StorageGRID® Webscale 10.3

# Swift Implementation Guide

September 2016 | 215-10827_A0
doccomments@netapp.com

# Contents

# OpenStack Swift API support in StorageGRID Webscale

Support for the OpenStack Swift Representational State Transfer Application Programming Interface (REST API) enables client applications developed for OpenStack Swift to store and retrieve objects on a StorageGRID Webscale system. Before using the API, you might benefit by understanding its implementation in StorageGRID Webscale.

StorageGRID Webscale supports the following versions:

| Item | Version |
|---|---|
| Swift specification | OpenStack Swift Object Storage API v1 as of November 2015 |
| HTTP | 1.1<br>For details about HTTP, see HTTP/1.1 (RFC 2616). |

**Related information**

*OpenStack: Object Storage API*

## History of Swift API support in StorageGRID Webscale

Understanding the initiation of and any changes to the support for the Swift API in the StorageGRID Webscale system might help you design your implementation.

The following table documents the StorageGRID Webscale system support for the Swift API:

| Date | Release | Comments |
|---|---|---|
| August 2016 | 10.3 | Administrative updates and corrections to the document. Removed sections for configuring custom server certificates. |
| December 2015 | 10.2 | Initial support of the Swift API by the StorageGRID Webscale system.<br>The currently supported version is OpenStack Swift Object Storage API v1. |

## How StorageGRID Webscale implements the Swift REST API

A client application can use Swift REST API calls to connect to storage nodes and API Gateway nodes to create containers and to store and retrieve objects. This enables service-oriented applications developed for OpenStack Swift to connect with on-premises object storage provided by the StorageGRID Webscale system.

To manage objects, the StorageGRID Webscale system uses information lifecycle management (ILM) rules.

For information about ILM rules, see the *Administrator Guide*.

## Consistency guarantees and controls

StorageGRID Webscale guarantees read-after-write consistency for newly created objects. Any GET following a successfully completed PUT will be able to read the newly written data. Overwrites of existing objects, metadata updates, and deletes remain eventually consistent.

StorageGRID Webscale now also allows the user to control consistency on a per container basis. This allows users to tradeoff consistency and availability as required by their application. By default, reads of non-existent object now require certain Storage Nodes to be available. When one or more Storage Nodes are unavailable, reading some non-existent object may fail with an HTTP 500 error. Reading with “weak” consistency will restore the previous behavior that values availability over consistency.

### Related information

*StorageGRID Webscale 10.3 Administrator Guide*

# Swift REST API supported operations

The StorageGRID Webscale system supports most operations in the OpenStack Swift API. If you are integrating Swift REST API clients with StorageGRID Webscale, understanding the implementation details for account, container, and object operations is helpful.

### Operations supported in StorageGRID Webscale

The following Swift API operations are supported:

- *Account operations* on page 9
- *Container operations* on page 10
- *Object operations* on page 13

### Common response headers for all operations

The StorageGRID Webscale system implements all common headers for supported operations as defined by the OpenStack Swift Object Storage API v1.

**Related concepts**

*OpenStack Swift API support in StorageGRID Webscale* on page 4

**Related information**

*OpenStack: Object Storage API*

# General information about Swift info, auth, and storage URLs

StorageGRID Webscale supports several Swift API endpoint types.

These Swift API endpoints are:

- info URL
- auth URL
- storage URL

### Swift capabilities and limitations with info URL

The capabilities and limitations of the StorageGRID Webscale Swift implementation can be queried through the Swift info URL. You obtain this information by issuing a GET request to the StorageGRID Webscale Swift base URL with the `/info` path.

```
https://FQDN | IP:Swift_Port/info/
```

The StorageGRID Webscale implementation of Swift allows unauthenticated access to the info URL.

A GET request to the info URL yields the capabilities of the Swift implementation as a JSON dictionary. A client tool can parse the returned JSON response to determine the capabilities of the implementation and employ them as constraints for subsequent storage operations.

### User authentication with auth URL

A client can authenticate a tenant user and procure a Swift token from the Swift auth URL. A successful authentication request yields a token and a storage URL, which are required for access on a StorageGRID Webscale CLB service on the Gateway Node or LDR service on a Storage Node.

```
https://FQDN | IP:Swift_Port/auth/v1.0/
```

The credentials include the user name and password as parameters and must be provided using request headers as follows:

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

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

The Swift tenant account information is used in the authentication process and consists of one of the following:

- If Identity Federation is enabled for the tenant account (for Active Directory or LDAP configurations), you should provide the username and password of the federated user from the AD or LDAP server. Alternatively, LDAP users can be referred to with their domain name, for example, `X-Auth-User: <Tenant_Account_ID>:<Username@Domain_Name>`
- For local accounts when LDAP is not configured, you should use "swiftadmin" as the user name and the password provided during tenant account creation.

A valid user name and password combination yields a valid token and storage URL through response headers as shown in the following:

```
X-Storage-Url   : https://FQDN | IP:Swift_Port/v1/Tenant_Account_ID
X-Auth-Token    : token
X-Storage-Token : token
```

By default, the token is valid for 24 hours from generation time.

Tokens are generated for a specific tenant account. A valid token for one account does not authorize a user to access another account.

### Swift API operations with storage URL

A client application can issue Swift REST API calls to perform supported account, container, and object operations against a CLB service on the Gateway Node or LDR service on a Storage Node. Storage requests can be addressed to the URL that is returned by the auth request in the `X-Storage-Url` response header. The request must include the `X-Auth-Token` header and value returned from the auth request.

```
https://FQDN | IP:Swift_Port/v1/
Tenant_Account_ID[/container][/object]
```

Because StorageGRID Webscale uses an eventually-consistent data model, some storage response headers that contain usage statistics might not reflect accurate numbers for recently modified objects. It might take a few minutes for accurate numbers to appear in these headers.

The following response headers are examples of those that contain usage statistics:

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

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

For details about responses, see information about account, container, and object operations.

**Related references**

*Account operations* on page 9
*Container operations* on page 10
*Object operations* on page 13
*Port numbers on API Gateway Nodes and Storage Nodes for Swift* on page 22

## Error responses to Swift API operations

Understanding the possible error responses can help you troubleshoot operations.

The following HTTP status codes might be returned when errors occur during an operation:

| Swift error name | HTTP status |
|---|---|
| AccountNameTooLong, ContainerNameTooLong, HeaderTooBig, InvalidContainerName, InvalidRequest,<br>InvalidURI, MetadataNameTooLong, MetadataValueTooBig, MissingSecurityHeader, ObjectNameTooLong, TooManyContainers, TooManyMetadataItems, TotalMetadataTooLarge | 400 Bad Request |
| AccessDenied | 403 Forbidden |
| ContainerNotEmpty, ContainerAlreadyExists | 409 Conflict |
| InternalError | 500 Internal Server Error |
| InvalidRange | 416 Requested Range Not Satisfiable |
| MethodNotAllowed | 405 Method Not Allowed |
| MissingContentLength | 411 Length Required |
| NotFound | 404 Not Found |
| NotImplemented | 501 Not Implemented |
| PreconditionFailed | 412 Precondition Failed |
| ResourceNotFound | 404 Not Found |
| Unauthorized | 401 Unauthorized |
| UnprocessableEntity | 422 Unprocessable Entity |

## Account operations

The following Swift API operations are performed on accounts.

| Operation | Implementation |
|---|---|
| GET account | Retrieves the container list associated with the account and account usage statistics.<br>The following request parameter is required:<br>• `Account`<br>The following request header is required:<br>• `X-Auth-Token`<br>The following supported request query parameters are optional:<br>• `Delimiter`<br>• `End_marker`<br>• `Format`<br>• `Limit`<br>• `Marker`<br>• `Prefix`<br>A successful execution returns the following headers with an “HTTP/1.1 204 No Content” response if the account is found and has no containers or the container list is empty; or an “HTTP/1.1 200 OK” response if the account is found and the container list is not empty:<br>• `Accept-Ranges`<br>• `Content-Length`<br>• `Content-Type`<br>• `Date`<br>• `X-Account-Bytes-Used`<br>• `X-Account-Container-Count`<br>• `X-Account-Object-Count`<br>• `X-Timestamp`<br>• `X-Trans-Id` |

| Operation | Implementation |
|---|---|
| HEAD account | Retrieves account information and statistics from a Swift account.<br>The following request parameter is required:<br>- `Account`<br>The following request header is required:<br>- `X-Auth-Token`<br>A successful execution returns the following headers with an “HTTP/1.1 204 No Content” response:<br>- `Accept-Ranges`<br>- `Content-Length`<br>- `Date`<br>- `X-Account-Bytes-Used`<br>- `X-Account-Container-Count`<br>- `X-Account-Object-Count`<br>- `X-Timestamp`<br>- `X-Trans-Id` |

## Container operations

The following Swift API operations are performed on containers.

The StorageGRID Webscale system supports a maximum of 100 containers per Swift account.

| Operation | Implementation |
|---|---|
| DELETE container | Removes an empty container from a Swift account in a StorageGRID Webscale system.<br>The following request parameters are required:<br>- `Account`<br>- `Container`<br>The following request header is required:<br>- `X-Auth-Token`<br>A successful execution returns the following headers with an "HTTP/1.1 204 No Content" response:<br>- `Content-Length`<br>- `Content-Type`<br>- `Date`<br>- `X-Trans-Id` |

| Operation | Implementation |
|---|---|
| GET container | Retrieves the object list associated with the container along with container statistics and metadata in a StorageGRID Webscale system.<br>The following request parameters are required:<br>• `Account`<br>• `Container`<br>The following request header is required:<br>• `X-Auth-Token`<br>The following supported request query parameters are optional:<br>• `Delimiter`<br>• `End_marker`<br>• `Format`<br>• `Limit`<br>• `Marker`<br>• `Path`<br>• `Prefix`<br>A successful execution returns the following headers with an "HTTP/1.1 200 Success" or a "HTTP/1.1 204 No Content" response:<br>• `Accept-Ranges`<br>• `Content-Length`<br>• `Content-Type`<br>• `Date`<br>• `X-Container-Bytes-Used`<br>• `X-Container-Object-Count`<br>• `X-Timestamp`<br>• `X-Trans-Id` |

| **Operation** | **Implementation** |
|---|---|
| HEAD container | Retrieves container statistics and metadata from a StorageGRID Webscale system.<br>The following request parameters are required:<br>• `Account`<br>• `Container`<br>The following request header is required:<br>• `X-Auth-Token`<br>A successful execution returns the following headers with an "HTTP/1.1 204 No Content" response:<br>• `Accept-Ranges`<br>• `Content-Length`<br>• `Date`<br>• `X-Container-Bytes-Used`<br>• `X-Container-Object-Count`<br>• `X-Timestamp`<br>• `X-Trans-Id` |
| PUT container | Creates a container for an account in a StorageGRID Webscale system.<br>The following request parameters are required:<br>• `Account`<br>• `Container`<br>The following request header is required:<br>• `X-Auth-Token`<br>A successful execution returns the following headers with an "HTTP/1.1 201 Created" or "HTTP/1.1 202 Accepted" (if the container already exists under this account) response:<br>• `Content-Length`<br>• `Date`<br>• `X-Timestamp`<br>• `X-Trans-Id`<br>A container name must be unique in the StorageGRID Webscale namespace. If the container exists under another account, the following header is returned: "HTTP/1.1 409 Conflict." |

## Object operations

The following Swift API operations are performed on objects.

| Operation | Implementation |
|---|---|
| DELETE object | Deletes an object's content and metadata from the StorageGRID Webscale system.<br>The following request parameters are required:<br>• `Account`<br>• `Container`<br>• `Object`<br>The following request header is required:<br>• `X-Auth-Token`<br>A successful execution returns the following response headers with an "HTTP/1.1 204 No Content" response:<br>• `Content-Length`<br>• `Content-Type`<br>• `Date`<br>• `X-Trans-Id` |

| Operation | Implementation |
|---|---|
| GET object | Retrieves the object content and gets the object metadata from a StorageGRID Webscale system.<br>The following request parameters are required:<br>• `Account`<br>• `Container`<br>• `Object`<br>The following request header is required:<br>• `X-Auth-Token`<br>The following request headers are optional:<br>• `Accept-Encoding`<br>• `If-Match`<br>• `If-Modified-Since`<br>• `If-None-Match`<br>• `If-Unmodified-Since`<br>• `Range`<br>A successful execution returns the following headers with an "HTTP/1.1 200 OK" response:<br>• `Accept-Ranges`<br>• `Content-Length`<br>• `Content-Type`<br>• `Date`<br>• `ETag`<br>• `Last-Modified`<br>• `X-Timestamp`<br>• `X-Trans-Id` |
| HEAD object | Retrieves metadata and properties of an ingested object from a StorageGRID Webscale system.<br>The following request parameters are required:<br>• `Account`<br>• `Container`<br>• `Object`<br>The following request header is required:<br>• `X-Auth-Token`<br>A successful execution returns the following headers with an "HTTP/1.1 200 OK" response:<br>• `Accept-Ranges`<br>• `Content-Length`<br>• `Content-Type`<br>• `Date`<br>• `ETag`<br>• `Last-Modified`<br>• `X-Timestamp`<br>• `X-Trans-Id` |

| Operation | Implementation |
|---|---|
| PUT object | Creates a new object with data and metadata, or replaces an existing object with data and metadata in a StorageGRID Webscale system.<br>The following request parameters are required:<br>• `Account`<br>• `Container`<br>• `Object`<br>The following request header is required:<br>• `X-Auth-Token`<br>The following request headers are optional:<br>• `Content-Encoding`<br>• `Content-Length`<br>• `Content-Type`<br>• `ETag`<br>• `Transfer-Encoding`<br>• `X-Object-Meta-<name>` (object-related metadata)<br>To record the object creation time, so that you can use the User Defined Creation Time option for the reference time in an ILM rule, you need to store the value in a user-defined header named `X-Object-Meta-Creation-Time`. For example: `X-Object-Meta-Creation-Time`=1443399726. This field is evaluated as seconds since Jan 1, 1970.<br>For details, see "Reference time" in the *Administrator Guide*.<br>• `X-Storage-Class:reduced_redundancy`<br>Specifies a single-commit ingest operation. This does not affect the information lifecycle management (ILM) policy and does not result in data being stored at lower levels of redundancy in the StorageGRID Webscale system.<br>For details, see information about ILM policies in the *Administrator Guide*.<br>A successful execution returns the following headers with an "HTTP/1.1 201 Created" response:<br>• `Content-Length`<br>• `Content-Type`<br>• `Date`<br>• `ETag`<br>• `Last-Modified`<br>• `X-Trans-Id` |

**Related information**

*StorageGRID Webscale 10.3 Administrator Guide*

## OPTIONS method

The OPTIONS request helps to check the availability of an individual Swift service. The OPTIONS request is handled by the LDR service on a Storage Node or the CLB service on the Gateway Node specified in the URL.

| Operation | Implementation |
|---|---|
| OPTIONS method | The OPTIONS method retrieves supported RESTful verbs for the following types of URLs: *info URL* and *storage URL* from a StorageGRID Webscale system.<br>The following request parameter is required:<br>• `Account`<br>The following request parameters are optional:<br>• `Container`<br>• `Object`<br>A successful execution returns the following headers with an "HTTP/1.1 204 No Content" response. The OPTIONS request to the storage URL does not require that the target exists.<br>• `Allow` (a list of supported verbs for the given URL, for example, HEAD, GET, OPTIONS, and PUT)<br>• `Content-Length`<br>• `Content-Type`<br>• `Date`<br>• `X-Trans-Id` |

## Operations tracked in the audit logs

All successful storage DELETE, GET, HEAD, and PUT operations are tracked in the StorageGRID Webscale audit log.

| Account operations | Container operations | Object operations |
|---|---|---|
| GET account | DELETE container | DELETE object |
| HEAD account | GET container | GET object |
|  | HEAD container | HEAD object |
|  | PUT container | PUT object |

# StorageGRID Webscale Swift REST API operations

There are operations added on to the Swift REST API that are specific to StorageGRID Webscale system.

## GET container consistency request

The GET container consistency request allows you to determine the consistency level being applied to a particular container.

### Request

| Request HTTP Header | Description |
|---|---|
| X-Auth-Token | Specifies the Swift authentication token for the account to use for the request. |
| x-ntap-sg-consistency | Specifies whether consistency controls are enabled (true) or not (false). |
| Host | The hostname to which the request is directed. |

### Request example

```
GET /v1/28544923908243208806/<Swift container> HTTP/1.1
x-ntap-sg-consistency: true
X-Auth-Token: SGRD_3a877009a2d24cb1801587bfa9050f29
Host: test.com
```

### Response

| Response HTTP Header | Description |
|---|---|
| Date | The date and time of the response. |
| Connection | Specifies whether the connection to the server is open or closed. |
| X-Trans-Id | The unique transaction identifier for the request. |
| Content-Length | The length of the response body. |

| Response HTTP Header | Description |
|---|---|
| x-ntap-sg-consistency | Specifies the consistency control level being applied to the container. The following values are supported:<br>• all: Provides the highest consistency guarantee. All nodes receive the data immediately or the request will fail.<br>• strong-global: Guarantees a consistent read-after-write view for all operations across all sites.<br>• strong-site: Guarantees a consistent read-after-write view for all operations within a site.<br>• weak: Ensures the highest availability, but provides no consistency guarantees and a reduced guarantee of data protection.<br>• default: Eventually consistent, highly available, with data protection guarantees. |

### Response example

```
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
```

## PUT container consistency request

The PUT container consistency request allows you to specify the consistency level to apply to operations performed on a container.

### Request

| Request HTTP Header | Description |
|---|---|
| X-Auth-Token | Specifies the Swift authentication token for the account to use for the request. |

| Request HTTP Header | Description |
|---|---|
| x-ntap-sg-consistency | Specifies the consistency control level to apply to operations on the container. The following values are supported:<br>• all: Provides the highest consistency guarantee. All nodes receive the data immediately or the request will fail.<br>• strong-global: Guarantees a consistent read-after-write view for all operations across all sites.<br>• strong-site: Guarantees a consistent read-after-write view for all operations within a site.<br>• weak: Ensures the highest availability, but provides no consistency guarantees and a reduced guarantee of data protection.<br>• default: Eventually consistent, highly available, with data protection guarantees. |
| Host | The hostname to which the request is directed. |

### Request example

```
PUT /v1/28544923908243208806/<Swift container> HTTP/1.1
x-ntap-sg-consistency: strong-site
X-Auth-Token: SGRD_3a877009a2d24cb1801587bfa9050f29
Host: test.com
```

### Response

| Response HTTP Header | Description |
|---|---|
| Date | The date and time of the response. |
| Connection | Specifies whether the connection to the server is open or closed. |
| X-Trans-Id | The unique transaction identifier for the request. |
| Content-Length | The length of the response body. |

### Response example

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

# Configuring tenant accounts and connections

Configuring StorageGRID Webscale to accept connections from Swift client applications requires creating and configuring a Swift tenant account.

### About this task

Creating and configuring tenant accounts and connections involves the following tasks:

- Create a tenant account.
- Identify IP addresses for API Gateway Nodes and Storage Nodes.
- Use accurate port numbers for API Gateway Nodes and Storage Nodes.

If your environment includes a form of identity federation (LDAP or AP), you should also complete the following tasks:

- Configure LDAP for identity federation.
- Edit group policies.

For details, see the *Administrator Guide*.

### Related information

*StorageGRID Webscale 10.3 Administrator Guide*

## Creating tenant accounts for Swift

You can create a Swift tenant account for each group that requires access to the StorageGRID Webscale system using the Swift REST API. A tenant account can be created for an organization, division, department, or any other internal or external group you want to use to define access to storage in your StorageGRID Webscale system. If you configured LDAP for this account, all groups and users in the LDAP domain can access Swift via this account.

### Before you begin

- You must have signed in to the Grid Management Interface using a supported browser.
- To perform this task, you need specific access permissions. For details, see information about controlling system access with administration user accounts and groups.

### About this task

The Swift tenant account information is used in the authentication process. Configuring a Swift client requires one of the following sets of user credentials:

- If Identity Federation is enabled for the tenant account (for Active Directory or LDAP configurations), you should provide the username and password of the federated user from the AD or LDAP server. Alternatively, LDAP users can be referred to with their domain name, for example, `X-Auth-User: *Tenant_Account_ID*:*Username@Domain_Name*`
- For local accounts when LDAP is not configured, you should use the "swiftadmin" as the user name and the password provided during tenant account creation.

### Steps

1. Select **Tenants**.
2. Click **Create**.
3. Configure the tenant account in the **Add Tenant Account** dialog box:
    a. Select **Swift** as the protocol.
    b. In the **Name** text box enter the name to display in the StorageGRID Webscale system.
    c. If you want to use the local Swift Administrator account, instead of or in addition to LDAP authentication, enter the password to use in the **Password** and **Confirm Password** text boxes.

       The password must be between 8 and 32 characters. You must enter a strong password to ensure the security of your StorageGRID Webscale system.
    d. Click **Save**.

### Related information

*StorageGRID Webscale 10.3 Administrator Guide*

## How client applications use HTTPS connections

Client applications use HTTPS connections to access and communicate with the StorageGRID Webscale system. Understanding HTTPS connections helps you understand StorageGRID Webscale requirements.

A client application can connect directly to an API Gateway Node or Storage Node to store and retrieve objects. To load balance ingests across the Storage Nodes in your grid, you can connect to an API Gateway Node, which handles the load balancing for you. Otherwise, you can connect directly to a Storage Node.

**Note:** IPv6 is supported only for client application connections through the API Gateway Node. For details about support for IPv6, see the *StorageGRID Webscale Administrator Guide*.

Client applications can issue `OPTIONS` HTTPS requests to the Swift port on a Storage Node, without providing Swift authentication credentials, to determine whether the LDR Service is available. You can use this request for monitoring or to allow external load balancers to identify when a Storage Node is down.

Setting up the connection to client applications involves the following tasks:

- Creating a Swift tenant account
- Identifying IP addresses for API Gateway Nodes and Storage Nodes
- Identifying Swift port numbers for API Gateway Nodes and Storage Nodes
- Copying the system's certificate authority (CA) certificate for client applications that require server validation

For details about setting up connections, see the *StorageGRID Webscale Administrator Guide*.

### Related information

*StorageGRID Webscale 10.3 Administrator Guide*

# Identifying IP addresses for API Gateway Nodes and Storage Nodes

You need the grid node's IP address to connect API client applications to StorageGRID Webscale.

### Steps

1. Sign in to the Grid Management Interface using a supported browser.
2. Select **Grid**.
3. In the **Grid Topology** tree, locate and expand the Storage Node or API Gateway Node to which you want to connect.

   The services for the selected grid node appear.
4. In the Storage Node or API Gateway Node, select **SSM > Resources**, and then scroll to the **Network Addresses** table.

   You can establish HTTPS connections from API client applications to any of the listed IP addresses.

# Port numbers on API Gateway Nodes and Storage Nodes for Swift

API Gateway Nodes and Storage Nodes are available for HTTPS connections from client applications to the StorageGRID Webscale system only on specific port numbers.

The following ports are used for Swift client applications to connect to the StorageGRID Webscale system:

| Grid node | Port number |
|---|---|
| API Gateway Node (CLB Swift Port) | 8083 |
| Storage Node (LDR Swift Port) | 18083 |

# Configuring security for the REST API

You need to understand the security measures implemented for the REST API and how to secure your system.

## How the StorageGRID Webscale system implements security for the REST API

The StorageGRID Webscale system employs the use of Transport Layer Security (TLS) connection security, server authentication, client authentication, and client authorization. When considering security issues, you might find it helpful to understand how the StorageGRID Webscale system implements security, authentication, and authorization for the S3 or Swift REST API.

The StorageGRID Webscale system accepts HTTPS commands submitted over a network connection that uses TLS to provide connection security, application authentication and, optionally, transport encryption. Commands that do not use TLS are rejected. If an object is encrypted when it is ingested, it stays encrypted for the lifetime of the object in the StorageGRID Webscale system.

TLS enables the exchange of certificates as entity credentials and allows a negotiation that can use hashing and encryption algorithms.

When a StorageGRID Webscale system is installed, a certificate authority (CA) certificate is generated for the system, as well as server certificates for each Storage Node. These server certificates are all signed by the system CA. You need to configure client applications to trust this CA certificate. When a client application connects to any Storage Node using TLS, the application can authenticate the Storage Node by verifying that the server certificate presented by the Storage Node is signed by the trusted system CA.

Alternatively, you can choose to supply a single, custom server certificate that should be used on all Storage Nodes rather than the generated ones. The custom server certificate must be signed by a CA selected by the administrator. The server authentication process by the client application is the same, but in this instance with a different trusted CA. For more information, see “Configuring certificates” in the *Administrator Guide*.

The following table shows how security issues are implemented for S3 and Swift API:

| Security issue | Implementation for REST API |
|---|---|
| Connection security | TLS |
| Server authentication | X.509 server certificate signed by system CA or custom server certificate supplied by administrator |
| Client authentication | • S3: S3 account (access key ID and secret access key)<br>• Swift: Swift account (credentials of user name and password) |
| Client authorization | • S3: Bucket ownership and all applicable access control policies<br>• Swift: Account admin role access |

**Related information**

*StorageGRID Webscale 10.3 Administrator Guide*

## How client applications use certificates for security with REST APIs

When a client application establishes a TLS session to the StorageGRID Webscale system, the system sends a server certificate to the client application for verification to ensure that the HTTPS connection is secure.

The client application loads the grid CA certificate and uses it to verify that the client application is communicating with the expected StorageGRID Webscale system. This process protects against man-in-the-middle and impersonation attacks.

## Supported hashing and encryption algorithms for TLS libraries

Client applications use the HTTPS protocol to communicate with the StorageGRID Webscale system over a network connection that uses Transport Layer Security (TLS). The StorageGRID Webscale system supports a limited set of hashing and encryption algorithms from the TLS libraries that client applications can use when establishing a TLS session. When you are setting up the communication processes, it is important for you to know which security algorithms the system uses.

The StorageGRID Webscale system supports the following cipher suite security algorithms:

- AES128-SHA
- AES256-SHA
- AES128-GCM
- AES256-GCM

AES128-SHA and AES256-SHA provide secure encryption and efficient processing of objects. AES128-GCM and AES256-GCM provide secure encryption and more efficient processing of large objects. The TLS session negotiates the connection, using either AES128 or AES256 based on the client application requirements, and the need to balance performance with encryption security.

# Testing your connection in the Swift API configuration

You can use the Swift CLI to test your connection to the StorageGRID Webscale system and to verify that you can read and write objects to the system.

### Before you begin

- You must have downloaded and installed *python-swiftclient*, the Swift command-line client, at *https://swiftstack.com/docs/integration/python-swiftclient.html*.
- You must have created a Swift tenant account in the StorageGRID Webscale system.

### About this task

If you have not configured security as described in configuring security information, then you must add the `--insecure` flag to each of these commands.

### Steps

1. Query the info URL for your StorageGRID Webscale Swift deployment:

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

   This is sufficient to test that your Swift deployment is functional. To further test account configuration by storing an object, continue with the additional steps.

2. Put an object in the container:

   ```
   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. Get the container to verify the object:

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

4. Delete the object:

   ```
   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. Delete the container:

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

### Related tasks

*Creating tenant accounts for Swift* on page 20

# Monitoring and auditing operations

You can monitor the health of your client application connections to the StorageGRID Webscale system by viewing summary attributes that list transaction counts for all LDR services, or you can view the transactions for a specific LDR service. Also, you can use audit messages to monitor the operations and transactions of the StorageGRID Webscale system.

### Steps

1. Viewing HTTPS transactions for Swift objects on page 27
   You can view the number of successful and failed attempts by client applications to read, write, and modify Swift objects in the StorageGRID Webscale system. You can view a summary of all transactions for all LDR services, or you can view the transactions for a specific LDR service. You might want to do this to evaluate the health of the system.
2. Viewing information about data objects on page 28
   You can use an object ID in the StorageGRID Webscale system to view information about the data object. You can check on the current location of the object and obtain any metadata associated with the object.
3. Accessing and reviewing audit logs on page 29
   The StorageGRID Webscale system securely and reliably transports audit messages from each service within the StorageGRID Webscale system to one or more audit repositories. API-specific (S3, Swift, and CDMI) audit messages provide critical security, operations, and performance monitoring data that can help you evaluate the health of your system.

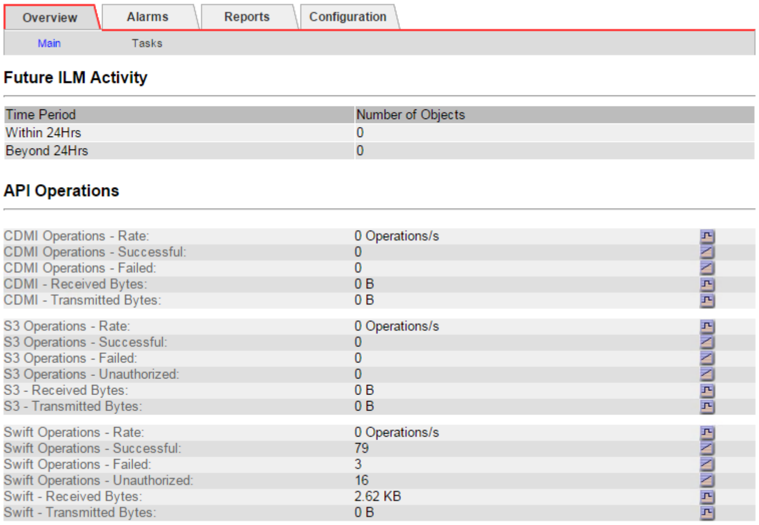
## Viewing HTTPS transactions for Swift objects

You can view the number of successful and failed attempts by client applications to read, write, and modify Swift objects in the StorageGRID Webscale system. You can view a summary of all transactions for all LDR services, or you can view the transactions for a specific LDR service. You might want to do this to evaluate the health of the system.

### Steps

1. Sign in to the Grid Management Interface using a supported browser.
2. Select **Grid**.
3. Select ***`deployment`*** > **Overview** > **Main**, and then view the **API Operations** area.

   The grid name is the top-level entry in the Grid Topology tree. The API Operations area displays a summary of information from all of the LDR services that support Swift client applications.



**Future ILM Activity**

| Time Period | Number of Objects |
|---|---|
| Within 24Hrs | 0 |
| Beyond 24Hrs | 0 |

**API Operations**

| | |
|---|---|
| CDMI Operations - Rate: | 0 Operations/s |
| CDMI Operations - Successful: | 0 |
| CDMI Operations - Failed: | 0 |
| CDMI - Received Bytes: | 0 B |
| CDMI - Transmitted Bytes: | 0 B |
| S3 Operations - Rate: | 0 Operations/s |
| S3 Operations - Successful: | 0 |
| S3 Operations - Failed: | 0 |
| S3 Operations - Unauthorized: | 0 |
| S3 - Received Bytes: | 0 B |
| S3 - Transmitted Bytes: | 0 B |
| Swift Operations - Rate: | 0 Operations/s |
| Swift Operations - Successful: | 79 |
| Swift Operations - Failed: | 3 |
| Swift Operations - Unauthorized: | 16 |
| Swift - Received Bytes: | 2.62 KB |
| Swift - Transmitted Bytes: | 0 B |

4. Select ***`Storage Node`*** **> LDR > Swift > Overview > Main** to view information for individual LDR services.

# Viewing information about data objects

You can use an object ID in the StorageGRID Webscale system to view information about the data object. You can check on the current location of the object and obtain any metadata associated with the object.

### Before you begin

- You must have signed in to the Grid Management Interface using a supported browser.
- To perform this task, you need specific access permissions. For details, see information about controlling system access with administration user accounts and groups.
- You must have the object ID from the client application. Object ID can be one of:
  - CBID (content block identifier)
  - UUID (universally unique identifier)
  - Object ID
  - Container/Object_Key

### Steps

1. Select **Grid**.
2. In the Grid Topology tree, select ***`primary Admin Node`*** **> CMN > Object Lookup > Configuration**.
3. In the **Object Identifier** box, enter an object ID, and click **Apply Changes**:

   **Note:** If you enter an invalid object ID, an error message appears.

4. Click the **Overview** tab to review the results.

### Related information

*StorageGRID Webscale 10.3 Administrator Guide*

## Accessing and reviewing audit logs

The StorageGRID Webscale system securely and reliably transports audit messages from each service within the StorageGRID Webscale system to one or more audit repositories. API-specific (S3, Swift, and CDMI) audit messages provide critical security, operations, and performance monitoring data that can help you evaluate the health of your system.

### About this task

The StorageGRID Webscale system compresses audit logs after one day and renames them using the format *`YYYY-MM-DD.txt.gz`* (where the original date is preserved).

### Steps

1. Log in to the server using the user name and password as recorded in the `Passwords.txt` file.
2. Access the audit log directory through a command line of the server that hosts the AMS service.
3. Go to the `/var/local/audit/export/` directory.
4. View the `audit.log` file.

# Copyright information

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.277-7103 (October 1988) and FAR 52-227-19 (June 1987).

# Trademark information

NetApp, the NetApp logo, Go Further, Faster, AltaVault, ASUP, AutoSupport, Campaign Express, Cloud ONTAP, Clustered Data ONTAP, Customer Fitness, Data ONTAP, DataMotion, Fitness, Flash Accel, Flash Cache, Flash Pool, FlashRay, FlexArray, FlexCache, FlexClone, FlexPod, FlexScale, FlexShare, FlexVol, FPolicy, GetSuccessful, LockVault, Manage ONTAP, Mars, MetroCluster, MultiStore, NetApp Insight, OnCommand, ONTAP, ONTAPI, RAID DP, RAID-TEC, SANtricity, SecureShare, Simplicity, Simulate ONTAP, Snap Creator, SnapCenter, SnapCopy, SnapDrive, SnapIntegrator, SnapLock, SnapManager, SnapMirror, SnapMover, SnapProtect, SnapRestore, Snapshot, SnapValidator, SnapVault, StorageGRID, Tech OnTap, Unbound Cloud, and WAFL and other names are trademarks or registered trademarks of NetApp, Inc., in the United States, and/or other countries. All other brands or products are trademarks or registered trademarks of their respective holders and should be treated as such. A current list of NetApp trademarks is available on the web.

*http://www.netapp.com/us/legal/netapptmlist.aspx*

# How to send comments about documentation and receive update notifications

You can help us to improve the quality of our documentation by sending us your feedback. You can receive automatic notification when production-level (GA/FCS) documentation is initially released or important changes are made to existing production-level documents.

If you have suggestions for improving this document, send us your comments by email.

*doccomments@netapp.com*

To help us direct your comments to the correct division, include in the subject line the product name, version, and operating system.

If you want to be notified automatically when production-level documentation is released or important changes are made to existing production-level documents, follow Twitter account @NetAppDoc.

You can also contact us in the following ways:

- NetApp, Inc., 495 East Java Drive, Sunnyvale, CA 94089 U.S.
- Telephone: +1 (408) 822-6000
- Fax: +1 (408) 822-4501
- Support telephone: +1 (888) 463-8277

# Index