



StorageGRID 如何實作 S3 REST API

StorageGRID software

NetApp
July 23, 2026

目錄

StorageGRID 如何實作 S3 REST API	1
StorageGRID S3 REST API 如何解決衝突的用戶端請求	1
StorageGRID S3 REST API 如何平衡可用性和一致性	1
一致性值	1
使用「新寫後讀取」和「可用」一致性	2
指定 API 操作的一致性	2
指定儲存桶的一致性	3
一致性與 ILM 規則如何互動以影響資料保護	3
一致性規則和 ILM 規則如何互動的範例	3
StorageGRID 如何使用物件版本控制	4
ILM 和版本控制	4
使用 S3 REST API 設定 StorageGRID S3 Object Lock	4
如何為儲存桶啟用 S3 物件鎖定	5
儲存桶的預設保留設定	5
如何為儲存桶設定預設保留期間	5
如何確定儲存桶的預設保留期間	6
如何為物件指定保留設定	8
如何更新物件的保留設定	9
如何使用 GOVERNANCE 模式	9
了解 StorageGRID 的 S3 生命週期組態	10
生命週期組態是什麼？	10
建立生命週期組態	10
將生命週期組態套用至儲存桶	13
驗證儲存貯體生命週期到期是否適用於物件	13
使用 StorageGRID 實作 S3 REST API 的建議	14
針對不存在物件的 HEAD 建議	14
物件鍵的建議	14
關於「範圍讀取」的建議	15

StorageGRID 如何實作 S3 REST API

StorageGRID S3 REST API 如何解決衝突的用戶端請求

如果用戶端請求發生衝突，例如兩個用戶端寫入同一個金鑰，則依照「最新優先」的原則進行解決。

「最新獲勝」評估的時間是根據 StorageGRID 系統完成給定請求的時間，而不是根據 S3 用戶端開始操作的時間。

StorageGRID S3 REST API 如何平衡可用性和一致性

一致性機制在物件的可用度和這些物件在不同儲存節點和站點間的一致性之間取得了平衡。您可以根據應用程式的需要變更一致性設定。

預設情況下，StorageGRID 保證新建立物件的寫入後讀取一致性。任何在 PUT 操作成功完成後執行的 GET 操作都可以讀取新寫入的資料。對現有物件的覆寫、中繼資料更新和刪除操作則最終一致。

如果要以不同的一致性執行物件操作，可以：

- 指定 **每個儲存桶** 的一致性。
- 指定 **每個 API 作業** 的一致性。
- 若要變更預設的網格級一致性，請執行下列操作之一：
 - 在 Grid Manager 中，前往 **Configuration > System > Storage settings > Default bucket consistency**。
 - 。



對網格整體一致性的變更僅適用於變更設定後所建立的儲存桶。若要了解變更的詳細資訊，請參閱位於 `/var/local/log` 的稽核日誌（搜尋 **consistencyLevel**）。

一致性值

一致性會影響 StorageGRID 用於追蹤物件的中繼資料在節點間的分佈方式。一致性會影響用戶端請求對物件的可用度。

您可以將儲存桶或 API 作業的一致性設定為下列值之一：

- **全部**：所有節點立即接收物件中繼資料，否則要求將失敗。
- **Strong-global**：保證所有站台上所有用戶端請求的寫入後讀取一致性。設定 Quorum 語意後，將套用以下行為：
 - 當網格具有三個或更多站台時，允許用戶端要求具有站台故障容許度。雙站台網格將不具備站台故障容許度。
 - 如果其中一個站台停機，以下 S3 作業將無法成功：

- DeleteBucketEncryption
- PutBucketBranch
- PutBucketEncryption
- PutBucketVersioning
- PutObjectLegalHold
- PutObjectLockConfiguration
- PutObjectRetention

如果需要，您可以 "`${post_edited_translations.segment}`"。

- **Strong-site**：物件中繼資料會立即分發至站台內的其他節點。保證站台內所有用戶端請求的寫後讀一致性。
- **Read-after-new-write**：為新物件提供寫入後讀取一致性，並為物件更新提供最終一致性。提供高可用度和資料保護保證。建議在大多數情況下使用。
- **Available**：為新物件和物件更新提供最終一致性。對於 S3 儲存貯體，請僅在需要時使用（例如，對於包含極少讀取之記錄值的儲存貯體，或針對不存在的金鑰進行 HEAD 或 GET 作業）。不支援 S3 FabricPool 儲存貯體。

使用「新寫後讀取」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詢，如下所示：

- 它首先使用低一致性查詢物件。
- 如果查詢失敗，則會在下一個一致性值處重複查詢，直到達到與 strong-global 的行為等效的一致性。

如果 HEAD 或 GET 操作使用「新寫後讀取」一致性，但物件不存在，則物件查找將一律達到與 strong-global 行為相同的一致性。由於此一致性要求每個站台都必須有物件中繼資料的多個複本，因此如果同一站台上有多個或多個儲存節點無法使用，您可能會收到大量 500 Internal Server 錯誤。

除非您需要類似 Amazon S3 的一致性保證，否則可以透過將一致性設為「Available」來避免 HEAD 和 GET 操作出現這些錯誤。當 HEAD 或 GET 操作使用「Available」一致性時，StorageGRID 僅提供最終一致性。它不會以提高一致性的方式重試失敗的操作，因此不需要提供多個物件中繼資料複本。

指定 API 操作的一致性

若要為單一 API 操作設定一致性，該操作必須支援對應的一致性值，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
```



對於 PutObject 和 GetObject 操作，必須使用相同的一致性。

指定儲存桶的一致性

若要設定儲存桶的一致性，可以使用 StorageGRID "PUT Bucket 一致性" 請求。或者，也可以"變更儲存桶的一致性"從 Tenant Manager 進行設定。

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，可決定對儲存桶中的物件或儲存桶組態執行的 S3 操作所使用的一致性。此設定不會影響對儲存桶本身的操作。
- 單一 API 操作的一致性會置換儲存桶的一致性。
- 通常情況下，儲存桶應使用預設的一致性原則「新寫入後讀取」。如果請求無法正常運作，請盡可能變更應用程式用戶端的行為。或者，設定用戶端為每個 API 請求指定一致性原則。只有在萬不得已的情況下才在儲存桶層級設定一致性原則。

一致性與 ILM 規則如何互動以影響資料保護

您選擇的一致性設定和 ILM 規則都會影響物件的保護方式。這些設定可能會互動。

例如，物件儲存時採用的一致性策略會影響物件中繼資料的初始位置，而為 ILM 規則選擇的擷取行為則會影響物件副本的初始位置。由於 StorageGRID 需要存取物件的中繼資料和資料才能滿足用戶端請求，因此為一致性策略和擷取行為選擇相符的保護層級可以提供更好的初始資料保護和更可預測的系統回應。

以下"擷取選項"適用於 ILM 規則：

雙重承諾

StorageGRID 會立即建立物件的臨時副本，並向用戶端傳回成功。ILM 規則中指定的副本會在可能的情況下建立。

嚴格

在向用戶端返回成功之前，必須完成 ILM 規則中指定的所有副本。

均衡

StorageGRID 會在擷取時嘗試建立 ILM 規則中指定的所有副本；如果無法全部建立，則會建立臨時副本並向用戶端傳回成功資訊。ILM 規則中指定的副本會在條件允許的情況下建立。

一致性規則和 ILM 規則如何互動的範例

假設您有一個三站台網格，具有以下 ILM 規則和以下一致性：

- **ILM 規則：**建立三個物件複本，一個在本機站台，每個遠端站台各一個。使用 Strict 擷取行為。
- **一致性：**強全域性（物件中繼資料立即分發至多個站台）。

當用戶端將物件儲存到網格時，StorageGRID 會建立所有三個物件複本，並將中繼資料分發到多個站台，然後向用戶端傳回成功。

在收到擷取成功訊息時，該物件已受到完整保護，可防止遺失。例如，如果本機站台在擷取後不久發生故障，物件資料和物件中繼資料的複本仍會存在於遠端站台。您可以從其他站台完整擷取該物件。

如果您改為使用相同的 ILM 規則和強站台一致性，則用戶端可能會在物件資料複寫到遠端站台之後，但在物件中繼資料發配到該處之前，收到成功訊息。在這種情況下，物件中繼資料的保護層級與物件資料的保護層級不

符。如果在本機站台擷取後不久即遺失，則物件中繼資料將會遺失。該物件將無法擷取。

一致性與 ILM 規則之間的相互關係可能非常複雜。如果您需要協助，請聯絡 NetApp。

StorageGRID 如何使用物件版本控制

如果您希望保留每個物件的多個版本，可以設定儲存貯體的版本控制狀態。啟用儲存貯體的版本控制有助於防止意外刪除物件，並可讓您擷取和還原物件的早期版本。

StorageGRID 系統實作版本控制，支援大多數功能，但有一些限制。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 儲存桶生命週期組態結合使用。您必須為每個儲存桶明確啟用版本控制。啟用儲存桶的版本控制後，新增至該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。

使用 MFA (多因素驗證) 時，不支援刪除功能。



版本控制只能在以 StorageGRID 版本 10.3 或更新版本建立的儲存桶上啟用。

ILM 和版本控制

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 原則重新評估它們。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。如果啟用了版本控制，則包含先前擷取的版本。ILM 掃描會將新的 ILM 變更套用至先前擷取的物件。

對於已啟用版本控制的 S3 儲存貯體中的物件，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 (在 ["建立 ILM 規則精靈的步驟 1"](#) 中，針對「僅將此規則套用至舊版物件？」此問題選擇*是*)。當物件更新時，其先前版本將變為非目前版本。使用「非目前時間」篩選器，您可以建立原則來減少物件先前版本對儲存設備的影響。



當您使用分段上傳作業上傳物件的新版本時，原始物件的非目前時間反映的是建立新版本分段上傳的時間，而不是分段上傳完成的時間。在少數情況下，原始版本的非目前時間可能比目前版本的時間早幾個小時甚至幾天。

相關資訊

- ["如何刪除 S3 版本化物件"](#)
- ["S3 版本化物件的 ILM 規則和原則 \(範例 4\)"](#)。

使用 S3 REST API 設定 StorageGRID S3 Object Lock

如果您的 StorageGRID 系統已啟用全域 S3 Object Lock 設定，則可以建立已啟用 S3 Object Lock 的儲存桶。您可以為每個儲存桶指定預設保留，也可以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 系統已啟用全域 S3 物件鎖定設定，您可以在建立每個儲存桶時選擇性地啟用 S3 物件鎖定。

S3 Object Lock 是一項永久性設定，只能在建立儲存桶時啟用。儲存桶建立後，無法新增或停用 S3 Object Lock。

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。參見["建立 S3 儲存桶"](#)。
- 使用含有 `x-amz-bucket-object-lock-enabled` 請求標頭的 `CreateBucket` 請求建立儲存桶。請參閱 ["儲存貯體上的作業"](#)。

S3 Object Lock 需要儲存桶版本控制，該功能在建立儲存桶時會自動啟用。您無法暫停儲存桶的版本控制。請參閱["物件版本控制"](#)。

儲存桶的預設保留設定

當為儲存桶啟用 S3 Object Lock 時，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間。

預設保留模式

- 處於合規模式：
 - 物件只有在達到其保留期限後才能刪除。
 - `${post_edited_translations.segment}`
 - `${post_edited_translations.segment}`
- 處於 GOVERNANCE 模式：
 - 擁有 ``s3:BypassGovernanceRetention`` 權限的使用者可以使用 ``x-amz-bypass-governance-retention: true`` 請求標頭繞過保留設定。
 - `${post_edited_translations.segment}`
 - 這些使用者可以增加、減少或移除物件的保留截止日期。

預設保留期間

每個儲存桶都可以設定預設保留期間，以年或天為單位。

如何為儲存桶設定預設保留期間

若要設定儲存桶的預設保留期限，可以使用下列任一方法：

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

PutObjectLockConfiguration

此 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間。您也可以移除先前設定的預設保留設定。

當新的物件版本被擷取至儲存桶時，如果未指定 `x-amz-object-lock-mode` 和 `x-amz-object-lock-retain-until-date`，則套用預設保留模式。如果未指定 `x-amz-object-lock-retain-until-date`，則使用預設保留期間來計算保留截止日期。

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

您必須擁有 `s3:PutBucketObjectLockConfiguration` 權限，或具備帳戶根目錄存取權，才能完成此操作。

The Content-MD5 請求標頭必須在 PUT 請求中指定。

請求範例

此範例為儲存桶啟用 S3 Object Lock，並將預設保留模式設為 COMPLIANCE，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>
```

如何確定儲存桶的預設保留期間

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

- 在租戶管理員中查看儲存桶。請參閱[查看 S3 儲存桶](#)。
- 發出 GetObjectLockConfiguration 請求。

GetObjectLockConfiguration

GetObjectLockConfiguration 請求可讓您確定是否為儲存桶啟用了 S3 物件鎖定，如果已啟用，則查看是否為該儲存桶配置了預設保留模式和保留期間。

當新的物件版本擷取至儲存桶時，如果未指定 `x-amz-object-lock-mode`，則套用預設保留模式。如果未指定 `x-amz-object-lock-retain-until-date`，則使用預設保留期間來計算保留截止日期。

您必須擁有 `s3:GetBucketObjectLockConfiguration` 權限，或具備帳戶根目錄存取權，才能完成此操作。

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/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>
```

如何為物件指定保留設定

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

物件層級的保留設定透過 S3 REST API 指定。物件的保留設定會置換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：COMPLIANCE 或 GOVERNANCE。
- **Retain-until-date**：指定 StorageGRID 必須保留物件版本多久的日期。
 - 在 COMPLIANCE 模式下，如果保留截止日期在未來，則可以擷取該物件，但無法修改或刪除。保留截止日期可以延長，但不能縮短或移除。
 - 在 GOVERNANCE 模式下，擁有特殊權限的使用者可以繞過保留期限設定。他們可以在物件版本的保留期間結束前將其刪除。他們還可以增加、減少甚至移除保留期限。
- 合法持有：對物件版本套用合法持有會立即鎖定該物件。例如，您可能需要對與調查或法律糾紛相關的物件套用合法持有。合法持有沒有失效日期，會一直有效，直到明確移除為止。

物件的合法持有設定獨立於保留模式和保留截止日期。如果某個物件版本處於合法持有，則任何人均無法刪除該版本。

若要在新增物件版本至儲存桶時指定 S3 Object Lock 設定，請發出 **"PutObject"**、**"CopyObject"** 或 **"CreateMultipartUpload"** 請求。

您可以使用以下方法：

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



如果指定了 `x-amz-object-lock-mode`，則必須也指定 `x-amz-object-lock-retain-until-date`。

- `x-amz-object-lock-retain-until-date`
 - 保留至日期的值必須採用特定格式 `2020-08-10T21:46:00Z`。允許使用秒的小數部分，但僅保留 3 位小數（毫秒精度）。不允許使用其他 ISO 8601 格式。
 - 保留截止日期必須在未來。
- `x-amz-object-lock-legal-hold`

如果合法持有設定為「ON」（區分大小寫），則物件將被置於合法持有狀態。如果合法持有設定為「OFF」，則不會進行任何合法持有。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：

- 如果 `PutObject` 要求中存在任何 `x-amz-object-lock-*` 要求標頭，則需要 `Content-MD5` 要求標頭。`Content-MD5` 不需要用於 `CopyObject` 或 `CreateMultipartUpload`。
- 如果儲存桶未啟用 S3 物件鎖定，且 `x-amz-object-lock-*` 請求標頭存在，則會傳回 400 Bad Request (InvalidRequest) 錯誤。

- 該 PutObject 請求支援使用 `x-amz-storage-class: REDUCED\_REDUNDANCY` 以符合 AWS 行為。但是，當物件擷取至已啟用 S3 物件鎖定的儲存桶時，StorageGRID 將一律執行雙重提交擷取。
- 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`（如果已配置，且請求發送者擁有正確的 `s3:Get*` 權限）。

您可以使用 `s3:object-lock-remaining-retention-days` 原則條件鍵來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

如果需要更新現有物件版本的合法持有或保留設定，可以執行下列物件子資源操作：

- PutObjectLegalHold

如果新的合法持有值為 ON，該物件將置於合法持有。如果合法持有值為 OFF，合法持有將解除。

- PutObjectRetention
 - 模式值可以是 COMPLIANCE 或 GOVERNANCE（區分大小寫）。
 - 保留至日期的值必須採用特定格式 2020-08-10T21:46:00Z。允許使用秒的小數部分，但僅保留 3 位小數（毫秒精度）。不允許使用其他 ISO 8601 格式。
 - 如果物件版本已設定保留至日期，則只能增加該日期值。新值必須為未來日期。

如何使用 GOVERNANCE 模式

擁有 `s3:BypassGovernanceRetention` 權限的使用者可以略過使用治理模式物件的作用中保留設定。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額外操作：

- 在物件版本的保留期間到期之前，執行 DeleteObject 或 DeleteObjects 操作以刪除物件版本。

受合法持有的物件無法刪除。合法持有必須關閉。

- 執行 PutObjectRetention 操作，在物件的保留期間到期之前，將物件版本的模式從 GOVERNANCE 變更為 COMPLIANCE。

絕對不允許將模式從 COMPLIANCE 變更為 GOVERNANCE。

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

相關資訊

- ["使用 S3 Object Lock 管理物件"](#)
- ["使用 S3 物件鎖定來保留物件"](#)
- ["Amazon Simple Storage Service 使用者指南：鎖定物件"](#)

了解 StorageGRID 的 S3 生命週期組態

您可以建立 S3 生命週期組態來控制何時從 StorageGRID 系統中刪除特定物件。

本節中的簡單範例說明如何透過 S3 生命週期組態來控制何時從特定的 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 生命週期組態的完整詳細資訊，請參閱 ["Amazon Simple Storage Service 使用者指南：物件生命週期管理"](#)。請注意，StorageGRID 僅支援過期動作，不支援轉換動作。

生命週期組態是什麼？

生命週期組態是一組應用於特定 S3 儲存桶中物件的規則。每條規則都指定哪些物件會受到影響，以及這些物件何時過期（在特定日期或經過一定天數後）。

StorageGRID 在生命週期組態中支援最多 1,000 條生命週期規則。每條規則可以包含以下 XML 元素：

- 過期時間：當達到指定日期或達到指定天數時（從物件被擷取之日起算），刪除該物件。
- NoncurrentVersionExpiration：當達到指定的天數時，從物件變成非目前版本之日起刪除該物件。
- 篩選（前綴，標籤）
- 狀態
- ID

每個物件都遵循 S3 儲存桶生命週期或 ILM 原則的保留設定。設定 S3 儲存桶生命週期後，對於符合該儲存桶生命週期篩選器的物件，生命週期過期動作會置換 ILM 原則。不符合該儲存桶生命週期篩選器的物件，則使用 ILM 原則的保留設定。如果物件符合儲存桶生命週期篩選器，但未明確指定過期動作，則不會使用 ILM 原則的保留設定，並且預設物件版本將永久保留。請參閱 ["S3 儲存桶生命週期和 ILM 原則的優先順序範例"](#)。

因此，即使 ILM 規則中的放置指令仍然適用於某個物件，該物件也可能從網格中移除。或者，即使針對該物件的任何 ILM 放置指令都已失效，該物件也可能保留在網格中。如需詳細資訊，請參閱 ["ILM 如何在物件的整個生命週期中運作"](#)。



儲存桶生命週期組態可用於已啟用 S3 Object Lock 的儲存桶，但舊版 Compliant 儲存桶不支援儲存桶生命週期組態。

StorageGRID 支援使用下列儲存桶操作來管理生命週期組態：

- DeleteBucketLifecycle
- GetBucketLifecycleConfiguration
- PutBucketLifecycleConfiguration

建立生命週期組態

建立生命週期組態的第一步是建立一個包含一條或多條規則的 JSON 檔案。例如，以下 JSON 檔案包含三條規則：

1. 規則 1 僅適用於前綴為 `category1/` 且 `key2` 值為 `tag2` 的物件。此 `Expiration` 參數指定符合篩選條件的物件將於 2020 年 8 月 22 日午夜過期。
2. 規則 2 僅適用於前綴為 `category2/` 的物件。此 `Expiration` 參數指定符合篩選條件的物件將在擷取後 100 天過期。



指定天數的規則是相對於物件擷取時間而言的。如果目前日期超過擷取日期加上指定天數，則在套用生命週期組態後，某些物件可能會立即從儲存桶中移除。

3. 規則 3 僅適用於前綴為 `category3/` 的物件。此 ``Expiration`` 參數指定任何符合物件的非目前版本將在變成非目前版本後 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"
    }
  ]
}

```

將生命週期組態套用至儲存桶

建立生命週期組態檔案後，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組態套用至名為 `testbucket` 的儲存貯體中的物件。

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

若要驗證生命週期組態是否已成功套用到儲存桶，請發出 `GetBucketLifecycleConfiguration` 請求。例如：

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

成功回應會列出您剛剛套用的生命週期組態。

驗證儲存貯體生命週期到期是否適用於物件

您可以在發出 `PutObject`、`HeadObject` 或 `GetObject` 請求時，確定生命週期組態中的過期規則是否適用於特定物件。如果規則適用，則回應將包含一個 `Expiration` 參數，用於指示物件的過期時間以及符合的過期規則。



由於儲存桶生命週期會置換 ILM，因此 `expiry-date` 顯示的日期是物件實際刪除的日期。如需詳細資訊，請參閱["如何確定物件保留"](#)。

例如，此 `PutObject` 請求於 2020 年 6 月 22 日發出，並將物件放入 `testbucket` 儲存桶中。

```
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` 請求用於取得 `testbucket` 儲存桶中相同物件的中繼資料。

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

成功回應包含物件的中繼資料，並指示該物件將在 100 天後過期，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貯體，`x-amz-expiration` 回應標頭僅適用於物件的目前版本。

使用 StorageGRID 實作 S3 REST API 的建議

在實作搭配 StorageGRID 使用的 S3 REST API 時，您應該遵循以下建議。

針對不存在物件的 **HEAD** 建議

如果您的應用程式會例行檢查某個物件是否存在於預期該物件實際上不存在的路徑，則應使用「Available」"[\\${post_edited_translations.segment}](#)"一致性。例如，如果您的應用程式在向某個位置執行 PUT 操作之前先執行 HEAD 操作，則應使用「Available」一致性。

否則，如果 HEAD 操作找不到物件，當同一站台的兩個或多個儲存節點無法使用或遠端站台無法連線時，您可能會收到大量 500 內部伺服器錯誤。

您可以使用"[PUT Bucket 一致性](#)"請求為每個儲存桶設定「可用」一致性，也可以在單一 API 操作的請求標頭中指定一致性。

物件鍵的建議

請根據儲存貯體首次建立時間，遵循以下物件索引鍵名稱建議。

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 先前關於鍵前綴的建議相反。相反，請使用非隨機、非唯一的前綴，例如 `image`。
- 如果您遵循 AWS 之前的建議，在鍵前綴中使用隨機且唯一的字元，請在物件鍵前加上目錄名稱作為前綴。也就是說，請使用以下格式：

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

限制物件金鑰名稱以符合效能最佳實務並非必要。在大多數情況下，您可以為物件金鑰名稱的前四個字元使用隨機值。



但有一種例外情況，即某些 S3 工作負載會在短時間內持續刪除所有物件。為了將此使用案例的效能影響降至最低，可以每隔幾千個物件變更金鑰名稱的前導部分，例如新增日期。例如，假設一個 S3 用戶端通常每秒寫入 2,000 個物件，而 ILM 或儲存桶生命週期原則在三天後刪除所有物件。為了將效能影響降至最低，您可以採用以下模式命名金鑰：

/mybucket/mydir/yyyymmddhhmmss-random_UUID.jpg

關於「範圍讀取」的建議

如果"[用於壓縮儲存物件的全域選項](#)"已啟用，S3 用戶端應用程式應避免執行指定要傳回位元組範圍的 `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所要求的位元組。從非常大的物件中要求少量位元組範圍的 `GetObject` 操作效率尤其低下；例如，從 50 GB 的壓縮物件中讀取 10 MB 範圍的效率很低。

如果從壓縮物件讀取範圍，用戶端請求可能會逾時。



如果需要壓縮物件，且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時間。

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。