# 如何實作S3 REST API StorageGRID

StorageGRID

NetApp
November 04, 2025

This PDF was generated from https://docs.netapp.com/zh-tw/storagegrid-118/s3/conflicting-client-requests.html on November 04, 2025. Always check docs.netapp.com for the latest.

# 目錄


如何實作S3 REST API StorageGRID 1
衝突的用戶端要求 1
一致性值 1
一致性值 1
使用「讀取後新寫入」和「可用」一致性 2
[[API- 作業 - 一致性控制 ] ] 指定 API 作業的一致性 2
指定貯體的一致性 2
[[how — consistency — controls — and — ILM — rules — arize] 一致性和 ILM 規則如何交互以影響數據保護 2
一致性與 ILM 規則如何互動的範例 3
物件版本管理 3
ILM與版本管理 4
使用 S3 REST API 來設定 S3 物件鎖定 4
如何為貯體啟用 S3 物件鎖定 4
貯體的預設保留設定 4
如何設定貯體的預設保留 5
如何決定貯體的預設保留 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生命週期組態 9
什麼是生命週期組態 9
建立生命週期組態 10
將生命週期組態套用至貯體 12
驗證目標是否適用庫位生命週期到期 12
實作S3 REST API的建議 13
針對不存在物件的使用者提出建議 13
物件金鑰建議 13
「範圍讀取」建議 14

# 如何實作S3 REST API StorageGRID

## 衝突的用戶端要求

衝突的用戶端要求（例如兩個寫入同一個金鑰的用戶端）會以「最新致勝」的方式解決。

「最新致勝」評估的時間取決於StorageGRID 何時由VMware系統完成指定的要求、而非S3用戶端開始作業的時間。

## 一致性值

一致性可在物件的可用度與這些物件在不同儲存節點和站台之間的一致性之間取得平衡。您可以根據應用程式的需求變更一致性。

根據預設StorageGRID 、此功能可確保新建立物件的寫入後讀取一致性。任何「Get」追蹤成功完成的「PUT」、都能讀取新寫入的資料。覆寫現有物件、更新中繼資料及刪除的動作最終一致。覆寫通常需要幾秒鐘或幾分鐘才能傳播、但可能需要15天的時間。

如果您想要以不同的一致性執行物件作業、您可以：

- 指定一致性 每個貯體。
- 指定一致性 每項 API 作業。
- 執行下列其中一項工作、變更預設的全網格一致性：
  - 在 Grid Manager 中、前往 * 組態 * > * 系統 * > * 儲存設定 * > * 預設一致性 * 。
  - 。 。

> 對全網格一致性所做的變更僅適用於變更設定後所建立的貯體。若要判斷變更的詳細資料、請參閱位於的稽核記錄 `/var/local/log` （搜尋 * 一致性層級 * ）。

### 一致性值

一致性會影響 StorageGRID 用來追蹤物件的中繼資料如何在節點之間散佈、進而影響用戶端要求的物件可用度。

您可以將貯體或 API 作業的一致性設定為下列其中一個值：

- * 全部 * ：所有節點都會立即接收資料、否則要求將會失敗。
- **Strong-globall** ：保證所有網站上所有用戶端要求的寫入後讀取一致性。
- **Strong-site** ：保證網站內所有用戶端要求的寫入後讀取一致性。
- * 新寫入後讀取 * ：（預設）提供新物件的寫入後讀取一致性、以及物件更新的最終一致性。提供高可用度與資料保護保證。建議大多數情況下使用。
- * 可用 * ：提供新物件和物件更新的最終一致性。對於 S3 貯體、請僅視需要使用（例如、包含很少讀取的記錄值之貯體、或用於對不存在的金鑰執行 head 或 Get 作業）。S3 FabricPool 儲存區不支援。

## 使用「讀取後新寫入」和「可用」一致性

當標頭或 GET 作業使用「讀取後新寫入」一致性時、 StorageGRID 會以多個步驟執行查詢、如下所示：

- 它會先使用低一致性來查詢物件。
- 如果該查詢失敗、它會在下一個一致性值重複查詢、直到達到等同於 Strong-global 行為的一致性為止。

如果 HEAD 或 GET 作業使用「讀取後新寫入」一致性、但物件不存在、則物件查詢一律會達到等同於 Strong-global 行為的一致性。由於這種一致性需要在每個站台上提供多個物件中繼資料複本、因此如果同一個站台上有兩個或多個儲存節點無法使用、您可能會收到大量 500 個內部伺服器錯誤。

除非您需要與 Amazon S3 類似的一致性保證、否則您可以將一致性設定為「可用」、以防止這些錯誤發生在 HEAD 和 GET 作業中。 當前端或 GET 作業使用「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作業、因此不需要物件中繼資料的多個複本。

## [[API- 作業 - 一致性控制 ] ] 指定 API 作業的一致性

若要設定個別 API 作業的一致性、作業必須支援一致性值、而且您必須在要求標頭中指定一致性。此範例會將 GetObject 作業的一致性設定為「 Strong-site 」。

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

PutObject 和 GetObject 作業必須使用相同的一致性。

### 指定貯體的一致性

若要設定貯體的一致性、您可以使用 StorageGRID "實現庫位一致性" 申請。您也可以 "改變貯體的一致性" 從租戶管理器。

設定貯體的一致性時、請注意下列事項：

- 設定貯體的一致性、可決定用於在貯體或貯體組態中的物件上執行 S3 作業的一致性。它不會影響儲存庫本身的作業。
- 個別 API 作業的一致性會覆寫貯體的一致性。
- 一般而言、貯體應使用預設的一致性「讀取後新寫入」。 如果要求無法正常運作、請盡可能變更應用程式用戶端行為。或者、設定用戶端以指定每個 API 要求的一致性。將貯體層級的一致性設為最後的方法。

## [[how － consistency － controls － and － ILM － rules － arize] 一致性和 ILM 規則如何交互以影響數據保護

您選擇的一致性和 ILM 規則都會影響物件的保護方式。這些設定可以互動。

例如、儲存物件時所使用的一致性會影響物件中繼資料的初始放置位置、而為 ILM 規則選取的擷取行為則會影

響物件複本的初始放置位置。由於 StorageGRID 需要同時存取物件的中繼資料及其資料、才能滿足用戶端要求、因此針對一致性和擷取行為選擇符合的保護層級、可以提供更好的初始資料保護、以及更可預測的系統回應。

以下內容 "擷取選項" 適用於 ILM 規則：

雙重承諾

StorageGRID 會立即製作物件的臨時複本、並將成功傳回給用戶端。在ILM規則中指定的複本會盡可能製作。

嚴格

在 ILM 規則中指定的所有複本都必須在成功傳回用戶端之前製作。

平衡

StorageGRID 會嘗試在擷取時製作 ILM 規則中指定的所有複本；如果不可能、則會製作過渡複本、並將成功傳回用戶端。ILM規則中指定的複本會盡可能製作。

### 一致性與 **ILM** 規則如何互動的範例

假設您有一個雙站台網格、其中包含下列 ILM 規則及下列一致性：

- * ILM規則*：建立兩個物件複本、一個在本機站台、一個在遠端站台。使用嚴格的擷取行為。
- * 一致性 * ：強式全域（物件中繼資料會立即發佈至所有站台）。

當用戶端將物件儲存到網格時、StorageGRID 在成功傳回用戶端之前、功能區會同時複製物件並將中繼資料散佈到兩個站台。

在擷取最成功的訊息時、物件會受到完整保護、不會遺失。例如、如果在擷取後不久即遺失本機站台、則物件資料和物件中繼資料的複本仍存在於遠端站台。物件可完全擷取。

如果您改用相同的 ILM 規則和強大的站台一致性、則在物件資料複寫到遠端站台、但在物件中繼資料散佈到該站台之前、用戶端可能會收到成功訊息。在此情況下、物件中繼資料的保護層級與物件資料的保護層級不符。如果在擷取後不久本機站台便會遺失、則物件中繼資料將會遺失。無法擷取物件。

一致性與 ILM 規則之間的相互關係可能很複雜。如需協助、請聯絡 NetApp 。

# 物件版本管理

## 如果您要保留每個物件的多個版本、可以設定貯體的版本設定狀態。啟用貯體版本設定有助於防止意外刪除物件、並可讓您擷取及還原物件的舊版。

支援大部分功能的支援功能、以及部分限制、可讓整個系統執行版本管理。StorageGRID支援多達1、000個版本的每個物件。StorageGRID

物件版本管理可與StorageGRID 資訊的生命週期管理（ILM）或S3生命週期組態結合使用。您必須明確啟用每個儲存區的版本設定。當貯體啟用版本設定時、新增至貯體的每個物件都會指派版本 ID 、由 StorageGRID 系統產生。

不支援使用MFA（多因素驗證）刪除。

版本管理只能在StorageGRID 以不含更新版本的版本資訊版本10.3所建立的儲存庫上啟用。

### ILM與版本管理

ILM原則會套用至物件的每個版本。ILM掃描程序會持續掃描所有物件、並根據目前的ILM原則重新評估這些物件。您對ILM原則所做的任何變更、都會套用至所有先前擷取的物件。如果啟用版本管理、則包括先前擷取的版本。ILM掃描會將新的ILM變更套用至先前擷取的物件。

對於啟用版本設定的儲存區中的 S3 物件、版本設定支援可讓您建立使用「非目前時間」做為參考時間的 ILM 規則（針對「僅將此規則套用至舊版物件？」這個問題、請選取 * 是 * 。 在中 "建立 ILM 規則精靈的步驟 1"）。更新物件時、其舊版本會變成非最新版本。使用「非目前時間」篩選器可讓您建立原則、以減少舊版物件的儲存影響。

當您使用多部分上傳作業上傳物件的新版本時、原始版本物件的非目前時間會反映新版本的多部分上傳時間、而非多部分上傳完成時。在有限的情況下、原始版本的非目前時間可能比目前版本的時間早上幾小時或幾天。

相關資訊

- "如何刪除S3版本控制物件"
- "S3版本化物件的ILM規則和原則（範例4）"。

# 使用 S3 REST API 來設定 S3 物件鎖定

如果 StorageGRID 系統已啟用全域 S3 物件鎖定設定、您可以在啟用 S3 物件鎖定的情況下建立儲存區。您可以針對每個物件版本、指定每個儲存區或保留設定的預設保留。

## 如何為貯體啟用 S3 物件鎖定

如果StorageGRID 您的整個S3物件鎖定設定已啟用、則您可以在建立每個儲存區時、選擇性地啟用S3物件鎖定。

S3 物件鎖定是永久性設定、只有在建立貯體時才能啟用。建立貯體後、您無法新增或停用 S3 物件鎖定。

若要為貯體啟用 S3 物件鎖定、請使用下列其中一種方法：

- 使用租戶管理程式建立桶。請參閱 "建立S3儲存區"。
- 使用的 CreateBucket 要求建立貯體 `x-amz-bucket-object-lock-enabled` 要求標頭：請參閱 "在貯體上作業"。

S3 物件鎖定需要儲存區版本設定、此功能會在建立儲存區時自動啟用。您無法暫停儲存區的版本設定。請參閱 "物件版本管理"。

## 貯體的預設保留設定

為貯體啟用 S3 物件鎖定時、您可以選擇性地啟用貯體的預設保留、並指定預設保留模式和預設保留期間。

預設保留模式

- 在法規遵循模式中：
  - 直到達到物件的保留日期、才能刪除物件。
  - 物件的保留日期可以增加、但不能減少。
  - 直到達到該日期為止、才能移除物件的保留日期。
- 在治理模式中：
  - 的使用者 `s3:BypassGovernanceRetention` 權限可以使用 `x-amz-bypass-governance-retention: true` 要求標頭略過保留設定。
  - 這些使用者可以在達到物件版本的保留截止日期之前刪除物件版本。
  - 這些使用者可以增加、減少或移除物件的保留到目前為止。

預設保留期間

每個貯體都可以有一段以年或日為單位指定的預設保留期間。

## 如何設定貯體的預設保留

若要設定貯體的預設保留、請使用下列其中一種方法：

- 從 Tenant Manager 管理貯體設定。請參閱 "建立S3儲存區" 和 "更新 S3 物件鎖定預設保留"。
- 針對貯體發出 PutObjectLockConfiguration 要求、以指定預設模式和預設天數或年數。

### PutObjectLockConfiguration

PutObjectLockConfiguration 要求可讓您設定及修改已啟用 S3 物件鎖定的儲存區的預設保留模式和預設保留期間。您也可以移除先前設定的預設保留設定。

將新物件版本擷取至貯體時、會套用預設保留模式 `x-amz-object-lock-mode` 和 `x-amz-object-lock-retain-until-date` 未指定。預設保留期間用於計算截止日期 IF `x-amz-object-lock-retain-until-date` 未指定。

如果在擷取物件版本之後修改預設保留期間、則物件版本的保留截止日期將維持不變、且不會使用新的預設保留期間重新計算。

您必須擁有 `s3:PutBucketObjectLockConfiguration` 完成此作業的權限、或是帳戶根目錄。

。 `Content-MD5` 必須在 PUT 要求中指定要求標頭。

#### 申請範例

此範例可為貯體啟用 S3 物件鎖定、並將預設保留模式設為符合法規、並將預設保留期間設為 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>
```

## 如何決定貯體的預設保留

若要判斷儲存區是否啟用 S3 物件鎖定、並查看預設保留模式和保留期間、請使用下列其中一種方法：

- 在租戶管理器中檢視貯體。請參閱 "檢視 S3 儲存區"。
- 發出 GetObjectLockConfiguration 要求。

### GetObjectLockConfiguration

GetObjectLockConfiguration 要求可讓您判斷儲存區是否已啟用 S3 物件鎖定、如果已啟用、請查看儲存區是否已設定預設保留模式和保留期間。

將新物件版本擷取至貯體時、會套用預設保留模式 `x-amz-object-lock-mode` 未指定。預設保留期間用於計算截止日期 IF `x-amz-object-lock-retain-until-date` 未指定。

您必須擁有 `s3:GetBucketObjectLockConfiguration` 完成此作業的權限、或是帳戶根目錄。

申請範例

```
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/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>
```

## 如何指定物件的保留設定

啟用 S3 物件鎖定的貯體可包含物件組合、並具有或不含 S3 物件鎖定保留設定。

物件層級保留設定是使用 S3 REST API 指定的。物件的保留設定會覆寫貯體的任何預設保留設定。

您可以為每個物件指定下列設定：

- * 保留模式 * ：法規遵循或治理。
- * 截止日期 * ：指定 StorageGRID 必須保留物件版本多久的日期。
  - 在規範模式中、如果保留截止日期是未來、則可以擷取物件、但無法修改或刪除物件。保留截止日期可以增加、但無法減少或移除此日期。
  - 在治理模式中、具有特殊權限的使用者可以略過保留到最新的設定。他們可以在物件版本的保留期間結

束之前刪除物件版本。他們也可以增加、減少或甚至移除截止日期的保留。

- 合法持有：將合法持有套用至物件版本、會立即鎖定該物件。例如、您可能需要對與調查或法律爭議相關的物件保留法律。合法持有沒有到期日、但在明確移除之前、仍會保留到位。

  物件的合法保留設定不受保留模式和保留截止日期的影響。如果物件版本處於合法保留狀態、則沒有人可以刪除該版本。

若要在將物件版本新增至貯體時指定 S3 物件鎖定設定、請發出 "PuttObject"、 "CopyObject"或 "建立多個部分上傳" 申請。

您可以使用下列項目：

- `x-amz-object-lock-mode`，可以是法規遵循或治理（區分大小寫）。

  > 如果您指定 `x-amz-object-lock-mode`，您也必須指定 `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格式 `2020-08-10T21:46:00Z`。允許分數秒、但只保留3個小數位數（毫秒精度）。不允許其他 ISO 8601 格式。
  - 保留截止日期必須為未來日期。
- `x-amz-object-lock-legal-hold`

  如果已開啟合法持有（區分大小寫）、則物件將置於合法持有之下。如果法律保留已關閉、則不會保留任何合法的保留。任何其他值都會導致400個錯誤要求（InvalidArgument）錯誤。

如果您使用上述任一要求標頭、請注意下列限制：

- 。 `Content-MD5` 如有任何要求、則要求標頭為必填欄位 `x-amz-object-lock-*` PuttObject 要求中存在要求標頭。 `Content-MD5` 複製物件或 CreateMultpartUpload 不需要。
- 如果儲存區未啟用S3物件鎖定和 `x-amz-object-lock-*` 出現要求標頭、傳回400個錯誤要求（InvalidRequest）錯誤。
- PutObject 要求支援使用 `x-amz-storage-class: REDUCED_REDUNDANCY` 以符合AWS行為。然而、當物件被擷取至啟用S3物件鎖定的儲存區時StorageGRID 、則會一律執行雙重認可擷取。
- 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`（如果已設定）以及要求傳送者是否正確 `s3:Get*` 權限：

您可以使用 `s3:object-lock-remaining-retention-days` 原則條件金鑰、可限制物件的最小和最大允許保留期間。

## 如何更新物件的保留設定

如果您需要更新現有物件版本的合法保留或保留設定、可以執行下列物件子資源作業：

- `PutObjectLegalHold`

如果新的合法持有值已開啟、則物件將置於合法持有之下。如果合法持有值為「關」、則合法持有將被解除。

- `PutObjectRetention`
  - 模式值可以是法規遵循或治理（區分大小寫）。
  - 保留截止日期值必須採用格式 `2020-08-10T21:46:00Z`。允許分數秒、但只保留3個小數位數（毫秒精度）。不允許其他 ISO 8601 格式。
  - 如果物件版本有現有的截至日期保留、您只能增加。新的價值必須是未來的價值。

### 如何使用治理模式

擁有的使用者 `s3:BypassGovernanceRetention` 權限可以略過使用治理模式之物件的作用中保留設定。任何刪除或 PutObjectRetention 作業都必須包含 `x-amz-bypass-governance-retention:true` 要求標頭：這些使用者可以執行這些額外作業：

- 執行 DeleteObject 或 DeleteObjects 作業、以在物件版本的保留期間結束之前刪除物件版本。

  合法持有的物件無法刪除。合法持有必須關閉。

- 在物件的保留期間結束之前、執行 PutObjectRetention 作業、將物件版本的模式從治理模式變更為法規遵循模式。

  永遠不允許將模式從法規遵循變更為治理。

- 執行 PutObjectRetention 作業以增加、減少或移除物件版本的保留期間。

相關資訊

- "使用S3物件鎖定來管理物件"
- "使用 S3 物件鎖定來保留物件"
- "Amazon簡易儲存服務使用者指南：使用S3物件鎖定"

# 建立S3生命週期組態

## 您可以建立S3生命週期組態、以控制何時從StorageGRID 作業系統刪除特定物件。

本節的簡單範例說明S3生命週期組態如何控制從特定S3儲存區刪除（過期）特定物件的時間。本節範例僅供說明用途。如需建立S3生命週期組態的完整詳細資料、請參閱 "Amazon Simple Storage Service 使用者指南：物件生命週期管理"。請注意StorageGRID 、僅支援過期行動、不支援轉換行動。

### 什麼是生命週期組態

生命週期組態是套用至特定S3儲存區中物件的一組規則。每個規則都會指定受影響的物件、以及這些物件何時到期（在特定日期或幾天之後）。

在生命週期組態中、支援多達1、000個生命週期規則。StorageGRID每個規則可包含下列XML元素：

- 過期：在達到指定日期或達到指定天數時刪除物件、從擷取物件開始算起。
- 非目前版本過期：在達到指定天數時刪除物件、從物件變成非目前的開始算起。

- 篩選器（前置、標記）
- 狀態
- ID

每個物件都遵循 S3 儲存區生命週期或 ILM 原則的保留設定。當 S3 貯體生命週期設定完成時、生命週期到期動作會覆寫符合貯體生命週期篩選器之物件的 ILM 原則。不符合貯體生命週期篩選器的物件、會使用 ILM 原則的保留設定。如果物件符合貯體生命週期篩選器、且未明確指定到期動作、則不會使用 ILM 原則的保留設定、而且會暗示物件版本會永遠保留。請參閱 "S3 儲存區生命週期和 ILM 原則的優先順序範例"。

因此、即使ILM規則中的放置指示仍套用至物件、也可能從網格中移除物件。或者、即使物件的任何ILM放置指示失效、物件仍可能保留在網格上。如需詳細資訊、請參閱 "ILM在物件生命週期內的運作方式"。

庫位生命週期組態可搭配已啟用S3物件鎖定的庫位使用、但庫位生命週期組態不支援舊型符合標準的庫位。

支援使用下列庫位作業來管理生命週期組態：StorageGRID

- 刪除 BucketLifecycle
- GetBucketLifecycleConfiguration
- PuttBucketLifecycleConfiguration

## 建立生命週期組態

建立生命週期組態的第一步、就是建立一個包含一或多個規則的Json檔案。例如、此Json檔案包含三個規則、如下所示：

1. 規則1僅適用於符合前置碼的物件 `category1`/而且有 `key2` 的價值 `tag2`。。 `Expiration` 參數指定符合篩選條件的物件將於2020年8月22日午夜到期。
2. 規則 2 僅適用於符合前置碼的物件 `category2/`。。 `Expiration` 參數指定符合篩選條件的物件在擷取後100天過期。

   指定天數的規則是相對於擷取物件的時間。如果目前日期超過擷取日期加上天數、則在套用生命週期組態後、部分物件可能會立即從儲存庫中移除。

3. 規則 3 僅適用於符合前置碼的物件 `category3/`。。 `Expiration` 參數指定任何非目前版本的相符物件在變成非目前物件50天後過期。

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

## 將生命週期組態套用至貯體

建立生命週期組態檔案之後、您可以發出 PuttBucketLifecycleConfiguration 要求、將其套用至儲存庫。

此要求會將範例檔案中的生命週期組態套用至名為的儲存區中的物件 `testbucket`。

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

若要驗證已成功將生命週期組態套用至貯體、請發出 GetBucketLifecycleConfiguration 要求。例如：

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

成功的回應會列出您剛套用的生命週期組態。

## 驗證目標是否適用庫位生命週期到期

您可以在發出 PutObject 、 HeadObject 或 GetObject 要求時、判斷生命週期組態中的到期規則是否適用於特定物件。如果適用規則、回應會包含 `Expiration` 指出物件到期時間及符合到期規則的參數。

> 因為儲存區生命週期會取代ILM `expiry-date` 顯示的是物件刪除的實際日期。如需詳細資訊、請參閱 "如何判斷物件保留"。

例如、本 PutObject 申請於 2020 年 6 月 22 日發出、並在中放置物件 `testbucket` 鏟斗。

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

成功回應表示物件將在100天（2020年10月1日）後過期、且符合生命週期組態的規則2。

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

例如、此 HeadObject 要求用於取得 testBucket 儲存區中相同物件的中繼資料。

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

成功回應包括物件的中繼資料、指出物件將在100天內過期、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區、 `x-amz-expiration` 回應標頭僅適用於物件的目前版本。

# 實作S3 REST API的建議

實作S3 REST API以搭配StorageGRID 使用時、請遵循以下建議。

## 針對不存在物件的使用者提出建議

如果您的應用程式會定期檢查某個物件是否存在於您不希望該物件實際存在的路徑、您應該使用「可用」 "一致性"。例如、如果您的應用程式在放置前先找到某個位置、則應使用「可用」一致性。

否則、如果前端作業找不到物件、如果同一站台上有兩個或多個儲存節點無法使用、或遠端站台無法連線、您可能會收到大量 500 個內部伺服器錯誤。

您可以使用設定每個貯體的「可用」一致性 "實現庫位一致性" 或者、您也可以在個別 API 作業的要求標頭中指定一致性。

## 物件金鑰建議

請根據第一次建立貯體的時間、遵循下列物件金鑰名稱建議。

在 **StorageGRID 11.4** 或更早版本中建立的貯體

- 請勿使用隨機值做為物件金鑰的前四個字元。這與前AWS關於金鑰前置碼的建議不同。請改用非隨機、非唯一的前置字元、例如 `image`。
- 如果您遵循前 AWS 的建議、在金鑰首碼中使用隨機和唯一字元、請在物件金鑰前加上目錄名稱。也就是使用此格式：

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

而非此格式：

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

在 **StorageGRID 11.4** 或更新版本中建立的貯體

不需要限制物件金鑰名稱以符合效能最佳實務做法。在大多數情況下、您可以對物件金鑰名稱的前四個字元使用隨機值。

S3 工作負載的例外情況是、它會在一段短時間後持續移除所有物件。為了將此使用案例的效能影響降至最低、請將金鑰名稱的前置部分變更為每數千個物件、例如日期。例如、假設 S3 用戶端通常每秒寫入 2 、 000 個物件、而 ILM 或儲存庫生命週期原則則會在三天後移除所有物件。若要將效能影響降至最低、您可以使用如下模式命名金鑰： `/mybucket/mydir/yyyymmddhhmmss-random_UUID.jpg`

## 「範圍讀取」建議

如果是 "用於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 GetObject 作業、以指定要傳回的位元組範圍。這些「範圍讀取」作業效率不彰、因為 StorageGRID 必須有效地解壓縮物件以存取要求的位元組。從非常大的物件要求少量位元組的 GetObject 作業尤其缺乏效率、例如從 50 GB 壓縮物件讀取 10 MB 範圍是效率不彰的。

如果從壓縮物件讀取範圍、用戶端要求可能會逾時。

如果您需要壓縮物件、而用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 © 2025 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。