Swift REST APIを使用する

StorageGRID 11.5

NetApp
April 11, 2024

目次

Swift を使用します

クライアントアプリケーションでOpenStack Swift APIを使用して、StorageGRID システムを操作する方法について説明します。

- "StorageGRID でのOpenStack Swift APIのサポート"
- "テナントアカウントと接続を設定する"
- "Swift REST API でサポートされている処理"
- "StorageGRID の Swift REST API 処理"
- "REST APIのセキュリティの設定"
- "処理の監視と監査"

StorageGRID でのOpenStack Swift APIのサポート

StorageGRID でサポートしている Swift 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
Swift の仕様	2015 年 11 月時点の OpenStack Swift Object Storage API v1
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

"OpenStack ：オブジェクトストレージ API"

StorageGRID での Swift API サポートの履歴

StorageGRID システムでの Swift REST API のサポートに関する変更点に注意する必要があります。

リリース。	コメント
11.5	弱い整合性制御を削除しました。代わりに、available 整合性レベルが使用されます。
11.4	TLS 1.3 のサポートの追加と、サポートされる TLS 暗号スイートのリストの更新CLB は廃止されました。ILM と整合性設定の間の相互関係の概要 が追加されました。

リリース。	コメント
11.3	PUT Object 処理が更新され、取り込み時に同期配置を使用する ILM ルールの影響（取り込み動作の Balanced オプションと Strict オプション）が記述されるようになりました。ロードバランサエンドポイントまたはハイアベイラビリティグループを使用するクライアント接続の概要 が追加されました。サポートされる TLS 暗号スイートのリストが更新されました。TLS 1.1 暗号はサポートされなくなりました。
11.2	ドキュメントに対する編集上の変更がいくつかあります。
11.1	グリッドノードへの Swift クライアント接続での HTTP の使用のサポートが追加されました。整合性制御の定義が更新されました。
11.0	テナントアカウントにつき 1 、 000 個のコンテナのサポートが追加されました。
10.3	ドキュメントの管理に関する記述の更新と修正カスタムサーバ証明書の設定に関するセクションが削除されました。
10.2	StorageGRID システムで Swift API が初めてサポートされました。現在サポートされているバージョンは、 OpenStack Swift Object Storage API v1 です。

StorageGRID でのSwift REST APIの実装

クライアントアプリケーションは、 Swift REST API 呼び出しを使用してストレージノードやゲートウェイノードに接続し、コンテナの作成やオブジェクトの格納と読み出しを行うことができます。これを利用して、 OpenStack Swift 向けに開発されたサービス指向アプリケーションを、 StorageGRID システムで利用できるオンプレミスのオブジェクトストレージに接続することができます。

Swift オブジェクトの管理

StorageGRID システムに取り込まれた Swift オブジェクトは、システムのアクティブな ILM ポリシー内の情報ライフサイクル管理（ ILM ）ルールによって管理されます。ILM ルールとポリシーは、 StorageGRID がオブジェクトデータのコピーを作成および分散し、一定の期間にわたって管理する方法を決定します。たとえば、 ILM ルールを特定の Swift コンテナ内のオブジェクトに適用し、複数のオブジェクトコピーを複数のデータセンターに一定期間保存するように指定できます。

グリッドの ILM ルールとポリシーが Swift テナントアカウントのオブジェクトに与える影響については、 StorageGRID 管理者にお問い合わせください。

競合するクライアント要求です

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、Swift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。

整合性の保証と制御

デフォルトでは、 StorageGRID は、新規作成された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と HEAD 処理には結果整合性を提供します。正常に完了した PUT に続く GET では、新しく書き込まれたデータを読み取ることができます。既存のオブジェクトの上書き、メタデータの更新、および削除の整合性レベルは、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。

StorageGRID では、コンテナごとに整合性を制御することもできます。アプリケーションでの必要に応じて、ストレージノード間およびサイト間でオブジェクトの可用性と整合性のトレードオフを行うために、整合性制御を変更できます。

関連情報

"ILM を使用してオブジェクトを管理する"

"GET コンテナセイコウセイヨウキユウ"

"PUT コンテナセイコウセイヨウキユウ"

Swift REST APIを実装する際の推奨事項

StorageGRID で使用するために Swift 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しないと思われるパスにオブジェクトが存在するかどうかをアプリケーションが定期的にチェックする場合は ' 使用可能な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ロケーションに対して PUT 操作を実行する前に ' そのロケーションに対して HEAD 操作を実行する場合は ' Available 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 コンテナ整合性要求を使用して、各コンテナに「 available 」整合性制御を設定できます。

オブジェクト名の推奨事項

オブジェクト名の最初の 4 文字に、ランダムな値を使用しないでください。代わりに、イメージなど、ランダムで一意でないプレフィックスを使用してください。

オブジェクト名のプレフィックスにランダムな一意の文字を使用する必要がある場合は、オブジェクト名の前にディレクトリ名を指定してください。つまり、次の形式を使用します。

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

次の形式は使用しないでください。

```
mycontainer/f8e3-image3132.jpg
```

「範囲の読み取り」に関する推奨事項

「格納オブジェクトの圧縮」オプション（* Configuration > System Settings > Grid Options *）を選択した場合、Swiftクライアントアプリケーションでは、バイト範囲を指定したGET object処理を実行しないでください。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常に非効率的です。

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

"GET コンテナセイコウセイヨウキユウ"

"PUT コンテナセイコウセイヨウキユウ"

"StorageGRID の管理"

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。

Swiftテナントアカウントを作成および設定します

Swift API クライアントで StorageGRID に対してオブジェクトの格納や読み出しを行うには、 Swift 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、専用のアカウント ID 、専用のグループとユーザ、および専用のコンテナとオブジェクトがあります。

Swift 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。

グリッド管理者は、 Swift テナントアカウントを作成する際に次の情報を指定します。

- テナントの表示名（テナントのアカウント ID は自動的に割り当てられ、変更できません）
- 必要に応じて、テナントアカウントのストレージクォータ — テナントのオブジェクトで使用可能な最大ギガバイト数、テラバイト数、ペタバイト数。テナントのストレージクォータは、物理容量（ディスクの

サイズ）ではなく、論理容量（オブジェクトのサイズ）を表します。

- StorageGRID システムでシングルサインオン（ SSO ）が使用されていない場合は、テナントアカウントが独自のアイデンティティソースを使用するか、グリッドのアイデンティティソースを共有するか、およびテナントのローカル root ユーザの初期パスワード。
- SSO が有効になっている場合は、テナントアカウントを設定するための Root Access 権限が割り当てられているフェデレーテッドグループ。

Swift テナントアカウントが作成されたら、 Root Access 権限を持つユーザは Tenant Manager にアクセスして、次のようなタスクを実行できます。

- アイデンティティフェデレーションの設定（グリッドとアイデンティティソースを共有する場合を除く）、およびローカルグループとユーザの作成
- ストレージ使用状況を監視しています

> Swift ユーザが Tenant Manager にアクセスするには、 Root Access 権限が必要です。ただし Root Access 権限では、 Swift REST API に認証してコンテナを作成したりオブジェクトを取り込んだりすることはできません。Swift REST API に認証するには、 Swift 管理者の権限が必要です。

関連情報

"StorageGRID の管理"

"テナントアカウントを使用する"

"サポートされている Swift API エンドポイント"

クライアント接続の設定方法

グリッド管理者は、 Swift 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。接続するために必要な具体的な情報は、選択した設定によって異なります。

クライアントアプリケーションは、次のいずれかに接続することで、オブジェクトを格納または読み出すことができます。

- 管理ノードまたはゲートウェイノード上のロードバランササービス、または必要に応じて、管理ノードまたはゲートウェイノードのハイアベイラビリティ（ HA ）グループの仮想 IP アドレス
- ゲートウェイノード上の CLB サービス、または必要に応じて、ゲートウェイノードのハイアベイラビリティグループの仮想 IP アドレス

> CLB サービスは廃止されました。StorageGRID 11.3 より前に設定されたクライアントは、ゲートウェイノード上の CLB サービスを引き続き使用できます。ロードバランシングに StorageGRID を使用する他のすべてのクライアントアプリケーションは、ロードバランササービスを使用して接続する必要があります。

- 外部ロードバランサを使用するかどうかに関係なく、ストレージノードに追加されます

StorageGRID を設定する場合、グリッド管理者はグリッドマネージャまたはグリッド管理 API を使用して次の手順を実行できます。これらはすべてオプションです。

1. ロードバランササービスのエンドポイントを設定する。

   ロードバランササービスを使用するようにエンドポイントを設定する必要があります。管理ノードまたはゲートウェイノード上のロードバランササービスは、クライアントアプリケーションからの受信ネットワーク接続を複数のストレージノードに分散します。ロードバランサエンドポイントを作成する際、StorageGRID 管理者は、ポート番号、エンドポイントで HTTP / HTTPS 接続を許可するかどうか、エンドポイントを使用するクライアントのタイプ（ S3 または Swift ）、 HTTPS 接続に使用する証明書（該当する場合）を指定します。

2. 信頼されていないクライアントネットワークを設定する

   StorageGRID 管理者がノードのクライアントネットワークを信頼されていないクライアントネットワークとして設定した場合、ノードはロードバランサエンドポイントとして明示的に設定されたポートでクライアントネットワークのインバウンド接続だけを受け入れます。

3. ハイアベイラビリティグループを設定する。

   管理者が HA グループを作成すると、複数の管理ノードまたはゲートウェイノードのネットワークインターフェイスがアクティブ / バックアップ構成になります。クライアント接続は、 HA グループの仮想 IP アドレスを使用して確立されます。

各オプションの詳細については、 StorageGRID の管理手順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

クライアントアプリケーションは、グリッドノードの IP アドレスおよびそのノード上のサービスのポート番号を使用して StorageGRID に接続します。ハイアベイラビリティ（ HA ）グループが設定されている場合は、 HA グループの仮想 IP アドレスを使用してクライアントアプリケーションを接続できます。

クライアント接続に必要な情報

次の表に、クライアントが StorageGRID に接続できるさまざまな方法、および各接続タイプで使用される IP アドレスとポートを示します。詳細については、 StorageGRID 管理者にお問い合わせください。または、StorageGRID for a 概要 の管理手順を参照して、グリッドマネージャでこの情報を確認してください。

接続が確立される場所	クライアントが接続するサービス	**IP** アドレス	ポート
HA グループ	ロードバランサ	HA グループの仮想 IP アドレス	• ロードバランサエンドポイントのポート
HA グループ	CLB の機能です • 注： * CLB サービスは廃止されました。	HA グループの仮想 IP アドレス	デフォルトの Swift ポート： • HTTPS ： 8083 • HTTP ： 8085
管理ノード	ロードバランサ	管理ノードの IP アドレス	• ロードバランサエンドポイントのポート

接続が確立される場所	クライアントが接続するサービス	**IP** アドレス	ポート
ゲートウェイノード	ロードバランサ	ゲートウェイノードの IP アドレス	• ロードバランサエンドポイントのポート
ゲートウェイノード	CLB の機能です • 注： * CLB サービスは廃止されました。	ゲートウェイノードの IP アドレス • 注：デフォルトでは、 CLB および LDR の HTTP ポートは有効になっていません。	デフォルトの Swift ポート： • HTTPS ： 8083 • HTTP ： 8085
ストレージノード	LDR	ストレージノードの IP アドレス	デフォルトの Swift ポート： • HTTPS ： 18083 • HTTP ： 18085

例

Swift クライアントをゲートウェイノードの HA グループのロードバランサエンドポイントに接続するには、次のように構造化された URL を使用します。

- `https://VIP-of-HA-group:LB-endpoint-port`

たとえば、 HA グループの仮想 IP アドレスが 192.0.2.6 で、 Swift ロードバランサエンドポイントのポート番号が 10444 の場合、 Swift クライアントは次の URL を使用して StorageGRID に接続できます。

- `https://192.0.2.6:10444`

クライアントが StorageGRID への接続に使用する IP アドレスに DNS 名を設定できます。ローカルネットワーク管理者にお問い合わせください。

HTTPS接続と**HTTP**接続のどちらを使用するかの判断

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またはゲートウェイノード上の CLB サービスへのクライアント接続に HTTP を使用する場合は、 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またはゲートウェイノード上の CLB 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 * Enable HTTP Connection * grid オプションを選択して、セキュアでない 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してください。

CLB サービスは廃止されました。

[Enable HTTP Connection*] オプションが選択されている場合、クライアントは HTTPS とは異なるポートを HTTP に使用する必要があります。StorageGRID の管理手順を参照してください。

関連情報

"StorageGRID の管理"

Swift API設定で接続をテストします

Swift の CLI を使用して、 StorageGRID 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。

必要なもの

- Swift のコマンドラインクライアント python-swiftclient をダウンロードしてインストールしておく必要があります。
- StorageGRID システムに Swift テナントアカウントが必要です。

このタスクについて

セキュリティを設定していない場合は、を追加する必要があります `--insecure` これらの各コマンドにフラグを設定します。

手順

1. StorageGRID Swift 環境の情報 URL を照会します。

```
swift
-U <Tenant_Account_ID:Account_User_Name>
-K <User_Password>
-A https://<FQDN | IP>:<Port>/info
capabilities
```

この手順で、 Swift 環境が機能することをテストできます。オブジェクトを格納してアカウント設定をさらにテストするには、以降の手順を実行します。

2. オブジェクトをコンテナに配置します。

```
touch test_object
swift
-U <Tenant_Account_ID:Account_User_Name>
-K <User_Password>
-A https://<FQDN | IP>:<Port>/auth/v1.0
upload test_container test_object
--object-name test_object
```

3. コンテナを取得してオブジェクトを確認します。

```
swift
-U <Tenant_Account_ID:Account_User_Name>
-K <User_Password>
-A https://<FQDN | IP>:<Port>/auth/v1.0
list test_container
```

4. オブジェクトを削除します。

```
swift
-U <Tenant_Account_ID:Account_User_Name>
-K <User_Password>
-A https://<FQDN | IP>:<Port>/auth/v1.0
delete test_container test_object
```

5. コンテナを削除します。

```
swift
-U `<_Tenant_Account_ID:Account_User_Name_>`
-K `<_User_Password_>`
-A `\https://<_FQDN_ | _IP_>:<_Port_>/auth/v1.0'
delete test_container
```

関連情報

"Swiftテナントアカウントを作成および設定します"

"REST APIのセキュリティの設定"

Swift REST API でサポートされている処理

StorageGRID システムは、 OpenStack Swift API のほとんどの処理をサポートしています。Swift REST API クライアントを StorageGRID に統合する前に、アカウント、コン

テナ、およびオブジェクトの処理の実装に関する詳細を確認します。

StorageGRID でサポートされている操作

次の Swift API 処理がサポートされています。

- "アカウントの処理"
- "コンテナの処理"
- "オブジェクトの処理"

すべての処理に共通の応答ヘッダー

StorageGRID システムでは、 OpenStack Swift Object Storage API v1 の定義に従って、サポートされるすべての処理に共通のヘッダーが実装されます。

関連情報

"OpenStack ：オブジェクトストレージ API"

サポートされている Swift API エンドポイント

StorageGRID でサポートされている Swift API エンドポイントは、情報 URL 、認証 URL 、およびストレージ URL です。

情報 URL

StorageGRID Swift 実装の機能と制限事項については、 Swift のベース URL に /info パスを付加して GET 要求を発行することで確認できます。

```
https://FQDN | Node IP:Swift Port/info/
```

要求の内容は次のとおりです。

- *FQDN* は完全修飾ドメイン名です。
- *Node IP* は、StorageGRID ネットワークのストレージノードまたはゲートウェイノードのIPアドレスです。
- *Swift Port* は、ストレージノードまたはゲートウェイノードのSwift API接続に使用するポート番号です。

たとえば、次の情報 URL は、 IP アドレスが 10.99.106.103 でポート 18083 を使用しているストレージノードから情報を要求します。

```
https://10.99.106.103:18083/info/
```

応答には、 Swift 実装の機能が JSON ディクショナリとして含まれます。クライアントツールは、 JSON 応答を解析して実装の機能を特定し、後続のストレージ処理で制約として使用できます。

StorageGRID 実装の Swift では、情報 URL への認証されていないアクセスが許可されます。

認証 URL

クライアントは、 Swift 認証 URL を使用してテナントアカウントユーザとして認証できます。

```
https://FQDN | Node IP:Swift Port/auth/v1.0/
```

で、テナントアカウントID、ユーザ名、およびパスワードをパラメータとして指定する必要があります `X-Auth-User` および `X-Auth-Key` 次のように要求ヘッダー

```
X-Auth-User: Tenant_Account_ID:Username
```

```
X-Auth-Key: Password
```

要求ヘッダーは次のようになります。

- *`Tenant_Account_ID`* は、Swiftテナントの作成時にStorageGRID によって割り当てられたアカウントIDです。Tenant Manager のサインインページで使用するテナントアカウント ID と同じです。
- *`Username`* は、Tenant Managerで作成されたテナントユーザの名前です。このユーザは、 Swift 管理者権限を持つグループに属している必要があります。テナントの root ユーザを、 Swift REST API を使用するように設定することはできません。

  テナントアカウントに対してアイデンティティフェデレーションが有効になっている場合は、 LDAP サーバからのフェデレーテッドユーザのユーザ名とパスワードを指定します。または、 LDAP ユーザのドメイン名を指定します。例：

  ```
  X-Auth-User: Tenant_Account_ID:Username@Domain_Name
  ```

- *`Password`* は、テナントユーザのパスワードです。ユーザパスワードは Tenant Manager で作成および管理します。

認証要求が成功すると、ストレージ URL と認証トークンが次のように返されます。

```
X-Storage-Url: https://FQDN | Node_IP:Swift_Port/v1/Tenant_Account_ID
```

```
X-Auth-Token: token
```

```
X-Storage-Token: token
```

デフォルトでは、トークンの有効期間は生成時刻から 24 時間です。

トークンは特定のテナントアカウントに対して生成されます。あるアカウントに対して有効なトークンで、別のアカウントにアクセスするユーザを許可することはできません。

ストレージ URL

クライアントアプリケーションは、ゲートウェイノードまたはストレージノードに対して、問題 の Swift REST API 呼び出しを使用して、アカウント、コンテナ、オブジェクトのサポートされる処理を実行できます。ストレージ要求は、認証応答で返されたストレージ URL にアドレスが指定されます。要求には、認証要求から返された X-Auth-Token ヘッダーと値も含める必要があります。

```
https://FQDN | IP:Swift_Port/v1/Tenant_Account_ID
```

`[/container][/object]`

`X-Auth-Token: token`

使用状況の統計が含まれるストレージ応答ヘッダーに、最近変更されたオブジェクトの正確な数が反映されない場合があります。このヘッダーに正確な数値が表示されるまでに数分かかることがあります。

使用状況の統計が含まれているアカウントおよびコンテナ処理の応答ヘッダーの例を次に示します。

- `X-Account-Bytes-Used`
- `X-Account-Object-Count`
- `X-Container-Bytes-Used`
- `X-Container-Object-Count`

関連情報

"クライアント接続の設定方法"

"Swiftテナントアカウントを作成および設定します"

"アカウントの処理"

"コンテナの処理"

"オブジェクトの処理"

アカウントの処理

アカウントに対して実行する Swift API 処理を次に示します。

GET アカウント

この処理は、アカウントに関連付けられているコンテナリストおよびアカウントの使用状況を示す統計を取得します。

次の要求パラメータが必要です。

- `Account`

次の要求ヘッダーが必要です。

- `X-Auth-Token`

次のサポートされている要求クエリパラメータはオプションです。

- `Delimiter`
- `End_marker`
- `Format`
- `Limit`

- Marker
- Prefix

実行が成功すると ' アカウントが見つかってコンテナがないかコンテナリストが空である場合 ' またはアカウントが見つかってコンテナリストが空でない場合には 'HTTP/1.1 204 No Content' の応答とともに ' 次のヘッダーが返され ' コンテナリストが空でない場合は 'HTTP/1.1 200 OK' の応答が返されます

- Accept-Ranges
- Content-Length
- Content-Type
- Date
- X-Account-Bytes-Used
- X-Account-Container-Count
- X-Account-Object-Count
- X-Timestamp
- X-Trans-Id

HEAD アカウント

この処理は、 Swift アカウントからアカウント情報と統計情報を取得します。

次の要求パラメータが必要です。

- Account

次の要求ヘッダーが必要です。

- X-Auth-Token

実行が成功すると、「 HTTP/1.1 204 No Content 」の応答とともに次のヘッダーが返されます。

- Accept-Ranges
- Content-Length
- Date
- X-Account-Bytes-Used
- X-Account-Container-Count
- X-Account-Object-Count
- X-Timestamp
- X-Trans-Id

関連情報

"監査ログで追跡される Swift 処理"

コンテナの処理

StorageGRID では、 Swift アカウントあたり最大で 1 、 000 個のコンテナがサポートされます。コンテナに対して実行する Swift API 処理を次に示します。

コンテナを削除します

この処理は、 StorageGRID システムの Swift アカウントから空のコンテナを削除します。

次の要求パラメータが必要です。

- `Account`
- `Container`

次の要求ヘッダーが必要です。

- `X-Auth-Token`

実行が成功すると、「 HTTP/1.1 204 No Content 」の応答とともに次のヘッダーが返されます。

- `Content-Length`
- `Content-Type`
- `Date`
- `X-Trans-Id`

GET コンテナ

この処理は、コンテナに関連付けられているオブジェクトリストを、 StorageGRID システム内のコンテナの統計情報およびメタデータとともに読み出します。

次の要求パラメータが必要です。

- `Account`
- `Container`

次の要求ヘッダーが必要です。

- `X-Auth-Token`

次のサポートされている要求クエリパラメータはオプションです。

- `Delimiter`
- `End_marker`
- `Format`
- `Limit`
- `Marker`

- `Path`
- `Prefix`

実行が成功すると、「 HTTP/1.1 200 Success 」または「 HTTP/1.1 204 No Content 」の応答とともに次のヘッダーが返されます。

- `Accept-Ranges`
- `Content-Length`
- `Content-Type`
- `Date`
- `X-Container-Bytes-Used`
- `X-Container-Object-Count`
- `X-Timestamp`
- `X-Trans-Id`

HEAD コンテナ

この処理は、 StorageGRID システムからコンテナの統計情報とメタデータを読み出します。

次の要求パラメータが必要です。

- `Account`
- `Container`

次の要求ヘッダーが必要です。

- `X-Auth-Token`

実行が成功すると、「 HTTP/1.1 204 No Content 」の応答とともに次のヘッダーが返されます。

- `Accept-Ranges`
- `Content-Length`
- `Date`
- `X-Container-Bytes-Used`
- `X-Container-Object-Count`
- `X-Timestamp`
- `X-Trans-Id`

PUT コンテナ

この処理は、 StorageGRID システムのアカウントにコンテナを作成します。

次の要求パラメータが必要です。

- `Account`
- `Container`

次の要求ヘッダーが必要です。

- `X-Auth-Token`

実行が成功すると、「 HTTP/1.1 201 Created 」または「 HTTP/1.1 202 Accepted 」の応答（このアカウントにコンテナがすでに存在する場合）とともに次のヘッダーが返されます。

- `Content-Length`
- `Date`
- `X-Timestamp`
- `X-Trans-Id`

コンテナ名は StorageGRID ネームスペース内で一意である必要があります。このコンテナが別のアカウントの下に存在する場合は、ヘッダー「 HTTP/1.1 409 Conflict 」が返されます。

関連情報

"監査ログで追跡される Swift 処理"

オブジェクトの処理

オブジェクトに対して実行する Swift API 処理を次に示します。

オブジェクトを削除します

この処理は、オブジェクトのコンテンツとメタデータを StorageGRID システムから削除します。

次の要求パラメータが必要です。

- `Account`
- `Container`
- `Object`

次の要求ヘッダーが必要です。

- `X-Auth-Token`

実行が成功すると、が指定された次の応答ヘッダーが返されます `HTTP/1.1 204 No Content` 対応：

- `Content-Length`
- `Content-Type`
- `Date`
- `X-Trans-Id`

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かった場合（格納場所が一時的に使用不能などの理由で）、StorageGRID は削除対象のコピーをキューに登録し、クライアントに処理が成功したことを通知します。

オブジェクトの削除方法の詳細については、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してください。

GET オブジェクト

この処理は、 StorageGRID から、オブジェクトのコンテンツを読み出し、オブジェクトメタデータを取得します。

次の要求パラメータが必要です。

- `Account`
- `Container`
- `Object`

次の要求ヘッダーが必要です。

- `X-Auth-Token`

次の要求ヘッダーはオプションです。

- `Accept-Encoding`
- `If-Match`
- `If-Modified-Since`
- `If-None-Match`
- `If-Unmodified-Since`
- `Range`

実行が成功すると、が指定された次のヘッダーが返されます `HTTP/1.1 200 OK` 対応：

- `Accept-Ranges`
- `Content-Disposition`の場合にのみ返されます `Content-Disposition` メタデータが設定されました
- `Content-Encoding`の場合にのみ返されます `Content-Encoding` メタデータが設定されました
- `Content-Length`
- `Content-Type`
- `Date`
- `ETag`
- `Last-Modified`
- `X-Timestamp`

- `X-Trans-Id`

HEAD オブジェクト

この処理は、取り込まれたオブジェクトのメタデータとプロパティを StorageGRID システムから読み出します。

次の要求パラメータが必要です。

- `Account`
- `Container`
- `Object`

次の要求ヘッダーが必要です。

- `X-Auth-Token`

実行が成功すると、「 HTTP/1.1 200 OK 」の応答とともに次のヘッダーが返されます。

- `Accept-Ranges`
- `Content-Disposition`の場合にのみ返されます `Content-Disposition` メタデータが設定されました
- `Content-Encoding`の場合にのみ返されます `Content-Encoding` メタデータが設定されました
- `Content-Length`
- `Content-Type`
- `Date`
- `ETag`
- `Last-Modified`
- `X-Timestamp`
- `X-Trans-Id`

PUT オブジェクト

この処理は、 StorageGRID システムで、データとメタデータを含む新しいオブジェクトを作成するか、データとメタデータを含む既存のオブジェクトを置換します。

StorageGRID は、サイズが最大5TBのオブジェクトをサポートします。

> 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、Swift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。

次の要求パラメータが必要です。

- `Account`

- `Container`
- `Object`

次の要求ヘッダーが必要です。

- `X-Auth-Token`

次の要求ヘッダーはオプションです。

- `Content-Disposition`
- `Content-Encoding`

  チャンクを使用しないでください `Content-Encoding` 環境 オブジェクトがサイズに基づいてオブジェクトをフィルタリングし、取り込み時に同期配置を使用するILMルール（取り込み動作にBalancedオプションまたはStrictオプション）の場合。

- `Transfer-Encoding`

  圧縮またはチャンクを使用しないでください `Transfer-Encoding` 環境 オブジェクトがサイズに基づいてオブジェクトをフィルタリングし、取り込み時に同期配置を使用するILMルール（取り込み動作にBalancedオプションまたはStrictオプション）の場合。

- `Content-Length`

  ILMルールで、オブジェクトがサイズでフィルタリングされ、取り込み時に同期配置が使用される場合は、を指定する必要があります `Content-Length`。

  > でこれらのガイドラインに従っていない場合は、を参照してください `Content-Encoding`、 `Transfer-Encoding`および `Content-Length`ではStorageGRID 、オブジェクトのサイズを確認してILMルールを適用する前に、オブジェクトを保存しておく必要があります。つまり、 StorageGRID で取り込み時にデフォルトでオブジェクトの中間コピーを作成する必要があります。つまり、 StorageGRID での取り込み動作には Dual Commit オプションを使用する必要があります。

  同期配置と ILM ルールの詳細については、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してください。

- `Content-Type`
- `ETag`
- `X-Object-Meta-<name\>` （オブジェクト関連のメタデータ）

  ILMルールの参照時間として* User Defined Creation Time *オプションを使用する場合は、という名前のユーザ定義のヘッダーに値を格納する必要があります `X-Object-Meta-Creation-Time`。例：

  ```
  X-Object-Meta-Creation-Time: 1443399726
  ```

  このフィールドの値は、 1970 年 1 月 1 日からの秒数となります。

- `X-Storage-Class: reduced_redundancy`

  このヘッダーは、取り込まれたオブジェクトに一致する 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。

  - * 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` ヘッダーは、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も適しています。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` 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るため、他の状況ではヘッダーを使用することは推奨されません。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。

  を指定することに注意してください `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。

実行が成功すると、「 HTTP/1.1 201 Created 」の応答とともに次のヘッダーが返されます。

- `Content-Length`
- `Content-Type`
- `Date`
- `ETag`
- `Last-Modified`
- `X-Trans-Id`

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 Swift 処理"

OPTIONS 要求

OPTIONS 要求は、個々の Swift サービスが使用可能かどうかを確認します。OPTIONS

要求は、 URL で指定されたストレージノードまたはゲートウェイノードによって処理されます。

OPTIONS メソッド

たとえば、クライアントアプリケーションでは、 Swift 認証クレデンシャルを入力することなく、ストレージノード上の Swift ポートに OPTIONS 要求を問題 で送信して、ストレージノードが使用可能かどうかを判別できます。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。

情報（ info ） URL またはストレージ（ storage ） URL と併用する場合、 OPTIONS メソッドは、 HEAD 、 GET 、 OPTIONS 、 PUT など、指定された URL でサポートされる動詞のリストを返します。AUTH URL にはオプションを使用できません。

次の要求パラメータが必要です。

- `Account`

次の要求パラメータはオプションです。

- `Container`
- `Object`

実行が成功すると、「 HTTP/1.1 204 No Content 」の応答とともに次のヘッダーが返されます。ストレージ URL への OPTIONS 要求には、ターゲットが存在する必要はありません。

- `Allow` （HEAD、GET、OPTIONSなど、指定されたURLでサポートされる動詞のリスト）およびPUT）
- `Content-Length`
- `Content-Type`
- `Date`
- `X-Trans-Id`

関連情報

"サポートされている Swift API エンドポイント"

Swift API 処理に対するエラー応答

エラー応答について理解しておくと、処理をトラブルシューティングする際に役立ちます。

処理中にエラーが発生した場合に返される HTTP ステータスコードを次に示します。

Swift エラーの名前	HTTP ステータス
AccountNameTooLong 、 ContainerNameTooLong 、 HeaderTooBig 、 InvalidContainerName 、 InvalidRequest 、 InvalidURI 、 MetadataNameTooLong 、 MetadataValueTooBig 、 MissingSecurityHeader 、 ObjectNameTooLong 、 TooManyContainers 、 TooManyMetadataItems 、 TotalMetadataTooLarge	400 不正な要求です
アクセスが拒否されました	403 禁止
ContainerNotEmpty 、 ContainerAlreadyExists です	409 競合
内部エラー	500 Internal Server Error （内部サーバエラー）
InvalidRange ：無効な範囲	416 リクエストされた範囲が適合しません
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
NOTFOUND	404 が見つかりません
実装なし	501 は実装されていません
PreconditionalFailed	412 事前条件が失敗しました
resourceNotFound です	404 が見つかりません
権限がありません	401 認証なし
UnprocessableEntity の場合	422 加工不能エンティティ

StorageGRID の Swift REST API 処理

StorageGRID システム固有の処理が Swift REST API に追加されています。

GET コンテナセイコウセイヨウキユウ

整合性レベルを設定する場合は、オブジェクトの可用性と、異なるストレージノードおよびサイト間におけるオブジェクトの整合性のどちらかを犠牲にしなければなりません。GET コンテナ整合性要求では、特定のコンテナに適用されている整合性レベルを確認できます。

リクエスト

要求の **HTTP** ヘッダー	説明
`X-Auth-Token`	要求に使用するアカウントの Swift 認証トークンを指定します。
`x-ntap-sg-consistency`	要求のタイプを指定します `true` = GETコンテナconsistency、および `false` =コンテナを取得します。
`Host`	要求の転送先のホスト名。

要求例

```
GET /v1/28544923908243208806/Swift container
X-Auth-Token: SGRD_3a877009a2d24cb1801587bfa9050f29
x-ntap-sg-consistency: true
Host: test.com
```

応答

応答の **HTTP** ヘッダー	説明
`Date`	応答の日時。
`Connection`	サーバへの接続が開いているかどうか。
`X-Trans-Id`	要求の一意のトランザクション ID 。
`Content-Length`	応答の本文の長さ。

応答の **HTTP** ヘッダー	説明
`x-ntap-sg-consistency`	コンテナに適用されている整合性制御レベルです。次の値がサポートされています。 • * all * ：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 * ：新規オブジェクトについて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ついて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。 • 注：存在しないオブジェクトに対してアプリケーションが HEAD 要求を使用すると、使用できないストレージノードがあると「 500 Internal Server Error 」が大量に返される可能性があります。これらのエラーを防ぐには、「 available 」レベルを使用します。 • * available * （ HEAD オペレーションについては結果整合性）：「 read-after-new-write 」整合性レベルと動作は同じですが、 HEAD オペレーションについては結果整合性のみを提供します。ストレージ・ノードが使用できない場合 ' リードアフター・新規ライトよりもヘッド操作の可用性が高くなります

応答例

```
HTTP/1.1 204 No Content
Date: Sat, 29 Nov 2015 01:02:18 GMT
Connection: CLOSE
X-Trans-Id: 1936575373
Content-Length: 0
x-ntap-sg-consistency: strong-site
```

関連情報

"テナントアカウントを使用する"

PUT コンテナセイコウセイヨウキユウ

PUT コンテナ整合性要求では、コンテナに対して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。デフォルトでは ' 新しいコンテナは ' リードアフター・ア・ニュー・ライトの整合性レベルを使用して作成されます

リクエスト

要求の **HTTP** ヘッダー	説明
`X-Auth-Token`	要求に使用するアカウントの Swift 認証トークンです。
`x-ntap-sg-consistency`	コンテナに対する処理に適用される整合性制御レベルです。次の値がサポートされています。 • * all * ：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 * ：新規オブジェクトについて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ついて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。 • 注：存在しないオブジェクトに対してアプリケーションが HEAD 要求を使用すると、使用できないストレージノードがあると「 500 Internal Server Error 」が大量に返される可能性があります。これらのエラーを防ぐには、「 available 」レベルを使用します。 • * available * （ HEAD オペレーションについては結果整合性）：「 read-after-new-write 」整合性レベルと動作は同じですが、 HEAD オペレーションについては結果整合性のみを提供します。ストレージ・ノードが使用できない場合 ' リードアフター・新規ライトよりもヘッド操作の可用性が高くなります
`Host`	要求の転送先のホスト名。

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。

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

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。オブジェクトを読み出すことができません。

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。

要求例

```
PUT /v1/28544923908243208806/_Swift container_
X-Auth-Token: SGRD_3a877009a2d24cb1801587bfa9050f29
x-ntap-sg-consistency: strong-site
Host: test.com
```

応答

応答の **HTTP** ヘッダー	説明
`Date`	応答の日時。
`Connection`	サーバへの接続が開いているかどうか。
`X-Trans-Id`	要求の一意のトランザクション ID 。
`Content-Length`	応答の本文の長さ。

応答例

```
HTTP/1.1 204 No Content
Date: Sat, 29 Nov 2015 01:02:18 GMT
Connection: CLOSE
X-Trans-Id: 1936575373
Content-Length: 0
```

関連情報

"テナントアカウントを使用する"

REST APIのセキュリティの設定

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およびゲートウェイノード上の CLB サービス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。

CLB サービスは廃止されました。

- 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接続するか、ストレージノードに直接接続するか、またはゲートウェイノード上の CLB サービスに直接接続することができます。

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をストレージノードまたはゲートウェイノード上の CLB サービスに直接接続する場合、 StorageGRID システムのインストール時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証明書）を使用します。 グリッド管理者がグリッド用に指定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、ストレージノードまたはゲートウェイノード上の CLB サービスへの直接 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方法については、 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

Security 問題 の略	REST API の実装
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。

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。

サポートされている暗号スイート

TLS バージョン	IANA 暗号スイートの名前
1/2	TLS_ECDHE_RSA_with _AES_256_GCM_SHA384
TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	TLS_ECDHE_RSA_With _AES_128_GCM_SHA256
1.3	TLS_AES_256_GCM_SHA384
TLS_CHACHA20_POLY1305_SHA256	TLS_AES_128_GCM _SHA256

廃止された暗号スイート

次の暗号スイートは廃止されました。これらの暗号のサポートは今後のリリースで廃止される予定です。

IANA 名
TLS_RSA_With _AES_128_GCM _SHA256

IANA 名
TLS_RSA_With _AES_256_GCM_SHA384

関連情報

"クライアント接続の設定方法"

処理の監視と監査

グリッド全体または特定のノードのトランザクションの傾向を確認することで、クライアント処理のワークロードと効率を監視できます。監査メッセージを使用して、クライアント処理とトランザクションを監視できます。

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. サポートされているブラウザを使用してGrid Managerにサインインします。
2. ダッシュボードで、プロトコル操作セクションを探します。

   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. [ノード（ Nodes）]を選択し
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。

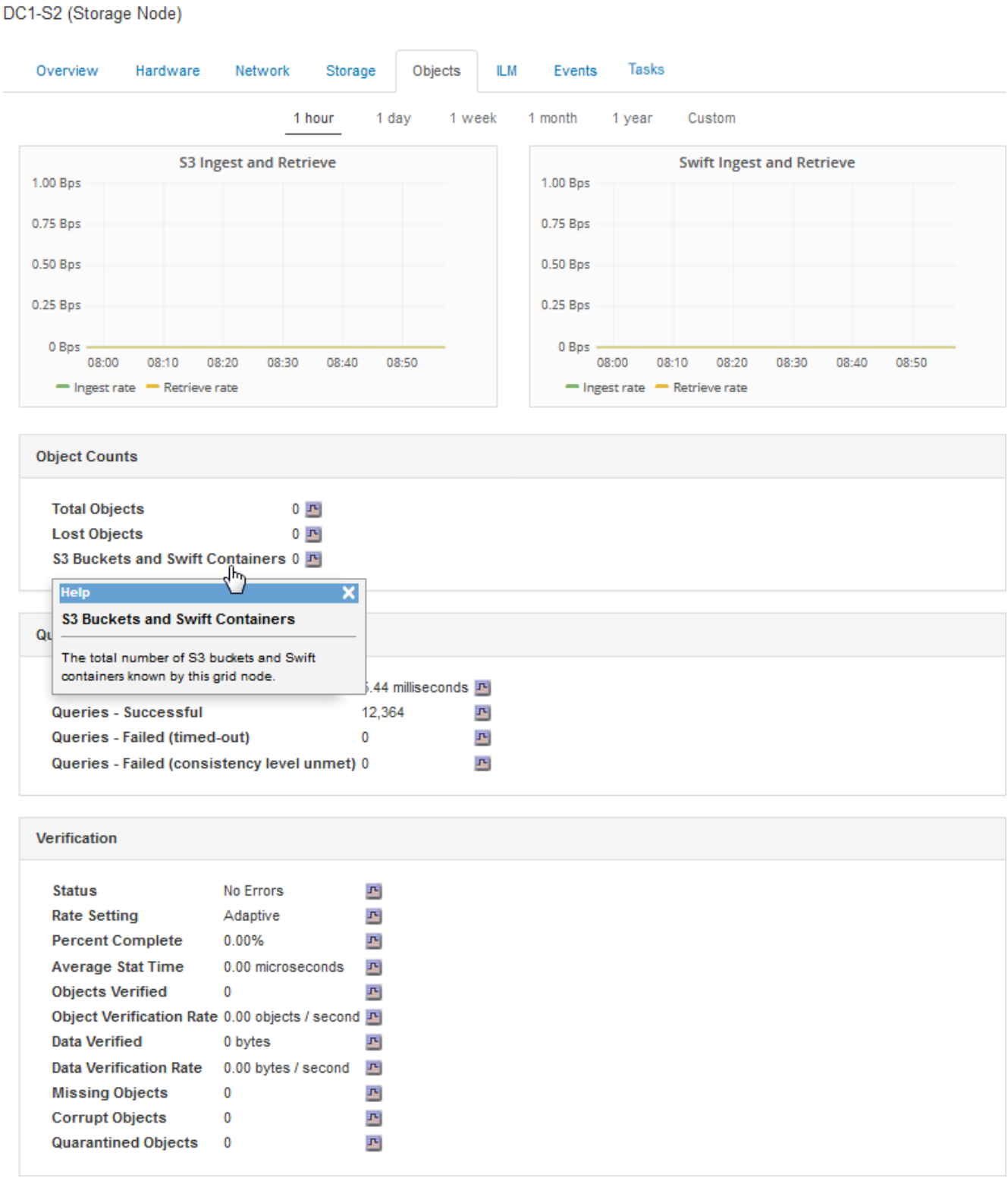
   このグラフには、グリッド内でロードバランサエンドポイントに送信されるすべてのクライアントトラフィックの傾向が表示されます。時間、日、週、月、年単位の間隔を選択できます。 または、カスタムの間隔を適用することもできます。

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. Support **>** Tools **>** Grid Topology *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。

監査ログへのアクセスと確認

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。

必要なもの

- 特定のアクセス権限が必要です。
- を用意しておく必要があります `Passwords.txt` ファイル。
- 管理ノードの IP アドレスを確認しておく必要があります。

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 audit.log ファイルが開始されます。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。

次の例は、アクティブなaudit.logファイル、前日のファイル（2018-04-15.txt）、および前日の圧縮されたファイルを示しています (`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。
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。

関連情報

"監査ログを確認します"

監査ログで追跡される **Swift** 処理

ストレージに対する成功した DELETE 、 GET 、 HEAD 、 POST 、 PUT の各処理は、 StorageGRID 監査ロ

グで追跡されます。エラーはログに記録されず、情報、認証、オプションの要求も記録されません。

次の Swift 処理で追跡される情報の詳細については、「監査メッセージの概要」を参照してください。

アカウントの処理

- GET アカウント
- HEAD アカウント

コンテナの処理

- コンテナを削除します
- GET コンテナ
- HEAD コンテナ
- PUT コンテナ

オブジェクトの処理

- オブジェクトを削除します
- GET オブジェクト
- HEAD オブジェクト
- PUT オブジェクト

関連情報

"監査ログを確認します"

"アカウントの処理"

"コンテナの処理"

"オブジェクトの処理"

著作権に関する情報

商標に関する情報