



# **StorageGRID 如何实施 S3 REST API**

## **StorageGRID 11.7**

NetApp  
April 12, 2024

# 目录

- StorageGRID 如何实施 S3 REST API ..... 1
  - 客户端请求冲突 ..... 1
  - 一致性控制 ..... 1
  - StorageGRID ILM 规则如何管理对象 ..... 3
  - 对象版本控制 ..... 4
  - 使用S3 REST API配置S3对象锁定 ..... 5
  - 创建 S3 生命周期配置 ..... 10
  - 实施 S3 REST API 的建议 ..... 14

# StorageGRID 如何实施 S3 REST API

## 客户端请求冲突

冲突的客户端请求（例如，两个客户端写入同一密钥）将以 "最新成功" 为基础进行解决。

"最新赢单" 评估的时间取决于 StorageGRID 系统何时完成给定请求，而不是 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：

- 全部：所有节点均立即接收数据、否则请求将失败。
- \* 强 - 全局 \*：保证所有站点中所有客户端请求的写入后读一致性。
- \* 强站点 \*：保证站点内所有客户端请求的写入后读一致性。
- 读后新写入：(默认)为新对象提供读后写入一致性、并最终为对象更新提供一致性。提供高可用性和数据保护保证。建议用于大多数情况。
- 可用：为新对象和对象更新提供最终一致性。对于S3存储分段、请仅在需要时使用(例如、对于包含很少读取的日志值的存储分段、或者对于不存在的密钥执行HEAD或GET操作)。S3 FabricPool 存储分段不支持。

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

如果 head 或 get 操作使用 read-after-new-write 一致性控制，则 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、则会在下一个一致性级别重复执行该查找、直到达到与强全局行为等效的一致性级别为止。

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与强全局行为等效的一致性级别。由于此一致性级别要求每个站点提供多个对象元数据副本、因此如果同一站点上的两个或

更多存储节点不可用、您可能会收到大量500个内部服务器错误。

除非您需要与 Amazon S3 类似的一致性保证，否则您可以通过将一致性控制设置为 "Available" 来防止机头和 GET 操作出现这些错误。当机头或 GET 操作使用 "Available" 一致性控制时，StorageGRID 仅提供最终一致性。它不会在一致性级别提高时重试失败的操作、因此不要求对象元数据的多个副本可用。

## 为 API 操作指定一致性控制

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: authorization name
Host: host
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。

## 为存储分段指定一致性控制

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、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。

## [[how—sistic-controls-and-ilm-交互]]如何通过一致性控制和ILM规则交互来影响数据保护

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：

- **\*Dual Commit\***：StorageGRID 会立即创建对象的临时副本并将成功结果返回给客户端。如果可能，将创建 ILM 规则中指定的副本。
- **\*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。



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之后，在将对象元数据分发到该远程站点之前，客户端可能会收到一条成功消息。在这种情况下，对象元数据的保护级别与对象数据的保护级别不匹配。如果本地站点在载入后不久丢失，则对象元数据将丢失。无法检索此对象。

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

相关信息

["使用 ILM 管理对象"](#)

["获取存储分段一致性"](#)

["PUT 存储分段一致性"](#)

## 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 存储分段启用上次访问时间更新。使用 Put Bucket 最后一个访问时间请求、租户经理(请参见 ["启用或禁用上次访问时间更新"](#))或租户管理 API。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

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"](#)

## 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(对于问题“仅将此规则应用于较早对象版本？”、请选择\*是\* 在中 ["创建ILM规则向导的第1步"](#))。对象更新后，其先前版本将变为非最新版本。通过使用“noncurrent time”筛选器、您可以创建可减少先前版本对象的存储影响的策略。



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。

请参见 ["S3 版本对象的 ILM 规则和策略（示例 4）"](#)。

## 使用S3 REST API配置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S3对象锁定的情况下创建分段。您可以为每个存储分段指定默认保留、也可以为每个对象版本指定保留设置。

### 如何为存储分段启用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。

S3对象锁定是一种永久性设置、只有在创建存储分段时才能启用。创建分段后、您无法添加或禁用S3对象锁定。

要为存储分段启用S3对象锁定、请使用以下方法之一：

- 使用租户管理器创建存储分段。请参见 ["创建 S3 存储分段"](#)。
- 使用PUT Bucket请求和创建存储分段 `x-amz-bucket-object-lock-enabled` 请求标题。请参见 ["对存储分段执行的操作"](#)。

S3对象锁定需要分段版本控制、创建分段时会自动启用此功能。您不能暂停分段的版本控制。请参见 ["对象版本控制"](#)。

### 存储分段的默认保留设置

为存储分段启用S3对象锁定后、您可以选择为存储分段启用默认保留、并指定默认保留模式和默认保留期限。

## 默认保留模式

- 在合规模式下：
  - 在达到保留截止日期之前、无法删除此对象。
  - 对象的保留截止日期可以增加、但不能减少。
  - 在达到该日期之前、无法删除对象的保留截止日期。
- 在监管模式下：
  - 使用的用户 `s3:BypassGovernanceRetention` 权限可以使用 `x-amz-bypass-governance-retention: true` 请求标头以绕过保留设置。
  - 这些用户可以在达到保留截止日期之前删除对象版本。
  - 这些用户可以增加、减少或删除对象的保留截止日期。

## 默认保留期限

每个存储分段都可以指定默认保留期限(以年或天为单位)。

## 如何设置存储分段的默认保留

要设置存储分段的默认保留时间、请使用以下方法之一：

- 通过租户管理器管理存储分段设置。请参见 ["创建 S3 存储区。"](#) 和 ["更新S3对象锁定默认保留"](#)。
- 问题描述 为分段指定默认模式和默认天数或年数的Put对象锁定配置请求。

## PUT 对象锁定配置

通过Put Object Lock Configuration请求、您可以设置和修改启用了S3对象锁定的存储分段的默认保留模式和默认保留期限。您还可以删除先前配置的默认保留设置。

如果将新对象版本写入存储分段、则会应用默认保留模式 `x-amz-object-lock-mode` 和 `x-amz-object-lock-retain-until-date` 未指定。默认保留期限用于计算保留截止日期IF `x-amz-object-lock-retain-until-date` 未指定。

如果在载入对象版本后修改了默认保留期限，则对象版本的保留日期将保持不变，不会使用新的默认保留期限重新计算。

您必须具有 `s3:PutBucketObjectLockConfiguration` 权限、或者作为帐户root用户来完成此操作。

◦ `Content-MD5` 必须在Put请求中指定请求标头。

### 请求示例

此示例为存储分段启用S3对象锁定、并将默认保留模式设置为合规、将默认保留期限设置为6年。



```
PUT /bucket?object-lock HTTP/1.1
Accept-Encoding: identity
Content-Length: 308
Host: host
Content-MD5: request header
User-Agent: s3sign/1.0.0 requests/2.24.0 python/3.8.2
X-Amz-Date: date
X-Amz-Content-SHA256: authorization-string
Authorization: authorization-string

<ObjectLockConfiguration>
  <ObjectLockEnabled>Enabled</ObjectLockEnabled>
  <Rule>
    <DefaultRetention>
      <Mode>COMPLIANCE</Mode>
      <Years>6</Years>
    </DefaultRetention>
  </Rule>
</ObjectLockConfiguration>
```

## 如何确定存储分段的默认保留

要确定是否为存储分段启用了S3对象锁定并查看默认保留模式和保留期限、请使用以下方法之一：

- 在租户管理器中查看存储分段。请参见 ["查看S3存储分段"](#)。
- 问题描述 获取对象锁定配置请求。

### 获取对象锁定配置

通过"Get Object Lock Configuration"(获取对象锁定配置)请求、您可以确定是否为存储分段启用了S3对象锁定、如果已启用、则查看是否为存储分段配置了默认保留模式和保留期限。

如果将新对象版本写入存储分段、则会应用默认保留模式 `x-amz-object-lock-mode` 未指定。默认保留期限用于计算保留截止日期IF `x-amz-object-lock-retain-until-date` 未指定。

您必须具有 `s3:GetBucketObjectLockConfiguration` 权限、或者作为帐户root用户来完成此操作。

### 请求示例

```
GET /bucket?object-lock HTTP/1.1
Host: host
Accept-Encoding: identity
User-Agent: aws-cli/1.18.106 Python/3.8.2 Linux/4.4.0-18362-Microsoft
botocore/1.17.29
x-amz-date: date
x-amz-content-sha256: authorization-string
Authorization: authorization-string
```

#### 响应示例

```
HTTP/1.1 200 OK
x-amz-id-2:
iVmcB7OXXJRkRH1FiVq1151/T24gRfpwpuZrEG11Bb9ImOMAAe98oxSpXlknabA0LTvBYJpSIX
k=
x-amz-request-id: B34E94CACB2CEF6D
Date: Fri, 04 Sep 2020 22:47:09 GMT
Transfer-Encoding: chunked
Server: AmazonS3

<?xml version="1.0" encoding="UTF-8"?>
<ObjectLockConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <ObjectLockEnabled>Enabled</ObjectLockEnabled>
  <Rule>
    <DefaultRetention>
      <Mode>COMPLIANCE</Mode>
      <Years>6</Years>
    </DefaultRetention>
  </Rule>
</ObjectLockConfiguration>
```

## 如何指定对象的保留设置

启用了S3对象锁定的存储分段可以包含具有和不具有S3对象锁定保留设置的对象组合。

对象级保留设置可通过S3 REST API来指定。对象的保留设置将覆盖存储分段的任何默认保留设置。

您可以为每个对象指定以下设置：

- 保留模式：合规性或监管。
- `*retain-until-date*`：指定StorageGRID 必须保留对象版本多长时间的日期。
  - 在合规模式下、如果保留截止日期为未来日期、则可以检索对象、但无法修改或删除它。保留截止日期可以增加、但不能减少或删除此日期。
  - 在监管模式下、具有特殊权限的用户可以绕过保留截止日期设置。他们可以在对象版本的保留期限到期

之前将其删除。它们还可以增加、减少甚至删除保留截止日期。

- **\* 合法保留 \***：对对象版本应用合法保留时，会立即锁定该对象。例如，您可能需要将与调查或法律争议相关的对象进行法律保留。合法保留没有到期日期，但在明确删除之前始终有效。

对象的合法保留设置与保留模式和保留截止日期无关。如果某个对象版本处于合法保留状态、则任何人都无法删除该版本。

要在向存储分段添加对象版本时指定S3对象锁定设置、请使用问题描述 A **"PUT 对象"**，**"PUT 对象—复制"**或 **"启动多部件上传"** 请求。

您可以使用以下命令：

- `x-amz-object-lock-mode`，可以是合规性或监管(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。

您可以使用 `s3:object-lock-remaining-retention-days` 策略条件关键字、用于限制对象允许的最短和最长保留期限。

## 如何更新对象的保留设置

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
  - 模式值可以是合规性或监管(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。

## 如何使用监管模式

拥有用户 `s3:BypassGovernanceRetention` 权限可以绕过使用监管模式的对象的活动保留设置。任何删除或放置对象保留操作都必须包含 `x-amz-bypass-governance-retention:true` 请求标题。这些用户可以执行以下附加操作：

- 执行删除对象或删除多个对象操作、以便在对象保留期限到期之前删除该对象版本。  
无法删除处于合法保留状态的对象。合法保留必须关闭。
- 执行Put对象保留操作、以便在对象保留期限结束之前将对象版本的模式从监管更改为合规。  
绝不允许将模式从合规性更改为监管。
- 执行Put Object保留操作以增加、减少或删除对象版本的保留期限。

### 相关信息

- ["使用 S3 对象锁定管理对象"](#)
- ["使用S3对象锁定保留对象"](#)
- ["《Amazon Simple Storage Service 用户指南：使用 S3 对象锁定》"](#)

## 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：对象生命周期管理》"](#)。请注意，StorageGRID 仅支持到期操作，不支持过渡操作。

### 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 ["ILM 如何在对象的整个生命周期内运行"](#)。



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。

```
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 ["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": {}
}
```

## 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

根据首次创建分段的时间、请遵循这些对象键名建议。

在**StorageGRID 11.4**或更早版本中创建的分段

- 不要使用随机值作为对象键的前四个字符。这与 AWS 以前针对密钥前缀的建议不同。请改用非随机、非唯一的前缀、例如 image。
- 如果按照以前的AWS建议在密钥前缀中使用随机和唯一字符、请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

在**StorageGRID 11.4**或更高版本中创建的分段

不需要限制对象密钥名称以满足性能最佳实践。在大多数情况下、对象密钥名称的前四个字符可以使用随机值。



但S3工作负载例外、它会在短时间后持续删除所有对象。为了最大限度地降低此使用情形对性能的影响、请每隔数千个对象更改一次密钥名称的前导部分、并使用日期之类的内容。例如、假设S3客户端通常每秒写入2、000个对象、而ILM或存储分段生命周期策略将在三天后删除所有对象。为了最大限度地降低对性能的影响、您可以使用如下模式命名密钥：

/mybucket/mydir/yyyymmddhhmmss-random\_UUID.jpg

## 关于"范围读取"的建议

如果 ["用于压缩存储对象的全局选项"](#) 已启用、则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获取对象操作效率尤其低下；例如，从 50 GB 压缩对象读取 10 MB 范围的操作效率低下。

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"](#)
- ["管理 StorageGRID"](#)

## 版权信息

版权所有 © 2024 NetApp, Inc.。保留所有权利。中国印刷。未经版权所有者事先书面许可，本文档中受版权保护的任何部分不得以任何形式或通过任何手段（图片、电子或机械方式，包括影印、录音、录像或存储在电子检索系统中）进行复制。

从受版权保护的 NetApp 资料派生的软件受以下许可和免责声明的约束：

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

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

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

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

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

## 商标信息

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