



StorageGRID による S3 REST API の実装方法

StorageGRID software

NetApp
July 23, 2026

目次

StorageGRID による S3 REST API の実装方法	1
StorageGRID S3 REST APIが競合するクライアント要求を解決する方法	1
StorageGRID S3 REST APIが可用性と一貫性のバランスをとる方法	1
一貫性値	1
「新規書き込み後の読み取り」と「利用可能」の整合性を使用する	2
API操作の整合性の指定	3
バケットの整合性を指定する	3
一貫性とILMルールがデータ保護に与える影響	3
一貫性とILMルールがどのように相互作用するかの例	4
StorageGRIDでのオブジェクトバージョン管理の使用方法	4
ILMとバージョン管理	5
S3 REST APIを使用してStorageGRID S3 Object Lock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	6
バケットのデフォルト保持期間を設定する方法	6
バケットのデフォルト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用方法	10
StorageGRIDのS3ライフサイクル設定について学ぶ	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構成を適用する	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ることを確認します。	14
StorageGRIDで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へのHEADに関する推奨事項	15
オブジェクトキーに関する推奨事項	16
「範囲読み取り」に関する推奨事項	16

StorageGRID による S3 REST API の実装方法

StorageGRID S3 REST APIが競合するクライアント要求を解決する方法

2つのクライアントが同じキーに書き込むなど、クライアントのリクエストが競合する場合は、「最新のリクエストが優先される」方式で解決されます。

「最新優先」評価のタイミングは、S3クライアントが操作を開始したときではなく、StorageGRID システムが特定のリクエストを完了したときを基準とします。

StorageGRID S3 REST APIが可用性と一貫性のバランスをとる方法

一貫性とは、オブジェクトの可用性と、異なるストレージノードやサイト間でのオブジェクトの一貫性とのバランスを取るものです。アプリケーションの要件に応じて、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について、書き込み後の読み取りの一貫性を保証します。PUT が正常に完了した後に行われる GET リクエストは、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、最終的に整合性が保たれます。

オブジェクト操作を異なる整合性で実行する場合は、次の操作を実行できます。

- **各バケット**の整合性を指定します。
- **各API操作**の一貫性を指定します。
- 以下のいずれかの操作を実行して、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、**Configuration > System > Storage settings > Default bucket consistency** に移動します。
 - 。



グリッド全体の整合性に関する変更は、設定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、`/var/local/log`にある監査ログを参照してください（**consistencyLevel** を検索）。

一貫性値

一貫性は、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ます。一貫性は、クライアントリクエストに対するオブジェクトの可用性に影響します。

バケットまたはAPI操作の一貫性は、以下のいずれかの値に設定できます：

- **すべて**：すべてのノードがオブジェクトメタデータを即座に受信します。受信しない場合、リクエストは失敗します。

- **Strong-global:** すべてのサイトにおけるすべてのクライアントリクエストに対して、書き込み後の読み取りの一貫性を保証します。Quorumのセマンティクスが設定されている場合、以下の動作が適用されます：
 - グリッドに3つ以上のサイトがある場合、クライアント要求に対するサイト障害の耐性を確保します。2サイトグリッドにはサイト障害の耐性はありません。
 - 1つのサイトがダウンしている場合、以下のS3操作は成功しません：
 - DeleteBucketEncryption
 - PutBucketBranch
 - PutBucketEncryption
 - PutBucketVersioning
 - PutObjectLegalHold
 - PutObjectLockConfiguration
 - PutObjectRetention

必要であれば、"[StorageGRID のクォーラムセマンティクスをstrong-global整合性に設定する](#)"。

- **Strong-site:** オブジェクトのメタデータは、サイト内の他のノードに即座に配信されます。サイト内のすべてのクライアントリクエストに対して、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：新規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結果整合性を提供します。高可用性とデータ保護を保証します。ほとんどの場合に推奨されます。
- 利用可能：新規オブジェクトとオブジェクト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、または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」の整合性を使用する

HEAD または GET 操作で「Read-after-new-write」整合性を使用する場合、StorageGRID は以下の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、次の整合性値で検索を繰り返し、strong-global の動作と同等の整合性に達するまで続けます。

HEAD または GET 操作で「新規書き込み後の読み取り」整合性が使用されている場合でも、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 の動作と同等の整合性に達します。この整合性を維持するには、各サイトにオブジェクトメタデータの複数のコピーが必要となるため、同じサイトにある 2 つ以上のストレージノードが利用できない場合、500 Internal Server Error が多数発生する可能性があります。

Amazon S3 と同様の整合性保証が不要な場合は、整合性を「Available」に設定することで、HEAD および GET 操作でこれらのエラーが発生するのを防ぐことができます。HEAD または GET 操作で「Available」整合性を使用する場合、StorageGRID は最終的な整合性のみを提供します。失敗した操作をより高い整合性で再試行しないため、オブジェクトメタデータの複数のコピーを用意する必要はありません。

API操作の整合性の指定

個々のAPI操作の一貫性を設定するには、その操作で一貫性の値がサポートされている必要があり、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 と GetObject の両方の操作に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の一貫性" リクエストを使用します。または、Tenant Manager から "バケットの整合性を変更する" することもできます。

バケットの整合性を設定する際には、以下の点に注意してください。

- バケットの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構成に対して実行されるS3操作に使用される整合性が決まります。これは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。
- 一般的に、バケットはデフォルトの整合性設定である「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てください。または、クライアント側で各 API リクエストの整合性を指定するように設定することもできます。バケットレベルでの整合性の設定は、最終手段としてのみ行ってください。

一貫性とILMルールがデータ保護に与える影響

一貫性の選択とILMルールは、どちらもオブジェクトの保護方法に影響を与えます。これらの設定は相互に影響し合う可能性があります。

例えば、オブジェクトを保存する際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を与え、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を与えます。StorageGRID が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データの両方へのアクセスを必要とするため、一貫性と取り込み動作に対して適切なレベルの保護を選択することで、より優れた初期データ保護とより予測可能なシステム応答を実現できます。

次の"取り込みオプション"は、ILMルールで使用できます：

Dual commit

StorageGRID はオブジェクトの暫定コピーを即座に作成し、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。

厳格

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

バランス

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。そ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ルールと次の整合性を持つ3サイトグリッドがあるとします。

- **ILMルール**：オブジェクトのコピーを3つ作成します。1つはローカルサイトに、残りの2つは各リモートサイトにそれぞれ1つずつ作成します。厳密な取り込み動作を使用します。
- **一貫性**：Strong-global（オブジェクトのメタデータは複数のサイトに即座に配信されます）。

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 はオブジェクトのコピーを3つすべて作成し、メタデータを複数のサイトに配布してから、クライアントに成功を返します。

取り込み成功メッセージの時点で、オブジェクトは損失から完全に保護されます。例えば、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。オブジェクトは他のサイトから完全に取得可能です。

代わりに同じILMルールとstrong-siteの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後、オブジェクトの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、オブジェクトの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。

整合性とILMルールの相互関係は複雑になる場合があります。サポートが必要な場合は、NetAppにお問い合わせください。

StorageGRIDでのオブジェクトバージョン管理の使用方法

各オブジェクトの複数のバージョンを保持したい場合は、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、オブジェクトの誤削除を防ぎ、オブジェクトの以前のバージョンを取得および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、各オブジェクトにつき最大 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バケットのライフサイクル構成と組み合わせることができます。各バケットでバージョン管理を明示的に有効にする必要があります。バケットで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バージョン ID が割り当てられます。このバージョン ID は StorageGRID システムによって生成されます。

多要素認証 (MFA) 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ポリシーに基づいて再評価します。ILMポリシーに加えた変更は、既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、これに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になっているバケット内の 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、参照時間として「非現在の時間」を使用する ILM ルールを作成できます（["ILMルール作成ウィザードのステップ1"](#)の「このルールを古いオブジェクトバージョンのみに適用しますか？」という質問に対して はいを選択します）。オブジェクトが更新されると、その以前のバージョンは最新ではなくなります。「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がストレージに与える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場合、元のバージョンのオブジェクトの非現在時刻は、マルチパートアップロードが完了した時刻ではなく、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刻を反映します。限られたケースでは、元のバージョンの非現在時刻が、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場合があります。

関連情報

- ["S3 バージョン管理されたオブジェクトはどのように削除されるのか"](#)
- ["S3バージョン管理対象オブジェクトに関するILMルールとポリシー（例4）"](#)。

S3 REST APIを使用してStorageGRID S3 Object Lockを設定する

グローバル S3 オブジェクトロック設定が StorageGRID システムで有効になっている場合、S3 オブジェクトロックを有効にしたバケットを作成できます。各バケットのデフォルト保持期間を指定したり、オブジェクトのバージョンごとに保持設定を指定したりできます。

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、各バケットを作成する際にオプションで S3 オブジェクトロックを有効にすることができます。

S3オブジェクトロックは、バケット作成時にのみ有効にできる永続的な設定です。バケット作成後に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。

バケットの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。["S3バケットを作成する"](#)を参照してください。

- CreateBucket リクエストと `x-amz-bucket-object-lock-enabled` リクエストヘッダーを使用してバケットを作成します。["バケットに対する操作"](#)を参照してください。

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あり、これはバケット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。["オブジェクトのバージョン管理"](#)を参照してください。

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、オプションでバケットのデフォルト保持を有効にし、デフォルト保持モードとデフォルト保持期間を指定できます。

デフォルトの保持モード

- コンプライアンスモードの場合：
 - オブジェクトは、保持期限が到来するまで削除できません。
 - オブジェクトの保持期限は延長できますが、短縮することはできません。
 - オブジェクトの保持期限日は、その日付が到来するまで削除することはできません。
- ガバナンスモードでは：
 - `s3:BypassGovernanceRetention` 権限を持つユーザーは、`x-amz-bypass-governance-retention: true` リクエストヘッダーを使用して保持設定をバイパスできます。
 - これらのユーザーは、オブジェクトのバージョンが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、年または日単位で指定できる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。["S3バケットを作成する"](#)および["S3オブジェクトロックのデフォルト保持期間を更新する"](#)を参照してください。
- バケットに対して PutObjectLockConfiguration リクエストを発行し、デフォルト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、`x-amz-object-lock-mode` および `x-amz-object-lock-retain-until-date` が指定されていない場合、デフォルトの保持モードが適用されます。`x-amz-object-lock-retain-until-date` が指定されていない場合、デフォルトの保持期間を使用して保持期限日が計算されます。

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変更されず、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。

この操作を完了するには、`s3:PutBucketObjectLockConfiguration`権限が必要です。またはアカウントのrootである必要があります。

`Content-MD5`リクエストヘッダーは、PUTリクエストで指定する必要があります。

リクエスト例

この例では、バケットに対して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、また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確認してください。["S3バケットを表示する"](#)を参照してください。
- `GetObjectLockConfiguration` リクエストを発行します。

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でき、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、`x-amz-object-lock-mode`が指定されていない場合、デフォルトの保持モードが適用されます。デフォルトの保持期間は、`x-amz-object-lock-retain-until-date`が指定されていない場合に保持期限日を計算するために使用されます。

この操作を完了するには、`s3:GetBucketObjectLockConfiguration`権限が必要です。またはアカウントの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
```

回答例

```
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適用されているオブジェクトと適用されていないオブジェクトが混在して格納される可能性があります。

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、以下の設定を指定できます。

- 保持モード：COMPLIANCE または GOVERNANCE のいずれか。
- 保持期限：オブジェクトのバージョンを 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、保持期限が未来の日付の場合、オブジェクトを取得することはできませんが、変更または削除することはできません。保持期限は延長できますが、短縮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、保持期限設定をバイパスできます。オブジェクトのバージョンは、保持期間が経過する前に削除することができます。また、保存期限を延長、短縮、あるいは削除することも可能です。
- 法的保留：オブジェクトの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即座にロックされます。例えば、調査や法的紛争に関連するオブジェクトに対して、法的保留を適用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なく、明示的に解除されるまで有効です。

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独立しています。オブジェクトのバージョンが法的保留状態にある場合、誰もそのバージョンを削除することはできません。

バケットにオブジェクトバージョンを追加する際に S3 Object Lock 設定を指定するには、["PutObject"](#)、["CopyObject"](#)、または ["CreateMultipartUpload"](#) リクエストを発行します。

以下のものを使用できます。

- ``x-amz-object-lock-mode`` これは、COMPLIANCE または GOVERNANCE のいずれかになります（大文字と小文字を区別します）。



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

- `x-amz-object-lock-retain-until-date`
 - 保持期限の値は、次の形式である必要があります 2020-08-10T21:46:00Z。小数秒も許容されますが、保持されるのは小数点以下3桁（ミリ秒単位の精度）のみです。その他のISO 8601形式は許可されていません。
 - 保持期限日は未来の日付でなければなりません。
- `x-amz-object-lock-legal-hold`

法的保留がオン（大文字小文字を区別）の場合、対象物は法的保留下に置かれます。法的保留がオフの場合、法的保留は行われません。その他の値では、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を使用する場合は、以下の制限事項に注意してください：

- ``Content-MD5`` リクエストヘッダーは、``x-amz-object-lock-*`` リクエストヘッダーがPutObjectリクエストに存在する場合に必須です。``Content-MD5`` はCopyObjectまたはCreateMultipartUploadには必要ありません。
- バケットで S3 Object Lock が有効になっておらず、``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、AWSの動作に合わせるために `x-amz-storage-class: REDUCED\_REDUNDANCY` の使用をサポートしています。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を取り込む場合、StorageGRIDは常にデュアルコミット取り込みを実行します。
- その後のGETまたはHeadObjectバージョンの応答には、設定されており、かつリクエスト送信者が適切な `s3:Get*` 権限を持っている場合、ヘッダー `x-amz-object-lock-mode`、`x-amz-object-lock-retain-until-date`、および `x-amz-object-lock-legal-hold` が含まれます。

```
`s3:object-lock-remaining-retention-  
days` ポリシー条件キーを使用して、オブジェクトの最小および最大許容保持期間を制限できま  
す。
```

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、対象物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のいずれかになります（大文字と小文字を区別します）。
 - 保持期限の値は、次の形式である必要があります 2020-08-10T21:46:00Z。小数秒も許容されますが、保持されるのは小数点以下3桁（ミリ秒単位の精度）のみです。その他のISO 8601形式は許可されていません。
 - オブジェクトのバージョンに保持期限が設定されている場合、その期限を延長することしかできません。新しい値は将来の日付である必要があります。

ガバナンスモードの使用方法

ユーザーは `s3:BypassGovernanceRetention` 権限によって、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をバイパスすることができます。DELETE または PutObjectRetention 操作には、`x-amz-bypass-governance-retention:true` リクエストヘッダーを含める必要があります。これらのユーザーは、以下の追加操作を実行できます：

- DeleteObject または DeleteObjects 操作を実行して、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します。

法的拘束を受けている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。

コンプライアンスモードからガバナンスモードへの変更は一切許可されていません。

- PutObjectRetention 操作を実行して、オブジェクトバージョンの保持期間を延長、短縮、または削除します。

関連情報

- ["S3 Object Lock 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"](#)

StorageGRIDのS3ライフサイクル設定について学ぶ

S3ライフサイクル構成を作成して、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によって、特定のS3バケットから特定のオブジェクトが削除（期限切れ）されるタイミングをどのように制御できるかを示しています。このセクションの例は、あくまで説明のためのものです。S3ライフサイクル設定の作成に関する詳細については、["Amazon Simple Storage Service ユーザーガイド：オブジェクトライフサイクル管理"](#)を参照してください。StorageGRID は Expiration アクションのみをサポートしており、Transition アクションはサポートしていません。

ライフサイクル構成とは

ライフサイクル設定とは、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のことです。各ルールは、どのオブジェクトが影響を受けるか、そしてそれらのオブジェクトがいつ期限切れになるか（特定の日付、または一定日数後）を指定します。

StorageGRID はライフサイクル構成において、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、以下のXML要素を含めることができます。

- 有効期限：オブジェクトが取り込まれた時点から、指定された日付に達したとき、または指定された日数が経過したときに、オブジェクトを削除します。
- NoncurrentVersionExpiration：オブジェクトが非現行状態になった時点から指定された日数が経過したら、その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ポリシーのいずれか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、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については、ライフサイクルの有効期限切れアクションがILMポリシーを上書きします。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ライフサイクルフィルタに一致し、かつ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、オブジェクトのバージョンは永久に保持されるとみなされます。["S3バケットのライフサイクルとILMポリシーの優先順位の例"](#)を参照してください。

その結果、ILMルール内の配置指示がオブジェクトに依然として適用されている場合でも、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。あるいは、オブジェクトに対するILM配置指示の有効期限が切れた後でも、オブジェクトがグリッド上に保持される場合があります。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を通してどのように機能するか"](#)を参照してください。



バケットライフサイクル設定は、S3 Object Lock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はサポートされていません。

StorageGRID は、ライフサイクル設定を管理するために以下のバケット操作の使用をサポートしています。

- DeleteBucketLifecycle
- GetBucketLifecycleConfiguration
- PutBucketLifecycleConfiguration

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ステップとして、1つ以上のルールを含むJSONファイルを作成します。例えば、このJSONファイルには以下の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 `category1/` に一致し、``key2``の値が ``tag2``であるオブジェクトにのみ適用されます。``Expiration``パラメータは、フィルターに一致するオブジェクトが2020年8月22日午前0時に期限切れになることを指定します。
2. ルール2は、プレフィックス `category2/`に一致するオブジェクトにのみ適用されます。``Expiration``パラメータは、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为准とします。現在の日付が取り込み日付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、プレフィックス `category3/`に一致するオブジェクトにのみ適用されます。``Expiration``パラメータは、一致するオブジェクトの非現行バージョンが非現行になってから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構成を適用する

ライフサイクル構成ファイルを作成したら、PutBucketLifecycleConfiguration リクエストを発行してバケットに適用します。

このリクエストは、サンプルファイル内のライフサイクル構成を、`testbucket` という名前のバケット内のオブジェクトに適用します。

```
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：

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

正常に応答すると、適用したライフサイクル構成が表示されます。

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ることを確認します。

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は、PutObject、HeadObject、またはGetObject リクエストの発行時に判断できます。ルールが適用される場合、応答にはオブジェクトの有効期限が切れるタイミングと一致した有効期限ルールを示す `Expiration` パラメータが含まれます。



バケットライフサイクルはILMを上書きするため、`expiry-date` 表示されているのは、オブジェクトが実際に削除される日付です。詳細については、["オブジェクト保持期間の決定方法"](#)を参照してください。

たとえば、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` バケットに配置します。

```
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）に期限切れとなり、ライフサイクル構成のルール2に一致したことを示しています。

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

例えば、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、およびRule 2に一致したことが示されています。

```
{
  "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になっているバケットの場合、`x-amz-expiration`レスポンスヘッダーは、オブジェクトの最新バージョンにのみ適用されます。

StorageGRIDでS3 REST APIを実装する際の推奨事項

StorageGRID で使用するための S3 REST API を実装する際には、以下の推奨事項に従ってください。

存在しないオブジェクトへのHEADに関する推奨事項

アプリケーションが、オブジェクトが実際には存在しないと想定さ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チェックする場合は、「Available」[一貫性](#)を使用する必要があります。例えば、アプリケーションがPUTする前にHEADで場所を確認する場合は、「Available」の整合性を使用する必要があります。

それ以外の場合、HEAD操作でオブジェクトが見つからない場合、同じサイトにある2つ以上のストレージノードが利用できない場合、またはリモートサイトにアクセスできない場合は、多数の500 Internal Server Errorが発生する可能性があります。

各バケットの「利用可能」整合性は、"[PUT Bucketの一貫性](#)"リクエストを使用して設定するか、個々のAPI操作のリクエストヘッダーで整合性を指定することができます。

オブジェクトキーに関する推奨事項

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時期に基づいて、以下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、以前のAWSのキープレフィックスに関する推奨事項とは対照的です。代わりに、`image`のような非ランダムで非一意のプレフィックスを使用してください。
- 以前のAWSの推奨事項に従って、キーのプレフィックスにランダムで一意の文字を使用する場合は、オブジェクトキーのプレフィックスにディレクトリ名を追加してください。つまり、この形式を使用してください：

```
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、オブジェクトの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、短時間後に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。このユースケースにおけるパフォーマンスへの影響を最小限に抑えるため、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の値で変更してください。例えば、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
```

「範囲読み取り」に関する推奨事項

"[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、返すバイト範囲を指定する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」操作は非効率的です。これは、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です。非常に大きなオブジェクトから少数のバイト範囲を要求するGetObject操作は特に非効率的です。たとえば、50 GBの圧縮オブジェクトから10 MBの範囲を読み取ることは非効率的です。

圧縮されたオブジェクトから範囲を読み取る場合、クライアントからのリクエスト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長くしてください。

著作権に関する情報

Copyright © 2026 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权を有し、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权については、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。