S3 REST APIを使用する

StorageGRID software

NetApp
July 23, 2026

目次

S3 REST APIを使用する

StorageGRID S3 REST API でサポートされているバージョンとアップデート

StorageGRID は、Simple Storage Service （S3） API をサポートしており、一連の Representational State Transfer （REST） Web サービスとして実装されています。

S3 REST APIのサポートにより、S3 Webサービス向けに開発されたサービス指向アプリケーションを、StorageGRIDシステムを使用するオンプレミスオブジェクトストレージに接続できます。クライアントアプリケーションで現在使用しているS3 REST API呼び出しへの変更は最小限で済みます。

サポート対象のバージョン

StorageGRID は、以下の特定のバージョンの STS、S3、および HTTP をサポートしています。

項目	**version**
STS AssumeRole API仕様	"Amazon Web Services (AWS) ドキュメント：Amazon Assumerole API リファレンス" AssumeRole の詳細については、"AssumeRole の設定" を参照してください。
S3 API仕様	"AWSドキュメント：Amazon Simple Storage Service API リファレンス"
HTTP	1.1 HTTPに関する詳細については、HTTP/1.1（RFCs 7230-35）を参照してください。 "IETF RFC 2616：ハイパーテキスト転送プロトコル（HTTP/1.1）" 注記：StorageGRID は HTTP/1.1 パイプライン処理をサポートしていません。

S3 REST APIサポートのアップデート

リリース	Comments
12.0	• 管理インターフェイスのクロスオリジンリソース共有（CORS）のサポートを追加しました。これにより、別のドメインが管理APIを使用してStorageGRID内のデータにアクセスできるようになります。"詳細情報"。 • STS AssumeRole およびセッションポリシーのサポートを追加しました。"セッションポリシーの例" を参照してください。テナントグループで AssumeRole を設定できます。
11.9	• 以下のリクエストおよびサポートされているヘッダーに対して、事前計算済みの SHA-256 チェックサム値のサポートを追加しました。この機能を使用すると、アップロードされたオブジェクトの整合性を検証できます。 ◦ CompleteMultipartUpload： `x-amz-checksum-sha256` ◦ CreateMultipartUpload： `x-amz-checksum-algorithm` ◦ GetObject： `x-amz-checksum-mode` ◦ HeadObject： `x-amz-checksum-mode` ◦ ListParts ◦ PutObject： `x-amz-checksum-sha256` ◦ UploadPart： `x-amz-checksum-sha256` • グリッド管理者がテナントレベルのデータ保持およびコンプライアンス設定を制御できる機能を追加しました。これらの設定は、S3オブジェクトロックの設定に影響します。 ◦ バケットのデフォルトの保持モードとオブジェクトの保持モード：グリッド管理者が許可している場合、Governance または Compliance。 ◦ バケットのデフォルトの保持期間とオブジェクトの保持終了日：グリッド管理者が設定した最大保持期間以下である必要があります。 • サポートの改善 `aws-chunked`コンテンツのエンコードとストリーミング `x-amz-content-sha256`値。制限事項： ◦ 存在する場合、 `chunk-signature`はオプションであり、検証は行われません ◦ 存在する場合、 `x-amz-trailer`コンテンツは無視されます
11.8	S3 操作の名前を、 "Amazon Web Services (AWS) ドキュメント：Amazon Simple Storage Service API リファレンス"で使用されている名前と一致するように更新しました。

リリース	Comments
11.7	• "クイックリファレンス：サポートされているS3 APIリクエスト"を追加しました。 • S3オブジェクトロックでガバナンスモードを使用するためのサポートを追加しました。 • GET Object および HEAD Object リクエストに対する StorageGRID 固有の `x-ntap-sg-cgr-replication-status`レスポンスヘッダーのサポートを追加しました。このヘッダーは、クロスグリッドレプリケーションにおけるオブジェクトのレプリケーション状態を示します。 • SelectObjectContent リクエストが Parquet オブジェクトをサポートするようになりました。
11.6	• GET Object および HEAD Object リクエストにおける `partNumber`リクエストパラメータの使用のサポートを追加しました。 • S3オブジェクトロックにおいて、バケットレベルでのデフォルトの保持モードとデフォルトの保持期間のサポートを追加しました。 • `s3:object-lock-remaining-retention-days`ポリシー条件キーのサポートが追加され、オブジェクトの許容保持期間の範囲を設定できるようになりました。 • 単一の PUT Object 操作における_推奨_最大サイズを 5 GiB（5,368,709,120 バイト）に変更しました。5 GiB を超えるオブジェクトがある場合は、代わりにマルチパートアップロードを使用してください。
11.5	• バケット暗号化の管理機能を追加しました。 • S3オブジェクトロックのサポートを追加し、従来のコンプライアンスリクエストを廃止しました。 • バージョン管理されたバケットで複数のオブジェクトを削除する機能のサポートを追加しました。 • The `Content-MD5`リクエストヘッダーが正しくサポートされるようになりました。
11.4	• DELETE Bucket tagging、GET Bucket tagging、PUT Bucket tagging のサポートを追加しました。コスト配分タグはサポートされていません。 • StorageGRID 11.4 で作成されたバケットについては、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なくなりました。 • `s3:ObjectRestore:Post`イベントタイプのバケット通知のサポートが追加されました。 • AWSのマルチパートパーツのサイズ制限が適用されるようになりました。マルチパートアップロードの各パートは、5 MiB以上5 GiB以下である必要があります。最後のパートは5 MiB未満でも構いません。 • TLS 1.3のサポートを追加しました。

リリース	Comments
11.3	• 顧客提供の鍵を使用したオブジェクトデータのサーバー側暗号化（SSE-C）のサポートを追加しました。 • DELETE、GET、PUT バケットライフサイクル操作（有効期限アクションのみ）と `x-amz-expiration`レスポンスヘッダーのサポートが追加されました。 • PUT Object、PUT Object - Copy、およびMultipart Uploadを更新し、取り込み時に同期配置を使用するILMルールの影響について説明を追加しました。 • TLS 1.1暗号はサポートされなくなりました。
11.2	クラウドストレージプールで使用するための、POST Object restoreのサポートを追加しました。グループポリシーおよびバケットポリシーにおいて、ARN、ポリシー条件キー、ポリシー変数に AWS 構文を使用できるようにサポートを追加しました。StorageGRID 構文を使用する既存のグループポリシーおよびバケットポリシーは、引き続きサポートされます。 注: カスタム StorageGRID 機能で使用されるものを含め、他の構成 JSON/XML での ARN/URN の使用は変更されていません。
11.1	クロスオリジンリソース共有（CORS）、グリッドノードへのS3クライアント接続のためのHTTP、およびバケットのコンプライアンス設定のサポートを追加しました。
11.0	プラットフォームサービスの構成に対するサポートを追加しました（CloudMirrorレプリケーション、通知、およびElasticsearch検索統合）。また、バケットに対するオブジェクトタグ付け位置制約と、Available整合性のサポートも追加しました。
10.4	ILM スキャンのバージョン管理への変更、エンドポイントドメイン名ページの更新、ポリシー内の条件と変数、ポリシーの例、および PutOverwriteObject 権限に対するサポートを追加しました。
10.3	バージョン管理機能のサポートを追加しました。
10.2	グループおよびバケットのアクセスポリシー、ならびにマルチパートコピー（Upload Part - Copy）のサポートを追加しました。
10.1	マルチパートアップロード、仮想ホスト型リクエスト、およびv4認証のサポートを追加しました。
10.0	StorageGRID システムによる S3 REST API の初期サポート。現在サポートされている *Simple Storage Service API Reference* のバージョンは 2006-03-01 です。

StorageGRID でサポートされている S3 API リクエスト

このページでは、StorageGRID が Amazon Simple Storage Service（S3）API をサポートする方法について説明します。

このページには、StorageGRID でサポートされている S3 オペレーションのみが含まれています。

各操作に関するAWSドキュメントを表示するには、見出しのリンクを選択してください。

一般的なURIクエリパラメータとリクエストヘッダー

特に明記されていない限り、以下の一般的なURIクエリパラメータがサポートされています：

- `versionId`（オブジェクト操作に必要な場合）

特に明記されていない限り、以下の一般的なリクエストヘッダーがサポートされています。

- `Authorization`
- `Connection`
- `Content-Length`
- `Content-MD5`
- `Content-Type`
- `Date`
- `Expect`
- `Host`
- `x-amz-date`

関連情報

- "S3 REST API実装の詳細"
- "Amazon Simple Storage Service API Reference：共通リクエストヘッダー"

"AbortMultipartUpload"

URIクエリパラメータとリクエストヘッダー

StorageGRID は、このリクエストに対してすべての共通パラメータとヘッダーをサポートしており、さらに以下の追加 URI クエリパラメータもサポートしています：

- `uploadId`

要求の本文

None

StorageGRID ドキュメント

"マルチパートアップロードの操作"

"CompleteMultipartUpload"

URIクエリパラメータとリクエストヘッダー

StorageGRID は、このリクエストに対してすべての共通パラメータとヘッダーをサポートしており、さらに以下の追加 URI クエリパラメータもサポートしています：

- `uploadId`
- `x-amz-checksum-sha256`

リクエストボディの**XML**タグ

StorageGRIDは以下のリクエストボディXMLタグをサポートしています：

- `ChecksumSHA256`
- `CompleteMultipartUpload`
- `ETag`
- `Part`
- `PartNumber`

StorageGRID ドキュメント

"CompleteMultipartUpload"

"CopyObject"

URIクエリパラメータとリクエストヘッダー

StorageGRID は、このリクエストに対してすべての共通パラメータとヘッダーをサポートしており、さらに以下の追加ヘッダーもサポートしています：

- `x-amz-copy-source`
- `x-amz-copy-source-if-match`
- `x-amz-copy-source-if-modified-since`
- `x-amz-copy-source-if-none-match`
- `x-amz-copy-source-if-unmodified-since`
- `x-amz-copy-source-server-side-encryption-customer-algorithm`
- `x-amz-copy-source-server-side-encryption-customer-key`
- `x-amz-copy-source-server-side-encryption-customer-key-MD5`
- `x-amz-metadata-directive`
- `x-amz-object-lock-legal-hold`
- `x-amz-object-lock-mode`
- `x-amz-object-lock-retain-until-date`
- `x-amz-server-side-encryption`
- `x-amz-server-side-encryption-customer-algorithm`
- `x-amz-server-side-encryption-customer-key`
- `x-amz-server-side-encryption-customer-key-MD5`
- `x-amz-storage-class`
- `x-amz-tagging`

- `x-amz-tagging-directive`
- `x-amz-meta-<metadata-name>`

要求の本文

None

StorageGRID ドキュメント

"CopyObject"

"CreateBucket"

URIクエリパラメータとリクエストヘッダー

StorageGRID は、このリクエストに対してすべての共通パラメータとヘッダーをサポートしており、さらに以下の追加ヘッダーもサポートしています：

- `x-amz-bucket-object-lock-enabled`

要求の本文

StorageGRID は、実装時点で Amazon S3 REST API によって定義されたすべてのリクエストボディパラメータをサポートしています。

StorageGRID ドキュメント

"バケットに対する操作"

"CreateMultipartUpload"

URIクエリパラメータとリクエストヘッダー

StorageGRID は、このリクエストに対してすべての共通パラメータとヘッダーをサポートしており、さらに以下の追加ヘッダーもサポートしています：

- `Cache-Control`
- `Content-Disposition`
- `Content-Encoding`
- `Content-Language`
- `Expires`
- `x-amz-checksum-algorithm`
- `x-amz-server-side-encryption`
- `x-amz-storage-class`
- `x-amz-server-side-encryption-customer-algorithm`
- `x-amz-server-side-encryption-customer-key`
- `x-amz-server-side-encryption-customer-key-MD5`
- `x-amz-tagging`
- `x-amz-object-lock-mode`

- `x-amz-object-lock-retain-until-date`
- `x-amz-object-lock-legal-hold`
- `x-amz-meta-<metadata-name>`

要求の本文
None

StorageGRID ドキュメント
"CreateMultipartUpload"

"DeleteBucket"

URIクエリパラメータとリクエストヘッダー
StorageGRID はこのリクエストに対して、すべての共通パラメータとヘッダーをサポートしています。

StorageGRID ドキュメント
"バケットに対する操作"

"DeleteBucketCors"

URIクエリパラメータとリクエストヘッダー
StorageGRID はこのリクエストに対して、すべての共通パラメータとヘッダーをサポートしています。

要求の本文
None

StorageGRID ドキュメント
"バケットに対する操作"

"DeleteBucketEncryption"

URIクエリパラメータとリクエストヘッダー
StorageGRID はこのリクエストに対して、すべての共通パラメータとヘッダーをサポートしています。

要求の本文
None

StorageGRID ドキュメント
"バケットに対する操作"

"DeleteBucketLifecycle"

URIクエリパラメータとリクエストヘッダー
StorageGRID はこのリクエストに対して、すべての共通パラメータとヘッダーをサポートしています。

要求の本文
None

StorageGRID ドキュメント

- "バケットに対する操作"
- "S3ライフサイクル構成を作成する"

"DeleteBucketPolicy"

URIクエリパラメータとリクエストヘッダー

StorageGRID はこのリクエストに対して、すべての共通パラメータとヘッダーをサポートしています。

要求の本文

None

StorageGRID ドキュメント

"バケットに対する操作"

"DeleteBucketReplication"

URIクエリパラメータとリクエストヘッダー

StorageGRID はこのリクエストに対して、すべての共通パラメータとヘッダーをサポートしています。

要求の本文

None

StorageGRID ドキュメント

"バケットに対する操作"

"DeleteBucketTagging"

URIクエリパラメータとリクエストヘッダー

StorageGRID はこのリクエストに対して、すべての共通パラメータとヘッダーをサポートしています。

要求の本文

None

StorageGRID ドキュメント

"バケットに対する操作"

"DeleteObject"

URIクエリパラメータとリクエストヘッダー

StorageGRID は、このリクエストに対するすべての共通パラメータとヘッダーに加え、以下の追加リクエストヘッダーをサポートしています：

- `x-amz-bypass-governance-retention`

要求の本文

None

StorageGRID ドキュメント

"オブジェクトに対する操作"

"DeleteObjects"

URIクエリパラメータとリクエストヘッダー

StorageGRID は、このリクエストに対するすべての共通パラメータとヘッダーに加え、以下の追加リクエストヘッダーをサポートしています：

- `x-amz-bypass-governance-retention`

要求の本文

StorageGRID は、実装時点で Amazon S3 REST API によって定義されたすべてのリクエストボディパラメータをサポートしています。

StorageGRID ドキュメント

"オブジェクトに対する操作"

"DeleteObjectTagging"

StorageGRID はこのリクエストに対して、すべての共通パラメータとヘッダーをサポートしています。

要求の本文

None

StorageGRID ドキュメント

"オブジェクトに対する操作"

"GetBucketAcl"

URIクエリパラメータとリクエストヘッダー

StorageGRID はこのリクエストに対して、すべての共通パラメータとヘッダーをサポートしています。

要求の本文

None

StorageGRID ドキュメント

"バケットに対する操作"

"GetBucketCors"

URIクエリパラメータとリクエストヘッダー

StorageGRID はこのリクエストに対して、すべての共通パラメータとヘッダーをサポートしています。

要求の本文

None

StorageGRID ドキュメント

"バケットに対する操作"

"GetBucketEncryption"

URIクエリパラメータとリクエストヘッダー

StorageGRID はこのリクエストに対して、すべての共通パラメータとヘッダーをサポートしています。

要求の本文

None

StorageGRID ドキュメント

"バケットに対する操作"

"GetBucketLifecycleConfiguration"

URIクエリパラメータとリクエストヘッダー

StorageGRID はこのリクエストに対して、すべての共通パラメータとヘッダーをサポートしています。

要求の本文

None

StorageGRID ドキュメント

- "バケットに対する操作"
- "S3ライフサイクル構成を作成する"

"GetBucketLocation"

URIクエリパラメータとリクエストヘッダー

StorageGRID はこのリクエストに対して、すべての共通パラメータとヘッダーをサポートしています。

要求の本文

None

StorageGRID ドキュメント

"バケットに対する操作"

"GetBucketNotificationConfiguration"

URIクエリパラメータとリクエストヘッダー

StorageGRID はこのリクエストに対して、すべての共通パラメータとヘッダーをサポートしています。

要求の本文

None

StorageGRID ドキュメント

"バケットに対する操作"

"GetBucketPolicy"

URIクエリパラメータとリクエストヘッダー

StorageGRID はこのリクエストに対して、すべての共通パラメータとヘッダーをサポートしています。

要求の本文

None

StorageGRID ドキュメント

"バケットに対する操作"

"GetBucketReplication"

URIクエリパラメータとリクエストヘッダー

StorageGRID はこのリクエストに対して、すべての共通パラメータとヘッダーをサポートしています。

要求の本文

None

StorageGRID ドキュメント

"バケットに対する操作"

"GetBucketTagging"

URIクエリパラメータとリクエストヘッダー

StorageGRID はこのリクエストに対して、すべての共通パラメータとヘッダーをサポートしています。

要求の本文

None

StorageGRID ドキュメント

"バケットに対する操作"

"GetBucketVersioning"

URIクエリパラメータとリクエストヘッダー

StorageGRID はこのリクエストに対して、すべての共通パラメータとヘッダーをサポートしています。

要求の本文

None

StorageGRID ドキュメント

"バケットに対する操作"

"GetObject"

URIクエリパラメータとリクエストヘッダー

StorageGRID は、このリクエストに対してすべての共通パラメータとヘッダーをサポートしており、さらに

以下の追加 URI クエリパラメータもサポートしています：

- `x-amz-checksum-mode`
- `partNumber`
- `response-cache-control`
- `response-content-disposition`
- `response-content-encoding`
- `response-content-language`
- `response-content-type`
- `response-expires`

追加のリクエストヘッダーは次のとおりです：

- `Range`
- `x-amz-server-side-encryption-customer-algorithm`
- `x-amz-server-side-encryption-customer-key`
- `x-amz-server-side-encryption-customer-key-MD5`
- `If-Match`
- `If-Modified-Since`
- `If-None-Match`
- `If-Unmodified-Since`

要求の本文
None

StorageGRID ドキュメント
"GetObject"

"GetObjectAcl"

URIクエリパラメータとリクエストヘッダー
StorageGRID はこのリクエストに対して、すべての共通パラメータとヘッダーをサポートしています。

要求の本文
None

StorageGRID ドキュメント
"オブジェクトに対する操作"

"GetObjectLegalHold"

URIクエリパラメータとリクエストヘッダー

StorageGRID はこのリクエストに対して、すべての共通パラメータとヘッダーをサポートしています。

要求の本文

None

StorageGRID ドキュメント

"S3 REST APIを使用してS3オブジェクトロックを設定する"

"GetObjectLockConfiguration"

URIクエリパラメータとリクエストヘッダー

StorageGRID はこのリクエストに対して、すべての共通パラメータとヘッダーをサポートしています。

要求の本文

None

StorageGRID ドキュメント

"S3 REST APIを使用してS3オブジェクトロックを設定する"

"GetObjectRetention"

URIクエリパラメータとリクエストヘッダー

StorageGRID はこのリクエストに対して、すべての共通パラメータとヘッダーをサポートしています。

要求の本文

None

StorageGRID ドキュメント

"S3 REST APIを使用してS3オブジェクトロックを設定する"

"GetObjectTagging"

URIクエリパラメータとリクエストヘッダー

StorageGRID はこのリクエストに対して、すべての共通パラメータとヘッダーをサポートしています。

要求の本文

None

StorageGRID ドキュメント

"オブジェクトに対する操作"

"HeadBucket"

URIクエリパラメータとリクエストヘッダー

StorageGRID はこのリクエストに対して、すべての共通パラメータとヘッダーをサポートしています。

要求の本文

None

StorageGRID ドキュメント

"バケットに対する操作"

"HeadObject"

URIクエリパラメータとリクエストヘッダー

StorageGRID は、このリクエストに対してすべての共通パラメータとヘッダーをサポートしており、さらに以下の追加ヘッダーもサポートしています：

- `x-amz-checksum-mode`
- `x-amz-server-side-encryption-customer-algorithm`
- `x-amz-server-side-encryption-customer-key`
- `x-amz-server-side-encryption-customer-key-MD5`
- `If-Match`
- `If-Modified-Since`
- `If-None-Match`
- `If-Unmodified-Since`
- `Range`

要求の本文

None

StorageGRID ドキュメント

"HeadObject"

"ListBuckets"

URIクエリパラメータとリクエストヘッダー

StorageGRID はこのリクエストに対して、すべての共通パラメータとヘッダーをサポートしています。

要求の本文

None

StorageGRID ドキュメント

サービス上での操作 › **ListBuckets**

"ListMultipartUploads"

URIクエリパラメータとリクエストヘッダー

StorageGRID は、このリクエストに対するすべての共通パラメータとヘッダーをサポートしており、さらに以下の追加パラメータもサポートしています：

- `encoding-type`
- `key-marker`
- `max-uploads`
- `prefix`
- `upload-id-marker`

要求の本文

None

StorageGRID ドキュメント

"ListMultipartUploads"

"ListObjects"

URIクエリパラメータとリクエストヘッダー

StorageGRID は、このリクエストに対するすべての共通パラメータとヘッダーをサポートしており、さらに以下の追加パラメータもサポートしています：

- `delimiter`
- `encoding-type`
- `marker`
- `max-keys`
- `prefix`

要求の本文

None

StorageGRID ドキュメント

"バケットに対する操作"

"ListObjectsV2を使用したチャンク アップロード署名要求がサポートされるようになりました。"

URIクエリパラメータとリクエストヘッダー

StorageGRID は、このリクエストに対するすべての共通パラメータとヘッダーをサポートしており、さらに以下の追加パラメータもサポートしています：

- `continuation-token`
- `delimiter`

- `encoding-type`
- `fetch-owner`
- `max-keys`
- `prefix`
- `start-after`

要求の本文

None

StorageGRID ドキュメント

"バケットに対する操作"

"ListObjectVersions"

URIクエリパラメータとリクエストヘッダー

StorageGRID は、このリクエストに対するすべての共通パラメータとヘッダーをサポートしており、さらに以下の追加パラメータもサポートしています：

- `delimiter`
- `encoding-type`
- `key-marker`
- `max-keys`
- `prefix`
- `version-id-marker`

要求の本文

None

StorageGRID ドキュメント

"バケットに対する操作"

"ListParts"

URIクエリパラメータとリクエストヘッダー

StorageGRID は、このリクエストに対するすべての共通パラメータとヘッダーをサポートしており、さらに以下の追加パラメータもサポートしています：

- `max-parts`
- `part-number-marker`
- `uploadId`

要求の本文

None

StorageGRID ドキュメント

"ListMultipartUploads"

"PutBucketCors"

URIクエリパラメータとリクエストヘッダー

StorageGRID はこのリクエストに対して、すべての共通パラメータとヘッダーをサポートしています。

要求の本文

StorageGRID は、実装時点で Amazon S3 REST API によって定義されたすべてのリクエストボディパラメータをサポートしています。

StorageGRID ドキュメント

"バケットに対する操作"

"PutBucketEncryption"

URIクエリパラメータとリクエストヘッダー

StorageGRID はこのリクエストに対して、すべての共通パラメータとヘッダーをサポートしています。

リクエストボディの**XML**タグ

StorageGRIDは以下のリクエストボディXMLタグをサポートしています：

- `ApplyServerSideEncryptionByDefault`
- `Rule`
- `ServerSideEncryptionConfiguration`
- `SSEAlgorithm`

StorageGRID ドキュメント

"バケットに対する操作"

"PutBucketLifecycleConfiguration"

URIクエリパラメータとリクエストヘッダー

StorageGRID はこのリクエストに対して、すべての共通パラメータとヘッダーをサポートしています。

リクエストボディの**XML**タグ

StorageGRIDは以下のリクエストボディXMLタグをサポートしています：

- `And`
- `Days`
- `Expiration`
- `ExpiredObjectDeleteMarker`
- `Filter`
- `ID`

- `Key`
- `LifecycleConfiguration`
- `NewerNoncurrentVersions`
- `NoncurrentDays`
- `NoncurrentVersionExpiration`
- `Prefix`
- `Rule`
- `Status`
- `Tag`
- `Value`

StorageGRID ドキュメント

- "バケットに対する操作"
- "S3ライフサイクル構成を作成する"

"PutBucketNotificationConfiguration"

URIクエリパラメータとリクエストヘッダー

StorageGRID はこのリクエストに対して、すべての共通パラメータとヘッダーをサポートしています。

リクエストボディの**XML**タグ

StorageGRIDは以下のリクエストボディXMLタグをサポートしています：

- `Event`
- `Filter`
- `FilterRule`
- `Id`
- `Name`
- `NotificationConfiguration`
- `Prefix`
- `S3Key`
- `Suffix`
- `Topic`
- `TopicConfiguration`
- `Value`

StorageGRID ドキュメント

"バケットに対する操作"

"PutBucketPolicy"

URIクエリパラメータとリクエストヘッダー

StorageGRID はこのリクエストに対して、すべての共通パラメータとヘッダーをサポートしています。

要求の本文

サポートされている JSON ボディ フィールドの詳細については、"バケットおよびグループのアクセスポリシーを使用する"を参照してください。

"PutBucketReplication"

URIクエリパラメータとリクエストヘッダー

StorageGRID はこのリクエストに対して、すべての共通パラメータとヘッダーをサポートしています。

リクエストボディの**XML**タグ

- `Bucket`
- `Destination`
- `Prefix`
- `ReplicationConfiguration`
- `Rule`
- `Status`
- `StorageClass`

StorageGRID ドキュメント

"バケットに対する操作"

"PutBucketTagging"

URIクエリパラメータとリクエストヘッダー

StorageGRID はこのリクエストに対して、すべての共通パラメータとヘッダーをサポートしています。

要求の本文

StorageGRID は、実装時点で Amazon S3 REST API によって定義されたすべてのリクエストボディパラメータをサポートしています。

StorageGRID ドキュメント

"バケットに対する操作"

"PutBucketVersioning"

URIクエリパラメータとリクエストヘッダー

StorageGRID はこのリクエストに対して、すべての共通パラメータとヘッダーをサポートしています。

リクエストボディのパラメータ

StorageGRID は、以下のリクエストボディパラメータをサポートしています：

- `VersioningConfiguration`
- `Status`

StorageGRID ドキュメント

"バケットに対する操作"

"PutObject"

URIクエリパラメータとリクエストヘッダー

StorageGRID は、このリクエストに対してすべての共通パラメータとヘッダーをサポートしており、さらに以下の追加ヘッダーもサポートしています：

- `Cache-Control`
- `Content-Disposition`
- `Content-Encoding`
- `Content-Language`
- `Expires`
- `x-amz-checksum-sha256`
- `x-amz-server-side-encryption`
- `x-amz-storage-class`
- `x-amz-server-side-encryption-customer-algorithm`
- `x-amz-server-side-encryption-customer-key`
- `x-amz-server-side-encryption-customer-key-MD5`
- `x-amz-tagging`
- `x-amz-object-lock-mode`
- `x-amz-object-lock-retain-until-date`
- `x-amz-object-lock-legal-hold`
- `x-amz-meta-<metadata-name>`

要求の本文

- オブジェクトのバイナリデータ

StorageGRID ドキュメント

"PutObject"

"PutObjectLegalHold"

URIクエリパラメータとリクエストヘッダー

StorageGRID はこのリクエストに対して、すべての共通パラメータとヘッダーをサポートしています。

要求の本文

StorageGRID は、実装時点で Amazon S3 REST API によって定義されたすべてのリクエストボディパラメータをサポートしています。

StorageGRID ドキュメント

"S3 REST APIを使用してS3オブジェクトロックを設定する"

"PutObjectLockConfiguration"

URIクエリパラメータとリクエストヘッダー

StorageGRID はこのリクエストに対して、すべての共通パラメータとヘッダーをサポートしています。

要求の本文

StorageGRID は、実装時点で Amazon S3 REST API によって定義されたすべてのリクエストボディパラメータをサポートしています。

StorageGRID ドキュメント

"S3 REST APIを使用してS3オブジェクトロックを設定する"

"PutObjectRetention"

URIクエリパラメータとリクエストヘッダー

StorageGRID は、このリクエストに対するすべての共通パラメータとヘッダーをサポートしており、さらに以下の追加ヘッダーもサポートしています：

- `x-amz-bypass-governance-retention`

要求の本文

StorageGRID は、実装時点で Amazon S3 REST API によって定義されたすべてのリクエストボディパラメータをサポートしています。

StorageGRID ドキュメント

"S3 REST APIを使用してS3オブジェクトロックを設定する"

"PutObjectTagging"

URIクエリパラメータとリクエストヘッダー

StorageGRID はこのリクエストに対して、すべての共通パラメータとヘッダーをサポートしています。

要求の本文

StorageGRID は、実装時点で Amazon S3 REST API によって定義されたすべてのリクエストボディパラメータをサポートしています。

StorageGRID ドキュメント

"オブジェクトに対する操作"

"RestoreObject"

URIクエリパラメータとリクエストヘッダー

StorageGRID はこのリクエストに対して、すべての共通パラメータとヘッダーをサポートしています。

要求の本文

サポートされている本文フィールドの詳細については、"RestoreObject"を参照してください。

"SelectObjectContent"

URIクエリパラメータとリクエストヘッダー

StorageGRID はこのリクエストに対して、すべての共通パラメータとヘッダーをサポートしています。

要求の本文

サポートされている本文フィールドの詳細については、以下を参照してください。

- "S3 Selectを使用する"
- "SelectObjectContent"

"UploadPart"

URIクエリパラメータとリクエストヘッダー

StorageGRID は、このリクエストに対してすべての共通パラメータとヘッダーをサポートしており、さらに以下の追加 URI クエリパラメータもサポートしています：

- `partNumber`
- `uploadId`

追加のリクエストヘッダーは次のとおりです：

- `x-amz-checksum-sha256`
- `x-amz-server-side-encryption-customer-algorithm`
- `x-amz-server-side-encryption-customer-key`
- `x-amz-server-side-encryption-customer-key-MD5`

要求の本文

- 部品のバイナリデータ

StorageGRID ドキュメント

"UploadPart"

"UploadPartCopy"

URIクエリパラメータとリクエストヘッダー

StorageGRID は、このリクエストに対してすべての共通パラメータとヘッダーをサポートしており、さらに以下の追加 URI クエリパラメータもサポートしています：

- partNumber
- uploadId

追加のリクエストヘッダーは次のとおりです：

- x-amz-copy-source
- x-amz-copy-source-if-match
- x-amz-copy-source-if-modified-since
- x-amz-copy-source-if-none-match
- x-amz-copy-source-if-unmodified-since
- x-amz-copy-source-range
- x-amz-server-side-encryption-customer-algorithm
- x-amz-server-side-encryption-customer-key
- x-amz-server-side-encryption-customer-key-MD5
- x-amz-copy-source-server-side-encryption-customer-algorithm
- x-amz-copy-source-server-side-encryption-customer-key
- x-amz-copy-source-server-side-encryption-customer-key-MD5

要求の本文

None

StorageGRID ドキュメント

"UploadPartCopy"

StorageGRID S3 REST API 構成をテストします

Amazon Web Services コマンドラインインターフェイス （AWS CLI） を使用すると、システムへの接続をテストしたり、オブジェクトの読み書きが可能かどうかを確認したりできます。

開始する前に

- AWS CLI を "aws.amazon.com/cli"からダウンロードしてインストールしました。
- オプションとして、"ロードバランサーエンドポイントを作成しました"。それ以外の場合は、接続先の Storage Node の IP アドレスと使用するポート番号がわかっている必要があります。"クライアント接続用のIPアドレスとポート"を参照してください。
- "S3テナントアカウントを作成しました"があります。
- テナントにサインインし、"アクセスキーを作成しました"。

これらの手順の詳細については、"クライアント接続を構成する"を参照してください。

手順

1. StorageGRID システムで作成したアカウントを使用するように AWS CLI の設定を構成します：
   a. 設定モードに入ります： `aws configure`
   b. 作成したアカウントのアクセスキーIDを入力してください。
   c. 作成したアカウントの秘密アクセスキーを入力してください。
   d. 使用するデフォルトのリージョンを入力してください。例えば、 `us-east-1`。
   e. 使用するデフォルトの出力形式を入力するか、**Enter** キーを押して JSON を選択してください。
2. バケットを作成

   この例では、ロードバランサーのエンドポイントをIPアドレス10.96.101.17とポート10443を使用するように構成済みであることを前提としています。

   ```
   aws s3api --endpoint-url https://10.96.101.17:10443
   --no-verify-ssl create-bucket --bucket testbucket
   ```

   バケットが正常に作成された場合、次の例に示すように、バケットの場所が返されます。

   ```
   "Location": "/testbucket"
   ```

3. オブジェクトをアップロードします。

   ```
   aws s3api --endpoint-url https://10.96.101.17:10443 --no-verify-ssl
   put-object --bucket testbucket --key s3.pdf --body C:\s3-
   test\upload\s3.pdf
   ```

   オブジェクトが正常にアップロードされると、オブジェクトデータのハッシュ値であるEtagが返されます。

4. バケットの内容を一覧表示して、オブジェクトがアップロードされたことを確認します。

   ```
   aws s3api --endpoint-url https://10.96.101.17:10443 --no-verify-ssl
   list-objects --bucket testbucket
   ```

5. オブジェクトを削除します。

   ```
   aws s3api --endpoint-url https://10.96.101.17:10443 --no-verify-ssl
   delete-object --bucket testbucket --key s3.pdf
   ```

6. バケットを削除します。

```
aws s3api --endpoint-url https://10.96.101.17:10443 --no-verify-ssl
delete-bucket --bucket testbucket
```

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の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/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>
```

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すが、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を基準とします。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/yyyymmddhhmmss-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。

Amazon S3 REST APIのサポート

StorageGRID における S3 REST API のサポートされている操作と制限

StorageGRID システムは、Simple Storage Service API（API Version 2006-03-01）を実装しており、ほとんどの操作をサポートしていますが、一部制限があります。S3 REST API クライアントアプリケーションを統合する際には、実装の詳細を理解しておく必要があります。

StorageGRIDシステムは、仮想ホスト形式のリクエストとパス形式のリクエストの両方をサポートしています。

日付処理

StorageGRID の S3 REST API の実装は、有効な HTTP 日付形式のみをサポートします。

StorageGRID システムは、日付値を受け入れるヘッダーに対して有効な HTTP 日付形式のみをサポートします。日付の時刻部分は、グリニッジ標準時（GMT）形式、またはタイムゾーンオフセットなしの協定世界時（UTC）形式（+0000 を指定する必要があります）で指定できます。リクエストに `x-amz-date`ヘッダーを含めると、Date リクエストヘッダーで指定された値よりも優先されます。AWS Signature Version 4 を使用する場合、日付ヘッダーはサポートされていないため、署名付きリクエストには `x-amz-date`ヘッダーが存在する必要があります。

一般的なリクエストヘッダー

StorageGRIDシステムは、 "Amazon Simple Storage Service API Reference：共通リクエストヘッダー"で定義される一般的なリクエストヘッダーをサポートしていますが、例外が1つあります。

リクエストヘッダー	導入
許可	AWS Signature Version 2 の完全サポート AWS Signature Version 4 をサポートします。ただし、以下の例外があります。 • 実際のペイロードのチェックサム値を `x-amz-content-sha256`に指定すると、その値は検証なしで受け入れられます。これは、値 `UNSIGNED-PAYLOAD`がヘッダーに指定された場合と同様です。`x-amz-content-sha256`ヘッダー値にストリーミング（例：STREAMING-AWS4-HMAC-SHA256-PAYLOAD）を意味する `aws-chunked`値を指定した場合、チャンク署名はチャンクデータに対して検証されません。

一般的なレスポンスヘッダー

StorageGRID システムは、*Simple Storage Service API Reference* で定義されているすべての一般的なレスポンスヘッダーをサポートしていますが、例外が1つあります。

レスポンスヘッダー	導入
x-amz-id-2	未使用

StorageGRID S3 REST APIによるリクエストの認証方法

StorageGRID システムは、S3 API を使用したオブジェクトへの認証済みアクセスと匿名アクセスの両方をサポートしています。

S3 APIは、S3 APIリクエストの認証において、署名バージョン2と署名バージョン4をサポートしています。

認証済みリクエストは、アクセスキーIDと秘密アクセスキーを使用して署名する必要があります。

StorageGRID システムは、HTTP `Authorization`ヘッダーとクエリパラメータの使用という2つの認証方法をサポートしています。

HTTP Authorizationヘッダーを使用する

HTTP `Authorization`ヘッダーは、バケットポリシーで許可されている匿名アクセスリクエストを除き、すべてのS3 API操作で使用されます。 `Authorization`ヘッダーには、リクエストを認証するために必要な署名情報がすべて含まれています。

クエリパラメータを使用する

クエリパラメータを使用すると、URLに認証情報を追加できます。これはURLの事前署名と呼ばれ、特定のリソースへの一時的なアクセスを許可するために使用できます。署名付きURLを持つユーザーは、リソースにアクセスするために秘密アクセスキーを知る必要がないため、サードパーティに対してリソースへのアクセスを制限できます。

StorageGRID でサポートされているサービス操作

StorageGRID システムは、サービスに対して以下の操作をサポートしています。

処理	導入
ListBuckets （以前は GET Service と呼ばれていました）	すべての Amazon S3 REST API の動作を実装済み。予告なく変更される場合があります。
ストレージ使用量を取得する	StorageGRID "ストレージ使用量を取得する" リクエストは、アカウントが使用しているストレージの総量と、そのアカウントに関連付けられている各バケットのストレージ容量を示します。これは、パスが / でカスタムクエリパラメータ (`?x-ntap-sg-usage`) が追加されたサービスに対する操作です。

処理	導入
OPTIONS /	クライアントアプリケーションは、S3 認証クレデンシャルを提供せずにストレージノードの S3 ポートへ `OPTIONS /`リクエストを発行して、ストレージノードが利用可能かどうかを確認できます。このリクエストは、監視目的、または外部ロードバランサーがストレージノードのダウンを検知できるようにするために使用できます。

StorageGRIDにおけるS3バケットの操作と実装の詳細

StorageGRIDシステムは、S3テナントアカウントごとに最大5,000個のバケットをサポートします。

各グリッドには最大100,000個のバケットを設定できます。

5,000個のバケットをサポートするには、グリッド内の各ストレージノードに最低64 GBのRAMが必要です。

バケット名の制限は AWS US Standard リージョンの制限に従いますが、S3 仮想ホスト型リクエストをサポートするために、さらに DNS 命名規則に基づいて制限する必要があります。

詳細については、次を参照してください。

- "Amazon Simple Storage Service ユーザーガイド：バケットのクォータ、制限、および制約"
- "S3エンドポイントのドメイン名を設定する"

ListObjects（GET Bucket）およびListObjectVersions（GET Bucket object versions）操作は StorageGRID "一貫性値"をサポートしています。

個々のバケットについて、最終アクセス時刻の更新が有効になっているか無効になっているかを確認できます。"GET Bucketの最終アクセス時間"を参照してください。

次の表は、StorageGRID が S3 REST API バケット操作を実装する方法を示しています。これらの操作を実行するには、アカウントに必要なアクセス認証情報を提供する必要があります。

処理	導入
CreateBucket	新しいバケットを作成します。バケットを作成すると、バケットの所有者になります。 • バケット名は、以下の規則に従う必要があります： ◦ 各 StorageGRID システム全体で一意である必要があります（テナントアカウント内だけでなく）。 ◦ DNS規格に準拠している必要があります。 ◦ 3文字以上63文字以内である必要があります。 ◦ 1つまたは複数のラベルの連続で、隣接するラベルはピリオドで区切られます。各ラベルは小文字の英字または数字で始まり、小文字の英字または数字で終わる必要があり、使用できる文字は小文字の英字、数字、およびハイフンのみです。 ◦ テキスト形式のIPアドレスのように見えてはなりません。 ◦ 仮想ホスト型リクエストではピリオドを使用しないでください。ピリオドは、サーバーのワイルドカード証明書の検証に問題を引き起こします。 • デフォルトでは、バケットは `us-east-1`リージョンに作成されます。ただし、リクエスト本文の `LocationConstraint`リクエスト要素を使用して、別のリージョンを指定することができます。 `LocationConstraint`要素を使用する場合は、Grid Manager または Grid Management API を使用して定義されたリージョンの正確な名前を指定する必要があります。使用すべきリージョン名がわからない場合は、システム管理者にお問い合わせください。 注：CreateBucketリクエストで StorageGRID に定義されていないリージョンを使用すると、エラーが発生します。 • `x-amz-bucket-object-lock-enabled`リクエストヘッダーを含めることで、S3 Object Lock を有効にしたバケットを作成できます。"S3 REST APIを使用してS3オブジェクトロックを設定する"を参照してください。 バケットを作成する際には、S3 Object Lockを有効にする必要があります。バケット作成後にS3 Object Lockを追加または無効にすることはできません。S3 Object Lockではバケットのバージョン管理が必要ですが、これはバケットを作成する際に自動的に有効になります。
DeleteBucket	バケットを削除します。
DeleteBucketCors	バケットのCORS設定を削除します。
DeleteBucketEncryption	バケットからデフォルトの暗号化を削除します。既存の暗号化されたオブジェクトは暗号化されたままですが、バケットに新しく追加されたオブジェクトは暗号化されません。
DeleteBucketLifecycle	バケットからライフサイクル設定を削除します。"S3ライフサイクル構成を作成する"を参照してください。

処理	導入
DeleteBucketPolicy	バケットに紐付けられているポリシーを削除します。
DeleteBucketReplication	バケットに紐付けられているレプリケーション構成を削除します。
DeleteBucketTagging	`tagging`サブリソースを使用して、バケットからすべてのタグを削除します。 注意：このバケットにデフォルト以外の ILM ポリシータグが設定されている場合、値が割り当てられた `NTAP-SG-ILM-BUCKET-TAG`バケットタグが存在します。 `NTAP-SG-ILM-BUCKET-TAG`バケットタグが存在する場合は、DeleteBucketTaggingリクエストを発行しないでください。代わりに、`NTAP-SG-ILM-BUCKET-TAG`タグとその割り当てられた値のみを指定してPutBucketTaggingリクエストを発行し、バケットから他のすべてのタグを削除してください。 `NTAP-SG-ILM-BUCKET-TAG`バケットタグを変更または削除しないでください。
GetBucketAcl	肯定的な応答と、バケット所有者のID、DisplayName、および権限を返します。これは、所有者がバケットへの完全なアクセス権を持っていることを示します。
GetBucketCors	バケットの `cors`設定を返します。
GetBucketEncryption	バケットのデフォルトの暗号化設定を返します。
GetBucketLifecycleConfiguration （以前は GET Bucket lifecycle と呼ばれていました）	バケットのライフサイクル構成を返します。"S3ライフサイクル構成を作成する"を参照してください。
GetBucketLocation	`LocationConstraint`要素を使用してCreateBucketリクエストで設定されたリージョンを返します。バケットのリージョンが `us-east-1`の場合、リージョンには空の文字列が返されます。
GetBucketNotificationConfiguration （以前はGET Bucket notificationという名称でした）	バケットに添付されている通知設定を返します。
GetBucketPolicy	バケットに添付されているポリシーを返します。

処理	導入
GetBucketReplication	バケットに添付されているレプリケーション構成を返します。
GetBucketTagging	`tagging`サブリソースを使用して、バケットのすべてのタグを返します。 注意：このバケットにデフォルト以外の ILM ポリシータグが設定されている場合、値が割り当てられた `NTAP-SG-ILM-BUCKET-TAG`バケットタグが存在します。このタグを変更または削除しないでください。
GetBucketVersioning	この実装では、 `versioning`サブリソースを使用してバケットのバージョン管理状態を返します。 • *blank*: バージョン管理が有効になっていません（バケットは「バージョン管理なし」です） • 有効：バージョン管理が有効になっています • 一時停止：バージョン管理は以前は有効でしたが、現在は一時停止されています
GetObjectLockConfiguration	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が設定されている場合は、それらを返します。 "S3 REST APIを使用してS3オブジェクトロックを設定する"を参照してください。
HeadBucket	バケットが存在するかどうか、また、それにアクセスする権限があるかどうかを判断します。 この操作は以下を返します： • `x-ntap-sg-bucket-id`：UUID形式のバケットのUUID。 • `x-ntap-sg-trace-id`：関連するリクエストの一意のトレースID。

処理	導入
ListObjectsおよびListObjectsV2 （以前はGET Bucketという名称でした）	バケット内のオブジェクトの一部または全部（最大1,000個）を返します。オブジェクトのストレージクラスは、オブジェクトが `REDUCED_REDUNDANCY` ストレージクラスオプションを使用して取り込まれた場合でも、2つの値のいずれかを持つことができます： • `STANDARD`これは、オブジェクトがストレージノードで構成されるストレージプールに格納されていることを示します。 • `GLACIER`これは、オブジェクトが Cloud Storage Pool によって指定された外部バケットに移動されたことを示します。 バケットに同じプレフィックスを持つ削除済みキーが多数含まれている場合、レスポンスにはキーを含まない `CommonPrefixes`が含まれる場合があります。 HeadObject および ListObject リクエストの場合、StorageGRID は以下の例に示すように異なる精度で LastModified タイムスタンプを返しますが、AWS は同じ精度でタイムスタンプを返します。 • StorageGRID HeadObject: 「LastModified」: 「2024-09-26T16:43:24+00:00」 • StorageGRID ListObject: 「LastModified」: 「2024-09-26T16:43:24.931000+00:00」 • AWS HeadObject："LastModified"："2023-10-17T00:19:54+00:00" • AWS ListObject："LastModified"："2023-10-17T00:19:54+00:00"
ListObjectVersions （以前は GET Bucket Object versions と呼ばれていました）	バケットへの READ アクセスがある場合、この操作を `versions`サブリソースとともに使用すると、バケット内のオブジェクトのすべてのバージョンのメタデータが一覧表示されます。
PutBucketCors	バケットのCORS設定を行い、バケットがクロスオリジンリクエストを処理できるようにします。クロスオリジンリソース共有（CORS）は、あるドメインのクライアントWebアプリケーションが別のドメインのリソースにアクセスできるようにするセキュリティメカニズムです。例えば、グラフィックを保存するために `images`という名前のS3バケットを使用するとします。 `images`バケットのCORS設定を行うことで、そのバケット内の画像をウェブサイト `http://www.example.com`に表示させることができます。

処理	導入
PutBucketEncryption	既存のバケットのデフォルトの暗号化状態を設定します。バケットレベルの暗号化が有効になっている場合、バケットに追加された新しいオブジェクトはすべて暗号化されます。StorageGRID は、StorageGRID が管理するキーを使用したサーバー側暗号化をサポートします。サーバー側の暗号化設定ルールを指定する場合は、 `SSEAlgorithm`パラメータを `AES256`に設定し、 `KMSMasterKeyID`パラメータは使用しないでください。 オブジェクトアップロード要求で既に暗号化が指定されている場合（つまり、要求に `x-amz-server-side-encryption-*`リクエストヘッダーが含まれている場合）、バケットのデフォルトの暗号化設定は無視されます。
PutBucketLifecycleConfiguration （以前は PUT Bucket lifecycle と呼ばれていました）	バケットの新しいライフサイクル構成を作成するか、既存のライフサイクル構成を置き換えます。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（日数、日付、ExpiredObjectDeleteMarker） • NoncurrentVersionExpiration（NewerNoncurrentVersions、NoncurrentDays） • フィルター（プレフィックス、タグ） • ステータス • ID StorageGRID は以下のアクションをサポートしていません： • AbortIncompleteMultipartUpload • 移行 "S3ライフサイクル構成を作成する"を参照してください。バケットライフサイクルの有効期限アクションがILM配置指示とどのように相互作用するかを理解するには、"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。 注：バケットライフサイクル構成は、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構成はサポートされていません。

処理	導入

PutBucketNotificationConfiguration
 	
（以前はPUT Bucket notificationと呼ばれていました）
	リクエスト本文に含まれる通知設定XMLを使用して、バケットの通知を設定します。以下の実装の詳細についてご留意ください：
	
	• StorageGRID は、Amazon Simple Notification Service (Amazon SNS) トピック、Kafka トピック、または Webhook エンドポイントを宛先としてサポートします。Simple Queue Service (SQS) または AWS Lambda のエンドポイントはサポートされていません。
	• 通知の宛先は、StorageGRID エンドポイントの URN として指定する必要があります。エンドポイントは、Tenant Manager または Tenant Management API を使用して作成できます。
	
	通知設定を成功させるには、エンドポイントが存在している必要があります。エンドポイントが存在しない場合、 `400 Bad Request`エラーがコード `InvalidArgument`とともに返されます。
	
	• 以下のイベントタイプについては、通知を設定できません。これらのイベントタイプはサポートされていません。
	◦ `s3:ReducedRedundancyLostObject`
	◦ `s3:ObjectRestore:Completed`
	• StorageGRID から送信されるイベント通知は標準の JSON 形式を使用しますが、一部のキーは含まれず、その他のキーには特定の値が使用されます。詳細は以下のリストを参照してください。
	◦ **eventSource**
	
	`sgws:s3`
	
	◦ **awsRegion**
	
	含まれていません
	
	◦ **x-amz-id-2**
	
	含まれていません
	
	◦ **arn**
	
	`urn:sgws:s3:::bucket_name`

PutBucketPolicy	バケットに適用されるポリシーを設定します。"バケットおよびグループのアクセスポリシーを使用する"を参照してください。

処理	導入
PutBucketReplication	リクエスト本文で提供されるレプリケーション設定XMLを使用して、バケットの"StorageGRID CloudMirror レプリケーション"を設定します。CloudMirrorレプリケーションについては、以下の実装の詳細に注意してください： • StorageGRID はレプリケーション設定の V1 のみをサポートしています。つまり、StorageGRID はルールに対して `Filter`要素の使用をサポートしておらず、オブジェクトバージョンの削除については V1 の規約に従います。詳細については、 "Amazon Simple Storage Service ユーザーガイド：レプリケーション設定"を参照してください。 • バケットのレプリケーションは、バージョン管理されたバケットとバージョン管理されていないバケットの両方で設定できます。 • レプリケーション構成XMLの各ルールで、異なる宛先バケットを指定できます。ソースバケットは、複数の宛先バケットにレプリケートできます。 • 宛先バケットは、Tenant Manager またはTenant Management APIで指定されているStorageGRID エンドポイントのURN として指定する必要があります。参照してください "CloudMirror レプリケーションの設定"。 レプリケーション構成を成功させるには、エンドポイントが存在している必要があります。エンドポイントが存在しない場合、リクエストは `400 Bad Request`として失敗します。エラーメッセージには次のように記載されています： `Unable to save the replication policy. The specified endpoint URN does not exist: URN.` • 設定 XML で `Role`を指定する必要はありません。この値は StorageGRID では使用されず、送信されても無視されます。 • 構成 XML からストレージクラスを省略すると、StorageGRID はデフォルトで `STANDARD`ストレージクラスを使用します。 • ソースバケットからオブジェクトを削除した場合、またはソースバケット自体を削除した場合、リージョン間レプリケーションの動作は次のとおりです： ◦ オブジェクトまたはバケットが複製される前に削除した場合、オブジェクト/バケットは複製されず、通知もされません。 ◦ オブジェクトまたはバケットが複製された後に削除した場合、StorageGRID はリージョン間レプリケーションの V1 に対する標準の Amazon S3 削除動作に従います。

処理	導入
PutBucketTagging	`tagging`サブリソースを使用して、バケットのタグセットを追加または更新します。バケットタグを追加する際は、以下の制限事項にご注意ください。 • StorageGRID と Amazon S3 はどちらも、各バケットにつき最大 50 個のタグをサポートします。 • バケットに関連付けられたタグには、一意のタグキーが必要です。タグキーは最大 128 文字の Unicode 文字で構成できます。 • タグの値は、最大256文字のUnicode文字を使用できます。 • キーと値は大文字と小文字を区別します。 注意：このバケットにデフォルト以外の ILM ポリシータグが設定されている場合、値が割り当てられた `NTAP-SG-ILM-BUCKET-TAG`バケットタグが存在します。割り当てられた値を持つ `NTAP-SG-ILM-BUCKET-TAG`バケットタグが、すべての PutBucketTagging リクエストに含まれていることを確認してください。このタグを変更または削除しないでください。 注：この操作を行うと、バケットに既に存在するタグはすべて上書きされます。セットから既存のタグが省略された場合、それらのタグはバケットから削除されます。
PutBucketVersioning	`versioning`サブリソースを使用して、既存のバケットのバージョン管理状態を設定します。バージョン管理の状態は、以下のいずれかの値で設定できます。 • 有効：バケット内のオブジェクトのバージョン管理を有効にします。バケットに追加されたすべてのオブジェクトには、一意のバージョン ID が付与されます。 • 一時停止: バケット内のオブジェクトのバージョン管理を無効にします。バケットに追加されたすべてのオブジェクトにはバージョンID `null`が付与されます。
PutObjectLockConfiguration	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または削除します。 デフォルトの保持期間が変更された場合、既存の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。 詳細については、"S3 REST APIを使用してS3オブジェクトロックを設定する"を参照してください。

オブジェクトに対する操作

StorageGRIDがオブジェクトの**S3 REST API**操作を実装する方法

このセクションでは、StorageGRID システムがオブジェクトに対して S3 REST API 操作を実装する方法について説明します。

以下の条件は、すべてのオブジェクト操作に適用されます。

- StorageGRID "一貫性値" は、以下の操作を除き、オブジェクトに対するすべての操作でサポートされています。
  - GetObjectAcl
  - `OPTIONS /`
  - PutObjectLegalHold
  - PutObjectRetention
  - SelectObjectContent
- 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、「最新の要求が優先される」方式で解決されます。「最新の要求が優先される」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に基づきます。
- StorageGRID バケット内のすべてのオブジェクトは、匿名ユーザーまたは別のアカウントによって作成されたオブジェクトを含め、バケットの所有者が所有します。

次の表は、StorageGRID が S3 REST API オブジェクト操作を実装する方法を示しています。

処理	導入
DeleteObject	DeleteObjectリクエストを処理する際、StorageGRIDはオブジェクトのすべてのコピーを、保存されているすべての場所から直ちに削除しようとします。成功した場合、StorageGRIDはクライアントに即座に応答を返します。すべてのコピーを30秒以内に削除できない場合（たとえば、場所が一時的に利用できない場合）、StorageGRIDは削除対象のコピーをキューに追加し、その後クライアントに成功を通知します。 制限事項 • 多要素認証（MFA）とレスポンスヘッダー `x-amz-mfa`はサポートされていません。 • `If-None`および `If-None-Match`ヘッダーは受け入れられますが、機能しません。 バージョン管理 特定のバージョンを削除するには、リクエスト者はバケットの所有者である必要があり、 `versionId`サブリソースを使用する必要があります。このサブリソースを使用すると、バージョンが完全に削除されます。 `versionId`が削除マーカーに対応する場合、レスポンスヘッダー `x-amz-delete-marker`は `true`に設定されて返されます。 • バージョン管理が有効になっているバケットで `versionId`サブリソースを指定せずにオブジェクトを削除すると、削除マーカーが生成されます。削除マーカーの `versionId`は `x-amz-version-id`レスポンスヘッダーを使用して返され、 `x-amz-delete-marker`レスポンスヘッダーは `true`に設定されて返されます。 • バージョン管理が停止されているバケットで `versionId`サブリソースを指定せずにオブジェクトを削除すると、既に存在する「null」バージョンまたは「null」削除マーカーが完全に削除され、新しい「null」削除マーカーが生成されます。 `x-amz-delete-marker`レスポンスヘッダーは `true`に設定されて返されます。 注：場合によっては、オブジェクトに複数の削除マーカーが存在する可能性があります。 GOVERNANCE モードでオブジェクトのバージョンを削除する方法については、"S3 REST APIを使用してS3オブジェクトロックを設定する"を参照してください。
DeleteObjects （以前は「複数オブジェクトの削除」という名称でした）	多要素認証（MFA）とレスポンスヘッダー `x-amz-mfa`はサポートされていません。 同じリクエストメッセージ内で複数のオブジェクトを削除できます。 GOVERNANCE モードでオブジェクトのバージョンを削除する方法については、"S3 REST APIを使用してS3オブジェクトロックを設定する"を参照してください。

処理	導入
DeleteObjectTagging	`tagging`サブリソースを使用して、オブジェクトからすべてのタグを削除します。 バージョン管理 `versionId`クエリパラメータがリクエストに指定されていない場合、この操作はバージョン管理されたバケット内のオブジェクトの最新バージョンからすべてのタグを削除します。オブジェクトの現在のバージョンが削除マーカーである場合、「MethodNotAllowed」ステータスが返され、`x-amz-delete-marker`レスポンスヘッダーは`true`に設定されます。
GetObject	"GetObject"
GetObjectAcl	アカウントに必要なアクセス認証情報が提供された場合、操作は肯定応答とID、DisplayName、およびオブジェクト所有者のPermissionを返し、所有者がオブジェクトへの完全なアクセス権を持っていることを示します。
GetObjectLegalHold	"S3 REST APIを使用してS3オブジェクトロックを設定する"
GetObjectRetention	"S3 REST APIを使用してS3オブジェクトロックを設定する"
GetObjectTagging	`tagging`サブリソースを使用して、オブジェクトのすべてのタグを返します。 バージョン管理 `versionId`クエリパラメータがリクエストで指定されていない場合、この操作はバージョン管理されたバケット内のオブジェクトの最新バージョンからすべてのタグを返します。オブジェクトの現在のバージョンが削除マーカーである場合、「MethodNotAllowed」ステータスが返され、`x-amz-delete-marker`レスポンスヘッダーは`true`に設定されます。
HeadObject	"HeadObject"
RestoreObject	"RestoreObject"
PutObject	"PutObject"
CopyObject （以前は PUT Object - Copy という名称でした）	"CopyObject"

処理	導入
PutObjectLegalHold	"S3 REST APIを使用してS3オブジェクトロックを設定する"
PutObjectRetention	"S3 REST APIを使用してS3オブジェクトロックを設定する"
PutObjectTagging	`tagging`サブリソースを使用して、既存のオブジェクトに一連のタグを追加します。 オブジェクトタグの制限 新規オブジェクトをアップロードする際にタグを追加することも、既存のオブジェクトにタグを追加することもできます。StorageGRID と Amazon S3 はどちらも、各オブジェクトにつき最大 10 個のタグをサポートします。オブジェクトに関連付けられたタグには、一意のタグキーが必要です。タグキーは最大 128 Unicode 文字、タグ値は最大 256 Unicode 文字で構成できます。キーと値は大文字と小文字を区別します。 タグの更新と取り込み動作 PutObjectTagging を使用してオブジェクトのタグを更新する場合、StorageGRID はオブジェクトを再取り込みしません。これは、一致する ILM ルールで指定された「取り込み動作」オプションが使用されないことを意味します。更新によってトリガーされるオブジェクト配置の変更は、通常のバックグラウンド ILM プロセスによって ILM が再評価された際に反映されます。 これは、ILMルールで取り込み動作に「Strict」オプションを使用している場合、必要なオブジェクトの配置ができない場合（たとえば、新たに必要となった場所が利用できない場合など）は、何もアクションが実行されないことを意味します。更新されたオブジェクトは、必要な配置が可能になるまで、現在の配置を維持します。 競合の解決 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、「最新の要求が優先される」方式で解決されます。「最新の要求が優先される」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に基づきます。 バージョン管理 `versionId`クエリパラメータがリクエストに指定されていない場合、この操作はバージョン管理されたバケット内のオブジェクトの最新バージョンにタグを追加します。オブジェクトの現在のバージョンが削除マーカーである場合、「MethodNotAllowed」ステータスが返され、 `x-amz-delete-marker`レスポンスヘッダーが `true`に設定されます。
SelectObjectContent	"SelectObjectContent"

StorageGRID でサポートされている **Amazon S3 Select** 操作

StorageGRID は、"SelectObjectContent コマンド"に対して以下の Amazon S3 Select 句、データ型、および演算子をサポートしています。

記載されていない項目はサポート対象外です。

構文については、"SelectObjectContent"を参照してください。S3 Selectの詳細については、 "S3 Select のAWSドキュメント"を参照してください。

S3 Select が有効になっているテナントアカウントのみが SelectObjectContent クエリを発行できます。"S3 Selectを使用する際の考慮事項と要件" を参照してください。

条項

- SELECTリスト
- FROM句
- WHERE句
- LIMIT句

データ型

- ブール値
- integer
- string
- float
- 小数、数値
- タイムスタンプ

有人対応

論理演算子

- および
- NOT
- または

比較演算子

- <
- >
- ⇐
- >=
- =

- =
- <>
- !=
- 間
- で

パターンマッチング演算子

- LIKE
- _
- %

ユニタリ演算子

- IS NULL
- IS NOT NULL

数学演算子

- +
- -
- *
- /
- %

StorageGRIDはAmazon S3 Selectの演算子の優先順位に従います。

集計関数

- AVG()
- COUNT(*)
- MAX()
- MIN()
- SUM()

条件関数

- ケース
- COALESCE
- NULLIF

変換関数

- CAST（サポートされているデータ型の場合）

日付関数

- DATE_ADD
- DATE_DIFF
- EXTRACT
- TO_STRING
- TO_TIMESTAMP
- UTCNOW

文字列関数

- CHAR_LENGTH、CHARACTER_LENGTH
- LOWER
- SUBSTRING
- TRIM
- アッパー

StorageGRIDがサーバー側の暗号化を使用する方法

サーバー側暗号化により、保存されているオブジェクトデータを保護できます。StorageGRID は、オブジェクトを書き込む際にデータを暗号化し、オブジェクトにアクセスする際にデータを復号化します。

サーバー側暗号化を使用する場合、暗号化キーの管理方法に基づいて、相互に排他的な2つのオプションのいずれかを選択できます：

- **SSE (StorageGRID** 管理キーを使用したサーバー側暗号化**)**：オブジェクトを保存するために S3 リクエストを発行すると、StorageGRID はオブジェクトを固有のキーで暗号化します。オブジェクトを取得するために S3 リクエストを発行すると、StorageGRID は保存されているキーを使用してオブジェクトを復号します。
- **SSE-C**（顧客提供キーによるサーバー側暗号化）：オブジェクトを保存するためにS3リクエストを発行する場合、独自の暗号化キーを提供します。オブジェクトを取得する際には、リクエストの一部として同じ暗号化キーを提供します。2つの暗号化キーが一致した場合、オブジェクトは復号化され、オブジェクトデータが返されます。

  StorageGRID はすべてのオブジェクトの暗号化および復号化操作を管理しますが、提供する暗号化キーはユーザーが管理する必要があります。

> ご提供いただいた暗号化キーは一切保存されません。暗号化キーを紛失した場合、対応するオブジェクトも失われます。

> オブジェクトがSSEまたはSSE-Cで暗号化されている場合、バケットレベルまたはグリッドレベルの暗号化設定はすべて無視されます。

SSE を使用

StorageGRID が管理する一意の暗号化キーでオブジェクトを暗号化するには、次のリクエストヘッダーを使用します：

```
x-amz-server-side-encryption
```

SSEリクエストヘッダーは、以下のオブジェクト操作でサポートされています。

- "PutObject"
- "CopyObject"
- "CreateMultipartUpload"

SSE-Cを使用する

管理する一意のキーでオブジェクトを暗号化するには、次の3つのリクエストヘッダーを使用します：

リクエストヘッダー	説明
`x-amz-server-side-encryption-customer-algorithm`	暗号化アルゴリズムを指定してください。ヘッダー値は `AES256`である必要があります。
`x-amz-server-side-encryption-customer-key`	オブジェクトの暗号化または復号化に使用する暗号化キーを指定します。キーの値は 256 ビットで、Base64 エンコードされている必要があります。
`x-amz-server-side-encryption-customer-key-MD5`	RFC 1321に従って暗号化キーのMD5ダイジェストを指定します。これは、暗号化キーがエラーなく送信されたことを確認するために使用されます。MD5ダイジェストの値は、Base64エンコードされた128ビットである必要があります。

SSE-Cリクエストヘッダーは、以下のオブジェクト操作でサポートされています：

- "GetObject"
- "HeadObject"
- "PutObject"
- "CopyObject"
- "CreateMultipartUpload"
- "UploadPart"
- "UploadPartCopy"

顧客提供キーを使用したサーバー側暗号化（SSE-C）の利用に関する考慮事項

SSE-Cを使用する前に、以下の点にご注意ください。

- https を使用する必要があります。

StorageGRID は、SSE-C を使用している場合、HTTP 経由で行われたすべてのリクエストを拒否します。セキュリティ上の理由から、誤って HTTP を使用して送信したキーはすべて漏洩している可能性があるとみなしてください。そのキーは破棄し、必要に応じてローテーションしてください。

- レスポンスに含まれるETagは、オブジェクトデータのMD5ではありません。
- 暗号化キーとオブジェクトのマッピングを管理する必要があります。StorageGRID は暗号化キーを保存しません。各オブジェクトに対して提供する暗号化キーを追跡する責任はお客様にあります。
- バケットでバージョン管理が有効になっている場合、各オブジェクトバージョンには独自の暗号化キーが必要です。各オブジェクトバージョンに使用される暗号化キーの追跡はお客様の責任となります。
- 暗号化キーをクライアント側で管理するため、鍵のローテーションなどの追加の安全対策もクライアント側で管理する必要があります。

ご提供いただいた暗号化キーは一切保存されません。暗号化キーを紛失した場合、対応するオブジェクトも失われます。

- バケットにクロスグリッドレプリケーションまたはCloudMirrorレプリケーションが設定されている場合、SSE-C オブジェクトを取り込むことはできません。取り込み処理は失敗します。

関連情報

"Amazon S3ユーザーガイド：カスタマー提供のキーを使用したサーバー側の暗号化（SSE-C）の使用"

S3 CopyObjectリクエストを使用して**StorageGRID**でオブジェクトのコピーを作成する

S3 CopyObject リクエストを使用して、S3 にすでに保存されているオブジェクトのコピーを作成できます。CopyObject 操作は、GetObject に続いて PutObject を実行することと同じです。

競合を解決する

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、「最新の要求が優先される」方式で解決されます。「最新の要求が優先される」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に基づきます。

オブジェクトサイズ

単一のPutObjectオペレーションの推奨最大サイズは 5 GiB（5,368,709,120 バイト）です。5 GiB を超えるオブジェクトがある場合は、代わりに"マルチパートアップロード"を使用してください。

単一のPutObject操作でサポートされる最大サイズは 5 TiB （5,497,558,138,880 バイト）です。

StorageGRID 11.6以前からアップグレードした場合、5 GiBを超えるオブジェクトをアップロードしようとすると、「S3 PUT Object size too large」アラートがトリガーされます。StorageGRID 11.7または11.8を新規インストールした場合、この場合はアラートはトリガーされません。ただし、AWS S3標準に準拠するため、StorageGRIDの今後のリリースでは5 GiBを超えるオブジェクトのアップロードはサポートされません。

ユーザーメタデータ内のUTF-8文字

リクエストに、ユーザー定義メタデータのキー名または値に（エスケープされていない）UTF-8値が含まれている場合、StorageGRID の動作は未定義です。

StorageGRID は、ユーザー定義メタデータのキー名または値に含まれるエスケープされた UTF-8 文字を解析または解釈しません。エスケープされた UTF-8 文字は ASCII 文字として扱われます：

- ユーザー定義のメタデータにエスケープされたUTF-8文字が含まれている場合、リクエストは成功します。
- StorageGRID は、キー名または値の解釈された値に印刷不可能な文字が含まれている場合、 `x-amz-missing-meta` ヘッダーを返しません。

サポートされているリクエストヘッダー

以下のリクエストヘッダーがサポートされています：

- `Content-Type`
- `x-amz-copy-source`
- `x-amz-copy-source-if-match`
- `x-amz-copy-source-if-none-match`
- `x-amz-copy-source-if-unmodified-since`
- `x-amz-copy-source-if-modified-since`
- `x-amz-meta-`続いて、ユーザー定義のメタデータを含む名前と値のペア
- `x-amz-metadata-directive`：デフォルト値は `COPY`で、オブジェクトとその関連メタデータをコピーできます。

  オブジェクトをコピーする際に既存のメタデータを上書きするか、オブジェクトのメタデータを更新するかを `REPLACE`指定できます。

- `x-amz-storage-class`
- `x-amz-tagging-directive`：デフォルト値は `COPY`で、オブジェクトとすべてのタグをコピーできます。

  オブジェクトをコピーする際に既存のタグを上書きするか、タグを更新するかを `REPLACE`で指定できます。

- S3オブジェクトロックリクエストヘッダー：
  - `x-amz-object-lock-mode`
  - `x-amz-object-lock-retain-until-date`
  - `x-amz-object-lock-legal-hold`

    これらのヘッダーなしでリクエストが行われた場合、バケットのデフォルトの保持設定を使用して、オブジェクトのバージョンモードと保持期限が計算されます。"S3 REST APIを使用してS3オブジェクトロックを設定する"を参照してください。

- SSEリクエストヘッダー：
  - `x-amz-copy-source-server-side-encryption-customer-algorithm`
  - `x-amz-copy-source-server-side-encryption-customer-key`
  - `x-amz-copy-source-server-side-encryption-customer-key-MD5`
  - `x-amz-server-side-encryption`
  - `x-amz-server-side-encryption-customer-key-MD5`
  - `x-amz-server-side-encryption-customer-key`
  - `x-amz-server-side-encryption-customer-algorithm`

    [サーバー側暗号化のためのリクエストヘッダー]を参照してください。

サポートされていないリクエストヘッダー

以下のリクエストヘッダーはサポートされていません：

- `Cache-Control`
- `Content-Disposition`
- `Content-Encoding`
- `Content-Language`
- `Expires`
- `If-Match`

  `If-Match header`は受け入れられるが、機能しない。

- `If-None-Match`

  `If-None-Match header`は受け入れられるが、機能しない。

- `x-amz-checksum-algorithm`

  オブジェクトをコピーする場合、ソースオブジェクトにチェックサムがあっても、StorageGRID はそのチェックサム値を新しいオブジェクトにコピーしません。この動作は、オブジェクトリクエストで `x-amz-checksum-algorithm`を使用しようとするかどうかに関わらず適用されます。

- `x-amz-website-redirect-location`

ストレージクラスのオプション

`x-amz-storage-class`リクエストヘッダーがサポートされており、一致するILMルールがデュアルコミットまたはバランスlink:../ilm/data-protection-options-for-ingest.html["取り込みオプション"]を使用している場合に、StorageGRIDが作成するオブジェクトコピーの数に影響します。

- STANDARD

  （デフォルト）ILMルールでデュアルコミットオプションを使用する場合、またはバランスオプションが中間コピーの作成にフォールバックする場合に、デュアルコミット取り込み操作を指定します。

- REDUCED_REDUNDANCY

  ILMルールでデュアルコミットオプションを使用する場合、またはバランスオプションが中間コピーの作成にフォールバックする場合に、単一コミットの取り込み操作を指定します。

  > 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、`REDUCED_REDUNDANCY`オプションは無視されます。レガシー準拠バケットにオブジェクトを取り込む場合、`REDUCED_REDUNDANCY`オプションはエラーを返します。StorageGRIDは、コンプライアンス要件を満たすため、常に二重コミットによるデータ取り込みを実行します。

x-amz-copy-source を **CopyObject** で使用する

`x-amz-copy-source`ヘッダーで指定されたソースバケットおよびキーが、コピー先のバケットおよびキーと異なる場合、ソースオブジェクトデータのコピーがコピー先に書き込まれます。

送信元と宛先が一致し、`x-amz-metadata-directive`ヘッダーが`REPLACE`として指定されている場合、オブジェクトのメタデータはリクエストで指定されたメタデータ値で更新されます。この場合、StorageGRID はオブジェクトを再取り込みしません。これには2つの重要な結果があります。

- CopyObjectを使用して、既存のオブジェクトをその場で暗号化したり、既存のオブジェクトの暗号化をその場で変更したりすることはできません。`x-amz-server-side-encryption`ヘッダーまたは`x-amz-server-side-encryption-customer-algorithm`ヘッダーを指定した場合、StorageGRID はリクエストを拒否し、`XNotImplemented`を返します。
- 一致するILMルールで指定されている「取り込み動作」オプションは使用されません。アップデートによって発生するオブジェクト配置の変更は、通常のバックグラウンドILMプロセスによってILMが再評価された際に反映されます。

  これは、ILMルールで取り込み動作に「Strict」オプションを使用している場合、必要なオブジェクトの配置ができない場合（たとえば、新たに必要となった場所が利用できない場合など）は、何もアクションが実行されないことを意味します。更新されたオブジェクトは、必要な配置が可能になるまで、現在の配置を維持します。

サーバー側暗号化のためのリクエストヘッダー

"サーバー側暗号化を使用する"を実行する場合、指定するリクエストヘッダーは、ソースオブジェクトが暗号

化されているかどうか、およびターゲットオブジェクトを暗号化する予定があるかどうかによって異なります。

- ソースオブジェクトが顧客提供キー（SSE-C）を使用して暗号化されている場合、オブジェクトを復号化してコピーできるように、CopyObjectリクエストに次の3つのヘッダーを含める必要があります：
  - `x-amz-copy-source-server-side-encryption-customer-algorithm`：`AES256`を指定します。
  - `x-amz-copy-source-server-side-encryption-customer-key`：ソースオブジェクトの作成時に指定した暗号化キーを指定します。
  - `x-amz-copy-source-server-side-encryption-customer-key-MD5`：ソースオブジェクトを作成した際に指定した MD5 ダイジェストを指定してください。
- 対象オブジェクト（コピー）を、ユーザーが提供および管理する固有のキーで暗号化する場合は、次の 3 つのヘッダーを含めてください。
  - `x-amz-server-side-encryption-customer-algorithm`：`AES256`を指定します。
  - `x-amz-server-side-encryption-customer-key`：対象オブジェクトに新しい暗号化キーを指定します。
  - `x-amz-server-side-encryption-customer-key-MD5`：新しい暗号化キーのMD5ダイジェストを指定します。

> ご提供いただいた暗号化キーは一切保存されません。暗号化キーを紛失した場合、対応するオブジェクトも失われます。オブジェクトデータを保護するために顧客提供のキーを使用する前に、"サーバー側暗号化を使用する"に関する考慮事項を確認してください。

- ターゲットオブジェクト（コピー）を StorageGRID で管理される一意のキーで暗号化する場合（SSE）は、このヘッダーを CopyObject リクエストに含めてください：
  - `x-amz-server-side-encryption`

> `server-side-encryption`オブジェクトの値は更新できません。代わりに、`server-side-encryption`の新しい値を持つコピーを `x-amz-metadata-directive`：`REPLACE`を使用して作成してください。

バージョン管理

ソースバケットがバージョン管理されている場合は、`x-amz-copy-source`ヘッダーを使用してオブジェクトの最新バージョンをコピーできます。オブジェクトの特定のバージョンをコピーするには、`versionId`サブリソースを使用してコピーするバージョンを明示的に指定する必要があります。宛先バケットがバージョン管理されている場合、生成されたバージョンは`x-amz-version-id`レスポンスヘッダーに返されます。対象バケットのバージョン管理が中断されている場合、`x-amz-version-id`は「null」値を返します。

StorageGRID のバケットから GetObject リクエストでオブジェクトを取得する

S3 GetObject リクエストを使用して、S3バケットからオブジェクトを取得できます。

GetObject とマルチパートオブジェクト

`partNumber`リクエストパラメータを使用して、マルチパートオブジェクトまたはセグメント化されたオブジェクトの特定の部分を取得できます。 `x-amz-mp-parts-count`レスポンス要素は、オブジェクトがいくつのパートで構成されているかを示します。

`partNumber`は、セグメント化/マルチパートオブジェクトと非セグメント化/非マルチパートオブジェクトの両方で `1` に設定できますが、 `x-amz-mp-parts-count`レスポンス要素は、セグメント化されたオブジェクトまたはマルチパートオブジェクトの場合にのみ返されます。

ユーザーメタデータ内の**UTF-8**文字

StorageGRID は、ユーザー定義メタデータ内のエスケープされた UTF-8 文字を解析または解釈しません。ユーザー定義メタデータにエスケープされた UTF-8 文字を含むオブジェクトに対する GET リクエストは、キー名または値に印刷不可能な文字が含まれている場合、 `x-amz-missing-meta` ヘッダーを返しません。

サポートされているリクエストヘッダー

以下のリクエストヘッダーがサポートされています：

- `x-amz-checksum-mode`：特定 `ENABLED`

  `Range`ヘッダーは、`GetObject`の `x-amz-checksum-mode`ではサポートされていません。リクエストに `Range`を含め、 `x-amz-checksum-mode`が有効になっている場合、`StorageGRID` はレスポンスにチェックサム値を返しません。

サポートされていないリクエストヘッダー

以下のリクエストヘッダーはサポートされておらず、 `XNotImplemented`を返します：

- `x-amz-website-redirect-location`

バージョン管理

`versionId`サブリソースが指定されていない場合、この操作はバージョン管理されたバケット内のオブジェクトの最新バージョンを取得します。オブジェクトの現在のバージョンが削除マーカーである場合、「見つかりません」ステータスが返され、 `x-amz-delete-marker`レスポンスヘッダーは `true`に設定されます。

顧客提供の暗号化キーを使用したサーバー側暗号化（**SSE-C**）のリクエストヘッダー

オブジェクトが指定した固有のキーで暗号化されている場合は、3つのヘッダーすべてを使用してください。

- `x-amz-server-side-encryption-customer-algorithm`： `AES256`を指定します。

- `x-amz-server-side-encryption-customer-key`：オブジェクトの暗号化キーを指定してください。
- `x-amz-server-side-encryption-customer-key-MD5`：オブジェクトの暗号化キーのMD5ダイジェストを指定します。

> ご提供いただいた暗号化キーは一切保存されません。暗号化キーを紛失した場合、対応するオブジェクトも失われます。オブジェクトデータの保護に顧客提供のキーを使用する前に、"サーバー側暗号化を使用する"の考慮事項を確認してください。

クラウドストレージプールオブジェクトに対する **GetObject** の動作

オブジェクトが"クラウド ストレージ プール"に保存されている場合、GetObjectリクエストの動作はオブジェクトの状態によって異なります。詳細については、"HeadObject"を参照してください。

> オブジェクトがクラウドストレージプールに保存され、かつそのオブジェクトのコピーがグリッド上にも1つ以上存在する場合、GetObjectリクエストは、クラウドストレージプールからデータを取得する前に、まずグリッドからデータを取得しようとします。

オブジェクトの状態	**GetObject** の動作
StorageGRID に取り込まれたがILMによってまだ評価されていないオブジェクト、または従来のストレージプールに保存されているオブジェクト、またはイレイジャー コーディングを使用して保存されているオブジェクト	`200 OK` オブジェクトのコピーが取得されます。
クラウドストレージプール内のオブジェクトだが、まだ取得不可能な状態に移行していない	`200 OK` オブジェクトのコピーが取得されます。
オブジェクトが復元不可能な状態に遷移しました	`403 Forbidden, InvalidObjectState` "RestoreObject" リクエストを使用して、オブジェクトを取得可能な状態に復元します。
復元不可能な状態から復元処理中のオブジェクト	`403 Forbidden, InvalidObjectState` RestoreObject リクエストが完了するまで待ちます。
オブジェクトがクラウドストレージプールに完全に復元されました。	`200 OK` オブジェクトのコピーが取得されます。

クラウドストレージプール内のマルチパートオブジェクトまたはセグメントオブジェクト

マルチパートオブジェクトをアップロードした場合、またはStorageGRIDが大きなオブジェクトをセグメントに分割した場合、StorageGRIDはオブジェクトの一部またはセグメントのサブセットをサンプリングすることにより、そのオブジェクトがクラウドストレージプールで利用可能かどうかを判断します。場合によっては、GetObjectリクエストが、オブジェクトの一部がすでに取得不可能な状態に移行している場合や、オブジ

ェクトの一部がまだ復元されていない場合に、誤って `200 OK`を返すことがあります。

このような場合：

- GetObjectリクエストは一部のデータを返す可能性がありますが、転送の途中で停止する場合があります。
- その後のGetObjectリクエストは `403 Forbidden`を返す可能性があります。

GetObjectおよびクロスグリッド複製

"グリッドフェデレーション"を使用していて、"グリッド間複製"がバケットに対して有効になっている場合、S3クライアントはGetObjectリクエストを発行することでオブジェクトのレプリケーションステータスを確認できます。応答にはStorageGRID固有の `x-ntap-sg-cgr-replication-status`レスポンスヘッダーが含まれ、以下のいずれかの値を持ちます：

Grid	レプリケーションステータス
ソース	• 完了：複製は成功しました。 • 保留中：オブジェクトはまだレプリケートされていません。 • **FAILURE**: レプリケーションが永続的な障害で失敗しました。ユーザーはエラーを解決する必要があります。
デスティネーション	レプリカ：オブジェクトはソースグリッドから複製されました。

StorageGRID は `x-amz-replication-status`ヘッダーをサポートしていません。

StorageGRID の S3 HeadObject リクエストを使用してオブジェクトからメタデータを取得する

S3 HeadObject リクエストを使用して、オブジェクト自体を返さずにオブジェクトからメタデータを取得できます。オブジェクトがクラウドストレージプールに保存されている場合は、HeadObject を使用してオブジェクトの移行状態を確認できます。

HeadObject とマルチパートオブジェクト

`partNumber`リクエストパラメータを使用して、マルチパートオブジェクトまたはセグメント化されたオブジェクトの特定部分のメタデータを取得できます。 `x-amz-mp-parts-count`レスポンス要素は、オブジェクトのパート数を示します。

`partNumber`は、セグメント化/マルチパートオブジェクトと非セグメント化/非マルチパートオブジェクトの両方で `1` に設定できますが、 `x-amz-mp-parts-count`レスポンス要素は、セグメント化されたオブジェクトまたはマルチパートオブジェクトの場合にのみ返されます。

ユーザーメタデータ内の**UTF-8**文字

StorageGRID は、ユーザー定義メタデータ内のエスケープされた UTF-8 文字を解析または解釈しません。ユーザー定義メタデータにエスケープされた UTF-8 文字を含むオブジェクトに対する HEAD リクエストは、キー名または値に印刷不可能な文字が含まれている場合、 `x-amz-missing-meta` ヘッダーを返しません。

サポートされているリクエストヘッダー

以下のリクエストヘッダーがサポートされています：

- `x-amz-checksum-mode`

> `partNumber`パラメータと `Range`ヘッダーは、 `x-amz-checksum-mode`では `HeadObject` に対してサポートされていません。リクエストに `x-amz-checksum-mode`を有効にして含める場合、`StorageGRID` はレスポンスにチェックサム値を返しません。

サポートされていないリクエストヘッダー

以下のリクエストヘッダーはサポートされておらず、 `XNotImplemented`を返します：

- `x-amz-website-redirect-location`

バージョン管理

> `versionId`サブリソースが指定されていない場合、この操作はバージョン管理されたバケット内のオブジェクトの最新バージョンを取得します。オブジェクトの現在のバージョンが削除マーカーである場合、「見つかりません」ステータスが返され、 `x-amz-delete-marker`レスポンスヘッダーは `true`に設定されます。

顧客提供の暗号化キーを使用したサーバー側暗号化（**SSE-C**）のリクエストヘッダー

オブジェクトが指定した固有のキーで暗号化されている場合は、これら3つのヘッダーをすべて使用してください。

- `x-amz-server-side-encryption-customer-algorithm`： `AES256`を指定します。
- `x-amz-server-side-encryption-customer-key`：オブジェクトの暗号化キーを指定してください。
- `x-amz-server-side-encryption-customer-key-MD5`：オブジェクトの暗号化キーのMD5ダイジェストを指定します。

> ご提供いただいた暗号化キーは一切保存されません。暗号化キーを紛失した場合、対応するオブジェクトも失われます。オブジェクトデータの保護に顧客提供のキーを使用する前に、"サーバー側暗号化を使用する"の考慮事項を確認してください。

HeadObject クラウドストレージプールオブジェクトに対する応答

オブジェクトが"クラウド ストレージ プール"に格納されている場合、次の応答ヘッダーが返されます：

- `x-amz-storage-class: GLACIER`
- `x-amz-restore`

レスポンスヘッダーには、オブジェクトがクラウドストレージプールに移動される際、オプションで復元不可能な状態に移行される際、および復元される際のオブジェクトの状態に関する情報が含まれます。

オブジェクトの状態	**HeadObject** への応答
StorageGRID に取り込まれたがILMによってまだ評価されていないオブジェクト、または従来のストレージプールに保存されているオブジェクト、またはイレイジャー コーディングを使用して保存されているオブジェクト	`200 OK`（特別なレスポンスヘッダーは返されません。）

クラウドストレージプール内のオブジェクトだが、まだ取得不可能な状態に移行していない	`200 OK`
	
	`x-amz-storage-class: GLACIER`
	
	`x-amz-restore: ongoing-request="false", expiry-date="Sat, 23 July 20 2030 00:00:00 GMT"`
	
	オブジェクトが取得不可能な状態に移行するまで、`expiry-date`の値は遠い未来のある時点に設定されています。移行の正確な時間は、StorageGRID システムによって制御されません。

オブジェクトは取得不可能な状態に移行しましたが、グリッド上には少なくとも1つのコピーが存在します。	`200 OK`
	
	`x-amz-storage-class: GLACIER`
	
	`x-amz-restore: ongoing-request="false", expiry-date="Sat, 23 July 20 2030 00:00:00 GMT"`
	
	`expiry-date`の値は、遠い将来のある時点に設定されています。
	
	注: グリッド上のコピーが利用できない場合（たとえば、ストレージノードがダウンしている場合）、オブジェクトを正常に取得するには、クラウドストレージプールからコピーを復元するための "RestoreObject" リクエストを発行する必要があります。

オブジェクトの状態	**HeadObject** への応答

オブジェクトが取得不可能な状態に移行し、グリッド上にコピーが存在しません	`200 OK`
	
	`x-amz-storage-class: GLACIER`

復元不可能な状態から復元処理中のオブジェクト	`200 OK`
	
	`x-amz-storage-class: GLACIER`
	
	`x-amz-restore: ongoing-request="true"`

オブジェクトがクラウドストレージプールに完全に復元されました。	`200 OK`
	
	`x-amz-storage-class: GLACIER`
	
	`x-amz-restore: ongoing-request="false", expiry-date="Sat, 23 July 20 2018 00:00:00 GMT"`
	
	`expiry-date`は、クラウドストレージプール内のオブジェクトがいつ復元不可能な状態に戻されるかを示します。

クラウドストレージプール内のマルチパートオブジェクトまたはセグメントオブジェクト

マルチパートオブジェクトをアップロードした場合、またはStorageGRIDが大きなオブジェクトをセグメントに分割した場合、StorageGRIDはオブジェクトの一部またはセグメントのサブセットをサンプリングすることにより、そのオブジェクトがクラウドストレージプールで利用可能かどうかを判断します。場合によっては、HeadObjectリクエストが誤って返される可能性があります `x-amz-restore: ongoing-request="false"`オブジェクトの一部がすでに取得不可能な状態に移行している場合、またはオブジェクトの一部がまだ復元されていない場合。

HeadObjectおよびクロスグリッド複製

使用している場合は"グリッドフェデレーション"、"グリッド間複製"がバケットに対して有効になっている場合、S3クライアントはHeadObjectリクエストを発行することでオブジェクトのレプリケーションステータスを確認できます。応答にはStorageGRID固有の `x-ntap-sg-cgr-replication-status`レスポンスヘッダーが含まれ、以下のいずれかの値を持ちます：

Grid	レプリケーションステータス
ソース	• 完了：複製は成功しました。 • 保留中：オブジェクトはまだレプリケートされていません。 • **FAILURE**: レプリケーションが永続的な障害で失敗しました。ユーザーはエラーを解決する必要があります。

Grid	レプリケーションステータス
デスティネーション	レプリカ：オブジェクトはソースグリッドから複製されました。

StorageGRID は `x-amz-replication-status`ヘッダーをサポートしていません。

S3 PutObjectリクエストを使用して**StorageGRID**のバケットにオブジェクトを追加する

S3 PutObject リクエストを使用して、バケットにオブジェクトを追加できます。

競合を解決する

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、「最新の要求が優先される」方式で解決されます。「最新の要求が優先される」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に基づきます。

オブジェクトサイズ

単一のPutObjectオペレーションの推奨最大サイズは 5 GiB（5,368,709,120 バイト）です。5 GiB を超えるオブジェクトがある場合は、代わりに"マルチパートアップロード"を使用してください。

単一のPutObject操作でサポートされる最大サイズは 5 TiB （5,497,558,138,880 バイト）です。

StorageGRID 11.6以前からアップグレードした場合、5 GiBを超えるオブジェクトをアップロードしようとすると、「S3 PUT Object size too large」アラートがトリガーされます。StorageGRID 11.7または11.8を新規インストールした場合、この場合はアラートはトリガーされません。ただし、AWS S3標準に準拠するため、StorageGRIDの今後のリリースでは5 GiBを超えるオブジェクトのアップロードはサポートされません。

ユーザーメタデータサイズ

Amazon S3 は、各 PUT リクエストヘッダー内のユーザー定義メタデータのサイズを 2 KB に制限します。StorageGRID はユーザーメタデータを 24 KiB に制限します。ユーザー定義メタデータのサイズは、各キーと値の UTF-8 エンコーディングにおけるバイト数の合計によって測定されます。

ユーザーメタデータ内の**UTF-8**文字

リクエストに、ユーザー定義メタデータのキー名または値に（エスケープされていない）UTF-8値が含まれている場合、StorageGRID の動作は未定義です。

StorageGRID は、ユーザー定義メタデータのキー名または値に含まれるエスケープされた UTF-8 文字を解析または解釈しません。エスケープされた UTF-8 文字は ASCII 文字として扱われます：

- PutObject、CopyObject、GetObject、およびHeadObjectリクエストは、ユーザー定義のメタデータにエスケープされた UTF-8 文字が含まれている場合に成功します。
- StorageGRID は、キー名または値の解釈された値に印刷不可能な文字が含まれている場合、 `x-amz-missing-meta` ヘッダーを返しません。

オブジェクトタグの制限

新規オブジェクトをアップロードする際にタグを追加することも、既存のオブジェクトにタグを追加することもできます。StorageGRID と Amazon S3 はどちらも、各オブジェクトにつき最大 10 個のタグをサポートします。オブジェクトに関連付けられたタグには、一意のタグキーが必要です。タグキーは最大 128 Unicode 文字、タグ値は最大 256 Unicode 文字で構成できます。キーと値は大文字と小文字を区別します。

オブジェクトの所有権

StorageGRID では、所有者以外のアカウントまたは匿名ユーザーによって作成されたオブジェクトを含め、すべてのオブジェクトはバケット所有者アカウントが所有します。

サポートされているリクエストヘッダー

以下のリクエストヘッダーがサポートされています：

- `Cache-Control`
- `Content-Disposition`
- `Content-Encoding`

  に対して `aws-chunked`を指定した場合`Content-Encoding`、StorageGRID は以下の項目を検証しません：

  - StorageGRID は、 `chunk-signature`をチャンクデータと照合して検証しません。
  - StorageGRID は、 `x-amz-decoded-content-length` に指定した値をオブジェクトに対して検証しません。
- `Content-Language`
- `Content-Length`
- `Content-MD5`
- `Content-Type`
- `Expires`
- `Transfer-Encoding`

  チャンク転送エンコーディングは、 `aws-chunked`ペイロード署名も使用されている場合にサポートされます。

- `x-amz-checksum-sha256`
- `x-amz-meta-`、ユーザー定義のメタデータを含む名前と値のペアが続きます。

  ユーザー定義メタデータの名前と値のペアを指定する場合は、次の一般的な形式を使用してください。

  ```
  x-amz-meta-name: value
  ```

  ILM ルールの参照時間として ユーザー定義の作成時間 オプションを使用する場合は、オブジェクトが作成された日時を記録するメタデータの名前として `creation-time`を使用する必要があります。例：

```
x-amz-meta-creation-time: 1443399726
```

`creation-time`の値は、1970年1月1日からの秒数として評価されます。

ILMルールでは、参照時間に「ユーザー定義の作成時間」と「バランス型」または「厳密型」の取り込みオプションの両方を使用することはできません。ILMルールを作成する際にエラーが返されます。

- `x-amz-tagging`
- S3オブジェクトロックリクエストヘッダー
  - `x-amz-object-lock-mode`
  - `x-amz-object-lock-retain-until-date`
  - `x-amz-object-lock-legal-hold`

    これらのヘッダーなしでリクエストが行われた場合、バケットのデフォルトの保持設定を使用して、オブジェクトのバージョンモードと保持期限が計算されます。"S3 REST APIを使用してS3オブジェクトロックを設定する"を参照してください。
- SSEリクエストヘッダー：
  - `x-amz-server-side-encryption`
  - `x-amz-server-side-encryption-customer-key-MD5`
  - `x-amz-server-side-encryption-customer-key`
  - `x-amz-server-side-encryption-customer-algorithm`

    [サーバー側暗号化のためのリクエストヘッダー]を参照してください。

サポートされていないリクエストヘッダー

以下のリクエストヘッダーはサポートされていません：

- `If-Match`

  `If-Match header`は受け入れられるが、機能しない。
- `If-None-Match`

  `If-None-Match header`は受け入れられるが、機能しない。
- `x-amz-acl`

- `x-amz-sdk-checksum-algorithm`
- `x-amz-trailer`
- `x-amz-website-redirect-location`

> `x-amz-website-redirect-location`ヘッダーは `XNotImplemented`を返します。

ストレージクラスのオプション

> `x-amz-storage-class`リクエストヘッダーはサポートされています。 `x-amz-storage-class`に送信された値は、取り込み中に `StorageGRID` がオブジェクトデータを保護する方法に影響しますが、`StorageGRID` システムにオブジェクトの永続的なコピーがいくつ保存されるか（`ILM` によって決定）には影響しません。

取り込まれたオブジェクトに一致する ILM ルールが厳密な取り込みオプションを使用する場合、`x-amz-storage-class`ヘッダーは効果がありません。

以下の値を `x-amz-storage-class`に使用できます：

- `STANDARD`（デフォルト）
  - デュアルコミット：ILMルールで取り込み動作にデュアルコミットオプションが指定されている場合、オブジェクトが取り込まれるとすぐに、そのオブジェクトの2番目のコピーが作成され、別のストレージノードに配布されます（デュアルコミット）。ILMを評価する際、StorageGRIDはこれらの最初の暫定コピーがルールに定められた配置指示を満たしているかどうかを判定します。満たしていない場合は、別の場所に新しいオブジェクトのコピーを作成する必要があり、最初の暫定コピーを削除する必要が生じる場合があります。
  - バランス：ILM ルールでバランスオプションが指定されており、StorageGRID がルールで指定されたすべてのコピーをすぐに作成できない場合、StorageGRID は異なるストレージノード上に2つの暫定コピーを作成します。

    StorageGRID がILM ルールで指定されたすべてのオブジェクトコピーを即座に作成できる場合（同期配置）、`x-amz-storage-class`ヘッダーは効果がありません。
- `REDUCED_REDUNDANCY`
  - デュアルコミット：ILMルールの取り込み動作にデュアルコミットオプションが指定されている場合、StorageGRID はオブジェクトの取り込み時に単一の中間コピーを作成します（シングルコミット）。
  - バランス：ILM ルールでバランス オプションが指定されている場合、StorageGRID はシステムがルールで指定されたすべてのコピーをすぐに作成できない場合にのみ、一時的なコピーを 1 つ作成します。StorageGRID が同期配置を実行できる場合、このヘッダーは効果がありません。`REDUCED_REDUNDANCY`オプションは、オブジェクトに一致する ILM ルールが単一の複製コピーを作成する場合に最適です。この場合、`REDUCED_REDUNDANCY`を使用することで、取り込み操作ごとに不要なオブジェクトコピーの作成と削除をなくすことができます。

`REDUCED_REDUNDANCY`オプションの使用は、その他の状況では推奨されません。`REDUCED_REDUNDANCY`取り込み時のオブジェクトデータ損失のリスクを高めます。たとえば、ILM評価が行われる前に障害が発生したストレージノードに単一のコピーが最初に保存された場合、データが失われる可能性があります。

一定期間にわたって複製コピーが1つしかない場合、データが永久に失われるリスクがあります。オブジェクトの複製コピーが1つしか存在しない場合、ストレージノードが故障したり重大なエラーが発生したりすると、そのオブジェクトは失われます。アップグレードなどのメンテナンス手順中は、一時的にオブジェクトへのアクセスができなくなる場合もあります。

指定する `REDUCED_REDUNDANCY`は、オブジェクトが最初に取り込まれたときに作成されるコピーの数にのみ影響します。これは、アクティブな ILM ポリシーによってオブジェクトが評価されるときに作成されるオブジェクトのコピーの数には影響せず、StorageGRID システムでより低いレベルの冗長性でデータが保存されることもありません。

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、`REDUCED_REDUNDANCY`オプションは無視されます。レガシー準拠バケットにオブジェクトを取り込む場合、`REDUCED_REDUNDANCY`オプションはエラーを返します。StorageGRIDは、コンプライアンス要件を満たすため、常に二重コミットによるデータ取り込みを実行します。

サーバー側暗号化のためのリクエストヘッダー

以下のリクエストヘッダーを使用すると、サーバー側暗号化でオブジェクトを暗号化できます。SSEオプションとSSE-Cオプションは相互に排他的です。

- **SSE**: StorageGRID が管理する一意のキーでオブジェクトを暗号化する場合は、次のヘッダーを使用します。
  - `x-amz-server-side-encryption`

`x-amz-server-side-encryption`ヘッダーが`PutObject`リクエストに含まれていない場合、グリッド全体の`link:../admin/changing-network-options-object-encryption.html["保存されたオブジェクトの暗号化設定"]`は`PutObject`レスポンスから省略されます。

- **SSE-C**：提供および管理する固有のキーを使用してオブジェクトを暗号化する場合は、これら 3 つのヘッダーすべてを使用してください。
  - `x-amz-server-side-encryption-customer-algorithm`： `AES256`を指定します。
  - `x-amz-server-side-encryption-customer-key`：新しいオブジェクトの暗号化キーを指定してください。
  - `x-amz-server-side-encryption-customer-key-MD5`：新しいオブジェクトの暗号化キーのMD5ダイジェストを指定します。

ご提供いただいた暗号化キーは一切保存されません。暗号化キーを紛失した場合、対応するオブジェクトも失われます。オブジェクトデータを保護するために顧客提供のキーを使用する前に、"サーバー側暗号化を使用する"に関する考慮事項を確認してください。

オブジェクトがSSEまたはSSE-Cで暗号化されている場合、バケットレベルまたはグリッドレベルの暗号化設定はすべて無視されます。

バージョン管理

バケットでバージョン管理が有効になっている場合、一意の `versionId`は、保存されるオブジェクトのバージョンに対して自動的に生成されます。この `versionId`は、 `x-amz-version-id`レスポンスヘッダーを使用してレスポンスにも返されます。

バージョン管理が中断された場合、オブジェクトのバージョンはnull `versionId`で保存され、nullバージョンが既に存在する場合は上書きされます。

Authorizationヘッダーの署名計算

`Authorization`ヘッダーを使用してリクエストを認証する場合、StorageGRID は次の点で AWS と異なります：

- StorageGRID は、 `host` ヘッダーを `CanonicalHeaders` 内に含める必要はありません。
- StorageGRID は、 `Content-Type` が `CanonicalHeaders` 内に含まれることを必要としません。
- StorageGRID は、 `x-amz-*` ヘッダーを `CanonicalHeaders` 内に含める必要はありません。

一般的なベストプラクティスとして、これらのヘッダーは常に `CanonicalHeaders`に含めて検証されるようにしてください。ただし、これらのヘッダーを除外しても、StorageGRID はエラーを返しません。

詳細については、 "認証ヘッダーの署名計算：ペイロードを単一のチャンクで転送する（AWS Signature Version 4）"を参照してください。

関連情報

- "ILMを使用してオブジェクトを管理する"
- "Amazon Simple Storage Service API リファレンス：PutObject"

S3 RestoreObjectリクエストを使用して**StorageGRID**でオブジェクトを復元する

S3 RestoreObject リクエストを使用して、Cloud Storage Pool に保存されているオブジェクトをリストアできます。

サポートされているリクエストタイプ

StorageGRID は、オブジェクトを復元するための RestoreObject リクエストのみをサポートしています。 `SELECT` タイプの復元はサポートしていません。Select リクエストは `XNotImplemented`を返します。

バージョン管理

オプションで、 `versionId`を指定して、バージョン管理されたバケット内のオブジェクトの特定バージョンを復元します。 `versionId`を指定しない場合、オブジェクトの最新バージョンが復元されます

クラウドストレージプールオブジェクトに対する **RestoreObject** の動作

オブジェクトが"クラウド ストレージ プール"に保存されている場合、RestoreObjectリクエストは、オブジェクトの状態に基づいて以下の動作をします。詳細については、"HeadObject"を参照してください。

> オブジェクトがクラウドストレージプールに保存され、かつそのオブジェクトのコピーがグリッド上にも1つ以上存在する場合、RestoreObjectリクエストを発行してオブジェクトをリストアする必要はありません。代わりに、GetObjectリクエストを使用して、ローカルコピーを直接取得できます。

オブジェクトの状態	**RestoreObject** の動作
StorageGRID に取り込まれたオブジェクトで、ILM による評価がまだ行われていないか、クラウドストレージプールに存在しないオブジェクト	`403 Forbidden, InvalidObjectState`
クラウドストレージプール内のオブジェクトだが、まだ取得不可能な状態に移行していない	`200 OK`変更は行われません。 注: オブジェクトが取得不可能な状態に移行する前は、その `expiry-date`を変更することはできません。
オブジェクトが復元不可能な状態に遷移しました	`202 Accepted`リクエスト本文で指定された日数だけ、オブジェクトの復元可能なコピーをクラウドストレージプールに復元します。この期間が終了すると、オブジェクトは復元不可能な状態に戻ります。 オプションで、 `Tier`リクエスト要素を使用して、復元ジョブの完了にかかる時間を決定します（`Expedited`、 `Standard`、または `Bulk`）。 `Tier`を指定しない場合、 `Standard`ティアが使用されます。 重要：オブジェクトが S3 Glacier Deep Archive に移行されている場合、またはクラウドストレージプールが Azure Blob ストレージを使用している場合、 `Expedited`階層を使用してオブジェクトを復元することはできません。以下のエラーが返されます `403 Forbidden`、 `InvalidTier`：`Retrieval option is not supported by this storage class`。
復元不可能な状態から復元処理中のオブジェクト	`409 Conflict, RestoreAlreadyInProgress`

オブジェクトの状態	**RestoreObject** の動作
オブジェクトがクラウドストレージプールに完全に復元されました。	`200 OK` 注： オブジェクトが取得可能な状態に復元されている場合、 `Days`に新しい値を指定してRestoreObjectリクエストを再発行することで、その `expiry-date`を変更できます。復元日は、リクエストの時間を基準に更新されます。

S3 SelectObjectContentリクエストを使用して**StorageGRID**でオブジェクトデータをフィルタリングする

S3 SelectObjectContent リクエストを使用して、単純なSQL文に基づいてS3オブジェクトの内容をフィルタリングできます。

詳細については、 "Amazon Simple Storage Service API Reference：SelectObjectContent"をご覧ください。

開始する前に

- テナントアカウントにはS3 Select権限が付与されています。
- クエリ対象のオブジェクトに対する `s3:GetObject`権限が必要です。
- クエリ対象のオブジェクトは、以下のいずれかの形式である必要があります。
  - **CSV**。そのまま使用することも、GZIPまたはBZIP2アーカイブに圧縮して使用することもできます。
  - **Parquet**。Parquetオブジェクトに関する追加要件：
    - S3 Selectは、GZIPまたはSnappyを使用したカラム型圧縮のみをサポートしています。S3 Selectは、Parquetオブジェクトのオブジェクト全体の圧縮をサポートしていません。
    - S3 SelectはParquet形式の出力に対応していません。出力形式はCSVまたはJSONのいずれかを指定する必要があります。
    - 圧縮されていない行グループの最大サイズは 512 MB です。
    - オブジェクトのスキーマで指定されているデータ型を使用する必要があります。
    - INTERVAL、JSON、LIST、TIME、またはUUIDの論理型は使用できません。
- SQL式の最大長は256 KBです。
- 入力または結果に含まれるレコードの最大長は 1 MiB です。

CSVリクエスト構文例

```
POST /{Key+}?select&select-type=2 HTTP/1.1
Host: Bucket.s3.abc-company.com
x-amz-expected-bucket-owner: ExpectedBucketOwner
<?xml version="1.0" encoding="UTF-8"?>
<SelectObjectContentRequest xmlns="http://s3.amazonaws.com/doc/2006-03-
01/">
   <Expression>string</Expression>
   <ExpressionType>string</ExpressionType>
   <RequestProgress>
      <Enabled>boolean</Enabled>
   </RequestProgress>
   <InputSerialization>
      <CompressionType>GZIP</CompressionType>
      <CSV>
         <AllowQuotedRecordDelimiter>boolean</AllowQuotedRecordDelimiter>
         <Comments>#</Comments>
         <FieldDelimiter>\t</FieldDelimiter>
         <FileHeaderInfo>USE</FileHeaderInfo>
         <QuoteCharacter>'</QuoteCharacter>
         <QuoteEscapeCharacter>\\</QuoteEscapeCharacter>
         <RecordDelimiter>\n</RecordDelimiter>
      </CSV>
   </InputSerialization>
   <OutputSerialization>
      <CSV>
         <FieldDelimiter>string</FieldDelimiter>
         <QuoteCharacter>string</QuoteCharacter>
         <QuoteEscapeCharacter>string</QuoteEscapeCharacter>
         <QuoteFields>string</QuoteFields>
         <RecordDelimiter>string</RecordDelimiter>
      </CSV>
   </OutputSerialization>
   <ScanRange>
      <End>long</End>
      <Start>long</Start>
   </ScanRange>
</SelectObjectContentRequest>
```

Parquetリクエスト構文の例

```
POST /{Key+}?select&select-type=2 HTTP/1.1
Host: Bucket.s3.abc-company.com
x-amz-expected-bucket-owner: ExpectedBucketOwner
<?xml version="1.0" encoding="UTF-8"?>
<SelectObjectContentRequest xmlns=http://s3.amazonaws.com/doc/2006-03-01/>
   <Expression>string</Expression>
   <ExpressionType>string</ExpressionType>
   <RequestProgress>
      <Enabled>boolean</Enabled>
   </RequestProgress>
   <InputSerialization>
      <CompressionType>GZIP</CompressionType>
      <PARQUET>
      </PARQUET>
   </InputSerialization>
   <OutputSerialization>
      <CSV>
         <FieldDelimiter>string</FieldDelimiter>
         <QuoteCharacter>string</QuoteCharacter>
         <QuoteEscapeCharacter>string</QuoteEscapeCharacter>
         <QuoteFields>string</QuoteFields>
         <RecordDelimiter>string</RecordDelimiter>
      </CSV>
   </OutputSerialization>
   <ScanRange>
      <End>long</End>
      <Start>long</Start>
   </ScanRange>
</SelectObjectContentRequest>
```

SQLクエリの例

このクエリは、米国国勢調査データから、州名、2010年の人口、2015年の推定人口、および変化率を取得します。ファイル内の州以外のレコードは無視されます。

```
SELECT STNAME, CENSUS2010POP, POPESTIMATE2015, CAST((POPESTIMATE2015 -
CENSUS2010POP) AS DECIMAL) / CENSUS2010POP * 100.0 FROM S3Object WHERE
NAME = STNAME
```

クエリ対象ファイルの最初の数行、 `SUB-EST2020_ALL.csv`は次のようになっています：

```
SUMLEV,STATE,COUNTY,PLACE,COUSUB,CONCIT,PRIMGEO_FLAG,FUNCSTAT,NAME,STNAME,
CENSUS2010POP,
ESTIMATESBASE2010,POPESTIMATE2010,POPESTIMATE2011,POPESTIMATE2012,POPESTIM
ATE2013,POPESTIMATE2014,
POPESTIMATE2015,POPESTIMATE2016,POPESTIMATE2017,POPESTIMATE2018,POPESTIMAT
E2019,POPESTIMATE042020,
POPESTIMATE2020
040,01,000,00000,00000,00000,0,A,Alabama,Alabama,4779736,4780118,4785514,4
799642,4816632,4831586,
4843737,4854803,4866824,4877989,4891628,4907965,4920706,4921532
162,01,000,00124,00000,00000,0,A,Abbeville
city,Alabama,2688,2705,2699,2694,2645,2629,2610,2602,
2587,2578,2565,2555,2555,2553
162,01,000,00460,00000,00000,0,A,Adamsville
city,Alabama,4522,4487,4481,4474,4453,4430,4399,4371,
4335,4304,4285,4254,4224,4211
162,01,000,00484,00000,00000,0,A,Addison
town,Alabama,758,754,751,750,745,744,742,734,734,728,
725,723,719,717
```

AWS-CLIの使用例（CSV）

```
aws s3api select-object-content --endpoint-url https://10.224.7.44:10443
--no-verify-ssl  --bucket 619c0755-9e38-42e0-a614-05064f74126d --key SUB-
EST2020_ALL.csv --expression-type SQL --input-serialization '{"CSV":
{"FileHeaderInfo": "USE", "Comments": "#", "QuoteEscapeCharacter": "\"",
"RecordDelimiter": "\n", "FieldDelimiter": ",", "QuoteCharacter": "\"",
"AllowQuotedRecordDelimiter": false}, "CompressionType": "NONE"}' --output
-serialization '{"CSV": {"QuoteFields": "ASNEEDED",
"QuoteEscapeCharacter": "#", "RecordDelimiter": "\n", "FieldDelimiter":
",", "QuoteCharacter": "\""}}' --expression "SELECT STNAME, CENSUS2010POP,
POPESTIMATE2015, CAST((POPESTIMATE2015 - CENSUS2010POP) AS DECIMAL) /
CENSUS2010POP * 100.0 FROM S3Object WHERE NAME = STNAME" changes.csv
```

出力ファイルの最初の数行、 `changes.csv`は次のようになります：

```
Alabama,4779736,4854803,1.570526070895965802295356898372629785 4
Alaska,710231,738430,3.970398363349389142405780654463125377 5
Arizona,6392017,6832810,6.895992297892824753125656580700583243 1
Arkansas,2915918,2979732,2.188470320495981025529524492801237894 9
California,37253956,38904296,4.429972483996062055798852610444914897 1
Colorado,5029196,5454328,8.453279609703022113276157859029554624 6
```

AWS-CLIの使用例（Parquet）

```
aws s3api select-object-content  -endpoint-url https://10.224.7.44:10443
--bucket 619c0755-9e38-42e0-a614-05064f74126d --key SUB-
EST2020_ALL.parquet --expression "SELECT STNAME, CENSUS2010POP,
POPESTIMATE2015, CAST((POPESTIMATE2015 - CENSUS2010POP) AS DECIMAL) /
CENSUS2010POP * 100.0 FROM S3Object WHERE NAME = STNAME" --expression-type
'SQL' --input-serialization '{"Parquet":{}}'  --output-serialization
'{"CSV": {}}' changes.csv
```

出力ファイル changes.csv の最初の数行は次のようになります：

```
Alabama,4779736,4854803,1.5705260708959658022953568983726297854
Alaska,710231,738430,3.9703983633493891424057806544631253775
Arizona,6392017,6832810,6.8959922978928247531256565807005832431
Arkansas,2915918,2979732,2.1884703204959810255295244928012378949
California,37253956,38904296,4.4299724839960620557988526104449148971
Colorado,5029196,5454328,8.453279609703022113276157859029554624 6
```

マルチパートアップロードの操作

StorageGRIDでサポートされているマルチパートアップロード操作

このセクションでは、StorageGRID がマルチパートアップロードの操作をサポートする方法について説明します。

以下の条件と注意事項は、すべてのマルチパートアップロード操作に適用されます。

- 1 つのバケットへの同時マルチパートアップロードは 1,000 を超えないようにしてください。そのバケットに対する ListMultipartUploads クエリが不完全な結果を返す可能性があるためです。
- StorageGRIDはAWSのマルチパートパーツのサイズ制限を適用します。S3クライアントは、以下のガイドラインに従う必要があります：
  - マルチパートアップロードの各パートは、5 MiB (5,242,880 バイト) から 5 GiB (5,368,709,120 バイト) の範囲内である必要があります。
  - 最後の部分は5 MiB（5,242,880バイト）未満でも構いません。
  - 一般的に、パートのサイズはできるだけ大きくする必要があります。たとえば、100 GiB のオブジェクトには 5 GiB のパートサイズを使用します。各パートは固有のオブジェクトとみなされるため、大きなパートサイズを使用すると、StorageGRID メタデータのオーバーヘッドが削減されます。
  - 5 GiB未満のオブジェクトの場合は、代わりにマルチパートではないアップロードの使用を検討してください。
- ILMは、マルチパートオブジェクトの各部分が取り込まれる際に、またILMルールがBalancedまたはStrictを使用している場合、マルチパートアップロードが完了した際にオブジェクト全体として評価されます"取り込みオプション"。これがオブジェクトやパーツの配置にどのような影響を与えるかを認識しておく必要があります。

- S3マルチパートアップロードの進行中にILMが変更された場合、マルチパートアップロードが完了した時点で、オブジェクトの一部が現在のILM要件を満たさなくなる可能性があります。正しく配置されていないパートはすべてILMによる再評価のためにキューに入れられ、後で正しい場所に移動されます。
- パーツのILMを評価する際、StorageGRIDはオブジェクトのサイズではなく、パーツのサイズに基づいてフィルタリングします。これは、オブジェクト全体に対するILM要件を満たさない場所に、オブジェクトの一部が保存される可能性があることを意味します。例えば、ルールで10GB以上のオブジェクトはすべてDC1に保存され、それより小さいオブジェクトはすべてDC2に保存されるように指定されている場合、10個のパートからなるマルチパートアップロードの各1GBパートは、取り込み時にDC2に保存されます。ただし、ILMをオブジェクト全体として評価する場合、オブジェクトのすべてのパーツがDC1に移動されます。

- すべてのマルチパートアップロード操作は、StorageGRID "一貫性値"をサポートしています。
- マルチパートアップロードを使用してオブジェクトを取り込む場合、"オブジェクト分割しきい値（1 GiB）"は適用されません。
- 必要に応じて、"サーバー側暗号化"をマルチパートアップロードで使用できます。SSE（StorageGRID管理キーを使用したサーバー側暗号化）を使用するには、 `x-amz-server-side-encryption`リクエストヘッダーをCreateMultipartUploadリクエストにのみ含めます。SSE-C（顧客提供キーによるサーバー側暗号化）を使用するには、同じ3つの暗号化キーリクエストヘッダーをCreateMultipartUploadリクエストおよびその後の各UploadPartリクエストに指定します。
- マルチパートアップロードオブジェクトは、取り込みがベースバケットの Before タイムスタンプより前に開始された場合、アップロードが完了したタイミングに関係なく、"ブランチバケット"に含まれます。

処理	導入
AbortMultipartUpload	すべての Amazon S3 REST API の動作を実装済み。予告なく変更される場合があります。
CompleteMultipartUpload	"CompleteMultipartUpload"を参照してください。
CreateMultipartUpload （以前は「マルチパートアップロードの開始」という名称でした）	"CreateMultipartUpload"を参照してください。
ListMultipartUploads	"ListMultipartUploads"を参照してください。
ListParts	すべての Amazon S3 REST API の動作を実装済み。予告なく変更される場合があります。
UploadPart	"UploadPart"を参照してください。
UploadPartCopy	"UploadPartCopy"を参照してください。

StorageGRID S3 REST APIによるマルチパートアップロードの完了方法

CompleteMultipartUpload オペレーションは、以前にアップロードされた各パートを組み立てることで、オブジェクトのマルチパートアップロードを完了します。

StorageGRID は、CompleteMultipartUpload の `partNumber`リクエストパラメータにおいて、昇順での連続しない値をサポートします。パラメータは任意の値から開始できます。

競合を解決する

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、「最新の要求が優先される」方式で解決されます。「最新の要求が優先される」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に基づきます。

サポートされているリクエストヘッダー

以下のリクエストヘッダーがサポートされています：

- `x-amz-checksum-sha256`
- `x-amz-storage-class`

  `x-amz-storage-class`ヘッダーは、一致する `ILM` ルールが`link:../ilm/data-protection-options-for-ingest.html["デュアルコミットまたはバランス型取り込みオプション"]`を指定している場合に `StorageGRID` が作成するオブジェクトコピーの数に影響します。

- `STANDARD`

  （デフォルト）ILMルールでデュアルコミットオプションを使用する場合、またはバランスオプションが中間コピーの作成にフォールバックする場合に、デュアルコミット取り込み操作を指定します。

- `REDUCED_REDUNDANCY`

  ILMルールでデュアルコミットオプションを使用する場合、またはバランスオプションが中間コピーの作成にフォールバックする場合に、単一コミットの取り込み操作を指定します。

  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、`REDUCED_REDUNDANCY`オプションは無視されます。レガシー準拠バケットにオブジェクトを取り込む場合、`REDUCED_REDUNDANCY`オプションはエラーを返します。StorageGRIDは、コンプライアンス要件を満たすため、常に二重コミットによるデータ取り込みを実行します。

マルチパートアップロードが15日以内に完了しない場合、その操作は非アクティブとしてマークされ、関連するすべてのデータがシステムから削除されます。

返される `ETag`値はデータのMD5合計ではなく、マルチパートオブジェクトに対するAmazon S3 APIの `ETag`値の実装に従います。

サポートされていないリクエストヘッダー

以下のリクエストヘッダーはサポートされていません：

- `If-Match`

  `If-Match header`は受け入れられるが、機能しない。

- `If-None-Match`

  `If-None-Match header`は受け入れられるが、機能しない。

- `x-amz-sdk-checksum-algorithm`
- `x-amz-trailer`

バージョン管理

この操作により、複数パートのアップロードが完了します。バケットでバージョン管理が有効になっている場合、マルチパートアップロードの完了後にオブジェクトバージョンが作成されます。

バケットでバージョン管理が有効になっている場合、一意の `versionId`は、保存されるオブジェクトのバージョンに対して自動的に生成されます。この `versionId`は、 `x-amz-version-id`レスポンスヘッダーを使用してレスポンスにも返されます。

バージョン管理が中断された場合、オブジェクトのバージョンはnull `versionId`で保存され、nullバージョンが既に存在する場合は上書きされます。

> バケットでバージョン管理が有効になっている場合、同じオブジェクトキーに対して同時に複数のマルチパートアップロードが完了した場合でも、マルチパートアップロードが完了すると必ず新しいバージョンが作成されます。バケットでバージョン管理が有効になっていない場合、マルチパートアップロードを開始した後、同じオブジェクトキーに対して別のマルチパートアップロードが先に開始され、完了する可能性があります。バージョン管理されていないバケットでは、最後に完了したマルチパートアップロードが優先されます。

レプリケーション、通知、またはメタデータ通知が失敗しました

マルチパートアップロードが行われるバケットがプラットフォームサービス用に構成されている場合、関連付けられたレプリケーションまたは通知アクションが失敗した場合でも、マルチパートアップロードは成功します。

テナントは、オブジェクトのメタデータまたはタグを更新することで、レプリケーションの失敗または通知をトリガーすることができます。テナントは、意図しない変更を避けるために、既存の値を再送信することができます。

"プラットフォームサービスのトラブルシューティング"を参照してください。

StorageGRID での **S3 CreateMultipartUpload** リクエストヘッダーとオプション

CreateMultipartUpload（以前は「マルチパートアップロードの開始」と呼ばれていました）操作は、オブジェクトのマルチパートアップロードを開始し、アップロードIDを返します。

`x-amz-storage-class`リクエストヘッダーはサポートされています。 `x-amz-storage-class`に送信された値は、取り込み中に StorageGRID がオブジェクトデータを保護する方法に影響しますが、StorageGRID システムにオブジェクトの永続的なコピーがいくつ保存されるか（ILM によって決定）には影響しません。

取り込まれたオブジェクトに一致する ILM ルールが Strict "取り込みオプション"を使用する場合、 `x-amz-storage-class`ヘッダーは効果がありません。

以下の値を `x-amz-storage-class`に使用できます：

- STANDARD（デフォルト）
  - デュアルコミット：ILMルールでデュアルコミット取り込みオプションが指定されている場合、オブジェクトが取り込まれるとすぐに、そのオブジェクトの2番目のコピーが作成され、別のストレージノードに配布されます（デュアルコミット）。ILMを評価する際、StorageGRIDはこれらの最初の暫定コピーがルールに定められた配置指示を満たしているかどうかを判定します。満たしていない場合は、別の場所に新しいオブジェクトのコピーを作成する必要があり、最初の暫定コピーを削除する必要が生じる場合があります。
  - バランス：ILM ルールでバランスオプションが指定されており、StorageGRID がルールで指定されたすべてのコピーをすぐに作成できない場合、StorageGRID は異なるストレージノード上に2つの暫定コピーを作成します。

    StorageGRID がILM ルールで指定されたすべてのオブジェクトコピーを即座に作成できる場合（同期配置）、 `x-amz-storage-class`ヘッダーは効果がありません。
- REDUCED_REDUNDANCY
  - デュアルコミット：ILM ルールでデュアルコミットオプションが指定されている場合、StorageGRID はオブジェクトが取り込まれる際に単一の中間コピーを作成します（シングルコミット）。
  - バランス：ILM ルールでバランス オプションが指定されている場合、StorageGRID はシステムがルールで指定されたすべてのコピーをすぐに作成できない場合にのみ、一時的なコピーを 1 つ作成します。StorageGRID が同期配置を実行できる場合、このヘッダーは効果がありません。 `REDUCED_REDUNDANCY`オプションは、オブジェクトに一致する ILM ルールが単一の複製コピーを作成する場合に最適です。この場合、 `REDUCED_REDUNDANCY`を使用することで、取り込み操作ごとに不要なオブジェクトコピーの作成と削除をなくすことができます。

    `REDUCED_REDUNDANCY`オプションの使用は、その他の状況では推奨されません。 `REDUCED_REDUNDANCY`取り込み時のオブジェクトデータ損失のリスクを高めます。たとえば、ILM評価が行われる前に障害が発生したストレージノードに単一のコピーが最初に保存された場合、データが失われる可能性があります。

一定期間にわたって複製コピーが1つしかない場合、データが永久に失われるリスクがあります。オブジェクトの複製コピーが1つしか存在しない場合、ストレージノードが故障したり重大なエラーが発生したりすると、そのオブジェクトは失われます。アップグレードなどのメンテナンス手順中は、一時的にオブジェクトへのアクセスができなくなる場合もあります。

指定する `REDUCED_REDUNDANCY`は、オブジェクトが最初に取り込まれたときに作成されるコピーの数

にのみ影響します。これは、アクティブな ILM ポリシーによってオブジェクトが評価されるときに作成されるオブジェクトのコピーの数には影響せず、StorageGRID システムでより低いレベルの冗長性でデータが保存されることもありません。

> 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、`REDUCED_REDUNDANCY`オプションは無視されます。レガシー準拠バケットにオブジェクトを取り込む場合、`REDUCED_REDUNDANCY`オプションはエラーを返します。StorageGRIDは、コンプライアンス要件を満たすため、常に二重コミットによるデータ取り込みを実行します。

サポートされているリクエストヘッダー

以下のリクエストヘッダーがサポートされています：

- `Content-Type`
- `x-amz-checksum-algorithm`

  現在、 `x-amz-checksum-algorithm`のSHA256値のみがサポートされています。

- `x-amz-meta-`続いて、ユーザー定義のメタデータを含む名前と値のペア

  ユーザー定義メタデータの名前と値のペアを指定する場合は、次の一般的な形式を使用してください。

  ```
  x-amz-meta-_name_: `value`
  ```

  ILM ルールの参照時間として ユーザー定義の作成時間 オプションを使用する場合は、オブジェクトが作成された日時を記録するメタデータの名前として `creation-time`を使用する必要があります。例：

  ```
  x-amz-meta-creation-time: 1443399726
  ```

  `creation-time`の値は、1970年1月1日からの秒数として評価されます。

  > `creation-time`をユーザー定義メタデータとして追加することは、レガシーコンプライアンスが有効になっているバケットにオブジェクトを追加する場合には許可されません。エラーが返されます。

- S3オブジェクトロックリクエストヘッダー：
  - `x-amz-object-lock-mode`
  - `x-amz-object-lock-retain-until-date`
  - `x-amz-object-lock-legal-hold`

    これらのヘッダーなしでリクエストが行われた場合、バケットのデフォルトの保持設定を使用して、オブジェクトバージョンの保持期限が計算されます。

"S3 REST APIを使用してS3オブジェクトロックを設定する"

- SSEリクエストヘッダー：
  - `x-amz-server-side-encryption`
  - `x-amz-server-side-encryption-customer-key-MD5`
  - `x-amz-server-side-encryption-customer-key`
  - `x-amz-server-side-encryption-customer-algorithm`

  [サーバー側暗号化のためのリクエストヘッダー]

  StorageGRID が UTF-8 文字を処理する方法については、"PutObject"を参照してください。

サーバー側暗号化のためのリクエストヘッダー

サーバー側暗号化を使用してマルチパートオブジェクトを暗号化するには、以下のリクエストヘッダーを使用できます。SSEオプションとSSE-Cオプションは相互に排他的です。

- **SSE**: StorageGRID が管理する一意のキーでオブジェクトを暗号化する場合は、CreateMultipartUpload リクエストで次のヘッダーを使用してください。このヘッダーは、UploadPart リクエストには指定しないでください。
  - `x-amz-server-side-encryption`
- **SSE-C**: 提供および管理する固有のキーでオブジェクトを暗号化する場合は、CreateMultipartUploadリクエスト（およびその後の各UploadPartリクエスト）でこれら3つのヘッダーをすべて使用します。
  - `x-amz-server-side-encryption-customer-algorithm`：`AES256`を指定します。
  - `x-amz-server-side-encryption-customer-key`：新しいオブジェクトの暗号化キーを指定してください。
  - `x-amz-server-side-encryption-customer-key-MD5`：新しいオブジェクトの暗号化キーのMD5ダイジェストを指定します。

ご提供いただいた暗号化キーは一切保存されません。暗号化キーを紛失した場合、対応するオブジェクトも失われます。オブジェクトデータを保護するために顧客提供のキーを使用する前に、"サーバー側暗号化を使用する"に関する考慮事項を確認してください。

サポートされていないリクエストヘッダー

以下のリクエストヘッダーはサポートされていません：

- `x-amz-website-redirect-location`

`x-amz-website-redirect-location`ヘッダーは `XNotImplemented`を返します。

バージョン管理

マルチパートアップロードは、アップロードの開始、アップロードの一覧表示、パートのアップロード、アップロードされたパートの組み立て、およびアップロードの完了という、それぞれ独立した操作で構成されます。オブジェクトは、CompleteMultipartUpload操作が実行されると作成されます（該当する場合はバージョン管理されます）。

S3 ListMultipartUploads 操作と **StorageGRID** でサポートされているパラメータ

ListMultipartUploads オペレーションは、バケットに対して進行中のマルチパートアップロードの一覧を表示します。

以下のリクエストパラメータがサポートされています：

- `encoding-type`
- `key-marker`
- `max-uploads`
- `prefix`
- `upload-id-marker`
- `Host`
- `Date`
- `Authorization`

バージョン管理

マルチパートアップロードは、アップロードの開始、アップロードの一覧表示、パートのアップロード、アップロードされたパートの組み立て、およびアップロードの完了という、それぞれ独立した操作で構成されます。オブジェクトは、CompleteMultipartUpload操作が実行されると作成されます（該当する場合はバージョン管理されます）。

StorageGRID の **S3 UploadPart** 処理でサポートされているリクエストヘッダーとサポートされていないリクエストヘッダー

UploadPart オペレーションは、オブジェクトのマルチパートアップロードにおけるパートをアップロードします。

サポートされているリクエストヘッダー

以下のリクエストヘッダーがサポートされています：

- `x-amz-checksum-sha256`
- `Content-Length`
- `Content-MD5`

サーバー側暗号化のためのリクエストヘッダー

CreateMultipartUploadリクエストにSSE-C暗号化を指定した場合は、各UploadPartリクエストに以下のリクエ

ストヘッダーも含める必要があります：

- `x-amz-server-side-encryption-customer-algorithm`： `AES256`を指定します。
- `x-amz-server-side-encryption-customer-key`：CreateMultipartUpload リクエストで指定したものと同じ暗号化キーを指定してください。
- `x-amz-server-side-encryption-customer-key-MD5`：CreateMultipartUpload リクエストで指定したものと同じ MD5 ダイジェストを指定します。

> ご提供いただいた暗号化キーは一切保存されません。暗号化キーを紛失した場合、対応するオブジェクトも失われます。オブジェクトデータの保護に顧客提供のキーを使用する前に、"サーバー側暗号化を使用する"の考慮事項を確認してください。

CreateMultipartUploadリクエスト時にSHA-256チェックサムを指定した場合は、各UploadPartリクエストに次のリクエストヘッダーも含める必要があります：

- `x-amz-checksum-sha256`：この部分の SHA-256 チェックサムを指定します。

サポートされていないリクエストヘッダー

以下のリクエストヘッダーはサポートされていません：

- `x-amz-sdk-checksum-algorithm`
- `x-amz-trailer`

バージョン管理

マルチパートアップロードは、アップロードの開始、アップロードの一覧表示、パートのアップロード、アップロードされたパートの組み立て、およびアップロードの完了という、それぞれ独立した操作で構成されます。オブジェクトは、CompleteMultipartUpload操作が実行されると作成されます（該当する場合はバージョン管理されます）。

S3 UploadPartCopy 操作を使用して **StorageGRID** にオブジェクトの一部をアップロードする

UploadPartCopy オペレーションは、既存のオブジェクトをデータソースとしてデータをコピーすることにより、オブジェクトの一部をアップロードします。

UploadPartCopy操作は、すべての Amazon S3 REST API の動作で実装されています。予告なく変更される場合があります。

このリクエストは、StorageGRIDシステム内の `x-amz-copy-source-range`で指定されたオブジェクトデータを読み書きします。

以下のリクエストヘッダーがサポートされています：

- `x-amz-copy-source-if-match`
- `x-amz-copy-source-if-none-match`
- `x-amz-copy-source-if-unmodified-since`
- `x-amz-copy-source-if-modified-since`

サーバー側暗号化のためのリクエストヘッダー

CreateMultipartUploadリクエストにSSE-C暗号化を指定した場合は、各UploadPartCopyリクエストに以下のリクエストヘッダーも含める必要があります：

- `x-amz-server-side-encryption-customer-algorithm`：`AES256`を指定します。
- `x-amz-server-side-encryption-customer-key`：CreateMultipartUpload リクエストで指定したものと同じ暗号化キーを指定してください。
- `x-amz-server-side-encryption-customer-key-MD5`：CreateMultipartUpload リクエストで指定したものと同じ MD5 ダイジェストを指定します。

ソースオブジェクトが顧客提供キー（SSE-C）を使用して暗号化されている場合、オブジェクトを復号化してコピーできるように、UploadPartCopy リクエストに次の 3 つのヘッダーを含める必要があります：

- `x-amz-copy-source-server-side-encryption-customer-algorithm`：`AES256`を指定します。
- `x-amz-copy-source-server-side-encryption-customer-key`：ソースオブジェクトの作成時に指定した暗号化キーを指定します。
- `x-amz-copy-source-server-side-encryption-customer-key-MD5`：ソースオブジェクトを作成した際に指定した MD5 ダイジェストを指定してください。

> ご提供いただいた暗号化キーは一切保存されません。暗号化キーを紛失した場合、対応するオブジェクトも失われます。オブジェクトデータの保護に顧客提供のキーを使用する前に、"サーバー側暗号化を使用する"の考慮事項を確認してください。

バージョン管理

マルチパートアップロードは、アップロードの開始、アップロードの一覧表示、パートのアップロード、アップロードされたパートの組み立て、およびアップロードの完了という、それぞれ独立した操作で構成されます。オブジェクトは、CompleteMultipartUpload操作が実行されると作成されます（該当する場合はバージョン管理されます）。

StorageGRID S3 REST API エラー応答

StorageGRID システムは、該当するすべての標準 S3 REST API エラーレスポンスをサポートしています。さらに、StorageGRID の実装ではいくつかのカスタムレスポンスが追加されています。

サポートされている**S3 API**エラーコード

Name	HTTPのステータス
AccessDenied	403 Forbidden
BadDigest	400 Bad Request
BucketAlreadyExists	409 Conflict

Name	HTTPのステータス
BucketNotEmpty	409 Conflict
IncompleteBody	400 Bad Request
InternalError	500 内部サーバーエラー
InvalidAccessKeyId	403 Forbidden
InvalidArgument	400 Bad Request
InvalidBucketName	400 Bad Request
InvalidBucketState	409 Conflict
InvalidDigest	400 Bad Request
InvalidEncryptionAlgorithmError	400 Bad Request
InvalidPart	400 Bad Request
InvalidPartOrder	400 Bad Request
InvalidRange	416 Requested Range Not Satisfiable
InvalidRequest	400 Bad Request
InvalidStorageClass	400 Bad Request
InvalidTag	400 Bad Request
無効なURI	400 Bad Request
KeyTooLong	400 Bad Request
不正なXML	400 Bad Request
MetadataTooLarge	400 Bad Request
MethodNotAllowed	405 Method Not Allowed
MissingContentLength	411 Length Required
MissingRequestBodyError	400 Bad Request

Name	HTTPのステータス
MissingSecurityHeader	400 Bad Request
NoSuchBucket	404 Not Found
NoSuchKey	404 Not Found
NoSuchUpload	404 Not Found
NotImplemented	501 未実装
NoSuchBucketPolicy	404 Not Found
ObjectLockConfigurationNotFoundError	404 Not Found
PreconditionFailed	412 Precondition Failed
RequestTimeTooSkewed	403 Forbidden
ServiceUnavailable	503 Service Unavailable
SignatureDoesNotMatch	403 Forbidden
TooManyBuckets	400 Bad Request
UserKeyMustBeSpecified	400 Bad Request

StorageGRID カスタムエラーコード

Name	説明	HTTPのステータス
XBucketLifecycleNotAllowed	従来の準拠バケットでは、バケットのライフサイクル構成は許可されていません。	400 Bad Request
XBucketPolicyParseException	受信したバケットポリシーのJSONを解析できませんでした。	400 Bad Request
XComplianceConflict	旧式のコンプライアンス設定のため、操作が拒否されました。	403 Forbidden
XComplianceReducedRedundancyForbidden	レガシー準拠バケットでは冗長性の削減は許可されていません	400 Bad Request

Name	説明	HTTPのステータス
XMaxBucketPolicyLengthExceeded	お客様のポリシーは、許可されている最大バケットポリシーの長さを超えています。	400 Bad Request
XMissingInternalRequestHeader	内部リクエストのヘッダーが欠落しています。	400 Bad Request
XNoSuchBucketCompliance	指定されたバケットでは、従来のコンプライアンスが有効になっていません。	404 Not Found
XNotAcceptable	リクエストには、満たすことのできなかった1つ以上のAcceptヘッダーが含まれています。	406 Not Acceptable
XNotImplemented	ご提示いただいたご要望は、実装されていない機能に関するものです。	501 未実装

StorageGRID カスタムオペレーション

StorageGRID でサポートされているカスタムバケット操作

StorageGRIDシステムは、S3 REST APIに追加されるカスタム操作をサポートします。

次の表は、StorageGRID でサポートされているカスタム操作の一覧です。

処理	説明
"GET Bucket consistency"	特定のバケットに適用されている整合性を返します。
"PUT Bucketの一貫性"	特定のバケットに適用される整合性を設定します。
"GET Bucketの最終アクセス時間"	特定のバケットに対して、最終アクセス時刻の更新が有効になっているか無効になっているかを返します。
"PUT Bucketの最終アクセス時刻"	特定のバケットの最終アクセス時刻の更新を有効または無効にすることができます。
"バケットメタデータ通知設定の削除"	特定のバケットに関連付けられているメタデータ通知設定XMLを削除します。
"バケットメタデータ通知設定を取得します"	特定のバケットに関連付けられたメタデータ通知設定XMLを返します。
"PUT Bucketメタデータ通知設定"	バケットのメタデータ通知サービスを設定します。

処理	説明
"ストレージ使用量を取得する"	アカウントで使用されているストレージの総量と、そのアカウントに関連付けられている各バケットごとのストレージ量を表示します。
"非推奨：CreateBucket（コンプライアンス設定付き）"	非推奨となり、サポート対象外です。コンプライアンスを有効にした状態で新しいバケットを作成することはできなくなりました。
"非推奨: GET Bucket compliance"	非推奨ですがサポートされています：既存のレガシー準拠バケットに対して現在有効なコンプライアンス設定を返します。
"非推奨: PUT Bucket compliance"	非推奨ですがサポートされています：既存のレガシー準拠バケットのコンプライアンス設定を変更できます。

StorageGRIDでGET Bucket整合性リクエストを使用してバケットの整合性レベルを取得する

GET Bucket 整合性リクエストを使用すると、特定のバケットに適用されている整合性を確認できます。

デフォルトの整合性設定では、新しく作成されたオブジェクトに対して書き込み後の読み取りを保証するようになっています。

この操作を完了するには、s3:GetBucketConsistency 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?x-ntap-sg-consistency HTTP/1.1
Date: date
Authorization: authorization string
Host: host
```

応答

レスポンスXMLでは、 `<Consistency>`は次のいずれかの値を返します。

一貫性	説明
all	すべてのノードがデータを即座に受信しなければ、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。

一貫性	説明
strong-site	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	（デフォルト）新規オブジェクトに対しては書き込み後の読み取り一貫性を、オブジェクト更新に対しては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。
使用可能	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。

回答例

```
HTTP/1.1 200 OK
Date: Fri, 18 Sep 2020 01:02:18 GMT
Connection: CLOSE
Server: StorageGRID/11.5.0
x-amz-request-id: 12345
Content-Length: 127
Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8"?>
<Consistency xmlns="http://s3.storagegrid.com/doc/2015-02-01/">read-after-
new-write</Consistency>
```

関連情報

"一貫性"

PUT Bucket整合性リクエストを使用してStorageGRIDで整合性レベルを指定する

PUT Bucket整合性リクエストを使用すると、バケットに対して実行される操作に適用する整合性を指定できます。

デフォルトの整合性設定では、新しく作成されたオブジェクトに対して書き込み後の読み取りを保証するようになっています。

開始する前に

この操作を完了するには、s3:PutBucketConsistency 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。

要求

`x-ntap-sg-consistency`パラメータには、以下のいずれかの値を含める必要があります：

一貫性	説明
all	すべてのノードがデータを即座に受信しなければ、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。
strong-site	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	（デフォルト）新規オブジェクトに対しては書き込み後の読み取り一貫性を、オブジェクト更新に対しては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。
使用可能	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。

注: 一般的には、「新規書き込み後の読み取り」一貫性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リクエストの一貫性を指定するように設定することもできます。最終手段としてのみ、バケットレベルでの一貫性設定を行ってください。

リクエスト例

```
PUT /bucket?x-ntap-sg-consistency=strong-global HTTP/1.1
Date: date
Authorization: authorization string
Host: host
```

関連情報

"一貫性"

GET Bucket last access timeリクエストを使用して**StorageGRID**でバケットの最終アクセス時間ステータスを取得する

GET Bucket last access time リクエストを使用すると、個々のバケットに対して最終アクセス時刻の更新が有効になっているか無効になっているかを判断できます。

この操作を完了するには、s3:GetBucketLastAccessTime 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?x-ntap-sg-lastaccesstime HTTP/1.1
Date: date
Authorization: authorization string
Host: host
```

回答例

この例は、バケットに対して最終アクセス時刻の更新が有効になっていることを示しています。

```
HTTP/1.1 200 OK
Date: Sat, 29 Nov 2015 01:02:18 GMT
Connection: CLOSE
Server: StorageGRID/10.3.0
x-amz-request-id: 12345
Content-Length: 127
Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8"?>
<LastAccessTime xmlns="http://s3.storagegrid.com/doc/2015-02-01/">enabled
</LastAccessTime>
```

StorageGRID で PUT Bucket 最終アクセス時刻リクエストを使用して最終アクセス時刻の更新を有効または無効にする

PUT Bucket last access timeリクエストを使用すると、個々のバケットの最終アクセス時刻の更新を有効または無効にすることができます。最終アクセス時刻の更新を無効にするとパフォーマンスが向上し、バージョン10.3.0以降で作成されたすべてのバケットのデフォルト設定となっています。

この操作を完了するには、バケットに対する s3:PutBucketLastAccessTime 権限、またはアカウントのルート権限が必要です。

> StorageGRID バージョン 10.3 以降、すべての新規バケットにおいて、最終アクセス時刻の更新はデフォルトで無効になっています。以前のバージョンの StorageGRID を使用して作成されたバケットがあり、新しいデフォルトの動作に合わせたい場合は、以前のバケットそれぞれについて、最終アクセス時刻の更新を明示的に無効にする必要があります。PUT Bucket last access time リクエストを使用するか、テナントマネージャのバケットの詳細ページから、最終アクセス時刻の更新を有効または無効にすることができます。参照してください "最終アクセス時刻の更新を有効または無効にする"。

バケットの最終アクセス時刻の更新が無効になっている場合、そのバケットに対する操作には以下の動作が適用されます。

- GetObject、GetObjectAcl、GetObjectTagging、およびHeadObjectリクエストでは最終アクセス時刻が更新されません。オブジェクトは、情報ライフサイクル管理（ILM）評価のキューに追加されません。
- CopyObject および PutObjectTagging のメタデータのみを更新するリクエストは、最終アクセス時刻も更新します。オブジェクトは ILM 評価のキューに追加されます。
- ソースバケットの最終アクセス時刻の更新が無効になっている場合、CopyObjectリクエストでは、ソースバケットの最終アクセス時刻は更新されません。コピーされたオブジェクトは、ソースバケットのILM評価キューには追加されません。ただし、コピー先については、CopyObjectリクエストは常に最終アクセス時刻を更新します。オブジェクトのコピーは、ILM評価のキューに追加されます。
- CompleteMultipartUpload リクエストは最終アクセス時刻を更新します。完成したオブジェクトは、ILM評価のキューに追加されます。

リクエスト例

この例では、バケットの最終アクセス時刻を有効にします。

```
PUT /bucket?x-ntap-sg-lastaccesstime=enabled HTTP/1.1
Date: date
Authorization: authorization string
Host: host
```

この例では、バケットの最終アクセス時刻を無効にします。

```
PUT /bucket?x-ntap-sg-lastaccesstime=disabled HTTP/1.1
Date: date
Authorization: authorization string
Host: host
```

StorageGRIDのDELETE Bucketメタデータ通知設定リクエストで検索統合サービスを無効にします

DELETEバケットメタデータ通知構成リクエストを使用すると、構成XMLを削除することで、個々のバケットの検索統合サービスを無効にできます。

この操作を完了するには、バケットに対する s3:DeleteBucketMetadataNotification 権限、またはアカウントのルート権限が必要です。

リクエスト例

この例では、バケットの検索統合サービスを無効にする方法を示します。

```
DELETE /test1?x-ntap-sg-metadata-notification HTTP/1.1
Date: date
Authorization: authorization string
Host: host
```

GET Bucketメタデータ通知設定リクエストを使用してStorageGRIDで検索統合を設定する

GET Bucket メタデータ通知設定リクエストを使用すると、個々のバケットの検索統合を設定するために使用される設定 XML を取得できます。

この操作を完了するには、s3:GetBucketMetadataNotification 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

このリクエストは、 `bucket`という名前のバケットのメタデータ通知設定を取得します。

```
GET /bucket?x-ntap-sg-metadata-notification HTTP/1.1
Date: date
Authorization: authorization string
Host: host
```

応答

レスポンスボディには、バケットのメタデータ通知設定が含まれます。メタデータ通知の設定により、検索統合のためにバケットをどのように構成するかを決定できます。つまり、どのオブジェクトがインデックス化されているか、およびそのオブジェクトのメタデータがどのエンドポイントに送信されているかを確認できます。

```
<MetadataNotificationConfiguration>
    <Rule>
        <ID>Rule-1</ID>
        <Status>rule-status</Status>
        <Prefix>key-prefix</Prefix>
        <Destination>
           <Urn>arn:aws:es:_region:account-
ID_:domain/_mydomain/myindex/mytype_</Urn>
        </Destination>
    </Rule>
    <Rule>
        <ID>Rule-2</ID>
         ...
    </Rule>
     ...
</MetadataNotificationConfiguration>
```

各メタデータ通知設定には、1つ以上のルールが含まれます。各ルールは、適用されるオブジェクトと、StorageGRID がオブジェクトメタデータを送信する宛先を指定します。宛先は StorageGRID エンドポイントの URN を使用して指定する必要があります。

Name	説明	必須
MetadataNotificationConfiguration	メタデータ通知の対象となるオブジェクトと宛先を指定するために使用されるルールを格納するタグ。 1つ以上のルール要素を含みます。	はい
Rule	メタデータを特定のインデックスに追加すべきオブジェクトを識別するルールのコンテナタグ。 接頭辞が重複するルールは拒否されます。 MetadataNotificationConfiguration 要素に含まれます。	はい
ID	ルールの一意の識別子。 ルール要素に含まれています。	いいえ
ステータス	ステータスは「有効」または「無効」のいずれかになります。無効化されたルールに対しては、何も措置は講じられません。 ルール要素に含まれています。	はい

Name	説明	必須
Prefix	プレフィックスに一致するオブジェクトはルールの影響を受け、そのメタデータは指定された宛先に送信されます。 すべてのオブジェクトに一致させるには、空のプレフィックスを指定します。 ルール要素に含まれています。	はい
デスティネーション	ルールの適用先を示すコンテナタグ。 ルール要素に含まれています。	はい
Urn	オブジェクトのメタデータが送信される宛先のURN。以下のプロパティを持つStorageGRIDエンドポイントのURNである必要があります： • `es`3番目の要素でなければなりません。 • URNは、メタデータが格納されているインデックスとタイプで終わる必要があります。形式は`domain-name/myindex/mytype`です。 エンドポイントは、Tenant ManagerまたはTenant Management APIを使用して設定します。形式は次のとおりです： • `arn:aws:es:_region:account-ID_:domain/mydomain/myindex/mytype` • `urn:mysite:es:::mydomain/myindex/mytype` エンドポイントは、設定XMLを送信する前に設定しておく必要があります。設定されていない場合、設定は404エラーで失敗します。 URN は Destination 要素に含まれています。	はい

回答例

`<MetadataNotificationConfiguration></MetadataNotificationConfiguration>`タグの間に含まれるXMLは、バケットに対して検索統合エンドポイントとの統合がどのように構成されているかを示しています。この例では、オブジェクトのメタデータが、 `records` という名前のAWSドメインでホストされている、 `current`という名前のElasticsearchインデックスと `2017`という名前のタイプに送信されています。

```
HTTP/1.1 200 OK
Date: Thu, 20 Jul 2017 18:24:05 GMT
Connection: KEEP-ALIVE
Server: StorageGRID/11.0.0
x-amz-request-id: 3832973499
Content-Length: 264
Content-Type: application/xml

<MetadataNotificationConfiguration>
    <Rule>
        <ID>Rule-1</ID>
        <Status>Enabled</Status>
        <Prefix>2017</Prefix>
        <Destination>
           <Urn>arn:aws:es:us-east-
1:3333333:domain/records/current/2017</Urn>
        </Destination>
    </Rule>
</MetadataNotificationConfiguration>
```

関連情報

"テナントアカウントを使用する"

PUT Bucketメタデータ通知設定リクエストを使用してStorageGRIDで検索統合サービスを有効にします

PUTバケットメタデータ通知設定リクエストを使用すると、個々のバケットに対して検索統合サービスを有効にできます。リクエスト本文で指定するメタデータ通知設定XMLは、メタデータが宛先検索インデックスに送信されるオブジェクトを指定します。

この操作を完了するには、バケットに対する s3:PutBucketMetadataNotification 権限、またはアカウントのルート権限が必要です。

要求

リクエストの本文には、メタデータ通知の設定を含める必要があります。各メタデータ通知設定には、1つ以上のルールが含まれます。各ルールは、適用対象のオブジェクトと、StorageGRID がオブジェクトメタデータを送信する宛先を指定します。

オブジェクトは、オブジェクト名のプレフィックスに基づいてフィルタリングできます。たとえば、プレフィックスが `/images`のオブジェクトのメタデータを1つの宛先に送信し、プレフィックスが `/videos`のオブジェクトを別の宛先に送信することができます。

プレフィックスが重複している設定は無効であり、送信時に拒否されます。例えば、プレフィックス `test`を持つオブジェクトに対する1つのルールと、プレフィックス `test2`を持つオブジェクトに対する2つ目のルールを含む設定は許可されません。

宛先は StorageGRID エンドポイントの URN を使用して指定する必要があります。メタデータ通知設定が送信される時点でエンドポイントが存在していない場合、リクエストは `400 Bad Request`として失敗します。エラーメッセージには次のように記載されています： `Unable to save the metadata notification (search) policy. The specified endpoint URN does not exist: *URN*.`

```
<MetadataNotificationConfiguration>
    <Rule>
        <ID>Rule-1</ID>
        <Status>rule-status</Status>
        <Prefix>key-prefix</Prefix>
        <Destination>
           <Urn>arn:aws:es:region:account-
ID:domain/mydomain/myindex/mytype</Urn>
        </Destination>
    </Rule>
    <Rule>
        <ID>Rule-2</ID>
         ...
    </Rule>
     ...
</MetadataNotificationConfiguration>
```

この表は、メタデータ通知設定XMLの要素について説明しています。

Name	説明	必須
MetadataNotificationConfiguration	メタデータ通知の対象となるオブジェクトと宛先を指定するために使用されるルールを格納するタグ。 1つ以上のルール要素を含みます。	はい
Rule	メタデータを特定のインデックスに追加すべきオブジェクトを識別するルールのコンテナタグ。 接頭辞が重複するルールは拒否されます。 MetadataNotificationConfiguration 要素に含まれます。	はい
ID	ルールの一意の識別子。 ルール要素に含まれています。	いいえ

Name	説明	必須
ステータス	ステータスは「有効」または「無効」のいずれかになります。無効化されたルールに対しては、何も措置は講じられません。 ルール要素に含まれています。	はい
Prefix	プレフィックスに一致するオブジェクトはルールの影響を受け、そのメタデータは指定された宛先に送信されます。 すべてのオブジェクトに一致させるには、空のプレフィックスを指定します。 ルール要素に含まれています。	はい
デスティネーション	ルールの適用先を示すコンテナタグ。 ルール要素に含まれています。	はい
Urn	オブジェクトのメタデータが送信される宛先のURN。以下のプロパティを持つStorageGRIDエンドポイントのURNである必要があります： • `es`3番目の要素でなければなりません。 • URNは、メタデータが格納されているインデックスとタイプで終わる必要があります。形式は`domain-name/myindex/mytype`です。 エンドポイントは、Tenant ManagerまたはTenant Management APIを使用して設定します。形式は次のとおりです： • `arn:aws:es:region:account-ID:domain/mydomain/myindex/mytype` • `urn:mysite:es:::mydomain/myindex/mytype` エンドポイントは、設定XMLを送信する前に設定しておく必要があります。設定されていない場合、設定は404エラーで失敗します。 URN は Destination 要素に含まれています。	はい

リクエスト例

この例では、バケットの検索統合を有効にする方法を示します。この例では、すべてのオブジェクトのオブジェクトメタデータが同じ宛先に送信されます。

```
PUT /test1?x-ntap-sg-metadata-notification HTTP/1.1
Date: date
Authorization: authorization string
Host: host

<MetadataNotificationConfiguration>
    <Rule>
        <ID>Rule-1</ID>
        <Status>Enabled</Status>
        <Prefix></Prefix>
        <Destination>
           <Urn>urn:sgws:es:::sgws-notifications/test1/all</Urn>
        </Destination>
    </Rule>
</MetadataNotificationConfiguration>
```

この例では、プレフィックス `/images`に一致するオブジェクトのオブジェクトメタデータは1つの宛先に送信され、プレフィックス `/videos`に一致するオブジェクトのオブジェクトメタデータは2番目の宛先に送信されます。

```
PUT /graphics?x-ntap-sg-metadata-notification HTTP/1.1
Date: date
Authorization: authorization string
Host: host

<MetadataNotificationConfiguration>
    <Rule>
        <ID>Images-rule</ID>
        <Status>Enabled</Status>
        <Prefix>/images</Prefix>
        <Destination>
           <Urn>arn:aws:es:us-east-1:3333333:domain/es-
domain/graphics/imagetype</Urn>
        </Destination>
    </Rule>
    <Rule>
        <ID>Videos-rule</ID>
        <Status>Enabled</Status>
        <Prefix>/videos</Prefix>
        <Destination>
           <Urn>arn:aws:es:us-west-1:22222222:domain/es-
domain/graphics/videotype</Urn>
        </Destination>
    </Rule>
</MetadataNotificationConfiguration>
```

検索統合サービスによって生成された**JSON**

バケットの検索統合サービスを有効にすると、オブジェクトのメタデータまたはタグが追加、更新、または削除されるたびに、JSONドキュメントが生成され、宛先エンドポイントに送信されます。

この例は、キー `SGWS/Tagging.txt`を持つオブジェクトが `test`という名前のバケットに作成された場合に生成される可能性のある JSON の例を示しています。 `test`バケットはバージョン管理されていないため、 `versionId`タグは空です。

```
{
  "bucket": "test",
  "key": "SGWS/Tagging.txt",
  "versionId": "",
  "accountId": "86928401983529626822",
  "size": 38,
  "md5": "3d6c7634a85436eee06d43415012855",
  "region":"us-east-1",
  "metadata": {
    "age": "25"
  },
  "tags": {
    "color": "yellow"
  }
}
```

メタデータ通知に含まれるオブジェクトメタデータ

この表には、検索統合が有効になっている場合に宛先エンドポイントに送信されるJSONドキュメントに含まれるすべてのフィールドが一覧表示されています。

ドキュメント名には、バケット名、オブジェクト名、およびバージョンID（存在する場合）が含まれます。

Type	商品名	説明
バケットとオブジェクトの情報	バケット	バケットの名前
バケットとオブジェクトの情報	キー	オブジェクトキー名
バケットとオブジェクトの情報	versionID	オブジェクトバージョン（バージョン管理されたバケット内のオブジェクトの場合）
バケットとオブジェクトの情報	リージョン	バケットリージョン（例： `us-east-1`）
システムメタデータ	サイズ	HTTPクライアントから見えるオブジェクトサイズ（バイト単位）
システムメタデータ	md5を使用したチャンク アップロード署名要求がサポートされるようになりました。	オブジェクトハッシュ

Type	商品名	説明
ユーザーメタデータ	メタデータ `key:value`	オブジェクトに関するすべてのユーザーメタデータ（キーと値のペアとして）
Tags	タグ `key:value`	オブジェクトに対して定義されたすべてのオブジェクトタグ（キーと値のペア）

タグとユーザーメタデータについては、StorageGRID は日付と数値を文字列またはS3イベント通知としてElasticsearchに渡します。Elasticsearchがこれらの文字列を日付または数値として解釈するように設定するには、Elasticsearchの動的フィールドマッピングおよび日付形式のマッピングに関する手順に従ってください。検索統合サービスを設定する前に、インデックス上で動的フィールドマッピングを有効にする必要があります。ドキュメントがインデックス化された後は、インデックス内のドキュメントのフィールドタイプを編集することはできません。

関連情報

"テナントアカウントを使用する"

StorageGRIDでGET Storage Usageリクエストを使用してアカウントとバケットのストレージ使用量を取得

GET Storage Usageリクエストは、アカウントが使用しているストレージの総量と、そのアカウントに関連付けられている各バケットごとのストレージ使用量を示します。

アカウントとそのバケットが使用するストレージ容量は、 `x-ntap-sg-usage`クエリパラメータを使用した修正済みのListBucketsリクエストで取得できます。バケットストレージの使用状況は、システムによって処理される PUT および DELETE リクエストとは別に追跡されます。リクエストの処理状況によっては、特にシステム負荷が高い場合、使用状況の値が期待値と一致するまでに多少の遅延が生じる可能性があります。

デフォルトでは、StorageGRID は強力なグローバル整合性を使用して使用状況情報を取得しようとします。強力なグローバル整合性が達成できない場合、StorageGRID はサイトの強力な整合性で使用状況情報を取得しようとします。

この操作を完了するには、s3:ListAllMyBuckets 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 /?x-ntap-sg-usage HTTP/1.1
Date: date
Authorization: authorization string
Host: host
```

回答例

この例は、2つのバケットに4つのオブジェクトと12バイトのデータを持つアカウントを示しています。各バケットには2つのオブジェクトと6バイトのデータが含まれています。

```
HTTP/1.1 200 OK
Date: Sat, 29 Nov 2015 00:49:05 GMT
Connection: KEEP-ALIVE
Server: StorageGRID/10.2.0
x-amz-request-id: 727237123
Content-Length: 427
Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8"?>
<UsageResult xmlns="http://s3.storagegrid.com/doc/2015-02-01">
<CalculationTime>2014-11-19T05:30:11.000000Z</CalculationTime>
<ObjectCount>4</ObjectCount>
<DataBytes>12</DataBytes>
<Buckets>
<Bucket>
<Name>bucket1</Name>
<ObjectCount>2</ObjectCount>
<DataBytes>6</DataBytes>
</Bucket>
<Bucket>
<Name>bucket2</Name>
<ObjectCount>2</ObjectCount>
<DataBytes>6</DataBytes>
</Bucket>
</Buckets>
</UsageResult>
```

バージョン管理

保存されるすべてのオブジェクトバージョンは、応答内の `ObjectCount`および `DataBytes`の値に反映されます。削除マーカーは `ObjectCount`の合計には追加されません。

関連情報

"一貫性"

レガシーコンプライアンス向けのバケットリクエストは非推奨です

レガシー **StorageGRID** コンプライアンス向けの非推奨のバケットリクエスト

従来のコンプライアンス機能を使用して作成されたバケットを管理するには、StorageGRID S3 REST API を使用する必要がある場合があります。

以前のバージョンの StorageGRID で利用可能だった StorageGRID コンプライアンス機能は非推奨となり、S3 Object Lock に置き換えられました。

以前にグローバルコンプライアンス設定を有効にした場合、グローバル S3 オブジェクトロック設定は StorageGRID 11.6 で有効になります。コンプライアンスが有効になっている新しいバケットを作成することはできなくなりましたが、必要に応じて、StorageGRID S3 REST API を使用して既存のレガシー準拠バケットを管理できます。

- "S3 REST APIを使用してS3オブジェクトロックを設定する"
- "ILMを使用してオブジェクトを管理する"
- "NetApp ナレッジベース：StorageGRID 11.5 でレガシー準拠バケットを管理する方法"

廃止されたコンプライアンス要求：

- "非推奨 - コンプライアンスのための PUT バケットリクエストの変更"

  SGCompliance XML要素は非推奨です。以前は、PUT Bucket リクエストのオプションの XML リクエスト本文にこの StorageGRID カスタム要素を含めることで、準拠バケットを作成することができました。

- "非推奨 - GET Bucket コンプライアンス"

  GET Bucket コンプライアンス要求は非推奨になりました。ただし、このリクエストを使用して、既存のレガシー準拠バケットに対して現在適用されているコンプライアンス設定を確認することは引き続き可能です。

- "非推奨 - PUT Bucket コンプライアンス"

  PUT Bucketコンプライアンスリクエストは非推奨です。ただし、このリクエストを使用して、既存のレガシー準拠バケットのコンプライアンス設定を変更することは引き続き可能です。例えば、既存のバケットを法的保留状態にしたり、保持期間を延長したりすることができます。

StorageGRIDのSGCompliance要素は廃止されました

SGCompliance XML要素は非推奨です。以前は、準拠バケットを作成するための CreateBucket リクエストのオプションの XML リクエスト本文に、この StorageGRID カスタム要素を含めることができました。

> 以前のバージョンの StorageGRID で利用可能だった StorageGRID コンプライアンス機能は非推奨となり、S3 Object Lock に置き換えられました。詳細については、以下をご覧ください。
>
> - "S3 REST APIを使用してS3オブジェクトロックを設定する"
> - "NetApp ナレッジベース：StorageGRID 11.5 でレガシー準拠バケットを管理する方法"

コンプライアンスが有効になっている状態では、新しいバケットを作成できなくなりました。新しい準拠バケットを作成するためにコンプライアンス向けの CreateBucket リクエストの変更を使用しようとすると、次のエラーメッセージが返されます：

```
The Compliance feature is deprecated.
Contact your StorageGRID administrator if you need to create new Compliant
buckets.
```

StorageGRID で廃止された **GET Bucket compliance** リクエスト

GET Bucket コンプライアンス要求は非推奨になりました。ただし、このリクエストを使用して、既存のレガシー準拠バケットに対して現在適用されているコンプライアンス設定を確認することは引き続き可能です。

以前のバージョンの StorageGRID で利用可能だった StorageGRID コンプライアンス機能は非推奨となり、S3 Object Lock に置き換えられました。詳細については、以下をご覧ください。

- "S3 REST APIを使用してS3オブジェクトロックを設定する"
- "NetApp ナレッジベース：StorageGRID 11.5 でレガシー準拠バケットを管理する方法"

この操作を完了するには、s3:GetBucketCompliance 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

このリクエスト例では、 `mybucket`という名前のバケットのコンプライアンス設定を確認できます。

```
GET /mybucket/?x-ntap-sg-compliance HTTP/1.1
Date: date
Authorization: authorization string
Host: host
```

回答例

レスポンスXMLでは、 `<SGCompliance>`バケットに適用されているコンプライアンス設定を一覧表示します。この応答例は、オブジェクトがグリッドに取り込まれた時点から1年間（525,600分）保持されるバケットのコンプライアンス設定を示しています。現在、このバケットには法的ホールドはありません。各オブジェクトは1年後に自動的に削除されます。

```
HTTP/1.1 200 OK
Date: date
Connection: connection
Server: StorageGRID/11.1.0
x-amz-request-id: request ID
Content-Length: length
Content-Type: application/xml

<SGCompliance>
  <RetentionPeriodMinutes>525600</RetentionPeriodMinutes>
  <LegalHold>false</LegalHold>
  <AutoDelete>true</AutoDelete>
</SGCompliance>
```

Name	説明
RetentionPeriodMinutes	このバケットに追加されたオブジェクトの保持期間（分単位）。保持期間は、オブジェクトがグリッドに取り込まれた時点から開始されます。
LegalHold	• True：このバケットは現在、法的保留状態にあります。このバケット内のオブジェクトは、保持期間が過ぎていても、法的保留が解除されるまで削除できません。 • False：このバケットは現在、法的保留下にはありません。このバケット内のオブジェクトは、保持期間が経過すると削除できます。
AutoDelete	• 正しい：このバケット内のオブジェクトは、バケットが法的保留下にない限り、保持期間が満了すると自動的に削除されます。 • False: このバケット内のオブジェクトは、保持期間が満了しても自動的に削除されません。これらのオブジェクトを削除する必要がある場合は、手動で削除してください。

エラー応答

バケットが準拠するように作成されていない場合、応答の HTTP ステータスコードは `404 Not Found`となり、S3 エラーコードは `XNoSuchBucketCompliance`です。

StorageGRID で廃止された **PUT Bucket compliance** リクエスト

PUT Bucketコンプライアンスリクエストは非推奨です。ただし、このリクエストを使用して、既存のレガシー準拠バケットのコンプライアンス設定を変更することは引き続き可能です。例えば、既存のバケットを法的保留状態にしたり、保持期間を延長したりすることができます。

以前のバージョンの StorageGRID で利用可能だった StorageGRID コンプライアンス機能は非推奨となり、S3 Object Lock に置き換えられました。詳細については、以下をご覧ください。

- "S3 REST APIを使用してS3オブジェクトロックを設定する"
- "NetApp ナレッジベース：StorageGRID 11.5 でレガシー準拠バケットを管理する方法"

この操作を完了するには、s3:PutBucketCompliance 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。

PUT Bucketコンプライアンスリクエストを発行する際には、コンプライアンス設定のすべてのフィールドに値を指定する必要があります。

リクエスト例

この例のリクエストは、 `mybucket`という名前のバケットのコンプライアンス設定を変更します。この例では、 `mybucket`内のオブジェクトがグリッドに取り込まれた時点から、保持期間は1年間ではなく2年間（1,051,200分）になります。このバケットにはリーガルホールドはありません。各オブジェクトは2年後に自動的に削除されます。

```
PUT /mybucket/?x-ntap-sg-compliance HTTP/1.1
Date: date
Authorization: authorization name
Host: host
Content-Length: 152

<SGCompliance>
  <RetentionPeriodMinutes>1051200</RetentionPeriodMinutes>
  <LegalHold>false</LegalHold>
  <AutoDelete>true</AutoDelete>
</SGCompliance>
```

Name	説明
RetentionPeriodMinutes	このバケットに追加されたオブジェクトの保持期間（分単位）。保持期間は、オブジェクトがグリッドに取り込まれた時点から開始されます。 重要 RetentionPeriodMinutes に新しい値を指定する場合は、バケットの現在の保持期間以上の値を指定する必要があります。バケットの保持期間が設定されると、その値を小さくすることはできません。大きくすることしかできません。
LegalHold	• True：このバケットは現在、法的保留状態にあります。このバケット内のオブジェクトは、保持期間が過ぎていても、法的保留が解除されるまで削除できません。 • False：このバケットは現在、法的保留下にはありません。このバケット内のオブジェクトは、保持期間が経過すると削除できます。

Name	説明
AutoDelete	• 正しい：このバケット内のオブジェクトは、バケットが法的保留下にない限り、保持期間が満了すると自動的に削除されます。 • False: このバケット内のオブジェクトは、保持期間が満了しても自動的に削除されません。これらのオブジェクトを削除する必要がある場合は、手動で削除してください。

コンプライアンス設定の一貫性

PUT Bucket コンプライアンスリクエストを使用して S3 バケットのコンプライアンス設定を更新すると、StorageGRID はグリッド全体でバケットのメタデータを更新しようとします。デフォルトでは、StorageGRID は **Strong-global** 整合性を使用して、バケットメタデータを含むすべてのデータセンターサイトとすべてのストレージノードが、変更されたコンプライアンス設定に対してライトアフターリード整合性を持つことを保証します。

StorageGRID がデータセンターサイトまたはサイト内の複数のストレージノードが利用できないために **Strong-global** 整合性を達成できない場合、レスポンスの HTTP ステータスコードは `503 Service Unavailable.`

この応答を受け取った場合は、必要なストレージサービスが可能な限り速やかに利用可能になるよう、グリッド管理者に連絡してください。グリッド管理者が各サイトのストレージノードを十分に利用可能にできない場合、テクニカルサポートから、**Strong-site** の整合性を強制することで失敗したリクエストを再試行するよう指示される場合があります。

> テクニカルサポートから指示された場合、およびこのレベルを使用することによる潜在的な影響を理解している場合を除き、PUT バケットコンプライアンスに対して **Strong-site** の整合性を強制しないでください。

一貫性が*ストロングサイト*に低下すると、StorageGRID は、更新されたコンプライアンス設定について、サイト内のクライアント要求に対してのみ書き込み後読み取りの一貫性を保証します。これはつまり、すべてのサイトとストレージノードが利用可能になるまで、StorageGRID システムはこのバケットに対して一時的に複数の矛盾した設定を持つ可能性があるということです。設定に一貫性がないと、予期しない望ましくない動作が発生する可能性があります。例えば、バケットを法的保留下に置く際に、より低い整合性を強制的に適用した場合、一部のデータセンターサイトでは、バケットの以前のコンプライアンス設定（つまり、法的保留オフ）が引き続き有効になる可能性があります。その結果、法的保留中と思われるオブジェクトは、保持期間が満了すると、ユーザーまたは AutoDelete（有効な場合）によって削除される可能性があります。

Strong-site の一貫性の使用を強制するには、PUT Bucket コンプライアンス要求を再発行し、以下のように `Consistency-Control` HTTP リクエストヘッダーを含めます：

```
PUT /mybucket/?x-ntap-sg-compliance HTTP/1.1
Consistency-Control: strong-site
```

エラー応答

- バケットが準拠するように作成されていない場合、応答の HTTP ステータスコードは `404 Not Found`です。
- リクエストの `RetentionPeriodMinutes`がバケットの現在の保持期間より短い場合、HTTP ステータスコ

ードは `400 Bad Request`です。

関連情報

"非推奨: PUT Bucket リクエストのコンプライアンス変更"

アクセスポリシーの管理

StorageGRID が AWS ポリシーを使用して S3 バケットとオブジェクトへのアクセスを制御する方法

StorageGRID は Amazon Web Services (AWS) のポリシー言語を使用して、S3 テナントがバケットおよびバケット内のオブジェクトへのアクセスを制御できるようにします。StorageGRID システムは、S3 REST API ポリシー言語のサブセットを実装しています。S3 API のアクセスポリシーは JSON 形式で記述されます。

アクセスポリシーの概要

StorageGRID は、3種類のアクセスポリシーをサポートしています。

- *バケットポリシー*は、GetBucketPolicy、PutBucketPolicy、およびDeleteBucketPolicy S3 API操作、またはテナントマネージャーもしくはテナント管理APIを使用して管理されます。バケットポリシーはバケットに紐付けられており、バケット所有者アカウントまたはその他のアカウントのユーザーによるバケットおよびバケット内のオブジェクトへのアクセスを制御するように構成されます。バケットポリシーは、1つのバケットと、場合によっては複数のグループにのみ適用されます。
- *グループポリシー*は、Tenant ManagerまたはTenant Management APIを使用して設定します。グループポリシーはアカウント内のグループに関連付けられており、そのグループがそのアカウントが所有する特定のリソースにアクセスできるように設定されます。グループポリシーは、1つのグループと、場合によっては複数のバケットにのみ適用されます。
- セッションポリシー は、AssumeRole リクエストの一部として含まれます。セッションポリシーは特定のセッションにのみ適用され、グループポリシーおよびバケットポリシーによって付与される権限に加えて、ユーザーが持つ権限をさらに詳細に定義します。

> グループポリシー、バケットポリシー、セッションポリシーの間には、優先順位の違いはありません。

StorageGRID バケットポリシーとグループポリシーは、Amazon が定義した特定の文法に従います。各ポリシーの中にはポリシーステートメントの配列があり、各ステートメントには以下の要素が含まれています。

- ステートメント ID （Sid）（オプション）
- 影響
- Principal/NotPrincipal
- リソース/NotResource
- アクション/NotAction
- 条件（任意）

ポリシーステートメントは、権限を指定するために次の構造を使用して作成されます：<Effect> を付与し

て、<Condition> が適用される場合に <Principal> が <Resource> に対して <Action> を実行することを許可/拒否します。

各ポリシー要素は、特定の機能に使用されます。

要素	説明
Sid	Sid要素は省略可能です。Sidは、ユーザー向けの説明としてのみ使用されるものです。StorageGRIDシステムによって保存されますが、解釈はされません。
影響	Effect要素を使用して、指定された操作が許可されるか拒否されるかを設定します。サポートされているAction要素キーワードを使用して、バケットまたはオブジェクトに対して許可（または拒否）する操作を指定する必要があります。
Principal/NotPrincipal	ユーザー、グループ、アカウントに対して、特定のリソースへのアクセス権限や特定のアクションの実行権限を付与できます。リクエストにS3署名が含まれていない場合、ワイルドカード文字（*）をプリンシパルとして指定することで、匿名アクセスが許可されます。デフォルトでは、アカウントのルートユーザーのみが、そのアカウントが所有するリソースにアクセスできます。 バケットポリシーでは、Principal要素のみを指定する必要があります。グループポリシーの場合、ポリシーが関連付けられているグループが暗黙のPrincipal要素となります。
リソース/NotResource	リソース要素は、バケットとオブジェクトを識別します。Amazon Resource Name （ARN）を使用してリソースを識別することで、バケットやオブジェクトに対するアクセス許可を許可または拒否できます。
アクション/NotAction	権限を構成する2つの要素は、アクション要素とエフェクト要素です。グループがリソースを要求すると、そのリソースへのアクセスが許可されるか、拒否されるかのいずれかになります。明示的に権限を割り当てない限りアクセスは拒否されますが、明示的な拒否を使用することで、別のポリシーによって付与された権限を上書きすることができます。
条件	Condition要素は省略可能です。条件を使用すると、ポリシーを適用するタイミングを決定する式を作成できます。

Action要素では、ワイルドカード文字（*）を使用して、すべての操作、または操作のサブセットを指定できます。例えば、このActionはs3:GetObject、s3:PutObject、s3:DeleteObjectなどの権限に一致します。

```
s3:*Object
```

Resource 要素では、ワイルドカード文字 (*) と (?) を使用できます。アスタリスク（*）は0文字以上の文字に一致しますが、疑問符（?）は任意の1文字に一致します。

Principal 要素では、すべてのユーザーに権限を付与する匿名アクセスの設定を除き、ワイルドカード文字は

サポートされていません。たとえば、ワイルドカード（*）を Principal の値として設定します。

```
"Principal":"*"
```

```
"Principal":{"AWS":"*"}
```

次の例では、ステートメントでEffect、Principal、Action、およびResource要素を使用しています。この例は、Effectの「Allow」を使用して、Principalである管理グループ `federated-group/admin`と財務グループ `federated-group/finance`に対して、指定した名前のバケットに対するAction `s3:ListBucket`の実行権限 `mybucket`、およびそのバケット内のすべてのオブジェクトに対するAction `s3:GetObject`の実行権限を付与する、完全なバケットポリシー記述を示しています。

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::27233906934684427525:federated-group/admin",
          "arn:aws:iam::27233906934684427525:federated-group/finance"
        ]
      },
      "Action": [
        "s3:ListBucket",
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::mybucket",
        "arn:aws:s3:::mybucket/*"
      ]
    }
  ]
}
```

バケットポリシーのサイズ制限は20,480バイト、グループポリシーのサイズ制限は5,120バイトです。

ポリシーの整合性

デフォルトでは、グループポリシーに加えた更新はすべて最終的に整合性が保たれます。グループポリシーが整合性を保つようになると、ポリシーのキャッシュ処理のため、変更が反映されるまでにさらに15分ほどかかる場合があります。デフォルトでは、バケットポリシーに加えた更新はすべて強整合性が保たれます。

必要に応じて、バケットポリシーの更新における整合性保証を変更できます。たとえば、サイト停止中にバケットポリシーの変更を有効にする場合などです。

この場合、PutBucketPolicyリクエストに `Consistency-Control`ヘッダーを設定するか、PUT Bucket 整合性リクエストを使用することができます。バケットポリシーが整合性を保つようになると、ポリシーのキャッシュのため、変更が反映されるまでにさらに8秒かかる場合があります。

> 一時的な状況に対処するために整合性を別の値に設定した場合は、作業が完了したら必ずバケットレベルの設定を元の値に戻してください。そうでない場合、今後のすべてのバケット要求は変更された設定を使用します。

セッションポリシーとは何ですか？

セッションポリシーとは、ユーザーがグループを代理で利用する場合など、特定のセッション中に利用可能な権限を一時的に制限するアクセスポリシーのことです。セッションポリシーでは、許可できる権限は一部のみであり、追加の権限を付与することはできません。グループ自体は、より広範な権限を持っている可能性があります。

ポリシーステートメントで**ARN**を使用する

ポリシーステートメントでは、ARN は Principal 要素と Resource 要素で使用されます。

- S3リソースARNを指定するには、以下の構文を使用します。

```
arn:aws:s3:::bucket-name
arn:aws:s3:::bucket-name/object_key
```

- IDリソースARN（ユーザーおよびグループ）を指定するには、以下の構文を使用します：

```
arn:aws:iam::account_id:root
arn:aws:iam::account_id:user/user_name
arn:aws:iam::account_id:group/group_name
arn:aws:iam::account_id:federated-user/user_name
arn:aws:iam::account_id:federated-group/group_name
```

その他の考慮事項：

- オブジェクトキー内の0個以上の文字に一致させるには、アスタリスク（*）をワイルドカードとして使用できます。
- オブジェクトキーで指定できる国際文字は、JSON UTF-8 または JSON \u エスケープシーケンスを使用してエンコードする必要があります。パーセント表記はサポートされていません。

  "RFC 2141 URN構文"

  PutBucketPolicy 操作の HTTP リクエストボディは、charset=UTF-8 でエンコードする必要があります。

ポリシーでリソースを指定する

ポリシーステートメントでは、Resource要素を使用して、アクセス許可または拒否するバケットまたはオブジェクトを指定できます。

- 各ポリシーステートメントには、Resource要素が必要です。ポリシーでは、リソースは要素 `Resource` で表されます。または、除外の場合は `NotResource`で表されます。
- S3リソースARNを使用してリソースを指定します。例：

```
"Resource": "arn:aws:s3:::mybucket/*"
```

- オブジェクトキーの中にポリシー変数を使用することもできます。例えば：

```
"Resource": "arn:aws:s3:::mybucket/home/${aws:username}/*"
```

- リソース値には、グループポリシー作成時にまだ存在しないバケットを指定できます。

ポリシーでプリンシパルを指定する

Principal要素を使用して、ポリシーステートメントによってリソースへのアクセスが許可または拒否されるユーザー、グループ、またはテナントアカウントを識別します。

- バケットポリシー内の各ポリシーステートメントには、Principal要素を含める必要があります。グループポリシーのポリシーステートメントには、グループ自体がプリンシパルであると解釈されるため、Principal要素は必要ありません。
- ポリシーでは、プリンシパルは要素「Principal」、または除外の場合は「NotPrincipal」で示されます。
- アカウントベースのIDは、IDまたはARNを使用して指定する必要があります。

```
"Principal": { "AWS": "account_id"}
"Principal": { "AWS": "identity_arn" }
```

- この例では、テナントアカウントID 272339069346844275 25を使用します。このアカウントには、アカウントのルートとアカウント内のすべてのユーザーが含まれます。

```
 "Principal": { "AWS": "27233906934684427525" }
```

- アカウントのルートのみを指定できます。

```
"Principal": { "AWS": "arn:aws:iam::27233906934684427525:root" }
```

- 特定のフェデレーションユーザー（「Alex」）を指定できます。

```
"Principal": { "AWS": "arn:aws:iam::27233906934684427525:federated-
user/Alex" }
```

- 特定のフェデレーショングループ（「Managers」）を指定できます：

```
"Principal": { "AWS": "arn:aws:iam::27233906934684427525:federated-group/Managers"  }
```

- 匿名プリンシパルを指定できます。

```
"Principal": "*"
```

- 曖昧さを避けるため、ユーザー名の代わりにユーザーUUIDを使用できます。

```
arn:aws:iam::27233906934684427525:user-uuid/de305d54-75b4-431b-adb2-eb6b9e546013
```

例えば、Alex が組織を離れ、ユーザー名 `Alex`が削除されたとします。新しい Alex が組織に加わり、同じ `Alex`ユーザー名が割り当てられた場合、新しいユーザーが意図せず元のユーザーに付与された権限を継承してしまう可能性があります。

- プリンシパル値には、バケットポリシー作成時にまだ存在しないグループ名またはユーザー名を指定できます。

ポリシーで権限を指定する

ポリシーにおいて、Action要素はリソースに対するアクセス許可を許可または拒否するために使用されます。ポリシーで指定できる一連の権限があり、それは要素「Action」、または除外の場合は「NotAction」で示されます。これらの各要素は、特定のS3 REST API操作にマッピングされます。

この表には、バケットに適用される権限とオブジェクトに適用される権限が一覧表示されています。

Amazon S3は現在、PutBucketReplicationアクションとDeleteBucketReplicationアクションの両方にs3:PutReplicationConfiguration権限を使用します。StorageGRIDは各アクションに個別の権限を使用します。これは、元のAmazon S3の仕様と一致しています。

削除は、put操作を使用して既存の値を上書きする際に実行されます。

バケットに適用される権限

権限	**S3 REST API** オペレーション	**StorageGRID** のカスタム
s3：CreateBucket	CreateBucket	○ 注：グループポリシーでのみ使用してください。

権限	**S3 REST API** オペレーション	**StorageGRID** のカスタム
s3:DeleteBucket	DeleteBucket	
s3:DeleteBucketMetadataNotification	バケットメタデータ通知設定の削除	はい
s3:DeleteBucketPolicy	DeleteBucketPolicy	
s3:DeleteReplicationConfiguration	DeleteBucketReplication	はい、PUT とDELETEには別々の権限があります
s3:GetBucketAcl	GetBucketAcl	
s3:GetBucketCompliance	GET Bucket compliance（非推奨）	はい
s3:GetBucketConsistency	GET Bucket consistency	はい
s3:GetBucketCORS	GetBucketCors	
s3:GetEncryptionConfiguration	GetBucketEncryption	
s3:GetBucketLastAccessTime	GET Bucketの最終アクセス時間	はい
s3:GetBucketLocation	GetBucketLocation	
s3:GetBucketMetadataNotification	バケットメタデータ通知設定を取得します	はい
s3:GetBucketNotification	GetBucketNotificationConfiguration	
s3:GetBucketObjectLockConfiguration	GetObjectLockConfiguration	
s3:GetBucketPolicy	GetBucketPolicy	
s3:GetBucketTagging	GetBucketTagging	
s3:GetBucketVersioning	GetBucketVersioning	
s3:GetLifecycleConfiguration	GetBucketLifecycleConfiguration	
s3:GetReplicationConfiguration	GetBucketReplication	

権限	**S3 REST API** オペレーション	**StorageGRID** のカスタム
s3:ListAllMyBuckets	• ListBuckets • ストレージ使用量を取得する	はい、GET Storage Usageの場合です。 注：グループポリシーでのみ使用してください。
s3：ListBucket	• ListObjects • HeadBucket • RestoreObject	
s3:ListBucketMultipartUploads	• ListMultipartUploads • RestoreObject	
s3:ListBucketVersions	GET Bucket versions	
s3:PutBucketCompliance	PUT Bucket compliance（非推奨）	はい
s3:PutBucketConsistency	PUT Bucketの一貫性	はい
s3:PutBucketCORS	• DeleteBucketCors† • PutBucketCors	
s3:PutEncryptionConfiguration	• DeleteBucketEncryption • PutBucketEncryption	
s3:PutBucketLastAccessTime	PUT Bucketの最終アクセス時刻	はい
s3:PutBucketMetadataNotification	PUT Bucketメタデータ通知設定	はい
s3:PutBucketNotification	PutBucketNotificationConfiguration	
s3:PutBucketObjectLockConfiguration	• CreateBucket と `x-amz-bucket-object-lock-enabled: true`リクエストヘッダー（s3:CreateBucket 権限も必要） • PutObjectLockConfiguration	
s3：PutBucketPolicy	PutBucketPolicy	
s3：PutBucketTagging	• DeleteBucketTagging† • PutBucketTagging	

権限	**S3 REST API** オペレーション	**StorageGRID** のカスタム
s3:PutBucketVersioning	PutBucketVersioning	
s3:PutLifecycleConfiguration	• DeleteBucketLifecycle† • PutBucketLifecycleConfiguration	
s3:PutReplicationConfiguration	PutBucketReplication	はい、PUT とDELETEには別々の権限があります

オブジェクトに適用される権限

権限	**S3 REST API** オペレーション	**StorageGRID** のカスタム
s3:AbortMultipartUpload	• AbortMultipartUpload • RestoreObject	
s3:BypassGovernanceRetention	• DeleteObject • DeleteObjects • PutObjectRetention	
s3:DeleteObject	• DeleteObject • DeleteObjects • RestoreObject	
s3:DeleteObjectTagging	DeleteObjectTagging	
s3:DeleteObjectVersionTagging	DeleteObjectTagging（オブジェクトの特定バージョン）	
s3:DeleteObjectVersion	DeleteObject（オブジェクトの特定バージョン）	
s3:GetObject	• GetObject • HeadObject • RestoreObject • SelectObjectContent	
s3:GetObjectAcl	GetObjectAcl	
s3:GetObjectLegalHold	GetObjectLegalHold	

権限	**S3 REST API** オペレーション	**StorageGRID** のカスタム
s3:GetObjectRetention	GetObjectRetention	
s3：GetObjectTagging	GetObjectTagging	
s3:GetObjectVersionTagging	GetObjectTagging（オブジェクトの特定バージョン）	
s3:GetObjectVersion	GetObject（オブジェクトの特定バージョン）	
s3:ListMultipartUploadParts	ListParts、RestoreObject	
s3:PutObject	• PutObject • CopyObject • RestoreObject • CreateMultipartUpload • CompleteMultipartUpload • UploadPart • UploadPartCopy	
s3:PutObjectLegalHold	PutObjectLegalHold	
s3:PutObjectRetention	PutObjectRetention	
s3:PutObjectTagging	PutObjectTagging	
s3:PutObjectVersionTagging	PutObjectTagging（オブジェクトの特定バージョン）	
s3:PutOverwriteObject	• PutObject • CopyObject • PutObjectTagging • DeleteObjectTagging • CompleteMultipartUpload	はい
s3:RestoreObject	RestoreObject	

PutOverwriteObject 権限を使用する

s3:PutOverwriteObject 権限は、オブジェクトの作成または更新を行う操作に適用されるカスタム StorageGRID 権限です。この権限の設定により、クライアントがオブジェクトのデータ、ユーザー定義のメ

タデータ、または S3 オブジェクトのタグ付けを上書きできるかどうかが決まります。

この権限の設定可能な項目は以下のとおりです：

- 許可：クライアントはオブジェクトを上書きできます。これはデフォルト設定です。
- 拒否：クライアントはオブジェクトを上書きできません。「拒否」に設定すると、PutOverwriteObject 権限の仕組みは以下のとおりです。
  - 同じパスに既存のオブジェクトが見つかった場合：
    - オブジェクトのデータ、ユーザー定義のメタデータ、またはS3オブジェクトのタグ付けは上書きできません。
    - 進行中の取り込み操作はすべてキャンセルされ、エラーが返されます。
    - S3 バージョン管理が有効になっている場合、拒否設定により、PutObjectTagging または DeleteObjectTagging 操作がオブジェクトとその非現行バージョンの TagSet を変更できなくなります。
  - 既存のオブジェクトが見つからない場合、この権限は無効です。
- この権限が存在しない場合、効果は「許可」が設定されている場合と同じです。

> ⓘ 現在のS3ポリシーで上書きが許可されており、PutOverwriteObject権限が「拒否」に設定されている場合、クライアントはオブジェクトのデータ、ユーザー定義のメタデータ、またはオブジェクトのタグ付けを上書きすることができません。さらに、クライアントの変更を防止する* チェックボックスが選択されている場合（*構成 > セキュリティ設定 > ネットワークとオブジェクト）、その設定はPutOverwriteObject権限の設定より優先されます。

ポリシーで条件を指定する

条件とは、ポリシーがいつ有効になるかを定めるものです。条件は、演算子とキーと値のペアで構成されます。

条件式は、評価にキーと値のペアを使用します。Condition要素には複数の条件を含めることができ、各条件には複数のキーと値のペアを含めることができます。条件ブロックは次の形式を使用します：

```
Condition: {
     condition_type: {
          condition_key: condition_values
```

次の例では、IpAddress 条件は SourceIp 条件キーを使用しています。

```
"Condition": {
    "IpAddress": {
      "aws:SourceIp": "54.240.143.0/24"
        ...
},
        ...
```

条件演算子は以下のように分類されます：

- 文字列
- 数字
- ブーリアン
- IPアドレス
- ヌルチェック

条件演算子	説明
StringEquals	キーと文字列値を完全一致（大文字小文字を区別）に基づいて比較します。
StringNotEquals	否定一致（大文字小文字を区別）に基づいて、キーと文字列値を比較します。
StringEqualsIgnoreCase	キーと文字列値を完全一致に基づいて比較します（大文字小文字は区別しません）。
StringNotEqualsIgnoreCase	否定一致に基づいてキーと文字列値を比較します（大文字と小文字は区別しません）。
StringLike	キーと文字列値を完全一致（大文字小文字を区別）に基づいて比較します。ワイルドカード文字として「*」と「?」を含めることができます。
StringNotLike	否定一致（大文字小文字を区別）に基づいて、キーと文字列値を比較します。ワイルドカード文字として * と？を含めることができます。
NumericEquals	キーと数値を完全一致に基づいて比較します。
NumericNotEquals	否定一致に基づいて、キーと数値を比較します。
NumericGreaterThan	キーと数値を「より大きい」という条件に基づいて比較します。
NumericGreaterThanEquals	キーと数値を「以上」の一致に基づいて比較します。
NumericLessThan	キーと数値を「より小さい」条件に基づいて比較します。
NumericLessThanEquals	キーと数値を「以下」のマッチングに基づいて比較します。
ブール値	キーをブール値と「true or false」の一致に基づいて比較します。
IpAddress	キーをIPアドレスまたはIPアドレスの範囲と比較します。

条件演算子	説明
NotIpAddress	否定一致に基づいて、キーをIPアドレスまたはIPアドレスの範囲と比較します。
Null	現在のリクエストコンテキストに条件キーが存在するかどうかを確認します。
IfExists	Null条件を除く、任意の条件演算子に追加して、その条件キーが存在しないかどうかを確認します。条件キーが存在しない場合は TRUE を返します。

サポートされている条件キー

条件キー	操作	説明
aws:SourceIp	IPオペレーター	リクエストが送信されたIPアドレスと比較します。バケット操作またはオブジェクト操作に使用できます。 注**:** S3 リクエストが管理ノードおよびゲートウェイノードのロードバランサーサービス経由で送信された場合、これはロードバランサーサービスの上流の IP アドレスと比較されます。 注: サードパーティ製の非透過型ロードバランサーを使用する場合、これはそのロードバランサーのIPアドレスと比較されます。 `X-Forwarded-For`ヘッダーは有効性を確認できないため、無視されます。
aws:username	リソース/アイデンティティ	リクエストが送信された送信者のユーザー名と比較します。バケット操作またはオブジェクト操作に使用できます。
s3:区切り文字	s3:ListBucket および s3:ListBucketVersions 権限	ListObjectsまたはListObjectVersionsリクエストで指定された区切り文字パラメータと比較されます。

条件キー	操作	説明
s3:ExistingObjectTag/<tag-key>	s3:DeleteObjectTagging s3:DeleteObjectVersionTagging s3:GetObject s3:GetObjectAcl 3：GetObjectTagging s3:GetObjectVersion s3:GetObjectVersionAcl s3:GetObjectVersionTagging s3:PutObjectAcl s3:PutObjectTagging s3:PutObjectVersionAcl s3:PutObjectVersionTagging	既存のオブジェクトに特定のタグキーと値が存在する必要があります。
s3:max-keys	s3:ListBucket および s3:ListBucketVersions 権限	ListObjectsまたはListObjectVersionsリクエストで指定された max-keys パラメータと比較します。
s3:object-lock-mode	s3:PutObject	`object-lock-mode`から展開された値と比較します。これは、PutObject、CopyObject、およびCreateMultipartUploadリクエストのリクエストヘッダーから展開されます。
s3:object-lock-mode	s3:PutObjectRetention	PutObjectRetentionリクエストのXML本文から展開された `object-lock-mode`と比較します。

条件キー	操作	説明
s3:object-lock-remaining-retention-days	s3:PutObject	`x-amz-object-lock-retain-until-date`リクエストヘッダーで指定された保持期限、またはバケットのデフォルト保持期間から計算された保持期限と比較して、これらの値が次のリクエストの許容範囲内にあることを確認します： • PutObject • CopyObject • CreateMultipartUpload
s3:object-lock-remaining-retention-days	s3:PutObjectRetention	PutObjectRetention リクエストで指定された保持期限日と比較し、許容範囲内であることを確認します。
s3:prefix	s3:ListBucket および s3:ListBucketVersions 権限	ListObjectsまたはListObjectVersionsリクエストで指定されたプレフィックスパラメータと比較します。
s3:RequestObjectTag/<tag-key>	s3:PutObject s3:PutObjectTagging s3:PutObjectVersionTagging	オブジェクト要求にタグ付けが含まれる場合、特定のタグキーと値が必要になります。
s3:x-amz-server-side-encryption-customer-algorithm	s3:PutObject	`sse-customer-algorithm`または`copy-source-sse-customer-algorithm`と比較します。これらは、PutObject、CopyObject、CreateMultipartUpload、UploadPart、UploadPartCopy、およびCompleteMultipartUploadリクエストのリクエストヘッダーから展開されます。

ポリシーで変数を指定する

ポリシー内で変数を使用すると、ポリシー情報が利用可能な場合に、その情報を入力できます。ポリシー変数は `Resource`要素と、 `Condition`要素内の文字列比較で使用できます。

この例では、変数 `${aws:username}`はリソース要素の一部です。

```
"Resource": "arn:aws:s3:::bucket-name/home/${aws:username}/*"
```

この例では、変数 `${aws:username}`は条件ブロック内の条件値の一部です。

```
"Condition": {
    "StringLike": {
      "s3:prefix": "${aws:username}/*"
        ...
},
        ...
```

変数	説明
`${aws:SourceIp}`	SourceIp キーを指定された変数として使用します。
`${aws:username}`	指定された変数としてユーザー名キーを使用します。
`${s3:prefix}`	サービス固有のプレフィックスキーを、指定された変数として使用します。
`${s3:max-keys}`	指定された変数として、サービス固有のmax-keysキーを使用します。
`${*}`	特殊文字。この文字をリテラルの「*」文字として使用します。
`${?}`	特殊文字。文字をリテラルの「?」文字として使用します。
`${$}`	特殊文字。文字をリテラルの $ 文字として使用します。

特別な取り扱いが必要なポリシーを作成する

ポリシーによっては、セキュリティ上危険であったり、継続的な運用に危険を及ぼすような権限を付与してしまう場合があります。例えば、アカウントのルートユーザーを締め出してしまうといった場合です。StorageGRID S3 REST API の実装は、ポリシー検証時には Amazon よりも制限が少ないですが、ポリシー評価時には Amazon と同程度に厳格です。

ポリシーの説明	ポリシータイプ	**Amazon** の動作	**StorageGRID**の動作
ルートアカウントに対するすべての権限を自分自身に拒否する	バケット	有効かつ強制適用されますが、root ユーザーアカウントはすべての S3 バケットポリシー操作に対する権限を保持します。	同じ
ユーザー/グループに対する自身の権限をすべて拒否する	グループ	有効かつ施行済み	同じ

ポリシーの説明	ポリシータイプ	**Amazon** の動作	**StorageGRID**の動作
外部アカウントグループに任意の権限を許可する	バケット	無効なプリンシパル	有効ですが、ポリシーで許可されている場合、すべてのS3 バケットポリシー操作の権限は 405 Method Not Allowed エラーを返します
外部アカウントのルートまたはユーザーにすべての権限を許可する	バケット	有効ですが、ポリシーで許可されている場合、すべてのS3 バケットポリシー操作の権限は 405 Method Not Allowed エラーを返します	同じ
すべてのアクションに対して全員に権限を付与する	バケット	有効ですが、すべての S3 バケットポリシー操作の権限は、外部アカウントの root とユーザーに対して 405 Method Not Allowed エラーを返します。	同じ
すべてのアクションに対する全員の権限を拒否する	バケット	有効かつ強制適用されますが、root ユーザーアカウントはすべての S3 バケットポリシー操作に対する権限を保持します。	同じ
プリンシパルは存在しないユーザーまたはグループです	バケット	無効なプリンシパル	有効
リソースは存在しないS3バケットです	グループ	有効	同じ
プリンシパルはローカルグループです	バケット	無効なプリンシパル	有効
このポリシーは、所有者以外のアカウント（匿名アカウントを含む）にオブジェクトを配置する権限を付与します。	バケット	有効。オブジェクトの所有権は作成者アカウントにあり、バケットポリシーは適用されません。作成者アカウントは、オブジェクトACLを使用してオブジェクトへのアクセス権限を付与する必要があります。	有効。オブジェクトの所有権は、バケット所有者のアカウントに帰属します。バケットポリシーが適用されます。

Write-once-read-many（WORM）保護

データ、ユーザー定義のオブジェクトメタデータ、およびS3オブジェクトのタグ付けを保護するために、Write Once Read Many（WORM）バケットを作成できます。WORMバケットを設定することで、新しい

オブジェクトの作成を許可し、既存コンテンツの上書きや削除を防止できます。ここで説明する方法のいずれかを使用してください。

上書きが常に拒否されるようにするには、次の方法があります。

- グリッドマネージャから、構成 > セキュリティ > セキュリティ設定 > ネットワークとオブジェクト に移動し、クライアントによる変更を防止する チェックボックスを選択します。
- 以下のルールとS3ポリシーを適用します。
  - S3ポリシーに PutOverwriteObject DENY 操作を追加します。
  - S3ポリシーに DeleteObject の DENY 操作を追加します。
  - S3ポリシーに PutObject の ALLOW 操作を追加します。

> S3ポリシーで DeleteObject を DENY に設定しても、「30日後にコピーをゼロにする」などのルールが存在する場合、ILM によるオブジェクトの削除を防ぐことはできません。

> これらのルールやポリシーをすべて適用したとしても、同時書き込みを防ぐことはできません（状況Aを参照）。これらは、完了した連続上書きを防ぐ対策を講じています（状況Bを参照）。

状況**A**：同時書き込み（対策なし）

```
/mybucket/important.doc
PUT#1 ---> OK
PUT#2 -------> OK
```

状況**B**：連続した上書き処理（対策済み）

```
/mybucket/important.doc
PUT#1 -------> PUT#2 ---X (denied)
```

関連情報

- "StorageGRID ILM ルールによるオブジェクトの管理方法"
- "バケットポリシーの例"
- "グループポリシーの例"
- "セッションポリシーの例"
- "ILMを使用してオブジェクトを管理する"
- "テナントアカウントを使用する"

StorageGRID S3 REST API セッションポリシーの例

次の例を使用して、StorageGRID セッションポリシーを作成します。

例：オブジェクトの取得を許可するセッションポリシーを設定する

この例では、セッションのプリンシパルはbucket1からオブジェクトを取得することのみが許可されています。その他のアクションはすべて暗黙的に拒否されますが、"s3:PutOverwriteObject"権限の使用などのStorageGRID固有のアクションは除きます。セッションポリシーは、AssumeRole APIの呼び出し時にJSONファイルとして提供できます。

```
{
  "Statement": [
    {
      "Action": "s3:GetObject",
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::bucket1/*"
    }
  ]
}
```

StorageGRID S3 REST API バケットポリシーの例

このセクションの例を使用して、バケットの StorageGRID アクセスポリシーを作成します。

バケットポリシーは、そのポリシーが適用されるバケットへのアクセス権限を指定します。S3 PutBucketPolicy API を使用して、以下のいずれかのツールからバケットポリシーを設定します：

- "Tenant Manager"。
- AWS CLI でこのコマンドを使用します（"バケットに対する操作" を参照）：

```
> aws s3api put-bucket-policy --bucket examplebucket --policy
file://policy.json
```

例：バケットへの読み取り専用アクセスを全員に許可する

この例では、匿名ユーザーを含むすべてのユーザーがバケット内のオブジェクトを一覧表示し、バケット内のすべてのオブジェクトに対してGetObject操作を実行することができます。その他のすべての操作は拒否されます。アカウントのルート以外にはバケットへの書き込み権限がないため、このポリシーはあまり役に立たない可能性があることに注意してください。

```
{
  "Statement": [
    {
      "Sid": "AllowEveryoneReadOnlyAccess",
      "Effect": "Allow",
      "Principal": "*",
      "Action": [ "s3:GetObject", "s3:ListBucket" ],
      "Resource":
["arn:aws:s3:::examplebucket","arn:aws:s3:::examplebucket/*"]
    }
  ]
}
```

例：あるアカウントの全員にバケットへのフルアクセス権限を、別のアカウントの全員には読み取り専用アクセス権限を付与する

この例では、指定された1つのアカウントのすべてのユーザーにはバケットへのフルアクセスが許可されますが、別の指定されたアカウントのすべてのユーザーには、バケットの一覧表示と、 `shared/`オブジェクトキープレフィックスで始まるバケット内のオブジェクトに対するGetObject操作の実行のみが許可されます。

> StorageGRID では、非所有者アカウント（匿名アカウントを含む）によって作成されたオブジェクトは、バケット所有者アカウントが所有します。バケットポリシーはこれらのオブジェクトに適用されます。

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "95390887230002558202"
      },
      "Action": "s3:*",
      "Resource": [
        "arn:aws:s3:::examplebucket",
        "arn:aws:s3:::examplebucket/*"
      ]
    },
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "31181711887329436680"
      },
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::examplebucket/shared/*"
    },
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "31181711887329436680"
      },
      "Action": "s3:ListBucket",
      "Resource": "arn:aws:s3:::examplebucket",
      "Condition": {
        "StringLike": {
          "s3:prefix": "shared/*"
        }
      }
    }
  ]
}
```

例：バケットへの読み取り専用アクセスを全員に許可し、指定されたグループにはフルアクセスを許可する

この例では、匿名ユーザーを含むすべてのユーザーがバケットを一覧表示し、バケット内のすべてのオブジェクトに対してGetObject操作を実行できますが、指定されたアカウントのグループ `Marketing`に属するユーザーのみがフルアクセス権限を持ちます。

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::95390887230002558202:federated-
group/Marketing"
      },
      "Action": "s3:*",
      "Resource": [
        "arn:aws:s3:::examplebucket",
        "arn:aws:s3:::examplebucket/*"
      ]
    },
    {
      "Effect": "Allow",
      "Principal": "*",
      "Action": ["s3:ListBucket","s3:GetObject"],
      "Resource": [
        "arn:aws:s3:::examplebucket",
        "arn:aws:s3:::examplebucket/*"
      ]
    }
  ]
}
```

例：クライアントが**IP**アドレス範囲内にいる場合、バケットへの読み取りおよび書き込みアクセスを全員に許可する

この例では、匿名ユーザーを含むすべてのユーザーが、指定されたIP範囲（54.240.143.0～54.240.143.255、ただし54.240.143.188を除く）からのリクエストであれば、バケットを一覧表示し、バケット内のすべてのオブジェクトに対して任意のオブジェクト操作を実行できます。その他の操作はすべて拒否され、IPアドレス範囲外からのすべてのリクエストも拒否されます。

```
{
  "Statement": [
    {
      "Sid": "AllowEveryoneReadWriteAccessIfInSourceIpRange",
      "Effect": "Allow",
      "Principal": "*",
      "Action": [ "s3:*Object", "s3:ListBucket" ],
      "Resource":
["arn:aws:s3:::examplebucket","arn:aws:s3:::examplebucket/*"],
      "Condition": {
        "IpAddress": {"aws:SourceIp": "54.240.143.0/24"},
        "NotIpAddress": {"aws:SourceIp": "54.240.143.188"}
      }
    }
  ]
}
```

例：指定されたフェデレーションユーザーのみにバケットへのフルアクセスを許可する

この例では、フェデレーションユーザーの Alex には、 `examplebucket`バケットとそのオブジェクトへのフルアクセスが許可されています。'root' を含むその他のすべてのユーザーは、すべての操作を明示的に拒否されます。ただし、'root' は Put/Get/DeleteBucketPolicy の権限を拒否されることはありません。

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::95390887230002558202:federated-user/Alex"
      },
      "Action": [
        "s3:*"
      ],
      "Resource": [
        "arn:aws:s3:::examplebucket",
        "arn:aws:s3:::examplebucket/*"
      ]
    },
    {
      "Effect": "Deny",
      "NotPrincipal": {
        "AWS": "arn:aws:iam::95390887230002558202:federated-user/Alex"
      },
      "Action": [
        "s3:*"
      ],
      "Resource": [
        "arn:aws:s3:::examplebucket",
        "arn:aws:s3:::examplebucket/*"
      ]
    }
  ]
}
```

例：**PutOverwriteObject** 権限

この例では、 `Deny` PutOverwriteObject および DeleteObject の Effect により、オブジェクトのデータ、ユーザー定義のメタデータ、および S3 オブジェクトタグが誰にも上書きまたは削除されないことが保証されます。

```
{
  "Statement": [
    {
      "Effect": "Deny",
      "Principal": "*",
      "Action": [
        "s3:PutOverwriteObject",
        "s3:DeleteObject",
        "s3:DeleteObjectVersion"
      ],
      "Resource": "arn:aws:s3:::wormbucket/*"
    },
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::95390887230002558202:federated-
group/SomeGroup"

},
      "Action": "s3:ListBucket",
      "Resource": "arn:aws:s3:::wormbucket"
    },
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::95390887230002558202:federated-
group/SomeGroup"

},
      "Action": "s3:*",
      "Resource": "arn:aws:s3:::wormbucket/*"
    }
  ]
}
```

StorageGRID S3 REST API グループポリシーの例

このセクションの例を使用して、グループ向けの StorageGRID アクセスポリシーを作成します。

グループポリシーは、そのポリシーが適用されるグループのアクセス権限を指定します。ポリシー内に`Principal`要素はありません。これは暗黙的であるためです。グループポリシーは、テナントマネージャーまたはAPIを使用して設定します。

例：**Tenant Manager** を使用してグループポリシーを設定する

テナントマネージャーでグループを追加または編集する際に、グループポリシーを選択して、そのグループのメンバーが持つS3アクセス権限を決定できます。"S3テナント用のグループを作成する"を参照してください。

- **S3**アクセスなし：デフォルトオプション。このグループのユーザーは、バケットポリシーによってアクセスが許可されない限り、S3リソースにアクセスできません。このオプションを選択すると、デフォルトではルートユーザーのみがS3リソースにアクセスできるようになります。
- 読み取り専用アクセス：このグループのユーザーは、S3リソースへの読み取り専用アクセス権限を持ちます。例えば、このグループのユーザーは、オブジェクトの一覧表示や、オブジェクトのデータ、メタデータ、タグの読み取りを行うことができます。このオプションを選択すると、読み取り専用グループポリシーのJSON文字列がテキストボックスに表示されます。この文字列は編集できません。
- フルアクセス：このグループのユーザーは、バケットを含む S3 リソースへのフルアクセス権限を持ちます。このオプションを選択すると、フルアクセス権限を持つグループポリシーの JSON 文字列がテキストボックスに表示されます。この文字列は編集できません。
- ランサムウェア対策：このサンプルポリシーは、このテナントのすべてのバケットに適用されます。このグループのユーザーは一般的な操作を実行できますが、オブジェクトのバージョン管理が有効になっているバケットからオブジェクトを完全に削除することはできません。

  「すべてのバケットを管理する」権限を持つテナントマネージャーユーザーは、このグループポリシーを上書きできます。「すべてのバケットを管理する」権限は信頼できるユーザーのみに制限し、利用可能な場合は多要素認証（MFA）を使用してください。

- カスタム：グループ内のユーザーには、テキストボックスで指定した権限が付与されます。

例：グループにすべてのバケットへのフルアクセスを許可する

この例では、バケットポリシーによって明示的に拒否されない限り、グループのすべてのメンバーは、テナントアカウントが所有するすべてのバケットへの完全なアクセスが許可されます。

```
{
  "Statement": [
    {
      "Action": "s3:*",
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::*"
    }
  ]
}
```

例：グループに対してすべてのバケットへの読み取り専用アクセスを許可する

この例では、バケットポリシーによって明示的に拒否されない限り、グループのすべてのメンバーはS3リソースへの読み取り専用アクセス権を持ちます。例えば、このグループのユーザーは、オブジェクトの一覧表示や、オブジェクトのデータ、メタデータ、タグの読み取りを行うことができます。

```
{
  "Statement": [
    {
      "Sid": "AllowGroupReadOnlyAccess",
      "Effect": "Allow",
      "Action": [
        "s3:ListAllMyBuckets",
        "s3:ListBucket",
        "s3:ListBucketVersions",
        "s3:GetObject",
        "s3:GetObjectTagging",
        "s3:GetObjectVersion",
        "s3:GetObjectVersionTagging"
      ],
      "Resource": "arn:aws:s3:::*"
    }
  ]
}
```

例：グループメンバーに、バケット内の自分の「フォルダー」のみへのフルアクセスを許可する

この例では、グループのメンバーは、指定されたバケット内の自分の特定のフォルダ（キープレフィックス）のみを一覧表示およびアクセスすることが許可されています。これらのフォルダのプライバシー設定を決定する際には、他のグループポリシーおよびバケットポリシーによるアクセス権限も考慮する必要があります。

```
{
  "Statement": [
    {
      "Sid": "AllowListBucketOfASpecificUserPrefix",
      "Effect": "Allow",
      "Action": "s3:ListBucket",
      "Resource": "arn:aws:s3:::department-bucket",
      "Condition": {
        "StringLike": {
          "s3:prefix": "${aws:username}/*"
        }
      }
    },
    {
      "Sid": "AllowUserSpecificActionsOnlyInTheSpecificUserPrefix",
      "Effect": "Allow",
      "Action": "s3:*Object",
      "Resource": "arn:aws:s3:::department-bucket/${aws:username}/*"
    }
  ]
}
```

StorageGRID監査ログで追跡されるS3操作

監査メッセージは StorageGRID サービスによって生成され、テキストログファイルに保存されます。監査ログにある S3 固有の監査メッセージを確認することで、バケットやオブジェクトの操作に関する詳細情報を取得できます。

監査ログに記録されるバケット操作

- CreateBucket
- DeleteBucket
- DeleteBucketTagging
- DeleteObjects
- GetBucketTagging
- HeadBucket
- ListObjects
- ListObjectVersions
- PUT Bucketコンプライアンス
- PutBucketTagging
- PutBucketVersioning

監査ログで追跡されるオブジェクト操作

- CompleteMultipartUpload
- CopyObject
- DeleteObject
- GetObject
- HeadObject
- PutObject
- RestoreObject
- SelectObject
- UploadPart（ILMルールがバランス型または厳密型の取り込みを使用する場合）
- UploadPartCopy（ILMルールがバランス型または厳密型の取り込みを使用する場合）

関連情報

- "アクセス監査ログファイル"
- "クライアントが監査メッセージを書き込む"
- "クライアント読み取り監査メッセージ"

著作権に関する情報

商標に関する情報