时，此对象将受到完全保护，不会丢失。例如，如果本地站点在载入后不久丢失，则远程站点上仍存在对象数据和对象元数据的副本。此对象完全可检索。

如果您改用相同的 ILM 规则和 `strong-site` 一致性级别，则在将对象数据复制到远程站点之后，在将对象元数据分发到该远程站点之前，客户端可能会收到一条成功消息。在这种情况下，对象元数据的保护级别与对象数据的保护级别不匹配。如果本地站点在载入后不久丢失，则对象元数据将丢失。无法检索此对象。

一致性级别和 ILM 规则之间的关系可能很复杂。如需帮助，请联系 NetApp 。

请求示例

```
PUT /v1/28544923908243208806/_Swift container_
X-Auth-Token: SGRD_3a877009a2d24cb1801587bfa9050f29
x-ntap-sg-consistency: strong-site
Host: test.com
```

响应

响应 **HTTP** 标头	**Description**
`Date`	响应的日期和时间。
`Connection`	与服务器的连接是打开还是关闭。
`X-Trans-Id`	请求的唯一事务标识符。
`Content-Length`	响应正文的长度。

响应示例

```
HTTP/1.1 204 No Content
Date: Sat, 29 Nov 2015 01:02:18 GMT
Connection: CLOSE
X-Trans-Id: 1936575373
Content-Length: 0
```

配置 REST API 的安全性

您应查看为 REST API 实施的安全措施，并了解如何保护系统安全。

StorageGRID 如何为REST API提供安全性

您应了解 StorageGRID 系统如何为 REST API 实施安全性，身份验证和授权。

StorageGRID 使用以下安全措施。

- 如果为负载平衡器端点配置了 HTTPS ，则客户端与负载平衡器服务的通信将使用 HTTPS 。

  当您 "配置负载平衡器端点"，也可以选择启用HTTP。例如，您可能希望将 HTTP 用于测试或其他非生产目的。

- 默认情况下、StorageGRID 使用HTTPS与存储节点进行客户端通信。

  （可选） "为这些连接启用HTTP"。例如，您可能希望将 HTTP 用于测试或其他非生产目的。

- StorageGRID 与客户端之间的通信使用 TLS 进行加密。
- 无论将负载平衡器端点配置为接受 HTTP 或 HTTPS 连接，网格中的负载平衡器服务和存储节点之间的通信都会进行加密。
- 客户端必须向 StorageGRID 提供 HTTP 身份验证标头，才能执行 REST API 操作。

安全证书和客户端应用程序

客户端可以直接连接到网关节点或管理节点上的负载平衡器服务和存储节点。

在所有情况下，客户端应用程序都可以使用网格管理员上传的自定义服务器证书或 StorageGRID 系统生成的证书进行 TLS 连接：

- 当客户端应用程序连接到负载平衡器服务时，它们会使用为用于建立连接的特定负载平衡器端点配置的证书来执行此操作。每个端点都有自己的证书，此证书可以是网格管理员上传的自定义服务器证书，也可以是网格管理员在配置端点时在 StorageGRID 中生成的证书。
- 当客户端应用程序直接连接到存储节点时、它们会使用系统生成的服务器证书、这些证书是在安装StorageGRID 系统时为存储节点生成的(由系统证书颁发机构签名)。 或网格管理员为网格提供的单个自定义服务器证书。

应将客户端配置为信任对用于建立 TLS 连接的任何证书签名的证书颁发机构。

请参见 "正在配置负载平衡器端点" 和 "添加单个自定义服务器证书" 用于直接连接到存储节点的TLS连接。

摘要

下表显示了如何在 S3 和 Swift REST API 中实施安全问题：

Security 问题描述	实施 **REST API**
连接安全性	TLS
服务器身份验证	系统 CA 签名的 X.509 服务器证书或管理员提供的自定义服务器证书
客户端身份验证	• S3： S3 帐户（访问密钥 ID 和机密访问密钥） • Swift： Swift 帐户（用户名和密码）
客户端授权	• S3： 存储分段所有权和所有适用的访问控制策略 • Swift： 管理员角色访问

支持 **TLS** 库的哈希和加密算法

StorageGRID 系统支持一组有限的密码套件，客户端应用程序可在建立传输层安全（ TLS ）会话时使用这些密码套件。要配置加密方法，请进入*configuration*>*Security*>*Security settings，然后选择*TLS和SSH policies*。

支持的 **TLS** 版本

StorageGRID 支持 TLS 1.2 和 TLS 1.3 。

不再支持 SSLv3 和 TLS 1.1 （或更早版本）。

相关信息

"配置租户帐户和连接"

监控和审核操作

您可以通过查看整个网格或特定节点的事务趋势来监控客户端操作的工作负载和效率。您可以使用审核消息监控客户端操作和事务。

监控对象载入和检索速率

您可以监控对象载入和检索速率，以及对象计数，查询和验证的指标。您可以查看客户端应用程序在 StorageGRID 系统中成功尝试读取，写入和修改对象的次数和失败的尝试次数。

步骤

1. 使用登录到网格管理器 "支持的 Web 浏览器"。
2. 在信息板上，选择*Performance*>*S3 operations*或*Performance*>*Swift operations*。

   本节总结了 StorageGRID 系统执行的客户端操作的数量。协议速率是过去两分钟的平均值。

3. 选择 * 节点 * 。
4. 从节点主页（部署级别）中，单击 * 负载平衡器 * 选项卡。

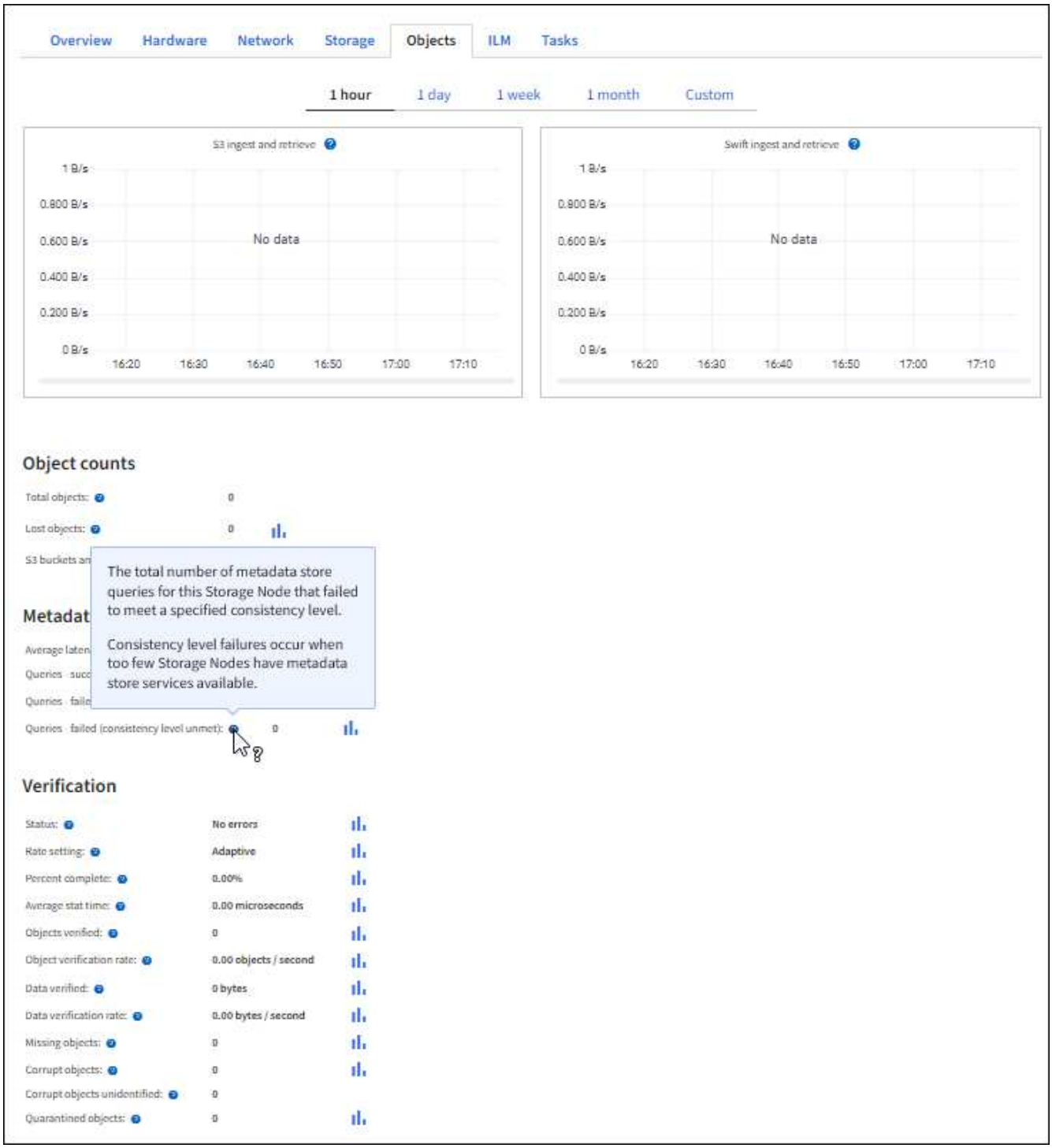
   这些图表显示了定向到网格中负载平衡器端点的所有客户端流量的趋势。您可以选择以小时，天，周，月或年为单位的时间间隔， 或者，您也可以应用自定义间隔。

5. 从节点主页（部署级别）中，单击 * 对象 * 选项卡。

   此图表显示了整个 StorageGRID 系统的载入和检索速率，以每秒字节数和总字节数为单位。您可以选择以小时，天，周，月或年为单位的时间间隔， 或者，您也可以应用自定义间隔。

6. 要查看特定存储节点的信息，请从左侧列表中选择该节点，然后单击 * 对象 * 选项卡。

   此图表显示了此存储节点的对象载入速率和检索速率。此选项卡还包括对象计数，查询和验证的指标。您可以单击这些标签以查看这些指标的定义。



7. 如果您需要更多详细信息：
   a. 选择 * 支持 * > * 工具 * > * 网格拓扑 * 。
   b. 选择 * 站点 _* > * 概述 * > * 主要 * 。

      API Operations 部分显示整个网格的摘要信息。

   c. 选择 * 存储节点 _* > * LDR* > * 客户端应用程序 _* > * 概述 * > * 主 *

      操作部分显示选定存储节点的摘要信息。

访问和查看审核日志

审核消息由 StorageGRID 服务生成并存储在文本日志文件中。审核日志中特定于 API 的审核消息提供关键的安全性，操作和性能监控数据，可帮助您评估系统的运行状况。

开始之前

- 您必须具有特定的访问权限。
- 您必须具有 `Passwords.txt` 文件
- 您必须知道管理节点的 IP 地址。

关于此任务

。 "活动审核日志文件" 名为 `audit.log`、并存储在管理节点上。

每天保存一次活动的 audit.log 文件，并启动一个新的 audit.log 文件。已保存文件的名称以格式指示其保存的时间 `yyyy-mm-dd.txt`。

一天之后、保存的文件将按格式进行压缩和重命名 `yyyy-mm-dd.txt.gz`、用于保留原始日期。

此示例显示了活动的audit.log文件、前一天的文件(2018-04-15.txt)以及前一天的压缩文件 (`2018-04-14.txt.gz`) 。

```
audit.log
2018-04-15.txt
2018-04-14.txt.gz
```

步骤

1. 登录到管理节点：
   a. 输入以下命令： `ssh admin@primary_Admin_Node_IP`
   b. 输入中列出的密码 `Passwords.txt` 文件
   c. 输入以下命令切换到root： `su -`
   d. 输入中列出的密码 `Passwords.txt` 文件

   以root用户身份登录后、提示符将从变为 `$` to `#`。
2. 转到包含审核日志文件的目录：`cd /var/local/audit/export`
3. 根据需要查看当前审核日志文件或已保存的审核日志文件。

审核日志中跟踪的 **Swift** 操作

系统会在中跟踪所有成功的存储删除、获取、机头、发布和放置操作 "StorageGRID 审核日志"。不会记录故障、也不会记录信息、身份验证或选项请求。

跟踪以下Swift操作的信息。

帐户操作

- "获取帐户"
- "主帐户"

容器操作

- "删除容器"
- "获取容器"
- "机头容器"
- "放入容器"

对象操作

- "删除对象"
- "获取对象"
- "head 对象"
- "PUT 对象"

版权信息

版权所有 © 2025 NetApp, Inc.。保留所有权利。中国印刷。未经版权所有者事先书面许可，本文档中受版权保护的任何部分不得以任何形式或通过任何手段（图片、电子或机械方式，包括影印、录音、录像或存储在电子检索系统中）进行复制。

从受版权保护的 NetApp 资料派生的软件受以下许可和免责声明的约束：

本软件由 NetApp 按“原样”提供，不含任何明示或暗示担保，包括但不限于适销性以及针对特定用途的适用性的隐含担保，特此声明不承担任何责任。在任何情况下，对于因使用本软件而以任何方式造成的任何直接性、间接性、偶然性、特殊性、惩罚性或后果性损失（包括但不限于购买替代商品或服务；使用、数据或利润方面的损失；或者业务中断），无论原因如何以及基于何种责任理论，无论出于合同、严格责任或侵权行为（包括疏忽或其他行为），NetApp 均不承担责任，即使已被告知存在上述损失的可能性。

NetApp 保留在不另行通知的情况下随时对本文档所述的任何产品进行更改的权利。除非 NetApp 以书面形式明确同意，否则 NetApp 不承担因使用本文档所述产品而产生的任何责任或义务。使用或购买本产品不表示获得 NetApp 的任何专利权、商标权或任何其他知识产权许可。

本手册中描述的产品可能受一项或多项美国专利、外国专利或正在申请的专利的保护。

有限权利说明：政府使用、复制或公开本文档受 DFARS 252.227-7013（2014 年 2 月）和 FAR 52.227-19（2007 年 12 月）中“技术数据权利 — 非商用”条款第 (b)(3) 条规定的限制条件的约束。

本文档中所含数据与商业产品和/或商业服务（定义见 FAR 2.101）相关，属于 NetApp, Inc. 的专有信息。根据本协议提供的所有 NetApp 技术数据和计算机软件具有商业性质，并完全由私人出资开发。 美国政府对这些数据的使用权具有非排他性、全球性、受限且不可撤销的许可，该许可既不可转让，也不可再许可，但仅限在与交付数据所依据的美国政府合同有关且受合同支持的情况下使用。除本文档规定的情形外，未经 NetApp, Inc. 事先书面批准，不得使用、披露、复制、修改、操作或显示这些数据。美国政府对国防部的授权仅限于 DFARS 的第 252.227-7015(b)（2014 年 2 月）条款中明确的权利。

商标信息

NetApp、NetApp 标识和 http://www.netapp.com/TM 上所列的商标是 NetApp, Inc. 的商标。其他公司和产品名称可能是其各自所有者的商标。