



# **StorageGRID での S3 REST API の実装**

## StorageGRID

NetApp  
November 04, 2025

# 目次

StorageGRID での S3 REST API の実装	1
競合するクライアント要求です	1
整合性の値	1
整合性の値	1
「Read-after-new-write」整合性と「available」整合性を使用	2
[api-operation-consistency-control]] API処理の整合性を指定します	2
バケットの整合性を指定する	2
整合性とILMルールの相互作用によるデータ保護への影響	3
整合性とILMルールの相互作用の例	3
オブジェクトのバージョン管理	4
ILM とバージョン管理	4
S3 REST APIを使用してS3オブジェクトロックを設定します	5
バケットでS3オブジェクトロックを有効にする方法	5
バケットのデフォルトの保持設定	5
バケットのデフォルトの保持期間を設定する方法	6
バケットのデフォルトの保持期間を確認する方法	7
オブジェクトの保持設定を指定する方法	8
オブジェクトの保持設定を更新する方法	10
ガバナンスマードの使用方法	10
S3 ライフサイクル設定を作成する	11
ライフサイクル構成とは	11
ライフサイクル構成を作成します	12
バケットにライフサイクル設定を適用	14
バケットライフサイクルの有効期限が環境 オブジェクトであることを検証します	14
S3 REST API を実装する際の推奨事項	15
存在しないオブジェクトに対する HEAD の推奨事項	15
オブジェクトキーの推奨事項	16
「範囲読み取り」に関する推奨事項	16

# StorageGRID での S3 REST API の実装

## 競合するクライアント要求です

同じキーに書き込む 2 つのクライアントなど、競合するクライアント要求は、「latest-wins」ベースで解決されます。

「latest-wins」評価は、S3 クライアントが処理を開始するタイミングではなく、StorageGRID システムが特定の要求を完了したタイミングで行われます。

## 整合性の値

整合性では、オブジェクトの可用性と、異なるストレージノードおよびサイト間でのオブジェクトの整合性のバランスが維持されます。アプリケーションの必要に応じて整合性を変更できます。

StorageGRID では、デフォルトで、新しく作成したオブジェクトのリードアフターライト整合性が保証されます。正常に完了した PUT に続く GET では、新しく書き込まれたデータを読み取ることができます。既存のオブジェクトの上書き、メタデータの更新、および削除の整合性レベルは、結果整合性です。上書きは通常、数秒から数分で反映されますが、最大で 15 日かかることがあります。

別の整合性でオブジェクトの処理を実行する場合は、次の操作を実行できます。

- 整合性の指定 [各バケット](#)。
- 整合性の指定 [各API処理](#)。
- 次のいずれかのタスクを実行して、グリッド全体のデフォルトの整合性を変更します。
  - Grid Manager で、[設定] > [ストレージ設定] > [デフォルトの整合性]\* の順に選択します。
  - 。



グリッド全体の整合性に対する変更は、設定の変更後に作成されたバケットにのみ適用されます。変更の詳細を確認するには、次の場所にある監査ログを参照してください。`/var/local/log (* consistencyLevel *を検索)`。

## 整合性の値

整合性は、StorageGRIDがオブジェクトの追跡に使用するメタデータがノード間でどのように分散されるかに影響し、その結果、クライアント要求に対するオブジェクトの可用性にも影響します。

バケットまたはAPI処理の整合性は、次のいずれかの値に設定できます。

- \* all \* : すべてのノードがすぐにデータを受信しないと、要求は失敗します。
- \* strong-global \* : すべてのサイトのすべてのクライアント要求について、リードアフターライト整合性が保証されます。
- \* strong-site \* : サイト内のすべてのクライアント要求に対してリードアフターライト整合性が保証されます。

- \* **Read-after-new-write** \* : (デフォルト) 新規オブジェクトにはリードアフターライト整合性を提供し、オブジェクトの更新には結果整合性を提供します。高可用性が確保され、データ保護が保証されます。ほとんどの場合に推奨されます。
- \* **available** \* : 新しいオブジェクトとオブジェクトの更新の両方について、結果整合性を提供します。S3 バケットの場合は、必要な場合にのみ使用します (読み取り頻度の低いログ値を含むバケットや、存在しないキーに対するHEAD処理やGET処理など)。S3 FabricPool バケットではサポートされません。

## 「Read-after-new-write」整合性と「available」整合性を使用

HEAD処理またはGET処理で「Read-after-new-write」整合性が使用されている場合、StorageGRIDは次のように複数の手順で検索を実行します。

- まず、低い整合性レベルを使用してオブジェクトを検索します。
- そのロックアップが失敗すると、次の整合性値でロックアップが繰り返され、strong-globalの動作と同等の整合性が得られるようになります。

HEAD処理またはGET処理で「Read-after-new-write」整合性が使用されているが、オブジェクトが存在しない場合、オブジェクト検索の整合性は常にstrong-globalの動作と同じになります。この整合性のためには、オブジェクトメタデータのコピーが各サイトで複数ある必要があるため、同じサイトで使用できないストレージノードが複数ある場合に「500 Internal Server Error」が大量に発生する可能性があります。

Amazon S3と同様の整合性の保証が必要ないかぎり、整合性を「available」に設定することで、HEAD処理とGET処理でのこれらのエラーを回避できます。HEAD処理またはGET処理で「available」整合性が使用されている場合、StorageGRIDでは結果整合性のみが提供されます。整合性が向上しても失敗した処理が再試行されることはないと、オブジェクトメタデータのコピーが複数ある必要はありません。

## [api-operation-consistency-control]] API処理の整合性を指定します。

個々のAPI処理に対して整合性を設定するには、処理でサポートされている整合性の値を要求ヘッダーで指定する必要があります。この例では、GetObject処理の整合性を「strong-site」に設定しています。

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



PutObject処理とGetObject処理では、同じ整合性を使用する必要があります。

## バケットの整合性を指定する

バケットの整合性を設定するには、StorageGRID "PUT Bucket consistency" リクエスト。または "バケットの整合性を変更する" Tenant Managerから削除します。

バケットに整合性を設定する場合は、次の点に注意してください。

- バケットに整合性を設定することで、バケット内のオブジェクトまたはバケット設定に対して実行されるS3処理に使用する整合性が決まります。バケット自体に対する処理には影響しません。

- ・個々のAPI処理の整合性がバケットの整合性よりも優先されます。
- ・通常、バケットではデフォルト整合性の「Read-after-new-write」を使用する必要があります。要求が正しく動作しない場合は、可能であればアプリケーションクライアントの動作を変更します。または、API要求ごとに整合性を指定するようにクライアントを設定します。バケットレベルの整合性は最後の手段として設定してください。

## 整合性とILMルールの相互作用によるデータ保護への影響

選択した整合性とILMルールは、どちらもオブジェクトの保護方法に影響します。これらの設定は対話的に操作できます。

たとえば、オブジェクトの格納時に使用される整合性はオブジェクトメタデータの初期配置に影響し、ILMルールで選択された取り込み動作はオブジェクトコピーの初期配置に影響します。StorageGRIDでは、クライアント要求に対応するためにオブジェクトのメタデータとそのデータの両方にアクセスする必要があるため、整合性と取り込み動作で同じ保護レベルを選択すると、初期データ保護が向上し、システム応答の予測性が向上します。

次のようになります "取り込みオプション" ILMルールで使用できます。

### デュアルコミット

StorageGRIDはオブジェクトの中間コピーをただちに作成し、クライアントに成功を返します。可能な場合は、ILMルールで指定されたコピーが作成されます。

### strict

クライアントに成功が返される前に、ILMルールで指定されたすべてのコピーが作成されている必要があります。

### 中間（Balanced）

StorageGRIDは、取り込み時にILMルールで指定されたすべてのコピーの作成を試みます。作成できない場合は中間コピーが作成され、クライアントに成功が返されます。可能な場合は、ILMルールで指定されたコピーが作成されます。

## 整合性とILMルールの相互作用の例

2サイトのグリッドで次のILMルールと整合性が設定されているとします。

- ・ \* ILM ルール \* : ローカルサイトとリモートサイトに1つずつ、2つのオブジェクトコピーを作成します。取り込み動作はStrictを使用します。
- ・ \* consistency \* : strong-global (オブジェクトメタデータがすべてのサイトに即座に分散されます)。

クライアントがオブジェクトをグリッドに格納すると、StorageGRIDは両方のオブジェクトをコピーし、両方のサイトにメタデータを分散してからクライアントに成功を返します。

オブジェクトは、取り込みが成功したことを示すメッセージが表示された時点で損失から完全に保護されます。たとえば、取り込み直後にローカルサイトが失われた場合、オブジェクトデータとオブジェクトメタデータの両方のコピーがリモートサイトに残っています。オブジェクトを完全に読み出し可能にしている。

同じILMルールでstrong-site整合性を使用した場合、オブジェクトデータがリモートサイトにレプリケートされたあと、オブジェクトメタデータが分散される前にクライアントに成功メッセージが返されることがあります。この場合、オブジェクトメタデータの保護レベルがオブジェクトデータの保護レベルと一致しません。取り込み直後にローカルサイトが失われると、オブジェクトメタデータが失われます。オブジェクトを取得でき

ません。

整合性ルールとILMルールの関係は複雑になる可能性があります。サポートが必要な場合は、NetAppにお問い合わせください。

## オブジェクトのバージョン管理

各オブジェクトの複数のバージョンを保持する場合は、バケットのバージョン管理状態を設定できます。バケットのバージョン管理を有効にすると、オブジェクトが誤って削除されないように保護したり、以前のバージョンのオブジェクトを読み出してリストアしたりできます。

StorageGRID システムでは、バージョン管理のほとんどの機能をサポートしていますが、いくつかの制限事項があります。StorageGRID では、オブジェクトごとに最大 1,000 個のバージョンをサポートしています。

オブジェクトのバージョン管理は、StorageGRID の情報ライフサイクル管理 (ILM) または S3 バケットのライフサイクル設定と組み合わせることができます。バージョン管理はバケットごとに明示的に有効にする必要があります。バケットでバージョン管理を有効にすると、バケットに追加される各オブジェクトにバージョンIDが割り当てられ、このIDがStorageGRIDシステムによって生成されます。

MFA (多要素認証) Delete の使用はサポートされていません。



バージョン管理は、StorageGRID バージョン 10.3 以降で作成されたバケットでのみ有効にすることができます。

## ILM とバージョン管理

ILM ポリシーはオブジェクトの各バージョンに適用されます。ILM のスキャン処理では、すべてのオブジェクトが継続的にスキャンされ、現在の ILM ポリシーに照らして再評価されます。ILM ポリシーに対する変更は、それまでに取り込まれたすべてのオブジェクトに適用されます。バージョン管理が有効になっている場合は、それまでに取り込まれたバージョンも対象に ILM のスキャン処理により、過去に取り込まれたオブジェクトに変更後の新しい ILM の内容が適用さ

バージョン管理が有効なバケット内のS3オブジェクトについては、バージョン管理のサポートにより、参照時間に「noncurrent time」を使用するILMルールを作成できます（質問、「Apply this rule to older object versions only？」に対して\* Yes \*を選択します）。イン "[ILMルール作成ウィザードのステップ1](#)"。オブジェクトが更新されると、それまでのバージョンは noncurrent になります。「noncurrent time」フィルタを使用すると、以前のバージョンのオブジェクトによるストレージへの影響を軽減するポリシーを作成できます。



マルチパートアップロード処理を使用してオブジェクトの新しいバージョンをアップロードすると、オブジェクトの元のバージョンの noncurrent の時間には、マルチパートアップロードの完了時ではなく、新しいバージョンのマルチパートアップロードが作成された時点が反映されます。ただし、オリジナルバージョンの最新でない時間は、現行バージョンの時間よりも数時間～数日早い場合があります。

### 関連情報

- ["S3 バージョン管理オブジェクトの削除方法"](#)
- ["S3 バージョン管理オブジェクトの ILM ルールとポリシー（例 4）"。](#)

# S3 REST APIを使用してS3オブジェクトロックを設定します

StorageGRID システムでS3オブジェクトロックのグローバル設定が有効になっている場合は、S3オブジェクトロックを有効にしてバケットを作成できます。デフォルトの保持設定はバケットごとに指定することも、オブジェクトバージョンごとに指定することもできます。

## バケットでS3オブジェクトロックを有効にする方法

StorageGRID システムでグローバルな S3 オブジェクトのロック設定が有効になっている場合は、各バケットの作成時に S3 オブジェクトのロックを必要に応じて有効にすることができます。

S3オブジェクトロックは永続的な設定で、バケットの作成時にのみ有効にできます。バケットの作成後にS3 オブジェクトロックを追加または無効にすることはできません。

バケットでS3オブジェクトロックを有効にするには、次のいずれかの方法を使用します。

- Tenant Manager を使用してバケットを作成します。を参照してください ["S3 バケットを作成する"](#)。
- CreateBucket要求を使用してバケットを作成します。 `x-amz-bucket-object-lock-enabled` 要求ヘッダー。を参照してください ["バケットの処理"](#)。

S3オブジェクトロックにはバケットのバージョン管理が必要です。バージョン管理はバケットの作成時に自動的に有効になります。バケットのバージョン管理を一時停止することはできません。を参照してください ["オブジェクトのバージョン管理"](#)。

## バケットのデフォルトの保持設定

バケットでS3オブジェクトロックが有効になっている場合は、必要に応じてバケットのデフォルトの保持を有効にし、デフォルトの保持モードとデフォルトの保持期間を指定できます。

### デフォルトの保持モード

- コンプライアンスマードの場合：
  - `retain-until-date`に達するまで、オブジェクトを削除できません。
  - オブジェクトの`retain-until-date`は増やすことはできますが、減らすことはできません。
  - オブジェクトの`retain-until-date`は、その日付に達するまで削除できません。
- ガバナンスマードの場合：
  - を使用するユーザ `s3:BypassGovernanceRetention` 権限はを使用できます `x-amz-bypass-governance-retention: true` 保持設定をバイパスする要求ヘッダー。
  - これらのユーザは、`retain-until-date`に達する前にオブジェクトバージョンを削除できます。
  - これらのユーザは、オブジェクトの`retain-until-date`を増減、または削除できます。

### デフォルトの保持期間

各バケットのデフォルトの保持期間は、年または日数で指定できます。

## バケットのデフォルトの保持期間を設定する方法

バケットのデフォルトの保持期間を設定するには、次のいずれかの方法を使用します。

- Tenant Managerからバケット設定を管理します。を参照してください ["S3 バケットを作成します。" および "S3オブジェクトロックのデフォルトの保持期間の更新"](#)。
- 問題デフォルトのモードとデフォルトの日数または年数を指定するための、バケットに対する PutObjectLockConfiguration 要求。

### PutObjectLockConfigurationの略

PutObjectLockConfiguration 要求を使用すると、S3オブジェクトロックが有効になっているバケットに対して、デフォルトの保持モードとデフォルトの保持期間を設定および変更できます。以前に設定したデフォルトの保持設定を削除することもできます。

新しいオブジェクトバージョンがバケットに取り込まれると、にデフォルトの保持モードが適用されます `x-amz-object-lock-mode` および `x-amz-object-lock-retain-until-date` は指定されていません。デフォルトの保持期間は、の `retain-until-date` の計算に使用されます `x-amz-object-lock-retain-until-date` が指定されていません。

オブジェクトバージョンの取り込み後にデフォルトの保持期間が変更された場合、オブジェクトバージョンの `retain-until` はそのまま残り、新しいデフォルトの保持期間を使用して再計算されることはありません。

を用意しておく必要があります `s3:PutBucketObjectLockConfiguration` この処理を完了するための権限 (rootアカウント)。

- Content-MD5 PUT 要求に要求ヘッダーを指定する必要があります。

### 要求例

この例では、バケットでS3オブジェクトロックを有効にし、デフォルトの保持モードを準拠に設定し、デフォルトの保持期間を6年に設定しています。

```
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>
```

## バケットのデフォルトの保持期間を確認する方法

バケットでS3オブジェクトロックが有効になっているかどうかを確認し、デフォルトの保持モードと保持期間を確認するには、次のいずれかの方法を使用します。

- Tenant Managerでバケットを表示します。を参照してください "[S3バケットを表示します](#)"。
- 問題GetObjectLockConfiguration要求。

### GetObjectLockConfigurationの略

GetObjectLockConfiguration要求を使用すると、S3 Object Lockがバケットで有効になっているかどうかを確認できます。有効になっている場合は、バケットにデフォルトの保持モードと保持期間が設定されているかどうかを確認できます。

新しいオブジェクトバージョンがバケットに取り込まれると、にデフォルトの保持モードが適用されます *x-amz-object-lock-mode* が指定されていません。デフォルトの保持期間は、の*retain-until-date*の計算に使用されます *x-amz-object-lock-retain-until-date* が指定されていません。

を用意しておく必要があります *s3:GetBucketObjectLockConfiguration* この処理を完了するための権限 (rootアカウント) 。

### 要求例

```
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
```

応答例

```
HTTP/1.1 200 OK
x-amz-id-2:
iVmcB70XXJRkRH1Fivq1151/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>
```

## オブジェクトの保持設定を指定する方法

S3オブジェクトロックが有効なバケットには、S3オブジェクトロックの保持設定の有無に関係なく、オブジェクトを組み合わせて含めることができます。

オブジェクトレベルの保持設定は、S3 REST APIを使用して指定します。オブジェクトの保持設定は、バケットのデフォルトの保持設定よりも優先されます。

オブジェクトごとに次の設定を指定できます。

- 保持モード：コンプライアンスまたはガバナンスのいずれか。
- \* retain-until-date \* : StorageGRID がオブジェクトバージョンを保持する期間を指定する日付。
  - コンプライアンスマードでは、retain-until-dateが将来の日付の場合、オブジェクトを読み出すことは

できますが、変更や削除はできません。retain-until-dateは増やすことができますが、この日付を減らすことも削除することもできません。

- ガバナンスモードでは、特別な権限を持つユーザーは、retain-until-date設定をバイパスできます。保持期間が経過する前にオブジェクトバージョンを削除できます。また、retain-until-dateを増減したり、削除したりすることもできます。
- \*リーガルホールド\*：オブジェクトバージョンにリーガルホールドを適用すると、そのオブジェクトがただちにロックされます。たとえば、調査または法的紛争に関連するオブジェクトにリーガルホールドを設定する必要がある場合があります。リーガルホールドには有効期限はありませんが、明示的に削除されるまで保持されます。

オブジェクトのリーガルホールド設定は、保持モードやretain-until-dateとは関係ありません。オブジェクトのバージョンがリーガルホールドの対象になっている場合、そのバージョンは誰も削除できません。

バケットにオブジェクトバージョンを追加するときにS3オブジェクトロックの設定を指定するには、問題Aを実行します ["PutObject"](#)、 ["CopyObject"](#) または ["CreateMultipartUpload を実行します"](#) リクエスト。

次のものを使用できます。

- `x-amz-object-lock-mode` コンプライアンスまたはガバナンス（大文字と小文字が区別されます）。



を指定する場合 x-amz-object-lock-mode、も指定する必要があります x-amz-object-lock-retain-until-date。

- x-amz-object-lock-retain-until-date

◦ retain-une-dateの値は、の形式で指定する必要があります 2020-08-10T21:46:00Z。秒数には分数を指定できますが、保持される 10 進数は 3 衔（ミリ秒単位）だけです。その他のISO 8601形式は使用できません。

◦ retain-une-date は将来の日付にする必要があります。

- x-amz-object-lock-legal-hold

リーガルホールドがオン（大文字と小文字が区別される）の場合、オブジェクトはリーガルホールドの対象になります。リーガルホールドがオフの場合、リーガルホールドは適用されません。それ以外の値を指定すると、400 Bad Request（InvalidArgument）エラーが発生します。

次のいずれかの要求ヘッダーを使用する場合は、次の制限事項に注意してください。

- Content-MD5 要求ヘッダーがある場合は必須です x-amz-object-lock-\* 要求ヘッダーがPutObject要求に存在します。Content-MD5 CopyObjectまたはCreateMultipartUploadには必要ありません。
- バケットでS3オブジェクトロックが有効になっていない場合は、とをクリックします x-amz-object-lock-\* 要求ヘッダーが存在し、400 Bad Request（InvalidRequest）エラーが返されます。
- PutObject要求は、次の使用をサポートします。x-amz-storage-class: REDUCED\_REDUNDANCY AWSの動作に合わせて調整できます。ただし、S3 オブジェクトのロックが有効になっているバケットにオブジェクトが取り込まれると、StorageGRID は常にデュアルコミットの取り込みを実行します。
- 後続のGETまたはHeadObjectバージョンの応答には、ヘッダーが含まれます。x-amz-object-lock-mode、x-amz-object-lock-retain-until-date`および`x-amz-object-lock-legal-hold`が設定されている場合、および要求の送信者が正しいかどうか `s3:Get\* 権限：

を使用できます `s3:object-lock-remaining-retention-days` オブジェクトの最小保持期間と最大保持期間を制限するポリシー条件キー。

## オブジェクトの保持設定を更新する方法

既存のオブジェクトのバージョンのリーガルホールドや保持の設定を更新する必要がある場合、次のオブジェクトサブリソース処理を実行できます。

- `PutObjectLegalHold`

新しいリーガルホールドの値が `on` の場合、オブジェクトはリーガルホールドの対象になります。リーガルホールドの値がオフの場合、リーガルホールドは解除されます。

- `PutObjectRetention`

- `mode` 値は `compliance` または `governance` です（大文字と小文字が区別されます）。
- `retain-until-date` の値は、の形式で指定する必要があります `2020-08-10T21:46:00Z`。秒数には分数を指定できますが、保持される 10 進数は 3 衔（ミリ秒単位）だけです。その他の ISO 8601 形式は使用できません。
- オブジェクトバージョンに既存の `retain-until` がある場合は、オブジェクトバージョンを増やすことはできますが、増やすことはできません。新しい値は将来の必要があります。

## ガバナンスモードの使用方法

を持つユーザ `s3:BypassGovernanceRetention` 権限は、ガバナンスモードを使用するオブジェクトのアクティブな保持設定をバイパスできます。DELETE処理または`PutObjectRetention`処理には、`x-amz-bypass-governance-retention:true` 要求ヘッダー。これらのユーザは、次の追加操作を実行できます。

- 保持期間が経過する前にオブジェクトバージョンを削除するには、`DeleteObject` 処理または`DeleteObjects` 処理を実行します。

リーガルホールドの対象になっているオブジェクトは削除できません。リーガルホールドをオフにする必要があります。

- オブジェクトの保持期間が経過する前にオブジェクトバージョンのモードをガバナンスからコンプライアンスに変更する`PutObjectRetention` 処理を実行します。

コンプライアンスモードからガバナンスモードに変更することはできません。

- `PutObjectRetention` 処理を実行して、オブジェクトバージョンの保持期間を増減、または削除します。

## 関連情報

- ["S3 オブジェクトロックでオブジェクトを管理します"](#)
- ["S3オブジェクトロックを使用してオブジェクトを保持します"](#)
- ["Amazon Simple Storage Service User Guide : Using S3 Object Lock"](#)

## S3 ライフサイクル設定を作成する

S3 ライフサイクル設定を作成して、特定のオブジェクトが StorageGRID システムから削除されるタイミングを制御できます。

このセクションの簡単な例では、S3 ライフサイクル設定で特定のオブジェクトが特定の S3 バケットから削除（期限切れ）されるタイミングを制御する方法を示します。このセクションの例は、説明のみを目的としています。S3 ライフサイクル設定の作成の詳細については、を参照してください ["Amazon Simple Storage Serviceユーザガイド：オブジェクトのライフサイクル管理"](#)。StorageGRID では、Expiration アクションのみがサポートされ、移行アクションはサポートされません。

### ライフサイクル構成とは

ライフサイクル設定は、特定の S3 バケット内のオブジェクトに適用される一連のルールです。各ルールは、影響を受けるオブジェクトと、それらのオブジェクトの有効期限（特定の日付または日数後）を指定します。

StorageGRID では、1 つのライフサイクル設定で最大 1,000 個のライフサイクルルールがサポートされます。各ルールには、次の XML 要素を含めることができます。

- Expiration : 指定した日付に達した場合、またはオブジェクトが取り込まれたときから指定した日数に達した場合にオブジェクトを削除します。
- NoncurrentVersionExpiration : 指定した日数に達したオブジェクトを削除します。これは、オブジェクトが最新でなくなったときからです。
- フィルタ（プレフィックス、タグ）
- ステータス
- ID

各オブジェクトは、S3 バケットライフサイクルまたは ILM ポリシーの保持設定に従います。S3 バケットライフサイクルが設定されている場合は、バケットライフサイクルフィルタに一致するオブジェクトの ILM ポリシーがライフサイクル有効期限のアクションで上書きされます。バケットライフサイクルフィルタに一致しないオブジェクトには、ILM ポリシーの保持設定が使用されます。オブジェクトがバケットライフサイクルフィルタに一致し、有効期限の操作が明示的に指定されていない場合、ILM ポリシーの保持設定は使用されず、オブジェクトのバージョンが無期限に保持されることが暗黙的に示されます。を参照してください ["S3 バケットライフサイクルと ILM ポリシーの優先度の例"](#)。

そのため、ILM ルールの配置手順がオブジェクトに引き続き適用されても、オブジェクトがグリッドから削除されることがあります。あるいは、ILM 配置手順がすべて終了したあとも、オブジェクトがグリッドに保持される場合があります。詳細については、を参照してください ["オブジェクトのライフサイクル全体にわたる ILM の動作"](#)。



バケットライフサイクル設定は S3 オブジェクトロックが有効になっているバケットで使用できますが、従来の準拠バケットではバケットライフサイクル設定がサポートされません。

StorageGRID では、次のバケット処理を使用してライフサイクル設定を管理できます。

- DeleteBucketLifecycle
- GetBucketLifecycleConfiguration
- PutBucketLifecycleConfiguration の略

## ライフサイクル構成を作成します

ライフサイクル設定を作成するための最初の手順として、1つ以上のルールを含む JSON ファイルを作成します。たとえば、この JSON ファイルには次の 3 つのルールが含まれています。

1. ルール1は、プレフィックスに一致するオブジェクトにのみ適用されます `category1/` とそれにはがります `key2` の値 `tag2`。 `Expiration` パラメータは、フィルタに一致するオブジェクトの有効期限が2020年8月22日の午前0時に切れるように指定します。
2. ルール2は、プレフィックスに一致するオブジェクトにのみ適用されます。 `category2/`。 `Expiration` パラメータは、フィルタに一致するオブジェクトの取り込みから100日後に期限切れにするを指定します。



日数を指定するルールは、オブジェクトが取り込まれた時点を基準とした相対的なルールです。現在の日付が取り込み日と日数を超えている場合は、ライフサイクル設定の適用後すぐに一部のオブジェクトがバケットから削除される可能性があります。

3. ルール3は、プレフィックスに一致するオブジェクトにのみ適用されます。 `category3/`。 `Expiration` パラメータは、最新でないバージョンの一致オブジェクトが最新でなくなったあと50日で期限切れになるように指定します。

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

## バケットにライフサイクル設定を適用

ライフサイクル設定ファイルを作成したら、PutBucketLifecycleConfiguration要求を発行してバケットに適用します。

次の要求は、サンプルファイル内のライフサイクル設定を、という名前のバケット内のオブジェクトに適用します testbucket。

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

ライフサイクル設定がバケットに正常に適用されたことを確認するには、GetBucketLifecycleConfiguration要求を問題します。例：

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

成功応答には、適用したライフサイクル設定が表示されます。

## バケットライフサイクルの有効期限が環境 オブジェクトであることを検証します

PutObject、HeadObject、またはGetObjectのいずれか環境の要求を発行するときに、ライフサイクル設定の有効期限ルールが特定のオブジェクトであるかどうかを確認できます。ルールが適用される場合、応答にはが含まれます Expiration オブジェクトの有効期限と一致する有効期限を示すパラメータ。



バケットライフサイクルはILMよりも優先されるため、を参照してください `expiry-date` 表示されているのは、オブジェクトが削除される実際の日付です。詳細については、を参照してください ["オブジェクト保持期間の決定方法"](#)。

たとえば、このPutObject要求は2020年6月22日に発行され、オブジェクトを testbucket バケット。

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

成功の応答は、オブジェクトの有効期限が 100 日（2020 年 10 月 1 日）に切れ、ライフサイクル設定のルール 2 に一致したことを示します。

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

たとえば、次のHeadObject要求を使用して、testbucketバケット内の同じオブジェクトのメタデータを取得しました。

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

成功の応答にはオブジェクトのメタデータが含まれ、オブジェクトが 100 日で期限切れになり、ルール 2 に一致したことが示されます。

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



バージョン管理が有効なバケットの場合は、`x-amz-expiration` 応答ヘッダーは、現在のバージョンのオブジェクトにのみ適用されます。

## S3 REST API を実装する際の推奨事項

StorageGRID で使用するためには S3 REST API を実装する場合は、次の推奨事項を考慮してください。

### 存在しないオブジェクトに対する HEAD の推奨事項

オブジェクトが実際に存在するとは思わないパスにオブジェクトが存在するかどうかをアプリケーションが定期的にチェックする場合は、「available」を使用する必要があります。**"一貫性"**。たとえば、アプリケーションがPUTを実行する前に特定の場所に移動する場合は、「available」整合性を使用する必要があります。

そうしないと、同じサイトに複数のストレージノードが使用できない場合やリモートサイトに到達できない場合に、HEAD処理でオブジェクトが見つからないと「500 Internal Server Error」が大量に返されることがあります。

バケットごとに「available」整合性を設定するには、"PUT Bucket consistency" または、個々のAPI処理の要求ヘッダーで整合性を指定できます。

## オブジェクトキーの推奨事項

オブジェクトキー名については、バケットが最初に作成された日時に基づいて次の推奨事項に従ってください。

### StorageGRID 11.4以前で作成されたバケット

- ・オブジェクトキーの最初の4文字にランダムな値を使用しないでください。これは、AWSが以前に推奨していたキープレフィックスの推奨事項とは異なります。代わりに、など、ランダムではなく一意ではないプレフィックスを使用します `image`。
- ・以前のAWSの推奨事項に従ってキープレフィックスにランダムな一意の文字を使用する場合は、オブジェクトキーの前にディレクトリ名を付けます。つまり、次の形式を使用します。

`mybucket/mydir/f8e3-image3132.jpg`

次の形式は使用しないでください。

`mybucket/f8e3-image3132.jpg`

### StorageGRID 11.4以降で作成されたバケット

パフォーマンスのベストプラクティスに合わせてオブジェクトキー名を制限する必要はありません。ほとんどの場合、オブジェクトキー名の最初の4文字にはランダムな値を使用できます。

ただし、短期間ですべてのオブジェクトを継続的に削除するS3ワークロードは例外です。このユースケースのパフォーマンスへの影響を最小限に抑えるには、キー名の先頭部分を数千個のオブジェクトごとに、日付などの値を変更します。たとえば、S3クライアントが1秒あたり2,000個のオブジェクトを書き込むのが一般的で、ILMまたはバケットライフサイクルポリシーで3日後にすべてのオブジェクトが削除されるとします。パフォーマンスへの影響を最小限に抑えるには、次のようなパターンを使用してキーに名前を付けます。

`/mybucket/mydir/yyyymmddhhmmss-random_UUID.jpg`

## 「範囲読み取り」に関する推奨事項

状況に応じて "格納オブジェクトを圧縮するグローバルオプション" が有効になっている場合は、S3クライアントアプリケーションで返されるバイト数の範囲を指定するGetObject処理を実行しないでください。これらの「範囲読み取り」処理は効率的ではありません。StorageGRIDでは、要求されたバイトにアクセスするためにオブジェクトの圧縮を実質的に解除する必要があるためです。非常に大きなオブジェクトから小さい範囲のバイト数を要求するGetObject処理は特に非効率的です。たとえば、50GBの圧縮オブジェクトから10MBの範囲を読み取る処理は非効率的です。

圧縮オブジェクトから範囲を読み取ると、クライアント要求がタイムアウトする可能性があります。

○ オブジェクトを圧縮する必要があり、クライアントアプリケーションが範囲読み取りを使用する必要がある場合は、アプリケーションの読み取りタイムアウトを増やしてください。

## 著作権に関する情報

Copyright © 2025 NetApp, Inc. All Rights Reserved. Printed in the U.S.このドキュメントは著作権によって保護されています。著作権所有者の書面による事前承諾がある場合を除き、画像媒体、電子媒体、および写真複写、記録媒体、テープ媒体、電子検索システムへの組み込みを含む機械媒体など、いかなる形式および方法による複製も禁止します。

ネットアップの著作物から派生したソフトウェアは、次に示す使用許諾条項および免責条項の対象となります。

このソフトウェアは、ネットアップによって「現状のまま」提供されています。ネットアップは明示的な保証、または商品性および特定目的に対する適合性の暗示的保証を含み、かつこれに限定されないいかなる暗示的な保証も行いません。ネットアップは、代替品または代替サービスの調達、使用不能、データ損失、利益損失、業務中断を含み、かつこれに限定されない、このソフトウェアの使用により生じたすべての直接的損害、間接的損害、偶発的損害、特別損害、懲罰的損害、必然的損害の発生に対して、損失の発生の可能性が通知されていたとしても、その発生理由、根拠とする責任論、契約の有無、厳格責任、不法行為（過失またはそうでない場合を含む）にかかわらず、一切の責任を負いません。

ネットアップは、ここに記載されているすべての製品に対する変更を隨時、予告なく行う権利を保有します。ネットアップによる明示的な書面による合意がある場合を除き、ここに記載されている製品の使用により生じる責任および義務に対して、ネットアップは責任を負いません。この製品の使用または購入は、ネットアップの特許権、商標権、または他の知的所有権に基づくライセンスの供与とはみなされません。

このマニュアルに記載されている製品は、1つ以上の米国特許、その他の国の特許、および出願中の特許によって保護されている場合があります。

権利の制限について：政府による使用、複製、開示は、DFARS 252.227-7013（2014年2月）およびFAR 5225.227-19（2007年12月）のRights in Technical Data -Noncommercial Items（技術データ - 非商用品目に関する諸権利）条項の(b)(3)項、に規定された制限が適用されます。

本書に含まれるデータは商用製品および / または商用サービス（FAR 2.101の定義に基づく）に関係し、データの所有権はNetApp, Inc.にあります。本契約に基づき提供されるすべてのネットアップの技術データおよびコンピュータソフトウェアは、商用目的であり、私費のみで開発されたものです。米国政府は本データに対し、非独占的かつ移転およびサブライセンス不可で、全世界を対象とする取り消し不能の制限付き使用権を有し、本データの提供の根拠となった米国政府契約に関連し、当該契約の裏付けとする場合にのみ本データを使用できます。前述の場合を除き、NetApp, Inc.の書面による許可を事前に得ることなく、本データを使用、開示、転載、改変するほか、上演または展示することはできません。国防総省にかかる米国政府のデータ使用権については、DFARS 252.227-7015(b)項（2014年2月）で定められた権利のみが認められます。

## 商標に関する情報

NetApp、NetAppのロゴ、<http://www.netapp.com/TM>に記載されているマークは、NetApp, Inc.の商標です。その他の会社名と製品名は、それを所有する各社の商標である場合があります。