



# StorageGRIDがS3 REST APIを実装する方法

## StorageGRID software

NetApp  
December 03, 2025

# 目次

StorageGRIDがS3 REST APIを実装する方法	1
競合するクライアント要求	1
一貫性の値	1
一貫性の値	1
「新規書き込み後の読み取り」と「利用可能」の一貫性を使用する	2
API操作の一貫性を指定する	2
バケットの一貫性を指定する	2
一貫性とILMルールがどのように相互作用してデータ保護に影響を与えるか	3
一貫性とILMルールがどのように相互作用するかの例	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する	5
バケットの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決定する方法	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
ガバナンスモードの使い方	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する	12
バケットにライフサイクル設定を適用する	14
バケットのライフサイクルの有効期限がオブジェクトに適用されていることを検証する	14
S3 REST API 実装に関する推奨事項	15
存在しないオブジェクトへのHEADの推奨事項	15
オブジェクトキーに関する推奨事項	15
「レンジリード」の推奨事項	16

# StorageGRIDがS3 REST APIを実装する方法

## 競合するクライアント要求

2つのクライアントが同じキーに書き込むなど、競合するクライアント要求は、「最新のもの」を優先して解決されます。

「最新のもの優先」評価のタイミングは、S3クライアントが操作を開始した時点ではなく、StorageGRIDシステムが特定のリクエストを完了した時点に基づきます。

## 一貫性の値

一貫性は、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一貫性との間のバランスを提供します。アプリケーションの必要に応じて一貫性を変更できます。

デフォルトでは、StorageGRIDは新しく作成されたオブジェクトの書き込み後の読み取り一貫性を保証します。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、および削除は、最終的に一貫性が保たれます。上書きが反映されるまでには通常数秒または数分かかりますが、最大15日かかる場合があります。

異なる一貫性でオブジェクト操作を実行する場合は、次の操作を実行できます。

- 一貫性を指定する [各バケツ](#)。
- 一貫性を指定する [各API操作](#)。
- 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一貫性を変更します。
  - グリッドマネージャーで、構成 > システム > ストレージ設定 > デフォルトの一貫性 に移動します。
  - 。



グリッド全体の一貫性の変更は、設定の変更後に作成されたバケツ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*`を検索してください)。

## 一貫性の値

一貫性は、StorageGRIDがオブジェクトを追跡するため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、したがってクライアント要求に対するオブジェクトの可用性に影響します。

バケツまたはAPI操作の一貫性を次のいずれかの値に設定できます。

- **すべて:** すべてのノードがデータを直ちに受信します。そうでない場合、要求は失敗します。
- **強力なグローバル:** すべてのサイトにわたるすべてのクライアント要求の書き込み後の読み取り一貫性を保証します。
- **強力なサイト:** サイト内のすべてのクライアント要求に対して、書き込み後の読み取りの一貫性を保証します。

- 新規書き込み後の読み取り: (デフォルト) 新しいオブジェクトに対して書き込み後の読み取りの一貫性を提供し、オブジェクトの更新に対して最終的な一貫性を提供します。高可用性とデータ保護の保証を提供します。ほとんどの場合に推奨されます。
- 利用可能: 新しいオブジェクトとオブジェクトの更新の両方に対して最終的な一貫性を提供します。S3 バケットの場合は、必要な場合にのみ使用してください (たとえば、めったに読み取られないログ値を含むバケットの場合や、存在しないキーに対する HEAD または GET 操作の場合など)。S3 FabricPoolバケットではサポートされていません。

## 「新規書き込み後の読み取り」と「利用可能」の一貫性を使用する

HEAD または GET 操作で「Read-after-new-write」一貫性を使用する場合、StorageGRID は次のように複数のステップで検索を実行します。

- まず、低い一貫性を使用してオブジェクトを検索します。
- その検索が失敗した場合、strong-global の動作と同等の一貫性に達するまで、次の一貫性値での検索を繰り返します。

HEAD または GET 操作で「新規書き込み後の読み取り」一貫性が使用され、オブジェクトが存在しない場合は、オブジェクトの検索は常に strong-global の動作と同等の一貫性に到達します。この一貫性を保つには、各サイトでオブジェクト メタデータの複数のコピーが使用可能である必要があるため、同じサイトにある 2 つ以上のストレージ ノードが使用できない場合、多数の 500 内部サーバー エラーが発生する可能性があります。

Amazon S3 と同様の一貫性の保証が必要ない場合は、一貫性を「利用可能」に設定することで、HEAD および GET 操作でこれらのエラーを防ぐことができます。HEAD または GET 操作で「使用可能」な一貫性が使用される場合、StorageGRID は最終的な一貫性のみを提供します。一貫性が増すにつれて失敗した操作を再試行することはないため、オブジェクト メタデータの複数のコピーが利用可能である必要はありません。

## API操作の一貫性を指定する

個々の API 操作の一貫性を設定するには、その操作に対して一貫性の値がサポートされている必要があります。リクエスト ヘッダーで一貫性を指定する必要があります。この例では、GetObject 操作の一貫性を「Strong-site」に設定します。

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



PutObject 操作と GetObject 操作の両方に同じ一貫性を使用する必要があります。

## バケットの一貫性を指定する

バケットの一貫性を設定するには、StorageGRIDを使用します。["PUT バケットの一貫性"](#)リクエスト。あるいは["バケットの一貫性を変更する"](#)テナントマネージャーより。

バケットの一貫性を設定するときは、次の点に注意してください。

- バケットの一貫性を設定すると、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 操作に使用される一貫性が決まります。バケット自体の操作には影響しません。
- 個々の API 操作の一貫性は、バケットの一貫性よりも優先されます。
- 一般に、バケットではデフォルトの一貫性である「新規書き込み後の読み取り」を使用する必要があります。リクエストが正しく機能しない場合は、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、各 API リクエストの一貫性を指定するようにクライアントを構成します。最後の手段としてのみ、バケット レベルで一貫性を設定します。

## 一貫性とILMルールがどのように相互作用してデータ保護に影響を与えるか

一貫性の選択と ILM ルールの両方が、オブジェクトの保護方法に影響します。これらの設定は相互作用する可能性があります。

たとえば、オブジェクトの保存時に使用される一貫性は、オブジェクト メタデータの初期配置に影響し、ILM ルールに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 はクライアントの要求を満たす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、一貫性と取り込み動作に一致する保護レベルを選択すると、初期データ保護が向上し、システム応答がより予測可能になります。

次の["取り込みオプション"](#)ILM ルールに使用できるもの:

### Dual commit

StorageGRID は直ちにオブジェクトの中間コピーを作成し、クライアントに成功を返します。ILM ルールで指定されたコピーは、可能な場合に作成されます。

### 厳しい

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を作成する必要があります。

### バランスの取れた

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を作成しようとします。これが不可能な場合は、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。ILM ルールで指定されたコピーは、可能な場合に作成されます。

## 一貫性とILMルールがどのように相互作用するかの例

次の ILM ルールと次の一貫性を持つ 2 つのサイト グリッドがあるとします。

- **ILM ルール:** ローカル サイトとリモート サイトに 1 つずつ、合計 2 つのオブジェクト コピーを作成します。厳密な取り込み動作を使用します。
- **一貫性:** 強力なグローバル (オブジェクト メタデータはすべてのサイトに直ちに配布されます)。

クライアントがオブジェクトをグリッドに保存すると、StorageGRID は両方のオブジェクトのコピーを作成し、両方のサイトにメタデータを配布してから、クライアントに成功を返します。

オブジェクトは、取り込み成功メッセージの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でも、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 サイトに残ります。オブジェクトは完全に取得可能です。

代わりに同じ ILM ルールと強力なサイト一貫性を使用した場合、オブジェクト データがリモート サイトにレ

プリアートされた後、オブジェクト メタデータがそこに配布される前に、クライアントは成功メッセージを受信する可能性があります。この場合、オブジェクト メタデータの保護レベルは、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 サイトが失われた場合、オブジェクト メタデータは失われます。オブジェクトを取得できません。

一貫性と ILM ルール間の相互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るのを防ぎ、オブジェクトの以前のバージョンを取得して復元できるようになります。

StorageGRIDシステムは、ほとんどの機能をサポートするバージョン管理を実装していますが、いくつかの制限があります。StorageGRID は、各オブジェクトの最大 10,000 バージョンをサポートします。

オブジェクトのバージョン管理は、StorageGRID情報ライフサイクル管理 (ILM) または S3 バケット ライフサイクル構成と組み合わせることができます。各バケットに対してバージョン管理を明示的に有効にする必要があります。バケットのバージョン管理が有効になっている場合、バケットに追加された各オブジェクトには、StorageGRIDシステムによって生成されるバージョン ID が割り当てられます。

MFA (多要素認証) を使用した削除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 スキャン プロセスは、すべてのオブジェクトを継続的にスキャンし、現在の ILM ポリシーに対して再評価します。ILM ポリシーに加えた変更は、以前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、以前に取り込まれたバージョンも含まれます。ILM スキャンは、以前に取り込まれたオブジェクトに新しい ILM 変更を適用します。

バージョン管理が有効なバケット内のS3オブジェクトの場合、バージョン管理のサポートにより、「非現在時刻」を参照時刻として使用するILMルールを作成できます（「このルールを古いオブジェクトバージョンにのみ適用しますか？」という質問に対して\*はい\*を選択します）。「[ILMルールの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、以前のバージョンは非最新になります。「非現在の時間」フィルターを使用すると、オブジェクトの以前のバージョン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操作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非現在の時間は、マルチパートアップロードが完了したときではなく、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ときに反映されます。場合によっては、元のバージョンの非現在の時刻が現在のバージョンの時刻よりも数時間または数日前になることがあります。

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する

StorageGRIDシステムでグローバル S3 オブジェクト ロック設定が有効になっている場合は、S3 オブジェクト ロックが有効になっているバケットを作成できます。各バケットのデフォルトの保持期間、または各オブジェクト バージョンの保持期間設定を指定できます。

### バケットの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 S3 オブジェクト ロック設定が有効になっている場合は、各バケットを作成するときにオプションで S3 オブジェクト ロックを有効にできます。

S3 オブジェクト ロックは、バケットを作成するときのみ有効にできる永続的な設定です。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。

バケットに対して 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。

- テナント マネージャーを使用してバケットを作成します。見る["S3バケットを作成する"](#)。
- CreateBucketリクエストを使用してバケットを作成します。`x-amz-bucket-object-lock-enabled` リクエスト ヘッダー。見る["バケットの操作"](#)。

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。これ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。見る["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

バケットに対して S3 オブジェクトロックが有効になっている場合は、オプションでバケットのデフォルトの保存を有効にし、デフォルトの保存モードとデフォルトの保存期間を指定できます。

#### デフォルトの保持モード

- コンプライアンスモードの場合:
  - オブジェクトは、保持期限に達するまで削除できません。
  - オブジェクトの保持期限を増やすことはできますが、減らすことはできません。
  - オブジェクトの保持期限は、その日付に達するまで削除できません。
- ガバナンスモードの場合:
  - ユーザーは `s3:BypassGovernanceRetention` 許可は `x-amz-bypass-governance-retention: true` 保持設定をバイパスするためのリクエスト ヘッダー。
  - これらのユーザーは、オブジェクト バージョンを、その保持期限に達する前に削除できます。
  - これらのユーザーは、オブジェクトの保持期限を増減または削除できます。

## デフォルト保持期間

各バケットには、年または日数で指定されたデフォルトの保持期間を設定できます。

## 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。

- テナント マネージャーからバケット設定を管理します。見る["S3バケットを作成する"](#)そして["S3 オブジェクトロックのデフォルト保持を更新"](#)。
- バケットに対して PutObjectLockConfiguration リクエストを発行し、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します。

## PutObjectLockConfiguration

PutObjectLockConfiguration リクエストを使用すると、S3 オブジェクトロックが有効になっているバケットの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。以前に構成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、次の場合、デフォルトの保持モードが適用されます。`x-amz-object-lock-mode`そして`x-amz-object-lock-retain-until-date`指定されていません。デフォルトの保存期間は、次の場合に保存期限日を計算するために使用されます。`x-amz-object-lock-retain-until-date`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 バージョンの保持期限は同じままとなり、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。

あなたは`s3:PutBucketObjectLockConfiguration`この操作を完了するには、権限を持っているか、アカウントの root である必要があります。

その`Content-MD5`PUT リクエストではリクエスト ヘッダーを指定する必要があります。

### リクエスト例

この例では、バケットの 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 COMPLIANCE に、デフォルトの保持期間を 6 年に設定します。

```
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決定する方法

バケットに対して 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。

- テナント マネージャーでバケットを表示します。見る["S3バケットを表示"](#)。
- `GetObjectLockConfiguration` 要求を発行します。

### GetObjectLockConfiguration

`GetObjectLockConfiguration` リクエストを使用すると、バケットに対して 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、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、次の場合、デフォルトの保持モードが適用されます。`x-amz-object-lock-mode` 指定されていません。デフォルトの保存期間は、次の場合に保存期限日を計算するために使用されます。`x-amz-object-lock-retain-until-date` 指定されていません。

あなたは `s3:GetBucketObjectLockConfiguration` この操作を完了するには、権限を持っているか、アカウントの root である必要があります。

リクエスト例

```
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:
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>
```

## 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になっているバケットには、S3 オブジェクトロックの保持設定があるオブジェクトとないオブジェクトの組み合わせを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 を使用して指定されます。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。

各オブジェクトに対して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:** オブジェクト バージョンをStorageGRIDで保持する必要がある期間を指定する日付。
  - COMPLIANCE モードでは、保持期限が将来の日付である場合、オブジェクトを取得することはでき

ますが、変更または削除することはできません。保持期限を増やすことはできますが、この日付を減らしたり削除したりすることはできません。

- GOVERNANCE 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せん。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、保持期限を延長、短縮、または削除することもできます。
- 法的保留: オブジェクト バージョンに法的保留を適用すると、そのオブジェクトは直ちにロックされます。たとえば、調査や法的紛争に関連するオブジェクトに対して法的保留を設定する必要がある場合があります。法的保留には有効期限はありませんが、明示的に削除されるまで有効のままになります。

オブジェクトの法的保留設定は、保持モードおよび保持期限とは無関係です。オブジェクト バージョンが法的保留中の場合、そのバージョンを削除することは誰にもできません。

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設定を指定するには、"[PutObject](#)"、"[CopyObject](#)"、または"[CreateMultipartUpload](#)"リクエスト。

以下を使用できます。

- `x-amz-object-lock-mode` COMPLIANCE または GOVERNANCE (大文字と小文字が区別されます) のいずれかになります。



指定する場合 x-amz-object-lock-mode、また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
  - 保持期限の値は、次の形式でなければなりません。2020-08-10T21:46:00Z。小数秒は許可されませんが、保持されるのは小数点第 3 位のみです (ミリ秒の精度)。その他の ISO 8601 形式は許可されません。
  - 保持期限は将来の日付にする必要があります。
- x-amz-object-lock-legal-hold

法的保留がオンの場合 (大文字と小文字が区別されます)、オブジェクトは法的保留の対象となります。法的保留がオフの場合、法的保留は設定されません。その他の値を指定すると、400 Bad Request (InvalidArgument) エラーが発生します。

これらのリクエスト ヘッダーのいずれかを使用する場合は、次の制限に注意してください。

- その Content-MD5 リクエストヘッダーは必須です `x-amz-object-lock-\*` リクエスト ヘッダーは PutObject リクエストに存在します。`Content-MD5 CopyObject` または CreateMultipartUpload には必要ありません。
- バケットにS3オブジェクトロックが有効になっていない場合、`x-amz-object-lock-\*` リクエスト ヘッダーが存在する場合、400 Bad Request (InvalidRequest) エラーが返されます。
- PutObjectリクエストは、x-amz-storage-class: REDUCED\_REDUNDANCY AWS の動作と一致させるためです。ただし、S3 オブジェクト 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取り込みを実行します。
- 後続のGETまたはHeadObjectバージョンのレスポンスにはヘッダーが含まれます x-amz-object-lock-mode、x-amz-object-lock-retain-until-date、そして `x-amz-object-lock-legal-hold` 設定されていて、リクエスト送信者が正しい `s3:Get\*` 権限。

使用することができます `s3:object-lock-remaining-retention-days` オブジェクトの許容される最小および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 バージョンの法的保留または保持設定を更新する必要がある場合は、次のオブジェクト サブリソース操作を実行できます。

- `PutObjectLegalHold`

新しい法的保留値が ON の場合、オブジェクトは法的保留下に置かれます。法的保留値が OFF の場合、法的保留は解除されます。

- `PutObjectRetention`

- モード値は、COMPLIANCE または GOVERNANCE (大文字と小文字が区別されます) になります。
- 保持期限の値は、次の形式でなければなりません。2020-08-10T21:46:00Z。小数秒は許可されませんが、保持されるのは小数点第 3 位のみです (ミリ秒の精度)。その他の ISO 8601 形式は許可されません。
- オブジェクト バージョンに既存の保持期限がある場合は、それを増やすことしかできません。新しい値は将来のものである必要があります。

## ガバナンスモードの使い方

ユーザーは `s3:BypassGovernanceRetention` 権限により、GOVERNANCE 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がバイパスされる可能性があります。DELETE または `PutObjectRetention` 操作には、`x-amz-bypass-governance-retention:true` リクエスト ヘッダー。これらのユーザー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` または `DeleteObjects` 操作を実行します。

法的保留中のオブジェクトは削除できません。法的保留は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 GOVERNANCE から COMPLIANCE に変更する `PutObjectRetention` 操作を実行します。

モードを COMPLIANCE から GOVERNANCE に変更することは許可されません。

- オブジェクト バージョンの保持期間を増加、減少、または削除するには、`PutObjectRetention` 操作を実行します。

### 関連情報

- ["S3 オブジェクトロックでオブジェクトを管理する"](#)
- ["S3 オブジェクトロックを使用してオブジェクトを保持する"](#)
- ["Amazon Simple Storage Service ユーザーガイド: オブジェクトのロック"](#)

# 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システムからいつ削除されるかを制御することができます。

このセクションの簡単な例は、S3 ライフサイクル設定によって、特定の S3 バケットから特定のオブジェクトがいつ削除される (期限切れになる) かを制御する方法を示しています。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、以下を参照してください。"[Amazon Simple Storage Service ユーザーガイド: オブジェクトのライフサイクル管理](#)"。StorageGRID は有効期限アクションのみをサポートし、遷移アクションはサポートしていないことに注意してください。

## 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。各ルールでは、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 (特定の日付または一定の日数後) を指定します。

StorageGRID は、ライフサイクル構成で最大 1,000 個のライフサイクル ルールをサポートします。各ルールには次の XML 要素を含めることができます。

- 有効期限: 指定した日付に達したとき、またはオブジェクトが取り込まれた時点から指定した日数に達したときに、オブジェクトを削除します。
- NoncurrentVersionExpiration: オブジェクトが非最新になった時点から指定された日数が経過すると、オブジェクトを削除します。
- フィルター (プレフィックス、タグ)
- ステータス
- ID

各オブジェクトは、S3 バケットの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 バケットのライフサイクルが設定されている場合、ライフサイクルの有効期限アクションは、バケットのライフサイクル フィルターに一致するオブジェクトの ILM ポリシーをオーバーライドします。バケットのライフサイクル フィルターに一致しないオブジェクトは、ILM ポリシーの保持設定を使用します。オブジェクトがバケットのライフサイクル フィルタに一致し、有効期限アクション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 バージョンは永久に保持されるものとみなされます。見る"[S3 バケット ライフサイクルと ILM ポリシーの優先順位の例](#)"。

その結果、ILM ルールの配置指示がオブジェクトに適用されているにもかかわらず、オブジェクトがグリッドから削除される可能性があります。または、オブジェクトに対する ILM 配置指示が失効した後でも、オブジェクトがグリッド上に保持されることがあります。詳細については、"[ILM がオブジェクトのライフサイクル全体にわたってどのように機能するか](#)"。



バケットの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のライフサイクル設定はサポートされていません。

StorageGRID は、ライフサイクル構成を管理するために次のバケット操作の使用をサポートしています。

- DeleteBucketLifecycle
- GetBucketLifecycleConfiguration

## ライフサイクル構成を作成する

ライフサイクル構成を作成する最初の手順として、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は接頭辞に一致するオブジェクトにのみ適用される `category1`/そして、`key2``の価値 ``tag2`。その ``Expiration``パラメーターは、フィルターに一致するオブジェクトが2020年8月22日の深夜に期限切れになることを指定します。
2. ルール2は接頭辞に一致するオブジェクトにのみ適用される `category2`。その ``Expiration``パラメーターは、フィルタに一致するオブジェクトが取り込まれてから100日後に期限切れになることを指定します。



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ます。現在の日付が取り込み日に日数を加えた日付を超えている場合、ライフサイクル設定が適用されると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は接頭辞に一致するオブジェクトにのみ適用される `category3`。その ``Expiration``パラメーターは、一致するオブジェクトの非現在のバージョンが非現在のバージョンになってから50日後に期限切れになることを指定します。

```

{
  "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する

ライフサイクル設定ファイルを作成したら、PutBucketLifecycleConfiguration リクエストを発行してそれをバケットに適用します。

このリクエストは、サンプルファイルのライフサイクル設定を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
```

ライフサイクル構成がバケットに正常に適用されたことを検証するには、GetBucketLifecycleConfiguration リクエストを発行します。例えば：

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

成功した応答には、適用したライフサイクル構成が一覧表示されます。

## バケットのライフサイクルの有効期限がオブジェクトに適用されていることを検証する

PutObject、HeadObject、または GetObject リクエストを発行するときに、ライフサイクル構成の有効期限ルールが特定のオブジェクトに適用されるかどうかを判断できます。ルールが適用される場合、応答には `Expiration` オブジェクトの有効期限と、どの有効期限ルールに一致したかを示すパラメーター。



バケットのライフサイクルはILMよりも優先されるため、`expiry-date`表示されるのは、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、["物体の保持がどのように決定されるか"](#)。

例えば、このPutObjectリクエストは2020年6月22日に発行され、オブジェクトを `testbucket` バケツ。

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

成功応答は、オブジェクトが 100 日後 (2020 年 10 月 1 日) に期限切れになり、ライフサイクル構成のルール 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、この HeadObject リクエストは、testbucket バケット内の同じオブジェクトのメタデータを取得するために使用されました。

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

成功応答にはオブジェクトのメタデータが含まれており、オブジェクトが 100 日後に期限切れになることと、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": {}
}
```



バージョン対応バケットの場合、`x-amz-expiration` 応答ヘッダーはオブジェクトの現在のバージョンにのみ適用されます。

## S3 REST API 実装に関する推奨事項

StorageGRIDで使用するために S3 REST API を実装する場合は、次の推奨事項に従う必要があります。

### 存在しないオブジェクトへのHEADの推奨事項

アプリケーションが、実際には存在しないと思われるパスにオブジェクトが存在するかどうかを定期的に確認する場合は、「使用可能」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 を実行する前に場所を HEAD する場合は、「Available」一貫性を使用する必要があります。

そうでない場合、HEAD 操作でオブジェクトが見つからないと、同じサイトにある 2 つ以上のストレージ ノードが使用できないか、リモート サイトにアクセスできないときに、多数の 500 内部サーバー エラーが発生する可能性があります。

各バケットの「利用可能」一貫性を設定するには、["PUT バケットの一貫性"](#) リクエストごとに一貫性を指定したり、個々の API 操作のリクエスト ヘッダーで一貫性を指定したりすることもできます。

### オブジェクトキーに関する推奨事項

バケットが最初に作成された時期に基づいて、オブジェクト キー名に関する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 文字としてランダムな値を使用しないでください。これは、キープレフィックスに関する以前の AWS 推奨事項とは対照的です。代わりに、ランダムではない、ユニークではない接頭辞を使用してください。 image。
- キープレフィックスにランダムで一意的文字を使用するという以前の AWS 推奨事項に従う場合は、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
```

この形式の代わりに:

```
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 プラクティスを満たすためにオブジェクト 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。



例外は、短時間ですべてのオブジェクトを継続的に削除する S3 ワークロードです。このユースケースでのパフォーマンスへの影響を最小限に抑えるには、数千個のオブジェクトごとにキー名の先頭部分を日付などに変更します。たとえば、S3 クライアントが通常 1 秒あたり 2,000 個のオブジェクトを書き込み、ILM またはバケット ライフサイクル ポリシーによって 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/yyyyymmddhhmmss-random_UUID.jpg
```

## 「レンジリード」の推奨事項

もし"保存された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、S3 クライアントアプリケーションは、返されるバイトの範囲を指定する GetObject 操作の実行を避ける必要があります。これらの「範囲読み取り」操作は、StorageGRID が要求されたバイトにアクセスするためにオブジェクトを効果的に解凍する必要があるため、非効率的です。非常に大きなオブジェクトから小さな範囲のバイトを要求する GetObject 操作は特に非効率的です。たとえば、50 GB の圧縮されたオブジェクトから 10 MB の範囲を読み取るのは非効率的です。

範囲が圧縮されたオブジェクトから読み取られる場合、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 アプリケーションで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ます。

## 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随時、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 5252.227-19（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。