



# StorageGRID如何實作 S3 REST API

## StorageGRID software

NetApp  
May 29, 2026

# 目錄

StorageGRID如何實作 S3 REST API	1
客戶端請求衝突	1
一致性值	1
一致性值	1
使用“新寫後讀”和“可用”一致性	2
指定 API 操作的一致性	2
指定 bucket 的一致性	2
一致性控制和 ILM 規則如何相互作用以影響資料保護	2
一致性和 ILM 規則如何交互的範例	3
物件版本控制	3
ILM 和版本控制	4
使用 S3 REST API 設定 S3 物件鎖	4
如何為儲存桶啟用 S3 物件鎖定	4
儲存桶的預設保留設定	4
如何設定儲存桶的預設保留	5
如何決定儲存桶的預設保留時間	6
如何指定物件的保留設定	7
如何更新物件的保留設定	8
如何使用治理模式	9
建立 S3 生命週期配置	9
什麼是生命週期配置	9
建立生命週期配置	10
將生命週期配置應用於儲存桶	12
驗證儲存桶生命週期到期是否適用於對象	12
實作 S3 REST API 的建議	13
對不存在對象的 HEAD 建議	13
對象鍵的建議	13
“範圍讀取”的建議	14

# StorageGRID如何實作 S3 REST API

## 客戶端請求衝突

衝突的客戶端請求（例如兩個客戶端寫入同一個金鑰）將根據「最新勝利」的原則解決。

「最新勝利」評估的時間取決於StorageGRID系統完成給定請求的時間，而不是 S3 用戶端開始操作的時間。

## 一致性值

一致性在物件的可用性和不同儲存節點和站點之間的物件的一致性之間提供了平衡。您可以根據應用程式的需要更改一致性。

預設情況下，StorageGRID保證新建立物件的讀寫一致性。成功完成 PUT 之後的任何 GET 都將能夠讀取新寫入的資料。現有物件的覆蓋、元資料更新和刪除最終是一致的。覆蓋通常需要幾秒鐘或幾分鐘才能傳播，但可能需要長達 15 天。

如果要以不同的一致性執行物件操作，您可以：

- 指定一致性**每個桶**。
- 指定一致性**每個 API 操作**。
- 透過執行下列任務之一來變更預設的網格範圍一致性：
  - 在網格管理員中，前往\*配置\* > 系統 > 儲存設定 > 預設一致性。
  - 。



對網格範圍一致性的變更僅適用於變更設定後建立的儲存桶。要確定更改的詳細信息，請參閱位於 `/var/local/log`（搜尋\*consistencyLevel\*）。

## 一致性值

一致性會影響StorageGRID用於追蹤物件的元資料在節點之間的分佈方式，進而影響物件對於客戶端請求的可用性。

您可以將儲存桶或 API 操作的一致性設定為下列值之一：

- 全部：所有節點立即接收數據，否則請求失敗。
- 強全域：保證所有網站上所有客戶端請求的讀寫一致性。
- 強站點：保證站點內所有客戶端請求的讀寫一致性。
- **Read-after-new-write**：（預設）為新物件提供讀取後寫入一致性，並為物件更新提供最終一致性。提供高可用性和資料保護保證。在大多數情況下建議使用。
- 可用：為新物件和物件更新提供最終一致性。對於 S3 儲存桶，僅在需要時使用（例如，對於包含很少讀取的日誌值的儲存桶，或對於不存在的鍵的 HEAD 或 GET 操作）。不支援 S3 FabricPool儲存桶。

## 使用“新寫後讀”和“可用”一致性

當 HEAD 或 GET 操作使用「新寫後讀取」一致性時，StorageGRID會分多個步驟執行查找，如下所示：

- 它首先使用低一致性來查找物件。
- 如果查找失敗，它會在下一個一致性值處重複查找，直到達到與強全域行為相當的一致性。

如果 HEAD 或 GET 操作使用「Read-after-new-write」一致性但物件不存在，則物件尋找將始終達到與強全域行為等效的一致性。由於這種一致性要求每個網站都有物件元資料的多個副本，因此如果同一網站上有兩個或多個儲存節點不可用，您可能會收到大量 500 內部伺服器錯誤。

除非您需要類似於 Amazon S3 的一致性保證，否則您可以透過將一致性設為「可用」來防止 HEAD 和 GET 操作出現這些錯誤。當 HEAD 或 GET 操作使用「可用」一致性時，StorageGRID僅提供最終一致性。它不會在增加一致性時重試失敗的操作，因此它不需要提供物件元資料的多個副本。

## 指定 API 操作的一致性

若要為單一 API 操作設定一致性，該操作必須支援一致性值，並且必須在請求標頭中指定一致性。此範例將 GetObject 操作的一致性設定為「Strong-site」。

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



您必須對 PutObject 和 GetObject 操作使用相同的一致性。

## 指定 bucket 的一致性

若要設定儲存桶的一致性，您可以使用StorageGRID"[PUT桶一致性](#)"要求。或者您可以"[改變桶的一致性](#)"來自租戶經理。

設定儲存桶的一致性時，請注意以下事項：

- 設定儲存桶的一致性決定了對儲存桶中的物件或儲存桶配置執行的 S3 操作使用哪種一致性。它不會影響儲存桶本身的操作。
- 單一 API 操作的一致性會覆蓋儲存桶的一致性。
- 一般來說，儲存桶應該使用預設一致性“新寫後讀取”。如果請求無法正常運作，請盡可能變更應用程式用戶端行為。或者，配置客戶端以指定每個 API 請求的一致性。僅在萬不得已的情況下才在儲存桶層級設定一致性。

## 一致性控制和 ILM 規則如何相互作用以影響資料保護

您的一致性選擇和 ILM 規則都會影響物件的保護方式。這些設定可以相互作用。

例如，儲存物件時使用的一致性會影響物件元資料的初始位置，而為 ILM 規則選擇的攝取行為會影響物件副本

的初始位置。由於StorageGRID需要存取物件的元資料及其資料來滿足客戶端請求，因此為一致性和攝取行為選擇相符的保護等級可以提供更好的初始資料保護和更可預測的系統回應。

下列"**攝取選項**"適用於 ILM 規則：

#### 雙重提交

StorageGRID立即製作物件的臨時副本並將成功傳回給客戶端。在可能的情況下，將製作 ILM 規則中指定的副本。

#### 嚴格的

在向客戶端返回成功之前，必須完成 ILM 規則中指定的所有複製。

#### 均衡

StorageGRID嘗試在攝取時製作 ILM 規則中指定的所有副本；如果無法做到這一點，則會製作臨時副本並將成功傳回給用戶端。在可能的情況下，將進行 ILM 規則中指定的複製。

## 一致性和 ILM 規則如何交互的範例

假設您有一個雙站點網格，具有以下 ILM 規則和以下一致性：

- **ILM 規則**：建立兩個物件副本，一個在本機站點，一個在遠端站點。使用嚴格的攝取行為。
- **一致性**：強全域（物件元資料立即分發到所有站點）。

當客戶端將物件儲存到網格時，StorageGRID會複製兩個物件並將元資料分發到兩個站點，然後再將成功返回給客戶端。

在接收成功訊息時，物件將受到完全保護，不會遺失。例如，如果本機站點在攝取後不久遺失，則物件資料和物件元資料的副本仍然存在於遠端站點。該物件完全可檢索。

如果您使用相同的 ILM 規則和強站點一致性，則用戶端可能會在物件資料複製到遠端站點之後但在物件元資料分發到那裡之前收到成功訊息。在這種情況下，物件元資料的保護等級與物件資料的保護等級不符。如果本地站點在攝取後不久遺失，則物件元資料就會遺失。無法檢索該物件。

一致性和 ILM 規則之間的相互關係可能很複雜。如果您需要協助，請聯絡NetApp。

## 物件版本控制

如果您想要保留每個物件的多個版本，您可以設定儲存桶的版本控制狀態。啟用儲存桶的版本控制有助於防止意外刪除對象，並使您能夠檢索和還原對象的早期版本。

StorageGRID系統實現了版本控制，支援大多數功能，但也有一些限制。StorageGRID支援每個物件最多 10,000 個版本。

物件版本控制可以與StorageGRID資訊生命週期管理 (ILM) 或 S3 儲存桶生命週期配置結合。您必須明確啟用每個儲存桶的版本控制。當為儲存桶啟用版本控制時，新增至儲存桶的每個物件都會被指派一個版本 ID，該 ID 由StorageGRID系統產生。

不支援使用 MFA（多因素身份驗證）刪除。



僅可在使用StorageGRID版本 10.3 或更高版本建立的儲存桶上啟用版本控制。

## ILM 和版本控制

ILM 策略適用於物件的每個版本。ILM 掃描過程會持續掃描所有物件並根據目前 ILM 策略重新評估它們。您對 ILM 策略所做的任何變更都將套用於所有先前攝取的物件。如果啟用了版本控制，這包括先前攝取的版本。ILM 掃描將新的 ILM 變更套用至先前擷取的物件。

對於啟用版本控制的儲存桶中的 S3 對象，版本控制支援可讓您建立使用“非目前時間”作為參考時間的 ILM 規則（對於“僅將此規則應用於較舊的對象版本？”問題，選擇“是”["建立 ILM 規則精靈的第 1 步"](#)）。當物件更新時，其先前版本將變為非目前版本。使用「非當前時間」篩選器可讓您建立減少物件先前版本儲存影響的策略。



當您使用分段上傳作業上傳物件的新版本時，物件原始版本的非當前時間反映的是新版本的分段上傳的建立時間，而不是分段上傳的完成時間。在有限的情況下，原始版本的非當前時間可能比目前版本的時間早幾小時或幾天。

### 相關資訊

- ["如何刪除 S3 版本控制對象"](#)
- ["S3 版本化物件的 ILM 規則和策略（範例 4）"](#)。

## 使用 S3 REST API 設定 S3 物件鎖

如果您的StorageGRID系統啟用了全域 S3 物件鎖定設置，則可以建立啟用了 S3 物件鎖定的儲存桶。您可以為每個儲存桶指定預設保留或為每個物件版本指定保留設定。

### 如何為儲存桶啟用 S3 物件鎖定

如果您的StorageGRID系統啟用了全域 S3 物件鎖定設置，則可以在建立每個儲存桶時選擇啟用 S3 物件鎖定。

S3 物件鎖定是一個永久設置，只有在建立儲存桶時才能啟用。建立儲存桶後，您無法新增或停用 S3 物件鎖定。

若要為儲存桶啟用 S3 物件鎖定，請使用下列任一方法：

- 使用租用戶管理器建立儲存桶。看["建立 S3 儲存桶"](#)。
- 使用 CreateBucket 請求建立儲存桶 `x-amz-bucket-object-lock-enabled` 請求標頭。看["對 bucket 的操作"](#)。

S3 物件鎖定需要儲存桶版本控制，該功能在建立儲存桶時自動啟用。您無法暫停儲存桶的版本控制。看["物件版本控制"](#)。

### 儲存桶的預設保留設定

當為儲存桶啟用 S3 物件鎖定时，您可以選擇為儲存桶啟用預設保留，並指定預設保留模式和預設保留期。

#### 預設保留模式

- 在合規模式下：

- 在達到保留截止日期之前，無法刪除該物件。
- 物件的保留截止日期可以增加，但不能減少。
- 在達到該日期之前，不能刪除物件的保留截止日期。
- 在治理模式下：
  - 使用者 `s3:BypassGovernanceRetention` 許可可以使用 `x-amz-bypass-governance-retention: true` 請求標頭以繞過保留設定。
  - 這些用戶可以在達到保留截止日期之前刪除物件版本。
  - 這些使用者可以增加、減少或刪除物件的保留期限。

## 預設保留期

每個儲存桶可以具有以年或天為單位指定的預設保留期。

## 如何設定儲存桶的預設保留

若要設定儲存桶的預設保留，請使用下列任一方法：

- 從租戶管理器管理儲存桶設定。看["建立 S3 儲存桶"](#)和["更新 S3 物件鎖定預設保留"](#)。
- 針對儲存桶發出 PutObjectLockConfiguration 請求，指定預設模式和預設天數或年數。

## PutObjectLock配置

PutObjectLockConfiguration 請求可讓您設定和修改已啟用 S3 物件鎖定的儲存桶的預設保留模式和預設保留期。您也可以刪除先前配置的預設保留設定。

當新的物件版本被提取到儲存桶時，如果 `x-amz-object-lock-mode` 和 `x-amz-object-lock-retain-until-date` 未指定。如果 `x-amz-object-lock-retain-until-date` 未指定。

如果在擷取物件版本後修改了預設保留期，則物件版本的保留截止日期保持不變，並且不會使用新的預設保留期重新計算。

你必須擁有 `s3:PutBucketObjectLockConfiguration` 權限，或成為帳戶 root，即可完成此操作。

這 `Content-MD5` 必須在 PUT 請求中指定請求標頭。

## 請求範例

此範例為儲存桶啟用 S3 物件鎖定，並將預設保留模式設為 COMPLIANCE，將預設保留期設為 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>
```

## 如何決定儲存桶的預設保留時間

若要確定儲存桶是否啟用了 S3 物件鎖定並查看預設保留模式和保留期，請使用下列任一方法：

- 在租用戶管理員中查看儲存桶。看["查看 S3 儲存桶"](#)。
- 發出 `GetObjectLockConfiguration` 請求。

### 取得物件鎖配置

`GetObjectLockConfiguration` 要求可讓您確定是否為儲存桶啟用了 S3 物件鎖定，如果啟用，則檢視是否為儲存桶配置了預設保留模式和保留期。

當新的物件版本被提取到儲存桶時，如果 ``x-amz-object-lock-mode`` 未指定。如果 ``x-amz-object-lock-retain-until-date`` 未指定。

你必須擁有 ``s3:GetBucketObjectLockConfiguration`` 權限，或成為帳戶 root，即可完成此操作。

### 請求範例

```
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
```

#### 回應範例

```
HTTP/1.1 200 OK
x-amz-id-2:
iVmcB7OXXJRkRH1FiVq1151/T24gRfpwpuZrEG11Bb9ImOMAAe98oxSpX1knabA0LTvBYJpSIX
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>
```

## 如何指定物件的保留設定

啟用了 S3 物件鎖的儲存桶可以包含具有和不具有 S3 物件鎖保留設定的物件的組合。

使用 S3 REST API 指定物件級保留設定。物件的保留設定會覆蓋儲存桶的任何預設保留設定。

您可以為每個物件指定以下設定：

- 保留模式：合規或治理。
- **Retain-until-date**：指定StorageGRID必須保留物件版本多久的日期。
  - 在 COMPLIANCE 模式下，如果保留截止日期是將來，則可以檢索對象，但不能修改或刪除它。保留截止日期可以增加，但不能減少或刪除。
  - 在治理模式下，具有特殊權限的使用者可以繞過保留截止日期設定。他們可以在保留期結束之前刪除物

件版本。他們還可以增加、減少甚至刪除保留截止日期。

- 合法保留：對物件版本套用合法保留會立即鎖定該物件。例如，您可能需要對與調查或法律糾紛相關的對象實施法律保留。合法保留沒有到期日，但會一直有效，直到明確取消。

物件的合法保留設定與保留模式和保留截止日期無關。如果物件版本處於合法保留狀態，則任何人都無法刪除該版本。

若要在將物件版本新增至儲存桶時指定 S3 物件鎖定設置，請發出"[放置對象](#)"，"[複製對象](#)"，或者"[建立多部分上傳](#)"要求。

您可以使用以下內容：

- `x-amz-object-lock-mode`，可以是 COMPLIANCE 或 GOVERNANCE（區分大小寫）。



如果您指定 `x-amz-object-lock-mode`，您還必須指定 `x-amz-object-lock-retain-until-date`。

- `x-amz-object-lock-retain-until-date`
  - 保留截止日期值必須採用以下格式 `2020-08-10T21:46:00Z`。允許使用小數秒，但只保留 3 位小數（毫秒精度）。不允許使用其他 ISO 8601 格式。
  - 保留截止日期必須是將來的日期。
- `x-amz-object-lock-legal-hold`

如果合法保留處於開啟狀態（區分大小寫），則該物件將被置於合法保留之下。如果合法保留處於關閉狀態，則不會設定合法保留。任何其他值都會導致 400 Bad Request (InvalidArgument) 錯誤。

如果您使用以下任何請求標頭，請注意以下限制：

- 這 ``Content-MD5`` 請求標頭（如果有）``x-amz-object-lock-*`` 請求標頭存在於 PutObject 請求中。``Content-MD5`` 對於 CopyObject 或 CreateMultipartUpload 來說不是必需的。
- 如果儲存桶未啟用 S3 物件鎖定，且 ``x-amz-object-lock-*`` 請求標頭存在，則傳回 400 錯誤請求 (InvalidRequest) 錯誤。
- PutObject 請求支援使用 ``x-amz-storage-class: REDUCED_REDUNDANCY`` 以匹配 AWS 行為。但是，當將物件提取到啟用了 S3 物件鎖定的儲存桶中時，StorageGRID 將始終執行雙重提交提取。
- 後續的 GET 或 HeadObject 版本回應將包含標頭 `x-amz-object-lock-mode`，`x-amz-object-lock-retain-until-date`，和 `x-amz-object-lock-legal-hold`，如果配置並且請求發送者有正確的 ``s3:Get*`` 權限。

您可以使用 ``s3:object-lock-remaining-retention-days`` 策略條件鍵用於限制物件的最小和最大允許保留期。

## 如何更新物件的保留設定

如果需要更新現有物件版本的合法保留或保留設置，您可以執行下列物件子資源操作：

- PutObjectLegalHold

如果新的合法保留值為 ON，則該物件將被置於合法保留之下。如果合法保留值為 OFF，則解除合法保留。

- PutObjectRetention
  - 模式值可以是 COMPLIANCE 或 GOVERNANCE（區分大小寫）。
  - 保留截止日期值必須採用以下格式 2020-08-10T21:46:00Z。允許使用小數秒，但只保留 3 位小數（毫秒精度）。不允許使用其他 ISO 8601 格式。
  - 如果物件版本具有現有的保留截止日期，則只能增加它。新值必須是未來的值。

## 如何使用治理模式

擁有 `s3:BypassGovernanceRetention` 權限可以繞過使用治理模式的物件的活動保留設定。任何 DELETE 或 PutObjectRetention 操作都必須包含 `x-amz-bypass-governance-retention:true` 請求標頭。這些使用者可以執行以下附加操作：

- 執行 DeleteObject 或 DeleteObjects 操作，在物件版本的保留期結束之前刪除該物件版本。  
處於合法保留狀態的物件無法被刪除。必須關閉合法保留。
- 在物件的保留期結束之前，執行 PutObjectRetention 操作，將物件版本的模式從 GOVERNANCE 變更為 COMPLIANCE。  
絕不允許將模式從合規性改為治理性。
- 執行 PutObjectRetention 操作以增加、減少或刪除物件版本的保留期。

### 相關資訊

- ["使用 S3 對象鎖管理對象"](#)
- ["使用 S3 對象鎖保留對象"](#)
- ["Amazon Simple Storage Service 使用者指南：鎖定對象"](#)

## 建立 S3 生命週期配置

您可以建立 S3 生命週期配置來控制何時從 StorageGRID 系統中刪除特定物件。

本節中的簡單範例說明了 S3 生命週期配置如何控制何時從特定 S3 儲存桶中刪除（過期）某些物件。本節中的範例僅用於說明目的。有關建立 S3 生命週期配置的完整詳細信息，請參閱 ["Amazon Simple Storage Service 使用者指南：物件生命週期管理"](#)。請注意，StorageGRID 僅支援 Expiration 操作；它不支援 Transition 操作。

### 什麼是生命週期配置

生命週期配置是一組應用於特定 S3 儲存桶中的物件的規則。每條規則指定哪些物件會受到影響以及這些物件何時過期（在特定日期或幾天後）。

StorageGRID 在生命週期配置中支援最多 1,000 條生命週期規則。每個規則可以包含以下 XML 元素：

- 過期：從物件被攝取時開始，當達到指定日期或指定天數時刪除物件。
- NoncurrentVersionExpiration：從物件變成非目前版本開始，當達到指定的天數時刪除該物件。
- 過濾器（前綴、標籤）

- 地位
- ID

每個物件都遵循 S3 儲存桶生命週期或 ILM 策略的保留設定。配置 S3 儲存桶生命週期後，生命週期到期作業將覆蓋與儲存桶生命週期過濾器相符的物件的 ILM 策略。與儲存桶生命週期過濾器不符的物件使用 ILM 策略的保留設定。如果物件與儲存桶生命週期過濾器匹配，且未明確指定到期操作，則不使用 ILM 策略的保留設置，這表示物件版本將永遠保留。看"[S3 儲存桶生命週期和 ILM 策略的範例優先級](#)"。

因此，即使 ILM 規則中的放置說明仍然適用於該對象，該對象也可能會從網格中移除。或者，即使物件的任何 ILM 放置指令已經失效，該物件仍可能保留在網格上。有關詳細信息，請參閱"[ILM 如何在物件的整個生命週期中運作](#)"。



儲存桶生命週期配置可與啟用了 S3 物件鎖的儲存桶一起使用，但舊版相容儲存桶不支援儲存桶生命週期配置。

StorageGRID支援使用下列儲存桶操作來管理生命週期配置：

- DeleteBucket生命週期
- 取得BucketLifecycleConfiguration
- PutBucket生命週期配置

## 建立生命週期配置

作為建立生命週期配置的第一步，您需要建立一個包含一個或多個規則的 JSON 檔案。例如，此 JSON 檔案包含三條規則，如下所示：

1. 規則 1 僅適用於與前綴相符的對象 `category1/` 並且有一個 `key2`` 的價值 ``tag2`。這 ``Expiration`` 參數指定與過濾器匹配的物件將於 2020 年 8 月 22 日午夜到期。
2. 規則 2 僅適用於與前綴相符的對象 `category2/`。這 ``Expiration`` 參數指定與篩選器相符的物件將在被攝取後 100 天過期。



指定天數的規則與物件被攝取的時間有關。如果目前日期超過攝取日期加上天數，則套用生命週期配置後，某些物件可能會從儲存桶中刪除。

3. 規則 3 僅適用於與前綴相符的對象 `category3/`。這 ``Expiration`` 參數指定匹配對象的任何非當前版本將在成為非當前版本後 50 天過期。

```

{
  "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"
    }
  ]
}

```

## 將生命週期配置應用於儲存桶

建立生命週期設定檔後，您可以透過發出 `PutBucketLifecycleConfiguration` 請求將其套用到儲存桶。

此請求將範例檔案中的生命週期配置套用至名為 `testbucket`。

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

若要驗證生命週期配置是否已成功套用於儲存桶，請發出 `GetBucketLifecycleConfiguration` 要求。例如：

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

成功的回應列出了您剛剛應用的生命週期配置。

## 驗證儲存桶生命週期到期是否適用於對象

您可以在發出 `PutObject`、`HeadObject` 或 `GetObject` 請求時確定生命週期配置中的過期規則是否適用於特定物件。如果適用規則，則回應包括 `Expiration` 指示物件何時過期以及符合哪個過期規則的參數。



由於 bucket 生命週期覆蓋 ILM，`expiry-date` 顯示的是物件被刪除的實際日期。有關詳細信息，請參閱["如何確定對象保留"](#)。

例如，此 `PutObject` 請求於 2020 年 6 月 22 日發出，並將一個物件放入 `testbucket` 桶。

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

成功回應表示該物件將在 100 天後（2020 年 10 月 1 日）過期，並且符合生命週期配置的規則 2。

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

例如，此 `HeadObject` 請求用於取得 `testbucket` 儲存桶中相同物件的元資料。

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

成功回應包括物件的元數據，並表明該物件將在 100 天後過期，並且符合規則 2。

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



對於啟用版本控制的儲存桶，`x-amz-expiration` 回應頭僅適用於物件的目前版本。

## 實作 S3 REST API 的建議

在實作用於 StorageGRID 的 S3 REST API 時，您應該遵循這些建議。

### 對不存在對象的 HEAD 建議

如果您的應用程式定期檢查某個物件是否存在於您認為該物件實際上不存在的路徑中，則應使用“可用”**一致性**。例如，如果您的應用程式在 PUT 之前 HEAD 某個位置，則您應該使用「可用」一致性。

否則，如果 HEAD 操作未找到該對象，且同一站點上有兩個或多個儲存節點不可用或遠端站點無法訪問，您可能收到大量 500 內部伺服器錯誤。

您可以使用**“PUT 桶一致性”**請求，或者您可以在單一 API 操作的請求標頭中指定一致性。

### 對象鍵的建議

根據儲存桶首次建立的時間，遵循以下物件鍵名稱建議。

在 **StorageGRID 11.4** 或更早版本中建立的儲存桶

- 不要使用隨機值作為物件鍵的前四個字元。這與 AWS 先前對鍵前綴的建議形成了對比。相反，使用非隨機、非唯一的前綴，例如 image。
- 如果您確實遵循先前的 AWS 建議在鍵前綴中使用隨機和唯一字符，請在物件鍵前加上目錄名稱。也就是說，使用這種格式：

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

而不是這種格式：

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

在**StorageGRID 11.4** 或更高版本中建立的儲存桶

不需要限制物件鍵名稱以滿足效能最佳實務。大多數情況下，您可以對物件鍵名稱的前四個字元使用隨機值。



一個例外是 S3 工作負載，它會在短時間後連續刪除所有物件。為了盡量減少此用例對效能的影響，每隔幾千個物件就會用日期之類的内容改變鍵名的前導部分。例如，假設 S3 用戶端通常每秒寫入 2,000 個對象，而 ILM 或儲存桶生命週期策略會在三天後刪除所有物件。為了最大限度地減少對效能的影響，您可以使用以下模式命名鍵：`/mybucket/mydir/yyyymddhhmmss-random_UUID.jpg`

## “範圍讀取”的建議

如果“[壓縮儲存物件的全域選項](#)”啟用後，S3 用戶端應用程式應避免執行指定傳回位元組範圍的 `GetObject` 操作。這些「範圍讀取」操作效率低下，因為 StorageGRID 必須有效地解壓縮物件才能存取請求的位元組。從非常大的物件中請求一小段位元組的 `GetObject` 操作效率特別低；例如，從 50 GB 的壓縮物件中讀取 10 MB 範圍的位元組效率很低。

如果從壓縮物件讀取範圍，客戶端請求可能會逾時。



如果您需要壓縮物件並且客戶端應用程式必須使用範圍讀取，請增加應用程式的讀取逾時。

## 版權資訊

Copyright © 2026 NetApp, Inc. 版權所有。台灣印製。非經版權所有人事先書面同意，不得將本受版權保護文件的任何部分以任何形式或任何方法（圖形、電子或機械）重製，包括影印、錄影、錄音或儲存至電子檢索系統中。

由 NetApp 版權資料衍伸之軟體必須遵守下列授權和免責聲明：

此軟體以 NETAPP「原樣」提供，不含任何明示或暗示的擔保，包括但不限於有關適售性或特定目的適用性之擔保，特此聲明。於任何情況下，就任何已造成或基於任何理論上責任之直接性、間接性、附隨性、特殊性、懲罰性或衍生性損害（包括但不限於替代商品或服務之採購；使用、資料或利潤上的損失；或企業營運中斷），無論是在使用此軟體時以任何方式所產生的契約、嚴格責任或侵權行為（包括疏忽或其他）等方面，NetApp 概不負責，即使已被告知有前述損害存在之可能性亦然。

NetApp 保留隨時變更本文所述之任何產品的權利，恕不另行通知。NetApp 不承擔因使用本文所述之產品而產生的責任或義務，除非明確經過 NetApp 書面同意。使用或購買此產品並不會在依據任何專利權、商標權或任何其他 NetApp 智慧財產權的情況下轉讓授權。

本手冊所述之產品受到一項（含）以上的美國專利、國外專利或申請中專利所保障。

有限權利說明：政府機關的使用、複製或公開揭露須受 DFARS 252.227-7013（2014 年 2 月）和 FAR 52.227-19（2007 年 12 月）中的「技術資料權利 - 非商業項目」條款 (b)(3) 小段所述之限制。

此處所含屬於商業產品和 / 或商業服務（如 FAR 2.101 所定義）的資料均為 NetApp, Inc. 所有。根據本協議提供的所有 NetApp 技術資料和電腦軟體皆屬於商業性質，並且完全由私人出資開發。美國政府對於該資料具有非專屬、非轉讓、非轉授權、全球性、有限且不可撤銷的使用權限，僅限於美國政府為傳輸此資料所訂合約所允許之範圍，並基於履行該合約之目的方可使用。除非本文另有規定，否則未經 NetApp Inc. 事前書面許可，不得逕行使用、揭露、重製、修改、履行或展示該資料。美國政府授予國防部之許可權利，僅適用於 DFARS 條款 252.227-7015(b)（2014 年 2 月）所述權利。

## 商標資訊

NETAPP、NETAPP 標誌及 <http://www.netapp.com/TM> 所列之標章均為 NetApp, Inc. 的商標。文中所涉及的所有其他公司或產品名稱，均為其各自所有者的商標，不得侵犯。