使用S3 REST API

StorageGRID

NetApp
October 03, 2025

目录

使用 S3

了解客户端应用程序如何使用S3 API与StorageGRID 系统相连接。

- "支持S3 REST API"
- "配置租户帐户和连接"
- "StorageGRID 如何实施S3 REST API"
- "S3 REST API 支持的操作和限制"
- "StorageGRID S3 REST API 操作"
- "存储分段和组访问策略"
- "为REST API配置安全性"
- "监控和审核操作"
- "活动，空闲和并发 HTTP 连接的优势"

支持S3 REST API

StorageGRID 支持简单存储服务（ S3 ） API ，该 API 作为一组表示状态传输（ Representational State Transfer ， REST ） Web 服务来实施。通过对 S3 REST API 的支持，您可以将为 S3 Web 服务开发的面向服务的应用程序与使用 StorageGRID 系统的内部对象存储连接起来。这需要对客户端应用程序当前使用 S3 REST API 调用的情况进行最少的更改。

- "对 S3 REST API 支持的更改"
- "支持的版本"
- "支持 StorageGRID 平台服务"

对 S3 REST API 支持的更改

您应了解 StorageGRID 系统对 S3 REST API 的支持发生了哪些变化。

版本。	注释
11.5	• 增加了对管理存储分段加密的支持。 • 增加了对 S3 对象锁定和已弃用旧合规性请求的支持。 • 增加了对在版本控制的存储分段上使用删除多个对象的支持。 • 。 `Content-MD5` 现在已正确支持请求标头。

版本。	注释
11.4	• 增加了对删除存储分段标记，获取存储分段标记和放置存储分段标记的支持。不支持成本分配标记。 • 对于在 StorageGRID 11.4 中创建的分段，不再需要限制对象密钥名称以满足性能最佳实践。 • 增加了对存储分段通知的支持 `s3:ObjectRestore:Post` 事件类型。 • 现在，多部件的 AWS 大小限制已强制实施。多部分上传中的每个部件必须介于 5 MiB 和 5 GiB 之间。最后一个部件可以小于 5 MiB 。 • 增加了对 TLS 1.3 的支持，并更新了支持的 TLS 密码套件列表。 • CLB 服务已弃用。
11.3	• 增加了对使用客户提供的密钥（ SSI-C ）对对象数据进行服务器端加密的支持。 • 增加了对删除、获取和放置分段生命周期操作(仅限到期操作)和的支持 `x-amz-expiration` 响应标头。 • 更新了 PUT 对象， PUT 对象 - 复制和多部件上传，以说明在载入时使用同步放置的 ILM 规则的影响。 • 更新了支持的 TLS 密码套件列表。不再支持 TLS 1.1 密码。
11.2.	增加了对用于云存储池的后对象还原的支持。增加了对在组和存储分段策略中使用 AWS 语法来处理 ARN ，策略条件密钥和策略变量的支持。仍支持使用 StorageGRID 语法的现有组和存储分段策略。 • 注意： * 在其他配置 JSON/XML 中使用 ARN/URN 的情况没有改变，包括在自定义 StorageGRID 功能中使用的情况。
11.1	增加了对跨源资源共享（ CORS ）， S3 客户端到网格节点连接的 HTTP 以及存储分段上的合规性设置的支持。
11.0	增加了对为存储分段配置平台服务（ CloudMirror 复制，通知和 Elasticsearch 搜索集成）的支持。此外，还增加了对存储分段的对象标记位置约束以及可用一致性控制设置的支持。

版本。	注释
10.4.	增加了对版本控制，端点域名页面更新，策略中的条件和变量，策略示例以及 PutOverwriteObject 权限的 ILM 扫描更改的支持。
10.3	增加了对版本控制的支持。
10.2	增加了对组和存储分段访问策略以及多部件副本（上传部件 - 复制）的支持。
10.1	增加了对多部分上传，虚拟托管模式请求和 v4 身份验证的支持。
10.0	StorageGRID 系统最初支持 S3 REST API 。当前支持的 _Simple Storage Service API 参考版本为 2006-03-01 。

支持的版本

StorageGRID 支持以下特定版本的 S3 和 HTTP 。

项目	**version**
S3 规范	_Simple Storage Service API 参考 _ 2006-03-01
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

"IETF RFC 2616 ： 超文本传输协议（ HTTP/1.1 ）"

"Amazon Web Services （ AWS ）文档： Amazon Simple Storage Service API 参考"

支持 **StorageGRID** 平台服务

通过 StorageGRID 平台服务， StorageGRID 租户帐户可以利用远程 S3 存储分段，简单通知服务（ SNS ）端点或 Elasticsearch 集群等外部服务来扩展网格提供的服务。

下表总结了可用的平台服务以及用于配置这些服务的 S3 API 。

平台服务	目的	用于配置服务的 **S3 API**
CloudMirror 复制	将对象从源 StorageGRID 存储分段复制到已配置的远程 S3 存储分段。	PUT 存储分段复制
通知	将有关源 StorageGRID 存储分段中事件的通知发送到已配置的简单通知服务（ Simple Notification Service ， SNS ）端点。	PUT 存储分段通知
搜索集成	将存储在 StorageGRID 存储分段中的对象的对象元数据发送到已配置的 Elasticsearch 索引。	PUT 存储分段元数据通知 • 注： * 这是 StorageGRID 自定义 S3 API 。

网格管理员必须先为租户帐户启用平台服务，然后才能使用这些服务。然后，租户管理员必须在租户帐户中创建一个表示远程服务的端点。要配置服务，必须执行此步骤。

使用平台服务的建议

在使用平台服务之前，您必须了解以下建议：

- NetApp 建议，对于需要进行 CloudMirror 复制，通知和搜索集成的 S3 请求，您允许的活动租户不超过 100 个。如果活动租户超过 100 个，则可能会导致 S3 客户端性能下降。
- 如果 StorageGRID 系统中的 S3 存储分段同时启用了版本控制和 CloudMirror 复制， NetApp 建议目标端点也启用 S3 存储分段版本控制。这样， CloudMirror 复制就可以在端点上生成类似的对象版本。
- 如果源存储分段启用了 S3 对象锁定，则不支持 CloudMirror 复制。
- 如果目标存储分段启用了原有合规性，则 CloudMirror 复制将失败并显示 AccessDenied 错误。

相关信息

"使用租户帐户"

"管理 StorageGRID"

"对存储分段执行的操作"

"PUT 存储分段元数据通知配置请求"

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**S3**租户帐户

S3 API 客户端必须先具有 S3 租户帐户，然后才能在 StorageGRID 上存储和检索对象。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。

S3 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。创建 S3 租户帐户时，网格管理员会指定以下信息：

- 租户的显示名称（租户的帐户 ID 会自动分配，不能更改）。
- 是否允许租户帐户使用平台服务。如果允许使用平台服务，则必须对网格进行配置，以支持使用这些服务。
- （可选）租户帐户的存储配额—租户对象可用的最大 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）。
- 如果为 StorageGRID 系统启用了身份联合，则哪个联合组具有 " 根访问 " 权限来配置租户帐户。
- 如果 StorageGRID 系统未使用单点登录（ SSO ），则表示租户帐户是使用自己的身份源还是共享网格的身份源，以及租户的本地 root 用户的初始密码。

创建 S3 租户帐户后，租户用户可以访问租户管理器以执行如下任务：

- 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
- 管理 S3 访问密钥
- 创建和管理 S3 存储分段，包括启用了 S3 对象锁定的存储分段
- 使用平台服务（如果已启用）
- 监控存储使用情况

> S3 租户用户可以使用租户管理器创建和管理 S3 存储分段，但他们必须具有 S3 访问密钥，并使用 S3 REST API 载入和管理对象。

相关信息

"管理 StorageGRID"

"使用租户帐户"

如何配置客户端连接

网格管理员可以选择影响 S3 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以下任一项来存储或检索对象：

- 管理节点或网关节点上的负载平衡器服务，或者也可以是管理节点或网关节点高可用性（ HA ）组的虚拟 IP 地址
- 网关节点上的 CLB 服务，或者也可以是网关节点高可用性组的虚拟 IP 地址

> CLB 服务已弃用。在 StorageGRID 11.3 版本之前配置的客户端可以继续在网关节点上使用 CLB 服务。所有其他依靠 StorageGRID 提供负载平衡的客户端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）。

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。

有关每个选项的详细信息，请参见有关管理 StorageGRID 的说明。

相关信息

"管理 StorageGRID"

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。有关详细信息，请与 StorageGRID 管理员联系，或者参见有关管理问题描述 StorageGRID 的说明，了解如何在网格管理器中查找此信息。

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
HA 组	CLB • 注： * CLB 服务已弃用。	HA 组的虚拟 IP 地址	默认 S3 端口： • HTTPS： 8082 • HTTP： 8084
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

建立连接的位置	客户端连接到的服务	IP 地址	Port
网关节点	CLB • 注： * CLB 服务已弃用。	网关节点的 IP 地址 • 注： * 默认情况下，CLB 和 LDR 的 HTTP 端口未启用。	默认 S3 端口： • HTTPS： 8082 • HTTP： 8084
存储节点	LDR	存储节点的 IP 地址	默认 S3 端口： • HTTPS： 18082 • HTTP： 18084

示例

要将 S3 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 192.0.2.5 ，而 S3 负载平衡器端点的端口号为 10443 ，则 S3 客户端可以使用以下 URL 连接到 StorageGRID：

- `https://192.0.2.5:10443`

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。

相关信息

"管理 StorageGRID"

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 连接到存储节点或网关节点上的 CLB 服务，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或网关节点上的 CLB 服务时，它们必须对所有连接使用加密 HTTPS 。您也可以选择网格管理器中的 * 启用 HTTP 连接 * 网格选项来启用不太安全的 HTTP 连接。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 时要小心，因为请求将以未加密方式发送。

> CLB 服务已弃用。

如果选择了 * 启用 HTTP 连接 * 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。请参见有关管理 StorageGRID 的说明。

相关信息

"管理 StorageGRID"

"活动，空闲和并发 HTTP 连接的优势"

S3 请求的端点域名

在对客户端请求使用 S3 域名之前， StorageGRID 管理员必须将系统配置为接受在 S3 路径模式和 S3 虚拟托管模式请求中使用 S3 域名的连接。

关于此任务

要使用 S3 虚拟托管模式请求，网格管理员必须执行以下任务：

- 使用网格管理器将 S3 端点域名添加到 StorageGRID 系统。
- 确保客户端用于与 StorageGRID 的 HTTPS 连接的证书已针对客户端所需的所有域名进行签名。

  例如、如果端点为 `s3.company.com`、网格管理员必须确保用于HTTPS连接的证书包含 `s3.company.com` 端点和端点的通配符使用者备用名称(SAN)： `*.s3.company.com`。

- 配置客户端使用的 DNS 服务器，使其包含与端点域名匹配的 DNS 记录，包括任何所需的通配符记录。

如果客户端使用负载平衡器服务进行连接，则网格管理员配置的证书是客户端使用的负载平衡器端点的证书。

每个负载平衡器端点都有自己的证书，并且可以对每个端点进行配置以识别不同的端点域名。

如果客户端连接存储节点或网关节点上的CLB服务、则网格管理员配置的证书是用于网格的单个自定义服务器证书。

CLB 服务已弃用。

有关详细信息，请参见有关管理 StorageGRID 的说明。

完成这些步骤后、您可以使用虚拟托管模式请求(例如 `bucket.s3.company.com`）。

相关信息

"管理 StorageGRID"

"为REST API配置安全性"

测试S3 REST API配置

您可以使用 Amazon Web Services 命令行界面（ AWS 命令行界面）测试与系统的连接，并验证是否可以向系统读取和写入对象。

您需要的内容

- 您必须已从下载并安装AWS命令行界面 "aws.amazon.com/cli"。
- 您必须已在StorageGRID 系统中创建S3租户帐户。

步骤

1. 配置 Amazon Web Services 设置以使用您在 StorageGRID 系统中创建的帐户：
   a. 进入配置模式： `aws configure`
   b. 输入您创建的帐户的 AWS 访问密钥 ID 。

c. 输入您创建的帐户的 AWS 机密访问密钥。

d. 输入要使用的默认区域，例如 us-east-1 。

e. 输入要使用的默认输出格式，或者按 * 输入 * 选择 JSON 。

2. 创建存储分段。

```
aws s3api --endpoint-url https://10.96.101.17:10443
--no-verify-ssl create-bucket --bucket testbucket
```

如果已成功创建存储分段，则会返回存储分段的位置，如以下示例所示：

```
"Location": "/testbucket"
```

3. 上传对象。

```
aws s3api --endpoint-url https://10.96.101.17:10443 --no-verify-ssl
put-object --bucket testbucket --key s3.pdf --body C:\s3-
test\upload\s3.pdf
```

如果对象上传成功，则返回一个 Etag ，该 Etag 是对象数据的哈希。

4. 列出存储分段的内容以验证是否已上传此对象。

```
aws s3api --endpoint-url https://10.96.101.17:10443 --no-verify-ssl
list-objects --bucket testbucket
```

5. 删除对象。

```
aws s3api --endpoint-url https://10.96.101.17:10443 --no-verify-ssl
delete-object --bucket testbucket --key s3.pdf
```

6. 删除存储分段。

```
aws s3api --endpoint-url https://10.96.101.17:10443 --no-verify-ssl
delete-bucket --bucket testbucket
```

StorageGRID 如何实施S3 REST API

客户端应用程序可以使用S3 REST API调用连接到StorageGRID 来创建、删除和修改存储分段以及存储和检索对象。

- "客户端请求冲突"
- "一致性控制"
- "StorageGRID ILM 规则如何管理对象"
- "对象版本控制"
- "实施S3 REST API的建议"

客户端请求冲突

冲突的客户端请求(例如、写入同一密钥的两个客户端)会按"`latest-WINS`"的原则进行解决。

"`latest-WINS`"评估的时间取决于StorageGRID 系统何时完成给定请求、而不是S3客户端何时开始操作。

一致性控制

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。

默认情况下， StorageGRID 保证新创建的对象的写入后读一致性。成功完成 PUT 后的任何 GET 都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。

如果要在不同的一致性级别执行对象操作，可以为每个分段或每个 API 操作指定一致性控制。

一致性控制

一致性控制会影响 StorageGRID 用于跟踪对象的元数据在节点之间的分布方式，从而影响对象用于客户端请求的可用性。

您可以将存储分段或 API 操作的一致性控制设置为以下值之一：

一致性控制	**Description**
全部	所有节点都会立即接收数据，否则请求将失败。
强大的全局功能	保证所有站点中所有客户端请求的写入后读一致性。
强大的站点	保证站点内所有客户端请求的写入后读一致性。

一致性控制	Description
读后写	（默认）为新对象提供写入后读一致性，并为对象更新提供最终一致性。提供高可用性和数据保护保证。与Amazon S3一致性保证匹配。 • 注意： * 如果应用程序对不存在的对象使用 head 请求，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。要防止出现这些错误，请将一致性控制设置为 "`Available` " ，除非您需要类似于 Amazon S3 的一致性保证。
可用（机头操作的最终一致性）	与 `read-after-new-write` 一致性级别相同，但仅为机头操作提供最终一致性。如果存储节点不可用，则为机头操作提供的可用性比 " `read-after-new-write` " 更高。与 Amazon S3 一致性保证不同，仅适用于机头操作。

使用"`read-after-new-write`"和"`available`"一致性控制

当head或get操作使用`read-after-new-write`一致性控制或get操作使用"`available`"一致性控制时、StorageGRID将执行多个步骤的查找、如下所示：

- 它首先使用低一致性查找对象。
- 如果该查找失败，它会在下一个一致性级别重复执行查找，直到达到最高一致性级别 "`All ，` " ，这要求对象元数据的所有副本都可用。

如果 head 或 get 操作使用 `read-after-new-write` 一致性控制，但对象不存在，则对象查找将始终达到 "`all` " 一致性级别。由于此一致性级别要求对象元数据的所有副本均可用，因此，如果一个或多个存储节点不可用，您可能会收到大量 500 个内部服务器错误。

除非您需要类似于Amazon S3的一致性保证、否则可以通过将一致性控制设置为"`Available"来防止机头操作出现这些错误。**当机头操作使用"`Available`"一致性控制时、`StorageGRID` 仅提供最终一致性。它不会重试失败的操作，直到达到 "`所有 " 一致性级别为止，因此它不要求对象元数据的所有副本都可用。**

指定**API**操作的一致性控制

要为单个 API 操作设置一致性控制，此操作必须支持一致性控制，并且必须在请求标题中指定一致性控制。此示例将 GET 对象操作的一致性控制设置为 `strong-site` 。

```
GET /bucket/object HTTP/1.1
Date: date
Authorization: <em>authorization name</em>
Host: <em>host</em>
Consistency-Control: strong-site
```

PUT 对象和 GET 对象操作必须使用相同的一致性控制。

指定存储分段的一致性控制

要为存储分段设置一致性控制，您可以使用 StorageGRID PUT 存储分段一致性请求和 GET 存储分段一致性请求。或者，您也可以使用租户管理器或租户管理 API 。

为存储分段设置一致性控制时，请注意以下事项：

- 设置存储分段的一致性控制可确定对存储分段中的对象或存储分段配置执行 S3 操作时使用的一致性控制。它不会影响存储分段本身的操作。
- 单个 API 操作的一致性控制将覆盖存储分段的一致性控制。
- 通常，存储分段应使用默认一致性控制 `read-after-new-write` 。 如果请求无法正常工作，请尽可能更改应用程序客户端的行为。或者，配置客户端以指定每个 API 请求的一致性控制。请仅作为最后一种方法在存储分段级别设置一致性控制。

一致性控制和 **ILM** 规则如何交互以影响数据保护

您选择的一致性控制和 ILM 规则都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，而为 ILM 规则选择的载入行为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。

ILM 规则可以使用以下载入行为：

- * 严格 * ： 必须创建 ILM 规则中指定的所有副本，才能将成功返回给客户端。
- * 平衡 * ： StorageGRID 尝试在载入时创建 ILM 规则中指定的所有副本；如果无法创建，则创建临时副本并将成功返回给客户端。在可能的情况下，将创建 ILM 规则中指定的副本。
- * 双提交 * ： StorageGRID 会立即为对象创建临时副本，并将成功返回给客户端。如果可能，将创建 ILM 规则中指定的副本。

> 在为 ILM 规则选择载入行为之前，请阅读有关通过信息生命周期管理管理对象的说明中有关这些设置的完整问题描述 。

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qe之后、但在该站点分发对象元数据之前、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 。

相关信息

"使用 ILM 管理对象"

"获取存储分段一致性请求"

"PUT 存储分段一致性请求"

StorageGRID ILM 规则如何管理对象

网格管理员创建信息生命周期管理（ ILM ）规则，用于管理从 S3 REST API 客户端应用程序输入到 StorageGRID 系统中的对象数据。然后，这些规则将添加到 ILM 策略中，以确定随着时间的推移对象数据的存储方式和存储位置。

ILM 设置可确定对象的以下方面：

- * 地理位置 *

  对象数据在 StorageGRID 系统（存储池）或云存储池中的位置。

- * 存储级别 *

  用于存储对象数据的存储类型：例如闪存或旋转磁盘。

- * 损失保护 *

  创建的副本数量以及创建的副本类型：复制，纠删编码或两者。

- * 保留 *

  随着时间的推移，对象数据的管理方式，存储位置以及保护数据免受丢失的方式会发生变化。

- * 载入期间的保护 *

  用于在载入期间保护对象数据的方法：同步放置（使用平衡或严格的 InGest 行为选项）或创建中间副本（使用双提交选项）。

ILM 规则可以筛选和选择对象。对于使用 S3 载入的对象， ILM 规则可以根据以下元数据筛选对象：

- 租户帐户
- Bucket Name
- 载入时间
- 密钥
- 上次访问时间

默认情况下，对所有 S3 分段禁用上次访问时间更新。如果您的 StorageGRID 系统包含使用上次访问时间选项的 ILM 规则，则必须为该规则中指定的 S3 分段启用对上次访问时间的更新。您可以使用租户管理器中的 PUT 分段上次访问时间请求， * S3* > * 分段 * > * 配置上次访问时间 * 复选框或租户管理 API 启用上次访问时间更新。启用上次访问时间更新时，请注意 StorageGRID 性能可能会降低，尤其是在具有小型对象的系统中。

- 位置限制
- 对象大小
- 用户元数据
- 对象标记

有关 ILM 的详细信息，请参见有关通过信息生命周期管理管理对象的说明。

相关信息

"使用租户帐户"

"使用 ILM 管理对象"

"PUT 分段上次访问时间请求"

对象版本控制

您可以使用版本控制来保留一个对象的多个版本，从而防止意外删除对象，并可用于检索和还原对象的早期版本。

StorageGRID 系统实施版本控制，并支持大多数功能，但存在一些限制。StorageGRID 最多支持 1 ， 000 个对象版本。

对象版本控制可以与 StorageGRID 信息生命周期管理（ ILM ）或 S3 存储分段生命周期配置结合使用。要为每个存储分段启用此功能，您必须明确启用版本控制。存储分段中的每个对象都分配有一个版本 ID ，该 ID 由 StorageGRID 系统生成。

不支持使用 MFA （多因素身份验证） Delete 。

只能在使用 StorageGRID 10.3 或更高版本创建的存储分段上启用版本控制。

ILM 和版本控制

ILM 策略将应用于对象的每个版本。ILM 扫描过程会持续扫描所有对象，并根据当前 ILM 策略重新评估这些对象。对 ILM 策略所做的任何更改都会应用于先前载入的所有对象。如果启用了版本控制，则包括先前载入的版本。ILM 扫描会将新的 ILM 更改应用于先前输入的对象。

对于启用了版本控制的分段中的 S3 对象，版本控制支持允许您创建使用非当前时间作为参考时间的 ILM 规则。对象更新后，其先前版本将变为非最新版本。使用非当前时间筛选器可以创建策略，以减少先前版本的对象对存储的影响。

使用多部分上传操作上传新版本的对象时，原始版本对象的非当前时间反映为新版本创建多部分上传的时间，而不是多部分上传完成的时间。在有限情况下，原始版本的非当前时间可能比当前版本的时间早数小时或数天。

有关 S3 版本控制对象的示例 ILM 策略，请参见有关通过信息生命周期管理管理对象的说明。

相关信息

"使用 ILM 管理对象"

实施**S3 REST API**的建议

在实施用于 StorageGRID 的 S3 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，如果您的应用程序在放置到某个位置之前一直位于某个位置，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 PUT 存储分段一致性请求为每个存储分段设置 "`Available` " 一致性控制，也可以在单个 API 操作的请求标题中指定一致性控制。

对象密钥建议

对于在 StorageGRID 11.4 或更高版本中创建的分段，不再需要限制对象密钥名称以满足性能最佳实践。例如，现在可以对对象密钥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分段，请继续对对象密钥名称遵循以下建议：

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

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

  而不是以下格式：

  ```
  mybucket/f8e3-image3132.jpg
  ```

关于"`范围读取`"的建议

如果选择了*压缩存储的对象*选项(配置>*网格选项*)、则S3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。

相关信息

"一致性控制"

"PUT 存储分段一致性请求"

"管理 StorageGRID"

S3 REST API 支持的操作和限制

StorageGRID 系统实施简单存储服务 API （ API 版本 2006-03-01 ），支持大多数操作，但有一些限制。在集成 S3 REST API 客户端应用程序时，您需要了解实施详细信息。

StorageGRID 系统既支持虚拟托管模式请求，也支持路径模式请求。

- "对请求进行身份验证"
- "对服务执行的操作"
- "对存储分段执行的操作"
- "对存储分段执行自定义操作"
- "对对象执行的操作"
- "多部分上传操作"
- "错误响应"

日期处理

S3 REST API 的 StorageGRID 实施仅支持有效的 HTTP 日期格式。

对于接受日期值的任何标头， StorageGRID 系统仅支持有效的 HTTP 日期格式。日期的时间部分可以使用格林威治标准时间（ GMT ）格式或通用协调时间（ UTC ）格式指定，并且不存在时区偏移（必须指定 +0000 ）。如果包括 `x-amz-date` 标题中指定的任何值。使用AWS签名版本4时、将显示 `x-amz-date` 签名请求中必须存在标题、因为不支持日期标题。

通用请求标头

StorageGRID 系统支持由_Simple Storage Service API参考_定义的通用请求标头、但有一个例外。

请求标题	实施
Authorization	完全支持 AWS 签名版本 2 支持 AWS 签名版本 4 ，但以下情况除外： • 不会为请求正文计算 SHA256 值。接受用户提交的值而不进行验证、就像该值一样 `UNSIGNED-PAYLOAD` 已为提供 `x-amz-content-sha256` 标题。
X-AMZ-securation-token	未实施。返回 `XNotImplemented`。

通用响应标头

StorageGRID 系统支持由 _Simple Storage Service API 参考 _ 定义的所有通用响应标头，但有一个例外。

响应标头	实施
X-AMZ-ID-2	未使用

相关信息

"Amazon Web Services （ AWS ）文档： Amazon Simple Storage Service API 参考"

对请求进行身份验证

StorageGRID 系统支持使用 S3 API 对对象进行身份验证和匿名访问。

S3 API 支持签名版本 2 和签名版本 4 对 S3 API 请求进行身份验证。

经过身份验证的请求必须使用您的访问密钥 ID 和机密访问密钥进行签名。

StorageGRID 系统支持两种身份验证方法：HTTP `Authorization` 标题和使用查询参数。

使用**HTTP**授权标头

HTTP `Authorization` 标头由所有S3 API操作使用、但在存储分段策略允许的情况下使用匿名请求除外。。 `Authorization` 标头包含对请求进行身份验证所需的所有签名信息。

使用查询参数

您可以使用查询参数向 URL 添加身份验证信息。这称为对 URL 进行预签名，可用于授予对特定资源的临时访问权限。使用预签名 URL 的用户无需知道机密访问密钥即可访问资源，这样您就可以为资源提供第三方受限访问权限。

对服务执行的操作

StorageGRID 系统支持对该服务执行以下操作。

操作	实施
获取服务	在所有 Amazon S3 REST API 行为下实施。
获取存储使用量	" 获取存储使用量 " 请求会告知您帐户正在使用的存储总量以及与帐户关联的每个存储分段的存储总量。这是对服务执行的操作、路径为/、并具有自定义查询参数 (`?x-ntap-sg-usage`)。
选项 /	客户端应用程序可以使用问题描述 `OPTIONS /` 向存储节点上的S3端口发出请求、但不提供S3身份验证凭据、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。

相关信息

"获取存储使用情况请求"

对存储分段执行的操作

对于每个 S3 租户帐户， StorageGRID 系统最多支持 1 ， 000 个分段。

存储分段名称限制遵循 AWS US 标准区域限制，但您应进一步将其限制为 DNS 命名约定，以便支持 S3 虚拟托管模式请求。

"Amazon Web Services （ AWS ）文档：存储分段限制"

"S3请求的端点域名"

获取分段（列出对象）和获取分段版本操作支持 StorageGRID 一致性控制。

您可以检查是否已为各个存储分段启用上次访问时间更新。

下表介绍了 StorageGRID 如何实施 S3 REST API 存储分段操作。要执行其中任何操作，必须为帐户提供必要的访问凭据。

操作	实施
删除存储分段	在所有 Amazon S3 REST API 行为下实施。
删除存储分段或	此操作将删除存储分段的 CORS 配置。
删除存储分段加密	此操作将从存储分段中删除默认加密。现有加密对象保持加密状态，但添加到存储分段中的任何新对象不会加密。
删除存储分段生命周期	此操作将从存储分段中删除生命周期配置。

操作	实施
删除存储分段策略	此操作将删除附加到存储分段的策略。
删除存储分段复制	此操作将删除附加到存储分段的复制配置。
删除存储分段标记	此操作使用 `tagging` 用于从存储分段中删除所有标记的子资源。
获取分段（列出对象）版本 1 和版本 2	此操作将返回一个存储分段中的部分或全部（最多 1 ， 000 个）对象。对象的存储类可以具有两个值之一、即使对象是随一起载入的 `REDUCED_REDUNDANCY` 存储类选项： • `STANDARD`、表示对象存储在由存储节点组成的存储池中。 • `GLACIER`、表示对象已移至云存储池指定的外部存储分段。 如果存储分段包含大量前缀相同的已删除密钥、则响应可能包括一些密钥 `CommonPrefixes` 不包含密钥的。
获取分段 ACL	此操作将返回肯定响应以及存储分段所有者的 ID ， DisplayName 和权限，指示所有者对存储分段具有完全访问权限。
获取分段存储器	此操作将返回 `cors` 存储分段的配置。
获取存储分段加密	此操作将返回存储分段的默认加密配置。
获取存储分段生命周期	此操作将返回存储分段的生命周期配置。
获取存储分段位置	此操作将返回使用设置的区域 `LocationConstraint` PUT分段请求中的元素。如果存储分段的区域为 `us-east-1`、将返回该区域的空字符串。
获取存储分段通知	此操作将返回附加到存储分段的通知配置。
获取 Bucket 对象版本	如果对存储分段具有读取访问权限、则此操作将使用 `versions` 子资源列出了存储分段中所有版本对象的元数据。
获取存储分段策略	此操作将返回附加到存储分段的策略。

操作	实施
获取存储分段复制	此操作将返回附加到存储分段的复制配置。
获取存储分段标记	此操作使用 `tagging` 用于返回存储分段的所有标记的子资源。
获取存储分段版本控制	此实施使用 `versioning` 用于返回存储分段版本控制状态的子资源。返回的版本控制状态指示存储分段是"`未版本控制`"还是存储分段是版本"`已启用`"或"已使用`S"。`
获取对象锁定配置	此操作将确定是否为存储分段启用了S3对象锁定。 "使用 S3 对象锁定"
头存储分段	此操作将确定某个存储分段是否存在，并且您有权访问该存储分段。

操作	实施
放入存储分段	此操作将创建一个新存储分段。创建存储分段后，您就会成为存储分段所有者。 • 存储分段名称必须符合以下规则： ◦ 每个 StorageGRID 系统必须是唯一的（而不仅仅是租户帐户中的唯一）。 ◦ 必须符合 DNS 要求。 ◦ 必须至少包含 3 个字符，并且不能超过 63 个字符。 ◦ 可以是一个或多个标签的序列，并使用一个句点分隔相邻标签。每个标签必须以小写字母或数字开头和结尾，并且只能使用小写字母，数字和连字符。 ◦ 不能与文本格式的 IP 地址类似。 ◦ 不应在虚拟托管模式请求中使用句点。句点会在验证服务器通配符证书时出现发生原因 问题。 • 默认情况下、将在中创建分段 `us-east-1` 区域；但是、您可以使用 `LocationConstraint` 请求正文中的请求元素以指定其他区域。使用时 `LocationConstraint` Element中、您必须指定已使用网格管理器或网格管理API定义的区域的确切名称。如果您不知道应使用的区域名称，请联系您的系统管理员。* 注 * ： 如果 PUT 存储分段请求使用的区域尚未在 StorageGRID 中定义，则会发生错误。 • 您可以包括 `x-amz-bucket-object-lock-enabled` 请求标题以创建启用了S3对象锁定的存储分段。 创建存储分段时，必须启用 S3 对象锁定。创建存储分段后，您无法添加或禁用 S3 对象锁定。S3 对象锁定需要分段版本控制，在创建分段时会自动启用分段版本控制。 "使用 S3 对象锁定"
放入存储分段箱	此操作会为存储分段设置 CORS 配置，以便存储分段可以处理跨源请求。跨源资源共享（ CORS ）是一种安全机制，允许一个域中的客户端 Web 应用程序访问不同域中的资源。例如、假设您使用名为的S3存储分段 `images` 以存储图形。通过设置的CORS配置 `images` 存储分段中的图像、您可以在网站上显示该存储分段中的图像 `http://www.example.com`。

操作	实施
PUT 存储分段加密	此操作将设置现有存储分段的默认加密状态。启用存储分段级别加密后，添加到存储分段中的任何新对象都会进行加密。 StorageGRID 支持使用 StorageGRID 管理的密钥进行服务器端加密。指定服务器端加密配置规则时、请设置 `SSEAlgorithm` 参数设置为 `AES256`和、请勿使用 `KMSMasterKeyID` 参数。 如果对象上传请求已指定加密(即、如果请求包含)、则存储分段默认加密配置将被忽略 `x-amz-server-side-encryption-*` 请求标题)。
PUT 存储分段生命周期	此操作将为存储分段创建新的生命周期配置或替换现有的生命周期配置。StorageGRID 在一个生命周期配置中最多支持 1 ， 000 条生命周期规则。每个规则可以包含以下 XML 元素： • 到期日期（天，日期） • 非当前版本到期（非当前日期） • 筛选器（前缀，标记） • Status • ID StorageGRID 不支持以下操作： • AbortIncompleteMultipartUpload • ExpiredObjectDeleteMarker • 过渡 要了解存储分段生命周期中的到期操作如何与 ILM 放置说明交互，请参见使用信息生命周期管理功能管理对象的说明中的 "`ILM 如何在对象的整个生命周期内运行` " 。 • 注 * ： 存储分段生命周期配置可用于启用了 S3 对象锁定的存储分段，但传统合规存储分段不支持存储分段生命周期配置。

操作	实施

PUT 存储分段通知	此操作将使用请求正文中包含的通知配置 XML 为存储分段配置通知。您应了解以下实施详细信息：
	
	• StorageGRID 支持将简单通知服务（ SNS ）主题作为目标。不支持简单队列服务（ SQS ）或 Amazon Lambda 端点。
	• 必须将通知目标指定为 StorageGRID 端点的 URN 。可以使用租户管理器或租户管理 API 创建端点。
	
	要成功配置通知，端点必须存在。如果端点不存在、则为 `400 Bad Request` 返回错误并显示代码 `InvalidArgument`。
	
	• 您不能为以下事件类型配置通知。这些事件类型 * 不 * 受支持。
	
	◦ `s3:ReducedRedundancyLostObject`
	
	◦ `s3:ObjectRestore:Completed`
	
	• 从 StorageGRID 发送的事件通知使用标准 JSON 格式，只是它们不包含某些密钥，而对其他密钥使用特定值，如以下列表所示：
	• * 事件源 *
	
	`sgws:s3`
	
	• * awsRegion*
	
	不包括
	
	• * 。 x-AMZ-id-2*
	
	不包括
	
	• * arn*
	
	`urn:sgws:s3:::bucket_name`

PUT 存储分段策略	此操作将设置附加到存储分段的策略。

操作	实施
PUT 存储分段复制	此操作将使用请求正文中提供的复制配置 XML 为存储分段配置 StorageGRID CloudMirror 复制。对于 CloudMirror 复制，您应了解以下实施详细信息： • StorageGRID 仅支持复制配置的 V1 。这意味着、StorageGRID 不支持使用 `Filter` Element中的规则、并遵循V1中有关删除对象版本的约定。有关详细信息，请参见有关复制配置的 Amazon 文档。 • 分段复制可以在分版本或未分版本的分段上配置。 • 您可以在复制配置 XML 的每个规则中指定不同的目标存储分段。一个源存储分段可以复制到多个目标存储分段。 • 必须将目标分段指定为租户管理器或租户管理 API 中指定的 StorageGRID 端点的 URN 。 要成功进行复制配置，必须存在此端点。如果端点不存在、则请求将以失败的形式出现 `400 Bad Request`。错误消息显示： `Unable to save the replication policy. The specified endpoint URN does not exist: URN.` • 您无需指定 `Role` 在配置XML中。StorageGRID 不使用此值，如果提交，则会忽略此值。 • 如果在配置XML中省略存储类、则StorageGRID 将使用 `STANDARD` 默认情况下、存储类。 • 如果从源存储分段中删除对象或删除源存储分段本身，则跨区域复制行为如下： ◦ 如果在复制对象或存储分段之前将其删除，则不会复制此对象 / 存储分段，您也不会收到通知。 ◦ 如果您在复制对象或存储分段后将其删除，则 StorageGRID 会对跨区域复制的 V1 遵循标准 Amazon S3 删除行为。

操作	实施
放置存储分段标记	此操作使用 `tagging` 用于为存储分段添加或更新一组标记的子资源。添加存储分段标记时，请注意以下限制： • StorageGRID 和 Amazon S3 为每个存储分段最多支持 50 个标签。 • 与存储分段关联的标记必须具有唯一的标记密钥。一个标记密钥的长度最多可包含 128 个 Unicode 字符。 • 标记值的长度最多可以为 256 个 Unicode 字符。 • 密钥和值区分大小写。
PUT 存储分版本	此实施使用 `versioning` 用于设置现有存储分段的版本控制状态的子资源。您可以使用以下值之一设置版本控制状态： • Enabled ： 为存储分段中的对象启用版本控制。添加到存储分段中的所有对象都会收到唯一的版本 ID 。 • suspended ： 为存储分段中的对象禁用版本控制。添加到存储分段中的所有对象都会收到版本ID `null`。

相关信息

"Amazon Web Services (AWS)文档：跨区域复制"

"一致性控制"

"获取分段上次访问时间请求"

"存储分段和组访问策略"

"使用 S3 对象锁定"

"审核日志中跟踪的 S3 操作"

"使用 ILM 管理对象"

"使用租户帐户"

创建**S3**生命周期配置

您可以创建 S3 生命周期配置，以控制何时从 StorageGRID 系统中删除特定对象。

本节中的简单示例说明了 S3 生命周期配置如何控制从特定 S3 存储分段中删除（过期）某些对象的时间。本节中的示例仅供说明。有关创建S3生命周期配置的完整详细信息、请参见_Amazon Simple Storage Service开发人员指南_中有关对象生命周期管理的章节。请注意， StorageGRID 仅支持到期操作，不支持过渡操作。

"《 Amazon Simple Storage Service 开发人员指南：对象生命周期管理》"

什么是生命周期配置

生命周期配置是一组应用于特定 S3 分段中的对象的规则。每个规则都指定受影响的对象以及这些对象的到期时间（在特定日期或一定天数后）。

StorageGRID 在一个生命周期配置中最多支持 1 ， 000 条生命周期规则。每个规则可以包含以下 XML 元素：

- 到期日期：从对象载入开始，在达到指定日期或达到指定天数时删除对象。
- NoncurrentVersionExpiration ：从对象变为非最新状态开始，在达到指定天数时删除对象。
- 筛选器（前缀，标记）
- Status
- ID

如果将生命周期配置应用于某个存储分段，则存储分段的生命周期设置始终会覆盖 StorageGRID ILM 设置。StorageGRID 使用存储分段的 " 到期 " 设置（而不是 ILM ）来确定是删除还是保留特定对象。

因此，即使 ILM 规则中的放置说明仍适用于某个对象，该对象也可能会从网格中删除。或者，即使对象的任何 ILM 放置指令已失效，该对象也可能会保留在网格中。有关详细信息、请参见使用信息生命周期管理管理对象的说明中的"`ILM在对象的整个生命周期中的运行方式`"。

> 存储分段生命周期配置可用于启用了 S3 对象锁定的存储分段，但旧版合规存储分段不支持存储分段生命周期配置。

StorageGRID 支持使用以下存储分段操作来管理生命周期配置：

- 删除存储分段生命周期
- 获取存储分段生命周期
- PUT 存储分段生命周期

创建生命周期配置

作为创建生命周期配置的第一步，您需要创建一个包含一个或多个规则的 JSON 文件。例如，此 JSON 文件包含三个规则，如下所示：

1. 规则1仅适用于与前缀匹配的对象 `category1/`并且具有 `key2` 的值 `tag2`。。 `Expiration` 参数指定与筛选器匹配的对象将在2020年8月22日午夜到期。
2. 规则2仅适用于与前缀匹配的对象 `category2/`。。 `Expiration` 参数指定与筛选器匹配的对象将在载入后100天过期。

   > 指定天数的规则与对象的载入时间相关。如果当前日期超过载入日期加上天数，则在应用生命周期配置后，可能会立即从存储分段中删除某些对象。

3. 规则3仅适用于与前缀匹配的对象 `category3/`。。 `Expiration` 参数指定任何非最新版本的匹配对象将在其变为非最新状态50天后过期。

```
{
    "Rules": [
        {
            "ID": "rule1",
            "Filter": {
                "And": {
                    "Prefix": "category1/",
                    "Tags": [
                        {
                            "Key": "key2",
                            "Value": "tag2"
                        }
                    ]
                }
            },
            "Expiration": {
                "Date": "2020-08-22T00:00:00Z"
            },
            "Status": "Enabled"
        },
        {
            "ID": "rule2",
            "Filter": {
                "Prefix": "category2/"
            },
            "Expiration": {
                "Days": 100
            },
            "Status": "Enabled"
        },
        {
            "ID": "rule3",
            "Filter": {
                "Prefix": "category3/"
            },
            "NoncurrentVersionExpiration": {
                "NoncurrentDays": 50
            },
            "Status": "Enabled"
        }
    ]
}
```

将生命周期配置应用于存储分段

创建生命周期配置文件后，您可以通过发出 PUT 存储分段生命周期请求将其应用于存储分段。

此请求会将示例文件中的生命周期配置应用于名为的存储分段中的对象 `testbucket`：分段

```
aws s3api --endpoint-url <StorageGRID endpoint> put-bucket-lifecycle-
configuration
--bucket testbucket --lifecycle-configuration file://bktjson.json
```

要验证是否已成功将生命周期配置应用于存储分段，请发送问题描述 获取存储分段生命周期请求。例如：

```
aws s3api --endpoint-url <StorageGRID endpoint> get-bucket-lifecycle-
configuration
 --bucket testbucket
```

成功的响应将列出您刚刚应用的生命周期配置。

验证存储分段生命周期到期适用场景 对象

在发出 PUT 对象， HEAD 对象或 GET 对象请求时，您可以确定生命周期配置适用场景 中的到期规则是否为特定对象。如果规则适用、响应将包括 `Expiration` 此参数用于指示对象何时到期以及匹配的到期规则。

由于存储分段生命周期会覆盖ILM、因此 `expiry-date` 显示的是删除对象的实际日期。有关详细信息、请参见执行StorageGRID 管理的说明中的"`如何确定对象保留`"。

例如、此PUT对象请求是在2020年6月22日发出的、并在中放置一个对象 `testbucket` 存储分段。

```
aws s3api --endpoint-url <StorageGRID endpoint> put-object
--bucket testbucket --key obj2test2 --body bktjson.json
```

成功响应表示此对象将在 100 天后（ 2020 年 10 月 1 日）过期，并且与生命周期配置的规则 2 匹配。

```
{
      *"Expiration": "expiry-date=\"Thu, 01 Oct 2020 09:07:49 GMT\", rule-
id=\"rule2\"",
      "ETag": "\"9762f8a803bc34f5340579d4446076f7\""
}
```

例如，此 head Object 请求用于获取测试分段中同一对象的元数据。

```
aws s3api --endpoint-url <StorageGRID endpoint> head-object
--bucket testbucket --key obj2test2
```

成功响应包括对象的元数据，并指示对象将在 100 天后过期，并且与规则 2 匹配。

```
{
      "AcceptRanges": "bytes",
      *"Expiration": "expiry-date=\"Thu, 01 Oct 2020 09:07:48 GMT\", rule-
id=\"rule2\"",
      "LastModified": "2020-06-23T09:07:48+00:00",
      "ContentLength": 921,
      "ETag": "\"9762f8a803bc34f5340579d4446076f7\""
      "ContentType": "binary/octet-stream",
      "Metadata": {}
}
```

相关信息

"对存储分段执行的操作"

"使用 ILM 管理对象"

对存储分段执行自定义操作

StorageGRID 系统支持添加到 S3 REST API 中且特定于系统的自定义存储分段操作。

下表列出了 StorageGRID 支持的自定义存储分段操作。

操作	Description	有关详细信息 ...
获取存储分段一致性	返回应用于特定存储分段的一致性级别。	"获取存储分段一致性请求"
PUT 存储分段一致性	设置应用于特定存储分段的一致性级别。	"PUT 存储分段一致性请求"
获取存储分段上次访问时间	返回为特定存储分段启用还是禁用上次访问时间更新。	"获取分段上次访问时间请求"
PUT 分段上次访问时间	用于启用或禁用特定存储分段的上次访问时间更新。	"PUT 分段上次访问时间请求"
删除存储分段元数据通知配置	删除与特定存储分段关联的元数据通知配置 XML 。	"删除存储分段元数据通知配置请求"

操作	Description	有关详细信息 ...
获取存储分段元数据通知配置	返回与特定存储分段关联的元数据通知配置 XML 。	"获取存储分段元数据通知配置请求"
PUT 存储分段元数据通知配置	配置存储分段的元数据通知服务。	"PUT 存储分段元数据通知配置请求"
为合规性修改存储分段	已弃用且不支持：您无法再在启用合规性的情况下创建新存储分段。	"已弃用：为满足合规性而修改存储分段请求"
获取存储分段合规性	已弃用但受支持：返回当前对现有旧版合规存储分段有效的合规性设置。	"已弃用：获取存储分段合规性请求"
PUT 存储分段合规性	已弃用但受支持：允许您修改现有旧版合规存储分段的合规性设置。	"已弃用： PUT 存储分段合规性请求"

相关信息

"审核日志中跟踪的 S3 操作"

对对象执行的操作

本节介绍 StorageGRID 系统如何对对象实施 S3 REST API 操作。

- "使用 S3 对象锁定"
- "使用服务器端加密"
- "获取对象"
- "HEAD 对象"
- "后对象还原"
- "PUT 对象"
- "PUT 对象—复制"

以下条件适用于所有对象操作：

- 对对象执行的所有操作均支持StorageGRID 一致性控制、但以下操作除外：
  - 获取对象 ACL
  - `OPTIONS /`
  - PUT 对象合法保留
  - 放置对象保留
- 冲突的客户端请求(例如、两个客户端写入同一密钥)将按"`latest-WINS`"的原则进行解决。"`latest-WINS`"评估的时间取决于StorageGRID 系统何时完成给定请求、而不是S3客户端何时开始操作。
- StorageGRID 存储分段中的所有对象均归存储分段所有者所有，包括由匿名用户或其他帐户创建的对象。

- 无法通过 S3 访问通过 Swift 载入到 StorageGRID 系统的数据对象。

下表介绍了 StorageGRID 如何实施 S3 REST API 对象操作。

操作	实施
删除对象	多因素身份验证(MFA)和响应标头 `x-amz-mfa` 不支持。 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，然后指示客户端成功删除。 • 版本控制 * 要删除特定版本、请求者必须是存储分段所有者并使用 `versionId` 子资源。使用此子资源将永久删除此版本。如果 `versionId` 对应于删除标记、即响应标头 `x-amz-delete-marker` 返回时设置为 `true`。 • 删除对象时不使用 `versionId` 子资源在已启用版本的存储分段上、将生成删除标记。。 `versionId` 对于删除标记、使用返回 `x-amz-version-id` 响应标头和 `x-amz-delete-marker` 返回的响应标头设置为 `true`。 • 删除对象时不使用 `versionId` 子资源在版本暂停的分段上、它会永久删除已存在的"null"版本或"null"删除标记、并生成新的"null"删除标记。。 `x-amz-delete-marker` 返回的响应标头设置为 `true`。 • 注意 *： 在某些情况下，一个对象可能存在多个删除标记。
删除多个对象	多因素身份验证(MFA)和响应标头 `x-amz-mfa` 不支持。 可以在同一请求消息中删除多个对象。

操作	实施
删除对象标记	使用 `tagging` 用于从对象中删除所有标记的子资源。在所有 Amazon S3 REST API 行为下实施。 • 版本控制 * 如果 `versionId` 请求中未指定查询参数、此操作将从受版本控制的存储分段中的对象的最新版本中删除所有标记。如果对象的当前版本为删除标记、则会使用返回`MethodNotAllowed`状态 `x-amz-delete-marker` 响应标头设置为 `true`。
获取对象	"获取对象"
获取对象 ACL	如果为帐户提供了必要的访问凭据，则此操作将返回肯定响应以及对象所有者的 ID ， DisplayName 和权限，指示所有者对对象具有完全访问权限。
获取对象合法保留	"使用 S3 对象锁定"
获取对象保留	"使用 S3 对象锁定"
获取对象标记	使用 `tagging` 子资源以返回对象的所有标记。在所有 Amazon S3 REST API 行为下实施 • 版本控制 * 如果 `versionId` 请求中未指定查询参数、此操作将返回受版本控制的存储分段中对象的最新版本中的所有标记。如果对象的当前版本为删除标记、则会使用返回`MethodNotAllowed`状态 `x-amz-delete-marker` 响应标头设置为 `true`。
HEAD 对象	"HEAD 对象"
后对象还原	"后对象还原"
PUT 对象	"PUT 对象"
PUT 对象—复制	"PUT 对象—复制"
PUT 对象合法保留	"使用 S3 对象锁定"
放置对象保留	"使用 S3 对象锁定"

操作	实施
PUT 对象标记	使用 `tagging` 用于向现有对象添加一组标记的子资源。在所有 Amazon S3 REST API 行为下实施 • 标记更新和载入行为 * 使用 PUT 对象标记更新对象的标记时， StorageGRID 不会重新载入对象。这意味着不会使用匹配 ILM 规则中指定的 " 载入行为 " 选项。通过正常后台 ILM 进程重新评估 ILM 时，更新触发的任何对象放置更改都会进行。 这意味着，如果 ILM 规则对载入行为使用严格选项，则在无法放置所需对象时（例如，由于新需要的位置不可用），不会执行任何操作。更新后的对象会保留其当前位置，直到可以进行所需的位置为止。 • 解决冲突 * 冲突的客户端请求(例如、两个客户端写入同一密钥)将按"`latest-WINS`"的原则进行解决。"`latest-WINS`"评估的时间取决于StorageGRID 系统何时完成给定请求、而不是S3客户端何时开始操作。 • 版本控制 * 如果 `versionId` 未在此请求中指定查询参数、此操作会将标记添加到受版本控制的存储分段中的对象的最新版本。如果对象的当前版本为删除标记、则会使用返回`MethodNotAllowed`状态 `x-amz-delete-marker` 响应标头设置为 `true`。

相关信息

"一致性控制"

"审核日志中跟踪的 S3 操作"

使用 **S3** 对象锁定

如果为 StorageGRID 系统启用了全局 S3 对象锁定设置，则可以在启用了 S3 对象锁定的情况下创建存储分段，然后为添加到存储分段的每个对象版本指定保留日期和合法保留设置。

通过 S3 对象锁定，您可以指定对象级别的设置，以防止对象在固定时间内或无限期地被删除或覆盖。

StorageGRID S3 对象锁定功能提供了一种保留模式，相当于 Amazon S3 合规模式。默认情况下，任何用户都无法覆盖或删除受保护的对象版本。StorageGRID S3 对象锁定功能不支持监管模式，并且不允许具有特殊权限的用户绕过保留设置或删除受保护的对象。

为存储分段启用S3对象锁定

如果为 StorageGRID 系统启用了全局 S3 对象锁定设置，则可以选择在创建每个分段时启用 S3 对象锁定。您可以使用以下任一方法：

- 使用租户管理器创建存储分段。

  "使用租户帐户"

- 使用PUT Bucket请求和创建存储分段 `x-amz-bucket-object-lock_enabled` 请求标题。

  "对存储分段执行的操作"

创建存储分段后，您无法添加或禁用 S3 对象锁定。S3 对象锁定需要分段版本控制，在创建分段时会自动启用分段版本控制。

启用了 S3 对象锁定的存储分段可以包含具有和不具有 S3 对象锁定设置的对象组合。StorageGRID 不支持S3对象锁定分段中的对象的默认保留、因此不支持PUT对象锁定配置分段操作。

确定是否为存储分段启用了S3对象锁定

要确定是否已启用S3对象锁定、请使用获取对象锁定配置请求。

"对存储分段执行的操作"

使用S3对象锁定设置创建对象

要在将对象版本添加到启用了 S3 对象锁定的存储分段时指定 S3 对象锁定设置，请问题描述 对 PUT 对象，PUT 对象 - 复制或启动多部件上传请求。请使用以下请求标头。

> 创建存储分段时，必须启用 S3 对象锁定。创建存储分段后，您无法添加或禁用 S3 对象锁定。

- `x-amz-object-lock-mode`、必须符合要求(区分大小写)。

  > 如果指定 `x-amz-object-lock-mode`、您还必须指定 `x-amz-object-lock-retain-until-date`。

- `x-amz-object-lock-retain-until-date`
  - 保留截止日期值必须采用格式 `2020-08-10T21:46:00Z`。允许使用小数秒，但仅保留 3 位小数（精确度为毫秒）。不允许使用其他 ISO 8601 格式。
  - 保留截止日期必须为未来日期。
- `x-amz-object-lock-legal-hold`

  如果处于合法保留状态（区分大小写），则对象将置于合法保留状态。如果关闭了合法保留，则不会进行合法保留。任何其他值都会导致 400 错误请求（ InvalidArgument ）错误。

如果您使用上述任一请求标头，请注意以下限制：

- 。 `Content-MD5` 如果有、则请求标头为必填项 `x-amz-object-lock-*` PUT对象请求中存在请求标头。

`Content-MD5` PUT对象-复制或启动多部件上传不需要。

- 如果存储分段未启用S3对象锁定和 `x-amz-object-lock-*` 存在请求标头、返回400错误请求(InvalidRequest)错误。
- PUT对象请求支持使用 `x-amz-storage-class: REDUCED_REDUNDANCY` 以匹配AWS行为。但是，如果在启用了 S3 对象锁定的情况下将对象载入存储分段，则 StorageGRID 将始终执行双提交载入。
- 后续的GET或HEAD对象版本响应将包括标题 `x-amz-object-lock-mode`， `x-amz-object-lock-retain-until-date`，和 `x-amz-object-lock-legal-hold`(如果已配置)以及请求发送方是否正确 `s3:Get*` 权限。
- 如果后续的删除对象版本或删除对象版本请求早于保留截止日期或处于合法保留状态，则此请求将失败。

正在更新**S3**对象锁定设置

如果需要更新现有对象版本的合法保留或保留设置，可以执行以下对象子资源操作：

- `PUT Object legal-hold`

  如果新的合法保留值为 on ，则对象将置于合法保留状态。如果合法保留值为 off ，则取消合法保留。

- `PUT Object retention`
  - 模式值必须符合要求（区分大小写）。
  - 保留截止日期值必须采用格式 `2020-08-10T21:46:00Z`。允许使用小数秒，但仅保留 3 位小数（精确度为毫秒）。不允许使用其他 ISO 8601 格式。
  - 如果对象版本具有现有的保留日期，则只能增加此保留日期。新的价值必须是未来的。

相关信息

"使用 ILM 管理对象"

"使用租户帐户"

"PUT 对象"

"PUT 对象—复制"

"启动多部件上传"

"对象版本控制"

"《 Amazon Simple Storage Service 用户指南：使用 S3 对象锁定》"

使用服务器端加密

服务器端加密可用于保护空闲对象数据。StorageGRID 会在写入对象时对数据进行加密，并在您访问对象时对数据进行解密。

如果要使用服务器端加密，可以根据加密密钥的管理方式从两个互斥选项中选择任一选项：

- *SSE （使用 StorageGRID 管理的密钥进行服务器端加密） *：在问题描述 S3 请求以存储对象时，StorageGRID 会使用唯一密钥对对象进行加密。在问题描述 S3 请求以检索对象时， StorageGRID 会使用

存储的密钥对对象进行解密。

- * SSI-C （使用客户提供的密钥进行服务器端加密） *：在问题描述 S3 请求以存储对象时，您可以提供自己的加密密钥。检索对象时，您可以在请求中提供相同的加密密钥。如果这两个加密密钥匹配，则会对对象进行解密，并返回您的对象数据。

  虽然 StorageGRID 负责管理所有对象加密和解密操作，但您必须管理提供的加密密钥。

  > 您提供的加密密钥永远不会存储。如果丢失加密密钥，则会丢失相应的对象。

  > 如果使用 SSE 或 SSI-C 对对象进行加密，则会忽略任何分段级别或网格级别的加密设置。

使用**SSe.**

要使用 StorageGRID 管理的唯一密钥对对象进行加密，请使用以下请求标头：

```
x-amz-server-side-encryption
```

以下对象操作支持此命令头：

- PUT 对象
- PUT 对象—复制
- 启动多部件上传

使用**SSE-C**

要使用您管理的唯一密钥对对象进行加密，请使用三个请求标头：

请求标题	**Description**
`x-amz-server-side-encryption-customer-algorithm`	指定加密算法。标题值必须为 `AES256`。
`x-amz-server-side-encryption-customer-key`	指定用于对对象进行加密或解密的加密密钥。密钥的值必须为 256 位 base64 编码。
`x-amz-server-side-encryption-customer-key-MD5`	根据 RFC 1321 指定加密密钥的 MD5 摘要，用于确保加密密钥的传输没有错误。MD5 摘要的值必须为 base64 编码的 128 位。

以下对象操作支持 SSI-C 请求标头：

- 获取对象
- HEAD 对象
- PUT 对象
- PUT 对象—复制
- 启动多部件上传

- 上传部件
- 上传部件—复制

将服务器端加密与客户提供的密钥（ **SSI-C** ）结合使用的注意事项

在使用 SSI-C 之前，请注意以下注意事项：

- 必须使用 https 。

> 使用 SSI-C 时， StorageGRID 会拒绝通过 http 发出的任何请求出于安全考虑，您应考虑使用 http 意外发送的任何密钥受到损坏。丢弃该密钥，并根据需要旋转。

- 响应中的 ETag 不是对象数据的 MD5 。
- 您必须管理加密密钥到对象的映射。StorageGRID 不存储加密密钥。您负责跟踪为每个对象提供的加密密钥。
- 如果您的存储分段已启用版本控制，则每个对象版本都应具有自己的加密密钥。您负责跟踪每个对象版本使用的加密密钥。
- 由于您在客户端上管理加密密钥，因此您还必须在客户端上管理任何其他保护措施，例如密钥轮换。

> 您提供的加密密钥永远不会存储。如果丢失加密密钥，则会丢失相应的对象。

- 如果为存储分段配置了 CloudMirror 复制，则无法载入 SSI-C 对象。载入操作将失败。

相关信息

"获取对象"

"HEAD 对象"

"PUT 对象"

"PUT 对象—复制"

"启动多部件上传"

"上传部件"

"上传部件—复制"

"Amazon S3 开发人员指南：使用客户提供的加密密钥（ SSI-C ）使用服务器端加密保护数据"

获取对象

您可以使用 S3 GET 对象请求从 S3 存储分段检索对象。

不支持**partnumber**请求参数

。 `partNumber` GET对象请求不支持请求参数。您不能执行获取请求来检索多部件对象的特定部分。返回501未实施错误、并显示以下消息：

```
GET Object by partNumber is not implemented
```

使用客户提供的加密密钥（ **SSI-C** ）进行服务器端加密的请求标头

如果使用您提供的唯一密钥对对象进行加密，请使用所有三个标头。

- `x-amz-server-side-encryption-customer-algorithm`：指定 `AES256`。
- `x-amz-server-side-encryption-customer-key`：指定对象的加密密钥。
- `x-amz-server-side-encryption-customer-key-MD5`：指定对象加密密钥的MD5摘要。

> 您提供的加密密钥永远不会存储。如果丢失加密密钥，则会丢失相应的对象。在使用客户提供的密钥保护对象数据之前、请查看"`使用服务器端加密中的注意事项。`

用户元数据中的 **UTF-8** 字符

StorageGRID 不会解析或解释用户定义的元数据中的转义 UTF-8 字符。对于用户定义的元数据中存在转义UTF-8字符的对象、获取请求不会返回 `x-amz-missing-meta` 如果密钥名称或值包含不可打印的字符、则为标题。

请求标头不受支持

不支持以下请求标头、并返回 `XNotImplemented`：

- `x-amz-website-redirect-location`

版本控制

如果为 `versionId` 未指定子资源、此操作将提取受版本控制的存储分段中的对象的最新版本。如果对象的当前版本为删除标记、则会使用返回"`not found`"状态 `x-amz-delete-marker` 响应标头设置为 `true`。

Cloud Storage Pool 对象的 **GET** 对象行为

如果某个对象已存储在云存储池中（请参见有关通过信息生命周期管理来管理对象的说明），则 GET 对象请求的行为取决于对象的状态。有关详细信息，请参见 "`head Object` " 。

> 如果某个对象存储在云存储池中，并且该对象的一个或多个副本也位于网格中，则获取对象请求将尝试从网格中检索数据，然后再从云存储池中检索数据。

对象的状态	**GET** 对象的行为
对象已载入 StorageGRID 但尚未通过 ILM 进行评估，或者存储在传统存储池中的对象或使用纠删编码	`200 OK` 检索对象的副本。
云存储池中的对象，但尚未过渡到无法检索的状态	`200 OK` 检索对象的副本。

对象的状态	**GET** 对象的行为
对象已过渡到无法检索的状态	`403 Forbidden， InvalidObjectState` 使用 POST 对象还原请求将对象还原到可检索的状态。
正在从不可检索状态还原的对象	`403 Forbidden， InvalidObjectState` 等待 POST 对象还原请求完成。
对象已完全还原到云存储池	`200 OK` 检索对象的副本。

云存储池中的多部分或分段对象

如果您上传的是多部分对象或 StorageGRID 将一个大型对象拆分为多个区块，则 StorageGRID 会通过取样该对象的部分或区块来确定该对象是否在云存储池中可用。在某些情况下、可能会错误地返回GET对象请求 `200 OK` 对象的某些部分已过渡到无法检索的状态、或者对象的某些部分尚未还原。

在这些情况下：

- GET 对象请求可能会返回一些数据，但会在传输过程中停止。
- 可能会返回后续的GET对象请求 `403 Forbidden`。

相关信息

"使用服务器端加密"

"使用 ILM 管理对象"

"后对象还原"

"审核日志中跟踪的 S3 操作"

HEAD 对象

您可以使用 S3 head Object 请求从对象检索元数据，而无需返回对象本身。如果对象存储在云存储池中，则可以使用 head 对象确定对象的过渡状态。

使用客户提供的加密密钥（ **SSI-C** ）进行服务器端加密的请求标头

如果对象使用您提供的唯一密钥进行加密，请使用所有这三个标头。

- `x-amz-server-side-encryption-customer-algorithm`：指定 `AES256`。
- `x-amz-server-side-encryption-customer-key`：指定对象的加密密钥。
- `x-amz-server-side-encryption-customer-key-MD5`：指定对象加密密钥的MD5摘要。

您提供的加密密钥永远不会存储。如果丢失加密密钥，则会丢失相应的对象。在使用客户提供的密钥保护对象数据之前、请查看"`使用服务器端加密中的注意事项。`

用户元数据中的 **UTF-8** 字符

StorageGRID 不会解析或解释用户定义的元数据中的转义 UTF-8 字符。对于用户定义的元数据中具有转义UTF-8字符的对象、如果对该对象发出机头请求、则不会返回 `x-amz-missing-meta` 如果密钥名称或值包含不可打印的字符、则为标题。

请求标头不受支持

不支持以下请求标头、并返回 `XNotImplemented`：

- `x-amz-website-redirect-location`

Cloud Storage Pool 对象的响应标头

如果对象存储在云存储池中（请参见有关通过信息生命周期管理来管理对象的说明），则返回以下响应标头：

- `x-amz-storage-class: GLACIER`
- `x-amz-restore`

响应标头提供了有关对象移动到云存储池，可选择过渡到不可检索状态并已还原时的状态的信息。

对象的状态	对 **head** 对象的响应
对象已载入 StorageGRID 但尚未通过 ILM 进行评估，或者存储在传统存储池中的对象或使用纠删编码	`200 OK` (不返回任何特殊的响应标头。)

云存储池中的对象，但尚未过渡到无法检索的状态	`200 OK`
	
	`x-amz-storage-class: GLACIER`
	
	`x-amz-restore: ongoing-request="false", expiry-date="Sat, 23 July 20 2030 00:00:00 GMT"`
	
	在将对象过渡到无法检索的状态之前、为提供的值 `expiry-date` 设置为未来的某个远程时间。确切的过渡时间不受 StorageGRID 系统控制。

对象的状态	对 **head** 对象的响应

对象已过渡到不可检索状态，但网格上至少也存在一个副本	`200 OK`
	
	`x-amz-storage-class: GLACIER`
	
	`x-amz-restore: ongoing-request="false", expiry-date="Sat, 23 July 20 2030 00:00:00 GMT"`
	
	的值 `expiry-date` 设置为未来的某个远程时间。
	
	• 注意 *：如果网格上的副本不可用（例如，存储节点已关闭），则必须先对后对象还原请求进行问题描述 处理，以便从云存储池还原此副本，然后才能成功检索此对象。

对象已过渡到无法检索的状态，网格上不存在任何副本	`200 OK`
	
	`x-amz-storage-class: GLACIER`

正在从不可检索状态还原的对象	`200 OK`
	
	`x-amz-storage-class: GLACIER`
	
	`x-amz-restore: ongoing-request="true"`

对象已完全还原到云存储池	`200 OK`
	
	`x-amz-storage-class: GLACIER`
	
	`x-amz-restore: ongoing-request="false", expiry-date="Sat, 23 July 20 2018 00:00:00 GMT"`
	
	。 `expiry-date` 指示何时将云存储池中的对象返回到无法检索的状态。

云存储池中的多部分或分段对象

如果您上传的是多部分对象或 StorageGRID 将一个大型对象拆分为多个区块，则 StorageGRID 会通过取样该对象的部分或区块来确定该对象是否在云存储池中可用。在某些情况下、可能会错误地返回HEAD对象请求 `x-amz-restore: ongoing-request="false"` 对象的某些部分已过渡到无法检索的状态、或者对象的某些部分尚未还原。

版本控制

如果为 `versionId` 未指定子资源、此操作将提取受版本控制的存储分段中的对象的最新版本。如果对象的当前版本为删除标记、则会使用返回"`not found`"状态 `x-amz-delete-marker` 响应标头设置为 `true`。

相关信息

"使用服务器端加密"

"使用 ILM 管理对象"

"后对象还原"

"审核日志中跟踪的 S3 操作"

后对象还原

您可以使用 S3 后对象还原请求还原存储在云存储池中的对象。

支持的请求类型

StorageGRID 仅支持后对象还原请求来还原对象。它不支持 `SELECT` 还原类型。选择返回请求 `XNotImplemented`。

版本控制

(可选)指定 `versionId` 还原受版本控制的存储分段中特定版本的对象。如果未指定 `versionId`、将还原对象的最新版本

对云存储池对象执行后对象还原的行为

如果某个对象存储在云存储池中（请参见有关通过信息生命周期管理管理来管理对象的说明），则根据对象的状态，后对象还原请求具有以下行为。有关详细信息，请参见 "`head Object` " 。

> 如果某个对象存储在云存储池中，并且该对象的一个或多个副本也位于网格中，则无需发出后对象还原请求来还原该对象。相反，可以使用 GET 对象请求直接检索本地副本。

对象的状态	**POST** 对象还原的行为
对象已载入 StorageGRID ，但尚未通过 ILM 进行评估，或者对象不在云存储池中	`403 Forbidden， InvalidObjectState`
云存储池中的对象，但尚未过渡到无法检索的状态	`200 OK` 不会进行任何更改。 注意：在将对象过渡到无法检索的状态之前、您无法更改其 `expiry-date`。

对象的状态	**POST** 对象还原的行为
对象已过渡到无法检索的状态	`202 Accepted` 在请求正文中指定的天数内将对象的可检索副本还原到云存储池。在此期间结束时，对象将返回到无法检索的状态。 或者、也可以使用 `Tier` 请求元素以确定还原作业完成所需的时间 (`Expedited`， `Standard`或 `Bulk` ）。如果未指定 `Tier`， `Standard` 已使用层。 注意：如果对象已过渡到S3 Glacier深度归档或云存储池使用Azure Blob Storage、则无法使用还原它 `Expedited` 层。返回以下错误 `403 Forbidden`， `InvalidTier: Retrieval option is not supported by this storage class`。
正在从不可检索状态还原的对象	`409 Conflict`， `RestoreAlreadyInProgress`
对象已完全还原到云存储池	`200 OK` *注意：*如果对象已还原到可检索状态、则可以更改其 `expiry-date` 通过使用新值重新发出POST对象还原请求 `Days`。还原日期将相对于请求时间进行更新。

相关信息

"使用 ILM 管理对象"

"HEAD 对象"

"审核日志中跟踪的 S3 操作"

PUT 对象

您可以使用 S3 PUT 对象请求将对象添加到存储分段中。

解决冲突

冲突的客户端请求(例如、两个客户端写入同一密钥)将按"`latest-WINS`"的原则进行解决。"`latest-WINS`"评估的时间取决于StorageGRID 系统何时完成给定请求、而不是S3客户端何时开始操作。

对象大小

StorageGRID 支持大小高达5 TB的对象。

用户元数据大小

Amazon S3 将每个 PUT 请求标头中用户定义的元数据的大小限制为 2 KB 。StorageGRID 将用户元数据限制为 24 KiB 。用户定义的元数据的大小是通过采用 UTF-8 编码的每个键和值的字节数之和来衡量的。

用户元数据中的 **UTF-8** 字符

如果某个请求在用户定义的元数据的密钥名称或值中包含（未转义） UTF-8 值，则会未定义 StorageGRID 行为。

StorageGRID 不会解析或解释用户定义的元数据的密钥名称或值中包含的转义 UTF-8 字符。转义的 UTF-8 字符被视为 ASCII 字符：

- 如果用户定义的元数据包含转义的 UTF-8 字符，则 PUT ， PUT 对象副本， GET 和 HEAD 请求将成功。
- StorageGRID 不会返回 `x-amz-missing-meta` 如果对密钥名称或值的解释值包含不可打印的字符、则为标题。

对象标记限制

您可以在上传新对象时为其添加标记，也可以将其添加到现有对象中。StorageGRID 和 Amazon S3 对每个对象最多支持 10 个标记。与对象关联的标记必须具有唯一的标记密钥。一个标记密钥的长度最多可以是 128 个 Unicode 字符，而标记值的长度最多可以是 256 个 Unicode 字符。密钥和值区分大小写。

对象所有权

在 StorageGRID 中，所有对象均归存储分段所有者帐户所有，包括由非所有者帐户或匿名用户创建的对象。

支持的请求标头

支持以下请求标头：

- `Cache-Control`
- `Content-Disposition`
- `Content-Encoding`

  指定时 `aws-chunked` 适用于 `Content-Encoding`StorageGRID 不会验证以下各项：

  - StorageGRID 不会验证 `chunk-signature` 针对区块数据。
  - StorageGRID 不会验证您为提供的值 `x-amz-decoded-content-length` 针对对象。
- `Content-Language`
- `Content-Length`
- `Content-MD5`
- `Content-Type`
- `Expires`
- `Transfer-Encoding`

  如果出现、则支持分块传输编码 `aws-chunked` 此外、还会使用有效负载签名。

- `x-amz-meta-`、后跟一个名称-值对、该对包含用户定义的元数据。

  为用户定义的元数据指定名称 - 值对时，请使用以下通用格式：

```
x-amz-meta-<em>name</em>: <em>value</em>
```

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、则必须使用 `creation-time` 作为创建对象时记录的元数据的名称。例如：

```
x-amz-meta-creation-time: 1443399726
```

的值 `creation-time` 评估为自1970年1月1日以来的秒数。

> ILM 规则不能同时使用 * 用户定义的创建时间 * 作为参考时间，也不能使用平衡或严格选项来执行载入行为。创建 ILM 规则时返回错误。

- `x-amz-tagging`
- S3 对象锁定请求标头
  - `x-amz-object-lock-mode`
  - `x-amz-object-lock-retain-until-date`
  - `x-amz-object-lock-legal-hold`

"使用 S3 对象锁定"

- SSA 请求标头：
  - `x-amz-server-side-encryption`
  - `x-amz-server-side-encryption-customer-key-MD5`
  - `x-amz-server-side-encryption-customer-key`
  - `x-amz-server-side-encryption-customer-algorithm`

"S3 REST API 支持的操作和限制"

请求标头不受支持

不支持以下请求标头：

- 。 `x-amz-acl` 不支持请求标头。
- 。 `x-amz-website-redirect-location` 不支持请求标头、将返回 `XNotImplemented`。

存储类选项

。 `x-amz-storage-class` 支持请求标头。为提交的值 `x-amz-storage-class` 影响StorageGRID 在载入期间保护对象数据的方式、而不影响StorageGRID 系统中存储的对象持久副本数(由ILM决定)。

如果与已载入对象匹配的ILM规则对载入行为使用strict选项、则为 `x-amz-storage-class` 标题无效。

可以使用以下值 `x-amz-storage-class`：

- STANDARD （默认）
  - * 双提交 * ： 如果 ILM 规则为载入行为指定了双提交选项，则在载入对象后，系统会立即创建该对象的第二个副本并将其分发到其他存储节点（双提交）。评估 ILM 后， StorageGRID 将确定这些初始临时副本是否满足规则中的放置说明。否则，可能需要在不同位置创建新的对象副本，并且可能需要删除初始中间副本。
  - * 已平衡 * ： 如果 ILM 规则指定 Balified 选项，而 StorageGRID 无法立即创建规则中指定的所有副本，则 StorageGRID 会在不同的存储节点上创建两个临时副本。

    如果StorageGRID 可以立即创建ILM规则(同步放置)中指定的所有对象副本、则会显示 `x-amz-storage-class` 标题无效。

- REDUCED_REDUNDANCY
  - * 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）。
  - * 已平衡 * ： 如果 ILM 规则指定 Balified 选项，则只有在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、则最好使用选项。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、不建议使用此选项。 REDUCED_REDUNDANCY 增加载入期间对象数据丢失的风险。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。

- 注意 * ： 在任意时间段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。

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。

注意：如果要在启用了S3对象锁定的情况下将对象载入存储分段、则 REDUCED_REDUNDANCY 选项将被忽略。如果要将对象载入旧的合规存储分段、则会显示 REDUCED_REDUNDANCY 选项返回错误。StorageGRID 将始终执行双提交载入，以确保满足合规性要求。

服务器端加密的请求标头

您可以使用以下请求标头通过服务器端加密对对象进行加密。SSE 和 SSI-C 选项是互斥的。

- * SSE* ： 如果要使用 StorageGRID 管理的唯一密钥对对象进行加密，请使用以下标题。
  - `x-amz-server-side-encryption`
- * SSI-C* ： 如果要使用您提供和管理的唯一密钥对对象进行加密，请使用所有这三个标头。
  - `x-amz-server-side-encryption-customer-algorithm`：指定 `AES256`。
  - `x-amz-server-side-encryption-customer-key`：指定新对象的加密密钥。
  - `x-amz-server-side-encryption-customer-key-MD5`：指定新对象加密密钥的MD5摘要。
- 注意： * 您提供的加密密钥永远不会存储。如果丢失加密密钥，则会丢失相应的对象。在使用客户提供的密钥保护对象数据之前、请查看"`使用服务器端加密中的注意事项。`

注：如果使用SSE或SSE-C对对象进行加密、则会忽略任何分段级别或网格级别的加密设置。

版本控制

如果为存储分段启用了版本控制、则为唯一的 `versionId` 会自动为所存储对象的版本生成。这 `versionId` 也会使用在响应中返回 `x-amz-version-id` 响应标头。

如果版本控制已暂停、则存储对象版本时为空 `versionId` 如果已存在空版本、则该版本将被覆盖。

相关信息

"使用 ILM 管理对象"

"对存储分段执行的操作"

"审核日志中跟踪的 S3 操作"

"使用服务器端加密"

"如何配置客户端连接"

PUT 对象—复制

您可以使用 S3 PUT 对象 - 复制请求为已存储在 S3 中的对象创建副本。PUT 对象 - 复制操作与执行 GET ，然后执行 PUT 操作相同。

解决冲突

冲突的客户端请求(例如、两个客户端写入同一密钥)将按"`latest-WINS`"的原则进行解决。"`latest-WINS`"评估的时间取决于StorageGRID 系统何时完成给定请求、而不是S3客户端何时开始操作。

对象大小

StorageGRID 支持大小高达5 TB的对象。

用户元数据中的 UTF-8 字符

如果某个请求在用户定义的元数据的密钥名称或值中包含（未转义） UTF-8 值，则会未定义 StorageGRID 行为。

StorageGRID 不会解析或解释用户定义的元数据的密钥名称或值中包含的转义 UTF-8 字符。转义的 UTF-8 字符被视为 ASCII 字符：

- 如果用户定义的元数据包含转义的 UTF-8 字符，则请求将成功。
- StorageGRID 不会返回 `x-amz-missing-meta` 如果对密钥名称或值的解释值包含不可打印的字符、则为标题。

支持的请求标头

支持以下请求标头：

- `Content-Type`

- `x-amz-copy-source`
- `x-amz-copy-source-if-match`
- `x-amz-copy-source-if-none-match`
- `x-amz-copy-source-if-unmodified-since`
- `x-amz-copy-source-if-modified-since`
- `x-amz-meta-`、后跟一个名称-值对、该对包含用户定义的元数据
- `x-amz-metadata-directive`：默认值为 `COPY`、用于复制对象和关联的元数据。

  您可以指定 `REPLACE` 复制对象时覆盖现有元数据、或者更新对象元数据。

- `x-amz-storage-class`
- `x-amz-tagging-directive`：默认值为 `COPY`、用于复制对象和所有标记。

  您可以指定 `REPLACE` 可在复制对象时覆盖现有标记、或更新标记。

- S3 对象锁定请求标头：
  - `x-amz-object-lock-mode`
  - `x-amz-object-lock-retain-until-date`
  - `x-amz-object-lock-legal-hold`

"使用 S3 对象锁定"

- SSA 请求标头：
  - `x-amz-copy-source-server-side-encryption-customer-algorithm`
  - `x-amz-copy-source-server-side-encryption-customer-key`
  - `x-amz-copy-source-server-side-encryption-customer-key-MD5`
  - `x-amz-server-side-encryption`
  - `x-amz-server-side-encryption-customer-key-MD5`
  - `x-amz-server-side-encryption-customer-key`
  - `x-amz-server-side-encryption-customer-algorithm`

"服务器端加密的请求标头"

请求标头不受支持

不支持以下请求标头：

- `Cache-Control`
- `Content-Disposition`
- `Content-Encoding`
- `Content-Language`

- `Expires`
- `x-amz-website-redirect-location`

存储类选项

。 `x-amz-storage-class` 如果匹配的ILM规则指定了双重提交或平衡的载入行为、则支持请求标头、并影响StorageGRID 创建的对象副本数。

- `STANDARD`

  （默认）指定在 ILM 规则使用双提交选项或 balanced-option 回退到创建中间副本时执行双提交载入操作。

- `REDUCED_REDUNDANCY`

  指定在 ILM 规则使用双提交选项或 balanced-option 回退为创建中间副本时执行单提交载入操作。

  如果要在启用了S3对象锁定的情况下将对象载入存储分段、则会显示 `REDUCED_REDUNDANCY` 选项将被忽略。如果要将对象载入旧的合规存储分段、则会显示 `REDUCED_REDUNDANCY` 选项返回错误。StorageGRID 将始终执行双提交载入，以确保满足合规性要求。

在 **PUT** 对象中使用 **x-AMZ-copy-source —**复制

如果源存储分段和密钥、请在中指定 `x-amz-copy-source` 标头与目标分段和密钥不同、源对象数据的副本将写入目标。

如果源和目标匹配、则使用和 `x-amz-metadata-directive` 标头指定为 `REPLACE`、对象的元数据将使用请求中提供的元数据值进行更新。在这种情况下， StorageGRID 不会重新载入对象。这有两个重要后果：

- 您不能使用 PUT 对象 - 复制对现有对象进行原位加密，也不能更改现有对象的加密。如果您提供 `x-amz-server-side-encryption` 标题或 `x-amz-server-side-encryption-customer-algorithm` 标头、StorageGRID 拒绝请求并返回 `XNotImplemented`。
- 不会使用匹配 ILM 规则中指定的 " 载入行为 " 选项。通过正常后台 ILM 进程重新评估 ILM 时，更新触发的任何对象放置更改都会进行。

  这意味着，如果 ILM 规则对载入行为使用严格选项，则在无法放置所需对象时（例如，由于新需要的位置不可用），不会执行任何操作。更新后的对象会保留其当前位置，直到可以进行所需的位置为止。

服务器端加密的请求标头

如果使用服务器端加密，则您提供的请求标头取决于源对象是否已加密以及是否计划对目标对象加密。

- 如果源对象使用客户提供的密钥（ SSI-C ）进行加密，则必须在 PUT Object - Copy 请求中包含以下三个标头，以便可以解密并复制此对象：
  - `x-amz-copy-source-server-side-encryption-customer-algorithm` 指定 `AES256`。
  - `x-amz-copy-source-server-side-encryption-customer-key` 指定在创建源对象时提供的加密密钥。
  - `x-amz-copy-source-server-side-encryption-customer-key-MD5`：指定在创建源对象时提

供的MD5摘要。

- 如果要使用您提供和管理的唯一密钥对目标对象（副本）进行加密，请包含以下三个标题：
  - `x-amz-server-side-encryption-customer-algorithm`：指定 `AES256`。
  - `x-amz-server-side-encryption-customer-key`：为目标对象指定新的加密密钥。
  - `x-amz-server-side-encryption-customer-key-MD5`：指定新加密密钥的MD5摘要。
- 注意： * 您提供的加密密钥永远不会存储。如果丢失加密密钥，则会丢失相应的对象。在使用客户提供的密钥保护对象数据之前、请查看"`使用服务器端加密中的注意事项。`
- 如果要使用由 StorageGRID （ SSE ）管理的唯一密钥对目标对象（副本）进行加密，请将此标头包括在 PUT 对象 - 复制请求中：
  - `x-amz-server-side-encryption`

注意： `server-side-encryption` 无法更新对象的值。而是使用新创建副本 `server-side-encryption` 价值使用 `x-amz-metadata-directive： REPLACE`。

版本控制

如果源存储分段已版本控制、则可以使用 `x-amz-copy-source` 用于复制最新版本对象的标题。要复制对象的特定版本、必须使用明确指定要复制的版本 `versionId` 子资源。如果目标存储分段已进行版本控制、则会在中返回生成的版本 `x-amz-version-id` 响应标头。如果目标分段的版本控制已暂停、则 `x-amz-version-id` 返回"`null`"值。

相关信息

"使用 ILM 管理对象"

"使用服务器端加密"

"审核日志中跟踪的 S3 操作"

"PUT 对象"

多部分上传操作

本节介绍 StorageGRID 如何支持多部件上传操作。

- "列出多部分上传"
- "启动多部件上传"
- "上传部件"
- "上传部件—复制"
- "完成多部件上传"

以下条件和注释适用于所有多部件上传操作：

- 一个存储分段的并发多部件上传数不应超过 1 ， 000 次，因为该存储分段的 List Multipart uploads 查询结果可能会返回不完整的结果。
- StorageGRID 对多部件部件强制实施 AWS 大小限制。S3 客户端必须遵循以下准则：

- 多部分上传中的每个部分必须介于 5 MiB （ 5 ， 242 ， 880 字节）和 5 GiB （ 5 ， 368 ， 709 ， 120 字节）之间。
- 最后一部分可以小于 5 MiB （ 5 ， 242 ， 880 字节）。
- 通常，部件大小应尽可能大。例如，对于 100 GiB 对象，请使用部件大小 5 GiB 。由于每个部件都被视为唯一对象，因此使用较大的部件大小可降低 StorageGRID 元数据开销。
- 对于小于 5 GiB 的对象，请考虑使用非多部分上传。

- 如果 ILM 规则使用严格或平衡的载入行为，则会在载入多部分对象时对其每个部分进行评估，并在多部分上传完成后对该对象作为一个整体进行评估。您应了解这会对对象和部件放置产生何种影响：
  - 如果在 S3 多部分上传过程中 ILM 发生更改，则在多部分上传完成后，对象的某些部分可能无法满足当前的 ILM 要求。任何放置不正确的部件都会排队等待 ILM 重新评估，并稍后移至正确的位置。
  - 在评估某个部件的 ILM 时， StorageGRID 会筛选该部件的大小，而不是对象的大小。这意味着，对象的某些部分可以存储在不满足整个对象的 ILM 要求的位置。例如，如果规则指定所有 10 GB 或更大的对象都存储在 DC1 中，而所有较小的对象存储在 DC2 中，则在载入时， 10 部分多部分上传的每个 1 GB 部分都存储在 DC2 中。在对对象整体进行 ILM 评估时，对象的所有部分都将移至 DC1 。
- 所有多部分上传操作均支持 StorageGRID 一致性控制。
- 您可以根据需要对多部分上传使用服务器端加密。要使用SSE (服务器端加密与StorageGRID管理的密钥)、您需要包括 `x-amz-server-side-encryption` 仅在"启动多部件上传请求"中显示请求标题。要对客户提供的密钥使用 SSI-C （服务器端加密），您可以在 " 启动多部件上传请求 " 和后续的每个 " 上传部件请求 " 中指定相同的三个加密密钥请求标头。

操作	实施
列出多部件上传	请参见 "列出多部件上传"
启动多部件上传	请参见 "启动多部件上传"
上传部件	请参见 "上传部件"
上传部件—复制	请参见 "上传部件—复制"
完成多部件上传	请参见 "完成多部件上传"
中止多部分上传	在所有 Amazon S3 REST API 行为下实施
列出部件	在所有 Amazon S3 REST API 行为下实施

相关信息

"一致性控制"

"使用服务器端加密"

列出多部件上传

" 列出多部件上传 " 操作会列出某个存储分段正在进行的多部件上传。

支持以下请求参数：

- encoding-type
- max-uploads
- key-marker
- prefix
- upload-id-marker

。 delimiter 不支持请求参数。

版本控制

多部分上传包括启动上传，发布上传，上传部件，组装上传的部件以及完成上传的操作。执行完 " 多部分上传 " 操作后，即创建对象（并在适用情况下进行版本控制）。

启动多部件上传

" 启动多部件上传 " 操作将为对象启动多部件上传，并返回上传 ID 。

。 x-amz-storage-class 支持请求标头。为提交的值 x-amz-storage-class 影响StorageGRID 在载入期间保护对象数据的方式、而不影响StorageGRID 系统中存储的对象持久副本数(由ILM决定)。

如果与已载入对象匹配的ILM规则对载入行为使用strict选项、则为 x-amz-storage-class 标题无效。

可以使用以下值 x-amz-storage-class：

- STANDARD （默认）
  - * 双提交 * ： 如果 ILM 规则为载入行为指定了双提交选项，则在载入对象后，系统会立即创建该对象的第二个副本并将其分发到其他存储节点（双提交）。评估 ILM 后， StorageGRID 将确定这些初始临时副本是否满足规则中的放置说明。否则，可能需要在不同位置创建新的对象副本，并且可能需要删除初始中间副本。
  - * 已平衡 * ： 如果 ILM 规则指定 Balified 选项，而 StorageGRID 无法立即创建规则中指定的所有副本，则 StorageGRID 会在不同的存储节点上创建两个临时副本。

    如果StorageGRID 可以立即创建ILM规则(同步放置)中指定的所有对象副本、则会显示 x-amz-storage-class 标题无效。
- REDUCED_REDUNDANCY
  - * 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）。
  - * 已平衡 * ： 如果 ILM 规则指定 Balified 选项，则只有在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、则最好使用选项。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、不建议使用此选项。 REDUCED_REDUNDANCY 增加载入期间对象数据丢失的风险。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。

- 注意 * ：在任意时间段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。

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。

注意：如果要在启用了S3对象锁定的情况下将对象载入存储分段、则 `REDUCED_REDUNDANCY` 选项将被忽略。如果要将对象载入旧的合规存储分段、则会显示 `REDUCED_REDUNDANCY` 选项返回错误。StorageGRID 将始终执行双提交载入，以确保满足合规性要求。

支持以下请求标头：

- `Content-Type`
- `x-amz-meta-`、后跟一个名称-值对、该对包含用户定义的元数据

  为用户定义的元数据指定名称 - 值对时，请使用以下通用格式：

  ```
  x-amz-meta-_name_: `value`
  ```

  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、则必须使用 `creation-time` 作为创建对象时记录的元数据的名称。例如：

  ```
  x-amz-meta-creation-time: 1443399726
  ```

  的值 `creation-time` 评估为自1970年1月1日以来的秒数。

  正在添加 `creation-time` 由于在将对象添加到启用了旧合规性的存储分段时不允许使用用户定义的元数据。此时将返回错误。

- S3 对象锁定请求标头：
  - `x-amz-object-lock-mode`
  - `x-amz-object-lock-retain-until-date`
  - `x-amz-object-lock-legal-hold`

"使用 S3 对象锁定"

- SSA 请求标头：
  - `x-amz-server-side-encryption`
  - `x-amz-server-side-encryption-customer-key-MD5`
  - `x-amz-server-side-encryption-customer-key`
  - `x-amz-server-side-encryption-customer-algorithm`

"S3 REST API 支持的操作和限制"

有关 StorageGRID 如何处理 UTF-8 字符的信息，请参见 PUT 对象的文档。

服务器端加密的请求标头

您可以使用以下请求标头通过服务器端加密对多部分对象进行加密。SSE 和 SSI-C 选项是互斥的。

- * SSE* ： 如果要使用 StorageGRID 管理的唯一密钥对对象进行加密，请在 " 启动多部分上传请求 " 中使用以下标题。请勿在任何上传部件请求中指定此标题。
  - `x-amz-server-side-encryption`
- * SSI-C* ： 如果要使用您提供和管理的唯一密钥对对象进行加密，请在 " 启动多部件上传请求 " （以及后续的每个 " 上传部件请求 " ）中使用所有这三个标头。
  - `x-amz-server-side-encryption-customer-algorithm`：指定 `AES256`。
  - `x-amz-server-side-encryption-customer-key`：指定新对象的加密密钥。
  - `x-amz-server-side-encryption-customer-key-MD5`：指定新对象加密密钥的MD5摘要。
- 注意： * 您提供的加密密钥永远不会存储。如果丢失加密密钥，则会丢失相应的对象。在使用客户提供的密钥保护对象数据之前、请查看"`使用服务器端加密中的注意事项。`

请求标头不受支持

不支持以下请求标头、并返回 `XNotImplemented`

- `x-amz-website-redirect-location`

版本控制

多部分上传包括启动上传，发布上传，上传部件，组装上传的部件以及完成上传的操作。在执行完整的多部件上传操作时，系统会创建对象（如果适用，还会进行版本控制）。

相关信息

"使用 ILM 管理对象"

"使用服务器端加密"

"PUT 对象"

上传部件

" 上传部件 " 操作会通过多部件上传方式为对象上传部件。

支持的请求标头

支持以下请求标头：

- `Content-Length`
- `Content-MD5`

服务器端加密的请求标头

如果您为启动多部件上传请求指定了 SSI-C 加密，则还必须在每个上传部件请求中包含以下请求标头：

- `x-amz-server-side-encryption-customer-algorithm`：指定 `AES256`。
- `x-amz-server-side-encryption-customer-key`：指定您在启动多部件上传请求中提供的相同加密密钥。
- `x-amz-server-side-encryption-customer-key-MD5`：指定您在启动多部件上传请求中提供的相同MD5摘要。

> 您提供的加密密钥永远不会存储。如果丢失加密密钥，则会丢失相应的对象。在使用客户提供的密钥保护对象数据之前、请查看"`使用服务器端加密中的注意事项。`

版本控制

多部分上传包括启动上传，发布上传，上传部件，组装上传的部件以及完成上传的操作。在执行完整的多部件上传操作时，系统会创建对象（如果适用，还会进行版本控制）。

相关信息

"使用服务器端加密"

上传部件—复制

上传部件 - 复制操作通过将现有对象中的数据复制为数据源来上传对象的一部分。

上传部件 - 复制操作可在所有 Amazon S3 REST API 行为下实施。

此请求读取和写入中指定的对象数据 `x-amz-copy-source-range` 在StorageGRID 系统中。

支持以下请求标头：

- `x-amz-copy-source-if-match`
- `x-amz-copy-source-if-none-match`
- `x-amz-copy-source-if-unmodified-since`
- `x-amz-copy-source-if-modified-since`

服务器端加密的请求标头

如果您为启动多部件上传请求指定了 SSI-C 加密，则还必须在每个上传部件 - 复制请求中包含以下请求标头：

- `x-amz-server-side-encryption-customer-algorithm`：指定 `AES256`。
- `x-amz-server-side-encryption-customer-key`：指定您在启动多部件上传请求中提供的相同加密密钥。
- `x-amz-server-side-encryption-customer-key-MD5`：指定您在启动多部件上传请求中提供的相同MD5摘要。

如果源对象使用客户提供的密钥（ SSI-C ）进行加密，则必须在上传部件 - 复制请求中包含以下三个标题，以便可以解密并复制此对象：

- `x-amz-copy-source-server-side-encryption-customer-algorithm`：指定 `AES256`。
- `x-amz-copy-source-server-side-encryption-customer-key`：指定在创建源对象时提供的加密密钥。
- `x-amz-copy-source-server-side-encryption-customer-key-MD5`：指定在创建源对象时提供的MD5摘要。

> 您提供的加密密钥永远不会存储。如果丢失加密密钥，则会丢失相应的对象。在使用客户提供的密钥保护对象数据之前、请查看"`使用服务器端加密中的注意事项。`

版本控制

多部分上传包括启动上传，发布上传，上传部件，组装上传的部件以及完成上传的操作。在执行完整的多部件上传操作时，系统会创建对象（如果适用，还会进行版本控制）。

完成多部件上传

完整的多部件上传操作通过整合先前上传的部件来完成对象的多部分上传。

解决冲突

冲突的客户端请求(例如、两个客户端写入同一密钥)将按"`latest-WINS`"的原则进行解决。"`latest-WINS`"评估的时间取决于StorageGRID 系统何时完成给定请求、而不是S3客户端何时开始操作。

对象大小

StorageGRID 支持大小高达5 TB的对象。

请求标题

。 `x-amz-storage-class` 如果匹配的ILM规则指定了双重提交或平衡的载入行为、则支持请求标头、并影响StorageGRID 创建的对象副本数。

- `STANDARD`

  （默认）指定在 ILM 规则使用双提交选项或 balanced-option 回退到创建中间副本时执行双提交载入操作。

- `REDUCED_REDUNDANCY`

  指定在 ILM 规则使用双提交选项或 balanced-option 回退为创建中间副本时执行单提交载入操作。

> 如果要在启用了S3对象锁定的情况下将对象载入存储分段、则会显示 `REDUCED_REDUNDANCY` 选项将被忽略。如果要将对象载入旧的合规存储分段、则会显示 `REDUCED_REDUNDANCY` 选项返回错误。StorageGRID 将始终执行双提交载入，以确保满足合规性要求。

> 如果多部分上传未在 15 天内完成，则此操作将标记为非活动，并从系统中删除所有关联数据。

> 。 `ETag` 返回的值不是数据的MD5之和、而是遵循的Amazon S3 API实施 `ETag` 多部分对象的值。

版本控制

此操作将完成多部分上传。如果为存储分段启用了版本控制，则在完成多部分上传后会创建对象版本。

如果为存储分段启用了版本控制、则为唯一的 `versionId` 会自动为所存储对象的版本生成。这 `versionId` 也会使用在响应中返回 `x-amz-version-id` 响应标头。

如果版本控制已暂停、则存储对象版本时为空 `versionId` 如果已存在空版本、则该版本将被覆盖。

> 如果为存储分段启用了版本控制，则完成多部分上传始终会创建新版本，即使在同一对象密钥上同时完成多部分上传也是如此。如果某个存储分段未启用版本控制，则可以先启动多部分上传，然后再对同一对象密钥启动并完成另一个多部分上传。在非版本控制的存储分段上，最后完成的多部分上传将优先。

复制，通知或元数据通知失败

如果为平台服务配置了进行多部分上传的存储分段，则即使关联的复制或通知操作失败，多部分上传也会成功。

如果发生这种情况，则会在网格管理器中针对总事件（ SMT ）发出警报。最后一个事件消息显示 "`failed to publish notifications for bucket-nameobject key` " for the last object whose notification failed.(要查看此消息、请选择*节点*>*存储节点_*>*事件*。在表顶部查看上次事件。)事件消息也会在中列出 `/var/local/log/bycast-err.log`。

租户可以通过更新对象的元数据或标记来触发失败的复制或通知。租户可以重新提交现有值，以避免进行不必要的更改。

相关信息

"使用 ILM 管理对象"

错误响应

StorageGRID 系统支持所有适用的标准 S3 REST API 错误响应。此外， StorageGRID 实施还添加了多个自定义响应。

支持的 **S3 API** 错误代码

Name	**HTTP** 状态
ACCESSDENIED	403 已禁用
BadDigest	400 个错误请求
BucketAlreadyExists	409 冲突
BucketNotEmpagty	409 冲突
实体不完整	400 个错误请求

Name	HTTP 状态
内部错误	500 内部服务器错误
InvalidAccessKeyId	403 已禁用
InvalidArgument	400 个错误请求
InvalidBucketName	400 个错误请求
InvalidBucketState	409 冲突
InvalidDigest	400 个错误请求
InvalidEncryptionAlgorithmError	400 个错误请求
InvalidPart	400 个错误请求
InvalidPartOrder	400 个错误请求
InvalidRange	416 无法满足请求的范围
InvalidRequest	400 个错误请求
InvalidStorageClass	400 个错误请求
InvalidTag	400 个错误请求
InvalidURI	400 个错误请求
KeyTooLong	400 个错误请求
MalformedXML	400 个错误请求
MetadataTooLarge	400 个错误请求
方法未使用	不允许使用 405 方法
MissingContent长度	411 需要长度
MissingRequestBodyError	400 个错误请求
MissingSecurityHeader	400 个错误请求

Name	HTTP 状态
NoSuchBucket	未找到 404
NoSuchKey	未找到 404
NoSuchUpload	未找到 404
未实施	501 未实施
NoSuchBucketPolicy	未找到 404
ObjectLockConfigurationNotFoundError	未找到 404
预条件已启用	412- 前提条件失败
已请求超时	403 已禁用
服务不可用	503 服务不可用
SignatureDoesNotMatch	403 已禁用
TooMany桶	400 个错误请求
用户密钥已规范	400 个错误请求

StorageGRID 自定义错误代码

Name	Description	HTTP 状态
XBucketLifeycleNotAllowed	旧版合规存储分段不支持存储分段生命周期配置	400 个错误请求
XBucketPolicyParseException	无法解析收到的存储分段策略 JSON 。	400 个错误请求
XComplianceConflict	操作因原有合规性设置而被拒绝。	403 已禁用
XComplianceReducedRedundancyFor禁用	原有的合规存储分段不允许减少冗余	400 个错误请求
XMaxBucketPolicyLengthExceeded	您的策略超出了允许的最大存储分段策略长度。	400 个错误请求

Name	Description	HTTP 状态
XMissingInternalRequestHeader	缺少内部请求的标题。	400 个错误请求
XNoSuchBucketCompliance	指定的存储分段未启用原有合规性。	未找到 404
XNotAcceptable	此请求包含一个或多个无法满足的接受标头。	406 不可接受
未实施	您提供的请求意味着未实施的功能。	501 未实施

StorageGRID S3 REST API 操作

S3 REST API 中添加了特定于 StorageGRID 系统的操作。

获取存储分段一致性请求

使用获取存储分段一致性请求，您可以确定应用于特定存储分段的一致性级别。

默认一致性控制设置为保证新创建的对象的写入后读。

要完成此操作、您必须具有S3：GetBucketConsistency权限或帐户root。

请求示例

```
GET /bucket?x-ntap-sg-consistency HTTP/1.1
Date: <em>date</em>
Authorization: <em>authorization string</em>
Host: <em>host</em>
```

响应

在响应XML中、 `<Consistency>` 将返回以下值之一：

一致性控制	Description
全部	所有节点都会立即接收数据，否则请求将失败。
强大的全局功能	保证所有站点中所有客户端请求的写入后读一致性。
强大的站点	保证站点内所有客户端请求的写入后读一致性。

一致性控制	Description
读后写	（默认）为新对象提供写入后读一致性，并为对象更新提供最终一致性。提供高可用性和数据保护保证。与Amazon S3一致性保证匹配。 • 注意： * 如果应用程序对不存在的对象使用 head 请求，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。要防止出现这些错误，请将一致性控制设置为 "`Available` "，除非您需要类似于 Amazon S3 的一致性保证。
可用（机头操作的最终一致性）	与 `read-after-new-write` 一致性级别相同，但仅为机头操作提供最终一致性。如果存储节点不可用，则为机头操作提供的可用性比 " `read-after-new-write` " 更高。与 Amazon S3 一致性保证不同，仅适用于机头操作。

响应示例

```
HTTP/1.1 200 OK
Date: Fri, 18 Sep 2020 01:02:18 GMT
Connection: CLOSE
Server: StorageGRID/11.5.0
x-amz-request-id: 12345
Content-Length: 127
Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8"?>
<Consistency xmlns="http://s3.storagegrid.com/doc/2015-02-01/">read-after-
new-write</Consistency>
```

相关信息

"一致性控制"

PUT 存储分段一致性请求

通过 PUT 分段一致性请求，您可以指定要应用于对分段执行的操作的一致性级别。

默认一致性控制设置为保证新创建的对象的写入后读。

要完成此操作、您必须具有S3：PutBucketConsistency权限或帐户root。

请求

。 `x-ntap-sg-consistency` 参数必须包含以下值之一：

一致性控制	Description
全部	所有节点都会立即接收数据，否则请求将失败。
强大的全局功能	保证所有站点中所有客户端请求的写入后读一致性。
强大的站点	保证站点内所有客户端请求的写入后读一致性。
读后写	（默认）为新对象提供写入后读一致性，并为对象更新提供最终一致性。提供高可用性和数据保护保证。与Amazon S3一致性保证匹配。 • 注意： * 如果应用程序对不存在的对象使用 head 请求，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。要防止出现这些错误，请将一致性控制设置为 "`Available` "，除非您需要类似于 Amazon S3 的一致性保证。
可用（机头操作的最终一致性）	与 `read-after-new-write` 一致性级别相同，但仅为机头操作提供最终一致性。如果存储节点不可用，则为机头操作提供的可用性比 " `read-after-new-write` " 更高。与 Amazon S3 一致性保证不同，仅适用于机头操作。

• 注： * 通常，您应使用 `read-after-new-write` 一致性控制值。如果请求无法正常工作，请尽可能更改应用程序客户端的行为。或者，配置客户端以指定每个 API 请求的一致性控制。请仅作为最后一种方法在存储分段级别设置一致性控制。

请求示例

```
PUT /bucket?x-ntap-sg-consistency=strong-global HTTP/1.1
Date: <em>date</em>
Authorization: <em>authorization string</em>
Host: <em>host</em>
```

相关信息

"一致性控制"

获取分段上次访问时间请求

通过获取分段上次访问时间请求，您可以确定是为单个分段启用还是禁用了上次访问时间更新。

要完成此操作、您必须具有S3：GetBucketLastAccessTime权限或帐户root。

请求示例

```
GET /bucket?x-ntap-sg-lastaccesstime HTTP/1.1
Date: <em>date</em>
Authorization: <em>authorization string</em>
Host: <em>host</em>
```

响应示例

此示例显示已为存储分段启用上次访问时间更新。

```
HTTP/1.1 200 OK
Date: Sat, 29 Nov 2015 01:02:18 GMT
Connection: CLOSE
Server: StorageGRID/10.3.0
x-amz-request-id: 12345
Content-Length: 127
Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8"?>
<LastAccessTime xmlns="http://s3.storagegrid.com/doc/2015-02-01/">enabled
</LastAccessTime>
```

PUT 分段上次访问时间请求

通过 PUT 分段上次访问时间请求，您可以为各个分段启用或禁用上次访问时间更新。禁用上次访问时间更新可提高性能，它是使用 10.3.0 或更高版本创建的所有存储分段的默认设置。

要完成此操作、您必须对某个存储分段拥有S3：PutBucketLastAccessTime权限、或者以root帐户身份登录。

> 从 StorageGRID 10.3 版开始，默认情况下，所有新存储分段都会禁用对上次访问时间的更新。如果您的存储分段是使用早期版本的 StorageGRID 创建的，并且您希望与新的默认行为匹配，则必须明确禁用上述每个存储分段的上次访问时间更新。您可以使用租户管理器中的 PUT 分段上次访问时间请求， * S3 > * 分段 * > * 更改上次访问设置 * 复选框或租户管理 API 启用或禁用对最后访问时间的更新。

如果禁用了某个存储分段的上次访问时间更新，则会对存储分段上的操作应用以下行为：

- GET 对象， GET 对象 ACL ， GET 对象标记和 HEAD 对象请求不会更新上次访问时间。此对象不会添加到用于信息生命周期管理（ ILM ）评估的队列中。
- PUT 对象—仅更新元数据的复制和 PUT 对象标记请求也会更新上次访问时间。对象将添加到队列中以进行 ILM 评估。
- 如果对源存储分段禁用了对最后访问时间的更新，则 PUT Object - Copy Requests 不会更新源存储分段的最后访问时间。复制的对象不会添加到源存储分段的 ILM 评估队列中。但是，对于目标， PUT 对象 - 复制请求始终更新上次访问时间。对象副本将添加到队列中以进行 ILM 评估。
- 完成多部件上传请求更新上次访问时间。已完成的对象将添加到队列中以进行 ILM 评估。

请求示例

此示例将为存储分段启用上次访问时间。

```
PUT /bucket?x-ntap-sg-lastaccesstime=enabled HTTP/1.1
Date: <em>date</em>
Authorization: <em>authorization string</em>
Host: <em>host</em>
```

此示例将禁用存储分段的上次访问时间。

```
PUT /bucket?x-ntap-sg-lastaccesstime=disabled HTTP/1.1
Date: <em>date</em>
Authorization: <em>authorization string</em>
Host: <em>host</em>
```

相关信息

"使用租户帐户"

删除存储分段元数据通知配置请求

通过删除存储分段元数据通知配置请求，您可以通过删除配置 XML 来禁用各个存储分段的搜索集成服务。

要完成此操作、您必须对某个存储分段拥有S3：DeleteBucketMetadataNotification权限、或者以root帐户身份登录。

请求示例

此示例显示了禁用存储分段的搜索集成服务。

```
DELETE /test1?x-ntap-sg-metadata-notification HTTP/1.1
Date: <em>date</em>
Authorization: <em>authorization string</em>
Host: <em>host</em>
```

获取存储分段元数据通知配置请求

使用获取分段元数据通知配置请求，您可以检索用于为各个分段配置搜索集成的配置 XML 。

要完成此操作、您必须具有S3：GetBucketMetadataNotification权限或帐户root。

请求示例

此请求将检索名为的存储分段的元数据通知配置 `bucket`。

```
GET /bucket?x-ntap-sg-metadata-notification HTTP/1.1
Date: <em>date</em>
Authorization: <em>authorization string</em>
Host: <em>host</em>
```

响应

响应正文包括存储分段的元数据通知配置。通过元数据通知配置，您可以确定如何配置存储分段以进行搜索集成。也就是说，您可以通过它确定哪些对象已编制索引，以及将其对象元数据发送到哪些端点。

```
<MetadataNotificationConfiguration>
    <Rule>
        <ID>Rule-1</ID>
        <Status>rule-status</Status>
        <Prefix>key-prefix</Prefix>
        <Destination>
           <Urn>arn:aws:es:_region:account-
ID_:domain/_mydomain/myindex/mytype_</Urn>
        </Destination>
    </Rule>
    <Rule>
        <ID>Rule-2</ID>
         ...
    </Rule>
     ...
</MetadataNotificationConfiguration>
```

每个元数据通知配置都包含一个或多个规则。每个规则都指定其适用场景 的对象以及 StorageGRID 应将对象元数据发送到的目标。必须使用 StorageGRID 端点的 URN 指定目标。

Name	Description	Required
MetadataNotificationConfiguration	用于指定元数据通知的对象和目标的规则的容器标记。 包含一个或多个规则元素。	是的。
规则	用于标识应将其元数据添加到指定索引中的对象的规则的容器标记。 拒绝前缀重叠的规则。 包含在 MetadataNotificationConfiguration 元素中。	是的。

Name	Description	Required
ID	规则的唯一标识符。 包含在 Rule 元素中。	否
Status	状态可以是 " 已启用 " 或 " 已禁用 " 。不会对已禁用的规则执行任何操作。 包含在 Rule 元素中。	是的。
前缀	与前缀匹配的对象受此规则的影响，其元数据将发送到指定目标。 要匹配所有对象，请指定一个空前缀。 包含在 Rule 元素中。	是的。
目标	规则目标的容器标记。 包含在 Rule 元素中。	是的。
URN	发送对象元数据的目标的 urn 。必须是具有以下属性的 StorageGRID 端点的 URN ： • `es` 必须是第三个元素。 • URN必须以存储元数据的索引和类型结尾、格式为 `domain-name/myindex/mytype`。 端点使用租户管理器或租户管理 API 进行配置。它们的形式如下： • `arn:aws:es:_region:account-ID_:domain/mydomain/myindex/mytype` • `urn:mysite:es:::mydomain/myindex/mytype` 必须在提交配置 XML 之前配置端点，否则配置将失败并显示 404 错误。 urn 包含在目标元素中。	是的。

响应示例

包含在之间的XML
`<MetadataNotificationConfiguration></MetadataNotificationConfiguration>` 标记显示了如何为存储分段配置与搜索集成端点的集成。在此示例中、对象元数据将发送到名为的Elasticsearch索引 `current` 并键入named `2017` 托管在名为的AWS域中 `records`。

```
HTTP/1.1 200 OK
Date: Thu, 20 Jul 2017 18:24:05 GMT
Connection: KEEP-ALIVE
Server: StorageGRID/11.0.0
x-amz-request-id: 3832973499
Content-Length: 264
Content-Type: application/xml

<MetadataNotificationConfiguration>
    <Rule>
        <ID>Rule-1</ID>
        <Status>Enabled</Status>
        <Prefix>2017</Prefix>
        <Destination>
           <Urn>arn:aws:es:us-east-
1:3333333:domain/records/current/2017</Urn>
        </Destination>
    </Rule>
</MetadataNotificationConfiguration>
```

相关信息

"使用租户帐户"

PUT 存储分段元数据通知配置请求

通过 PUT Bucket 元数据通知配置请求，您可以为各个存储分段启用搜索集成服务。您在请求正文中提供的元数据通知配置 XML 用于指定将其元数据发送到目标搜索索引的对象。

要完成此操作、您必须对某个存储分段拥有S3：PutBucketMetadataNotification权限、或者以root帐户身份登录。

请求

此请求必须在请求正文中包含元数据通知配置。每个元数据通知配置都包含一个或多个规则。每个规则都指定其适用场景 的对象以及 StorageGRID 应将对象元数据发送到的目标。

可以按对象名称的前缀筛选对象。例如、您可以发送具有前缀的对象的元数据 `/images` 到一个目标、以及具有前缀的对象 `/videos` 另一个。

前缀重叠的配置无效，在提交时会被拒绝。例如、一种配置、其中包含一个规则、用于具有前缀的对象 `test` 和第二个规则、用于具有前缀的对象 `test2` 不允许。

必须使用 StorageGRID 端点的 URN 指定目标。如果提交元数据通知配置、或者请求以失败的形式出现故障、则端点必须存在 `400 Bad Request`。错误消息显示： `Unable to save the metadata notification (search) policy. The specified endpoint URN does not exist: URN.`

```
<MetadataNotificationConfiguration>
    <Rule>
        <ID>Rule-1</ID>
        <Status>rule-status</Status>
        <Prefix>key-prefix</Prefix>
        <Destination>
           <Urn>arn:aws:es:region:account-
ID:domain/mydomain/myindex/mytype</Urn>
        </Destination>
    </Rule>
    <Rule>
        <ID>Rule-2</ID>
         ...
    </Rule>
     ...
</MetadataNotificationConfiguration>
```

下表介绍了元数据通知配置 XML 中的元素。

Name	Description	Required
MetadataNotificationConfiguration	用于指定元数据通知的对象和目标的规则的容器标记。 包含一个或多个规则元素。	是的。
规则	用于标识应将其元数据添加到指定索引中的对象的规则的容器标记。 拒绝前缀重叠的规则。 包含在 MetadataNotificationConfiguration 元素中。	是的。
ID	规则的唯一标识符。 包含在 Rule 元素中。	否

Name	Description	Required
Status	状态可以是 " 已启用 " 或 " 已禁用 " 。不会对已禁用的规则执行任何操作。 包含在 Rule 元素中。	是的。
前缀	与前缀匹配的对象受此规则的影响，其元数据将发送到指定目标。 要匹配所有对象，请指定一个空前缀。 包含在 Rule 元素中。	是的。
目标	规则目标的容器标记。 包含在 Rule 元素中。	是的。
URN	发送对象元数据的目标的 urn 。必须是具有以下属性的 StorageGRID 端点的 URN ： • `es` 必须是第三个元素。 • URN必须以存储元数据的索引和类型结尾、格式为 `domain-name/myindex/mytype`。 端点使用租户管理器或租户管理 API 进行配置。它们的形式如下： • `arn:aws:es:region:account-ID:domain/mydomain/myindex/mytype` • `urn:mysite:es:::mydomain/myindex/mytype` 必须在提交配置 XML 之前配置端点，否则配置将失败并显示 404 错误。 urn 包含在目标元素中。	是的。

请求示例

此示例显示了为存储分段启用搜索集成。在此示例中，所有对象的对象元数据都将发送到同一目标。

```
PUT /test1?x-ntap-sg-metadata-notification HTTP/1.1
Date: <em>date</em>
Authorization: <em>authorization string</em>
Host: <em>host</em>

<MetadataNotificationConfiguration>
    <Rule>
        <ID>Rule-1</ID>
        <Status>Enabled</Status>
        <Prefix></Prefix>
        <Destination>
           <Urn>urn:sgws:es:::sgws-notifications/test1/all</Urn>
        </Destination>
    </Rule>
</MetadataNotificationConfiguration>
```

在此示例中、是指与前缀匹配的对象的对象元数据 `/images` 发送到一个目标、而与前缀匹配的对象的对象元数据则发送到一个目标 `/videos` 发送到另一个目标。

```
PUT /graphics?x-ntap-sg-metadata-notification HTTP/1.1
Date: <em>date</em>
Authorization: <em>authorization string</em>
Host: <em>host</em>

<MetadataNotificationConfiguration>
    <Rule>
        <ID>Images-rule</ID>
        <Status>Enabled</Status>
        <Prefix>/images</Prefix>
        <Destination>
           <Urn>arn:aws:es:us-east-1:3333333:domain/es-
domain/graphics/imagetype</Urn>
        </Destination>
    </Rule>
    <Rule>
        <ID>Videos-rule</ID>
        <Status>Enabled</Status>
        <Prefix>/videos</Prefix>
        <Destination>
           <Urn>arn:aws:es:us-west-1:22222222:domain/es-
domain/graphics/videotype</Urn>
        </Destination>
    </Rule>
</MetadataNotificationConfiguration>
```

相关信息

"使用租户帐户"

由搜索集成服务生成的 **JSON**

为存储分段启用搜索集成服务后，每次添加，更新或删除对象元数据或标记时，系统都会生成一个 JSON 文档并将其发送到目标端点。

此示例显示了使用密钥的对象时可能生成的JSON示例 `SGWS/Tagging.txt` 在名为的存储分段中创建 `test`。。 `test` 存储分段未进行版本控制、因此 `versionId` 标记为空。

```
{
  "bucket": "test",
  "key": "SGWS/Tagging.txt",
  "versionId": "",
  "accountId": "86928401983529626822",
  "size": 38,
  "md5": "3d6c7634a85436eee06d43415012855",
  "region":"us-east-1"
  "metadata": {
    "age": "25"
  },
  "tags": {
    "color": "yellow"
  }
}
```

元数据通知中包含的对象元数据

下表列出了启用搜索集成后发送到目标端点的 JSON 文档中包含的所有字段。

文档名称包括存储分段名称，对象名称和版本 ID （如果存在）。

Type	项目名称	Description
存储分段和对象信息	存储分段	存储分段的名称
存储分段和对象信息	key	对象密钥名称
存储分段和对象信息	版本 ID	对象版本，用于受版本控制的分段中的对象
存储分段和对象信息	region	分段区域、例如 `us-east-1`
系统元数据	size	HTTP 客户端可见的对象大小（以字节为单位）

Type	项目名称	Description
系统元数据	MD5	对象哈希
用户元数据	元数据 *key:value*	对象的所有用户元数据，作为键值对
Tags	tags *key:value*	为对象定义的所有对象标记，作为键值对

- 注： * 对于标记和用户元数据， StorageGRID 会将日期和数字作为字符串或 S3 事件通知传递给 Elasticsearch 。要配置 Elasticsearch 以将这些字符串解释为日期或数字，请按照 Elasticsearch 说明进行动态字段映射和映射日期格式。在配置搜索集成服务之前，必须在索引上启用动态字段映射。为文档编制索引后，您无法在索引中编辑文档的字段类型。

获取存储使用情况请求

" 获取存储使用量 " 请求会告知您帐户正在使用的存储总量以及与帐户关联的每个存储分段的存储总量。

帐户及其存储分段使用的存储量可通过修改后的GET服务请求获得 `x-ntap-sg-usage` 查询参数。存储分段使用量与系统处理的 PUT 和 DELETE 请求分开跟踪。根据请求处理情况，使用量值与预期值匹配可能会有一定的延迟，尤其是在系统负载较重时。

默认情况下， StorageGRID 会尝试使用强全局一致性检索使用情况信息。如果无法实现强全局一致性， StorageGRID 将尝试以强站点一致性检索使用情况信息。

要完成此操作、您必须具有S3：ListAllMy桶 权限或帐户root。

请求示例

```
GET /?x-ntap-sg-usage HTTP/1.1
Date: <em>date</em>
Authorization: <em>authorization string</em>
Host: <em>host</em>
```

响应示例

此示例显示了一个帐户，该帐户在两个存储分段中包含四个对象和 12 字节的数据。每个存储分段包含两个对象和六个字节的数据。

```
HTTP/1.1 200 OK
Date: Sat, 29 Nov 2015 00:49:05 GMT
Connection: KEEP-ALIVE
Server: StorageGRID/10.2.0
x-amz-request-id: 727237123
Content-Length: 427
Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8"?>
<UsageResult xmlns="http://s3.storagegrid.com/doc/2015-02-01">
<CalculationTime>2014-11-19T05:30:11.000000Z</CalculationTime>
<ObjectCount>4</ObjectCount>
<DataBytes>12</DataBytes>
<Buckets>
<Bucket>
<Name>bucket1</Name>
<ObjectCount>2</ObjectCount>
<DataBytes>6</DataBytes>
</Bucket>
<Bucket>
<Name>bucket2</Name>
<ObjectCount>2</ObjectCount>
<DataBytes>6</DataBytes>
</Bucket>
</Buckets>
</UsageResult>
```

版本控制

存储的每个对象版本都将用于 `ObjectCount` 和 `DataBytes` 响应中的值。删除标记不会添加到 `ObjectCount` 总计。

相关信息

"一致性控制"

已弃用旧合规性存储分段请求

您可能需要使用 StorageGRID S3 REST API 来管理使用原有合规性功能创建的分段。

已弃用合规性功能

先前 StorageGRID 版本中提供的 StorageGRID 合规性功能已弃用，并已被 S3 对象锁定取代。

如果先前启用了全局合规性设置、则在升级到StorageGRID 11.5时、系统会自动启用全局S3对象锁定设置。您不能再在启用了合规性的情况下创建新的存储分段；但是，您可以根据需要使用 StorageGRID S3 REST API 管理任何现有的旧合规存储分段。

"使用 S3 对象锁定"

"使用 ILM 管理对象"

"NetApp 知识库：如何在 StorageGRID 11.5 中管理原有的合规存储分段"

已弃用：为满足合规性而修改存储分段请求

SGCompliance XML 元素已弃用。以前，您可以将此 StorageGRID 自定义元素包含在 PUT 存储分段请求的可选 XML 请求正文中，以创建合规存储分段。

先前 StorageGRID 版本中提供的 StorageGRID 合规性功能已弃用，并已被 S3 对象锁定取代。

"使用 S3 对象锁定"

"使用 ILM 管理对象"

"NetApp 知识库：如何在 StorageGRID 11.5 中管理原有的合规存储分段"

您不能再在已启用合规性的情况下创建新存储分段。如果您尝试使用 PUT 分段请求修改以满足合规性要求来创建新的合规分段，则会返回以下错误消息：

```
The Compliance feature is deprecated.
Contact your StorageGRID administrator if you need to create new Compliant
buckets.
```

相关信息

"使用 ILM 管理对象"

"使用租户帐户"

已弃用：获取存储分段合规性请求

获取存储分段合规性请求已弃用。但是，您可以继续使用此请求来确定当前对现有旧版合规存储分段有效的合规性设置。

先前 StorageGRID 版本中提供的 StorageGRID 合规性功能已弃用，并已被 S3 对象锁定取代。

"使用 S3 对象锁定"

"使用 ILM 管理对象"

"NetApp 知识库：如何在 StorageGRID 11.5 中管理原有的合规存储分段"

要完成此操作、您必须具有S3：GetBucketCompliance权限或帐户root。

请求示例

通过此示例请求、您可以确定名为的存储分段的合规性设置 `mybucket`。

```
GET /mybucket/?x-ntap-sg-compliance HTTP/1.1
Date: <em>date</em>
Authorization: <em>authorization string</em>
Host: <em>host</em>
```

响应示例

在响应XML中、 `<SGCompliance>` 列出了对存储分段有效的合规性设置。此示例响应显示了一个存储分段的合规性设置，从将对象载入网格开始，每个对象将保留一年（ 525600 分钟）。此存储分段当前没有法律上的保留。每个对象将在一年后自动删除。

```
HTTP/1.1 200 OK
Date: <em>date</em>
Connection: <em>connection</em>
Server: StorageGRID/11.1.0
x-amz-request-id: <em>request ID</em>
Content-Length: <em>length</em>
Content-Type: application/xml

<SGCompliance>
  <RetentionPeriodMinutes>525600</RetentionPeriodMinutes>
  <LegalHold>false</LegalHold>
  <AutoDelete>true</AutoDelete>
</SGCompliance>
```

Name	Description
RetentionPeriodMinutes	添加到此存储分段的对象的保留期限长度，以分钟为单位。保留期限从将对象载入网格时开始。
乐高积木	• true ： 此存储分段当前处于合法保留状态。除非取消合法保留，否则无法删除此存储分段中的对象，即使其保留期限已过期也是如此。 • false ： 此存储分段当前未处于合法保留状态。此存储分段中的对象可以在保留期限到期时删除。
自动删除	• true ： 此存储分段中的对象将在保留期限到期时自动删除，除非此存储分段处于合法保留状态。 • false ： 保留期限到期后，不会自动删除此存储分段中的对象。如果需要删除这些对象，必须手动将其删除。

错误响应

如果未创建符合要求的存储分段、则响应的HTTP状态代码为 `404 Not Found`、带有S3错误代码 `XNoSuchBucketCompliance`。

相关信息

"使用 ILM 管理对象"

"使用租户帐户"

已弃用： **PUT** 存储分段合规性请求

PUT 存储分段合规性请求已弃用。但是，您可以继续使用此请求修改现有旧版合规存储分段的合规性设置。例如，您可以将现有存储分段置于合法保留状态或延长其保留期限。

先前 StorageGRID 版本中提供的 StorageGRID 合规性功能已弃用，并已被 S3 对象锁定取代。

"使用 S3 对象锁定"

"使用 ILM 管理对象"

"NetApp 知识库：如何在 StorageGRID 11.5 中管理原有的合规存储分段"

要完成此操作、您必须具有S3：PutBucketCompliance权限或帐户root。

发出 PUT 存储分段合规性请求时，必须为合规性设置的每个字段指定一个值。

请求示例

此示例请求修改名为的存储分段的合规性设置 `mybucket`。在此示例中、对象位于中 `mybucket` 现在将保留两年(1、051、200分钟)、而不是一年、从将对象载入网格开始。此存储分段没有法律上的保留。每个对象将在两年后自动删除。

```
PUT /mybucket/?x-ntap-sg-compliance HTTP/1.1
Date: <em>date</em>
Authorization: <em>authorization name</em>
Host: <em>host</em>
Content-Length: 152

<SGCompliance>
  <RetentionPeriodMinutes>1051200</RetentionPeriodMinutes>
  <LegalHold>false</LegalHold>
  <AutoDelete>true</AutoDelete>
</SGCompliance>
```

Name	Description
RetentionPeriodMinutes	添加到此存储分段的对象的保留期限长度，以分钟为单位。保留期限从将对象载入网格时开始。 • 注意： * 为 RetentionPeriodMinutes 指定新值时，必须指定一个等于或大于存储分段当前保留期限的值。设置存储分段的保留期限后，您不能减小该值；您只能增加该值。
乐高积木	• true ： 此存储分段当前处于合法保留状态。除非取消合法保留，否则无法删除此存储分段中的对象，即使其保留期限已过期也是如此。 • false ： 此存储分段当前未处于合法保留状态。此存储分段中的对象可以在保留期限到期时删除。
自动删除	• true ： 此存储分段中的对象将在保留期限到期时自动删除，除非此存储分段处于合法保留状态。 • false ： 保留期限到期后，不会自动删除此存储分段中的对象。如果需要删除这些对象，必须手动将其删除。

合规性设置的一致性级别

当您使用 PUT 存储分段合规性请求更新 S3 存储分段的合规性设置时， StorageGRID 会尝试更新整个网格中存储分段的元数据。默认情况下， StorageGRID 使用 * 强 - 全局 * 一致性级别来保证所有数据中心站点以及包含存储分段元数据的所有存储节点在更改的合规性设置中具有读 - 写一致性。

如果由于某个站点上的数据中心站点或多个存储节点不可用而导致StorageGRID 无法达到*强-全局*一致性级别、则响应的HTTP状态代码为 `503 Service Unavailable.`

如果收到此响应，您必须联系网格管理员，以确保所需的存储服务尽快可用。如果网格管理员无法在每个站点提供足够的存储节点，技术支持可能会指示您通过强制执行 * 强站点 * 一致性级别来重试失败的请求。

除非技术支持指示您这样做，并且您了解使用此级别可能产生的后果，否则切勿强制使用 * 强站点 * 一致性级别来满足 PUT 存储分段合规性要求。

当一致性级别降低到 * 强站点 * 时， StorageGRID 保证更新后的合规性设置仅对站点中的客户端请求具有读写后一致性。这意味着，在所有站点和存储节点均可用之前， StorageGRID 系统可能会暂时为此存储分段设置多个不一致的设置。设置不一致可能导致意外和意外的行为。例如，如果您将某个存储分段置于合法保留状态并强制降低一致性级别，则某些数据中心站点上可能仍会继续使用存储分段先前的合规性设置（即合法保留）。因此，您认为处于合法保留状态的对象可能会在保留期限到期时被用户删除，或者如果启用了自动删除，也可以删除。

要强制使用*强站点*一致性级别、请重新发出PUT存储分段合规性请求并加入 `Consistency-Control` HTTP请求标头、如下所示：

```
PUT /mybucket/?x-ntap-sg-compliance HTTP/1.1
Consistency-Control: strong-site
```

错误响应

- 如果未创建符合要求的存储分段、则响应的HTTP状态代码为 `404 Not Found`。
- 条件 `RetentionPeriodMinutes` 在请求小于存储分段的当前保留期限时、HTTP状态代码为 `400 Bad Request`。

相关信息

"已弃用：为满足合规性而修改存储分段请求"

"使用租户帐户"

"使用 ILM 管理对象"

存储分段和组访问策略

StorageGRID 使用 Amazon Web Services （ AWS ）策略语言允许 S3 租户控制对这些存储分段和对象的访问。StorageGRID 系统实施 S3 REST API 策略语言的一个子集。S3 API 的访问策略以 JSON 格式写入。

访问策略概述

StorageGRID 支持两种访问策略。

- * 分段策略 * ，使用 GET 分段策略， PUT 分段策略和 DELETE 分段策略 S3 API 操作进行配置。存储分段策略附加到存储分段，因此，可以对其进行配置，以控制存储分段所有者帐户或其他帐户中的用户对存储分段及其对象的访问。一个存储分段策略适用场景 只能包含一个存储分段，并且可能包含多个组。
- * 组策略 * ，使用租户管理器或租户管理 API 配置。组策略会附加到帐户中的某个组，因此，这些策略会配置为允许该组访问该帐户拥有的特定资源。一个组策略只对一个组进行适用场景 ，并且可能对多个存储分段进行。

StorageGRID 存储分段和组策略遵循由 Amazon 定义的特定语法。每个策略中都包含一组策略语句，每个语句都包含以下元素：

- 语句 ID （ SID ）（可选）
- 影响
- 主体 / 不重要
- 资源 /NotResource
- 操作 / 未操作
- 条件（可选）

策略语句是使用此结构构建的，用于指定权限： Grant <Effic> to allow/deny <Principe> to Perform <Action> on <Resource> when <condition> applies 。

每个策略元素都用于特定功能：

Element	Description
SID	Sid 元素是可选的。SID 仅用作用户的问题描述 。它会被存储，但不会被 StorageGRID 系统解释。
影响	使用 Effect 元素确定是否允许或拒绝指定的操作。您必须使用支持的 Action Element 关键字来确定允许（或拒绝）对存储分段或对象执行的操作。
主体 / 不重要	您可以允许用户，组和帐户访问特定资源并执行特定操作。如果请求中不包含 S3 签名，则可以通过指定通配符（ * ）作为主体来进行匿名访问。默认情况下，只有帐户 root 有权访问该帐户拥有的资源。 您只需要在存储分段策略中指定主体元素。对于组策略，附加该策略的组为隐式主体元素。
资源 /NotResource	资源元素用于标识分段和对象。您可以使用 Amazon 资源名称（ ARN ）来标识资源，从而允许或拒绝对存储分段和对象的权限。
操作 / 未操作	操作和效果元素是权限的两个组成部分。当组请求资源时，它们会被授予或拒绝访问该资源。除非您明确分配权限，否则访问将被拒绝，但您可以使用显式拒绝覆盖由其他策略授予的权限。
条件	条件元素是可选的。通过条件，您可以构建表达式以确定何时应用策略。

在 Action 元素中，您可以使用通配符（ * ）指定所有操作或部分操作。例如，此操作与 S3： GetObject ， S3 ： PutObject 和 S3： DeleteObject 等权限匹配。

```
s3:*Object
```

在资源元素中，可以使用通配符（ * ）和（？ ）。星号（ * ）与 0 个或更多字符匹配时，问号（？ ） 匹配任意单个字符。

在 Principal 元素中，除了设置匿名访问权限之外，不支持使用通配符，此权限会授予所有人权限。例如，您将通配符（ * ）设置为 Principal 值。

```
"Principal":"*"
```

在以下示例中，该语句使用的是 " 影响 " ， " 主体 " ， " 操作 " 和 " 资源 " 元素。此示例显示了一个完整的存储分段策略语句、该语句使用"allow"的效果为Principals即管理组 `federated-group/admin` 和财务团队 `federated-group/finance`、执行操作的权限 `s3:ListBucket` 位于名为的存储分段上 `mybucket` 和操作

`s3:GetObject` 存储在该存储分段内的所有对象上。

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::27233906934684427525:federated-group/admin",
          "arn:aws:iam::27233906934684427525:federated-group/finance"
        ]
      },
      "Action": [
        "s3:ListBucket",
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:iam:s3:::mybucket",
        "arn:aws:iam:s3:::mybucket/*"
      ]
    }
  ]
}
```

存储分段策略的大小限制为 20 ， 480 字节，而组策略的大小限制为 5 ， 120 字节。

相关信息

"使用租户帐户"

策略的一致性控制设置

默认情况下，对组策略所做的任何更新最终都是一致的。由于策略缓存，一旦组策略保持一致，所做的更改可能还需要 15 分钟才能生效。默认情况下，对存储分段策略进行的任何更新最终也会保持一致。

您可以根据需要更改存储分段策略更新的一致性保证。例如，出于安全原因，您可能希望对存储分段策略所做的更改尽快生效。

在这种情况下、您可以设置 `Consistency-Control` 标题、或者您也可以使用PUT存储分段一致性请求。更改此请求的一致性控制时，必须使用值 * 全部 * ，这可以为读写一致性提供最高保证。如果在 PUT 存储分段一致性请求的标题中指定任何其他一致性控制值，则此请求将被拒绝。如果为 PUT 存储分段策略请求指定任何其他值，则此值将被忽略。存储分段策略保持一致后，由于策略缓存，更改可能需要额外 8 秒才能生效。

如果将一致性级别设置为 "**all** " 以强制新的存储分段策略更快生效，请确保在完成后将存储分段级别控制设置回其原始值。否则，所有未来的存储分段请求将使用 * 全部 * 设置。

在策略语句中使用ARN

在策略语句中， ARN 用于 Principal 和 Resource Element 。

- 使用以下语法指定 S3 资源 ARN ：

```
arn:aws:s3:::bucket-name
arn:aws:s3:::bucket-name/object_key
```

- 使用以下语法指定身份资源 ARN （用户和组）：

```
arn:aws:iam::account_id:root
arn:aws:iam::account_id:user/user_name
arn:aws:iam::account_id:group/group_name
arn:aws:iam::account_id:federated-user/user_name
arn:aws:iam::account_id:federated-group/group_name
```

其他注意事项：

- 您可以使用星号（ * ）作为通配符，以匹配对象密钥中的零个或多个字符。
- 可以在对象密钥中指定的国际字符应使用 JSON UTF-8 或 JSON \u 转义序列进行编码。不支持百分比编码。

  "RFC 2141 URN 语法"

  PUT 存储分段策略操作的 HTTP 请求正文必须使用 charset=UTF-8 进行编码。

在策略中指定资源

在策略语句中，您可以使用资源元素指定允许或拒绝权限的分段或对象。

- 每个策略语句都需要一个资源元素。在策略中、资源由元素表示 `Resource`或者、 `NotResource` 以排除。
- 您可以使用 S3 资源 ARN 指定资源。例如：

```
"Resource": "arn:aws:s3:::mybucket/*"
```

- 您也可以在对象密钥中使用策略变量。例如：

```
"Resource": "arn:aws:s3:::mybucket/home/${aws:username}/*"
```

- 资源值可以指定创建组策略时尚不存在的存储分段。

相关信息

"在策略中指定变量"

在策略中指定主体

使用 Principal 元素标识策略语句允许 / 拒绝访问资源的用户，组或租户帐户。

- 存储分段策略中的每个策略语句都必须包含一个主体元素。组策略中的策略语句不需要主体元素，因为组被理解为主体。
- 在策略中，主体由元素 "`Principal ，` " 或 "`NotPrincipal` " 表示以表示排除。
- 必须使用 ID 或 ARN 指定基于帐户的身份：

```
"Principal": { "AWS": "account_id"}
"Principal": { "AWS": "identity_arn" }
```

- 此示例使用租户帐户 ID 27233906934684427525 ，其中包括帐户 root 和帐户中的所有用户：

```
 "Principal": { "AWS": "27233906934684427525" }
```

- 您只能指定帐户 root ：

```
"Principal": { "AWS": "arn:aws:iam::27233906934684427525:root" }
```

- 您可以指定一个特定的联合用户（ "Alex" ）：

```
"Principal": { "AWS": "arn:aws:iam::27233906934684427525:federated-
user/Alex" }
```

- 您可以指定特定的联合组（ "Managers" ）：

```
"Principal": { "AWS": "arn:aws:iam::27233906934684427525:federated-
group/Managers"  }
```

- 您可以指定匿名主体：

```
"Principal": "*"
```

- 为避免歧义，您可以使用用户 UUID ，而不是用户名：

```
arn:aws:iam::27233906934684427525:user-uuid/de305d54-75b4-431b-adb2-
eb6b9e546013
```

例如、假设Alex离开了组织和用户名 `Alex` 已删除。如果新的Alex加入了该组织并获得了相同的分配 `Alex` 用户名、新用户可能会意外继承授予原始用户的权限。

- 主体值可以指定在创建存储分段策略时尚不存在的组 / 用户名称。

在策略中指定权限

在策略中， Action 元素用于允许 / 拒绝对资源的权限。您可以在策略中指定一组权限，这些权限由元素 "Action" 或 "NotAction" 表示以表示排除。其中每个元素都映射到特定的 S3 REST API 操作。

下表列出了应用于存储分段的权限以及应用于对象的权限。

Amazon S3 现在对 PUT 和 DELETE 分段复制操作使用 S3： PutReplicationConfiguration 权限。StorageGRID 对每个操作使用单独的权限，这些权限与原始 Amazon S3 规范匹配。

如果使用 PUT 覆盖现有值，则会执行删除。

应用于存储分段的权限

权限	**S3 REST API** 操作	为 **StorageGRID** 自定义
S3： CreateBucket	放入存储分段	
S3： DeleteBucket	删除存储分段	
S3： DeleteBucketMetadataNotification	删除存储分段元数据通知配置	是的。
S3： DeleteBucketPolicy	删除存储分段策略	
S3： DeleteReplicationConfiguration	删除存储分段复制	是， PUT 和 DELETE 的权限不同 *
S3： GetBucketAcl	获取分段 ACL	
S3： GetBucketCompliance	获取存储分段合规性（已弃用）	是的。
S3： GetBucketConsistency	获取存储分段一致性	是的。
S3： GetBucketCORS	获取分段存储器	

权限	**S3 REST API** 操作	为 **StorageGRID** 自定义
S3： GetEncryptionConfiguration	获取存储分段加密	
S3： GetBucketLastAccessTime	获取存储分段上次访问时间	是的。
S3： GetBucketLocation	获取存储分段位置	
S3： GetBucketMetadataNotification	获取存储分段元数据通知配置	是的。
S3： GetBucketNotification	获取存储分段通知	
S3： GetBucketObjectLockConfiguration	获取对象锁定配置	
S3： GetBucketPolicy	获取存储分段策略	
S3： GetBucketTagging	获取存储分段标记	
S3： GetBucketVersioning	获取存储分段版本控制	
S3： GetLifeycleConfiguration	获取存储分段生命周期	
S3： GetReplicationConfiguration	获取存储分段复制	
S3： ListAllMy桶	• 获取服务 • 获取存储使用量	是，适用于获取存储使用量
S3： ListBucket	• 获取存储分段（列出对象） • 头存储分段 • 后对象还原	
S3： ListBucketMultipartUploads	• 列出多部件上传 • 后对象还原	
S3： ListBucketVersions	获取存储分段版本	
S3： PutBucketCompliance	PUT 存储分段合规性（已弃用）	是的。
S3： PutBucketConsistency	PUT 存储分段一致性	是的。

权限	**S3 REST API** 操作	为 **StorageGRID** 自定义
S3： PutBucketCORS	• 删除存储分段或† • 放入存储分段箱	
S3： PutEncryptionConfiguration	• 删除存储分段加密 • PUT 存储分段加密	
S3： PutBucketLastAccessTime	PUT 分段上次访问时间	是的。
S3： PutBucketMetadataNotification	PUT 存储分段元数据通知配置	是的。
S3： PutBucketNotification	PUT 存储分段通知	
S3： PutBucketObjectLockConfiguration	使用PUT存储分段 `x-amz-bucket-object-lock-enabled: true` 请求标头(也需要S3：CreateBucket权限)	
S3： PutBucketPolicy	PUT 存储分段策略	
S3： PutBucketTagging	• 删除存储分段标记† • 放置存储分段标记	
S3： PutBucketVersioning	PUT 存储分版本	
S3： PutLifeycleConfiguration	• 删除存储分段生命周期† • PUT 存储分段生命周期	
S3： PutReplicationConfiguration	PUT 存储分段复制	是， PUT 和 DELETE 的权限不同 *

应用于对象的权限

权限	**S3 REST API** 操作	为 **StorageGRID** 自定义
S3： AbortMultipartUpload	• 中止多部分上传 • 后对象还原	
S3： DeleteObject	• 删除对象 • 删除多个对象 • 后对象还原	

权限	S3 REST API 操作	为 StorageGRID 自定义
S3： DeleteObjectTagging	删除对象标记	
S3： DeleteObjectVersionTagging	删除对象标记（对象的特定版本）	
S3： DeleteObjectVersion	删除对象（对象的特定版本）	
S3： GetObject	• 获取对象 • HEAD 对象 • 后对象还原	
S3： GetObjectAcl	获取对象 ACL	
S3： GetObjectLegend	获取对象合法保留	
S3： GetObjectRetention	获取对象保留	
S3： GetObjectTagging	获取对象标记	
S3： GetObjectVersionTagging	获取对象标记（对象的特定版本）	
S3： GetObjectVersion	GET 对象（对象的特定版本）	
S3： ListMultipartUploadPart	列出部件， POST 对象还原	
S3： PutObject	• PUT 对象 • PUT 对象—复制 • 后对象还原 • 启动多部件上传 • 完成多部件上传 • 上传部件 • 上传部件—复制	
S3： PutObjectLegend	PUT 对象合法保留	
S3： PutObjectRetention	放置对象保留	
S3： PutObjectTagging	放置对象标记	
S3： PutObjectVersionTagging	PUT 对象标记（对象的特定版本）	

权限	**S3 REST API** 操作	为 **StorageGRID** 自定义
S3： PutOverwriteObject	• PUT 对象 • PUT 对象—复制 • PUT 对象标记 • 删除对象标记 • 完成多部件上传	是的。
S3： RestoreObject	后对象还原	

使用**PutOverwriteObject**权限

S3： PutOverwriteObject 权限是一种自定义 StorageGRID 权限，适用场景 可通过此权限创建或更新对象。此权限的设置可确定客户端是否可以覆盖对象的数据，用户定义的元数据或 S3 对象标记。

此权限的可能设置包括：

- * 允许 *：客户端可以覆盖对象。这是默认设置。
- * 拒绝 *：客户端无法覆盖对象。如果设置为 deny ，则 PutOverwriteObject 权限的工作原理如下：
  - 如果在同一路径中找到现有对象：
    - 无法覆盖对象的数据，用户定义的元数据或 S3 对象标记。
    - 正在执行的任何载入操作均会取消，并返回错误。
    - 如果启用了 S3 版本控制，则 deny 设置将阻止 PUT 对象标记或删除对象标记操作修改对象及其非最新版本的标记集。
  - 如果未找到现有对象，此权限将不起作用。
- 如果不存在此权限，则效果与设置了 allow 时相同。

> 如果当前 S3 策略允许覆盖，并且 PutOverwriteObject 权限设置为 deny ，则客户端无法覆盖对象的数据，用户定义的元数据或对象标记。此外、如果选中了*阻止客户端修改*复选框(配置>*网格选项*)、则该设置将覆盖PutOverwriteObject权限的设置。

相关信息

"S3 组策略示例"

指定策略中的条件

条件用于定义策略何时生效。条件包括运算符和键值对。

条件使用键值对进行评估。一个条件元素可以包含多个条件，每个条件可以包含多个键值对。条件块使用以下格式：

```
Condition: {
     <em>condition_type</em>: {
          <em>condition_key</em>: <em>condition_values</em>
```

在以下示例中， ipaddress 条件使用 SourceIp 条件密钥。

```
"Condition": {
    "IpAddress": {
      "aws:SourceIp": "54.240.143.0/24"
        ...
},
        ...
```

支持的条件运算符

条件运算符分为以下几类：

- string
- 数字
- 布尔值
- IP 地址
- 空检查

条件运算符	**Description**
StringEquals	根据完全匹配（区分大小写）将键与字符串值进行比较。
StringNotEquals	根据否定匹配（区分大小写）将键与字符串值进行比较。
StringEqualsIgnoreCase	根据完全匹配将键与字符串值进行比较（忽略大小写）。
StringNotEqualsIgnoreCase	根据否定的匹配将键与字符串值进行比较（忽略大小写）。
StringLike	根据完全匹配（区分大小写）将键与字符串值进行比较。可以包括 * 和？通配符。
StringNotLike	根据否定匹配（区分大小写）将键与字符串值进行比较。可以包括 * 和？通配符。

条件运算符	Description
数值方程式	根据精确匹配将键与数字值进行比较。
NumericNotEquals	根据否定匹配将键与数字值进行比较。
数值 GreaterThan	根据 "`大于` " 匹配将键与数值进行比较。
NumericGreaterThals.	根据 "`大于或等于` " 匹配将键与数值进行比较。
数值细小	根据 "`小于` " 匹配将键与数值进行比较。
数值 ThalEquals	根据 "`小于或等于` " 匹配将键与数值进行比较。
池	根据 "`true 或 false` " 匹配将键与布尔值进行比较。
IP 地址	将密钥与 IP 地址或 IP 地址范围进行比较。
NotIpAddress	根据否定匹配将密钥与 IP 地址或 IP 地址范围进行比较。
空	检查当前请求上下文中是否存在条件密钥。

支持的条件密钥

类别	适用的条件密钥	Description
IP 运算符	AWS ： 源 Ip	将与发送请求的 IP 地址进行比较。可用于存储分段或对象操作。 • 注意： * 如果 S3 请求是通过管理节点和网关节点上的负载平衡器服务发送的，则此请求将与负载平衡器服务上游的 IP 地址进行比较。 • 注 * ： 如果使用第三方非透明负载平衡器，则此负载平衡器将与该负载平衡器的 IP 地址进行比较。任意 `X-Forwarded-For` 由于无法确定标题的有效性、因此将忽略标题。
资源 / 身份	AWS ： 用户名	将与发送请求的发件人用户名进行比较。可用于存储分段或对象操作。

类别	适用的条件密钥	Description
S3：ListBucket和 S3：ListBucketVersions权限	S3 ： 分隔符	将与 GET 分段或 GET 分段对象版本请求中指定的分隔符参数进行比较。
S3：ListBucket和 S3：ListBucketVersions权限	S3 ： 最大密钥	将与获取分段或获取分段对象版本请求中指定的 max-keys 参数进行比较。
S3：ListBucket和 S3：ListBucketVersions权限	S3 ： 前缀	将与获取分段或获取分段对象版本请求中指定的前缀参数进行比较。

在策略中指定变量

您可以在策略中使用变量填充可用的策略信息。您可以在中使用策略变量 `Resource` 中的元素和字符串比较 `Condition` Element。

在此示例中、为变量 `${aws:username}` 是资源元素的一部分：

```
"Resource": "arn:aws:s3:::_bucket-name/home_/${aws:username}/*"
```

在此示例中、为变量 `${aws:username}` 是条件块中条件值的一部分：

```
"Condition": {
    "StringLike": {
      "s3:prefix": "${aws:username}/*"
        ...
},
        ...
```

变量	Description
`${aws:SourceIp}`	使用 SourceIp 键作为提供的变量。
`${aws:username}`	使用 username 密钥作为提供的变量。
`${s3:prefix}`	使用特定于服务的前缀密钥作为提供的变量。
`${s3:max-keys}`	使用特定于服务的 max-keys 键作为提供的变量。
`${*}`	特殊字符。使用字符作为文字 * 字符。

变量	Description
`${?}`	特殊字符。使用字符作为文字？字符。
`${$}`	特殊字符。使用字符作为文字 $ 字符。

创建需要特殊处理的策略

有时，策略可能会授予对安全性有危险或对持续操作（例如锁定帐户的 root 用户）有危险的权限。在策略验证期间， StorageGRID S3 REST API 实施的限制性要低于 Amazon ，但在策略评估期间同样严格。

策略问题描述	Policy type	Amazon 行为	StorageGRID 行为
拒绝向自己授予对 root 帐户的任何权限	存储分段	有效且强制实施，但 root 用户帐户保留所有 S3 存储分段策略操作的权限	相同
拒绝用户 / 组的任何权限	组	有效且强制实施	相同
允许外部帐户组拥有任何权限	存储分段	主体无效	有效，但如果某个策略允许，则所有 S3 存储分段策略操作的权限均会返回 405 Method not allowed 错误
允许外部帐户 root 或用户拥有任何权限	存储分段	有效，但如果某个策略允许，则所有 S3 存储分段策略操作的权限均会返回 405 Method not allowed 错误	相同
允许所有人对所有操作拥有权限	存储分段	有效，但对所有 S3 存储分段策略操作的权限会为外部帐户 root 和用户返回 405 Method not allowed 错误	相同
拒绝任何人对所有操作的权限	存储分段	有效且强制实施，但 root 用户帐户保留所有 S3 存储分段策略操作的权限	相同
主体是不存在的用户或组	存储分段	主体无效	有效
资源不是 S3 存储分段	组	有效	相同
主体是一个本地组	存储分段	主体无效	有效

策略问题描述	Policy type	Amazon 行为	StorageGRID 行为
策略授予非所有者帐户（包括匿名帐户）放置对象的权限	存储分段	有效。对象由创建者帐户拥有，并且存储分段策略不适用。创建者帐户必须使用对象 ACL 为对象授予访问权限。	有效。对象由存储分段所有者帐户拥有。存储分段策略适用。

一次写入多读（ WORM ）保护

您可以创建一次写入多读（ Write Once Read-Many ， WORM ）分段来保护数据，用户定义的对象元数据和 S3 对象标记。您可以配置 WORM 分段，以便创建新对象并防止覆盖或删除现有内容。请使用此处所述的方法之一。

为了确保覆盖始终被拒绝，您可以：

- 在网格管理器中、转到*配置*>*网格选项*、然后选中*阻止客户端修改*复选框。
- 应用以下规则和 S3 策略：
  - 向 S3 策略添加 PutOverwriteObject deny 操作。
  - 将 DeleteObject deny 操作添加到 S3 策略中。
  - 向 S3 策略添加 PUT 对象允许操作。

> 在 S3 策略中将 DeleteObject 设置为 deny 不会阻止 ILM 在存在 "`zero copies after 30 days` " 等规则时删除对象。

> 即使应用了所有这些规则和策略，它们也不会防止并发写入（请参见情况 A ）。它们可以防止顺序完成的覆盖（请参见情况 B ）。

- 情形 A* ： 并发写入（不受保护）

```
/mybucket/important.doc
PUT#1 ---> OK
PUT#2 -------> OK
```

- 情形 B* ： 顺序完成的覆盖（防止）

```
/mybucket/important.doc
PUT#1 -------> PUT#2 ---X (denied)
```

相关信息

"使用 ILM 管理对象"

"创建需要特殊处理的策略"

"StorageGRID ILM 规则如何管理对象"

"S3 组策略示例"

S3 策略示例

使用本节中的示例为分段和组构建 StorageGRID 访问策略。

S3 存储分段策略示例

存储分段策略用于指定附加此策略的存储分段的访问权限。存储分段策略使用 S3 PutBucketPolicy API 进行配置。

可以按照以下命令使用 AWS 命令行界面配置存储分段策略：

```
> aws s3api put-bucket-policy --bucket examplebucket --policy
<em>file://policy.json</em>
```

示例：允许每个人对某个存储分段进行只读访问

在此示例中，允许包括匿名用户在内的所有人列出存储分段中的对象，并对存储分段中的所有对象执行 GET Object 操作。所有其他操作都将被拒绝。请注意，此策略可能不会特别有用，因为除了帐户 root 之外，没有其他人有权写入存储分段。

```
{
  "Statement": [
    {
      "Sid": "AllowEveryoneReadOnlyAccess",
      "Effect": "Allow",
      "Principal": "*",
      "Action": [ "s3:GetObject", "s3:ListBucket" ],
      "Resource":
["arn:aws:s3:::examplebucket","arn:aws:s3:::examplebucket/*"]
    }
  ]
}
```

示例：允许一个帐户中的每个人完全访问某个存储分段，而另一帐户中的每个人只读访问某个存储分段

在此示例中、一个指定帐户中的每个人都可以完全访问某个存储分段、而另一个指定帐户中的每个人只能列出存储分段并对以开头的存储分段中的对象执行GetObject操作 `shared/` 对象密钥前缀。

> 在 StorageGRID 中，非所有者帐户创建的对象（包括匿名帐户）归存储分段所有者帐户所有。存储分段策略适用场景 这些对象。

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "95390887230002558202"
      },
      "Action": "s3:*",
      "Resource": [
        "arn:aws:s3:::examplebucket",
        "arn:aws:s3:::examplebucket/*"
      ]
    },
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "31181711887329436680"
      },
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::examplebucket/shared/*"
    },
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "31181711887329436680"
      },
      "Action": "s3:ListBucket",
      "Resource": "arn:aws:s3:::examplebucket",
      "Condition": {
        "StringLike": {
          "s3:prefix": "shared/*"
        }
      }
    }
  ]
}
```

示例：允许每个人对某个存储分段进行只读访问，并允许指定组进行完全访问

在此示例中、允许包括anonymous在内的所有人列出存储分段并对存储分段中的所有对象执行GET Object操作、而只允许用户属于该组 `Marketing` 在指定帐户中、允许完全访问。

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::95390887230002558202:federated-
group/Marketing"
      },
      "Action": "s3:*",
      "Resource": [
        "arn:aws:s3:::examplebucket",
        "arn:aws:s3:::examplebucket/*"
      ]
    },
    {
      "Effect": "Allow",
      "Principal": "*",
      "Action": ["s3:ListBucket","s3:GetObject"],
      "Resource": [
        "arn:aws:s3:::examplebucket",
        "arn:aws:s3:::examplebucket/*"
      ]
    }
  ]
}
```

示例：如果客户端位于 **IP** 范围内，则允许每个人对存储分段进行读写访问

在此示例中，允许包括匿名用户在内的所有人列出存储分段并对存储分段中的所有对象执行任何对象操作，前提是这些请求来自指定的 IP 范围（ 54.240.143.0 到 54.240.143.255 ，但 54.240.143.188 除外）。所有其他操作都将被拒绝，并且 IP 范围以外的所有请求都将被拒绝。

```
{
  "Statement": [
    {
      "Sid": "AllowEveryoneReadWriteAccessIfInSourceIpRange",
      "Effect": "Allow",
      "Principal": "*",
      "Action": [ "s3:*Object", "s3:ListBucket" ],
      "Resource":
["arn:aws:s3:::examplebucket","arn:aws:s3:::examplebucket/*"],
      "Condition": {
        "IpAddress": {"aws:SourceIp": "54.240.143.0/24"},
        "NotIpAddress": {"aws:SourceIp": "54.240.143.188"}
      }
    }
  ]
}
```

示例：允许指定的联合用户完全访问某个存储分段

在此示例中、允许联合用户Alex完全访问 `examplebucket` 存储分段及其对象。包括 "`root` " 在内的所有其他用户均被明确拒绝所有操作。但请注意， "`root` " 从不会被拒绝 PUT ， Get/DeleteBucketPolicy 的权限。

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::95390887230002558202:federated-user/Alex"
      },
      "Action": [
        "s3:*"
      ],
      "Resource": [
        "arn:aws:s3:::examplebucket",
        "arn:aws:s3:::examplebucket/*"
      ]
    },
    {
      "Effect": "Deny",
      "NotPrincipal": {
        "AWS": "arn:aws:iam::95390887230002558202:federated-user/Alex"
      },
      "Action": [
        "s3:*"
      ],
      "Resource": [
        "arn:aws:s3:::examplebucket",
        "arn:aws:s3:::examplebucket/*"
      ]
    }
  ]
}
```

示例： **PutOverwriteObject** 权限

在此示例中、将显示 `Deny` 对PutOverwriteObject和DeleteObject的影响可确保任何人都不能覆盖或删除对象的数据、用户定义的元数据和S3对象标记。

```
{
  "Statement": [
    {
      "Effect": "Deny",
      "Principal": "*",
      "Action": [
        "s3:PutOverwriteObject",
        "s3:DeleteObject",
        "s3:DeleteObjectVersion"
      ],
      "Resource": "arn:aws:s3:::wormbucket/*"
    },
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::95390887230002558202:federated-
group/SomeGroup"

},
      "Action": "s3:ListBucket",
      "Resource": "arn:aws:s3:::wormbucket"
    },
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::95390887230002558202:federated-
group/SomeGroup"

},
      "Action": "s3:*",
      "Resource": "arn:aws:s3:::wormbucket/*"
    }
  ]
}
```

相关信息

"对存储分段执行的操作"

S3 组策略示例

组策略用于指定附加此策略的组的访问权限。没有 `Principal` 策略中的元素、因为它是隐式的。组策略可使用租户管理器或 API 进行配置。

示例：使用租户管理器设置组策略

使用租户管理器添加或编辑组时，您可以选择要如何创建组策略，以定义此组的成员将具有的 S3 访问权限，如

下所示：

- * 无 S3 访问 * ： 默认选项。此组中的用户无权访问 S3 资源，除非使用存储分段策略授予访问权限。如果选择此选项，则默认情况下，只有 root 用户才能访问 S3 资源。
- * 只读访问 * ： 此组中的用户对 S3 资源具有只读访问权限。例如，此组中的用户可以列出对象并读取对象数据，元数据和标记。选择此选项后，只读组策略的 JSON 字符串将显示在文本框中。您不能编辑此字符串。
- * 完全访问 * ： 此组中的用户对 S3 资源（包括分段）具有完全访问权限。选择此选项后，完全访问组策略的 JSON 字符串将显示在文本框中。您不能编辑此字符串。
- * 自定义 * ： 组中的用户将获得您在文本框中指定的权限。

  在此示例中，组成员只能列出并访问指定存储分段中的特定文件夹（密钥前缀）。

示例：允许组完全访问所有存储分段

在此示例中，除非 bucket 策略明确拒绝，否则允许组中的所有成员对租户帐户拥有的所有分段进行完全访问。

```
{
  "Statement": [
    {
      "Action": "s3:*",
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::*"
    }
  ]
}
```

示例：允许组对所有分段进行只读访问

在此示例中，组的所有成员都对 S3 资源具有只读访问权限，除非 bucket 策略明确拒绝。例如，此组中的用户可以列出对象并读取对象数据，元数据和标记。

```
{
  "Statement": [
    {
      "Sid": "AllowGroupReadOnlyAccess",
      "Effect": "Allow",
      "Action": [
        "s3:ListAllMyBuckets",
        "s3:ListBucket",
        "s3:ListBucketVersions",
        "s3:GetObject",
        "s3:GetObjectTagging",
        "s3:GetObjectVersion",
        "s3:GetObjectVersionTagging"
      ],
      "Resource": "arn:aws:s3:::*"
    }
  ]
}
```

示例：仅允许组成员对存储分段中的"`folder`"具有完全访问权限

在此示例中，组成员只能列出并访问指定存储分段中的特定文件夹（密钥前缀）。请注意，在确定其他组策略和存储分段策略的隐私时，应考虑这些文件夹的访问权限。

```
{
  "Statement": [
    {
      "Sid": "AllowListBucketOfASpecificUserPrefix",
      "Effect": "Allow",
      "Action": "s3:ListBucket",
      "Resource": "arn:aws:s3:::department-bucket",
      "Condition": {
        "StringLike": {
          "s3:prefix": "${aws:username}/*"
        }
      }
    },
    {
      "Sid": "AllowUserSpecificActionsOnlyInTheSpecificUserPrefix",
      "Effect": "Allow",
      "Action": "s3:*Object",
      "Resource": "arn:aws:s3:::department-bucket/${aws:username}/*"
    }
  ]
}
```

相关信息

"使用租户帐户"

"使用PutOverwriteObject权限"

"一次写入多读（ WORM ）保护"

为REST API配置安全性

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 。

  配置负载平衡器端点时，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。有关详细信息，请参见有关管理 StorageGRID 的说明。

- 默认情况下， StorageGRID 使用 HTTPS 与存储节点进行客户端通信，并在网关节点上使用 CLB 服务。

可以选择为这些连接启用 HTTP 。例如，您可能希望将 HTTP 用于测试或其他非生产目的。有关详细信息，请参见有关管理 StorageGRID 的说明。

CLB 服务已弃用。

- 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

客户端可以连接到网关节点或管理节点上的负载平衡器服务，直接连接到存储节点或网关节点上的 CLB 服务。

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或网关节点上的 CLB 服务时，它们会使用安装 StorageGRID 系统时为存储节点生成的系统生成的服务器证书（由系统证书颁发机构签名）， 或网格管理员为网格提供的一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。

有关配置负载平衡器端点的信息，请参见管理 StorageGRID 的说明，以及有关为直接连接到存储节点或网关节点上的 CLB 服务添加单个自定义服务器证书的说明。

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

相关信息

"管理 StorageGRID"

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。

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）。

支持的密码套件

TLS 版本	密码套件的 IANA 名称
1.2	tls_ECDHE_RSA_WIT_AES_256_GCM_SHA384
1.2	tls_ECDHE_RSA_WIT_CHACHA20_POLY1305_SHA256
1.2	tls_ECDHE_RSA_WIT_AES_128_GCM_SHA256
1.3	tls_aes_256_gcm_SHA384
1.3	tls_chacHA20_POLY1305_SHA256
1.3	tls_aes_128_gcm_SHA256

已弃用密码套件

以下密码套件已弃用。未来版本将删除对这些密码的支持。

IANA 名称
tls_rsa_and_aes_128_gcm_SHA256
tls_rsa_and_aes_256_gcm_SHA384

相关信息

"如何配置客户端连接"

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"
- "访问和查看审核日志"

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. 使用支持的浏览器登录到网格管理器。
2. 在信息板上，找到协议操作部分。

   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。

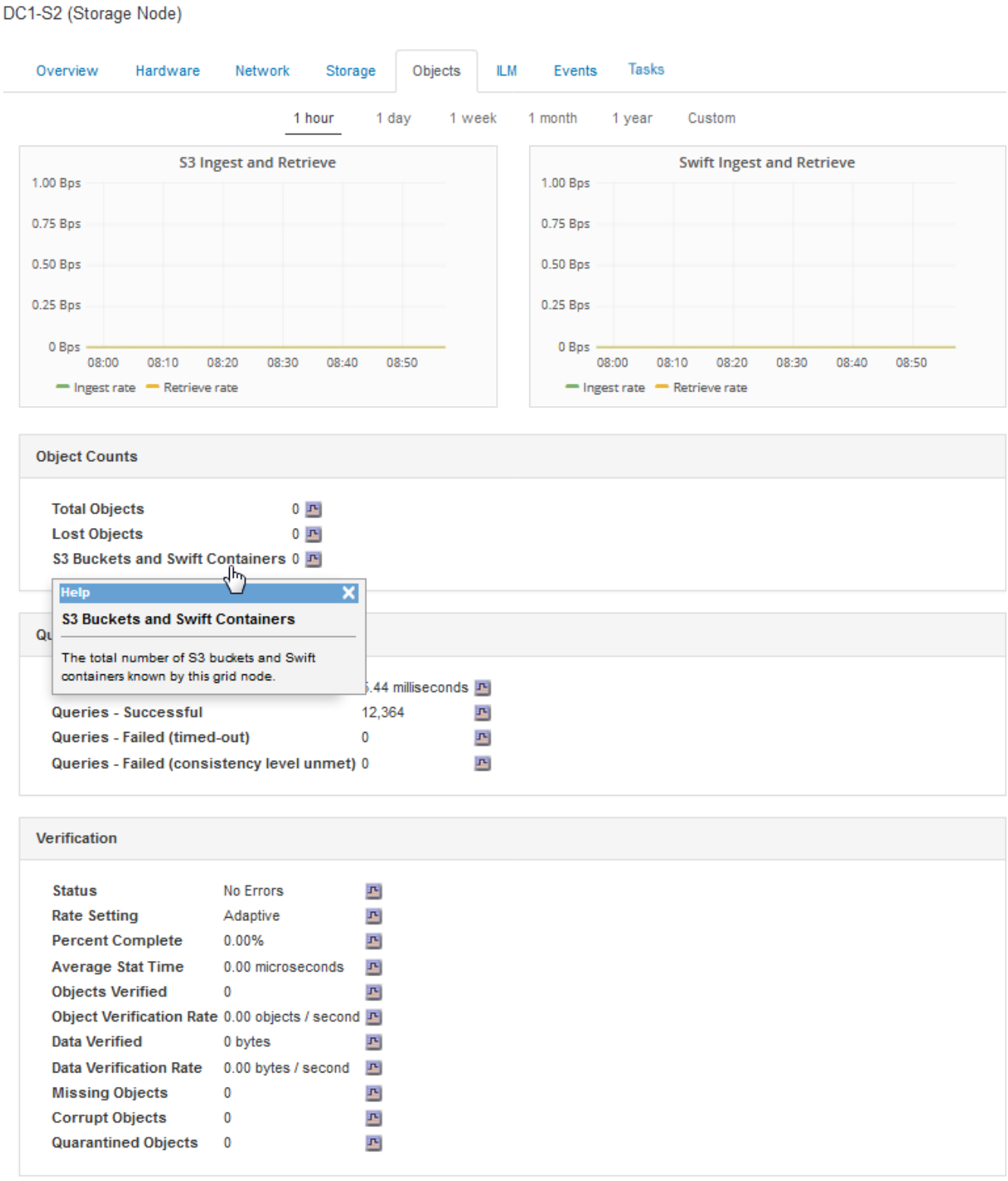
   这些图表显示了定向到网格中负载平衡器端点的所有客户端流量的趋势。您可以选择以小时，天，周，月或年为单位的时间间隔， 或者，您也可以应用自定义间隔。

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。

您需要的内容

- 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、并保存一个新文件 `audit.log` 文件已启动。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` file、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
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 **S3** 操作

StorageGRID 审核日志会跟踪多个存储分段操作和对象操作。

审核日志中跟踪的存储分段操作

- 删除存储分段
- 删除存储分段标记
- 删除多个对象
- 获取存储分段（列出对象）
- 获取 Bucket 对象版本
- 获取存储分段标记
- 头存储分段
- 放入存储分段
- PUT 存储分段合规性
- 放置存储分段标记
- PUT 存储分版本

审核日志中跟踪的对象操作

- 完成多部件上传
- 上传部件（ ILM 规则使用严格或平衡的载入行为时）
- 上传部件—复制（当 ILM 规则使用严格或平衡的载入行为时）
- 删除对象
- 获取对象
- HEAD 对象
- 后对象还原
- PUT 对象
- PUT 对象—复制

相关信息

"对存储分段执行的操作"

"对对象执行的操作"

活动，空闲和并发 **HTTP** 连接的优势

如何配置 HTTP 连接可能会影响 StorageGRID 系统的性能。根据 HTTP 连接是活动连接还是空闲连接，或者您有多个并发连接，配置会有所不同。

您可以确定以下类型的 HTTP 连接的性能优势：

- 空闲 HTTP 连接
- 活动 HTTP 连接

- 并发 HTTP 连接

相关信息

- "保持空闲 HTTP 连接处于打开状态的优势"
- "活动 HTTP 连接的优势"
- "并发 HTTP 连接的优势"
- "为读取和写入操作分隔 HTTP 连接池"

保持空闲 **HTTP** 连接处于打开状态的优势

即使客户端应用程序处于闲置状态，您也应保持 HTTP 连接处于打开状态，以允许客户端应用程序通过打开的连接执行后续事务。根据系统测量结果和集成经验，您应将闲置的 HTTP 连接保持打开状态最多 10 分钟。StorageGRID 可能会自动关闭保持打开和闲置超过 10 分钟的 HTTP 连接。

开放式和空闲 HTTP 连接具有以下优势：

- 从 StorageGRID 系统确定必须执行 HTTP 事务的时间缩短到 StorageGRID 系统可以执行此事务的时间

  缩短延迟是主要优势，尤其是在建立 TCP/IP 和 TLS 连接所需的时间方面。

- 通过在先前执行的传输中填充 TCP/IP 慢速启动算法来提高数据传输速率
- 瞬时通知多种中断客户端应用程序与 StorageGRID 系统之间连接的故障情况

- 保持闲置连接打开多长时间是对与现有连接相关的慢速启动优势与将连接分配给内部系统资源的理想平衡。

活动 **HTTP** 连接的优势

要直接连接到存储节点或网关节点上的 CLB 服务（已弃用），您应将活动 HTTP 连接的持续时间限制为最多 10 分钟，即使 HTTP 连接持续执行事务也是如此。

- 连接应保持打开状态的最长持续时间是为了权衡连接持久性的优势与将连接分配给内部系统资源的理想方式。

对于客户端与存储节点或 CLB 服务的连接，限制活动 HTTP 连接具有以下优势：

- 在 StorageGRID 系统之间实现最佳负载平衡。

  使用 CLB 服务时，您应防止使用长 - 寿命的 TCP/IP 连接，以优化整个 StorageGRID 系统的负载平衡。您应将客户端应用程序配置为跟踪每个 HTTP 连接的持续时间，并在设置的时间后关闭 HTTP 连接，以便可以重新建立和重新平衡 HTTP 连接。

  在客户端应用程序建立 HTTP 连接时， CLB 服务会在整个 StorageGRID 系统中平衡负载。随着时间的推移，随着负载平衡要求的变化， HTTP 连接可能不再是最佳连接。当客户端应用程序为每个事务建立单独的 HTTP 连接时，系统会执行最佳的负载平衡，但这会抵消与持久连接相关的更有价值的收益。

  CLB 服务已弃用。

- 允许客户端应用程序将 HTTP 事务定向到具有可用空间的 LDR 服务。

- 允许开始维护过程。

  某些维护过程仅在所有正在进行的 HTTP 连接完成后才会启动。

对于客户端与负载平衡器服务的连接，限制打开连接的持续时间对于允许某些维护过程立即启动非常有用。如果客户端连接的持续时间不受限制，则自动终止活动连接可能需要几分钟的时间。

并发 **HTTP** 连接的优势

您应保持与 StorageGRID 系统的多个 TCP/IP 连接处于开放状态，以实现并行处理，从而提高性能。并行连接的最佳数量取决于多种因素。

并发 HTTP 连接具有以下优势：

- 缩短延迟

  事务可以立即启动，而不是等待其他事务完成。

- 提高吞吐量

  StorageGRID 系统可以执行并行事务并提高聚合事务吞吐量。

客户端应用程序应建立多个 HTTP 连接。当客户端应用程序必须执行事务时，它可以选择并立即使用当前未处理事务的任何已建立连接。

在性能开始下降之前，每个 StorageGRID 系统的拓扑对于并发事务和连接具有不同的峰值吞吐量。峰值吞吐量取决于计算资源，网络资源，存储资源和 WAN 链路等因素。服务器和服务的数量以及 StorageGRID 系统支持的应用程序的数量也是因素。

StorageGRID 系统通常支持多个客户端应用程序。在确定客户端应用程序所使用的最大并发连接数时，应牢记这一点。如果客户端应用程序包含多个软件实体，每个软件实体都与 StorageGRID 系统建立连接，则应添加这些实体之间的所有连接。在以下情况下，您可能需要调整并发连接的最大数量：

- StorageGRID 系统的拓扑会影响系统可以支持的并发事务和连接的最大数量。
- 如果客户端应用程序通过带宽有限的网络与 StorageGRID 系统进行交互，则可能需要降低并发程度，以确保各个事务在合理时间内完成。
- 当许多客户端应用程序共享 StorageGRID 系统时，您可能需要降低并发程度，以避免超过系统限制。

为读取和写入操作分隔 **HTTP** 连接池

您可以使用单独的 HTTP 连接池执行读写操作，并控制每个连接池要使用的池容量。通过单独的 HTTP 连接池，您可以更好地控制事务并平衡负载。

客户端应用程序可以创建检索占主导地位（读取）或存储占主导地位（写入）的负载。由于读取和写入事务使用单独的 HTTP 连接池，因此您可以调整每个池中用于读取或写入事务的数量。