S3 REST APIを使用する

StorageGRID 11.7

NetApp
April 12, 2024

目次

S3 REST APIを使用する

S3 REST APIでサポートされるバージョンと更新

StorageGRID は、 Representational State Transfer （ REST ）の Web サービスのセットとして実装される Simple Storage Service （ S3 ）をサポートします。

S3 REST APIのサポートにより、S3 Webサービス用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の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S3 REST API呼び出しを使用している場合は、変更を最小限に抑える必要があります。

サポートされるバージョン

StorageGRID でサポートしている S3 および HTTP のバージョンは次のとおりです。

項目	バージョン
S3 仕様	_Simple Storage Service API Reference_2006-03-01
HTTP	1.1 HTTP の詳細については、 HTTP/1.1 （ RFC 7230~7235 ）を参照してください。 • 注： StorageGRID は、 HTTP/1.1 パイプラインをサポートしません。

関連情報

"IETF RFC 2616 ：『 Hypertext Transfer Protocol （ HTTP/1.1 ）』"

"Amazon Web Services （ AWS ）ドキュメント：「 Amazon Simple Storage Service API Reference"

S3 REST APIのサポートが更新されました

リリース。	コメント
11.7	• を追加しました "クイックリファレンス：サポートされるS3 API要求"。 • S3オブジェクトロックでのガバナンスモードの使用のサポートが追加されました。 • StorageGRID固有のサポートが追加されました `x-ntap-sg-cgr-replication-status` GET Object要求とHEAD Object要求の応答ヘッダー。このヘッダーは、グリッド間レプリケーションのオブジェクトのレプリケーションステータスを示します。 • SelectObjectContent要求でParquetオブジェクトがサポートされるようになりました。

リリース。	コメント
11.6	• の使用のサポートが追加されました `partNumber` GET Object要求とHEAD Object要求のRequestパラメータ。 • S3 オブジェクトロックのデフォルト保持モードとデフォルトの保持期間がバケットレベルでサポートされるようになりました。 • のサポートが追加されました `s3:object-lock-remaining-retention-days` オブジェクトに許可される保持期間の範囲を設定するためのPolicy Conditionキー。 • 単一のPUT Object処理のmaximum_recommended_sizeを5GiB（5、368、709、120バイト）に変更しました。5GB より大きいオブジェクトがある場合は、マルチパートアップロードを使用してください。
11.5	• バケットの暗号化の管理のサポートが追加されました。 • S3 オブジェクトのロックと廃止された従来の準拠要求のサポートを追加しました。 • バージョン管理されたバケットでの DELETE Multiple Objects の使用のサポートが追加されました。 • 。 `Content-MD5` 要求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 のサポートが追加されました。コスト割り当て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では、オブジェクトキー名がパフォーマンスのベストプラクティスに適合するように制限する必要はなくなりました。 • でバケット通知のサポートが追加されました `s3:ObjectRestore:Post` イベントタイプ。 • マルチパートの AWS サイズの上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のサイズは 5MiB から 5GiB の間にする必要があります。最後の部分は 5MiB より小さくすることができます。 • TLS 1.3のサポートが追加されました
11.3	• ユーザ指定のキーによるオブジェクトデータのサーバ側暗号化（ SSE-C ）がサポートされるようになりました。 • DELETE Bucket lifecycle、GET Bucket lifecycle、PUT Bucket lifecycleの各処理（Expirationアクションのみ）とがサポートされるようになりました `x-amz-expiration` 応答ヘッダー。 • PUT Object 、 PUT Object - Copy 、 Multipart Upload が更新されて、取り込み時に同期配置を使用する ILM ルールの影響を受けるようになりました。 • TLS 1.1 暗号はサポートされなくなりました。

リリース。	コメント
11.2	クラウドストレージプールで POST Object restore を使用できるようになりました。グループポリシーとバケットポリシーの ARN 、ポリシー条件キー、およびポリシー変数で AWS 構文を使用できるようになりました。StorageGRID 構文を使用する既存のグループポリシーと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 ARN やその他の構成 JSON / XML での使用に変更はありませんでした。
11.1	Cross-Origin Resource Sharing（CORS）、グリッドノードへのS3クライアント接続でのHTTP、バケットでの準拠設定のサポートが追加されました。
11.0	バケットでのプラットフォームサービス（ CloudMirror レプリケーション、通知、および Elasticsearch 検索統合）の設定がサポートされるようになりました。また、バケットに対するオブジェクトタグ付け機能の場所の制約、および整合性制御設定「 available 」がサポートされるようになりました。
10.4.	ILM スキャンのバージョン管理、エンドポイントドメインの名前ページの更新、ポリシーの条件と変数、ポリシーの例、および PutOverwriteObject 権限の変更のサポートが追加されました。
10.3	バージョン管理のサポートが追加されました。
10.2	グループとバケットのアクセスポリシー、およびマルチパートコピー（ Upload Part - Copy ）のサポートが追加されました。
10.1	マルチパートアップロード、仮想ホスト形式の要求、および v4 認証のサポートが追加されました。
10.0	StorageGRID システムで S3 REST API のサポートが初めて導入されました。現在サポートされているバージョンの _Simple Storage Service API Reference_is 2006-03-01 。

クイックリファレンス：サポートされる**S3 API**要求

このページでは、StorageGRID がAmazon Simple Storage Service（S3）APIをどのようにサポートしているかをまとめます。

このページには、StorageGRID でサポートされるS3処理のみが含まれています。

各処理のAWSドキュメントを参照するには、見出しのリンクを選択します。

一般的な**URI**クエリパラメータと要求ヘッダー

特に記載がない限り、次の一般的なURIクエリパラメータがサポートされます。

- `versionId`（オブジェクトの処理に必要な場合）

特に記載がないかぎり、次の一般的な要求ヘッダーがサポートされます。

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

関連情報

- "S3 REST APIの実装の詳細"
- "Amazon Simple Storage Service API Reference：Common Request Headers"

"AbortMultipartUpload の略"

URIクエリパラメータと要求ヘッダー

StorageGRID はすべてをサポートします 共通のパラメータとヘッダー この要求に加えて、次の追加のURIクエリパラメータを指定します。

- `uploadId`

本文を要求します

なし

StorageGRID のドキュメント

"マルチパートアップロードの処理"

"CompleteMultipartUpload"

URIクエリパラメータと要求ヘッダー

StorageGRID はすべてをサポートします 共通のパラメータとヘッダー この要求に加えて、次の追加のURIクエリパラメータを指定します。

- `uploadId`

本文**XML**タグを要求します

StorageGRID は、次の要求本文XMLタグをサポートしています。

- `CompleteMultipartUpload`

- `Part`
- `ETag`
- `PartNumber`

StorageGRID のドキュメント

"Complete Multipart Upload の実行"

"CopyObject"

URIクエリパラメータと要求ヘッダー

StorageGRID はすべてをサポートします 共通のパラメータとヘッダー この要求に加え、次のヘッダーが追加されています。

- `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>`

本文を要求します

なし

StorageGRID のドキュメント

"PUT Object - Copyの略"

"CreateBucketを選択します"

URIクエリパラメータと要求ヘッダー

StorageGRID はすべてをサポートします 共通のパラメータとヘッダー この要求に加え、次のヘッダーが追加されています。

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

本文を要求します

StorageGRID は、実装時にAmazon S3 REST APIで定義されたすべての要求本文パラメータをサポートします。

StorageGRID のドキュメント

"バケットの処理"

"CreateMultipartUpload を実行します"

URIクエリパラメータと要求ヘッダー

StorageGRID はすべてをサポートします 共通のパラメータとヘッダー この要求に加え、次のヘッダーが追加されています。

- `Cache-Control`
- `Content-Disposition`
- `Content-Encoding`
- `Content-Language`
- `Expires`
- `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>`

本文を要求します

なし

StorageGRID のドキュメント

"マルチパートアップロードを開始します"

"DeleteBucketの場合"

URIクエリパラメータと要求ヘッダー

StorageGRID はすべてをサポートします 共通のパラメータとヘッダー この要求のために。

StorageGRID のドキュメント

"バケットの処理"

"DeleteBucketCors"

URIクエリパラメータと要求ヘッダー

StorageGRID はすべてをサポートします 共通のパラメータとヘッダー この要求のために。

本文を要求します

なし

StorageGRID のドキュメント

"バケットの処理"

"DeleteBucketEncryption"

URIクエリパラメータと要求ヘッダー

StorageGRID はすべてをサポートします 共通のパラメータとヘッダー この要求のために。

本文を要求します

なし

StorageGRID のドキュメント

"バケットの処理"

"DeleteBucketLifecycle"

URIクエリパラメータと要求ヘッダー

StorageGRID はすべてをサポートします 共通のパラメータとヘッダー この要求のために。

本文を要求します

なし

StorageGRID のドキュメント

- "バケットの処理"
- "S3 ライフサイクル設定を作成する"

"DeleteBucketPolicyのようになります"

URIクエリパラメータと要求ヘッダー

StorageGRID はすべてをサポートします 共通のパラメータとヘッダー この要求のために。

本文を要求します

なし

StorageGRID のドキュメント

"バケットの処理"

"DeleteBucketReplication"

URIクエリパラメータと要求ヘッダー

StorageGRID はすべてをサポートします 共通のパラメータとヘッダー この要求のために。

本文を要求します

なし

StorageGRID のドキュメント

"バケットの処理"

"DeleteBucketTagging"

URIクエリパラメータと要求ヘッダー

StorageGRID はすべてをサポートします 共通のパラメータとヘッダー この要求のために。

本文を要求します

なし

StorageGRID のドキュメント

"バケットの処理"

"deleteObject"

URIクエリパラメータと要求ヘッダー

StorageGRID はすべてをサポートします 共通のパラメータとヘッダー この要求と追加の要求ヘッダー：

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

本文を要求します

なし

StorageGRID のドキュメント

"オブジェクトの処理"

"オブジェクトを削除します"

URIクエリパラメータと要求ヘッダー

StorageGRID はすべてをサポートします 共通のパラメータとヘッダー この要求と追加の要求ヘッダー：

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

本文を要求します

StorageGRID は、実装時にAmazon S3 REST APIで定義されたすべての要求本文パラメータをサポートします。

StorageGRID のドキュメント

"オブジェクトの処理" （複数のオブジェクトの削除）

"DeleteObjectTagging の場合"

StorageGRID はすべてをサポートします 共通のパラメータとヘッダー この要求のために。

本文を要求します

なし

StorageGRID のドキュメント

"オブジェクトの処理"

"GetBucketAcl"

URIクエリパラメータと要求ヘッダー

StorageGRID はすべてをサポートします 共通のパラメータとヘッダー この要求のために。

本文を要求します

なし

StorageGRID のドキュメント

"バケットの処理"

"GetBucketCors"

URIクエリパラメータと要求ヘッダー

StorageGRID はすべてをサポートします 共通のパラメータとヘッダー この要求のために。

本文を要求します

なし

StorageGRID のドキュメント

"バケットの処理"

"GetBucketEncryptionの略"

URIクエリパラメータと要求ヘッダー

StorageGRID はすべてをサポートします 共通のパラメータとヘッダー この要求のために。

本文を要求します

なし

StorageGRID のドキュメント

"バケットの処理"

"GetBucketLifecycleConfiguration"

URIクエリパラメータと要求ヘッダー

StorageGRID はすべてをサポートします 共通のパラメータとヘッダー この要求のために。

本文を要求します

なし

StorageGRID のドキュメント

- "バケットの処理" （GET Bucket lifecycle）
- "S3 ライフサイクル設定を作成する"

"GetBucketLocation"

URIクエリパラメータと要求ヘッダー

StorageGRID はすべてをサポートします 共通のパラメータとヘッダー この要求のために。

本文を要求します

なし

StorageGRID のドキュメント

"バケットの処理"

"GetBucketNotificationConfigurationを参照してください"

URIクエリパラメータと要求ヘッダー

StorageGRID はすべてをサポートします 共通のパラメータとヘッダー この要求のために。

本文を要求します

なし

StorageGRID のドキュメント

"バケットの処理" （バケット通知を取得）

"GetBucketPolicyのようになります"

URIクエリパラメータと要求ヘッダー

StorageGRID はすべてをサポートします 共通のパラメータとヘッダー この要求のために。

本文を要求します

なし

StorageGRID のドキュメント

"バケットの処理"

"GetBucketReplicationの略"

URIクエリパラメータと要求ヘッダー

StorageGRID はすべてをサポートします 共通のパラメータとヘッダー この要求のために。

本文を要求します

なし

StorageGRID のドキュメント

"バケットの処理"

"GetBucketTagging"

URIクエリパラメータと要求ヘッダー

StorageGRID はすべてをサポートします 共通のパラメータとヘッダー この要求のために。

本文を要求します

なし

StorageGRID のドキュメント

"バケットの処理"

"GetBucketVersioningの各ノードの設定"

URIクエリパラメータと要求ヘッダー

StorageGRID はすべてをサポートします 共通のパラメータとヘッダー この要求のために。

本文を要求します

なし

StorageGRID のドキュメント

"バケットの処理"

"GetObject"

URIクエリパラメータと要求ヘッダー

StorageGRID はすべてをサポートします 共通のパラメータとヘッダー この要求に加えて、次の追加のURIクエリパラメータを使用します。

- `partNumber`
- `response-cache-control`
- `response-content-disposition`
- `response-content-encoding`
- `response-content-language`
- `response-content-type`

- response-expires

追加の要求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

本文を要求します
なし

StorageGRID のドキュメント
"オブジェクトの取得"

"GetObjectAcl"

URIクエリパラメータと要求ヘッダー
StorageGRID はすべてをサポートします 共通のパラメータとヘッダー この要求のために。

本文を要求します
なし

StorageGRID のドキュメント
"オブジェクトの処理"

"GetObjectLegalHold"

URIクエリパラメータと要求ヘッダー
StorageGRID はすべてをサポートします 共通のパラメータとヘッダー この要求のために。

本文を要求します
なし

StorageGRID のドキュメント
"S3 REST APIを使用してS3オブジェクトロックを設定します"

"GetObjectLockConfigurationの略"

URIクエリパラメータと要求ヘッダー

StorageGRID はすべてをサポートします 共通のパラメータとヘッダー この要求のために。

本文を要求します

なし

StorageGRID のドキュメント

"S3 REST APIを使用してS3オブジェクトロックを設定します"

"GetObjectRetentionの略"

URIクエリパラメータと要求ヘッダー

StorageGRID はすべてをサポートします 共通のパラメータとヘッダー この要求のために。

本文を要求します

なし

StorageGRID のドキュメント

"S3 REST APIを使用してS3オブジェクトロックを設定します"

"GetObjectTagging の 2 つの機能を"

URIクエリパラメータと要求ヘッダー

StorageGRID はすべてをサポートします 共通のパラメータとヘッダー この要求のために。

本文を要求します

なし

StorageGRID のドキュメント

"オブジェクトの処理"

"ヘッドバケット"

URIクエリパラメータと要求ヘッダー

StorageGRID はすべてをサポートします 共通のパラメータとヘッダー この要求のために。

本文を要求します

なし

StorageGRID のドキュメント

"バケットの処理"

"HeadObject （ヘッドオブジェクト）"

URIクエリパラメータと要求ヘッダー

StorageGRID はすべてをサポートします 共通のパラメータとヘッダー この要求に加え、次のヘッダーが追加されています。

- `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`

本文を要求します

なし

StorageGRID のドキュメント

"HEAD Object の実行"

"ListBuckets"

URIクエリパラメータと要求ヘッダー

StorageGRID はすべてをサポートします 共通のパラメータとヘッダー この要求のために。

本文を要求します

なし

StorageGRID のドキュメント

"サービス> Get Serviceに対する操作"

"ListMultipartUploads"

URIクエリパラメータと要求ヘッダー

StorageGRID はすべてをサポートします 共通のパラメータとヘッダー この要求に加え、次の追加パラメータが含まれます。

- `delimiter`
- `encoding-type`
- `key-marker`
- `max-uploads`
- `prefix`

- `upload-id-marker`

本文を要求します

なし

StorageGRID のドキュメント

"マルチパートアップロードをリストします"

"ListObjects"

URIクエリパラメータと要求ヘッダー

StorageGRID はすべてをサポートします 共通のパラメータとヘッダー この要求に加え、次の追加パラメータが含まれます。

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

本文を要求します

なし

StorageGRID のドキュメント

"バケットの処理" （GET Bucket）

"ListObjectsV2"

URIクエリパラメータと要求ヘッダー

StorageGRID はすべてをサポートします 共通のパラメータとヘッダー この要求に加え、次の追加パラメータが含まれます。

- `continuation-token`
- `delimiter`
- `encoding-type`
- `fetch-owner`
- `max-keys`
- `prefix`
- `start-after`

本文を要求します

なし

StorageGRID のドキュメント

"バケットの処理" （GET Bucket）

"ListObjectVersions"

URIクエリパラメータと要求ヘッダー

StorageGRID はすべてをサポートします 共通のパラメータとヘッダー この要求に加え、次の追加パラメータが含まれます。

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

本文を要求します

なし

StorageGRID のドキュメント

"バケットの処理" （バケットオブジェクトのバージョンを取得）

"ListParts"

URIクエリパラメータと要求ヘッダー

StorageGRID はすべてをサポートします 共通のパラメータとヘッダー この要求に加え、次の追加パラメータが含まれます。

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

本文を要求します

なし

StorageGRID のドキュメント

"マルチパートアップロードをリストします"

"PutBucketCorsの略"

URIクエリパラメータと要求ヘッダー

StorageGRID はすべてをサポートします 共通のパラメータとヘッダー この要求のために。

本文を要求します

StorageGRID は、実装時にAmazon S3 REST APIで定義されたすべての要求本文パラメータをサポートします。

StorageGRID のドキュメント

"バケットの処理"

"PutBucketEncryptionの略"

URIクエリパラメータと要求ヘッダー

StorageGRID はすべてをサポートします 共通のパラメータとヘッダー この要求のために。

本文**XML**タグを要求します

StorageGRID は、次の要求本文XMLタグをサポートしています。

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

StorageGRID のドキュメント

"バケットの処理"

"PutBucketLifecycleConfigurationの略"

URIクエリパラメータと要求ヘッダー

StorageGRID はすべてをサポートします 共通のパラメータとヘッダー この要求のために。

本文**XML**タグを要求します

StorageGRID は、次の要求本文XMLタグをサポートしています。

- `NewerNoncurrentVersions`
- `LifecycleConfiguration`
- `Rule`
- `Expiration`
- `Days`
- `Filter`
- `And`
- `Prefix`
- `Tag`
- `Key`
- `Value`
- `Prefix`
- `Tag`
- `Key`

- `Value`
- `ID`
- `NoncurrentVersionExpiration`
- `NoncurrentDays`
- `Prefix`
- `Status`

StorageGRID のドキュメント

- "バケットの処理" （PUT Bucket lifecycle）
- "S3 ライフサイクル設定を作成する"

"PutBucketNotificationConfigurationの略"

URIクエリパラメータと要求ヘッダー

StorageGRID はすべてをサポートします 共通のパラメータとヘッダー この要求のために。

本文**XML**タグを要求します

StorageGRID は、次の要求本文XMLタグをサポートしています。

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

StorageGRID のドキュメント

"バケットの処理" （PUT Bucket通知）

"PutBucketPolicyのように指定します"

URIクエリパラメータと要求ヘッダー

StorageGRID はすべてをサポートします 共通のパラメータとヘッダー この要求のために。

本文を要求します

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

"PutBucketReplicationの略"

URIクエリパラメータと要求ヘッダー

StorageGRID はすべてをサポートします 共通のパラメータとヘッダー この要求のために。

本文**XML**タグを要求します

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

StorageGRID のドキュメント

"バケットの処理"

"PutBucketTaggingの略"

URIクエリパラメータと要求ヘッダー

StorageGRID はすべてをサポートします 共通のパラメータとヘッダー この要求のために。

本文を要求します

StorageGRID は、実装時にAmazon S3 REST APIで定義されたすべての要求本文パラメータをサポートします。

StorageGRID のドキュメント

"バケットの処理"

"PutBucketVersioningの各ノードの設定"

URIクエリパラメータと要求ヘッダー

StorageGRID はすべてをサポートします 共通のパラメータとヘッダー この要求のために。

本文パラメータを要求します

StorageGRID は、次の要求本文パラメータをサポートしています。

- `VersioningConfiguration`
- `Status`

StorageGRID のドキュメント

"バケットの処理"

"PutObject"

URIクエリパラメータと要求ヘッダー

StorageGRID はすべてをサポートします 共通のパラメータとヘッダー この要求に加え、次のヘッダーが追加されています。

- `Cache-Control`
- `Content-Disposition`
- `Content-Encoding`
- `Content-Language`
- `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>`

本文を要求します

- オブジェクトのバイナリデータ

StorageGRID のドキュメント

"PUT Object の場合"

"PutObjectLegalHold"

URIクエリパラメータと要求ヘッダー

StorageGRID はすべてをサポートします 共通のパラメータとヘッダー この要求のために。

本文を要求します

StorageGRID は、実装時にAmazon S3 REST APIで定義されたすべての要求本文パラメータをサポートします。

StorageGRID のドキュメント

"S3 REST APIを使用してS3オブジェクトロックを設定します"

"PutObjectLockConfigurationの略"

URIクエリパラメータと要求ヘッダー

StorageGRID はすべてをサポートします 共通のパラメータとヘッダー この要求のために。

本文を要求します

StorageGRID は、実装時にAmazon S3 REST APIで定義されたすべての要求本文パラメータをサポートします。

StorageGRID のドキュメント

"S3 REST APIを使用してS3オブジェクトロックを設定します"

"PutObjectRetentionの略"

URIクエリパラメータと要求ヘッダー

StorageGRID はすべてをサポートします 共通のパラメータとヘッダー この要求とこの追加ヘッダー：

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

本文を要求します

StorageGRID は、実装時にAmazon S3 REST APIで定義されたすべての要求本文パラメータをサポートします。

StorageGRID のドキュメント

"S3 REST APIを使用してS3オブジェクトロックを設定します"

"PutObjectTagging の 2 つのグループが"

URIクエリパラメータと要求ヘッダー

StorageGRID はすべてをサポートします 共通のパラメータとヘッダー この要求のために。

本文を要求します

StorageGRID は、実装時にAmazon S3 REST APIで定義されたすべての要求本文パラメータをサポートします。

StorageGRID のドキュメント

"オブジェクトの処理"

"SelectObjectContent の順に選択します"

URIクエリパラメータと要求ヘッダー

StorageGRID はすべてをサポートします 共通のパラメータとヘッダー この要求のために。

本文を要求します

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"
- "オブジェクトコンテンツを選択します"

"UploadPart のアップロード"

URIクエリパラメータと要求ヘッダー

StorageGRID はすべてをサポートします 共通のパラメータとヘッダー この要求に加えて、次の追加のURIクエリパラメータを使用します。

- `partNumber`
- `uploadId`

追加の要求ヘッダーは次のとおりです。

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

本文を要求します

- 部品のバイナリデータ

StorageGRID のドキュメント

"パーツをアップロードします"

"UploadPartCopyをクリックします"

URIクエリパラメータと要求ヘッダー

StorageGRID はすべてをサポートします 共通のパラメータとヘッダー この要求に加えて、次の追加のURIクエリパラメータを使用します。

- `partNumber`
- `uploadId`

追加の要求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`

本文を要求します

なし

StorageGRID のドキュメント

"パーツのアップロード - コピー"

テナントアカウントと接続を設定する

クライアントアプリケーションからの接続を受け入れるように StorageGRID を設定するには、テナントアカウントを 1 つ以上作成し、接続を設定する必要があります。

S3 テナントアカウントを作成して設定します

S3 API クライアントが StorageGRID でオブジェクトの格納や読み出しを行うには、 S3 テナントアカウントが必要です。各テナントアカウントには、独自のアカウントID、グループ、ユーザ、バケット、およびオブジェクトがあります。

S3 テナントアカウントは、 StorageGRID のグリッド管理者がグリッドマネージャまたはグリッド管理 API を使用して作成します。を参照してください "テナントを管理します" を参照してください。S3テナントアカウントが作成されると、テナントユーザはTenant Managerにアクセスしてグループ、ユーザ、アクセスキー、およびバケットを管理できるようになります。を参照してください "テナントアカウントを使用する" を参照してください。

S3テナントユーザはTenant Managerを使用してS3アクセスキーとバケットを作成、管理できますが、オブジェクトを取り込み、管理するにはS3クライアントアプリケーションを使用する必要があります。を参照してください "S3 REST APIを使用する" を参照してください。

クライアント接続の設定方法

グリッド管理者は、 S3 クライアントがデータの格納と読み出しを行うために StorageGRID に接続する方法に関連する設定を行います。StorageGRID を任意のS3アプリケーションに接続するには、基本的に次の4つの手順を実行します。

- クライアントアプリケーションがStorageGRID に接続する方法に基づいて、StorageGRID で前提条件となるタスクを実行します。
- StorageGRID を使用して、アプリケーションがグリッドに接続するために必要な値を取得します。どちらでもかまいません "S3セットアップウィザードを使用します" または、各StorageGRID エンティティを手動で設定します。
- S3アプリケーションを使用して、StorageGRID への接続を完了します。DNSエントリを作成して、使用するドメイン名にIPアドレスを関連付けます。
- アプリケーションとStorageGRID で継続的なタスクを実行し、時間の経過に伴うオブジェクトストレージの管理と監視を行います。

これらの手順の詳細については、を参照してください "クライアント接続を設定します"。

クライアント接続に必要な情報

S3クライアントアプリケーションは、オブジェクトの格納や読み出しを行うために、すべての管理ノードおよびゲートウェイノードに含まれるロードバランササービスまたはすべてのストレージノードに含まれるLocal Distribution Router（LDR）サービスに接続します。

クライアントアプリケーションは、グリッドノードのIPアドレスとそのノード上のサービスのポート番号を使用してStorageGRID に接続できます。必要に応じて、ロードバランシングノードのハイアベイラビリティ（HA）グループを作成して、仮想IP（VIP）アドレスを使用する可用性の高い接続を確立できます。IPアドレスまたはVIPアドレスの代わりに完全修飾ドメイン名（FQDN）を使用してStorageGRID に接続する場合は、DNSエントリを設定できます。

を参照してください "Summary ：クライアント接続の IP アドレスとポート" を参照してください。

HTTPS 接続または **HTTP** 接続を使用するかどうかを決定します

ロードバランサエンドポイントを使用してクライアント接続を行う場合は、そのエンドポイントに指定されているプロトコル（ HTTP または HTTPS ）を使用して接続を確立する必要があります。ストレージノードへのクライアント接続にHTTPを使用するには、HTTPの使用を有効にする必要があります。

デフォルトでは、クライアントアプリケーションがストレージノードに接続する際に、すべての接続に暗号化されたHTTPSを使用する必要があります。必要に応じて、Grid Managerで*>[セキュリティ設定]>[ネットワークとオブジェクト]>[ストレージノード接続用のHTTPを有効にする]*を選択して、安全性の低いHTTP接続を有効にすることができます。たとえば、非本番環境でストレージノードへの接続をテストする際に、クライアントアプリケーションで HTTP を使用できます。

> 要求と応答が暗号化されずに送信されるため、本番環境のグリッドでHTTPを有効にする場合は注意が必要です。

関連情報

"StorageGRID の管理"

"アクティブ、アイドル、および同時 HTTP 接続のメリット"

S3要求のS3エンドポイントのドメイン名

クライアント要求にS3エンドポイントのドメイン名を使用するには、StorageGRID 管理者が、S3パス形式およびS3仮想ホスト形式の要求でS3エンドポイントのドメイン名を使用する接続を受け入れるようにシステムを設定する必要があります。

このタスクについて

S3 仮想ホスト形式の要求を使用できるようにするには、グリッド管理者が次のタスクを実行する必要があります。

- Grid Manager を使用して、 S3 エンドポイントのドメイン名を StorageGRID システムに追加します。
- クライアントが StorageGRID への HTTPS 接続に使用する証明書が、クライアントが必要とするすべてのドメイン名に対して署名されていることを確認します。

  たとえば、S3 APIサービスエンドポイントのドメインエンドポイントがの場合などです `s3.company.com`グリッド管理者は、HTTPS接続に使用する証明書にがあることを確認する必要があります `s3.company.com` サブジェクトの共通名として、およびサブジェクトの別名として、およびを使

用します `*.s3.company.com` サブジェクトの別名。

- "DNSサーバを設定します" クライアントが使用して、S3エンドポイントのドメイン名に一致するDNSレコード（必要なワイルドカードレコードを含む）を追加します。

クライアントがロードバランササービスを使用して接続する場合、グリッド管理者は、クライアントが使用するロードバランサエンドポイントの証明書を設定します。

ロードバランサエンドポイントにはそれぞれ独自の証明書があり、各エンドポイントは異なるS3エンドポイントのドメイン名を認識するように設定できます。

クライアントがストレージノードに接続する場合、グリッド管理者は、グリッドに使用される単一のカスタムサーバ証明書を設定します。

の手順を参照してください "StorageGRID の管理" を参照してください。

これらの手順が完了したら、仮想ホスト形式の要求を使用できます。

S3 REST API の設定をテストします

Amazon Web Services コマンドラインインターフェイス（ AWS CLI ）を使用してシステムへの接続をテストし、システムに対するオブジェクトの読み取りと書き込みが可能であることを確認できます。

作業を開始する前に

- AWS CLI をからダウンロードしてインストールしておきます "aws.amazon.com/cli"。
- StorageGRID システムで S3 テナントアカウントを作成しておきます。
- テナントアカウントでアクセスキーを作成しておきます。

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を設定します。
   a. コンフィギュレーションモードを開始します。 `aws configure`
   b. 作成したアカウントのアクセスキーIDを入力します。
   c. 作成したアカウントのシークレットアクセスキーを入力します。
   d. 使用するデフォルトのリージョン（ us-east-1 など）を入力します。
   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 を選択します。
2. バケットを作成する。

   この例では、IPアドレス10.96.101.17とポート10443を使用するようにロードバランサエンドポイントが設定されていると想定しています。

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

   バケットの作成が完了すると、次の例の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
```

オブジェクトのアップロードが完了すると、オブジェクトデータのハッシュである Etag が返されます。

4. バケットの内容をリストして、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 プラットフォームサービスのサポート

StorageGRID プラットフォームサービスでは、 StorageGRID のテナントアカウントでリモート S3 バケット、 Simple Notification Service （ SNS ）エンドポイント、 Elasticsearch クラスタなどの外部サービスを利用して、グリッドが提供するサービスを拡張できます。

次の表に、使用可能なプラットフォームサービスとその設定に使用する S3 API を示します。

プラットフォームサービス	目的	サービスの設定に使用する **S3 API**
CloudMirror レプリケーション	ソースの StorageGRID バケットから、設定したリモートの S3 バケットにオブジェクトをレプリケートします。	PUT Bucket replication （を参照 "バケットの処理")

プラットフォームサービス	目的	サービスの設定に使用する **S3 API**
通知	ソースの StorageGRID バケットでのイベントに関する通知を、設定した Simple Notification Service （ SNS ）エンドポイントに送信します。	PUT Bucket通知（を参照 "バケットの処理"）
検索統合	StorageGRID バケットに格納されているオブジェクトメタデータを、設定した Elasticsearch インデックスに送信します。	"PUT Bucket metadata notification configuration のコマンドです" • 注： * これは StorageGRID のカスタム S3 API です。

グリッド管理者がテナントアカウントでプラットフォームサービスの使用を有効にするには、事前にプラットフォームサービスを使用できるようにする必要があります。を参照してください "StorageGRID の管理"。その後、テナント管理者が、テナントアカウントのリモートサービスを表すエンドポイントを作成する必要があります。この手順は、サービスを設定する前に実行する必要があります。を参照してください "テナントアカウントを使用する"。

プラットフォームサービスの使用に関する推奨事項

プラットフォームサービスを使用する前に、次の推奨事項を確認してください。

- CloudMirror のレプリケーション、通知、検索統合を必要とする S3 要求ではアクティブなテナントが 100 個を超えないようにすることを推奨します。アクティブなテナントが 100 を超えると、 S3 クライアントのパフォーマンスが低下する可能性があります。
- StorageGRID システムのS3バケットで、バージョン管理とCloudMirrorレプリケーションの両方が有効になっている場合は、デスティネーションエンドポイントでもS3バケットのバージョン管理を有効にすることを推奨します。これにより、 CloudMirror レプリケーションでエンドポイントに同様のオブジェクトバージョンを生成できます。
- ソースバケットで S3 オブジェクトのロックが有効になっている場合、 CloudMirror レプリケーションはサポートされません。
- デスティネーションバケットでレガシー準拠が有効になっていると、 CloudMirror レプリケーションは AccessDenied エラーで失敗します。

StorageGRID での S3 REST API の実装

競合するクライアント要求です

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

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

整合性制御

整合性制御では、アプリケーションの必要に応じて、オブジェクトの可用性と異なるストレージノードおよびサイト間でのオブジェクトの整合性のバランスを調整できます。

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

別の整合性レベルでオブジェクトの処理を実行する場合は、各バケットまたは各 API 処理に対して整合性制御を指定できます。

整合性制御

整合性制御は、 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に分散される方法、つまりクライアント要求で使用できるオブジェクトの有無に影響します。

バケットまたは API 処理の整合性制御は、次のいずれかの値に設定できます。

- * all *：すべてのノードがすぐにデータを受信しないと、要求は失敗します。
- * strong-global * ：すべてのサイトにおけるすべてのクライアント要求について、リードアフターライト整合性が保証されます。
- * strong-site * ： 1 つのサイトにおけるすべてのクライアント要求について、リードアフターライト整合性が保証されます。
- * read-after-new-write *：（デフォルト）新規オブジェクトにはリードアフターライト整合性が提供され、オブジェクトの更新には結果整合性が提供されます。高可用性が確保され、データ保護が保証されます。ほとんどの場合に推奨されます。
- * available *：新しいオブジェクトとオブジェクトの更新の両方について、結果整合性を提供します。S3 バケットの場合は、必要な場合にのみ使用します（読み取り頻度の低いログ値を含むバケットや、存在しないキーに対するHEAD処理やGET処理など）。S3 FabricPool バケットではサポートされません。

「**read-after-new-write**」および「**available**」の整合性制御を使用します

HEAD 操作または GET 操作で「 read-after-new-write 」整合性制御を使用する場合、 StorageGRID は次のように複数の手順で検索を実行します。

- まず、低い整合性レベルを使用してオブジェクトを検索します。
- そのルックアップが失敗した場合は、次の整合性レベルでルックアップを繰り返し、strong-globalの動作と同じ整合性レベルに達します。

HEAD処理またはGET処理で「read-after-new-write」整合性制御が使用されているが、オブジェクトが存在しない場合、オブジェクトの検索は常にstrong-globalの動作と同じ整合性レベルに達します。この整合性レベルでは、オブジェクトメタデータのコピーが各サイトで複数ある必要があるため、同じサイトで使用できないストレージノードが複数ある場合に「500 Internal Server Error」が大量に発生する可能性があります。

Amazon S3 と同様の整合性の保証が必要でない限り、整合性制御を「 available 」に設定することで、 HEAD 処理と GET 処理でのこれらのエラーを防ぐことができます。 HEAD 操作または GET 操作で「 available 」整合性制御を使用する場合、 StorageGRID は結果整合性のみを提供します。失敗した処理が整合性レベルを上げて再試行されることはないため、オブジェクトメタデータの複数のコピーがある必要はありません。

API 処理に対して整合性制御を指定する

個々の API 処理に対して整合性制御を設定するには、その処理でサポートされている整合性制御を要求ヘッダーで指定する必要があります。次の例では、 GET Object 処理に対して、整合性制御を「 strong-site 」に設定しています。

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

PUT Object 処理と GET Object 処理には、同じ整合性制御を使用する必要があります。

バケットの整合性制御を指定します

バケットに対して整合性制御を設定するには、 StorageGRID の PUT Bucket 整合性要求および GET Bucket 整合性要求を使用できます。または、 Tenant Manager またはテナント管理 API を使用できます。

バケットの整合性制御を設定する際は、次の点に注意してください。

- バケットの整合性制御を設定することで、バケット内のオブジェクトまたはバケット設定に対して実行される S3 処理に、どの整合性制御を使用するかを指定できます。バケット自体に対する処理には影響しません。
- 個々の API 処理の整合性制御は、バケットの整合性制御よりも優先されます。
- 通常、バケットではデフォルトの整合性制御である「read-after-new-write」を使用する必要があります。要求が正しく動作しない場合は、可能であればアプリケーションクライアントの動作を変更します。または、 API 要求ごとに整合性制御を指定するようにクライアントを設定します。バケットレベルの整合性制御は最後の手段と考えてください。

[how-consistency-controls-and-ilm-rules-interact]]整合性制御と**ILM**ルールの相互作用によるデータ保護への影響

整合性制御と ILM ルールのどちらを選択した場合も、オブジェクトの保護方法に影響します。これらの設定は対話的に操作できます。

たとえば、オブジェクトの格納に使用される整合性制御はオブジェクトメタデータの初期配置に影響し、 ILM ルールで選択される取り込み動作はオブジェクトコピーの初期配置に影響します。StorageGRID では、クライアント要求に対応するためにオブジェクトのメタデータとそのデータの両方にアクセスする必要があるため、整合性レベルと取り込み動作に一致する保護レベルを選択することで、より適切な初期データ保護と予測可能なシステム応答を実現できます。

ILM ルールでは、次の取り込み動作を使用できます。

- * Dual commit *：StorageGRID はオブジェクトの中間コピーをただちに作成し、クライアントに成功を返します。可能な場合は、 ILM ルールで指定されたコピーが作成されます。
- * Strict * ： ILM ルールに指定されたすべてのコピーを作成しないと、クライアントに成功が返されません。

- * Balanced * ： StorageGRID は、取り込み時に ILM ルールで指定されたすべてのコピーを作成しようとします。作成できない場合、中間コピーが作成されてクライアントに成功が返されます。可能な場合は、ILM ルールで指定されたコピーが作成されます。

> ILM ルールの取り込み動作を選択する前に、情報ライフサイクル管理を使用してオブジェクトを管理する手順の設定の完全な概要 を確認してください。

整合性制御と **ILM** ルールの連動の例

次の ILM ルールと次の整合性レベル設定の 2 サイトグリッドがあるとします。

- * ILM ルール * ：ローカルサイトとリモートサイトに 1 つずつ、 2 つのオブジェクトコピーを作成します。Strict 取り込み動作が選択されています。
- * 整合性レベル *:"Strong-GLOBAL" (オブジェクトメタデータはすべてのサイトにただちに分散されます)

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

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

代わりに同じ ILM ルールと「 strong-site 」整合性レベルを使用する場合は、オブジェクトデータがリモートサイトにレプリケートされたあとで、オブジェクトメタデータがそこに分散される前に、クライアントに成功メッセージが送信され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がオブジェクトデータの保護レベルと一致しません。取り込み直後にローカルサイトが失われると、オブジェクトメタデータが失われます。オブジェクトを取得できません。

整合性レベルと ILM ルールの間の関係は複雑になる可能性があります。サポートが必要な場合は、ネットアップにお問い合わせください。

関連情報

"ILM を使用してオブジェクトを管理する"

"GET Bucket consistency"

"PUT Bucket consistency"

StorageGRID の ILM ルールによるオブジェクトの管理

グリッド管理者が情報ライフサイクル管理（ ILM ）ルールを作成して、 S3 REST API クライアントアプリケーションから StorageGRID システムに取り込まれたオブジェクトデータを管理します。これらのルールは、以降のオブジェクトデータを格納する方法と場所を指定するために、 ILM ポリシーに追加されます。

ILM の設定によって、オブジェクトの次の要素が決まります。

- * 地域 *

  StorageGRID システム（ストレージプール）内またはクラウドストレージプール内のオブジェクトのデータの場所。

- * ストレージグレード *

  フラッシュや回転式ディスクなど、オブジェクトデータの格納に使用されるストレージのタイプ。

- * 損失の保護 *

  作成されるコピーの数と作成されるコピーのタイプ（レプリケーション、イレイジャーコーディング、またはその両方）。

- * 保持 *

  オブジェクトのデータの管理方法、格納場所、損失からの保護方法の経過時間に応じて変更が加えられます。

- * 取り込み中の保護 *

  取り込み時にオブジェクトデータを保護する方法。同期配置（取り込み動作に Balanced オプションまたは Strict オプションを使用）または中間コピー作成（ Dual commit オプションを使用）のいずれかです。

ILM ルールではオブジェクトをフィルタして選択できます。S3 を使用して取り込まれたオブジェクトは、ILM ルールによって次のメタデータに基づいてフィルタできます。

- テナントアカウント
- バケット名
- 取り込み時間
- キーを押します
- 最終アクセス時間

> デフォルトでは、すべての S3 バケットで最終アクセス時間の更新が無効になっています。StorageGRID システムに[Last access time]オプションを使用するILMルールが含まれている場合は、そのルールで指定されたS3バケットに対して最終アクセス時間の更新を有効にする必要があります。Tenant ManagerでPUT Bucket last access time要求を使用します（を参照） "最終アクセス日時の更新を有効または無効にします"）をクリックするか、テナント管理APIを使用します。最終アクセス時間の更新を有効にする場合は、特に小さなオブジェクトを含むシステムで StorageGRID のパフォーマンスが低下する可能性があることに注意してください。

- 場所の制約
- オブジェクトのサイズ
- ユーザメタデータ
- オブジェクトタグ

関連情報

"テナントアカウントを使用する"

"ILM を使用してオブジェクトを管理する"

"PUT Bucket last access time のように指定します"

オブジェクトのバージョン管理

バージョン管理の機能を使用してオブジェクトの複数のバージョンを保持することで、オブジェクトが偶発的に削除される事態に対応したり、以前のバージョンのオブジェクトを読み出してリストアしたりできます。

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

オブジェクトのバージョン管理は、 StorageGRID の情報ライフサイクル管理（ ILM ）または S3 バケットのライフサイクル設定と組み合わせることができます。バケットでバージョン管理機能を有効にするには、各バケットに対して明示的に有効にする必要があります。バケット内の各オブジェクトには、 StorageGRID システムによって生成されるバージョン ID が割り当てられます。

MFA （多要素認証） Delete の使用はサポートされていません。

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

ILM とバージョン管理

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

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

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

を参照してください "S3 バージョン管理オブジェクトの ILM ルールとポリシー（例 4 ）"。

S3 REST APIを使用してS3オブジェクトロックを設定します

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

バケットでS3オブジェクトロックを有効にする方法

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

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

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

- Tenant Manager を使用してバケットを作成します。を参照してください "S3 バケットを作成する"。
- を指定したPUT Bucket要求を使用してバケットを作成します `x-amz-bucket-object-lock-enabled` 要求ヘッダー。を参照してください "バケットの処理"。

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

バケットのデフォルトの保持設定

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

デフォルトの保持モード

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

デフォルトの保持期間

各バケットのデフォルトの保持期間は、年または日数で指定できます。

バケットのデフォルトの保持期間を設定する方法

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

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

PUT Object Lock の設定を指定します

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

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

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

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

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

要求例

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

```
PUT /bucket?object-lock HTTP/1.1
Accept-Encoding: identity
Content-Length: 308
Host: host
Content-MD5: request header
User-Agent: s3sign/1.0.0 requests/2.24.0 python/3.8.2
X-Amz-Date: date
X-Amz-Content-SHA256: authorization-string
Authorization: authorization-string

<ObjectLockConfiguration>
    <ObjectLockEnabled>Enabled</ObjectLockEnabled>
    <Rule>
        <DefaultRetention>
            <Mode>COMPLIANCE</Mode>
            <Years>6</Years>
        </DefaultRetention>
    </Rule>
</ObjectLockConfiguration>
```

バケットのデフォルトの保持期間を確認する方法

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

- Tenant Managerでバケットを表示します。を参照してください "S3バケットを表示します"。
- 問題 GET Object Lock Configuration要求。

オブジェクトロック設定の取得

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

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

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

要求例

```
GET /bucket?object-lock HTTP/1.1
Host: host
Accept-Encoding: identity
User-Agent: aws-cli/1.18.106 Python/3.8.2 Linux/4.4.0-18362-Microsoft
botocore/1.17.29
x-amz-date: date
x-amz-content-sha256: authorization-string
Authorization: authorization-string
```

応答例

```
HTTP/1.1 200 OK
x-amz-id-2:
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なバケットには、S3オブジェクトロックの保持設定の有無に関係なく、オブジェクトを組み合わせて含めることができます。

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

オブジェクトごとに次の設定を指定できます。

- 保持モード：コンプライアンスまたはガバナンスのいずれか。
- * retain-until-date *：StorageGRID がオブジェクトバージョンを保持する期間を指定する日付。
  - コンプライアンスモードでは、retain-until-dateが将来の日付の場合、オブジェクトを読み出すことはできますが、変更や削除はできません。retain-until-dateは増やすことができますが、この日付を減らすことも削除することもできません。
  - ガバナンスモードでは、特別な権限を持つユーザーは、retain-until-date設定をバイパスできます。保持期間が経過する前にオブジェクトバージョンを削除できます。また、retain-until-dateを増減したり、削除したりすることもできます。
- * リーガルホールド * ：オブジェクトバージョンにリーガルホールドを適用すると、そのオブジェクトがただちにロックされます。たとえば、調査または法的紛争に関連するオブジェクトにリーガルホールドを設定する必要がある場合があります。リーガルホールドには有効期限はありませんが、明示的に削除されるまで保持されます。

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

バケットにオブジェクトバージョンを追加するときにS3オブジェクトロックの設定を指定するには、問題Aを実行します "PUT Object の場合"、 "PUT Object - Copy の各コマンドを実行します"または "マルチパートアップロードを開始します" リクエスト。

次のものを使用できます。

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

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

- `x-amz-object-lock-retain-until-date`
  - retain-une-dateの値は、の形式で指定する必要があります `2020-08-10T21:46:00Z`。秒数には分数を指定できますが、保持される 10 進数は 3 桁（ミリ秒単位）だけです。その他のISO 8601形式は使用できません。
  - retain-une-date は将来の日付にする必要があります。
- `x-amz-object-lock-legal-hold`

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

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

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

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

オブジェクトの保持設定を更新する方法

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

- `PUT Object legal-hold`

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

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

ガバナンスモードの使用方法

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

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

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

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

  コンプライアンスモードからガバナンスモードに変更することはできません。

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

関連情報

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

S3 ライフサイクル設定を作成する

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

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

ライフサイクル構成とは

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

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

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

バケットにライフサイクル設定を適用する場合、バケットのライフサイクル設定は常に StorageGRID の ILM 設定よりも優先されます。StorageGRID は、 ILM ではなくバケットの Expiration 設定を使用して、特定のオブジェクトを削除するか保持するかを決定します。

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

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

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

- バケットライフサイクルを削除
- GET Bucket lifecycle
- PUT Bucket lifecycle の場合

ライフサイクル構成を作成します

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

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

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

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

```
{
    "Rules": [
        {
            "ID": "rule1",
            "Filter": {
                "And": {
                    "Prefix": "category1/",
                    "Tags": [
                        {
                            "Key": "key2",
                            "Value": "tag2"
                        }
                    ]
                }
            },
            "Expiration": {
                "Date": "2020-08-22T00:00:00Z"
            },
            "Status": "Enabled"
        },
        {
            "ID": "rule2",
            "Filter": {
                "Prefix": "category2/"
            },
            "Expiration": {
                "Days": 100
            },
            "Status": "Enabled"
        },
        {
            "ID": "rule3",
            "Filter": {
                "Prefix": "category3/"
            },
            "NoncurrentVersionExpiration": {
                "NoncurrentDays": 50
            },
            "Status": "Enabled"
        }
    ]
}
```

バケットにライフサイクル設定を適用

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

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

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

ライフサイクル設定がバケットに正常に適用されたことを検証するために、問題 には GET Bucket lifecycle 要求があります。例：

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

成功応答には、適用したライフサイクル設定が表示されます。

バケットライフサイクルの有効期限が環境 オブジェクトであることを検証します

PUT Object 、 HEAD Object 、または GET Object 要求の発行時に、ライフサイクル設定の有効期限ルールが環境 の特定のオブジェクトかどうかを確認できます。ルールが適用される場合、応答にはが含まれます `Expiration` オブジェクトの有効期限と一致する有効期限を示すパラメータ。

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

たとえば、このPUT Object要求は2020年6月22日に実行され、にオブジェクトが配置されます `testbucket` バケット。

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

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

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

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

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

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

```
{
      "AcceptRanges": "bytes",
      *"Expiration": "expiry-date=\"Thu, 01 Oct 2020 09:07:48 GMT\", rule-
id=\"rule2\"",
      "LastModified": "2020-06-23T09:07:48+00:00",
      "ContentLength": 921,
      "ETag": "\"9762f8a803bc34f5340579d4446076f7\""
      "ContentType": "binary/octet-stream",
      "Metadata": {}
}
```

S3 REST API を実装する際の推奨事項

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

存在しないオブジェクトに対する HEAD の推奨事項

オブジェクトが実際に存在するとは思わないパスにオブジェクトが存在するかどうかをアプリケーションが定期的にチェックする場合は、「使用可能」整合性制御を使用する必要があります。たとえば ' アプリケーションがその場所に配置する前にその場所に注意する場合は ' 利用可能な整合性制御を使用する必要があります

そうしないと、使用できないストレージノードがある場合に HEAD 処理でオブジェクトが見つからないと、「 500 Internal Server Error 」が大量に返される可能性があります。

PUT Bucket consistency 要求を使用して各バケットに「 available 」整合性制御を設定するか、または個々の API 処理の要求ヘッダーで整合性制御を指定できます。

オブジェクトキーの推奨事項

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

StorageGRID 11.4以前で作成されたバケット

- オブジェクトキーの最初の4文字にランダムな値を使用しないでください。これは、 AWS が以前に推奨していたキープレフィックスの推奨事項とは異なります。代わりに、など、ランダムではなく一意ではないプレフィックスを使用します `image`。

- 以前のAWSの推奨事項に従ってキープレフィックスにランダムな一意の文字を使用する場合は、オブジェクトキーの前にディレクトリ名を付けます。つまり、次の形式を使用します。

  ```
  mybucket/mydir/f8e3-image3132.jpg
  ```

  次の形式は使用しないでください。

  ```
  mybucket/f8e3-image3132.jpg
  ```

StorageGRID 11.4以降で作成されたバケット

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

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

「範囲の読み取り」に関する推奨事項

状況に応じて "格納オブジェクトを圧縮するグローバルオプション" が有効になっている場合は、S3クライアントアプリケーションで返されるバイト数の範囲を指定するGET Object処理を実行しないようにする必要があります。StorageGRID は要求されたバイトにアクセスするためにオブジェクトを圧縮解除する必要があるため ' これらの “range read” 操作は非効率的です非常に大きなオブジェクトから小さい範囲のバイト数を要求する GET Object 処理は特に効率が悪く、たとえば、 50GB の圧縮オブジェクトから 10MB の範囲を読み取る処理は非効率的です。

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

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

関連情報

- "整合性制御"
- "PUT Bucket consistency"
- "StorageGRID の管理"

Amazon S3 REST APIのサポート

S3 REST APIの実装の詳細

StorageGRID システムは Simple Storage Service API （ API バージョン 2006-03-01 ）を実装しており、ほとんどの処理をサポートしていますが、いくつかの制限事項があります。S3 REST API クライアントアプリケーションを統合するときは、実装の詳細を理

解しておく必要があります。

StorageGRID システムでは、仮想ホスト形式の要求とパス形式の要求の両方がサポートされます。

日付の処理

S3 REST API の StorageGRID 実装では、有効な HTTP の日付形式のみをサポートしています。

StorageGRID システムでは、日付の値を設定できるすべてのヘッダーで、有効な HTTP の日付形式のみがサポートされます。日付の時刻の部分は、 Greenwich Mean Time （ GMT ；グリニッジ標準時）の形式で指定するか、タイムゾーンのオフセットなし（ +0000 を指定）の Universal Coordinated Time （ UTC ；協定世界時）の形式で指定できます。を指定する場合は `x-amz-date` 要求のヘッダー。Date要求ヘッダーで指定された値を上書きします。AWS署名バージョン4を使用している場合は、を参照してください `x-amz-date` 日付ヘッダーがサポートされていないため、署名済み要求にヘッダーが含まれている必要があります。

代表的な要求ヘッダー

StorageGRID システムは、で定義されている共通の要求ヘッダーをサポートします "Amazon Simple Storage Service API Reference：Common Request Headers"1 つの例外を除いて。

要求ヘッダー	実装
承認	AWS 署名バージョン 2 は完全にサポートされます AWS 署名バージョン 4 は次の例外を除いてサポートされます。 • 要求の本文の SHA256 の値は計算されません。ユーザが送信した値は、値の場合と同様に、検証なしで受け入れられます `UNSIGNED-PAYLOAD` は用に提供されていた `x-amz-content-sha256` ヘッダー。
x-amz-security-token を指定します	実装されていませんを返します `XNotImplemented`。

共通の応答ヘッダー

StorageGRID システムでは、以下の例外を除き、 _Simple Storage Service API Reference_で 定義されている共通の応答ヘッダーがすべてサポートされます。

応答ヘッダー	実装
x-amz-id-2	使用されません

要求を認証します

StorageGRID システムでは、 S3 API を使用したオブジェクトへのアクセスについて、認証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 では、 S3 API 要求の認証で署名バージョン 2 と署名バージョン 4 がサポートされます。

認証された要求は、アクセスキー ID とシークレットアクセスキーを使用して署名する必要があります。

StorageGRID システムでは、HTTPという2つの認証方式がサポートされています `Authorization` ヘッダーを使用し、クエリパラメータを使用する。

HTTP Authorization ヘッダーを使用します

HTTP `Authorization` ヘッダーは、バケットポリシーで許可された匿名の要求を除き、すべてのS3 API処理で使用されます。。 `Authorization` ヘッダーには、要求の認証に必要なすべての署名情報が含まれています。

クエリパラメータを使用します

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。これは署名付き URL 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場合に使用できます。指定されたURLを持つユーザは、リソースにアクセスする際にシークレットアクセスキーを知っている必要はありません。これにより、リソースへのサードパーティの制限付きアクセスを提供できます。

サービスの処理

StorageGRID システムでは、サービスに対して次の処理をサポートしています。

操作	実装
GET Service の略 `(ListBuckets)`	Amazon S3 REST API のすべての動作が実装されています。予告なく変更される場合があります。
GET Storage Usage の略	GET Storage Usage 要求を使用すると、アカウントで使用しているストレージの総容量とアカウントに関連付けられているバケットごとの使用容量を確認できます。これは、パス/とカスタムクエリパラメータを使用したサービスに対する処理です (`?x-ntap-sg-usage`)が追加されました
オプション /	クライアントアプリケーションは問題 を実行できます `OPTIONS /` S3認証クレデンシャルを入力せずにストレージノード上のS3ポートに要求し、ストレージノードが使用可能かどうかを確認します。この要求は監視に使用できるほか、外部のロードバランサがストレージノードの停止を特定する目的でも使用できます。

関連情報

"GET Storage Usage の略"

バケットの処理

StorageGRID システムでは、 S3 テナントアカウントあたり最大 1 、 000 個のバケットがサポートされます。

バケット名にはAWS US Standardリージョンの制限事項が適用されますが、S3仮想ホスト形式の要求をサポートするためにDNSの命名規則にも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 Web Services （ AWS ）ドキュメント：「 Bucket Restrictions and Limitations"
- "S3エンドポイントのドメイン名を設定"

GET Bucket （ List Objects ）処理と GET Bucket versions 処理では、 StorageGRID の整合性制御がサポートされます。

最終アクセス時間の更新が個々のバケットで有効になっているか無効になっているかを確認することができます。

次の表に、 StorageGRID での S3 REST API バケット処理の実装方法を示します。これらの処理を実行するには、アカウントに必要なアクセスクレデンシャルが付与されている必要があります。

操作	実装
バケットを削除します	この処理では、バケットが削除されます。
バケットの CORS を削除します	この処理は、バケットの CORS 設定を削除します。
バケットの暗号化を削除	この処理は、バケットからデフォルトの暗号化を削除します。既存の暗号化オブジェクトは暗号化されたままですが、バケットに追加された新しいオブジェクトは暗号化されません。
バケットライフサイクルを削除	この処理は、バケットからライフサイクル設定を削除します。を参照してください "S3 ライフサイクル設定を作成する"。
バケットポリシーを削除	この処理は、バケットに関連付けられているポリシーを削除します。
バケットレプリケーションを削除します	この処理は、バケットに関連付けられているレプリケーション設定を削除します。
バケットのタグ付けを削除します	この処理にはを使用します `tagging` サブリソース：バケットからすべてのタグを削除します。

操作	実装
GET Bucket （ListObjects） （ListObjectsV2）	この処理は、バケット内のオブジェクトの一部またはすべて（最大 1 、 000 ）を返します。を使用してオブジェクトを取り込んだ場合でも、オブジェクトのストレージクラスには2つの値が設定されます `REDUCED_REDUNDANCY` ストレージクラスのオプション： • `STANDARD`を指定します。このオブジェクトは、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ます。 • `GLACIER`を指定します。このオブジェクトは、クラウドストレージプールで指定された外部バケットに移動されています。 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、応答に一部のキーが含まれることがあります `CommonPrefixes` 鍵が入っていないものです
GET Bucket Object versions （ListObjectVersions）	バケットに対する読み取りアクセスで、を使用した処理 `versions` サブリソースには、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表示されます。
GET Bucket ACL の場合	この処理では、バケットの所有者にバケットに対するフルアクセスがあることを示す応答が返され、所有者の ID 、表示名、および権限が表示されます。
GET Bucket CORS	この処理を実行するとが返されます `cors` バケットの設定。
GET Bucket encryption	この処理は、バケットのデフォルトの暗号化設定を返します。
GET Bucket lifecycle （GetBucketLifecycleConfiguration）	この処理は、バケットのライフサイクル設定を返します。を参照してください "S3 ライフサイクル設定を作成する"。
GET Bucket location の各ノードで使用でき	この操作は、を使用して設定されたリージョンを返します `LocationConstraint` PUT Bucket要求の要素。バケットのリージョンがの場合 `us-east-1`を指定すると、リージョンに対して空の文字列が返されます。
GET Bucket notification （GetBucketNotificationConfiguration）	この処理は、バケットに関連付けられている通知設定を返します。
GET Bucket policy の場合	この処理は、バケットに関連付けられているポリシーを返します。
GET Bucket replication	この処理は、バケットに関連付けられているレプリケーション設定を返します。

操作	実装
GET Bucket tagging	この処理にはを使用します `tagging` サブリソース：バケットのすべてのタグを返す
GET Bucket versioning	この実装ではを使用します `versioning` サブリソース：バケットのバージョン管理の状態を返します。 • *blank*: バージョン管理は有効になっていません (バケットはバージョン管理されていません) • 有効：バージョン管理が有効になっています • 中断：バージョン管理は以前有効になっていて、中断されています
オブジェクトロック設定の取得	この処理では、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（設定されている場合）が返されます。 を参照してください "S3 REST APIを使用してS3オブジェクトロックを設定します"。
HEAD Bucket （ヘッドバケット）	この処理は、バケットが存在し、そのバケットへのアクセス権限があるかどうかを判断します。 この処理から返される情報は次の • `x-ntap-sg-bucket-id`：バケットのUUID（UUID形式）。 • `x-ntap-sg-trace-id`：関連付けられた要求の一意のトレースID。

操作	実装
PUT Bucket の場合	この処理は、新しいバケットを作成します。バケットを作成すると、そのバケットの所有者になります。 • バケット名は次のルールを満たす必要があります。 ◦ StorageGRID システム全体で（テナントアカウント内だけではなく）一意である必要があります。 ◦ DNS に準拠している必要があります。 ◦ 3 文字以上 63 文字以下にする必要があります。 ◦ 1 つ以上のラベルを連続して指定できます。隣接するラベルはピリオドで区切ります。各ラベルの先頭と末尾の文字は小文字のアルファベットか数字にする必要があり、使用できる文字は小文字のアルファベット、数字、ハイフンのみです。 ◦ テキスト形式の IP アドレスのようにはできません。 ◦ 仮想ホスト形式の要求でピリオドを使用しないでください。ピリオドを使用すると、サーバワイルドカード証明書の検証で原因 の問題が発生します。 • デフォルトでは、バケットはに作成されます `us-east-1` リージョン。ただし、を使用することはできます `LocationConstraint` 別のリージョンを指定するように要求本文内の要求要素。を使用する場合 `LocationConstraint` 要素：Grid Managerまたはグリッド管理APIを使用して定義されているリージョンの正確な名前を指定する必要があります。使用する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： StorageGRID で定義されていないリージョンを PUT Bucket 要求で使用すると、エラーが発生します。 • を含めることができます `x-amz-bucket-object-lock-enabled` S3オブジェクトのロックを有効にしてバケットを作成する要求ヘッダー。を参照してください "S3 REST APIを使用してS3オブジェクトロックを設定します"。 バケットの作成時に S3 オブジェクトのロックを有効にする必要があります。バケットの作成後にS3オブジェクトロックを追加または無効にすることはできません。S3 オブジェクトロックにはバケットのバージョン管理が必要です。バケットの作成時に自動的に有効になります。
PUT Bucket CORS	この処理は、バケットの CORS 設定を指定し、クロスオリジン要求を処理できるようにします。Cross-Origin Resource Sharing （ 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機能です。たとえば、というS3バケットを使用するとします `images` グラフィックを保存します。のCORS設定を指定します `images` バケットを使用すると、そのバケット内の画像をWebサイトに表示できます `http://www.example.com`。

操作	実装
PUT Bucket encryption	この処理は、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な場合は、バケットに追加されたすべての新しいオブジェクトが暗号化されます。 StorageGRID では、 StorageGRID で管理されるキーによるサーバ側の暗号化がサポートされます。サーバ側の暗号化設定ルールを指定する場合は、を設定します `SSEAlgorithm` パラメータの値 `AES256`を使用しないでください `KMSMasterKeyID` パラメータ バケットのデフォルトの暗号化設定は、オブジェクトのアップロード要求ですでに暗号化が指定されている場合（要求にが含まれている場合）は無視されます `x-amz-server-side-encryption-*` 要求ヘッダー）。
PUT Bucket lifecycle の場合 （PutBucketLifecycleConfiguration）	この処理は、バケットの新しいライフサイクル設定を作成するか、既存のライフサイクル設定を置き換えます。StorageGRID では、 1 つのライフサイクル設定で最大 1 、 000 個のライフサイクルルールがサポートされます。各ルールには、次の XML 要素を含めることができます。 • 有効期限（日数、日付） • NoncurrentVersionExpiration （ NoncurrentDays ） • フィルタ（プレフィックス、タグ） • ステータス • ID StorageGRID では、次のアクションはサポートされません。 • AbortIncompleteMultipartUpload の略 • ExpiredObjectDeleteMarker • 移行 を参照してください "S3 ライフサイクル設定を作成する"。バケットライフサイクルのExpirationアクションとILMの配置手順の相互作用については、を参照してください "オブジェクトのライフサイクル全体にわたる ILM の動作"。 • 注：バケットライフサイクル設定は S3 オブジェクトロックが有効なバケットで使用できますが、従来の準拠バケットではバケットライフサイクル設定がサポートされません。

操作	実装

PUT Bucket notification
 	
（PutBucketNotificationConfiguration）
	この処理は、要求の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。実装に関する次の詳細事項に注意してください。
	
	• StorageGRID では、 Simple Notification Service （ SNS ）のトピックがデスティネーションとしてサポートされます。Simple Queue Service（SQS）またはAmazon Lambdaエンドポイントはサポートされていません。
	• 通知のデスティネーションは、 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 またはテナント管理 API を使用して作成できます。
	
	通知設定が機能するためには、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は、 `400 Bad Request` エラーがコードとともに返されます `InvalidArgument`。
	
	• 次のイベントタイプに対して通知を設定することはできません。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、次のリストに示すように、一部のキーが含まれず、他のキーには特定の値が使用されます。
	◦ * eventSource*
	
	`sgws:s3`
	
	◦ * awsRegion *
	
	含まれません
	
	◦ * x-amz-id-2 *
	
	含まれません
	
	◦ * arn *
	
	`urn:sgws:s3:::bucket_name`

PUT Bucket policy の場合	この処理は、バケットに関連付けられているポリシーを設定します。

操作	実装
PUT Bucket replication	この操作は、を設定します "StorageGRID CloudMirrorレプリケーション" （バケット用）。要求の本文に含まれるレプリケーション設定XMLを使用します。CloudMirror レプリケーションについては、実装に関する次の詳細事項に注意してください。 • StorageGRID では、 V1 のレプリケーション設定のみがサポートされます。つまり、StorageGRID では、の使用はサポートされていません `Filter` ルールのエレメント。V1の規則に従ってオブジェクトバージョンを削除します。詳細については、を参照してください "レプリケーション設定に関する Amazon S3 のドキュメント"。 • バケットレプリケーションは、バージョン管理されているバケットでもバージョン管理されていないバケットでも設定でき • レプリケーション設定 XML の各ルールで異なるデスティネーションバケットを指定できます。1 つのソースバケットを複数のデスティネーションバケットにレプリケートできます。 • デスティネーションバケットは、テナントマネージャまたはテナント管理 API で指定された StorageGRID エンドポイントの URN として指定する必要があります。を参照してください "CloudMirror レプリケーションを設定します"。 レプリケーション設定が機能するためには、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は、として要求が失敗します `400 Bad Request`。エラーメッセージ： `Unable to save the replication policy. The specified endpoint URN does not exist: URN.` • を指定する必要はありません `Role` 設定XMLを使用します。この値は StorageGRID では使用されず、送信されても無視されます。 • 設定XMLでストレージクラスを省略した場合、StorageGRID ではを使用します `STANDARD` デフォルトのストレージクラス。 • ソースバケットからオブジェクトを削除する場合、またはソースバケット自体を削除する場合、クロスリージョンレプリケーションは次のように動作します。 ◦ レプリケートの前にオブジェクトまたはバケットを削除した場合、オブジェクトまたはバケットはレプリケートされず、通知も送信されません。 ◦ レプリケートのあとにオブジェクトまたはバケットを削除すると、StorageGRID は、 V1 のクロスリージョンレプリケーションに対する Amazon S3 の通常の削除動作に従います。

操作	実装
PUT Bucket tagging	この処理にはを使用します `tagging` サブリソース：バケットの一連のタグを追加または更新できます。バケットタグを追加する場合は、次の制限事項に注意してください。 • StorageGRID と Amazon S3 はどちらもバケットごとに最大 50 個のタグをサポートします。 • バケットに関連付けられているタグには、一意のタグキーが必要です。タグキーには Unicode 文字を 128 文字まで使用できます。 • タグ値には、 Unicode 文字を 256 文字以内で指定します。 • キーと値では大文字と小文字が区別されます。
PUT Bucket versioning の場合	この実装ではを使用します `versioning` サブリソース：既存のバケットのバージョン管理の状態を設定できます。バージョン管理の状態は、次のいずれかの値に設定できます。 • Enabled ：バケット内のオブジェクトに対してバージョン管理を有効にします。バケットに追加されるすべてのオブジェクトに、一意のバージョン ID が割り当てられます。 • Suspended ：バケット内のオブジェクトに対してバージョン管理を無効にします。バケットに追加されるすべてのオブジェクトに、バージョンIDが割り当てられます `null`。
PUT Object Lock の設定を指定します	この処理は、バケットのデフォルト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を変更した場合、既存のオブジェクトバージョンの retain-until はそのまま残り、新しいデフォルトの保持期間を使用して再計算されることはありません。 を参照してください "S3 REST APIを使用してS3オブジェクトロックを設定します" を参照してください。

関連情報

"整合性制御"

"GET Bucket last access time の場合"

"バケットとグループのアクセスポリシーを使用"

"監査ログで追跡される S3 処理"

バケットのカスタム処理

StorageGRID システムでは、 S3 REST API に追加されたシステム固有のカスタムバケット処理をサポートしています。

次の表に、 StorageGRID でサポートされるカスタムバケット処理を示します。

操作	説明	を参照してください。
GET Bucket consistency	特定のバケットに適用されている整合性レベルを返します。	"GET Bucket consistency"
PUT Bucket consistency	特定のバケットに適用される整合性レベルを設定します。	"PUT Bucket consistency"
GET Bucket last access time の場合	特定のバケットで最終アクセス時間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。	"GET Bucket last access time の場合"
PUT Bucket last access time のように指定します	特定のバケットの最終アクセス時間の更新を有効または無効にできます。	"PUT Bucket last access time のように指定します"
バケットのメタデータ通知設定を削除します	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。	"バケットのメタデータ通知設定を削除します"
GET Bucket metadata notification configuration	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 を返します。	"GET Bucket metadata notification configuration"
PUT Bucket metadata notification configuration のコマンドです	バケットのメタデータ通知サービスを設定します。	"PUT Bucket metadata notification configuration のコマンドです"
準拠設定の PUT Bucket	廃止およびサポート終了：準拠を有効にした新しいバケットを作成できなくなりました。	"廃止：準拠設定を指定した PUT Bucket"
GET Bucket compliance で確認します	廃止されましたがサポートされています：既存の古い準拠バケットに対して現在有効な準拠設定を返します。	"廃止予定：バケット準拠を取得します"
PUT Bucket compliance で確認してください	廃止されましたがサポートされています：既存の古い準拠バケットの準拠設定を変更できます。	"廃止予定：PUT Bucket compliance"

関連情報

"監査ログで追跡される S3 処理"

オブジェクトの処理

このセクションでは、 StorageGRID システムでオブジェクトの S3 REST API 処理を実

装する方法について説明します。

すべてのオブジェクトの処理に次の条件が適用されます。

- StorageGRID "整合性制御" オブジェクトに対するすべての操作でサポートされます。ただし、次の操作はサポートされません。
  - GET Object ACL の場合
  - `OPTIONS /`
  - オブジェクトのリーガルホールドを適用します
  - PUT Object retention のことです
  - オブジェクトコンテンツを選択します
- 同じキーに書き込む 2 つのクライアントなど、競合するクライアント要求は、「 latest-wins 」ベースで解決されます。「latest-wins」評価のタイミングは、S3クライアントが処理を開始するタイミングではなく、StorageGRID システムが特定の要求を完了したタイミングに基づいています。
- StorageGRID バケット内のオブジェクトは、匿名ユーザまたは別のアカウントが作成したオブジェクトも含めて、すべてバケット所有者によって所有されます。
- Swiftを使用してStorageGRID システムに取り込まれたデータオブジェクトにS3を使用してアクセスすることはできません。

次の表に、 StorageGRID での S3 REST API オブジェクト処理の実装方法を示します。

操作	実装
オブジェクトを削除します	多要素認証（MFA）と応答ヘッダー `x-amz-mfa` はサポートされていません。 StorageGRID は、 DELETE Object 要求を処理する際に、オブジェクトのすべてのコピーをすべての格納場所からただちに削除しようとします。成功すると、 StorageGRID はただちにクライアントに応答を返します。30秒以内にすべてのコピーを削除できない場合（場所が一時的に使用できない場合など）、StorageGRID は削除対象のコピーをキューに登録し、クライアントに成功を通知します。 バージョン管理 特定のバージョンを削除するには、バケットの所有者を要求元にしてを使用する必要があります `versionId` サブリソース：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。状況に応じて `versionId` 削除マーカー、応答ヘッダーに対応します `x-amz-delete-marker` はに設定されています `true`。 • を使用せずにオブジェクトが削除された場合 `versionId` バージョンが有効になっているバケットのサブリソースが表示されると、削除マーカーが生成されます。。 `versionId` 削除マーカーの場合は、を使用して戻ります `x-amz-version-id` 応答ヘッダー、および `x-amz-delete-marker` 応答ヘッダーがに設定されて返されます `true`。 • を使用せずにオブジェクトが削除された場合 `versionId` バージョンが一時停止中のバケットについて、既存の「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` 応答ヘッダーがに設定されて返されます `true`。 • 注 * ：特定の場合、 1 つのオブジェクトに複数の削除マーカーが存在することがあります。 を参照してください "S3 REST APIを使用してS3オブジェクトロックを設定します" ガバナンスモードでオブジェクトバージョンを削除する方法については、を参照してください。
複数のオブジェクトを削除します （DeleteObjects）	多要素認証（MFA）と応答ヘッダー `x-amz-mfa` はサポートされていません。 同じ要求メッセージで複数のオブジェクトを削除できます。 を参照してください "S3 REST APIを使用してS3オブジェクトロックを設定します" ガバナンスモードでオブジェクトバージョンを削除する方法については、を参照してください。

操作	実装
オブジェクトのタグ付けを削除します	を使用します `tagging` サブリソース：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 状況に応じて `versionId` クエリパラメータが要求で指定されていない場合、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が削除されます。オブジェクトの現在のバージョンが削除マーカーの場合は、"MethodNotAllowed`"ステータスがとともに返されます `x-amz-delete-marker` 応答ヘッダーをに設定しました `true`。
オブジェクトの取得	"オブジェクトの取得"
GET Object ACL の場合	アカウントに必要なアクセスクレデンシャルがある場合、オブジェクトの所有者にオブジェクトに対するフルアクセスがあることを示す応答が返され、所有者の ID 、表示名、および権限が表示されます。
オブジェクトのリーガルホールドを取得します	"S3 REST APIを使用してS3オブジェクトロックを設定します"
GET Object retention のことです	"S3 REST APIを使用してS3オブジェクトロックを設定します"
GET Object tagging	を使用します `tagging` サブリソース：オブジェクトのすべてのタグを返すために使用します。 バージョン管理 状況に応じて `versionId` クエリパラメータが要求で指定されていない場合、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が返されます。オブジェクトの現在のバージョンが削除マーカーの場合は、"MethodNotAllowed`"ステータスがとともに返されます `x-amz-delete-marker` 応答ヘッダーをに設定しました `true`。
HEAD Object の実行	"HEAD Object の実行"
POST Object restore の実行	"POST Object restore の実行"
PUT Object の場合	"PUT Object の場合"
PUT Object - Copy の各コマンドを実行します	"PUT Object - Copy の各コマンドを実行します"
オブジェクトのリーガルホールドを適用します	"S3 REST APIを使用してS3オブジェクトロックを設定します"

操作	実装
PUT Object retention のことです	"S3 REST APIを使用してS3オブジェクトロックを設定します"
PUT Object tagging	を使用します `tagging` サブリソース：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 タグは、新しいオブジェクトをアップロードするときに追加することも、既存のオブジェクトに追加することもできます。StorageGRID と Amazon S3 はどちらも、オブジェクトごとに最大 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。タグキーには Unicode 文字を 128 文字まで、タグ値には Unicode 文字を 256 文字まで使用できます。キーと値では大文字と小文字が区別されます。 タグの更新と取り込み動作 PUT Object tagging を使用してオブジェクトのタグを更新した場合、 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 ルールで指定されている取り込み動作が使用されないことを意味します。更新によって発生したオブジェクト配置の変更は、通常のバックグラウンド ILM プロセスで ILM が再評価されるときに実施されます。 つまり、ILMルールの取り込み動作にStrictオプションが使用されている場合、必要なオブジェクト配置を実行できない場合（新たに必要な場所が使用できない場合など）は処理されません。更新されたオブジェクトは、必要な配置を実行可能になるまで現在の配置が維持されます。 競合の解決 同一キーに書き込む2つのクライアントなど'競合するクライアント要求は'最新のWINS形式で解決されます「latest-wins」評価のタイミングは、S3クライアントが処理を開始するタイミングではなく、StorageGRID システムが特定の要求を完了したタイミングに基づいています。 バージョン管理 状況に応じて `versionId` クエリパラメータが要求で指定されていません。処理は、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の場合は、"MethodNotAllowed`"ステータスがとともに返されます `x-amz-delete-marker` 応答ヘッダーをに設定しました `true`。
SelectObjectContent の順に選択します	"SelectObjectContent の順に選択します"

関連情報

"監査ログで追跡される S3 処理"

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

リストされていない項目はサポートされていません。

構文については、を参照してください "SelectObjectContent の順に選択します"。S3 Select の詳細については、を参照してください "S3 Select に関する AWS のドキュメント"。

問題 SelectObjectContent クエリを実行できるのは、 S3 Select が有効になっているテナントアカウントのみです。を参照してください "S3 Select を使用する際の考慮事項と要件"。

句

- リストを選択します
- FROM 句
- WHERE 句
- Limit 句

データ型

- ブール値
- 整数
- 文字列
- 浮動小数点
- 10 進数、数値
- タイムスタンプ

演算子

論理演算子

- および
- ありません
- または

比較演算子

- <
- >
- ⇐
- >=
- =

- =
- <>
- !=
- 間（ Between ）
- インチ

パターンマッチング演算子

- いいね
- _
- %

単一の演算子

- は NULL です
- は NULL ではありません

数学演算子

- [+]
- -
- *
- /
- %

StorageGRID はAmazon S3 Selectオペレータの優先順位に従います。

集合関数

- 平均 ()
- カウント（ * ）
- 最大 ()
- 最小（）
- 合計 ()

条件付き関数

- ケース
- 集合体
- NULLIF

変換関数

- CAST （サポートされているデータタイプ用）

日付関数

- date_add
- DATE_DIFF
- 抽出（ Extract ）
- 文字列まで（ _STRING ）
- 終了タイムスタンプ
- UTCNOW

文字列関数

- char_length 、 character_length
- 低い
- サブストリング
- トリム（ Trim ）
- 上限

サーバ側の暗号化を使用します

サーバ側の暗号化を使用して、保存中の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、ユーザが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。

サーバ側の暗号化を使用する場合は、暗号化キーの管理方法に基づいて、次の 2 つのオプションを同時に選択できます。

- * SSE （ StorageGRID で管理されるキーによるサーバ側の暗号化） * ：オブジェクトを格納する S3 要求を問題 で暗号化すると、 StorageGRID は一意のキーでオブジェクトを暗号化します。オブジェクトを読み出す S3 要求を問題 で実行すると、 StorageGRID は格納されているキーを使用してオブジェクトを復号化します。
- * SSE-C （ユーザ指定のキーによるサーバ側の暗号化） * ：オブジェクトを格納する S3 要求を問題 で処理するときに、独自の暗号化キーを指定します。オブジェクトを読み出すときは、同じ暗号化キーを要求に指定します。2 つの暗号化キーが一致すると、オブジェクトが復号化されてオブジェクトデータが返されます。

  オブジェクトの暗号化処理と復号化処理はすべて StorageGRID で管理されますが、指定する暗号化キーはユーザが管理する必要があります。

  指定した暗号化キーが格納されることはありません。暗号化キーを紛失すると、対応するオブジェクトが失われます。

  SSE または SSE-C で暗号化されたオブジェクトは、バケットレベルまたはグリッドレベルの暗号化設定が無視されます。

SSE を使用します

StorageGRID で管理される一意のキーでオブジェクトを暗号化する場合は、次の要求ヘッダーを使用します。

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

SSE 要求ヘッダーは、次のオブジェクト処理でサポートされます。

- "PUT Object の場合"
- "PUT Object - Copy の各コマンドを実行します"
- "マルチパートアップロードを開始します"

SSE-C を使用します

ユーザ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 3 つの要求ヘッダーを使用します。

要求ヘッダー	説明
`x-amz-server-side-encryption-customer-algorithm`	暗号化アルゴリズムを指定します。ヘッダー値はである必要があります `AES256`。
`x-amz-server-side-encryption-customer-key`	オブジェクトの暗号化と復号化に使用する暗号化キーを指定します。キーの値は、 Base64 でエンコードされた 256 ビットであることが必要です。
`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しで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 でエンコードされた 128 ビットであることが必要です。

SSE-C 要求ヘッダーは、次のオブジェクト処理でサポートされます。

- "オブジェクトの取得"
- "HEAD Object の実行"
- "PUT Object の場合"
- "PUT Object - Copy の各コマンドを実行します"
- "マルチパートアップロードを開始します"
- "パーツをアップロードします"
- "パーツのアップロード - コピー"

ユーザ指定のキーによるサーバ側の暗号化（ SSE-C ）を使用する場合の考慮事項

SSE-C を使用する場合は、次の考慮事項に注意してください。

- HTTPS を使用する必要があります。

SSE-C を使用すると、 http 経由の要求が StorageGRID ですべて拒否され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漏洩する可能性があります。キーを破棄し、必要に応じてローテーションします。

- 応答内の 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対応関係を管理する必要があります。StorageGRID では暗号化キーは格納されません。各オブジェクトに対して指定した暗号化キーを管理する責任はユーザにあります。
- バケットのバージョン管理が有効になっている場合は、オブジェクトのバージョンごとに固有の暗号化キーが必要です。各オブジェクトバージョンで使用される暗号化キーを管理する責任はユーザにあります。
- 暗号化キーはクライアント側で管理するため、キーローテーションなどの追加の防護策もクライアント側で管理する必要があります。

指定した暗号化キーが格納されることはありません。暗号化キーを紛失すると、対応するオブジェクトが失われます。

- 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は、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 開発者ガイド：「お客様が用意した暗号化キーによるサーバ側の暗号化（ SSE-C ）を使用したデータの保護」"

オブジェクトの取得

S3 GET Object 要求を使用して、 S3 バケットからオブジェクトを読み出すことができます。

オブジェクトとマルチパートオブジェクトを取得する

を使用できます `partNumber` マルチパー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読み出す要求パラメータ。。 `x-amz-mp-parts-count` response要素は、オブジェクトに含まれるパーツの数を示します。

設定できます `partNumber` セグメント化されたオブジェクト/マルチパートオブジェクトとセグメント化されていないオブジェクト/マルチパート以外のオブジェクトの場合は1。ただし、 `x-amz-mp-parts-count` 応答要素は、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。

ユーザメタデータ内の **UTF-8** 文字

StorageGRID は、ユーザ定義メタデータ内のエスケープされた UTF-8 文字を解析も解釈もしません。ユーザ定義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オブジェクトに対するGET要求では、が返されません `x-amz-missing-meta` キーの名前または値に印刷できない文字が含まれている場合は、ヘッダーを指定します。

サポートされない要求ヘッダーです

次の要求ヘッダーはサポートされていません `XNotImplemented`：

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

の場合 `versionId` サブリソースが指定されていません。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が取得されます。オブジェクトの現在のバージョンが削除マーカーの場合は、「見つからない」ステータスがとともに返されます `x-amz-delete-marker` 応答ヘッダーをに設定しました `true`。

ユーザ指定の暗号化キーによるサーバ側の暗号化（ **SSE-C** ）の要求ヘッダー

指定した一意のキーでオブジェクトが暗号化されている場合は、 3 つのヘッダーをすべて使用します。

- `x-amz-server-side-encryption-customer-algorithm`:指定します `AES256`。
- `x-amz-server-side-encryption-customer-key`:オブジェクトの暗号化キーを指定します
- `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。

> 指定した暗号化キーが格納されることはありません。暗号化キーを紛失すると、対応するオブジェクトが失われます。ユーザ指定のキーを使用してオブジェクトデータを保護する前に、の考慮事項を確認してください "サーバ側の暗号化を使用します"。

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

オブジェクトがに格納されている場合 "クラウドストレージプール"の場合、GET Object要求の動作はオブジェクトの状態によって異なります。を参照してください "HEAD Object の実行" 詳細：

> オブジェクトがクラウドストレージプールに格納され、かつそのオブジェクトのコピーがグリッドに 1 つ以上存在する場合、 GET Object 要求はクラウドストレージプールからデータを読み出す前に、グリッドからデータを読み出そうとします。

オブジェクトの状態	**GET Object** の動作
StorageGRID に取り込まれているがまだ ILM によって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格納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いるオブジェクト	`200 OK` オブジェクトのコピーが読み出されます。
クラウドストレージプール内にあるが、まだ読み出し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読み出されます。
オブジェクトを読み出し不可能な状態に移行した	`403 Forbidden、 InvalidObjectState` を使用します "POST Object restore の実行" 読み出し可能な状態へのオブジェクトのリストア要求。
読み出し不可能な状態からリストア中である	`403 Forbidden、 InvalidObjectState` POST Object restore 要求が完了するまで待ちます。

オブジェクトの状態	**GET Object** の動作
クラウドストレージプールへのリストアが完了している	`200 OK` オブジェクトのコピーが読み出されます。

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化されたオブジェクト

マルチパートオブジェクトをアップロードした場合や 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 はオブジェクトのパーツ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でクラウドストレージプール内のオブジェクトが使用可能かどうかを判断します。GET Object要求が誤って返されることがあります `200 OK` オブジェクトの一部のパートがすでに読み出し不可能な状態に移行されている場合や、オブジェクトの一部のパートがまだリストアされていない場合。

このような場合は、次のよう

- GET Object 要求がデータの一部を返し、転送の途中で停止することがあります。
- 後続のGET Object要求が返されることがあります `403 Forbidden`。

GET Objectとクロスグリッドレプリケーション

使用するポート "グリッドフェデレーション" および "グリッド間レプリケーション" バケットで有効になっている場合、S3クライアントはGET Object要求を発行して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固有の情報が含まれます `x-ntap-sg-cgr-replication-status` 応答ヘッダー。次のいずれかの値が設定されます。

グリッド（ **Grid** ）	レプリケーションのステータス
ソース	• 成功：レプリケーションは成功しました。 • * pending*：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エラーで失敗しました。ユーザーはエラーを解決する必要があります。
宛先	**replica**:オブジェクトはソースグリッドからレプリケートされました。

StorageGRID ではがサポートされません `x-amz-replication-status` ヘッダー。

関連情報

"監査ログで追跡される S3 処理"

HEAD Object の実行

S3 HEAD Object 要求を使用すると、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読み出すことができます。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は、 HEAD Object を使用してオブジェクトの移行状態を特定できます。

HEAD オブジェクトおよびマルチパートオブジェクト

を使用できます `partNumber` マルチパー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のメタデータを読み出す要求パラメータ。。 `x-amz-mp-parts-count` response要素は、オブジェクトに含まれるパーツの数を示します。

設定できます `partNumber` セグメント化されたオブジェクト/マルチパートオブジェクトとセグメント化されていないオブジェクト/マルチパート以外のオブジェクトの場合は1。ただし、 `x-amz-mp-parts-count` 応答要素は、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。

ユーザメタデータ内の UTF-8 文字

StorageGRID は、ユーザ定義メタデータ内のエスケープされた UTF-8 文字を解析も解釈もしません。ユーザ定義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オブジェクトに対するHEAD要求では、が返されません `x-amz-missing-meta` キーの名前または値に印刷できない文字が含まれている場合は、ヘッダーを指定します。

サポートされない要求ヘッダーです

次の要求ヘッダーはサポートされていません `XNotImplemented`：

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

バージョン管理

の場合 `versionId` サブリソースが指定されていません。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が取得されます。オブジェクトの現在のバージョンが削除マーカーの場合は、「見つからない」ステータスがとともに返されます `x-amz-delete-marker` 応答ヘッダーをに設定しました `true`。

ユーザ指定の暗号化キーによるサーバ側の暗号化（ SSE-C ）の要求ヘッダー

指定した一意のキーでオブジェクトが暗号化されている場合は、次の 3 つのヘッダーをすべて使用します。

- `x-amz-server-side-encryption-customer-algorithm`:指定します `AES256`。
- `x-amz-server-side-encryption-customer-key`:オブジェクトの暗号化キーを指定します
- `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。

> ⓘ 指定した暗号化キーが格納されることはありません。暗号化キーを紛失すると、対応するオブジェクトが失われます。ユーザ指定のキーを使用してオブジェクトデータを保護する前に、の考慮事項を確認してください "サーバ側の暗号化を使用します"。

クラウドストレージプールオブジェクトに対するHEAD Object応答

オブジェクトがに格納されている場合 "クラウドストレージプール"を指定すると、次の応答ヘッダーが返されます。

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

応答ヘッダーは、オブジェクトがクラウドストレージプールに移動され、必要に応じて読み出し不可能な状態

に移行されてリストアされるときの状態に関する情報を提供します。

オブジェクトの状態	**HEAD Object** への応答
StorageGRID に取り込まれているがまだ ILM によって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格納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いるオブジェクト	`200 OK` （特別な応答ヘッダーは返されません）。

クラウドストレージプール内にあるが、まだ読み出し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"`
	
	オブジェクトが読み出し不可能な状態に移行されるまでの間、の値 `expiry-date` は、将来の特定の日時に設定されます。移行の正確な時間は、StorageGRID システムでは制御されません。

オブジェクトが読み出し不可能な状態に移行したが、少なくとも 1 つのコピーがグリッドに存在する	`200 OK`
	
	`x-amz-storage-class: GLACIER`
	
	`x-amz-restore: ongoing-request="false", expiry-date="Sat, 23 July 20 2030 00:00:00 GMT"`
	
	の値 `expiry-date` は、将来の特定の日時に設定されます。
	
	注：グリッド上のコピーを使用できない場合（ストレージノードが停止している場合など）は、問題 を実行する必要があります "POST Object restore の実行" オブジェクトを読み出す前にクラウドストレージプールからコピーをリストアする要求。

読み出し不可能な状態に移行しており、グリッドにコピーが存在しない	`200 OK`
	
	`x-amz-storage-class: GLACIER`

読み出し不可能な状態からリストア中である	`200 OK`
	
	`x-amz-storage-class: GLACIER`
	
	`x-amz-restore: ongoing-request="true"`

オブジェクトの状態	**HEAD Object** への応答

クラウドストレージプールへのリストアが完了している	`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` クラウドストレージプール内のオブジェクトが読み出し不可能な状態に戻るタイミングを示します。

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化されたオブジェクト

マルチパートオブジェクトをアップロードした場合や 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 はオブジェクトのパーツ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でクラウドストレージプール内のオブジェクトが使用可能かどうかを判断します。HEAD Object要求が誤って返されることがあります `x-amz-restore: ongoing-request="false"` オブジェクトの一部のパートがすでに読み出し不可能な状態に移行されている場合や、オブジェクトの一部のパートがまだリストアされていない場合。

HEAD Objectとクロスグリッドレプリケーション

使用するポート "グリッドフェデレーション" および "グリッド間レプリケーション" バケットで有効になっている場合、S3クライアントはHEAD Object要求を発行して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固有の情報が含まれます `x-ntap-sg-cgr-replication-status` 応答ヘッダー。次のいずれかの値が設定されます。

グリッド（ **Grid** ）	レプリケーションのステータス
ソース	• 成功：レプリケーションは成功しました。 • * pending*：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エラーで失敗しました。ユーザーはエラーを解決する必要があります。
宛先	**replica**:オブジェクトはソースグリッドからレプリケートされました。

StorageGRID ではがサポートされません `x-amz-replication-status` ヘッダー。

関連情報

"監査ログで追跡される S3 処理"

POST Object restore の実行

S3 POST Object restore 要求を使用して、クラウドストレージプールに格納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要求タイプ

StorageGRID では、オブジェクトのリストアに POST Object restore 要求のみがサポートされます。ではサポートされません `SELECT` リストアのタイプ。戻り要求を選択してください `XNotImplemented`。

バージョン管理

必要に応じて、と指定します `versionId` 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のバージョンをリストアする。指定しない場合 `versionId`オブジェクトの最新バージョンがリストアされます

クラウドストレージプールオブジェクトでの **POST Object restore** の動作

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（情報ライフサイクル管理を使用してオブジェクトを管理する手順を参照）、 POST Object restore 要求はオブジェクトの状態に基づいて次のように動作します。詳細については、「 head Object 」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格納され、かつそのオブジェクトのコピーがグリッドに 1 つ以上存在する場合は、 POST Object restore 要求を実行してオブジェクトをリストアする必要はありません。GET Object 要求を使用してローカルコピーを直接読み出すことができます。

オブジェクトの状態	**POST Object restore** の動作
StorageGRID に取り込まれているがまだ ILM によって評価されていない、またはオブジェクトがクラウドストレージプールにない	`403 Forbidden、 InvalidObjectState`
クラウドストレージプール内にあるが、まだ読み出し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：オブジェクトが読み出し不可能な状態に移行されるまでは変更できません `expiry-date`。
オブジェクトを読み出し不可能な状態に移行した	`202 Accepted` 要求の本文で指定されている日数、オブジェクトの読み出し可能なコピーをクラウドストレージプールにリストアします。この期間が終了すると、オブジェクトは読み出し不可能な状態に戻ります。 必要に応じて、を使用します `Tier` リストアジョブの完了までにかかる時間を確認するための要求要素 (`Expedited、 Standard`または `Bulk`）。指定しない場合 `Tier、 Standard` 階層を使用しています。 重要：オブジェクトがS3 Glacier Deep Archiveに移行された場合、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は、を使用してリストアできません `Expedited` 階層：次のエラーが返されます `403 Forbidden、 InvalidTier： Retrieval option is not supported by this storage class`。
読み出し不可能な状態からリストア中である	`409 Conflict、 RestoreAlreadyInProgress`

オブジェクトの状態	**POST Object restore** の動作
クラウドストレージプールへのリストアが完了している	`200 OK` *注：*オブジェクトが読み出し可能な状態にリストアされている場合は、オブジェクトを変更できます `expiry-date` 用の新しい値を指定してPOST Object restore要求を再発行する `Days`。要求が実行された日時に基づいてリストア日が更新されます。

関連情報

"ILM を使用してオブジェクトを管理する"

"HEAD Object の実行"

"監査ログで追跡される S3 処理"

PUT Object の場合

S3 PUT Object 要求を使用すると、オブジェクトをバケットに追加できます。

競合を解決します

同じキーに書き込む 2 つのクライアントなど、競合するクライアント要求は、「 latest-wins 」ベースで解決されます。「 latest-wins 」評価は、 S3 クライアントが処理を開始するタイミングではなく、 StorageGRID システムが特定の要求を完了したタイミングで行われます。

オブジェクトのサイズ

単一 PUT Object 処理の maximum_recommended_size は 5GiB （ 5 、 368 、 709 、 120 バイト）です。5GB より大きいオブジェクトがある場合は、マルチパートアップロードを使用してください。

単一のPUT Object処理のmaximum_supported_sizeは5TiB（5、497、558、138、880バイト）です。ただし、5GiB 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。

ユーザメタデータのサイズ

Amazon S3 では、各 PUT 要求ヘッダー内のユーザ定義メタデータのサイズが 2KB に制限されます。StorageGRID では、ユーザメタデータが 24KiB に制限されます。ユーザ定義のメタデータのサイズは、各キーと値の UTF-8 エンコードでのバイト数の合計で測定されます。

ユーザメタデータ内の **UTF-8** 文字

要求のユーザ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 文字が含まれている場合、 StorageGRID の動作は定義されていません。

ユーザ定義メタデータのキー名または値に含まれているエスケープされた UTF-8 文字は、 StorageGRID で解析も解釈もされません。エスケープされた UTF-8 文字は ASCII 文字として扱われます。

- ユーザ定義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、 PUT 、 PUT Object-Copy 、 GET 、 HEAD の各要求は正常に実行されます。

- StorageGRID からが返されない `x-amz-missing-meta` キーの名前または値の解釈後の値に印刷不能文字が含まれている場合は、ヘッダー。

オブジェクトタグの制限

タグは、新しいオブジェクトをアップロードするときに追加することも、既存のオブジェクトに追加することもできます。StorageGRID と Amazon S3 はどちらも、オブジェクトごとに最大 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。タグキーには Unicode 文字を 128 文字まで、タグ値には Unicode 文字を 256 文字まで使用できます。キーと値では大文字と小文字が区別されます。

オブジェクトの所有権

StorageGRID では、非所有者アカウントまたは匿名ユーザによって作成されたオブジェクトを含むすべてのオブジェクトが、バケット所有者アカウントによって所有されます。

サポートされる要求ヘッダー

次の要求ヘッダーがサポートされています。

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

  を指定する場合 `aws-chunked` の場合 `Content-Encoding`StorageGRID では、次の項目は検証されません。

  - StorageGRID ではが検証されません `chunk-signature` チャンクデータに対して。
  - StorageGRID は、ユーザが指定した値を検証しません `x-amz-decoded-content-length` をクリックします。
- `Content-Language`
- `Content-Length`
- `Content-MD5`
- `Content-Type`
- `Expires`
- `Transfer-Encoding`

  チャンク転送エンコードは、の場合にサポートされます `aws-chunked` ペイロード署名も使用されます。

- `x-amz-meta-`をクリックし、続けてユーザ定義のメタデータを含む名前と値のペアを作成します。

  ユーザ定義メタデータの名前と値のペアを指定する場合、一般的な形式は次のとおりです。

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

  ILMルールの参照時間に*[ユーザ定義の作成時間]*オプションを使用する場合は、を使用する必要がありま

す `creation-time` を、オブジェクトの作成時に記録されたメタデータの名前として指定します。例：

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

の値 `creation-time` 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「ユーザ定義の作成時間」*を使用し、取り込み動作に「Balanced」または「Strict」オプションを使用することはできません。ILM ルールの作成時にエラーが返されます。

- `x-amz-tagging`
- S3 Object Lock 要求のヘッダー
  - `x-amz-object-lock-mode`
  - `x-amz-object-lock-retain-until-date`
  - `x-amz-object-lock-legal-hold`

    これらのヘッダーを指定せずに要求を行うと、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モードとretain-until-dateが計算されます。を参照してください "S3 REST APIを使用してS3オブジェクトロックを設定します"。
- SSE 要求ヘッダー：
  - `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`

    を参照してください [サーバ側の暗号化を行うための要求ヘッダー]

サポートされない要求ヘッダーです

次の要求ヘッダーはサポートされていません。

- 。 `x-amz-acl` 要求ヘッダーはサポートされていません。
- 。 `x-amz-website-redirect-location` 要求ヘッダーはサポートされておらず、返されます `XNotImplemented`。

ストレージクラスのオプション

。 `x-amz-storage-class` 要求ヘッダーがサポートされています。に送信された値 `x-amz-storage-class` StorageGRID が取り込み中にオブジェクトデータを保護する方法に影響し、StorageGRID システム（ILMで決定）に格納されるオブジェクトの永続的コピーの数には影響しません。

取り込まれたオブジェクトに一致するILMルールの取り込み動作がStrictオプションに指定されている場合、はを使用します `x-amz-storage-class` ヘッダーに影響はありません。

には次の値を使用できます `x-amz-storage-class`：

- `STANDARD` （デフォルト）
  - * Dual commit * ： ILM ルールの取り込み動作が Dual commit オプションに指定されている場合は、オブジェクトの取り込み直後にオブジェクトの 2 つ目のコピーが作成されて別のストレージノードに配置されます（デュアルコミット）。ILMが評価されると、StorageGRID はこれらの初期中間コピーがルールの配置手順を満たしているかどうかを判断します。作成されていない場合は、新しいオブジェクトコピーを別の場所に作成し、最初の中間コピーを削除しなければならないことがあります。
  - * Balanced *：ILMルールでBalancedオプションが指定されていて、ルールで指定されたすべてのコピーをStorageGRID がすぐに作成できない場合、StorageGRID は2つの中間コピーを別 々 のストレージノードに作成します。

    StorageGRID がILMルールに指定されたすべてのオブジェクトコピーをただちに作成できる場合（同期配置）は、を参照してください `x-amz-storage-class` ヘッダーに影響はありません。

- `REDUCED_REDUNDANCY`
  - * Dual commit * ： ILM ルールの取り込み動作が Dual commit オプションに指定されている場合は、オブジェクトの取り込み時に StorageGRID が中間コピーを 1 つ作成します（シングルコミット）。
  - * Balanced *：ILMルールでBalancedオプションが指定されている場合、StorageGRID は、ルールで指定されたすべてのコピーをただちに作成できない場合にのみ中間コピーを1つ作成します。StorageGRID で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ルールで単一のレプリケートコピーが作成される場合に最適です。この場合は、を使用します `REDUCED_REDUNDANCY` 取り込み処理のたびに追加のオブジェクトコピーを不要に作成および削除する必要がなくなります。

  を使用する `REDUCED_REDUNDANCY` それ以外の場合は、このオプションは推奨されません。 `REDUCED_REDUNDANCY` 取り込み中にオブジェクトデータが失われるリスクが高まります。たとえば、ILM 評価の前にコピーが 1 つだけ格納されていたストレージノードに障害が発生すると、データが失われる可能性があります。

> レプリケートコピーを一定期間に 1 つだけ作成すると、データが永続的に失われるリスクがあります。オブジェクトのレプリケートコピーが 1 つしかない場合、ストレージノードに障害が発生したり、重大なエラーが発生すると、そのオブジェクトは失われます。また、アップグレードなどのメンテナンス作業中は、オブジェクトへのアクセスが一時的に失われます。

を指定します `REDUCED_REDUNDANCY` オブジェクトの初回取り込み時に作成されるコピー数のみに影響します。オブジェクトがアクティブな ILM ポリシーで評価される際に作成されるオブジェクトのコピー数には影響せず、 StorageGRID システムでデータが格納されるときの冗長性レベルが低下することもありません。

> S3オブジェクトロックを有効にしてオブジェクトをバケットに取り込む場合は、を使用します `REDUCED_REDUNDANCY` オプションは無視されます。古い準拠バケットにオブジェクトを取り込む場合は、を参照してください `REDUCED_REDUNDANCY` オプションを指定するとエラーが返されます。StorageGRID では、常にデュアルコミットの取り込みが実行され、コンプライアンス要件が満たされます。

サーバ側の暗号化を行うための要求ヘッダー

オブジェクトをサーバ側の暗号化で暗号化するには、次の要求ヘッダーを使用します。SSE オプションと SSE-C オプションを同時に指定することはできません。

- * SSE * ： StorageGRID で管理される一意のキーでオブジェクトを暗号化するには、次のヘッダーを使用します。
  - `x-amz-server-side-encryption`
- * SSE-C * ：ユーザが指定および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 3 つのヘッダーをすべて使用します。
  - `x-amz-server-side-encryption-customer-algorithm`:指定します `AES256`。
  - `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ます。
  - `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。

> 指定した暗号化キーが格納されることはありません。暗号化キーを紛失すると、対応するオブジェクトが失われます。ユーザ指定のキーを使用してオブジェクトデータを保護する前に、の考慮事項を確認してください "サーバ側の暗号化を使用する"。

> SSE または SSE-C で暗号化されたオブジェクトは、バケットレベルまたはグリッドレベルの暗号化設定が無視されます。

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は、一意です `versionId` は、格納されてい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。これ `versionId` は、を使用して応答としても返されます `x-amz-version-id` 応答ヘッダー。

バージョン管理が一時停止中の場合は、オブジェクトバージョンはnullで格納されます `versionId` また、null バージョンがすでに存在する場合は上書きされます。

Authorizationヘッダーのシグニチャ計算

を使用する場合 `Authorization` 要求を認証するためのヘッダー。StorageGRID はAWSと次の点で異なります。

- StorageGRID は必要ありません `host` に含めるヘッダー `CanonicalHeaders`。
- StorageGRID は必要ありません `Content-Type` に含まれています `CanonicalHeaders`。
- StorageGRID は必要ありません `x-amz-*` に含めるヘッダー `CanonicalHeaders`。

> 一般的なベストプラクティスとして、には常にこれらのヘッダーを含めてください `CanonicalHeaders` これらのヘッダーが検証されるようにするためですが、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、を参照してください "Authorizationヘッダーのシグニチャ計算：単一チャンクでのペイロードの転送（AWS Signature Version 4）"。

関連情報

"ILM を使用してオブジェクトを管理する"

"バケットの処理"

"監査ログで追跡される S3 処理"

"クライアント接続の設定方法"

PUT Object - Copy の各コマンドを実行します

S3 PUT Object - Copy 要求を使用すると、すでに S3 に格納されているオブジェクトのコピーを作成できます。PUT Object - Copy 処理は、 GET を実行してから PUT を実行する処理と同じです。

競合を解決します

同じキーに書き込む 2 つのクライアントなど、競合するクライアント要求は、「 latest-wins 」ベースで解決されます。「 latest-wins 」評価は、 S3 クライアントが処理を開始するタイミングではなく、 StorageGRID システムが特定の要求を完了したタイミングで行われます。

オブジェクトのサイズ

単一 PUT Object 処理の maximum_recommended_size は 5GiB （ 5 、 368 、 709 、 120 バイト）です。5GB より大きいオブジェクトがある場合は、マルチパートアップロードを使用してください。

単一のPUT Object処理のmaximum_supported_sizeは5TiB（5、497、558、138、880バイト）です。ただし、5GiB 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。

ユーザメタデータ内の **UTF-8** 文字

要求のユーザ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 文字が含まれている場合、 StorageGRID の動作は定義されていません。

ユーザ定義メタデータのキー名または値に含まれているエスケープされた UTF-8 文字は、 StorageGRID で解析も解釈もされません。エスケープされた UTF-8 文字は ASCII 文字として扱われます。

- ユーザ定義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、要求は正常に実行されます。
- StorageGRID からが返されない `x-amz-missing-meta` キーの名前または値の解釈後の値に印刷不能文字が含まれている場合は、ヘッダー。

サポートされる要求ヘッダー

次の要求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-`をクリックし、続けてユーザ定義のメタデータを含む名前と値のペアを作成します
- `x-amz-metadata-directive`:デフォルト値はです `COPY`をクリックすると、オブジェクトおよび関連するメタデータをコピーできます。

  を指定できます `REPLACE` オブジェクトのコピー時に既存のメタデータを上書きする場合、またはオブジェクトメタデータを更新する場合。

- `x-amz-storage-class`
- `x-amz-tagging-directive`:デフォルト値はです `COPY`をクリックすると、オブジェクトとすべてのタグをコピーできます。

  を指定できます `REPLACE` オブジェクトのコピー時に既存のタグを上書きする場合、またはタグを更新する場合。

- S3 オブジェクトロック要求のヘッダー：
  - `x-amz-object-lock-mode`
  - `x-amz-object-lock-retain-until-date`
  - `x-amz-object-lock-legal-hold`

    これらのヘッダーを指定せずに要求を行うと、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モードとretain-until-dateが計算されます。を参照してください "S3 REST APIを使用してS3オブジェクトロックを設定します"。

- SSE 要求ヘッダー：
  - `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`

    を参照してください [サーバ側の暗号化を行うための要求ヘッダー]

サポートされない要求ヘッダーです

次の要求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`

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

ストレージクラスのオプション

。 `x-amz-storage-class` 要求ヘッダーがサポートされ、一致するILMルールで取り込み動作にDual commitまたはBalancedが指定されている場合にStorageGRID で作成され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 Dual commit オプションが使用されている場合、または Balanced オプションによって中間コピーが作成される場合に、デュアルコミットの取り込み処理を指定します。

- `REDUCED_REDUNDANCY`

  ILM ルールで Dual commit オプションが使用されている場合、または Balanced オプションによって中間コピーが作成される場合に、シングルコミットの取り込み処理を指定します。

> S3オブジェクトロックを有効にしてオブジェクトをバケットに取り込む場合は、を使用します `REDUCED_REDUNDANCY` オプションは無視されます。古い準拠バケットにオブジェクトを取り込む場合は、を参照してください `REDUCED_REDUNDANCY` オプションを指定するとエラーが返されます。StorageGRID では、常にデュアルコミットの取り込みが実行され、コンプライアンス要件が満たされます。

PUT Object - Copy で **x-amz-copy-source** を使用しています

ソースのバケットとキーの場合は、で指定します `x-amz-copy-source` ヘッダーはデスティネーションのバケットおよびキーとは異なり、ソースオブジェクトデータのコピーがデスティネーションに書き込まれます。

送信元と宛先が一致している場合は、および `x-amz-metadata-directive` ヘッダーはのように指定します `REPLACE`では、要求で指定されたメタデータの値に基づいてオブジェクトのメタデータが更新されます。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。

- PUT Object - Copyを使用して既存のオブジェクトを暗号化したり、既存のオブジェクトの暗号化を変更したりすることはできません。を用意する場合は `x-amz-server-side-encryption` ヘッダーまたは `x-amz-server-side-encryption-customer-algorithm` ヘッダー。StorageGRID は要求を拒否し、戻ります `XNotImplemented`。
- 一致する ILM ルールで指定されている取り込み動作のオプションが使用されません。更新によって発生したオブジェクト配置の変更は、通常のバックグラウンド ILM プロセスで ILM が再評価されるときに実施されます。

  つまり、ILMルールの取り込み動作にStrictオプションが使用されている場合、必要なオブジェクト配置を実行できない場合（新たに必要な場所が使用できない場合など）は処理されません。更新されたオブジェクトは、必要な配置を実行可能になるまで現在の配置が維持されます。

サーバ側の暗号化を行うための要求ヘッダー

サーバ側の暗号化を使用する場合は、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かどうかによって、指定する要求ヘッダーが異なります。

- ソースオブジェクトがユーザ指定のキーを使用して暗号化されている場合（ SSE-C ）は、オブジェクト

を復号化してコピーできるように、 PUT Object - Copy 要求に次の 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`:指定します `AES256`。
- `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の作成時に指定したMD5ダイジェストを指定します。

- ユーザが指定および管理する一意のキーでターゲットオブジェクト（コピー）を暗号化する場合は、次の 3 つのヘッダーを含めます。
  - `x-amz-server-side-encryption-customer-algorithm`:指定します `AES256`。
  - `x-amz-server-side-encryption-customer-key`:ターゲットオブジェクトの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。

> ⓘ 指定した暗号化キーが格納されることはありません。暗号化キーを紛失すると、対応するオブジェクトが失われます。ユーザ指定のキーを使用してオブジェクトデータを保護する前に、の考慮事項を確認してください "サーバ側の暗号化を使用する"。

- StorageGRID で管理される一意のキーでターゲットオブジェクト（コピー）を暗号化する（ SSE ）には、 PUT Object - Copy 要求に次のヘッダーを含めます。
  - `x-amz-server-side-encryption`

> ⓘ 。 `server-side-encryption` オブジェクトの値を更新できません。代わりに、新しいを使用してコピーを作成します `server-side-encryption` を使用した値 `x-amz-metadata-directive`： `REPLACE`。

バージョン管理

ソースバケットがバージョン管理に対応している場合は、を使用できます `x-amz-copy-source` オブジェクトの最新バージョンをコピーするヘッダー。オブジェクトの特定のバージョンをコピーするには、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 `versionId` サブリソース：デスティネーションバケットがバージョン管理に対応している場合は、で生成されたバージョンが返されます `x-amz-version-id` 応答ヘッダー。ターゲットバケットのバージョン管理が一時停止中の場合は、を実行します `x-amz-version-id` 「null」値を返します。

関連情報

"ILM を使用してオブジェクトを管理する"

"監査ログで追跡される S3 処理"

"PUT Object の場合"

SelectObjectContent の順に選択します

S3 SelectObjectContent 要求を使用すると、シンプルな SQL ステートメントに基づいて

S3 オブジェクトのコンテンツをフィルタリングできます。

詳細については、を参照してください "SelectObjectContent に関する AWS ドキュメント"。

作業を開始する前に

- テナントアカウントには S3 Select 権限が割り当てられます。
- これで完了です `s3:GetObject` 照会するオブジェクトの権限。
- 照会するオブジェクトは、次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やbzip2のアーカイブに圧縮して使用することもできます。
  - 寄木細工。寄木細工オブジェクトの追加要件：
    - S3 Selectでは、GZIPまたはSnappyを使用したカラムナ圧縮のみがサポートされます。S3 Selectでは、寄木細工オブジェクトのオブジェクト全体の圧縮はサポートされません。
    - S3 Selectは寄木細工の出力をサポートしていません。出力形式はCSVまたはJSONで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512MB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論理型は使用できません。
- SQL 式の最大長は 256KB です。
- 入力または結果のすべてのレコードの最大長は 1MiB です。

ScanRangeの使用はサポートされていません。

CSV要求の構文例

```
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>
```

寄木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、州名、 2010 年人口、 2015 年推定人口、米国の人口調査データからの変化率を取得します。状態でないファイル内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
```

照会するファイルの最初の数行 `SUB-EST2020_ALL.csv`次のようになります。

```
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`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（寄木細工）

```
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
```

出力ファイルの最初のいくつかの行は、.csvを変更します。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,5029196,5454328,8.4532796097030221132761578590295546246
```

マルチパートアップロードの処理

このセクションでは、 StorageGRID でのマルチパートアップロードの処理のサポートについて説明します。

マルチパートアップロードのすべての処理に、次の条件と注意事項が適用されます。

- 1 つのバケットに対して同時に実行するマルチパートアップロードが 1 、 000 件を超えないようにしてください。 1 、 000 件を超えると、そのバケットに対する List Multipart Uploads のクエリで完全な結果が返されないことがあります。
- StorageGRID は、マルチパートに AWS のサイズ制限を適用します。S3 クライアントは次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のサイズは 5MiB （ 5 、 242 、 880 バイト）と 5GiB （ 5 、 368 、 709 、 120 バイト）の間にする必要があります。
  - 最後の部分は 5MiB （ 5,242,880 バイト）より小さくできます。
  - 一般に、パーツ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、 100GiB オブジェクトの場合、 5GB 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。
  - 5GB 未満のオブジェクトでは、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ルールの取り込み動作がBalancedまたはStrictの場合は、マルチパートオブジェクトの各パートについて、ILMルールの取り込み動作がBalancedまたはStrictの場合はマルチパートアップロードの完了時にオブジェクト全体についてILMが評価されます。これがオブジェクトとパートの配置にどのように影響するかに注意する必要があります。
  - S3 マルチパートアップロードの進行中に ILM が変更されると、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のパートが現在の ILM 要件を満たしていないことがあります。正しく配

置されていないパートは ILM ルールによる再評価の対象としてキューに登録され、あとで正しい場所に移動されます。

- パートに対して 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トのサイズでフィルタリングします。つまり、オブジェクト全体のILM要件を満たしていない場所にオブジェクトの一部を格納できます。たとえば、 10GB 以上のオブジェクトをすべて DC1 に格納し、それより小さいオブジェクトをすべて DC2 に格納するルールの場合、 10 パートからなるマルチパートアップロードの 1GB の各パートは取り込み時に DC2 に格納されます。オブジェクト全体に対して ILM が評価されると、オブジェクトのすべてのパートが DC1 に移動されます。

- マルチパートアップロードでは、すべての処理で StorageGRID の整合性制御がサポートされます。
- マルチパートアップロードでは、必要に応じてサーバ側の暗号化を使用できます。SSE（StorageGRIDで管理されるキーによるサーバ側の暗号化）を使用するには、を指定します `x-amz-server-side-encryption` Initiate Multipart Upload要求のみの要求ヘッダー。SSE-C （ユーザ指定のキーによるサーバ側の暗号化）を使用する場合は、 Initiate Multipart Upload 要求と後続の各 Upload Part 要求に、同じ 3 つの暗号化キー要求ヘッダーを指定します。

操作	実装
マルチパートアップロードをリストします	を参照してください "マルチパートアップロードをリストします"
マルチパートアップロードを開始します	を参照してください "マルチパートアップロードを開始します"
パーツをアップロードします	を参照してください "パーツをアップロードします"
パーツのアップロード - コピー	を参照してください "パーツのアップロード - コピー"
Complete Multipart Upload の実行	を参照してください "Complete Multipart Upload の実行"
マルチパートアップロードを中止します	Amazon S3 REST API のすべての動作が実装されています。予告なく変更される場合があります。
パーツをリストします	Amazon S3 REST API のすべての動作が実装されています。予告なく変更される場合があります。

関連情報

- "整合性制御"
- "サーバ側の暗号化を使用します"

マルチパートアップロードをリストします

List Multipart Uploads 処理では、バケットの進行中のマルチパートアップロードがリストされます。

次の要求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、アップロードのリストの表示、パートのアップロード、アップロードしたパートのアセンブル、およびアップロードの完了の個別の処理に分けられます。Complete Multipart Upload 処理が実行されると、オブジェクトが作成される時点（およびバージョン管理されている場合）になります。

マルチパートアップロードを開始します

Initiate Multipart Upload（CreateMultipartUpload）処理を実行すると、オブジェクトのマルチパートアップロードが開始され、アップロードIDが返されます。

。 x-amz-storage-class 要求ヘッダーがサポートされています。に送信された値 x-amz-storage-class StorageGRID が取り込み中にオブジェクトデータを保護する方法に影響し、StorageGRID システム（ILMで決定）に格納されるオブジェクトの永続的コピーの数には影響しません。

取り込まれたオブジェクトに一致するILMルールの取り込み動作がStrictオプションに指定されている場合、はを使用します x-amz-storage-class ヘッダーに影響はありません。

には次の値を使用できます x-amz-storage-class：

- STANDARD （デフォルト）
  - * Dual commit * ： ILM ルールの取り込み動作が Dual commit オプションに指定されている場合は、オブジェクトの取り込み直後にオブジェクトの 2 つ目のコピーが作成されて別のストレージノードに配置されます（デュアルコミット）。ILMが評価されると、StorageGRID はこれらの初期中間コピーがルールの配置手順を満たしているかどうかを判断します。作成されていない場合は、新しいオブジェクトコピーを別の場所に作成し、最初の中間コピーを削除しなければならないことがあります。
  - * Balanced *：ILMルールでBalancedオプションが指定されていて、ルールで指定されたすべてのコピーをStorageGRID がすぐに作成できない場合、StorageGRID は2つの中間コピーを別 々 のストレージノードに作成します。

    StorageGRID がILMルールに指定されたすべてのオブジェクトコピーをただちに作成できる場合（同期配置）は、を参照してください x-amz-storage-class ヘッダーに影響はありません。
- REDUCED_REDUNDANCY
  - * Dual commit * ： ILM ルールの取り込み動作が Dual commit オプションに指定されている場合は、オブジェクトの取り込み時に StorageGRID が中間コピーを 1 つ作成します（シングルコミット）。
  - * Balanced *：ILMルールでBalancedオプションが指定されている場合、StorageGRID は、ルールで指

定されたすべてのコピーをただちに作成できない場合にのみ中間コピーを1つ作成します。StorageGRID で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ルールで単一のレプリケートコピーが作成される場合に最適です。この場合は、を使用します `REDUCED_REDUNDANCY` 取り込み処理のたびに追加のオブジェクトコピーを不要に作成および削除する必要がなくなります。

を使用する `REDUCED_REDUNDANCY` それ以外の場合は、このオプションは推奨されません。 `REDUCED_REDUNDANCY` 取り込み中にオブジェクトデータが失われるリスクが高まります。たとえば、ILM 評価の前にコピーが 1 つだけ格納されていたストレージノードに障害が発生すると、データが失われる可能性があります。

> ⓘ レプリケートコピーを一定期間に 1 つだけ作成すると、データが永続的に失われるリスクがあります。オブジェクトのレプリケートコピーが 1 つしかない場合、ストレージノードに障害が発生したり、重大なエラーが発生すると、そのオブジェクトは失われます。また、アップグレードなどのメンテナンス作業中は、オブジェクトへのアクセスが一時的に失われます。

を指定します `REDUCED_REDUNDANCY` オブジェクトの初回取り込み時に作成されるコピー数のみに影響します。オブジェクトがアクティブな ILM ポリシーで評価される際に作成されるオブジェクトのコピー数には影響せず、 StorageGRID システムでデータが格納されるときの冗長性レベルが低下することもありません。

> ⓘ S3オブジェクトロックを有効にしてオブジェクトをバケットに取り込む場合は、を使用します `REDUCED_REDUNDANCY` オプションは無視されます。古い準拠バケットにオブジェクトを取り込む場合は、を参照してください `REDUCED_REDUNDANCY` オプションを指定するとエラーが返されます。StorageGRID では、常にデュアルコミットの取り込みが実行され、コンプライアンス要件が満たされます。

次の要求ヘッダーがサポートされています。

- `Content-Type`
- `x-amz-meta-`をクリックし、続けてユーザ定義のメタデータを含む名前と値のペアを作成します

  ユーザ定義メタデータの名前と値のペアを指定する場合、一般的な形式は次のとおりです。

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

  ILMルールの参照時間に*[ユーザ定義の作成時間]*オプションを使用する場合は、を使用する必要があります `creation-time` を、オブジェクトの作成時に記録されたメタデータの名前として指定します。例：

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

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

  > ⓘ 追加中です `creation-time` レガシー準拠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、ユーザ定義メタデータは許可されません。エラーが返されます。

- S3 オブジェクトロック要求のヘッダー：

- `x-amz-object-lock-mode`
- `x-amz-object-lock-retain-until-date`
- `x-amz-object-lock-legal-hold`

  これらのヘッダーがない状態で要求を送信した場合、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 retain-date 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します"

- SSE 要求ヘッダー：
  - `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`

  [サーバ側の暗号化を行うための要求ヘッダー]

> StorageGRID でのUTF-8文字の処理方法については、PUT Objectのドキュメントを参照してください。

サーバ側の暗号化を行うための要求ヘッダー

マルチパートオブジェクトをサーバ側の暗号化で暗号化するには、次の要求ヘッダーを使用します。SSE オプションと SSE-C オプションを同時に指定することはできません。

- * SSE * ： StorageGRID で管理される一意のキーでオブジェクトを暗号化する場合は、 Initiate Multipart Upload 要求で次のヘッダーを使用します。Upload Partリクエストでは、このヘッダーを指定しないでください。
  - `x-amz-server-side-encryption`
- * SSE-C * ：ユーザが指定および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、 Initiate Multipart Upload 要求（および後続の各 Upload Part 要求）で、次の 3 つのヘッダーをすべて使用します。
  - `x-amz-server-side-encryption-customer-algorithm`:指定します `AES256`。
  - `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ます。
  - `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。

> 指定した暗号化キーが格納されることはありません。暗号化キーを紛失すると、対応するオブジェクトが失われます。ユーザ指定のキーを使用してオブジェクトデータを保護する前に、の考慮事項を確認してください "サーバ側の暗号化を使用する"。

サポートされない要求ヘッダーです

次の要求ヘッダーはサポートされていません `XNotImplemented`

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

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、アップロードのリストの表示、パートのアップロード、アップロードしたパートのアセンブル、およびアップロードの完了の個別の処理に分けられます。Complete Multipart Upload 処理が実行されると、オブジェクトが作成されます（該当する場合はバージョン管理されます）。

関連情報

"ILM を使用してオブジェクトを管理する"

"PUT Object の場合"

パーツをアップロードします

Upload Part 処理では、オブジェクトのマルチパートアップロード内のパートがアップロードされます。

サポートされる要求ヘッダー

次の要求ヘッダーがサポートされています。

- `Content-Length`
- `Content-MD5`

サーバ側の暗号化を行うための要求ヘッダー

Initiate Multipart Upload 要求に SSE-C 暗号化を指定した場合は、各 Upload Part 要求に次の要求ヘッダーも含める必要があります。

- `x-amz-server-side-encryption-customer-algorithm`:指定します `AES256`。
- `x-amz-server-side-encryption-customer-key`：Initiate Multipart Upload要求で指定した暗号化キーを指定します。
- `x-amz-server-side-encryption-customer-key-MD5`：Initiate Multipart Upload要求で指定したMD5ダイジェストを指定します。

> 指定した暗号化キーが格納されることはありません。暗号化キーを紛失すると、対応するオブジェクトが失われます。お客様提供の鍵を使用してオブジェクト・データを保護する前に ' サーバ側の暗号化を使用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、アップロードのリストの表示、パートのアップロード、アップロードしたパートのアセンブル、およびアップロードの完了の個別の処理に分けられます。Complete Multipart Upload 処理が実行されると、オブジェクトが作成されます（該当する場合はバージョン管理されます）。

関連情報

"サーバ側の暗号化を使用します"

パーツのアップロード - コピー

Upload Part - Copy 処理は、データソースとしての既存のオブジェクトからデータをコピーすることで、オブジェクトのパートをアップロードします。

Upload Part - Copy 処理には、すべての Amazon S3 REST API の動作が実装されています。予告なく変更される場合があります。

この要求は、で指定されたオブジェクトデータの読み取りと書き込みを行います `x-amz-copy-source-range` StorageGRID システム内で実行する。

次の要求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`

サーバ側の暗号化を行うための要求ヘッダー

Initiate Multipart Upload 要求に SSE-C 暗号化を指定した場合は、各 Upload Part - Copy 要求に次の要求ヘッダーも含める必要があります。

- `x-amz-server-side-encryption-customer-algorithm`:指定します `AES256`。
- `x-amz-server-side-encryption-customer-key`：Initiate Multipart Upload要求で指定した暗号化キーを指定します。
- `x-amz-server-side-encryption-customer-key-MD5`：Initiate Multipart Upload要求で指定したMD5ダイジェストを指定します。

ソースオブジェクトがユーザ指定のキーを使用して暗号化されている場合（ SSE-C ）は、オブジェクトを復号化してコピーできるように、 Upload Part - Copy 要求に次の 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`:指定します `AES256`。
- `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の作成時に指定したMD5ダイジェストを指定します。

> 指定した暗号化キーが格納されることはありません。暗号化キーを紛失すると、対応するオブジェクトが失われます。お客様提供の鍵を使用してオブジェクト・データを保護する前に ' サーバ側の暗号化を使用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、アップロードのリストの表示、パートのアップロード、アップロードしたパートのアセンブル、およびアップロードの完了の個別の処理に分けられます。Complete

Multipart Upload 処理が実行されると、オブジェクトが作成されます（該当する場合はバージョン管理されます）。

Complete Multipart Upload の実行

Complete Multipart Upload 処理では、以前にアップロードされたパートをアセンブルすることで、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。

競合を解決します

同じキーに書き込む 2 つのクライアントなど、競合するクライアント要求は、「 latest-wins 」ベースで解決されます。「 latest-wins 」評価は、 S3 クライアントが処理を開始するタイミングではなく、 StorageGRID システムが特定の要求を完了したタイミングで行われます。

要求ヘッダー

。 `x-amz-storage-class` 要求ヘッダーがサポートされ、一致するILMルールで取り込み動作にDual commitまたはBalancedが指定されている場合にStorageGRID で作成され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 Dual commit オプションが使用されている場合、または Balanced オプションによって中間コピーが作成される場合に、デュアルコミットの取り込み処理を指定します。

- `REDUCED_REDUNDANCY`

  ILM ルールで Dual commit オプションが使用されている場合、または Balanced オプションによって中間コピーが作成される場合に、シングルコミットの取り込み処理を指定します。

  S3オブジェクトロックを有効にしてオブジェクトをバケットに取り込む場合は、を使用します `REDUCED_REDUNDANCY` オプションは無視されます。古い準拠バケットにオブジェクトを取り込む場合は、を参照してください `REDUCED_REDUNDANCY` オプションを指定するとエラーが返されます。StorageGRID では、常にデュアルコミットの取り込みが実行され、コンプライアンス要件が満たされます。

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

。 `ETag` 返される値はデータのMD5サムではなく、のAmazon S3 APIの実装に従います `ETag` マルチパートオブジェクトの値。

バージョン管理

マルチパートアップロードは、この処理で完了します。バケットでバージョン管理が有効になっている場合は、マルチパートアップロードの完了後にオブジェクトの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は、一意です `versionId` は、格納されてい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。これ `versionId` は、を使用して応答としても返されます `x-amz-version-id` 応答ヘッダー。

バージョン管理が一時停止中の場合は、オブジェクトバージョンはnullで格納されます `versionId` また、null バージョンがすで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ときは、同じオブジェクトキーで同時に複数のマルチパートアップロードが実行されている場合でも、マルチパートアップロードが完了するたびに常に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ときは、マルチパートアップロードの開始後に、同じオブジェクトキーで別のマルチパートアップロードが開始されて先に完了することがあります。バージョン管理が有効になっ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に失敗しました

マルチパートアップロードが行われるバケットでプラットフォームサービスが設定されている場合、関連するレプリケーション操作や通知操作が失敗してもマルチパートアップロードは正常に実行されます。

この状況が発生すると、 Total Events （ SMTT ）のアラームがグリッドマネージャで生成されます。Last Event メッセージに、通知が失敗した最後のオブジェクトについて、「 Failed to publish notifications for bucket-name object key 」と表示されます。（このメッセージを表示するには、 * nodes * > * _ Storage Node_* > * Events * を選択します。表の一番上にLast Eventが表示されます）。イベントメッセージは、にも表示されます `/var/local/log/bycast-err.log`。

テナントでは、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、失敗したレプリケーションまたは通知をトリガーできます。テナントでは、既存の値を再送信し、不要な変更を回避できます。

関連情報

"ILM を使用してオブジェクトを管理する"

エラー応答

StorageGRID システムでは、該当する S3 REST API の標準のエラー応答をすべてサポートしています。また、 StorageGRID の実装では、カスタム応答もいくつか追加されています。

サポートされている **S3 API** のエラーコード

名前	**HTTP** ステータス
アクセスが拒否されました	403 禁止
BadDigest の略	400 不正な要求です
BucketAlreadyExists のようになりました	409 競合
BucketNotEmpty のように入力します	409 競合
IncompleteBody	400 不正な要求です
内部エラー	500 Internal Server Error （内部サーバエラー）

名前	**HTTP** ステータス
InvalidAccessKeyId	403 禁止
アンヴァリッドドキュメント	400 不正な要求です
InvalidBucketName の略	400 不正な要求です
InvalidBucketState の場合	409 競合
InvalidDigest の略	400 不正な要求です
InvalidEncryptionAlgorithmError	400 不正な要求です
InvalidPart	400 不正な要求です
InvalidPartOrder	400 不正な要求です
InvalidRange ：無効な範囲	416 リクエストされた範囲が適合しません
InvalidRequest	400 不正な要求です
InvalidStorageClass	400 不正な要求です
InvalidTag	400 不正な要求です
InvalidURI	400 不正な要求です
KeyTooLong の 2 つのグループがあります	400 不正な要求です
MalformedXML の場合	400 不正な要求です
MetadataTooLarge	400 不正な要求です
MethodNotAllowed のように入力します	405 メソッドは許可されていません
MissingContentLength （ MissingContentLength ）	411 長さが必要です
MissingRequestBodyError	400 不正な要求です
MissingSecurityHeader	400 不正な要求です
NoSuchBucket	404 が見つかりません

名前	**HTTP** ステータス
NoSuchKey	404 が見つかりません
NoSuchUpload	404 が見つかりません
実装なし	501 は実装されていません
NoSuchBucketPolicy のようになります	404 が見つかりません
ObjectLockConfigurationNotFoundError	404 が見つかりません
PreconditionalFailed	412 事前条件が失敗しました
RequestTimeTooSkewed	403 禁止
サービスを利用できません	503 Service Unavailable （ 503 サービスが利用でき
SignatureDoesNotMatch のように指定します	403 禁止
TooManyBuckets	400 不正な要求です
UserKeyMustBeSpecified	400 不正な要求です

StorageGRID カスタムのエラーコード

名前	説明	**HTTP** ステータス
XBucketLifecycleNotAllowed のようになりました	バケットライフサイクル設定は従来の準拠バケットには適用されません	400 不正な要求です
XBucketPolicyParseException	受信したバケットポリシー JSON を解析できませんでした。	400 不正な要求です
XCompliConflict	準拠設定が古いため、処理が拒否されました。	403 禁止
XCompliReducedRedundancyForbidden	レガシー準拠バケットでは冗長性の低下は許可されません	400 不正な要求です
XMaxBucketPolicyLengthExceeded （ XMaxBucketLengthExceeded ）	ポリシーが許容される最大バケットポリシー長を超えています。	400 不正な要求です
XMissingInternalRequestHeader	内部要求のヘッダーがありません。	400 不正な要求です

名前	説明	HTTP ステータス
XNoSuchBucketCompliance です	指定したバケットで従来の準拠が有効になっていません。	404 が見つかりません
XNotAcceptable	要求に含まれている Accept ヘッダーの 1 つ以上を満たすことができませんでした。	406 は許容されません
XNotImplemented	指定した要求の処理には、実装されていない機能が含まれます。	501 は実装されていません

StorageGRID S3要求

GET Bucket consistency

GET Bucket consistency 要求を使用すると、特定のバケットに適用されている整合性レベルを確認できます。

新たに作成したオブジェクトに対しては、リードアフターライト整合性を保証するようにデフォルトの整合性制御が設定されます。

この処理を完了するには、 s3 ： GetBucketConsistency 権限または root アカウントが必要です。

要求例

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

応答

応答XMLで、 `<Consistency>` は次のいずれかの値を返します。

整合性制御	説明
すべて	すべてのノードが即座にデータを受け取り、受け取れない場合は要求が失敗します。
strong-global	すべてのサイトのすべてのクライアント要求について、リードアフターライト整合性が保証されます。
strong-site	1 つのサイト内のすべてのクライアント要求について、リードアフターライト整合性が保証されます。

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

応答例

```
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

"整合性制御"

PUT Bucket consistency

PUT Bucket consistency 要求を使用すると、バケットで実行される処理に適用する整合性レベルを指定できます。

新たに作成したオブジェクトに対しては、リードアフターライト整合性を保証するようにデフォルトの整合性制御が設定されます。

作業を開始する前に

この処理を完了するには、 s3 ： PutBucketConsistency 権限または root アカウントが必要です。

リクエスト

。 `x-ntap-sg-consistency` パラメータには次のいずれかの値を指定する必要があります。

整合性制御	説明
すべて	すべてのノードが即座にデータを受け取り、受け取れない場合は要求が失敗します。
strong-global	すべてのサイトのすべてのクライアント要求について、リードアフターライト整合性が保証されます。
strong-site	1 つのサイト内のすべてのクライアント要求について、リードアフターライト整合性が保証されます。
read-after-new-write の場合	（デフォルト）新規オブジェクトにはリードアフターライト整合性を、オブジェクトの更新には結果整合性を提供します。高可用性が確保され、データ保護が保証されます。ほとんどの場合に推奨されます。
利用可能	新規オブジェクトとオブジェクトの更新の両方について結果整合性を提供します。S3バケットの場合は、必要な場合にのみ使用します（読み取り頻度の低いログ値を含むバケットや、存在しないキーに対するHEAD処理やGET処理など）。S3 FabricPool バケットではサポートされません。

- 注： * 一般的には、「 read-after-new-write 」整合性制御値を使用する必要があります。要求が正しく動作しない場合は、可能であればアプリケーションクライアントの動作を変更します。または、 API 要求ごとに整合性制御を指定するようにクライアントを設定します。バケットレベルの整合性制御は最後の手段と考えてください。

要求例

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

関連情報

"整合性制御"

GET Bucket last access time の場合

GET Bucket last access time 要求を使用すると、最終アクセス時間の更新が個々のバケットで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。

この処理を完了するには、 s3 ： GetBucketLastAccessTime 権限または root アカウントが必要です。

要求例

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

応答例

次の例では、バケットの最終アクセス時間の更新が有効になっています。

```
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>
```

PUT Bucket last access time のように指定します

PUT Bucket last access time 要求を使用すると、最終アクセス時間の更新を個々のバケットで有効または無効にできます。最終アクセス時間の更新を無効にするとパフォーマンスが向上します。バージョン 10.3.0 以降で作成されたバケットに対しては、いずれもデフォルトで無効になります。

この処理を完了するには、バケットの s3 ： PutBucketLastAccessTime 権限または root アカウントが必要です。

> StorageGRID バージョン 10.3 以降では、すべての新規バケットで最終アクセス時間の更新がデフォルトで無効になります。以前のバージョンの StorageGRID で作成されたバケットにこの新たなデフォルトの動作を適用する場合は、対象となるバケットごとに最終アクセス時間の更新を無効にする必要があります。最終アクセス時間の更新を有効または無効にするには、Tenant Managerの* S3 > Buckets > Change Last Access Setting*チェックボックス、またはテナント管理APIを使用します。

バケットで最終アクセス時間の更新が無効になっている場合、バケットの処理の動作は次のようになります。

- GET Object、GET Object ACL、GET Object Tagging、HEAD Objectの各要求では、最終アクセス時間が更新されません。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。
- メタデータのみを更新する PUT Object - Copy 要求と PUT Object Tagging 要求では、最終アクセス時間も更新されます。オブジェクトは ILM 評価のキューに追加されます。
- ソースバケットで最終アクセス時間の更新が無効になっている場合は、PUT Object - Copy要求でソース

バケットの最終アクセス時間が更新されません。コピーされたオブジェクトは、ソースバケットの ILM 評価のキューに追加されません。ただし、デスティネーションについては、 PUT Object - Copy 要求で常に最終アクセス時間が更新されます。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。

- Complete Multipart Upload 要求では、最終アクセス時間が更新されます。完了したオブジェクトは、 ILM 評価のキューに追加されます。

例をリクエストする

この例では、バケットの最終アクセス時間を有効にしています。

```
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間を無効にしています。

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

関連情報

"テナントアカウントを使用する"

バケットのメタデータ通知設定を削除します

DELETE Bucket metadata notification configuration 要求では、設定 XML を削除することで、個々のバケットで検索統合サービスを無効化できます。

この処理を完了するには、バケットの s3 ： DeleteBucketMetadataNotification 権限または root アカウントが必要です。

要求例

次の例は、バケットの検索統合サービスを無効にする方法を示しています。

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

GET Bucket metadata notification configuration

GET Bucket metadata notification configuration 要求では、個々のバケットで検索統合を設定するために使用する設定 XML を読み出すことができます。

この処理を完了するには、 s3 ： GetBucketMetadataNotification 権限または root アカウントが必要です。

要求例

次の要求は、というバケットのメタデータ通知設定を読み出します `bucket`。

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

応答

応答の本文には、バケットのメタデータ通知設定が含まれます。メタデータ通知設定では、バケットでの検索統合の設定を確認できます。つまり、どのオブジェクトにインデックスが付けられ、そのオブジェクトメタデータがどのエンドポイントに送信され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ています。各ルールは、環境 がオブジェクトを指定し、 StorageGRID がオブジェクトメタデータを送信するデスティネーションを指定します。デスティネーションは、 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。

名前	説明	必須
MetadataNotificationConfiguration のページです	メタデータ通知でオブジェクトとデスティネーションの指定に使用されるルール用のコンテナタグ。 1 つ以上の Rule 要素を含みます。	はい。
ルール	指定したインデックスにメタデータを追加する必要があるオブジェクトを特定するルール用のコンテナタグ。 プレフィックスが重複しているルールは拒否されます。 MetadataNotificationConfiguration 要素に含まれています。	はい。
ID	ルールの一意の識別子。 Rule 要素に含まれています。	いいえ
ステータス	Status には「 Enabled 」または「 Disabled 」を指定できます。無効になっているルールについては操作が実行されません。 Rule 要素に含まれています。	はい。
プレフィックス	プレフィックスと一致するオブジェクトにルールが適用され、そのメタデータが指定したデスティネーションに送信されます。 すべてのオブジェクトを照合するには、空のプレフィックスを指定します。 Rule 要素に含まれています。	はい。
宛先	ルールのデスティネーションのコンテナタグ。 Rule 要素に含まれています。	はい。

名前	説明	必須
URN	オブジェクトメタデータが送信されるデスティネーションの URN 。次のプロパティを持つ StorageGRID エンドポイントの URN を指定する必要があります。 • `es` 3番目のエレメントである必要があります。 • URNの末尾に、メタデータが格納されるインデックスとタイプを、の形式で指定する必要があります `domain-name/myindex/mytype`。 エンドポイントは、 Tenant Manager またはテナント管理 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する必要があります。そうしないと、 404 エラーで設定が失敗します。 Urn は Destination 要素に含まれています。	はい。

応答例

間に含まれるXML
`<MetadataNotificationConfiguration></MetadataNotificationConfiguration>` タグは、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設定されているかを示します。次の例では、という名前のElasticsearchインデックスにオブジェクトメタデータが送信されています `current` と入力します `2017` という名前のAWSドメインでホストされている `records`。

```
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 metadata notification configuration のコマンドです

PUT Bucket metadata notification configuration 要求を使用すると、個々のバケットで検索統合サービスを有効化できます。要求の本文に含めるメタデータ通知設定 XML では、デスティネーション検索インデックスにメタデータを送信するオブジェクトを指定します。

この処理を完了するには、バケットの s3 ： PutBucketMetadataNotification 権限または root アカウントが必要です。

リクエスト

要求の本文にメタデータ通知設定が含まれている必要があります。各メタデータ通知設定には、 1 つ以上のルールが含まれています。各ルールは、環境 がオブジェクトを指定し、 StorageGRID がオブジェクトメタデータを送信するデスティネーションを指定します。

オブジェクトはオブジェクト名のプレフィックスでフィルタリングできます。たとえば、というプレフィックスのオブジェクトのメタデータを送信できます `/images` を1つのデスティネーションに、プレフィックスがのオブジェクトに追加します `/videos` 別のノードに移動します

プレフィックスが重複している設定は有効ではなく、送信時に拒否されます。たとえば、プレフィックスがのオブジェクト用のルールを1つ含む設定などです `test` プレフィックスが付いたオブジェクトの2番目のルールです `test2` 許可されません。

デスティネーションは、 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。エンドポイントは、メタデータ通知設定が送信されたときに存在している必要があります。存在していない場合、要求がとして失敗します `400 Bad Request`。エラーメッセージ： `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>
```

次の表に、メタデータ通知設定 XML の要素を示します。

名前	説明	必須
MetadataNotificationConfiguration のページです	メタデータ通知でオブジェクトとデスティネーションの指定に使用されるルール用のコンテナタグ。 1 つ以上の Rule 要素を含みます。	はい。
ルール	指定したインデックスにメタデータを追加する必要があるオブジェクトを特定するルール用のコンテナタグ。 プレフィックスが重複しているルールは拒否されます。 MetadataNotificationConfiguration 要素に含まれています。	はい。
ID	ルールの一意の識別子。 Rule 要素に含まれています。	いいえ

名前	説明	必須
ステータス	Status には「 Enabled 」または「 Disabled 」を指定できます。無効になっているルールについては操作が実行されません。 Rule 要素に含まれています。	はい。
プレフィックス	プレフィックスと一致するオブジェクトにルールが適用され、そのメタデータが指定したデスティネーションに送信されます。 すべてのオブジェクトを照合するには、空のプレフィックスを指定します。 Rule 要素に含まれています。	はい。
宛先	ルールのデスティネーションのコンテナタグ。 Rule 要素に含まれています。	はい。
URN	オブジェクトメタデータが送信されるデスティネーションの URN 。次のプロパティを持つ StorageGRID エンドポイントの URN を指定する必要があります。 • `es` 3番目のエレメントである必要があります。 • URNの末尾に、メタデータが格納されるインデックスとタイプを、の形式で指定する必要があります `domain-name/myindex/mytype`。 エンドポイントは、 Tenant Manager またはテナント管理 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する必要があります。そうしないと、 404 エラーで設定が失敗します。 Urn は Destination 要素に含まれています。	はい。

例をリクエストする

次の例は、バケットの検索統合を有効にする方法を示しています。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デスティネーションに送信されます。

```
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、プレフィックスに一致するオブジェクトのオブジェクトメタデータを指定します `/images` が1つのデスティネーションに送信され、プレフィックスに一致するオブジェクトのオブジェクトメタデータが送信されます `/videos` 2番目の送信先に送信されます。

```
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>
```

検索統合サービスで生成される **JSON**

バケットで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の追加、更新、削除が行われるたびに、 JSON ドキュメントが生成されてデスティネーションエンドポイントに送信されます。

次の例は、キーを含むオブジェクトの場合に生成されるJSONを示しています `SGWS/Tagging.txt` は、という名前のバケットに作成されます `test`。。 `test` バケットはバージョン管理されていないため、を使用します `versionId` タグが空です。

```
{
  "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ているオブジェクトメタデータ

次の表に、検索統合が有効になっている場合にデスティネーションエンドポイントに送信される JSON ドキュメント内のすべてのフィールドを示します。

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

を入力します	項目名	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	キーを押します	オブジェクトキーの名前
バケットとオブジェクトの情報	versionId	バージョン管理されたバケット内のオブジェクトのオブジェクトバージョン
バケットとオブジェクトの情報	リージョン	たとえば、バケットのリージョンのように指定します `us-east-1`
システムメタデータ	サイズ	HTTP クライアントから認識できるオブジェクトのサイズ（バイト）
システムメタデータ	MD5	オブジェクトのハッシュ
ユーザメタデータ	メタデータ *`key:value`*	オブジェクトのすべてのユーザメタデータをキーと値のペアとして格納

を入力します	項目名	説明
タグ	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をキーと値のペアとして使用します

タグとユーザメタデータの場合、 StorageGRID は文字列または S3 イベント通知として Elasticsearch に日付と番号を渡します。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 Elasticsearch を設定するには、動的フィールドマッピングおよびマッピング日付形式に関する Elasticsearch の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、インデックスの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。ドキュメントのインデックス作成後は、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"

GET Storage Usage 要求の略

GET Storage Usage 要求を使用すると、アカウントで使用しているストレージの総容量とアカウントに関連付けられているバケットごとの使用容量を確認できます。

アカウントとそのバケットで使用されているストレージの量は、でGET Service要求を変更して取得できます `x-ntap-sg-usage` クエリパラメータ。バケットによるストレージの使用量は、システムで処理される PUT 要求や DELETE 要求とは別に追跡されます。特にシステムの負荷が高い場合などは、使用量の値が要求の処理に基づく想定値と同じになるまでに少し時間がかかることがあります。

デフォルトでは、 StorageGRID は strong-global 整合性を使用して、使用状況の情報を取得します。strong-global整合性を達成できない場合、StorageGRID はstrong-site整合性で使用状況情報を取得しようとします。

この処理を完了するには、 s3 ： ListAllMyBuckets 権限または root アカウントが必要です。

要求例

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

応答例

次の例は、 2 つのバケットに 4 つのオブジェクトと 12 バイトのデータが格納されたアカウントです。各バケットには、 2 つのオブジェクトと 6 バイトのデータが格納されています。

```
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

には、格納されているすべてのオブジェクトバージョンが関連します `ObjectCount` および `DataBytes` 応答の値。削除マーカーはに追加されません `ObjectCount` 合計。

関連情報

"整合性制御"

従来の準拠のためのバケット要求が廃止されました

従来の準拠機能で作成されたバケットの管理には、 StorageGRID S3 REST API の使用が必要になる場合があります。

コンプライアンス機能は廃止されました

以前のバージョンの StorageGRID で提供されていた StorageGRID 準拠機能は廃止され、 S3 オブジェクトロックに置き換えられました。

グローバル準拠設定を有効にしている場合は、 StorageGRID 11.6 でグローバル S3 オブジェクトロック設定

が有効になっています。準拠を有効にした新しいバケットは作成できなくなりました。ただし、必要に応じて、 StorageGRID S3 REST API を使用して、従来の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します"
- "ILM を使用してオブジェクトを管理する"
- "ネットアップのナレッジベース： StorageGRID 11.5 でレガシー準拠バケットを管理する方法"

廃止された準拠要求：

- "DEPRECATED - PUT Bucket request modifications for compliance"

  SGCompliance XML 要素は廃止されました。これまでは、この StorageGRID カスタム要素を PUT Bucket 要求のオプションの XML 要求の本文に含めて準拠バケットを作成できました。

- "廃止予定- GET Bucket compliance"

  GET Bucket compliance 要求は廃止されました。ただし、既存のレガシー準拠バケットに対して現在有効な準拠設定を引き続き確認することができます。

- "廃止されました。PUT Bucket compliance"

  PUT Bucket compliance 要求は廃止されました。ただし、この要求を引き続き使用して、既存のレガシー準拠バケットの準拠設定を変更できます。たとえば、既存のバケットをリーガルホールドの対象にしたり、バケットの保持期間を長くしたりできます。

廃止：準拠のための **PUT Bucket** 要求の変更

SGCompliance XML 要素は廃止されました。これまでは、この StorageGRID カスタム要素を PUT Bucket 要求のオプションの XML 要求の本文に含めて準拠バケットを作成できました。

以前のバージョンの StorageGRID で提供されていた StorageGRID 準拠機能は廃止され、 S3 オブジェクトロックに置き換えられました。

"S3 REST APIを使用してS3オブジェクトロックを設定します"

"ILM を使用してオブジェクトを管理する"

"ネットアップのナレッジベース： StorageGRID 11.5 でレガシー準拠バケットを管理する方法"

準拠を有効にした新しいバケットを作成することはできなくなりました。準拠バケットを新しく作成するために PUT Bucket 要求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.
```

廃止予定： **GET Bucket compliance** 要求

GET Bucket compliance 要求は廃止されました。ただし、既存のレガシー準拠バケットに対して現在有効な準拠設定を引き続き確認することができます。

以前のバージョンの StorageGRID で提供されていた StorageGRID 準拠機能は廃止され、 S3 オブジェクトロックに置き換えられました。

"S3 REST APIを使用してS3オブジェクトロックを設定します"

"ILM を使用してオブジェクトを管理する"

"ネットアップのナレッジベース： StorageGRID 11.5 でレガシー準拠バケットを管理する方法"

この処理を完了するには、 s3 ： GetBucketCompliance 権限または root アカウントが必要です。

要求例

次の要求例では、という名前のバケットの準拠設定を確認できます `mybucket`。

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

応答例

応答XMLで、 `<SGCompliance>` バケットで有効な準拠設定が表示されます。次の応答例では、バケットの準拠設定が示されており、各オブジェクトはグリッドに取り込まれてから 1 年間（ 525 、 600 分）保持されます。このバケットには現在リーガル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>
```

名前	説明
RetentionPeriodMinutes です	このバケットに追加されたオブジェクトの保持期間を分で指定します。保持期間は、オブジェクトがグリッドに取り込まれたときからの期間です。
LegalHold のようになります	• True ：このバケットは、現在リーガルホールドの対象です。このバケット内のオブジェクトは、保持期間が過ぎても、リーガルホールドが解除されるまで削除できません。 • False ：このバケットは、現在リーガルホールドの対象ではありません。このバケット内のオブジェクトは、保持期間が過ぎたら削除できます。
自動削除	• True ：このバケット内のオブジェクトは、バケットがリーガルホールドの対象である場合を除き、保持期間が過ぎ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、保持期間が過ぎても自動的には削除されません。これらのオブジェクトを削除する必要がある場合は、手動で削除する必要があります。

エラー応答

バケットが準拠バケットとして作成されていない場合、応答のHTTPステータスコードはになります `404 Not Found`を返します `XNoSuchBucketCompliance`。

廃止予定： **PUT Bucket compliance** 要求

PUT Bucket compliance 要求は廃止されました。ただし、この要求を引き続き使用して、既存のレガシー準拠バケットの準拠設定を変更できます。たとえば、既存のバケットをリーガルホールドの対象にしたり、バケットの保持期間を長くしたりできます。

以前のバージョンの StorageGRID で提供されていた StorageGRID 準拠機能は廃止され、 S3 オブジェクトロックに置き換えられました。

"S3 REST APIを使用してS3オブジェクトロックを設定します"

"ILM を使用してオブジェクトを管理する"

"ネットアップのナレッジベース： StorageGRID 11.5 でレガシー準拠バケットを管理する方法"

この処理を完了するには、 s3 ： PutBucketCompliance 権限または root アカウントが必要です。

PUT Bucket compliance 要求を発行する際は、準拠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。

要求例

次の要求例では、という名前のバケットの準拠設定を変更します `mybucket`。この例では、のオブジェクトが表示されています `mybucket` オブジェクトがグリッドに取り込まれてから1年間ではなく2年間（1、051

、200分）保持されます。このバケットに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>
```

名前	説明
RetentionPeriodMinutes です	このバケットに追加されたオブジェクトの保持期間を分で指定します。保持期間は、オブジェクトがグリッドに取り込まれたときからの期間です。 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の設定後は、その値を減らすことはできず、増やすことしかできません。
LegalHold のようになります	• True ：このバケットは、現在リーガルホールドの対象です。このバケット内のオブジェクトは、保持期間が過ぎても、リーガルホールドが解除されるまで削除できません。 • False ：このバケットは、現在リーガルホールドの対象ではありません。このバケット内のオブジェクトは、保持期間が過ぎたら削除できます。
自動削除	• True ：このバケット内のオブジェクトは、バケットがリーガルホールドの対象である場合を除き、保持期間が過ぎ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、保持期間が過ぎても自動的には削除されません。これらのオブジェクトを削除する必要がある場合は、手動で削除する必要があります。

準拠設定の整合性レベル

PUT Bucket compliance 要求によって S3 バケットの準拠設定を更新すると、 StorageGRID は、グリッド全体の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 * 整合性レベルを使用し、バケットのメタデータを含むすべてのデータセンターサイトおよびストレージノードで、変更された準拠設定のリードアフターライト整合性を保証します。

データセンターサイトまたはサイトの複数のストレージノードが利用できないために、StorageGRID が*

strong-global *整合性レベルを達成できない場合、応答のHTTPステータスコードはになります `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をできるだけ早く利用可能にするために、グリッド管理者に問い合わせる必要があります。グリッド管理者が各サイトで十分な数のストレージノードを利用可能にできない場合は、テクニカルサポートから、 * strong-site * 整合性レベルを強制的に適用することで、失敗した要求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があった場合や、このレベルを使用した場合の影響を理解している場合を除き、 PUT Bucket compliance で * strong-site * 整合性レベルを強制的に適用することは避けてください。

整合性レベルを * strong-site * 」に下げると、 StorageGRID は、サイト内のクライアント要求に対してのみ、更新された準拠設定のリードアフターライト整合性を保証します。そのため、すべてのサイトおよびストレージノードが利用可能になるまでの間、 StorageGRID システムにはこのバケットに対して複数の異なる設定が一時的に存在することになる場合があります。整合性のない設定を使用すると、予期せぬ望ましくない動作が生じる可能性がありますたとえば、あるバケットをリーガルホールドの対象にして、低い整合性レベルを強制的に適用すると、一部のデータセンターサイトでバケットの以前の準拠設定（つまり、リーガルホールドの対象外の状態）が引き続き適用される場合があります。したがって、リーガルホールドの対象と思われるオブジェクトは、保持期間が経過すると、ユーザによって削除される場合と、 AutoDelete によって削除される場合があります。

strong-site *整合性レベルを強制的に適用するには、PUT Bucket compliance要求を再発行し、を含めてください `Consistency-Control` HTTP要求ヘッダー。

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

エラー応答

- バケットが準拠バケットとして作成されていない場合、応答のHTTPステータスコードはになります `404 Not Found`。
- 状況 `RetentionPeriodMinutes` 要求がバケットの現在の保持期間よりも短い場合、HTTPステータスコードはになります `400 Bad Request`。

関連情報

"廃止：準拠のための PUT Bucket 要求の変更"

バケットとグループのアクセスポリシー

バケットとグループのアクセスポリシーを使用

StorageGRID では、 Amazon Web Services （ AWS ）ポリシー言語を使用して、 S3 テナント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できます。StorageGRID システムには、 S3 REST API ポリシー言語のサブセットが実装されています。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 では 2 種類のアクセスポリシーがサポートされています。

- * バケットポリシー * 。 GET Bucket policy 、 PUT Bucket policy 、 DELETE Bucket policy の各 S3 API 処理を使用して設定します。バケットポリシーはバケットに関連付けられ、バケットとそのオブジェクトへのバケット所有者アカウントやその他のアカウントのユーザによるアクセスを制御するために使用されます。バケットポリシー環境 は 1 つのバケットのみで、場合によっては複数のグループに分かれています。
- * グループポリシー * 。 Tenant Manager またはテナント管理 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のグループに関連付けられ、そのアカウントが所有する特定のリソースにそのグループがアクセスできるように設定されます。グループポリシー環境 は 1 つのグループに限定され、場合によっては複数のバケットに適用されます。

グループポリシーとバケットポリシーの優先度に違いはありません。

StorageGRID のバケットとグループのポリシーは、 Amazon が定義している特定の文法に従って記述されます。各ポリシーは一連のステートメントからなり、各ステートメントは次の要素で構成されます。

- ステートメント ID （ SID ）（オプション）
- 効果
- プリンシパル / NotPrincipal
- リソース / メモリソース
- アクション / NotAction
- Condition （オプション）

次の構造を使用して、権限を指定するポリシーステートメントが構築されます。 <Effect> を付与して、 <Condition> に該当する場合に <Principal> に <Resource> に対する <Action> の実行を許可または拒否します。

各ポリシー要素は、特定の機能に使用されます。

要素（ **Element** ）	説明
SID	Sid 要素はオプションです。SID は、ユーザの概要 としてのみ使用されます。StorageGRID システムに格納はされますが、システムで解釈されません。
効果	Effect 要素では、指定した処理を許可するか拒否するかを指定します。Action 要素でサポートされるキーワードを使用して、バケットやオブジェクトで許可（または拒否）する処理を指定する必要があります。

要素（ **Element** ）	説明
プリンシパル / NotPrincipal	ユーザ、グループ、およびアカウントに特定のリソースへのアクセスと特定の操作の実行を許可できます。要求に S3 の署名が含まれていない場合は、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、アカウントが所有するリソースへのアクセスは root アカウントにのみ許可されます。 Principal 要素を指定する必要があるのはバケットポリシーだけです。グループポリシーの場合は、ポリシーが関連付けられたグループが暗黙的にプリンシパルになります。
リソース / メモリソース	Resource 要素では、バケットとオブジェクトを指定します。Amazon リソースネーム（ ARN ）を使用してリソースを指定し、バケットやオブジェクトに対する権限を許可または拒否することができます。
アクション / NotAction	権限は Action 要素と Effect 要素の 2 つで構成されます。グループがリソースを要求すると、リソースへのアクセスが許可または拒否されます。権限を明示的に割り当てていないかぎりアクセスは拒否されますが、明示的な拒否を使用して別のポリシーで付与された権限を上書きすることもできます。
条件	Condition 要素はオプションです。条件を使用すると、ポリシーを適用する条件を示す式を作成できます。

Action 要素では、ワイルドカード文字（ * ）を使用してすべての処理または処理のサブセットを指定できます。たとえば、次の Action の値は、 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（ * ）および（？）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、疑問符（？）は 0 文字以上の文字に一致します。 任意の 1 文字に一致します。

Principal要素では、匿名アクセスを設定してすべてのユーザに権限を付与する場合を除き、ワイルドカード文字はサポートされません。たとえば、 Principal の値としてワイルドカード（ * ）を設定します。

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

次の例では、 Effect 、 Principal 、 Action 、および Resource の各要素を使用して記述します。次の例は、「許可」の効果を使用してプリンシパル、adminグループを指定したバケットポリシーのステートメントを示しています `federated-group/admin` 財務グループなどです `federated-group/finance`、アクションを実行する権限 `s3:ListBucket` をバケットにインストールします `mybucket` そしてアクション `s3:GetObject` そのバケット内のすべてのオブジェクト。

```
{
  "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:iam:s3:::mybucket",
        "arn:aws:iam:s3:::mybucket/*"
      ]
    }
  ]
}
```

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

ポリシーの整合性制御設定

デフォルトでは、グループポリシーに対するすべての更新の整合性レベルは結果整合性です。グループポリシーが整合した状態になっても、ポリシーキャッシュのために、変更が有効になるまでさらに 15 分を要することがあります。デフォルトでは、バケットポリシーに対するすべての更新の整合性レベルも結果整合性です。

バケットポリシーの更新の整合性保証は必要に応じて変更できます。たとえば、セキュリティ上の理由から、できるだけ早くバケットポリシーの変更を有効にしなければならない場合があります。

この場合は、を設定できます `Consistency-Control` PUT Bucket policy要求のヘッダーを指定するか、PUT Bucket整合性要求を使用できます。この要求で整合性制御を変更する場合は、値「 * all * 」を使用して最高レベルのリードアフターライト整合性を保証する必要があります。それ以外の整合性制御値を PUT Bucket consistency 要求のヘッダーで指定すると、要求は拒否されます。PUT Bucket policy 要求でそれ以外の値を指定した場合は、値が無視されます。バケットポリシーが整合した状態になっても、ポリシーキャッシュのために、変更が有効になるまでさらに 8 秒を要することがあります。

新しいバケットポリシーを速やかに有効にするために整合性レベルを * all * に設定する場合は、処理が完了したあとに必ずバケットレベルの制御を元の値に戻してください。そうしないと、それ以降のすべてのバケット要求で * all * 設定が使用されます。

ポリシーステートメントでは **ARN** を使用します

ポリシーステートメントでは、 Principal 要素と Resource 要素で ARN を使用します。

- S3 リソースの ARN の指定には次の構文を使用します。

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

- アイデンティティリソースの ARN （ユーザおよびグループ）の指定には次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：

- オブジェクトキーの一部にワイルドカードとしてアスタリスク（ * ）を使用すると、 0 文字以上の文字に一致します。
- オブジェクトキーで指定できる国際文字は、 JSON UTF-8 形式または 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。パーセントエンコーディングはサポートされていません。

  "RFC 2141 の URN 構文"

  PUT Bucket policy 処理の HTTP 要求の本文は、 charset=UTF-8 でエンコードする必要があります。

ポリシー内のリソースを指定します

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

- Resource 要素はポリシーの各ステートメントに必要です。ポリシーでは、リソースは要素で示されます `Resource`または、 `NotResource` 除外のため。
- リソースは S3 リソースの ARN で指定します。例：

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

- オブジェクトキーの内部でポリシー変数を使用することもできます。例：

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

- グループポリシーの作成時は、まだ存在しないバケットもリソースの値で指定することができます。

ポリシーでプリンシパルを指定します

ポリシーステートメントでリソースへのアクセスを許可または拒否するユーザ、グループ、またはテナントアカウントを指定するには、 Principal 要素を使用します。

- バケットポリシー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。グループはプリンシパルとみなされるため、グループポリシーのポリシーステートメントではPrincipal要素は必要ありません。
- ポリシーでは ' 主体は ' 主 (Principal)' または除外のためにもう 1 つの "NotPrincipal" という要素によって示されます
- ID または ARN を使用してアカウントベースのアイデンティティを指定する必要があります。

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

- 次の例では、テナントアカウント ID 27233906934684427525 を使用しています。この場合、 root アカウントとそのすべてのユーザが含まれます。

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

- root アカウントのみを指定する場合は次のようになります。

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

- 特定のフェデレーテッドユーザ（「 Alex 」）を指定する場合は次のようになります。

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

- 特定のフェデレーテッドグループ（「 Managers 」）のみを指定する場合は次のようになります。

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

- 匿名プリンシパルを指定する場合は次のようになります。

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

- あいまいさを排除するために、ユーザ名の代わりに UUID を使用できます。

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

たとえば、Alexが組織とユーザ名を退職するとします `Alex` が削除されました。新しいAlexが組織に参加し、同じが割り当てられている場合 `Alex` ユーザ名。元のユーザに付与された権限が、新しいユーザに意図せず継承されることがあります。

- バケットポリシーの作成時は、まだ存在しないグループ / ユーザの名前もプリンシパルの値で指定することができます。

ポリシーで権限を指定します

ポリシーでは、 Action 要素を使用してリソースに対する権限を許可または拒否します。ポリシーには、「 Action 」要素で示される一連の権限、または除外する「 NotAction 」要素で指定できる一連の権限があります。それぞれが特定の S3 REST API 処理に対応しています。

次の表に、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を示します。

Amazon S3 では、 PUT と DELETE Bucket の両方のレプリケーション処理に s3 ： PutReplicationConfiguration 権限が使用されるようになりました。StorageGRID では、元の Amazon S3 仕様に一致する個別の権限が各アクションに使用されます。

DELETE は、 PUT を使用して既存の値を上書きするときに実行されます。

バケットに適用される権限

権限	**S3 REST API** の処理	**StorageGRID** のカスタム
S3 ： CreateBucket を指定します	PUT Bucket の場合	
S3 ： DeleteBucket	バケットを削除します	
S3 ： DeleteBucketMetadataNotification	バケットのメタデータ通知設定を削除します	はい。
S3 ： DeleteBucketPolicy	バケットポリシーを削除	
S3 ： DeleteReplicationConfiguration	バケットレプリケーションを削除します	はい。 PUT および DELETE の権限は分離されています
S3 ： GetBucketAcl	GET Bucket ACL の場合	
S3 ： GetBucketCompliance	GET Bucket compliance （廃止）	はい。

権限	**S3 REST API** の処理	**StorageGRID** のカスタム
S3 ： GetBucketConsistency	GET Bucket consistency	はい。
S3 ： GetBucketCORS	GET Bucket CORS	
S3 ： GetEncryptionConfiguration	GET Bucket encryption	
S3 ： GetBucketLastAccessTime	GET Bucket last access time の場合	はい。
S3 ： GetBucketLocation	GET Bucket location の各ノードで使用でき	
S3 ： GetBucketMetadataNotification	GET Bucket metadata notification configuration	はい。
S3 ： GetBucketNotification	GET Bucket notification	
S3 ： GetBucketObjectLockConfiguration	オブジェクトロック設定の取得	
S3 ： GetBucketPolicy	GET Bucket policy の場合	
S3 ： GetBucketTagging	GET Bucket tagging	
S3 ： GetBucketVersioning	GET Bucket versioning	
S3 ： GetLifecycleConfiguration	GET Bucket lifecycle	
S3 ： GetReplicationConfiguration	GET Bucket replication	
S3 ： ListAllMyBuckets	• GET Service の略 • GET Storage Usage の略	GET Storage Usage の場合は、はい
S3 ： ListBucket	• GET Bucket （ List Objects ） • HEAD Bucket （ヘッドバケット） • POST Object restore の実行	
S3 ： ListBucketMultipartUploads	• マルチパートアップロードをリストします • POST Object restore の実行	

権限	**S3 REST API** の処理	**StorageGRID** のカスタム
S3 ： ListBucketVersions	GET Bucket versions （バケットバージョンの取得	
S3 ： PutBucketCompliance	PUT Bucket compliance （廃止）	はい。
S3 ： PutBucketConsistency	PUT Bucket consistency	はい。
S3 ： PutBucketCORS	• バケットの CORS を削除† • PUT Bucket CORS	
S3 ： PutEncryptionConfiguration	• バケットの暗号化を削除 • PUT Bucket encryption	
S3 ： PutBucketLastAccessTime	PUT Bucket last access time のように指定します	はい。
S3 ： PutBucketMetadataNotification	PUT Bucket metadata notification configuration のコマンドです	はい。
S3 ： PutBucketNotification	PUT Bucket notification	
S3 ： PutBucketObjectLockConfiguration	• PUT Bucketにで接続します `x-amz-bucket-object-lock-enabled: true` 要求ヘッダー（s3 ：CreateBucket権限も必要） • PUT Object Lock の設定を指定します	
S3 ： PutBucketPolicy	PUT Bucket policy の場合	
S3 ： PutBucketTagging	• バケットタグを削除† • PUT Bucket tagging	
S3 ： PutBucketVersioning	PUT Bucket versioning の場合	
S3 ： PutLifecycleConfiguration	• バケットライフサイクルを削除† • PUT Bucket lifecycle の場合	
S3 ： PutReplicationConfiguration	PUT Bucket replication	はい。 PUT および DELETE の権限は分離されています

オブジェクトに適用される権限

権限	**S3 REST API** の処理	**StorageGRID** のカスタム
S3 ： AbortMultipartUpload	• マルチパートアップロードを中止します • POST Object restore の実行	
S3：Bypassガバナー 保持	• オブジェクトを削除します • 複数のオブジェクトを削除します • PUT Object retention のことです	
S3 ： DeleteObject	• オブジェクトを削除します • 複数のオブジェクトを削除します • POST Object restore の実行	
S3 ： DeleteObjectTagging	オブジェクトのタグ付けを削除します	
S3 ： DeleteObjectVersionTagging	DELETE Object Tagging （オブジェクトの特定のバージョン）	
S3 ： DeleteObjectVersion	DELETE Object （オブジェクトの特定のバージョン）	
S3 ： GetObject	• オブジェクトの取得 • HEAD Object の実行 • POST Object restore の実行 • オブジェクトコンテンツを選択します	
S3 ： GetObjectAcl	GET Object ACL の場合	
S3 ： GetObjectLegalHold	オブジェクトのリーガルホールドを取得します	
S3 ： GetObjectRetention	GET Object retention のことです	
S3 ： GetObjectTagging	GET Object Tagging の場合	
S3 ： GetObjectVersionTagging	GET Object Tagging （オブジェクトの特定のバージョン）	

権限	**S3 REST API** の処理	**StorageGRID** のカスタム
S3 ： GetObjectVersion	GET Object （オブジェクトの特定のバージョン）	
S3 ： ListMultipartUploadParts	パーツを表示し、 POST Object restore を実行します	
S3 ： PutObject	• PUT Object の場合 • PUT Object - Copy の各コマンドを実行します • POST Object restore の実行 • マルチパートアップロードを開始します • Complete Multipart Upload の実行 • パーツをアップロードします • パーツのアップロード - コピー	
S3 ： PutObjectLegalHold	オブジェクトのリーガルホールドを適用します	
S3 ： PutObjectRetention	PUT Object retention のことです	
S3 ： PutObjectTagging	PUT Object Tagging の場合	
S3 ： PutObjectVersionTagging	PUT Object Tagging （オブジェクトの特定のバージョン）	
S3 ： PutOverwriteObject	• PUT Object の場合 • PUT Object - Copy の各コマンドを実行します • PUT Object tagging • オブジェクトのタグ付けを削除します • Complete Multipart Upload の実行	はい。
S3 ： RestoreObject	POST Object restore の実行	

PutOverwriteObject 権限を使用します

s3 ： PutOverwriteObject 権限は、オブジェクトの作成または更新を行う環境 処理のカスタムの StorageGRID 権限です。この権限の設定により、オブジェクトのデータ、ユーザ定義メタデータ、または S3 オブジェクトのタグをクライアントが上書きできるかどうかが決まります。

この権限で可能な設定は次のとおりです。

- * allow * ：クライアントはオブジェクトを上書きできます。これがデフォルト設定です。
- **Deny**:クライアントはオブジェクトを上書きできません。PutOverwriteObject 権限が Deny に設定されている場合の動作は次のとおりです。
  - 同じパスで既存のオブジェクトが見つかった場合は、次の手順を実行します。
    - オブジェクトのデータ、ユーザ定義メタデータ、またはS3オブジェクトのタグを上書きすることはできません。
    - 実行中の取り込み処理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は、 Deny に設定すると、 PUT Object tagging 処理または DELETE Object tagging 処理によって、オブジェクトとその最新ではないバージョンの TagSet が変更されなくなります。
  - 既存のオブジェクトが見つからない場合は、この権限の設定は影響しません。
- この権限がない場合、 Allow が設定されたものと同じ結果になります。

> 現在のS3ポリシーで上書きが許可されていて、PutOverwriteObject権限がDenyに設定されている場合、オブジェクトのデータ、ユーザ定義メタデータ、またはオブジェクトのタグをクライアントが上書きすることはできません。また、**[Prevent client modification]***チェックボックスが選択されている場合（ configuration **>** Security settings **>** Network and objects *）、この設定はPutOverwriteObject権限の設定よりも優先されます。

ポリシーの条件を指定します

条件は、ポリシーが有効になるタイミングを定義します。条件は演算子とキーと値のペアで構成されます。

条件はキーと値のペアを使用して評価されます。Condition 要素には複数の条件を指定でき、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。条件ブロックの形式は次のとおりです。

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

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

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

サポートされる条件演算子は次の

条件演算子は次のように分類されます。

- 文字列
- 数値
- ブール値
- IP アドレス
- Null チェック

条件演算子	説明
StringEquals	キーを文字列値と比較し、完全一致であるかを確認します（大文字と小文字の区別あり）。
StringNotEquals	キーを文字列値と比較し、不一致であるかを確認します（大文字と小文字の区別あり）。
StringEqualsIgnoreCase	キーを文字列値と比較し、完全一致であるかを確認します（大文字と小文字の区別なし）。
StringNotEqualsIgnoreCase	キーを文字列値と比較し、不一致であるかを確認します（大文字と小文字の区別なし）。
StringLike	キーを文字列値と比較し、完全一致であるかを確認します（大文字と小文字の区別あり）。含めることができる * と？ワイルドカード文字を使用できます。
StringNotLike	キーを文字列値と比較し、不一致であるかを確認します（大文字と小文字の区別あり）。含めることができる * と？ワイルドカード文字を使用できます。
NumericEquals （数値機器）	キーを数値と比較し、完全一致であるかを確認します。
NumericNotEquals	キーを数値と比較し、不一致であるかを確認します。
NumericGreaterThan	キーを数値と比較し、「大なり」の一致であるかを確認します。
NumericGreaterThanEquals	キーを数値と比較し、「大なり」または「等しい」の一致であるかを確認します。
NumericLessThan	キーを数値と比較し、「より小さい」の一致であるかを確認します。
NumericLessThanEquals	キーを数値と比較し、「より小さい」または「等しい」の一致であるかを確認します。
ブール値	キーをブール値と比較し、「 true 」または「 false 」の一致であるかを確認します。

条件演算子	説明
IP アドレス	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。
NotIpAddress	キーを IP アドレスまたは IP アドレスの範囲と比較し、不一致であるかを確認します。
null	現在の要求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。

サポートされている条件キー

カテゴリ	適用される条件キー	説明
IP 演算子	AWS ： sourceIP	要求の送信元の IP アドレスと比較します。バケットまたはオブジェクトの処理に使用できます。 • 注： S3 要求が管理ノードおよびゲートウェイノード上のロードバランササービスを介して送信された場合は、ロードバランササービスのアップストリームの IP アドレスと比較します。 • 注 * ：サードパーティ製の非透過型ロードバランサを使用する場合は、そのロードバランサの IP アドレスと比較します。任意 `X-Forwarded-For` ヘッダーの有効性を確認できないため、ヘッダーは無視されます。
リソース / ID	AWS ：ユーザ名	要求の送信者のユーザ名と比較します。バケットまたはオブジェクトの処理に使用できます。
S3 ： ListBucket と S3 ： ListBucketVersions 権限	S3 ：デリミタ	GET Bucket 要求または GET Bucket Object versions 要求で指定された delimiter パラメータと比較します。
S3 ： ListBucket と S3 ： ListBucketVersions 権限	S3 ： max-keys	GET Bucket 要求または GET Bucket Object versions 要求で指定された max-keys パラメータと比較します。
S3 ： ListBucket と S3 ： ListBucketVersions 権限	S3 ：プレフィックス	GET Bucket 要求または GET Bucket Object versions 要求で指定された prefix パラメータと比較します。

カテゴリ	適用される条件キー	説明
S3 ： PutObject	S3 ： object-lock-remaining-retention-days	で指定されたretain-until-dateと比較します `x-amz-object-lock-retain-until-date` 次の要求について、これらの値が許容範囲内であることを確認するために、要求ヘッダーまたはバケットのデフォルト保持期間から計算されます。 • PUT Object の場合 • PUT Object - Copy の各コマンドを実行します • マルチパートアップロードを開始します
S3 ： PutObjectRetention	S3 ： object-lock-remaining-retention-days	PUT Object Retention 要求で指定された retain-until 日と比較して、許容範囲内にあることを確認します。

ポリシーで変数を指定します

ポリシーで変数を使用すると、該当するポリシーの情報を設定できます。でポリシー変数を使用できます `Resource` の要素と文字列比較 `Condition` 要素（Element）：

この例では、変数を使用しています `${aws:username}` はResource要素の一部です。

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

この例では、変数を使用しています `${aws:username}` は、条件ブロックの条件値の一部です。

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

変数（ **Variable** ）	説明
`${aws:SourceIp}`	SourceIp キーを指定の変数として使用します。
`${aws:username}`	username キーを指定の変数として使用します。
`${s3:prefix}`	サービス固有のプレフィックスキーを指定の変数として使用します。
`${s3:max-keys}`	サービス固有の max-keys キーを指定の変数として使用します。

変数（ **Variable** ）	説明
`${*}`	特殊文字です。文字をリテラル * 文字として使用します。
`${?}`	特殊文字です。文字をリテラル文字として使用しますか？を押します。
`${$}`	特殊文字です。文字「 $ 」をリテラル文字として使用します。

特別な処理を必要とするポリシーを作成します

ポリシーで付与される権限によって、アカウントの root ユーザがロックアウトされるなど、セキュリティや継続的な運用に支障が生じることがあります。StorageGRID の S3 REST API の実装では、ポリシーの検証時の制限は Amazon よりも厳しくありませんが、評価時は同等の制限が適用されます。

Policy 概要 の略	ポリシータイプ	**Amazon** の動作	**StorageGRID** の動作
自身に対し、 root アカウントに対するすべての権限を拒否する	バケット	有効で適用されるが、 S3 バケットのすべてのポリシー処理に対する権限は引き続き root ユーザアカウントに付与される	同じ
自身に対しユーザ / グループに対するすべての権限を拒否する	グループ	有効で適用されます	同じ
外部アカウントグループに対し任意の権限を許可します	バケット	無効なプリンシパルです	有効だが、 S3 バケットのすべてのポリシー処理に対する権限をポリシーで許可すると 405 Method Not Allowed エラーが返されます
外部アカウントの root またはユーザに任意の権限を許可します	バケット	有効だが、 S3 バケットのすべてのポリシー処理に対する権限をポリシーで許可すると 405 Method Not Allowed エラーが返されます	同じ
すべてのユーザにすべての処理に対する権限を許可します	バケット	有効だが、外部アカウントの root およびユーザについては、 S3 バケットのすべてのポリシー処理に対する権限で 405 Method Not Allowed エラーが返されます	同じ

Policy 概要 の略	ポリシータイプ	**Amazon** の動作	**StorageGRID** の動作
すべてのユーザに対してすべての処理に対する権限を拒否する	バケット	有効で適用されるが、 S3 バケットのすべてのポリシー処理に対する権限は引き続き root ユーザアカウントに付与される	同じ
プリンシパルとして新規のユーザまたはグループを指定します	バケット	無効なプリンシパルです	有効
リソースとして新規の S3 バケットを指定する必要があります	グループ	有効	同じ
プリンシパルとしてローカルグループを指定します	バケット	無効なプリンシパルです	有効
ポリシーでは、非所有者アカウント（匿名アカウントを含む）にオブジェクトを PUT する権限が付与されます	バケット	有効。オブジェクトは作成者アカウントによって所有され、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、オブジェクトの ACL を使用してオブジェクトにアクセス権限を付与する必要があります。	有効。オブジェクトはバケット所有者アカウントによって所有され、バケットポリシーが適用される。

Write-Once-Read-Many （ WORM ）による保護

データ、ユーザ定義オブジェクトのメタデータ、 S3 オブジェクトのタグを保護するために、 Write-Once-Read-Many （ WORM ）バケットを作成することができます。新しいオブジェクトの作成を許可し、既存のコンテンツの上書きや削除を防止するように WORM バケットを設定します。ここで説明するいずれかの方法を使用します。

上書きを常に拒否するには、次の操作を実行します。

- Grid Managerで、* configuration **>** Security **>** Security settings **>** Network and objects の順に選択し、Prevent client modification *チェックボックスを選択します。
- 次のルールと S3 ポリシーを適用します。
  - S3 ポリシーに PutOverwriteObject DENY 処理を追加します。
  - S3 ポリシーに DeleteObject DENY 処理を追加します。
  - S3 ポリシーに PUT Object ALLOW 処理を追加します。

S3 ポリシーで DeleteObject を DENY に設定しても、「 zero copies after 30 days 」のようなルールに基づく ILM によるオブジェクトの削除は実行されます。

これらのルールとポリシーがすべて適用されても、同時書き込みからは保護されません（状況Aを参照）。保護の対象になるのはシーケンシャルな上書きです（状況 B を参照）。

- 状況 A * ：同時書き込み（保護対象外）

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

- 状況 B * ：シーケンシャルな上書き（保護対象）

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

関連情報

- "StorageGRID の ILM ルールによるオブジェクトの管理"
- "バケットポリシーの例"
- "グループポリシーの例"
- "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"

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。

バケットポリシーでは、そのポリシーが関連付けられたバケットに対するアクセス権限を指定します。バケットポリシーは、 S3 PutBucketPolicy API を使用して設定します。を参照してください "バケットの処理"。

バケットポリシーを設定するには、 AWS CLI で次のコマンドを使用します。

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

例：すべてのユーザに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、匿名ユーザを含むすべてのユーザにバケット内のオブジェクトのリストとバケット内のすべてのオブジェクトの GET Object 処理を許可しています。それ以外の処理はすべて拒否されます。バケットへの書き込み権限がrootアカウント以外に付与されていないため、このポリシーは特に有用ではない場合があります。

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

例：あるアカウントのすべてのユーザにフルアクセスを許可し、別のアカウントのすべてのユーザに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、指定したアカウントのすべてのユーザにバケットへのフルアクセスを許可しています。さらに、アカウントをもう1つ指定し、そのアカウントのすべてのユーザには、で始まるバケットのオブジェクトのList処理とGetObject処理のみを許可しています `shared/` オブジェクトキープレフィックス。

> 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が、バケット所有者アカウントによって所有されます。バケットポリシーで、これらのオブジェクトの環境 を設定します。

```
{
  "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：すべてのユーザにバケットへの読み取り専用アクセスを許可し、指定したグループにフルアクセスを許可する

この例では、匿名ユーザを含むすべてのユーザにバケットのList処理とバケット内のすべてのオブジェクトのGET Object処理を許可し、グループに属するユーザのみを許可しています `Marketing` 指定したアカウントでは、フルアクセスが許可されています。

```
{
  "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：クライアントの **IP** 範囲を限定して、すべてのユーザにバケットへの読み取り **/** 書き込みアクセスを許可する

この例では、指定した IP 範囲（ 54.240.143.0~54.240.143.255 で 54.240.143.188 を除く）からの要求についてのみ、匿名ユーザを含むすべてのユーザにバケットの List 処理とバケット内のすべてのオブジェクトの全処理を許可しています。それ以外の処理はすべて拒否され、 IP 範囲外の要求はすべて拒否されます。

```
{
  "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：指定したフェデレーテッドユーザにのみバケットへのフルアクセスを許可します

この例では、フェデレーテッドユーザのAlexがへのフルアクセスを許可しています `examplebucket` バケットとそのオブジェクト。'root' を含む他のすべてのユーザは ' すべての操作を明示的に拒否されますただし、「 root 」による Put/Get/DeleteBucketPolicy は拒否されません。

```
{
  "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、を使用しています `Deny` PutOverwriteObjectとDeleteObjectの効果は、オブジェクトのデータ、ユーザ定義メタデータ、S3オブジェクトのタグを上書きまたは削除できないようにします。

```
{
  "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/*"
    }
  ]
}
```

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、そのポリシーが関連付けられたグループに対するアクセス権限を指定します。はいません `Principal` 要素は暗黙的であるため、ポリシーに含まれます。グループポリシーは Tenant Manager 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します

Tenant Managerでグループを追加または編集するときに、グループポリシーを選択して、このグループのメンバーに付与するS3アクセス権限を決定できます。を参照してください "S3 テナント用のグループを作成します"。

- * No S3 Access * ：デフォルトオプション。バケットポリシーでアクセスが許可されていないかぎり、このグループのユーザは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 root ユーザにのみ S3 リソースへのアクセスが許可されます。
- * 読み取り専用アクセス * ：このグループのユーザには、 S3 リソースへの読み取り専用アクセスが許可されます。たとえば、オブジェクトをリストして、オブジェクトデータ、メタデータ、タグを読み取ることができます。このオプションを選択すると、テキストボックスに読み取り専用グループポリシーの JSON 文字列が表示されます。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には、バケットを含む S3 リソースへのフルアクセスが許可されます。このオプションを選択すると、テキストボックスにフルアクセスグループポリシーの JSON 文字列が表示されます。この文字列は編集できません。
- ランサムウェアの軽減：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を環境 します。このグループのユーザは共通の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。

  Manage All Buckets権限を持つTenant Managerユーザは、このグループポリシーよりも優先できます。[すべてのバケットを管理]権限を信頼できるユーザに制限し、可能な場合は多要素認証（MFA）を使用します。

- * 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、バケットポリシーで明示的に拒否されている場合を除き、グループのすべてのメンバーにテナントアカウントが所有するすべてのバケットへのフル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、バケットポリシーで明示的に拒否されている場合を除き、グループのすべてのメンバーに S3 リソースへの読み取り専用アクセスが許可されます。たとえば、オブジェクトをリストして、オブジェクトデータ、メタデータ、タグを読み取ることができます。

```
{
  "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各自の「フォルダ」のみへのフルアクセスを許可します

この例では、指定したバケット内の特定のフォルダ（キープレフィックス）のリストおよびアクセスのみがグループのメンバーに許可されます。これらのフォルダのプライバシー設定を決めるときは、他のグループポリシーやバケットポリシーのアクセス権限を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}/*"
    }
  ]
}
```

REST API のセキュリティを設定する

REST API のセキュリティの実装を確認し、システムの保護方法について理解しておく必要があります。

StorageGRID がREST APIのセキュリティを提供する仕組み

StorageGRID システムで REST API のセキュリティ、認証、および許可がどのように実装されるかを理解しておく必要があります。

StorageGRID では、次のセキュリティ対策が使用されます。

- ロードバランサエンドポイントで HTTPS が設定されている場合は、ロードバランササービスとのクライアント通信に HTTPS が使用されます。

  ロードバランサエンドポイントを設定する際に、オプションで HTTP を有効にすることができます。たとえば、非本番環境でのテストなどに HTTP を使用できます。詳細については、 StorageGRID の管理手順を参照してください。

- StorageGRID は、ストレージノードとのクライアント通信にデフォルトでHTTPSを使用します。

  これらの接続に対して HTTP を有効にすることもできます。たとえば、非本番環境でのテストなどに HTTP を使用できます。詳細については、 StorageGRID の管理手順を参照してください。

- StorageGRID とクライアント間の通信は、 TLS を使用して暗号化されます。
- ロードバランササービスとグリッド内のストレージノードの間の通信は、ロードバランサエンドポイントが HTTP と HTTPS どちらの接続を受け入れるように設定されているかに関係なく暗号化されます。
- REST API 処理を実行するには、クライアントが StorageGRID に HTTP 認証ヘッダーを提供する必要があります。

セキュリティ証明書とクライアントアプリケーション

クライアントは、ゲートウェイノードまたは管理ノード上のロードバランササービスに、ストレージノードに直接接続できます。

いずれの場合も、クライアントアプリケーションは、グリッド管理者がアップロードしたカスタムサーバ証明書または StorageGRID システムが生成した証明書を使用して、 TLS 接続を確立できます。

- ロードバランササービスに接続する場合、クライアントアプリケーションは、接続に使用するロードバランサエンドポイント用に設定された証明書を使用します。各エンドポイントには独自の証明書があり、グリッド管理者がアップロードしたカスタムサーバ証明書か、グリッド管理者がエンドポイントの設定時に StorageGRID で生成した証明書のいずれかです。
- クライアントアプリケーションは、ストレージノードに直接接続する場合、StorageGRID システムのインストール時にストレージノード用に生成されたシステム生成のサーバ証明書（システム認証局によって署名されたもの）を使用します。 または、グリッド管理者がグリッド用に提供した単一のカスタムサーバ証明書。

TLS 接続の確立に使用する証明書に署名した認証局を信頼するよう、クライアントを設定する必要があります。

ロードバランサエンドポイントの設定に関する情報や、ストレージノードへの直接TLS接続に使用する単一のカスタムサーバ証明書を追加する手順については、StorageGRID の管理手順を参照してください。

まとめ

次の表に、 S3 および Swift の REST API におけるセキュリティの問題に対する実装を示します。

Security 問題 の略	**REST API** の実装
接続のセキュリティ	TLS
サーバ認証	システム CA によって署名された X.509 サーバ証明書、または管理者から提供されたカスタムサーバ証明書
クライアント認証	• S3 ： S3 アカウント（アクセスキー ID とシークレットアクセスキー） • Swift ： Swift アカウント（ユーザ名とパスワード）
クライアント許可	• S3 ：バケットの所有権と適用可能なすべてのアクセス制御ポリシー • Swift ：管理者ロールのアクセス

関連情報

"StorageGRID の管理"

TLS ライブラリのハッシュアルゴリズムと暗号化アルゴリズムがサポートされます

StorageGRID システムでは、クライアントアプリケーションが Transport Layer Security （ TLS ）セッションを確立する際に使用できる暗号スイートに制限があります。暗号を設定するには、**[設定]>*[セキュリティ設定]***に移動し、 TLSおよびSSHポリシー*を選択します。

サポートされる **TLS** のバージョン

StorageGRID では、 TLS 1.2 と TLS 1.3 がサポートされています。

SSLv3 と TLS 1.1 （またはそれ以前のバージョン）はサポートされなくなりました。

関連情報

"テナントアカウントと接続を設定する"

監視と監査の処理

オブジェクトの取り込み速度と読み出し速度を監視する

オブジェクトの取り込み速度と読み出し速度、およびオブジェクト数、クエリ、検証関連の指標を監視できます。StorageGRID システムのオブジェクトに対してクライアントアプリケーションが試みた読み取り、書き込み、変更の各処理について、成功した回数と失敗した回数を表示できます。

手順

1. を使用して Grid Manager にサインインします "サポートされている Web ブラウザ"。
2. ダッシュボードで、**[パフォーマンス]>*** S3処理]または[パフォーマンス]**>** Swift処理*を選択します。

   このセクションには、 StorageGRID システムによって実行されたクライアント処理の回数に関する概要が表示されます。プロトコル速度は過去 2 分間の平均値です。

3. [* nodes （ノード）] を選択します
4. ノードのホームページ（導入レベル）で、 * ロードバランサ * タブをクリックします。

   このグラフには、グリッド内でロードバランサエンドポイントに送信されるすべてのクライアントトラフィックの傾向が表示されます。時間、日、週、月、年単位の間隔を選択できます。 または、カスタムの間隔を適用することもできます。

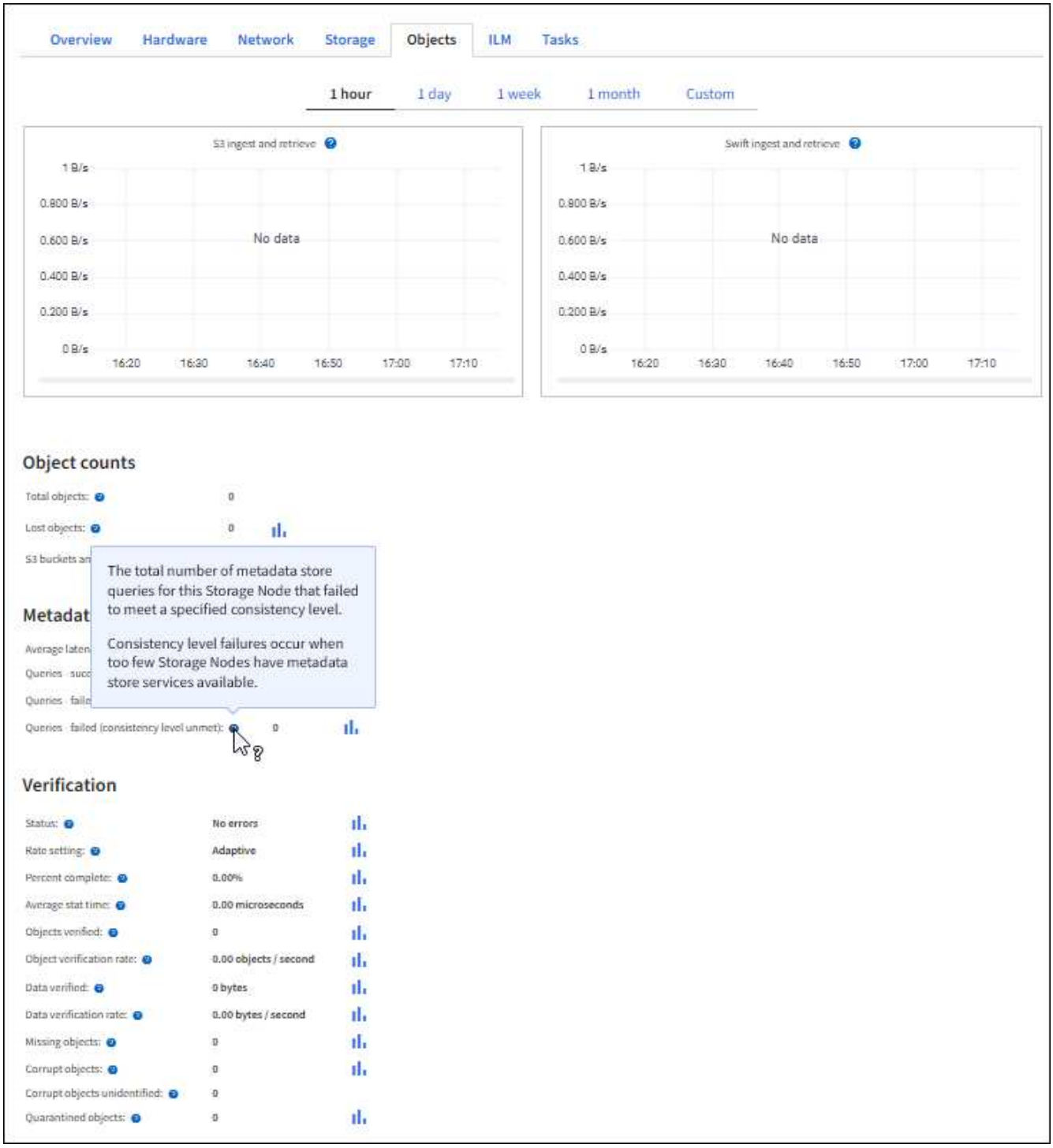
5. ノードのホームページ（導入レベル）で、 * Objects * タブをクリックします。

   グラフには、 StorageGRID システム全体の取り込み速度と読み出し速度が、 1 秒あたりのバイト数と合計バイト数で表示されます。時間、日、週、月、年単位の間隔を選択できます。 または、カスタムの間隔を適用することもできます。

6. 特定のストレージノードに関する情報を表示するには、左側のリストからノードを選択し、 * Objects * タ

ブをクリックします。

グラフには、このストレージノードのオブジェクトの取り込み速度と読み出し速度が表示されます。このタブには、オブジェクト数、クエリ、検証関連の指標も表示されます。ラベルをクリックすると、これらの指標の定義を確認できます。



7. さらに詳細な情報が必要な場合は、次の手順に従います
   a. サポート * > * ツール * > * グリッドトポロジ * を選択します。
   b. [**_site *>] > [** Overview **] > [** Main*] を選択します。

API Operations セクションには、グリッド全体の概要情報が表示されます。

c. 「 * _ ストレージノード _ * > * LDR * > * _ クライアントアプリケーション _ * > * 概要 * > * Main * 」を選択します

Operations セクションには、選択したストレージノードに関する概要情報が表示されます。

監査ログにアクセスして確認する

監査メッセージは StorageGRID サービスによって生成され、テキスト形式のログファイルに保存されます。監査ログの API 固有の監査メッセージにより、セキュリティ、運用、およびパフォーマンスについて、システムの健全性の評価に役立つ重要な監視データが提供されます。

作業を開始する前に

- 特定のアクセス権限が必要です。
- を使用することができます `Passwords.txt` ファイル。
- 管理ノードの IP アドレスを確認しておきます。

このタスクについて

アクティブな監査ログファイルの名前はです `audit.log`をクリックし、を管理ノードに格納します。

1日に1回、アクティブなaudit.logファイルが保存され、新しいファイルが作成されます `audit.log` ファイルが開始されました。保存されたファイルの名前は、保存された日時をの形式で示しています *`yyyy-mm-dd.txt`*。

1日後、保存されたファイルは圧縮され、という形式で名前が変更されます `*yyyy-mm-dd.txt.gz*`元の日付を保持します。

この例は、アクティブを示しています `audit.log` ファイル。前日のファイルです (`2018-04-15.txt`）、および前日の圧縮ファイルです (`2018-04-14.txt.gz`）。

```
audit.log
2018-04-15.txt
2018-04-14.txt.gz
```

手順

1. 管理ノードにログインします。
   a. 次のコマンドを入力します。 `ssh admin@primary_Admin_Node_IP`
   b. に記載されているパスワードを入力します `Passwords.txt` ファイル。
   c. 次のコマンドを入力してrootに切り替えます。 `su -`
   d. に記載されているパスワードを入力します `Passwords.txt` ファイル。

   rootとしてログインすると、プロンプトがから変わります `$` 終了： `#`。

2. 監査ログファイルが保存されているディレクトリに移動します。

```
cd /var/local/audit/export
```

3. 必要に応じて、現在の監査ログファイルまたは保存された監査ログファイルを表示します。

監査ログで追跡される **S3** 処理

バケットおよびオブジェクトのいくつかの処理は、 StorageGRID の監査ログで追跡されます。

監査ログで追跡されるバケットの処理

- バケットを削除します
- バケットのタグ付けを削除します
- 複数のオブジェクトを削除します
- GET Bucket （ List Objects ）
- GET Bucket Object versions
- GET Bucket tagging
- HEAD Bucket （ヘッドバケット）
- PUT Bucket の場合
- PUT Bucket compliance で確認してください
- PUT Bucket tagging
- PUT Bucket versioning の場合

監査ログで追跡されるオブジェクトの処理

- Complete Multipart Upload の実行
- Upload Part（ILMルールの取り込み動作がBalancedまたはStrictの場合）
- Upload Part - Copy（ILMルールの取り込み動作がBalancedまたはStrictの場合）
- オブジェクトを削除します
- オブジェクトの取得
- HEAD Object の実行
- POST Object restore の実行
- PUT Object の場合
- PUT Object - Copy の各コマンドを実行します

関連情報

"バケットの処理"

"オブジェクトの処理"

アクティブ、アイドル、および同時 **HTTP** 接続のメリット

StorageGRID システムのパフォーマンスに影響するのは、 HTTP 接続の設定方法です。設定は、 HTTP 接続がアクティブであるかアイドルであるか、同時に複数の接続を使用するかによって異なります。

次の種類の HTTP 接続について、パフォーマンスのメリットを特定することができます。

- アイドル HTTP 接続
- アクティブ HTTP 接続
- 同時 HTTP 接続

アイドル **HTTP** 接続を開いておくメリット

クライアントアプリケーションがアイドル状態のときも HTTP 接続を開いておくと、クライアントアプリケーションで以降のトランザクションが発生したときに、それらの開いている接続を使用して実行することができます。ネットアップでは、アイドル HTTP 接続を開いておく時間を 10 分までにすることを推奨します。HTTP 接続をアイドル状態のまま 10 分以上開いていると、 StorageGRID によって自動的に閉じられることがあります。

アイドル HTTP 接続を開いておくと、次のようなメリットがあります。

- HTTP トランザクションの実行が StorageGRID 必要と判断されてから StorageGRID システムでトランザクションが実行されるまでのレイテンシが短縮されます

  レイテンシの短縮は、特に TCP / IP 接続と TLS 接続の確立に時間がかかる場合に大きなメリットとなります。

- 実行済みの転送が増えるにしたがって TCP / IP のスロースタートアルゴリズムによってデータ転送速度が向上します
- クライアントアプリケーションと StorageGRID システムの間の接続が中断された、複数の障害状況の瞬時通知

アイドル接続を開いておく適切な時間は、既存の接続のスロースタートから得られるメリットと、内部システムリソースへの理想的な接続の割り当てとのバランスによって決まります。

アクティブ **HTTP** 接続のメリット

ストレージノードに直接接続する場合は、HTTP接続でトランザクションを継続的に実行する場合でも、アクティブHTTP接続の継続時間を10分に制限する必要があります。

接続を開いておく最大継続時間は、接続を維持することで得られるメリットと内部システムリソースへの理想的な接続の割り当てとのバランスによって決まります。

ストレージノードへのクライアント接続でアクティブHTTP接続を制限すると、次のようなメリットがあります。

- StorageGRID システム全体で負荷を最適に分散できます。

  時間の経過とともに負荷分散の要件が変わったため、 HTTP 接続が最適な状態でなくなることがありま

す。クライアントアプリケーションでトランザクションごとに別の HTTP 接続を確立すれば、システムによる負荷分散は最適になりますが、この場合、接続を維持することで得られるより大きなメリットを失うことになります。

- クライアントアプリケーションからの HTTP トランザクションを使用可能な空きスペースがある LDR サービスに転送できる
- メンテナンス手順を開始できます。

  メンテナンス手順の中には、実行中のすべての HTTP 接続が完了してからでないと開始されないものがあります。

ロードバランササービスへのクライアント接続では、接続時間を制限することで一部のメンテナンス手順をすぐに開始できます。クライアント接続の時間が制限されていない場合、アクティブな接続が自動的に終了するまでに数分かかることがあります。

同時 **HTTP** 接続のメリット

StorageGRID システムへの TCP / IP 接続を複数開いて並列処理を可能にしておくと、パフォーマンスが向上します。最適な並列接続数は、さまざまな要因によって異なります。

同時 HTTP 接続には、次のようなメリットがあります。

- レイテンシが短縮されます

  他のトランザクションが完了するのを待たずに、トランザクションをすぐに開始できます。

- スループットの向上

  StorageGRID システムでは、トランザクションの並列処理が可能なため、全体的なトランザクションのスループットが向上します。

クライアントアプリケーションで複数の HTTP 接続を確立する必要があります。クライアントアプリケーションでトランザクションの実行が必要になったときは、確立された接続の中からトランザクションの処理に現在使用されていない接続を選択してすぐに使用することができます。

同時トランザクションや同時接続の最大スループットは StorageGRID システムのトポロジごとに異なり、それを超えるとパフォーマンスが低下し始めます。最大スループットは、コンピューティングリソース、ネットワークリソース、ストレージリソース、 WAN リンクなどの要因によって決まります。また、サーバやサービスの数、 StorageGRID システムでサポートするアプリケーションの数も影響します。

StorageGRID システムでは、複数のクライアントアプリケーションをサポートすることがよくありますクライアントアプリケーションで使用する同時接続の最大数を決定する場合は、この点に注意してください。クライアントアプリケーションを構成する複数のソフトウェアエンティティのそれぞれで StorageGRID システムへの接続を確立する場合は、それらのエンティティのすべての接続を合計して考慮する必要があります。次のような場合は、同時接続の最大数の調整が必要になることがあります。

- StorageGRID システムのトポロジによって、システムでサポートできる同時トランザクションや同時接続の最大数が異なります。
- クライアントアプリケーションがネットワークの限られた帯域幅で StorageGRID システムと通信する場合は、個々のトランザクションが妥当な時間で完了するように、必要に応じて同時実行の数を少なくします。

- 多くのクライアントアプリケーションで StorageGRID システムを共有する場合は、システムの制限を超えないように、同時実行の数を少なくする必要があります。

読み取り処理用と書き込み処理用に別々の **HTTP** 接続プールを使用する

読み取り処理と書き込み処理に別々の HTTP 接続プールを使用して、それぞれに使用するプールの容量を制御できます。HTTP 接続のプールを分けることで、トランザクションや負荷分散をより細かく制御できます。

クライアントアプリケーションで生成される負荷には、読み出し中心（読み取り）の負荷と格納中心（書き込み）の負荷があります。読み取りと書き込みで HTTP 接続プールを分けることで、各プールの量を調整してそれぞれのトランザクション専用に使用することができます。