Oracle® Cloud

Using Oracle Cloud Infrastructure Object Storage Classic

E37261-43
May 2020

Oracle Cloud Using Oracle Cloud Infrastructure Object Storage Classic,

E37261-43

Primary Author: Oracle Corporation

Contents

Preface

Audience vi
Related Resources vi
Conventions vii

1 Getting Started with Object Storage Classic

Interfaces to Object Storage Classic 1-1
Tasks Supported by the Interfaces of Object Storage Classic 1-3
Workflow for Getting Started with Object Storage Classic 1-6
About Replication Policy for Your Object Storage Classic Instance 1-7
Authenticating Access to Object Storage Classic 1-7
- Authenticating Access When Using the REST API 1-8
- Authenticating Access When Using the Java Library 1-12

About Oracle Cloud Infrastructure Object Storage Classic 1-14
- Oracle Cloud Infrastructure Object Storage Classic vs. Other Storage Solutions 1-15
- Features of Oracle Cloud Infrastructure Object Storage Classic 1-16
- Architectural Overview 1-20

Before You Begin with Object Storage Classic 1-21
How to Begin with Object Storage Classic Subscriptions 1-21
Accessing Object Storage Classic 1-21
- Accessing Oracle Cloud Infrastructure Object Storage Classic Using the Web Console 1-22
- Accessing Oracle Cloud Infrastructure Object Storage Classic Using the REST API 1-22
- About REST URLs for Oracle Cloud Infrastructure Object Storage Classic Resources 1-23
- Accessing Oracle Cloud Infrastructure Object Storage Classic Using the Java Library 1-33
- Accessing Oracle Cloud Infrastructure Object Storage Classic Using Oracle Cloud Infrastructure Storage Software Appliance 1-33
- Accessing Oracle Cloud Infrastructure Object Storage Classic Using File Transfer Manager API 1-34

Accessing Oracle Cloud Infrastructure Object Storage Classic Using the FTM CLI 1-34
Accessing Objects in Oracle Cloud Infrastructure Object Storage Classic Using Temporary URLs 1-34
About Object Storage Classic Roles and Users 1-34
About Access Control Lists 1-36
Create a Custom Role for Cloud Accounts with Identity Cloud Service 1-37

2 Managing Containers in Object Storage Classic

Typical Workflow for Managing Containers 2-1
Creating Containers 2-2
Creating Archive Containers 2-5
Listing Containers 2-8
Setting a Container-Specific Policy 2-11
Setting a Container-Specific Policy Using the Web Console 2-12
Setting a Container-Specific Policy Using the REST API 2-13
Deleting Containers 2-19
Enabling Server-Side Encryption 2-22
Getting Container Metadata 2-27
Deleting Container Metadata 2-31
Setting Container Metadata 2-34
Setting Container ACLs 2-34
Setting Container Quotas 2-39
Setting Custom Metadata for Containers 2-41
Enabling CORS for a Container 2-44
Making Objects in a Container Immutable 2-48

3 Managing Objects in Object Storage Classic

Typical Workflow for Managing Objects 3-1
Roles Required for Managing Objects in Object Storage Classic 3-2
Listing Objects in a Container 3-2
Creating Objects 3-6
Creating a Single Object 3-7
Uploading Multiple Objects in a Single Operation 3-11
Uploading Large Objects 3-13
Making an Object Immutable 3-19
Getting Object Metadata 3-21
Finding Out the Status of Objects in an Archive Container 3-26
Restoring Archived Objects 3-29
Tracking Restoration of an Object in an Archive Container 3-32

Downloading Objects 3-36
Downloading an Object 3-36
Downloading a Large Object 3-40
Downloading an Object Using a Temporary URL 3-42
Deleting Objects 3-46
Deleting a Single Object 3-46
Deleting a Large Object 3-49
Deleting Multiple Objects in a Single Operation 3-51
Copying Objects 3-54
Encrypting Objects 3-57
Updating Object Metadata 3-58
Updating Custom Metadata for Objects 3-58
Scheduling Automatic Deletion of Objects 3-63

4 Managing Your Object Storage Classic Account

Setting Account Metadata 4-1
Getting Account Metadata 4-4
Deleting Account Metadata 4-6
Enabling Audit Logging 4-9

5 Frequently Asked Questions for Object Storage Classic

6 Troubleshooting Object Storage Classic

A Error Code Reference for Object Storage Classic

B Certified Third-Party Products

C Java Code Samples for Using the File Transfer Manager API

D Character Restrictions

E About Replication Policy for Accounts Created Before March 2018

Preface

Using Oracle Cloud Infrastructure Object Storage Classic describes how to access and use Oracle Cloud Infrastructure Object Storage Classic APIs to store and manage content in the cloud.

Topics:

- Audience
- Related Resources
- Conventions

Audience

Using Oracle Cloud Infrastructure Object Storage Classic is intended for administrators and users who want to store, access, and manage files and unstructured data in the cloud.

Related Resources

For more information, see these Oracle resources:

- *REST API for Standard Storage in Oracle Cloud Infrastructure Object Storage Classic*
- *REST API for Archive Storage in Oracle Cloud Infrastructure Object Storage Classic*
- *REST API for Identity in Oracle Cloud Infrastructure Object Storage Classic*
- *Using Oracle Cloud Infrastructure Storage Software Appliance*
- *Licensing Information User Manual for Oracle Cloud Infrastructure Object Storage Classic*
- *Java API Reference for Oracle Cloud Infrastructure Object Storage Classic File Transfer Manager*
- *Oracle Cloud Java API Reference for Oracle Cloud Infrastructure Object Storage Classic*
- *Command-Line Reference for Oracle Cloud Infrastructure Object Storage Classic*
- *Oracle Cloud Known Issues for Oracle Cloud Infrastructure Object Storage Classic*
- *Oracle Cloud What's New for Oracle Cloud Infrastructure Object Storage Classic*

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

Getting Started with Object Storage Classic

This section describes how to get started with Oracle Cloud Infrastructure Object Storage Classic.

Topics:

- About Oracle Cloud Infrastructure Object Storage Classic
- Before You Begin with Oracle Cloud Infrastructure Object Storage Classic
- Workflow for Getting Started with Oracle Cloud Infrastructure Object Storage Classic
- How to Begin with Oracle Cloud Infrastructure Object Storage Classic Subscriptions
- About Replication Policy for Your Object Storage Classic Instance
- Accessing Oracle Cloud Infrastructure Object Storage Classic
- About Oracle Cloud Infrastructure Object Storage Classic Roles and Users
- About Access Control Lists

For the definitions of the terms used in this and other documents in the Oracle Cloud library, see Oracle Cloud Terminology in *Getting Started with Oracle Cloud*.

Interfaces to Object Storage Classic

The following table summarizes the Oracle-provided interfaces to Oracle Cloud Infrastructure Object Storage Classic.

> **Note:**
>
> Oracle has certified certain third-party products for use with Oracle Cloud Infrastructure Object Storage Classic. For more information, see Certified Third-Party Products.

Interface	Description	More Information
Web Console (Not available on Oracle Cloud at Customer)	A web-based console to manage your service instances, containers, and objects.	• Accessing Object Storage Classic • Tasks Supported by the Interfaces of Object Storage Classic

Interface	Description	More Information
RESTful Web Service API	Oracle Cloud Infrastructure Object Storage Classic provides REST APIs that are compatible with OpenStack Swift. The following major additions have been made: • Centralized identity management across Oracle Cloud • Centralized reporting of usage metrics • Global namespace URL to access the service • Archiving and restoring objects Oracle Cloud Infrastructure Object Storage Classic *does not* support the following OpenStack Swift features: • Object versioning • Static website support • Container synchronization • Form post • Account ACLs • Rate limits The RESTful web service API is available only over HTTPS.	• About REST URLs for Oracle Cloud Infrastructure Object Storage Classic Resources • Accessing Oracle Cloud Infrastructure Object Storage Classic Using the REST API • *REST API for Standard Storage in Oracle Cloud Infrastructure Object Storage Classic* • *REST API for Archive Storage in Oracle Cloud Infrastructure Object Storage Classic* • *REST API for Identity in Oracle Cloud Infrastructure Object Storage Classic* • Tasks Supported by the Interfaces of Object Storage Classic
Oracle Cloud Infrastructure Storage Software Appliance (Not available on Oracle Cloud at Customer)	Oracle Cloud Infrastructure Storage Software Appliance is a tool that you can install on-premises and then use to easily connect your on-premises applications and workflows to Oracle Cloud Infrastructure Object Storage Classic.	• Getting Started with Oracle Cloud Infrastructure Storage Software Appliance in *Using Oracle Cloud Infrastructure Storage Software Appliance* • Tasks Supported by the Interfaces of Object Storage Classic
Oracle Cloud Infrastructure Object Storage Classic File Transfer Manager API	Oracle Cloud Infrastructure Object Storage Classic File Transfer Manager API is a Java library that provides a simple interface to upload or download individual and multiple objects of various sizes to `standard` and `archive` containers in Oracle Cloud Infrastructure Object Storage Classic.	• Accessing Oracle Cloud Infrastructure Object Storage Classic Using File Transfer Manager API • Java API Reference for Oracle Cloud Infrastructure Object Storage Classic File Transfer Manager • Java Code Samples for Using the File Transfer Manager API • Tasks Supported by the Interfaces of Object Storage Classic

Interface	Description	More Information
Oracle Cloud Infrastructure Object Storage Classic CLI	The Oracle Cloud Infrastructure Object Storage Classic CLI is a cross-platform Java-based command line tool that you can use to upload and download objects of various sizes to `standard` and `archive` containers in Oracle Cloud Infrastructure Object Storage Classic.	• Command-Line Reference for Oracle Cloud Infrastructure Object Storage Classic • Tasks Supported by the Interfaces of Object Storage Classic
Java Library	A Java library that wraps the RESTful web service API. The Java library supports most of the major features of the RESTful web service API. The Java library also provides client-side encryption utilities.	• Tasks Supported by the Interfaces of Object Storage Classic • Accessing Oracle Cloud Infrastructure Object Storage Classic Using the Java Library • Java API Reference for Oracle Cloud Infrastructure Object Storage Classic

Tasks Supported by the Interfaces of Object Storage Classic

Use the following table as a guide to select the interface to Oracle Cloud Infrastructure Object Storage Classic. *Yes* indicates that the task can be performed using the interface.

Task	Web Console (Not available on Oracle Cloud at Customer)	RESTful API	Java Library	Storage Software Appliance 1.4 (Not available on Oracle Cloud at Customer)	FTM API (version 2.4)	FTM CLI (version 2.4)
Setting account metadata	Yes	Yes	No	No	Yes	Yes
Getting account metadata	Yes	Yes	No	No	Yes	Yes
Creating containers	Yes	Yes	Yes	Yes See **Note**.	Yes	Yes
Creating `archive` containers	Yes	Yes	No	Yes	Yes	Yes
Deleting containers	Yes	Yes	Yes	No	Yes	Yes
Listing containers	Yes	Yes	Yes	No	Yes	Yes

Task	Web Console (Not available on Oracle Cloud at Customer)	RESTful API	Java Library	Storage Software Appliance 1.4 (Not available on Oracle Cloud at Customer)	FTM API (version 2.4)	FTM CLI (version 2.4)
Setting custom metadata for containers	Yes	Yes	Yes	No	Yes	Yes
Setting container ACLs	Yes	Yes	Yes	No	Yes	Yes
Setting container quotas	Yes	Yes	No	No	Yes	Yes
Setting a replication policy for a container	No	Yes	No	No	Yes	Yes
Getting container metadata	Yes	Yes	No	No	Yes	Yes
Making objects in a container immutable	No	Yes	No	No	Yes	Yes
Uploading files to `standard` containers	Yes	Yes	Yes	Yes	Yes	Yes
Uploading files to `archive` containers	Yes	Yes	No	No	Yes	Yes
Uploading multiple objects in parallel	No	No	No	Yes	Yes	Yes
Listing objects	Yes	Yes	Yes	Yes	Yes	Yes
Deleting objects	Yes	Yes	Yes	Yes	Yes	Yes
Uploading multiple files in one operation	No	Yes	No	No	Yes	Yes

Task	Web Console (Not available on Oracle Cloud at Customer)	RESTful API	Java Library	Storage Software Appliance 1.4 (Not available on Oracle Cloud at Customer)	FTM API (version 2.4)	FTM CLI (version 2.4)
Deleting multiple objects in one operation	No	Yes	No	No	Yes	Yes
Downloading objects	Yes	Yes	Yes	Yes	Yes	Yes
Downloading multiple objects	No	No	No	Yes	Yes	Yes
Updating custom metadata for objects	No	Yes	Yes	No	Yes	Yes
Updating special metadata for objects	No	Yes	No	No	Yes	Yes
Copying objects	No	Yes	No	No	Yes	Yes
Encrypting objects	No	No	Yes	Yes	Yes	Yes
Restoring archive objects	Yes	Yes	No	Yes	Yes	Yes
Resuming interrupted uploads	No	No	No	Yes	Yes	Yes
Ensuring the integrity of uploaded data	No	No	No	Yes	Yes	Yes
Supporting NFS protocol	No	No	No	Yes	No	No
Monitoring upload activity	No	No	No	Yes	Yes	No
Rotating encryption keys	No	No	Yes	Yes	Yes	Yes
Local disk caching	No	No	No	Yes	No	No

Task	Web Console (Not available on Oracle Cloud at Customer)	RESTful API	Java Library	Storage Software Appliance 1.4 (Not available on Oracle Cloud at Customer)	FTM API (version 2.4)	FTM CLI (version 2.4)
Network throttling	No	No	No	Yes	No	No
Automatic segmentation of large files	No	No	No	Yes	Yes	Yes

> **Note:**
>
> During an upload operation, if the container specified doesn't exist, then it is created. But you can't create an empty container.

Workflow for Getting Started with Object Storage Classic

 This topic does not apply to Oracle Cloud Machine.

Use the following table as a guide to the workflow for getting started with Oracle Cloud Infrastructure Object Storage Classic:

Task	Description	More Information
Request a trial or purchase a subscription to Oracle Cloud Infrastructure Object Storage Classic	Provide your information, and sign up for a free trial or purchase a subscription. After activation, create accounts for your users and assign them appropriate privileges and roles.	How to Begin with Oracle Cloud Infrastructure Object Storage Classic Subscriptions
Monitor the service	Check on the day-to-day operation of your service, monitor performance, and review important notifications.	Managing and Monitoring Oracle Cloud Services in *Managing and Monitoring Oracle Cloud*
Access the service	Access the service through the REST API or Java library interface.	Accessing Oracle Cloud Infrastructure Object Storage Classic

About Replication Policy for Your Object Storage Classic Instance

 This topic does not apply to Oracle Cloud at Customer.

Topics

- About Georeplication
- Verifying the Replication Policy for Your Service Instance

About Georeplication

If your account was created **after** March 2018, then once your Oracle Cloud Infrastructure Object Storage Classic subscription is activated, the replication policy for your instance is set to `any`, by default.

With the `any` policy set by default at the service instance level, you can now create a container and set any authorized replication policy to the container.

If your account was created **before** March 2018, then see About Replication Policy for Accounts Created Before March 2018.

Verifying the Replication Policy for Your Service Instance

Using the Web Console

To find out the replication policy that is set for your Oracle Cloud Infrastructure Object Storage Classic service instance, sign in to the web console. Expand **Account Information**. The details of your service instances are displayed in the **Account Information** pane. Look for the **Georeplication Policy** field.

Using the REST API

Send a `HEAD` request to view the replication policy details for your service instance. Look for the header `X-Account-Meta-Policy-Georeplication` in the output. See Getting Account Metadata.

Authenticating Access to Object Storage Classic

Oracle Cloud Infrastructure Object Storage Classic requires authentication when executing operations against your service instance. Authentication is provided to the service instance in the form of an authentication token.

You request an authentication token from the service by sending your user credentials to the service. Authentication tokens are temporary and expire in 30 minutes. This is a session time out and not an idle time out, which means that tokens expire even if they are in use. You must include your current authentication token with every operation against your service instance.

Topics:

- Authenticating Access When Using REST API

- Authenticating Access When Using the Java Library

Authenticating Access When Using the REST API

To request an authentication token, send a `GET` request to the authentication URL for your account. You'll need your sign-in credentials to access your account and obtain the authentication URL to perform any operation on the resources.

Topics:

- Obtaining the Authentication URL
- Using Your Account Details
- Requesting an Authentication Token

Obtaining the Authentication URL

1. Sign in to your Oracle Cloud account.
   If you see Infrastructure Classic at the top of the page when you sign in to Oracle Cloud, then you are using the Infrastructure Classic Console to access your services and your subscription does not support access to the Infrastructure Console. See Signing In to Your Cloud Account in *Getting Started with Oracle Cloud*.

   If you can access the service from the Infrastructure Console, see Signing In to the Console in Oracle Cloud Infrastructure documentation.
2. If you can access the service from the Infrastructure Console, perform the following steps to obtain the REST Endpoint URL:
   a. On the navigation menu, under **More Oracle Cloud Services**, point to **Classic Infrastructure Services**, and then click **Storage Classic**.
      The **Oracle Cloud Infrastructure Object Storage Classic** console is displayed.
   b. Click the **Account** tab.
   c. Note the Rest Endpoint URL which is displayed in the **Rest Endpoint** field.
      You can construct the authentication URL from the Rest Endpoint URL.
3. If you can access the service from the Infrastructure Classic Console, perform the following steps to obtain the REST Endpoint URL and authentication URL:
   a. On the dashboard, look for **Storage Classic**.
   b. Select **View Details** from the **Actions** menu. Alternatively, click the **Storage Classic** link on the **Dashboard** page.
      On the resulting page, the details of your Oracle Cloud Infrastructure Object Storage Classic account are displayed.
   c. Note the authentication URL, which is displayed in the **Auth V1 Endpoint** field under the **Additional Information** section.
      For example: `https://acme.storage.oraclecloud.com/auth/v1.0`
   d. If the **Auth V1 Endpoint** field is not displayed, then you must construct the authentication URL from the REST Endpoint URL. Note the REST Endpoint URL, which is displayed in the **REST Endpoint** field under the **Additional Information** section.
4. To construct the authentication URL when the authentication URL is not directly available in the web console:

a. Let's consider that the REST Endpoint URL that you have noted is `https://acme.storage.oraclecloud.com/v1/Storage-acme`.

b. Delete the following portion of the REST Endpoint URL: `v1/Storage-acme`

   The edited URL is: `https://acme.storage.oraclecloud.com/`

c. Append the following to the edited URL: `auth/v1.0`

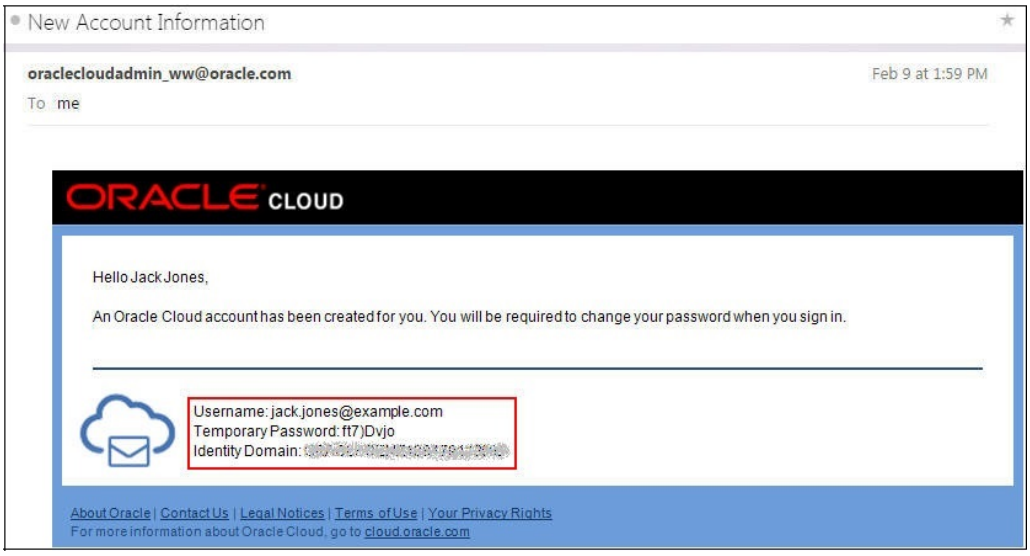
   The equivalent authentication URL would be `https://acme.storage.oraclecloud.com/auth/v1.0`.

Using Your Account Details

You'll need your sign-in credentials to access your account and perform any operation on the resources.

When you sign up for the account, you'll receive a Welcome mail with your sign-in credentials.

Make a note of your sign-in credentials, as shown in the following example:



If you don't have your Welcome email, ask your account administrator for your Oracle Cloud user name, password, and identity domain.

> **Note:**
>
> For traditional accounts, you can request Oracle Cloud to send the email to the administrator again. See Resending Welcome Email with Administrator Sign-in Credentials.

If your user credentials are not authenticated, the service returns an HTTP response with a status code of 401 and no authentication token is returned.

If the credentials are authenticated, the service either returns the currently active authentication token or generates a new authenticate token. Authentication tokens are returned as the value of the HTTP header `X-Auth-Token` in the HTTP response. Requesting an authentication token with credentials that already have an active authentication token will not extend the expiration time of the existing authentication token.

Requesting an Authentication Token

When you send the `GET` request to the authentication URL, include the user credentials in the following headers:

- Depending on the REST Endpoint URL in your account:
  - If you use the **REST Endpoint** URL, then include the header `X-Storage-User`: `Storage-`*`account_name`*`:`*`userName`*
  - If you use the **REST Endpoint URL (Permanent)**, then include the header `X-Storage-User`: `Storage-`*`GUID`*`:`*`userName`*

> **Note:**
>
> The REST Endpoint (Permanent) URL is displayed for the accounts created after November 2017.

- `X-Storage-Pass:` *`password`*
- `http://`*`account_name`*`.storage.oraclecloud.com/auth/v1.0` is the authentication URL.

> **Note:**
>
> For traditional accounts, the identity domain name and the account name are the same.

cURL Command Examples

Sample Cloud account with the following details:

- Account name: `acme`
  For traditional accounts, account name and identity domain name are the same.
- REST Endpoint URL: `https://acme.storage.oraclecloud.com/v1/Storage-acme`
- REST Endpoint (Permanent) URL: `https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/Storage-7b16fede61e1417ab83eb52e06f0e365`
- Authentication URL: `https://acme.storage.oraclecloud.com/auth/v1.0`

cURL Command Examples – Requesting an Authentication Token:

- Using the account name `acme` obtained from the **REST Endpoint** URL in the header `X-Storage-User`:

```
curl -v -X GET \
     -H "X-Storage-User: Storage-acme:myUsername" \
```

```
    -H "X-Storage-Pass: myPassword" \
    https://acme.storage.oraclecloud.com/auth/v1.0
```

The following is an example of the output of this command:

```
> GET /auth/v1.0 HTTP/1.1
> Host: acme.storage.oraclecloud.com
> Accept: */*
> X-Storage-User: Storage-acme:myUsername
> X-Storage-Pass: myPassword

< HTTP/1.1 200 OK
< X-Storage-Url: https://acme.storage.oraclecloud.com/v1/Storage-acme
< X-Storage-Token: AUTH_tk209f7f2ea1265a0d3f29d28a2dc8ced6
< X-Auth-Token: AUTH_tk209f7f2ea1265a0d3f29d28a2dc8ced6
< X-Trans-Id: txba4aa8f776164c33b7aa587554c29fb6
< Content-Length: 0
< Cache-Control: no-cache
< Pragma: no-cache
< Content-Type: text/plain
< Content-Language: en
```

- Using the GUID `7b16fede61e1417ab83eb52e06f0e365` obtained from the **REST Endpoint (Permanent)** URL in the header `X-Storage-User`:

```
curl -v -X GET \
    -H "X-Storage-User: Storage-7b16fede61e1417ab83eb52e06f0e365:myUsername" \
    -H "X-Storage-Pass: myPassword" \
    https://acme.storage.oraclecloud.com/auth/v1.0
```

The following is an example of the output of this command:

```
> GET /auth/v1.0 HTTP/1.1
> Host: acme.storage.oraclecloud.com
> Accept: */*
> X-Storage-User: Storage-7b16fede61e1417ab83eb52e06f0e365:myUsername
> X-Storage-Pass: myPassword

< HTTP/1.1 200 OK
< X-Storage-Url: https://acme.storage.oraclecloud.com/v1/
Storage-7b16fede61e1417ab83eb52e06f0e365
< X-Storage-Token: AUTH_tk209f7f2ea1265a0d3f29d28a2dc8ced6
< X-Auth-Token: AUTH_tk209f7f2ea1265a0d3f29d28a2dc8ced6
< X-Trans-Id: txba4aa8f776164c33b7aa587554c29fb6
< Content-Length: 0
< Cache-Control: no-cache
< Pragma: no-cache
< Content-Type: text/plain
< Content-Language: en
```

To use your authentication token, include it as the value of the `X-Auth-Token` HTTP header in every HTTP request to the service instance. If your authentication token is not valid, or has expired, the service returns an HTTP response with the status code 401 and the requested operation will fail. If the authentication token has expired, you must request a new token. If you are reading publicly accessible objects, you do not need to provide an authentication token in your HTTP request; anonymously accessible objects do not need an authentication token.

cURL Command Examples – Storing an Object in an Account Using an Authentication Token:

- Using the REST Endpoint URL obtained from the **REST Endpoint** field:

```
curl -v -X PUT \
     -H "X-Auth-Token: AUTH_tk209f7f2ea1265a0d3f29d28a2dc8ced6" \
     -d "Hello, World!" \
     https://acme.storage.oraclecloud.com/v1/Storage-acme/myContainer/myObject
```

The following is an example of the output of this command:

```
> PUT /v1/Storage-acme/myContainer/myObject HTTP/1.1
> Host: acme.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tk209f7f2ea1265a0d3f29d28a2dc8ced6
> Content-Length: 13
> Content-Type: application/x-www-form-urlencoded

< HTTP/1.1 201 Created
< Content-Length: 0
< Etag: 65a8e27d8879283831b664bd8b7f0ad4
< Content-Type: text/html; charset=UTF-8
< X-Trans-Id: tx287a1a8e33cc45e5a1431817e3e87621
< Cache-Control: no-cache
< Pragma: no-cache
< Content-Language: en
```

- Using the Service Permanent REST Endpoint URL obtained from the **REST Endpoint (Permanent)** field:
  This cURL command example applies to the accounts created after November 2017.

```
curl -v -X PUT \
     -H "X-Auth-Token: AUTH_tk209f7f2ea1265a0d3f29d28a2dc8ced6" \
     -d "Hello, World!" \
     https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/
Storage-7b16fede61e1417ab83eb52e06f0e365/myContainer/myObject
```

The following is an example of the output of this command:

```
> PUT /v1/Storage-7b16fede61e1417ab83eb52e06f0e365/myContainer/myObject HTTP/1.1
> Host: storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tk209f7f2ea1265a0d3f29d28a2dc8ced6
> Content-Length: 13
> Content-Type: application/x-www-form-urlencoded

< HTTP/1.1 201 Created
< Content-Length: 0
< Etag: 65a8e27d8879283831b664bd8b7f0ad4
< Content-Type: text/html; charset=UTF-8
< X-Trans-Id: tx287a1a8e33cc45e5a1431817e3e87621
< Cache-Control: no-cache
< Pragma: no-cache
< Content-Language: en
```

Authenticating Access When Using the Java Library

When using the Java library, you do not need to manually request and use an authentication token; the Java library will automatically request and use it.

The Java library will also try to request a new authentication token when the current authentication token expires.

When using the Java library, you must pass your user credentials to a `CloudStorageConfig` object. The `CloudStorageConfig` object is then passed to a `CloudStorageFactory` object and, upon successful authentication, ultimately returns a `CloudStorage` object. The `CloudStorage` object provides the methods for all supported functionality to the service instance.

Code Examples

Sample Cloud account with the following details:

- Account name: `acme`
  For traditional accounts, account name and identity domain name are the same.
- REST Endpoint URL: `https://acme.storage.oraclecloud.com/v1/Storage-acme`
- REST Endpoint (Permanent) URL: `https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/Storage-7b16fede61e1417ab83eb52e06f0e365`

> **Note:**
>
> The REST Endpoint (Permanent) URL is displayed for the accounts created after November 2017.

The following are examples of Java code for providing user credentials and creating an object. The values used in the examples are illustrative. While developing Java code to use Oracle Cloud Infrastructure Object Storage Classic, ensure that the code complies with the IT security standards and requirements of your organization.

- Using the REST Endpoint URL obtained from the **REST Endpoint** field:

```
CloudStorageConfig myConfig = new CloudStorageConfig();
myConfig.setServiceName("Storage-acme")
   .setUsername("john.doe@example.com")
   .setPassword("Password".toCharArray())
   .setServiceUrl("https://acme.storage.oraclecloud.com");
CloudStorage myConnection = CloudStorageFactory.getStorage(myConfig);
FileInputStream fis = new FileInputStream("hello_world.txt");
myConnection.storeObject("MyContainer", "hello_world.txt", "text/plain", fis);
```

- Using the Service Permanent REST Endpoint URL obtained from the **REST Endpoint (Permanent)** field:
  This cURL command example applies to the accounts created after November 2017.

```
CloudStorageConfig myConfig = new CloudStorageConfig();
myConfig.setServiceName("Storage-7b16fede61e1417ab83eb52e06f0e365")
   .setUsername("john.doe@example.com ")
   .setPassword("Password".toCharArray())
   .setServiceUrl("https://
storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com");
CloudStorage myConnection = CloudStorageFactory.getStorage(myConfig);
FileInputStream fis = new FileInputStream("hello_world.txt");
myConnection.storeObject("MyContainer", "hello_world.txt", "text/plain", fis);
```

To identify the service URL and service name (for the `setServiceName` and `setServiceUrl` methods in this code example) of your Oracle Cloud Infrastructure Object Storage Classic instance:

1. Sign in to your Oracle Cloud account.

   If you see Infrastructure Classic at the top of the page when you sign in to Oracle Cloud, then you are using the Infrastructure Classic Console to access your services and your subscription does not support access to the Infrastructure Console. See Signing In to Your Cloud Account in *Getting Started with Oracle Cloud*.

   If you can access the service from the Infrastructure Console, see Signing In to the Console in Oracle Cloud Infrastructure documentation.

2. If you can access the service from the Infrastructure Console, perform the following steps to obtain the REST Endpoint URL:

   a. On the navigation menu, under **More Oracle Cloud Services**, point to **Classic Infrastructure Services**, and then click **Storage Classic**.

      The **Oracle Cloud Infrastructure Object Storage Classic** console is displayed.

   b. Click the **Account** tab.

   c. Note the Rest Endpoint URL which is displayed in the **Rest Endpoint** field.

3. If you can access the service from the Infrastructure Classic Console, perform the following steps to obtain the REST Endpoint URL:

   a. On the Dashboard, look for **Storage Classic**.

   b. Select **View Details** from the **Actions** menu. Alternatively, click the **Storage Classic** link on the **Dashboard**.
      On the resulting page, the details of your Oracle Cloud Infrastructure Object Storage Classic account are displayed.

   c. Note the REST Endpoint URL, which is displayed in the **REST Endpoint** field in the **Additional Information** section. For example: `https://acme.storage.oraclecloud.com/v1/Storage-acme`.

4. You can identify the value for the **service URL** and **service name** of your Oracle Cloud Infrastructure Object Storage Classic instance from the REST Endpoint URL.

   For example: `https://acme.storage.oraclecloud.com/v1/Storage-acme` is the REST Endpoint URL.

   In this example, the following are the service URL and name:

   - Service URL: `https://acme.storage.oraclecloud.com`
   - Service Name: `Storage-acme`

About Oracle Cloud Infrastructure Object Storage Classic

Oracle Cloud Infrastructure Object Storage Classic is an Infrastructure as a Service (IaaS) product, which provides an enterprise-grade, large-scale, object storage solution for files and unstructured data.

Topics:

- Oracle Cloud Infrastructure Object Storage Classic vs. Other Storage Solutions
- Features of Oracle Cloud Infrastructure Object Storage Classic

- Architectural Overview

Oracle Cloud Infrastructure Object Storage Classic vs. Other Storage Solutions

Storage is a fundamental requirement for any enterprise application workload. Traditional storage solutions pose certain scalability, performance, and management challenges that Oracle Cloud Infrastructure Object Storage Classic helps you overcome.

- With *direct attached storage*, such as the hard disk drive in a laptop, the operating system underlying the applications manages data storage, retrieval, and organization through a *file system*, which is a schema that the operating system uses to organize data on locally attached disks. Direct attached storage provides convenient, low latency, durable storage. However, because storage capacity is spread between isolated devices, direct attached storage does not scale well.
- In *network-attached storage* (NAS), the storage device is physically separate from the servers hosting the applications. To the application hosts, the storage device is available as a network drive. A network file system on the storage device manages data storage, retrieval, and organization. NAS enables applications running on multiple hosts to share storage. It enables centralized management of storage resources and high performance over a local network. But this architecture is feasible only within a limited geographical area, and it offers limited room for scaling. Like in directly attached storage, in NAS as well, applications rely on the underlying operating system and on the network file system of the storage device.
- *Block storage* enables applications such as OLTP databases that have high IOPS (input/output operations per second) requirements to store and retrieve data efficiently, by bypassing the host operating system and interacting directly with virtual block devices. Chunks of data are stored in blocks, each with an address, but with no other metadata. Applications decide where data is stored, and they retrieve data by calling the appropriate block addresses directly. Block storage optimizes storage for IOPS and block-based access and provides POSIX-compliant file systems for Oracle Cloud Infrastructure Object Storage Classic instances. It is limited in terms of scalability and does not support the definition of granular metadata for stored data.
- *Object storage* provides an optimal blend of performance, scalability, and manageability when storing large amounts of unstructured data. Multiple storage nodes form a single, shared, horizontally scalable pool in which data is stored as *objects* (blobs of data) in a flat hierarchy of *containers*. Each object stores data, the associated metadata, and a unique ID. You can assign custom metadata to containers and objects, making it easier to find, analyze, and manage data. Applications use the unique object IDs to access data directly via REST API calls. Object storage is simple to use, performs well, and scales to a virtually unlimited capacity.

Oracle Cloud Infrastructure Object Storage Classic provides a low cost, reliable, secure, and scalable object-storage solution for storing unstructured data and accessing it anytime from anywhere. It is ideal for data backup, archival, file sharing, and for storing large amounts of unstructured data like logs, sensor-generated data, and VM images.

Features of Oracle Cloud Infrastructure Object Storage Classic

Oracle Cloud Infrastructure Object Storage Classic is an Infrastructure as a Service (IaaS) product, which provides an enterprise-grade, large-scale, object storage solution for files and unstructured data.

You can use Oracle Cloud Infrastructure Object Storage Classic to back up content to an off-site location, programmatically store and retrieve content, and share content with peers. The following are the features of the service:

Object storage

Oracle Cloud Infrastructure Object Storage Classic stores data as objects within a flat hierarchy of containers.

- An object is most commonly created by uploading a file. It can also be created from ephemeral unstructured data. Objects are created within a container. A single object can hold up to 5 GB of data, but multiple objects can be linked together to hold more than 5 GB of contiguous data.
- A container is a user-created resource, which can hold an unlimited number of objects, unless you specify a quota for the container. Note that containers cannot be nested.

Custom metadata can be defined for both objects and containers.

Replication within the data center

All objects or containers created in Oracle Cloud Infrastructure Object Storage Classic are replicated to multiple separate storage nodes in the data center. If one of the nodes fails, other copies of the object or container will continue to be available.

Note that by default, data is *eventually consistent* across the nodes in the data center. When an object or container is created or modified, it is not replicated instantaneously to the other nodes. Until the replication is completed, a container or an object's data may not be consistent across the nodes. Over time, all changes to all objects or containers are replicated, and the data becomes consistent across the nodes.

Automatic error detection and healing

Object copies are actively scanned for data corruption. If a bad copy is found, it is replaced, automatically, with a new copy.

Fine-grained read/write access control to containers

Read and write access to an object is controlled via access control lists for its container. Each container can be assigned its own read and write access control lists. By default, access to a container and its objects is private (that is only the user who created the container can access it), but read access can be made public if required.

Multiple Oracle-provided interfaces

You can access Oracle Cloud Infrastructure Object Storage Classic using any of the following interfaces:

- Web Console
  You can use a web-based graphical user interface to easily manage containers and objects in your service instances.

(Not available on Oracle Cloud at Customer)

- RESTful Web Service API
  Your applications can access Oracle Cloud Infrastructure Object Storage Classic programmatically using calls to a RESTful web service, which is compatible with OpenStack Swift. The service can be accessed from anywhere over the Internet, at any time, and from any device.
- Java Library
  A Java library that wraps the RESTful web service is available. No special hardware is required to start using the service.
- Oracle Cloud Infrastructure Storage Software Appliance
  Oracle Cloud Infrastructure Storage Software Appliance is a tool that you can install on-premises and then use to easily connect your on-premises applications and workflows to Oracle Cloud Infrastructure Object Storage Classic.
- Oracle Cloud Infrastructure Object Storage Classic File Transfer Manager API
  The Oracle Cloud Infrastructure Object Storage Classic File Transfer Manager API is a Java library that provides a simple interface to upload or download individual and multiple objects to `standard` and `archive` containers in Oracle Cloud Infrastructure Object Storage Classic.
- Oracle Cloud Infrastructure Object Storage Classic File Transfer Manager CLI
  The Oracle Cloud Infrastructure Object Storage Classic File Transfer Manager CLI is a cross-platform Java-based command line tool that you can use to manage containers and objects in Oracle Cloud Infrastructure Object Storage Classic.
- Temporary URL
  You can create time-limited temporary URLs to provide secure, temporary access to download an object in your Oracle Cloud Infrastructure Object Storage Classic account.

See Interfaces to Object Storage Classic.

Certified third-party clients

(Not available on Oracle Cloud at Customer)
A wide range of third-party products are certified for use as gateways and clients to store and manage your data in Oracle Cloud Infrastructure Object Storage Classic. Oracle continues to add to the list of certified third-party clients.

See Certified Third-Party Products.

Replication to a geographically distant data center

(Not available on Oracle Cloud at Customer)
For the accounts created **after** March 2018, the replication policy is set to `any`, by default, in the account.

With the `any` policy set by default at the account level, you can now create a container and set any authorized replication policy to the container. See About Replication Policy for Your Object Storage Classic Instance.

For the accounts created **before** March 2018:

After activating your Oracle Cloud Infrastructure Object Storage Classic subscription, you must choose a *replication policy*, which defines your primary data center and also specifies whether your data should be replicated to a geographically distant (secondary) data center. Data is written to the primary data center and replicated

asynchronously to the secondary data center. The primary and secondary data centers are eventually consistent. In addition to being billed for storage capacity used at each data center, you will also be billed for bandwidth used during replication between data centers.

> **Note:**
>
> After you select a replication policy, you cannot change the policy.
>
> Data in containers of the `Archive` storage class is currently not replicated to the secondary data center, even if a replication policy is selected for the account.

For information about the available policies, see About Replication Policy for Accounts Created Before March 2018.

Global namespace URL to access the service

(Not available on Oracle Cloud at Customer)
Regardless of the data center where your service instance is provisioned, you can access Oracle Cloud Infrastructure Object Storage Classic by using a global namespace URL. Requests sent to the global namespace URL are routed to the data center where your service instance is provisioned.

> **Note:**
>
> You can find the global namespace URL for your service instance in Oracle Cloud Infrastructure Classic Console.

- **For accounts created before November 2017** The URL is displayed in the **REST Endpoint** field under the **Additional Information** section.
- **For accounts created after November 2017** Two REST Endpoint URLs are displayed under the **Additional Information** section. See About REST URLs for Oracle Cloud Infrastructure Object Storage Classic Resources.

If the primary data center is unavailable, read requests to the global namespace URL are routed to the secondary data center. This is known as *failover*. While the primary data center is unavailable, write requests will fail with the 403 – Forbidden error. When the primary data center is available again, all requests to the global namespace URL are routed to the primary data center. This is known as *failback*.

> **Note:**
>
> During failover and failback, the DNS records of your service instance's global namespace URL are updated to point to the currently active data center. But for a short while after the DNS records are updated, usually a few minutes, requests to the global namespace URL may return a 500-series error. This error occurs because propagation of the DNS changes across all the intermediate nodes on the Internet between your client and Oracle Public Cloud may not yet have been completed. To continue using Oracle Cloud Infrastructure Object Storage Classic during DNS propagation, you can send requests directly to the URL of the active data center (see About REST URLs for Oracle Cloud Infrastructure Object Storage Classic Resources). In the following situations, DNS propagation can take longer than a few minutes:
>
> - ISP DNS resolvers set their own DNS TTL values. Check about this with your ISP provider.
> - Long-running network clients may cache resolved DNS records. For information about tuning the DNS host name caching behavior on your client, see the documentation for your programming platform.

Low-cost data archival

(Not available on Oracle Cloud at Customer)
In metered accounts, you can create containers of two *storage classes*, `Standard` (default) and `Archive`. You can use `Archive` containers to store large data sets that you don't need to access frequently, at a fraction of the cost of storing data in `Standard` containers. Note that to download data stored in `Archive` containers, you must first *restore* the objects. The restoration process can take up to four hours depending on the size of the object. A few features, such as bulk upload and deletion are not supported for `Archive` containers. `Archive` containers are ideal for storing data such as email archives, data backups, and digital video masters. For information about the pricing and other terms for the `Archive` storage class in Oracle Cloud Infrastructure Object Storage Classic, go to https://cloud.oracle.com/storage-classic?tabID=1406491833493.

CORS support

Cross-Origin Resource Sharing (CORS) allows browser-based programs (like JavaScript) to access resources in another domain. This enables web applications to access Oracle Cloud Infrastructure Object Storage Classic, overcoming the **Same-Origin** policy that's used by browsers to prevent access to resources in other domains.

Setting a Container-Specific Replication Policy

You can specify a different replication policy for a container other than the policy that's defined for your service instance. The container-specific policy overrides the policy that's set for the service instance. This enables you to control, at a more granular level, what data gets replicated to a geographical distant data center.

See Setting a Container-Specific Policy Using the REST API.

Making Objects Immutable

You can make an object immutable by setting it's Write-Once-Read-Many (WORM) policy when uploading it to the container to prevent the users from modifying or deleting it for a specified duration.

See Making an Object Immutable.

You can make the objects in your container immutable by setting the WORM policy for your container to prevent the users from modifying and deleting the objects in the container for a specified duration. The container-level WORM policy applies to all the objects that are uploaded to the container, unless an object has it's own object-level WORM policy set during upload.

See Making Objects in a Container Immutable.

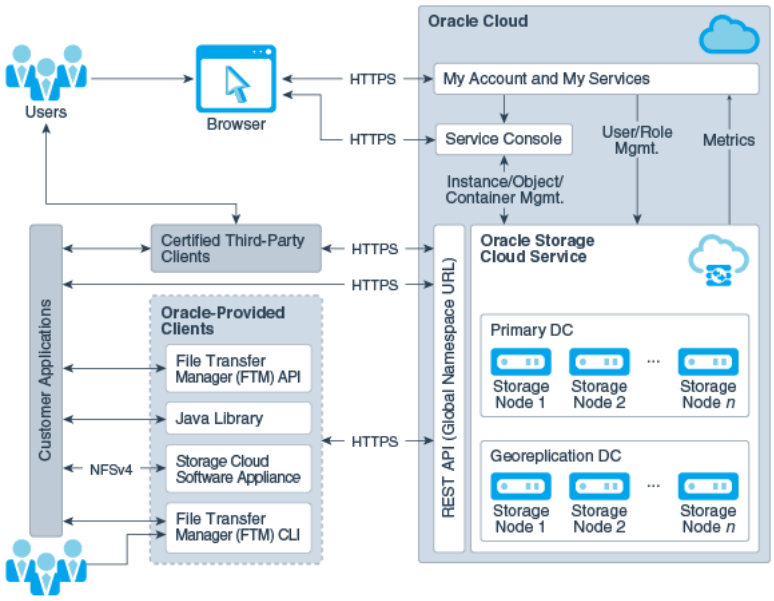
Architectural Overview

This topic does not apply to Oracle Cloud Machine.

The Oracle Cloud Infrastructure Object Storage Classic architecture is highly available and redundant. It provides support for external access methods, including customer applications, Java SDK, and REST clients.

When objects are stored in Oracle Cloud Infrastructure Object Storage Classic, the data is replicated across multiple storage nodes in the data center. This replication strategy ensures that stored object data can survive hardware failure. There can only be one Oracle Cloud Infrastructure Object Storage Classic instance per identity domain.

The following diagram presents an architectural overview of Oracle Cloud Infrastructure Object Storage Classic:



Before You Begin with Object Storage Classic


This topic does not apply to Oracle Cloud at Customer.

Before you begin using Oracle Cloud Infrastructure Object Storage Classic:

- Create and configure your account on Oracle Cloud. For more information, see Sign Up for the Free Oracle Cloud Promotion in *Getting Started with Oracle Cloud*.
- Ensure you are familiar with the following:
  - How HTTP requests and responses work
  - The Java programming language, if you intend to use the Java library to access Oracle Cloud Infrastructure Object Storage Classic. Specifically, you must be familiar with reading and writing local files using the `InputStream` interface

How to Begin with Object Storage Classic Subscriptions


This topic does not apply to Oracle Cloud at Customer.

Here's a summary of the key steps to get started with Oracle Cloud Infrastructure Object Storage Classic trials and paid subscriptions:

1. Use any of the following subscription methods to start with Oracle Cloud Infrastructure Object Storage Classic:
   - Request a trial subscription. See Sign Up for the Free Oracle Cloud Promotion in *Getting Started with Oracle Cloud*.
   - Purchase a subscription to an Oracle Cloud Service. See Buying an Oracle Cloud Subscription in *Getting Started with Oracle Cloud.*
2. Activate and verify the service. See Activating Your Order in *Getting Started with Oracle Cloud*.
3. Verify activation. See Managing Your Oracle Cloud Service in *Getting Started with Oracle Cloud*.
4. Learn about the users and roles. See About Oracle Cloud Infrastructure Object Storage Classic Roles and Users.
5. Create accounts for your users and assign them appropriate privileges and roles. See Managing Users, User Accounts, and Roles and Managing User Roles in *Managing and Monitoring Oracle Cloud*.

Accessing Object Storage Classic

Topics:

- Accessing Oracle Cloud Infrastructure Object Storage Classic Using the Web Console

- Accessing Oracle Cloud Infrastructure Object Storage Classic Using the REST API
- About REST URLs for Oracle Cloud Infrastructure Object Storage Classic Resources
- Accessing Oracle Cloud Infrastructure Object Storage Classic Using the Java Library
- Accessing Oracle Cloud Infrastructure Object Storage Classic Using Oracle Cloud Infrastructure Storage Software Appliance
- Accessing Oracle Cloud Infrastructure Object Storage Classic Using File Transfer Manager API
- Accessing Oracle Cloud Infrastructure Object Storage Classic Using the FTM CLI
- Accessing Objects in Oracle Cloud Infrastructure Object Storage Classic Using Temporary URLs

Accessing Oracle Cloud Infrastructure Object Storage Classic Using the Web Console

 This topic does not apply to Oracle Cloud at Customer.

1. Sign in to your Oracle Cloud account.
   If you see Infrastructure Classic at the top of the page when you sign in to Oracle Cloud, then you are using the Infrastructure Classic Console to access your services and your subscription does not support access to the Infrastructure Console. See Signing In to Your Cloud Account in *Getting Started with Oracle Cloud*.

   If you can access the service from the Infrastructure Console, see Signing In to the Console in Oracle Cloud Infrastructure documentation.
2. You can access the service in one of the following ways depending on whether you are using the Infrastructure Classic Console or Infrastructure Console.
   - If you are using Infrastructure Classic Console, on the navigation menu, click **Storage Classic**.
   - If you are using Infrastructure Console, on the navigation menu, under **More Oracle Cloud Services**, point to **Classic Infrastructure Services**, and then click **Storage Classic**.

   The **Oracle Cloud Infrastructure Object Storage Classic** console is displayed.

Accessing Oracle Cloud Infrastructure Object Storage Classic Using the REST API

The REST API can be accessed from any application or programming platform that correctly and completely understands the Hypertext Transfer Protocol (HTTP) and has Internet connectivity. The REST API uses advanced facets of HTTP such as secure communication over HTTPS, HTTP headers, and specialized HTTP verbs (PUT, DELETE).

Some applications that meet these requirements are:

- cURL — cURL is a command-line tool that you can use to invoke REST API calls by sending HTTP requests.
  To use cURL, see http://curl.haxx.se.
- Web browsers — Support varies across vendors. Some browser plugins may be needed for full support.

Many programming platforms (Java, Ruby, Perl, PHP, .NET, and so on) also meet these requirements, although some may require the use of third party libraries for full support. See your programming platform's documentation for guidance.

About REST URLs for Oracle Cloud Infrastructure Object Storage Classic Resources

Accounts, containers, and objects in an Oracle Cloud Infrastructure Object Storage Classic instance are represented as REST resources and are accessible through HTTP uniform resource locators (URLs).

The REST API endpoint in the Oracle Cloud Infrastructure Classic Console is the global namespace URL.

> **Note:**
>
> Depending on when your account was created, you may find multiple REST API Endpoint URLs for your account.

Topics

- REST Endpoint URL Formats in Oracle Cloud Accounts
- Finding the REST Endpoint URL for Your Cloud Account
- Using Your Account Details

REST Endpoint URL Formats in Oracle Cloud Accounts

When you sign in to your Oracle Cloud account, you'll have access to the following types of Cloud accounts:

- Cloud accounts with Identity Cloud Service
- Traditional Cloud accounts (also known as Cloud Service accounts)

To know more about the types of Oracle Cloud accounts, see Types of Oracle Cloud Accounts.

The following table describes the REST Endpoint URL formats in Oracle Cloud accounts.

Account Type	When Created	REST Endpoint URL Formats
Cloud Accounts with Identity Cloud Service	After November 2017	See REST Endpoint URL Formats in Cloud Accounts with Identity Cloud Service (Created after November 2017).

Account Type	When Created	REST Endpoint URL Formats
	Before November 2017	See REST Endpoint URL Formats in Cloud Accounts with Identity Cloud Service (Created before November 2017).
Traditional Account	After November 2017	See REST Endpoint URL Formats in Traditional Accounts (Created after November 2017).
	Before November 2017	See REST Endpoint URL Formats in Traditional Accounts (Created before November 2017).

See Using Your Account Details to access your account using your sign-in credentials.

Finding the REST Endpoint URL for Your Cloud Account

The REST Endpoint URL is also the URL for the account.

To find the REST Endpoint URL of your Oracle Cloud Infrastructure Object Storage Classic account:

1. Sign in to your Oracle Cloud account.
   If you see Infrastructure Classic at the top of the page when you sign in to Oracle Cloud, then you are using the Infrastructure Classic Console to access your services and your subscription does not support access to the Infrastructure Console. See Signing In to Your Cloud Account in *Getting Started with Oracle Cloud*.

   If you can access the service from the Infrastructure Console, see Signing In to the Console in Oracle Cloud Infrastructure documentation.
2. If you can access the service from the Infrastructure Console, perform the following steps to obtain the REST Endpoint URL:
   a. On the navigation menu, under **More Oracle Cloud Services**, point to **Classic Infrastructure Services**, and then click **Storage Classic**.
      The **Oracle Cloud Infrastructure Object Storage Classic** console is displayed.
   b. Click the **Account** tab.
   c. Note the Rest Endpoint URL which is displayed in the **Rest Endpoint** field.
3. If you can access the service from the Infrastructure Classic Console, perform the following steps to obtain the REST Endpoint URL and authentication URL:
   a. On the dashboard, look for **Storage Classic**.
   b. Select **View Details** from the **Actions** menu. Alternatively, click the **Storage Classic** link on the **Dashboard** page.
      On the resulting page, the details of your Oracle Cloud Infrastructure Object Storage Classic account are displayed.
      - **For accounts created before November 2017**
        The URL is displayed in the **REST Endpoint** field under the **Additional Information** section.

- **For accounts created after November 2017**
  The following REST Endpoint URLs are displayed under the **Additional Information** section.

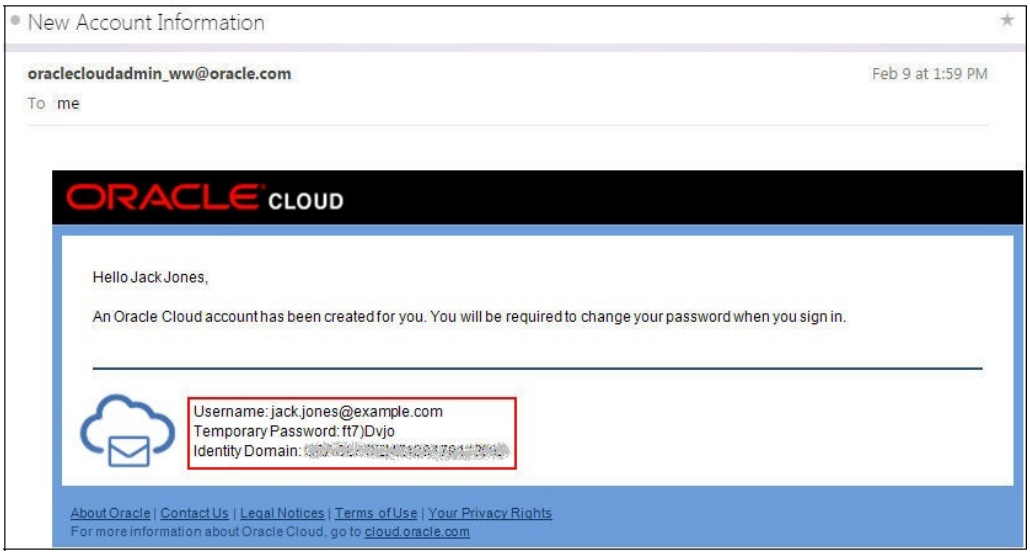
REST Endpoint URL Type	Description	When to Use?	Example
Service Permanent REST Endpoint URL	This URL is displayed in the **REST Endpoint (Permanent)** field. This URL contains the auto-generated GUID for the account and remains *constant* for your account.	Use this REST Endpoint URL: – To set a replication policy for a container in your account. See Setting a Container-Specific Policy Using the REST API. – To ensure that the URL remains *constant* under all circumstances. For example, you can hard-code the permanent URL in your code scripts to perform any operation on the Storage Classic resources. .	`https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/Storage-7b16fede61e1417ab83eb52e06f0e365`
Service Friendly REST Endpoint URL	This URL is displayed in the **REST Endpoint** field. **Note**: If you change the Storage Classic account name, then the Service Friendly REST Endpoint URL and the authentication URL will also change. Ensure that you are using the appropriate and latest URL.	Use this URL to perform *any* operation on the Storage Classic resources through the desired interface.	`https://acme.storage.oraclecloud.com/v1/Storage-acme`

Using Your Account Details

You'll need your sign-in credentials to access your account and perform any operation using the desired interface.

When you sign up for the account, you'll receive a Welcome mail with your sign-in credentials.

Make a note of your sign-in credentials, as shown in the following example:



If you don't have your Welcome mail, ask your account administrator for your sign-in credentials.

> **Note:**
>
> For traditional accounts, you can request Oracle Cloud to send the email to the administrator again. See Resending Welcome Email with Administrator Sign-in Credentials.

REST Endpoint URL Formats in Cloud Accounts with Identity Cloud Service (Created after November 2017)

Make a note of the following details of your account from the Welcome email:

- IDCS GUID
- Account name
- User name

For example - Sample Cloud account with the following details:

- IDCS GUID: `idcs-b75f75ed2528447fb59a798c1f08a38d`
- Account name: `acme`
- User name: `john.doe@example.com`

URL	Format

REST Endpoint URLs:
• Service Permanent REST Endpoint
• Service Friendly REST Endpoint
	Check in the following fields under **Additional Information** in Oracle Cloud Infrastructure Classic Console.
	• **REST Endpoint URL (Permanent)**
	Format:
	`https://storage-GUID.storage.oraclecloud.com/v1/Storage-GUID`
	REST Endpoint URL (Permanent) URL for the sample account:
	`https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/Storage-7b16fede61e1417ab83eb52e06f0e365`
	• **REST Endpoint**
	Format:
	`https://account_name.storage.oraclecloud.com/v1/Storage-account_name`
	Service Friendly REST Endpoint URL for the sample account:
	`https://acme.storage.oraclecloud.com/v1/Storage-acme`

Authentication URL	Check in **Auth V1 Endpoint** field under **Additional Information** in Oracle Cloud Infrastructure Classic Console.
	Format:
	`https://account_name.storage.oraclecloud.com/auth/v1.0`
	Note:
	If the authentication URL is not available in the Oracle Cloud Infrastructure Classic Console, then you must construct the authentication URL. See Authenticating Access to Object Storage Classic.
	Auth URL for the sample account:
	`https://acme.storage.oraclecloud.com/auth/v1.0`

Example: cURL command and output - Authentication token request for the above sample Cloud account

cURL command:

- Using the account name in the header `X-Storage-User`:

```
curl -v -X GET -H "X-Storage-User: Storage-acme:john.doe@example.com"
-H "X-Storage-Pass: Welcome1" http://acme.storage.oraclecloud.com/
auth/v1.0
```

Output:

- Using the GUID from the **Service Permanent REST Endpoint** in the header `X-Storage-User`:

  ```
  curl -v -X GET -H "X-Storage-User:
  Storage-7b16fede61e1417ab83eb52e06f0e365:john.doe@example.com" -H "X-
  Storage-Pass: Welcome1" http://acme.storage.oraclecloud.com/auth/v1.0
  ```

Output:

REST Endpoint URL Formats in Cloud Accounts with Identity Cloud Service (Created before November 2017)

Make a note of the following details of your account from the Welcome email:

- IDCS GUID
- Account name
- User name

For example - Sample Cloud account with the following details:

- IDCS GUID: `idcs-b75f75ed2528447fb59a798c1f08a38d`
- Account name: `acme`
- User name: `john.doe@example.com`

URL	Format
REST Endpoint URL	Check in **REST Endpoint** field under **Additional Information** in Oracle Cloud Infrastructure Classic Console. **Format**: `https://`*`account_name`*`.storage.oraclecloud.com/v1/Storage-`*`account_name`* REST Endpoint URL for the sample account: `https://acme.storage.oraclecloud.com/v1/Storage-acme`

URL	Format

Authentication URL	Check in **Auth V1 Endpoint** field under **Additional Information** in Oracle Cloud Infrastructure Classic Console.
	Format:
	`https://account_name.storage.oraclecloud.com/auth/v1.0`
	Note:
	If the authentication URL is not available in the Infrastructure Classic Console, then you must construct the authentication URL. See Authenticating Access to Object Storage Classic.
	Auth URL for the sample account:
	`https://acme.storage.oraclecloud.com/auth/v1.0`

Example: cURL command and output - Authentication token request for the above sample Cloud account

cURL command:

```
curl -v -X GET -H "X-Storage-User: Storage-acme:john.doe@example.com" -H
"X-Storage-Pass: Welcome1" http://acme.storage.oraclecloud.com/auth/v1.0
```

Output:

REST Endpoint URL Formats in Traditional Accounts (Created after November 2017)

Make a note of the following details of your account from the Welcome email:

- Identity domain
- Account name
- User name

For example - Sample traditional account with the following details:

- Identity domain: `acme`
- Account name: `acme`
- User name: `john.doe@example.com`

> **Note:**
>
> In traditional accounts, the identity domain name and account name are the same.

URL	Format

REST Endpoint URLs
• Service Permanent REST Endpoint
• Service Friendly REST Endpoint
	Check in the following fields under **Additional Information** in Infrastructure Classic Console.
	• **REST Endpoint URL (Permanent)**
	Format:
	`https://storage-GUID.storage.oraclecloud.com/v1/Storage-GUID`
	REST Endpoint URL (Permanent) URL for the sample account:
	`https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/Storage-7b16fede61e1417ab83eb52e06f0e365`
	• **REST Endpoint**
	Format:
	`https://`*`account_name`*`.storage.oraclecloud.com/v1/Storage-`*`account_name`*
	Service Friendly REST Endpoint URL for the sample account:
	`https://acme.storage.oraclecloud.com/v1/Storage-acme`

Authentication URL	Check in **Auth V1 Endpoint** field under **Additional Information** in Infrastructure Classic Console.
	Format:
	`https://`*`account_name`*`.oraclecloud.com/auth/v1.0`
	Note:
	If the authentication URL is not available in the Infrastructure Classic Console, then you must construct the authentication URL. See Authenticating Access to Object Storage Classic.
	Auth URL for the sample account:
	`https://acme.storage.oraclecloud.com/auth/v1.0`

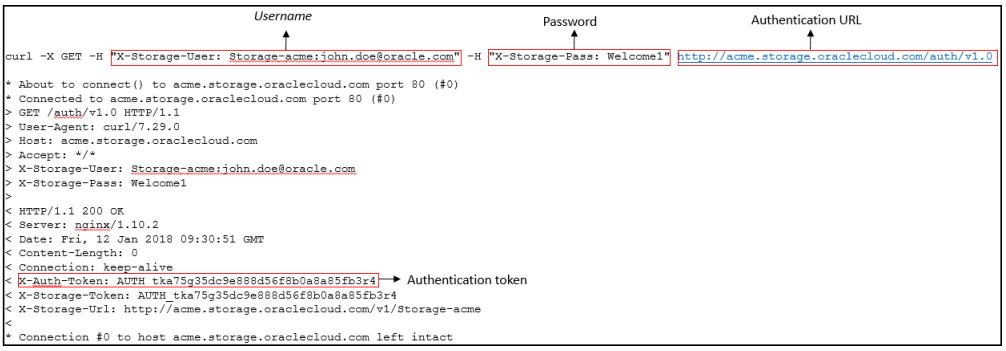
Example: cURL command and output - Authentication token request for the above sample traditional account

cURL command:

- Using the identity domain name in the header `X-Storage-User`:

  ```
  curl -v -X GET -H "X-Storage-User: Storage-acme:john.doe@example.com"
  -H "X-Storage-Pass: Welcome1" http://acme.storage.oraclecloud.com/
  auth/v1.0
  ```

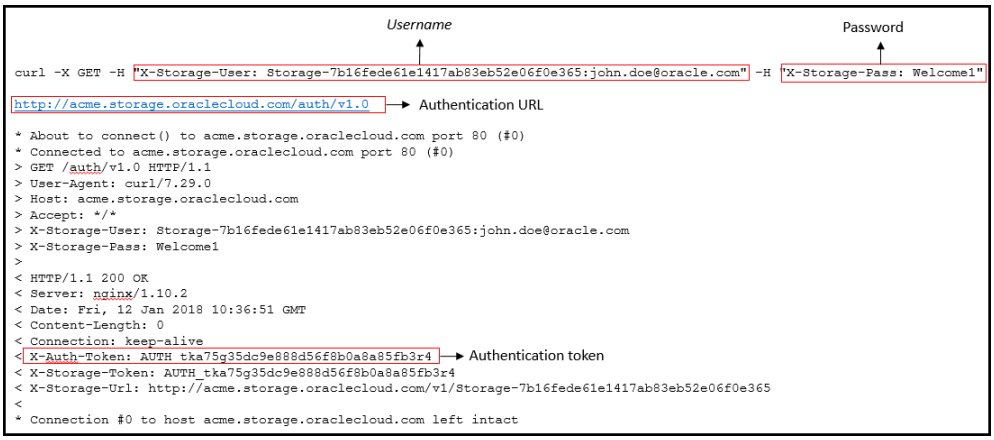
  Output:



- Using the GUID from the **Service Permanent REST Endpoint** in the header `X-Storage-User`:

  ```
  curl -v -X GET -H "X-Storage-User:
  Storage-7b16fede61e1417ab83eb52e06f0e365:john.doe@example.com" -H "X-
  Storage-Pass: Welcome1" http://acme.storage.oraclecloud.com/auth/v1.0
  ```

Output:



REST Endpoint URL Formats in Traditional Accounts (Created before November 2017)

Make a note of the following details of your account from the Welcome email:

- Identity domain
- Account name
- User name

For example - Sample traditional account with the following details:

- Identity domain: `acme`
- Account name: `acme`
- User name: `john.doe@example.com`

> **Note:**
>
> In traditional accounts, the identity domain name and account name are the same.

URL	Format

REST Endpoint URL	Check in **REST Endpoint** field under **Additional Information** in Infrastructure Classic Console.
	Format:
	`https://account_name.storage.oraclecloud.com/v1/Storage-account_name`
	REST Endpoint URL for the sample account:
	`https://acme.storage.oraclecloud.com/v1/Storage-acme`

Authentication URL	Check in **Auth V1 Endpoint** field under **Additional Information** in Infrastructure Classic Console.
	Format:
	`https://account_name.oraclecloud.com/auth/v1.0`
	Note:
	If the authentication URL is not available under **Additional Information**, then you must construct the authentication URL. See Authenticating Access to Object Storage Classic.
	Auth URL for the sample account:
	`https://acme.storage.oraclecloud.com/auth/v1.0`

Example: cURL command and output - Authentication token request for the above sample traditional account

cURL command:

```
curl -v -X GET -H "X-Storage-User: Storage-acme:john.doe@example.com" -H
"X-Storage-Pass: Welcome1" http://acme.storage.oraclecloud.com/auth/v1.0
```

Output:

Accessing Oracle Cloud Infrastructure Object Storage Classic Using the Java Library

The Java library uses the REST API. So the Java library, too, requires Internet connectivity.

The Java library requires Java Runtime Environment (JRE) version 1.6 or later. The Java library has several runtime-dependent Java libraries, all of which are included in the downloadable Java SDK.

> **Note:**
>
> You cannot create `archive` containers by using the Java API.

To use the Java library in your own Java applications:

1. Download the Oracle Cloud Infrastructure Object Storage Classic Java SDK from: http://www.oracle.com/technetwork/topics/cloud/downloads/cloud-service-java-sdk-2121032.html
2. Extract the Java library's classes and runtime dependencies somewhere onto your Java application's class path.
3. Import the Java library's classes and interfaces into your Java application.

For information about using the Java library to perform specific operations for containers and objects, see Managing Containers in Oracle Cloud Infrastructure Object Storage Classic and Managing Objects in Oracle Cloud Infrastructure Object Storage Classic.

Accessing Oracle Cloud Infrastructure Object Storage Classic Using Oracle Cloud Infrastructure Storage Software Appliance

Oracle Cloud Infrastructure Storage Software Appliance is a tool that you can install on-premises and then use to easily connect your on-premises applications and workflows to Oracle Cloud Infrastructure Object Storage Classic.

Using Oracle Cloud Infrastructure Storage Software Appliance, your applications can interact with Oracle Cloud Infrastructure Object Storage Classic through standard file-based network protocols, without invoking direct REST API calls to the service.

For information on the list of tasks that you can perform in your Oracle Cloud Infrastructure Object Storage Classic account using Oracle Cloud Infrastructure Storage Software Appliance, see Tasks Supported by the Interfaces of Object Storage Classic.

Accessing Oracle Cloud Infrastructure Object Storage Classic Using File Transfer Manager API

File Transfer Manager API is a Java library that provides a simple interface to upload or download individual and multiple objects of various sizes to `standard` and `archive` containers in Oracle Cloud Infrastructure Object Storage Classic.

To use the FTM API:

1. Download the SDK from http://www.oracle.com/technetwork/topics/cloud/downloads/file-transfer-manager-2956858.html.
2. Extract the SDK and include the Java libraries from `libs` folder into the class path of your Java application.
3. Import the Java classes and interfaces into your Java application. Before you import, see the sample code from the `samples` folder in the SDK.

Accessing Oracle Cloud Infrastructure Object Storage Classic Using the FTM CLI

The Oracle Cloud Infrastructure Object Storage Classic File Transfer Manager CLI (FTM CLI) is a cross-platform Java-based command line tool that you can use to upload and download objects to `standard` and `archive` containers in Oracle Cloud Infrastructure Object Storage Classic.

See *Command-Line Reference for Oracle Cloud Infrastructure Object Storage Classic*.

Accessing Objects in Oracle Cloud Infrastructure Object Storage Classic Using Temporary URLs

Temporary URLs are time-limited URLs that expire after a configured time period. You can create temporary URLs to provide secure, temporary access to protected resources like objects in your Oracle Cloud Infrastructure Object Storage Classic account. Users who do not have access to Oracle Cloud Infrastructure Object Storage Classic can download an object from the service using a temporary URL that you provide.

See Downloading an Object Using a Temporary URL.

About Object Storage Classic Roles and Users

The following table summarizes the Oracle Cloud Infrastructure Object Storage Classic roles used for accessing, administering, and using Oracle Cloud Infrastructure Object Storage Classic instances.

Role Name	Description	More Information
`TenantAdminGroup` (Identity Domain Administrator)	Users who are assigned this role can perform all tasks in the Infrastructure Classic Console, including user and role management tasks. Note that Oracle Cloud assigns this role to all trial users.	Oracle Cloud User Roles and Privileges in *Getting Started with Oracle Cloud*
`Storage``.Storage_Administrator` (Service Administrator) For nonmetered subscriptions, the role name would be *`service-instance-name`*`.Storage_Administrator`.	Users who are assigned this role can perform the following tasks: • Perform all tasks for an Oracle Cloud Infrastructure Object Storage Classic instance, including user management • Monitor and manage service usage in Oracle Cloud • Grant roles to users • Create and delete containers • Modify container ACLs The account administrator can create more storage administrators, as required, by assigning this role.	• Managing Containers in Oracle Cloud Infrastructure Object Storage Classic • Managing Objects in Oracle Cloud Infrastructure Object Storage Classic
`Storage``.Storage_ReadWriteGroup` For nonmetered subscriptions, the role name would be *`service-instance-name`*`.Storage_ReadWriteGroup`.	Users who are assigned this role can perform the following tasks: • Create, read, modify, and delete objects within containers • List containers (note that they *cannot* create, modify, or delete containers) • List objects within containers unless the roles has been removed from the containers's read ACL	• Managing Containers in Oracle Cloud Infrastructure Object Storage Classic • Managing Objects in Oracle Cloud Infrastructure Object Storage Classic
`Storage``.Storage_ReadOnlyGroup` For nonmetered subscriptions, the role name would be *`service-instance-name`*`.Storage_ReadOnlyGroup`.	Can perform the following tasks: • Read objects • List containers • List objects within containers unless the role has been removed from the container's read ACL Given the default ACLs added to containers, users who are assigned this role can read the contents of all containers.	• Managing Containers in Oracle Cloud Infrastructure Object Storage Classic • Managing Objects in Oracle Cloud Infrastructure Object Storage Classic

Note that the containers ACLs can be rewritten. So while the predefined roles have semantics based on the default behavior, access to a container is governed entirely by the values set for the container's `X-Container-Read` and `X-Container-Write` metadata fields, and not by the role. For more information, see Setting Container ACLs.

About Access Control Lists

The ability to read and write objects in a container is governed by the Access Control Lists (ACLs) assigned to the container. These ACLs are written to two metadata fields: `X-Container-Read` and `X-Container-Write`.

Users with roles assigned to these metadata fields can perform the following actions:

- `X-Container-Read`: Users can read objects and associated metadata in the given container.
- `X-Container-Write`: Users can create and delete objects and associated metadata in the given container.

The metadata field values are a comma-separated list of identity domain ID and role pairs. This allows service administrators to grant read or write access to users in other identity domains. Users with the `Storage_Administrator` role may define their own roles in the Users page in Infrastructure Classic Console and assign them to the `X-Container-Read` and `X-Container-Write` headers on containers, as required.

For creating custom roles for a traditional Cloud account. See Adding a Custom Role in *Managing and Monitoring Oracle Cloud*.

For creating custom roles for accounts with Identity Cloud Service, see Create a Custom Role for Cloud Accounts with Identity Cloud Service.

Users with the `Storage_Administrator` role will always have read and write access to all containers in their service instance.

All non-administrator users are subject to the ACLs for a given container.

The service instance root path is an exception to this, because it does not have ACLs associated with it. For this path, all users can obtain a list of containers, but only users with the `Storage_Administrator` role can create or delete containers.

By default, when a container is created in the Oracle Cloud Infrastructure Object Storage Classic, the following ACLs are assigned:

- `X-Container-Read`: *`identity_domain_ID.storage_service.`*`Storage_ReadOnlyGroup,`*`identity_domain_ID.storage_service`*`.Storage_ReadWriteGroup`
- `X-Container-Write`: *`identity_domain_ID.storage_service`*`.Storage_ReadWriteGroup`

Example:

The following are the newly created container ACL values for a service instance named `Storage` in an identity domain named `myIdentity3`.

- `X-Container-Read: myIdentityDomainID.Storage.Storage_ReadOnlyGroup, myIdentityDomainID.Storage.Storage_ReadWriteGroup`
- `X-Container-Write: myIdentityDomainID.Storage.Storage_ReadWriteGroup`

To learn how to restrict read and write access to containers by using ACLs, see Setting Container ACLs.

Create a Custom Role for Cloud Accounts with Identity Cloud Service

To create a custom role for cloud accounts with identity cloud service:

1. Find your Oracle Identity Cloud Service tenant name.
   a. From the dashboard in Infrastructure Classic Console, click **Identity Cloud**.
   b. In the **Additional Information** section of the **Overview** tab, copy the tenant name from the **Identity Service Id** field. The tenant name begins with the characters **idcs-** and then is followed by a string of numbers and letters, for example, **idcs-6572bfeb183b4becad9e649bfa14a488**.
2. Create a Confidential application in Oracle Identity Cloud Service. See Working with OAuth 2 to Access the REST API in *REST API for Oracle Identity Cloud Service*.
3. After activating the application, identify and note down the SCIM Application ID from the application link that's available in the browser. For example, if `https://idcs-6572bfeb183b4becad9e649bfa14a488.identity.oraclecloud.com/ui/v1/adminconsole?root=apps&app=e947cd3a3573975980930d52dfc111fb` is the application link in the browser, then the SCIM application ID is `e947cd3a3573975980930d52dfc111fb`.
4. Base64 encode the client ID and client secret that you had noted down while creating the application, and then obtain an access token. See Working with OAuth 2 to Access the REST API in *REST API for Oracle Identity Cloud Service*.
5. In the response, you can see `Status: 200`. Copy the value of `access_token` from the response as you will have to provide this value while sending a REST request to the Oracle Identity Cloud Service REST API.
6. Run the following command to create a custom role for the application. Provide the tenant base URL, access token value, and SCIM application ID based on your environment.

```
curl --request POST \
  --url https://idcs-<tenant-base-url>.identity.oraclecloud.com/
admin/v1/AppRoles \
  --header 'authorization: Bearer <access token that you have obtained
in the previous step>' \
  --header 'content-type: application/json' \
  --data ' {
  "displayName": "My_Custom_Application_Role",
  "adminRole": false,
  "description": "My custom application role",
  "availableToUsers": true,
  "availableToGroups": true,
  "availableToClients": true,
  "app": {
    "value": "<SCIM Application ID that you have identified>"
  },
  "schemas": [
    "urn:ietf:params:scim:schemas:oracle:idcs:AppRole"
  ]
}'
```

2

Managing Containers in Object Storage Classic

This section provides documentation about managing containers in Oracle Cloud Infrastructure Object Storage Classic.

Topics:

- Typical Workflow for Managing Containers
- Creating Containers
- Creating Archive Containers
- Listing Containers
- Setting a Container-Specific Policy Using the REST API
- Deleting Containers
- Enabling Server-Side Encryption
- Setting Container Metadata
- Getting Container Metadata
- Deleting Container Metadata

Typical Workflow for Managing Containers

Use the following table as a guide to the workflow for managing containers. If you have not yet subscribed to or set up your service, see Workflow for Getting Started with Object Storage Classic.

Task	Description	More Information
Access the service	Access the service through any of the available interfaces.	Accessing Oracle Cloud Infrastructure Object Storage Classic
Generate an authentication token	Generate an authentication token if you are accessing the service through the REST API interface. If you are accessing the service through Java library, the authentication token would be automatically generated and applied.	Authenticating Access When Using REST API
Create containers	Create containers to organize your data.	Creating Containers
Delete containers	Delete empty containers when no longer needed.	Deleting Containers
Update container metadata	Set custom metadata or special metadata on a container.	Updating Container Metadata

Creating Containers

A container is a storage compartment that provides a way to organize the data stored in Oracle Cloud Infrastructure Object Storage Classic.

Any user with the Service Administrator role can create containers. You should create at least one container for your account. Containers are similar to a directory structure but with a key distinction: unlike directories, containers cannot be nested. By default, all containers are of the `standard` storage class (as opposed to the `archive` storage class).

> **Note:**
>
> Before you create your first container, check the replication policy set for your account. See About Replication Policy for Your Object Storage Classic Instance.

You can create containers by using the following interfaces:

Interface	Resources
Web Console (Not available on Oracle Cloud at Customer)	See Creating a Container Using the Web Console.
RESTful API	See: • Creating a Container Using the REST API • Create Container in *REST API for Standard Storage in Oracle Cloud Infrastructure Object Storage Classic*
Java Library	See `createContainer` in *Java API Reference for Oracle Cloud Infrastructure Object Storage Classic*.
File Transfer Manager API	See `createContainer` in *Java API Reference for Oracle Cloud Infrastructure Object Storage Classic File Transfer Manager*.
File Transfer Manager CLI	See Creating a Container in *Command-Line Reference for Oracle Cloud Infrastructure Object Storage Classic*.

To create an `archive` container, you must set the `X-Storage-Class` header to `Archive`. For more information, see Creating Archive Containers. (Not available on Oracle Cloud at Customer)

Creating a Container Using the Web Console

(Not available on Oracle Cloud at Customer)

1. Sign in to the Oracle Cloud Infrastructure Object Storage Classic console.
2. Click **Create Container**.
   The **Create Container** dialog box is displayed.
3. Enter a name for the container.

> **Note:**
>
> Ensure that the container name complies with the input restrictions mentioned in Character Restrictions.

4. Select `Standard` in the **Storage Class** field.
5. To set a replication policy for the container, see Setting a Container-Specific Policy Using the Web Console.

> **Note:**
>
> This option is available only for accounts created **after** March 2018.

6. Click **Create**.
   The container is created and displayed in the console.

Creating a Container Using the REST API

cURL Command Syntax

```
curl -v -X PUT \
     -H "X-Auth-Token: token" \
     accountURL/containerName
```

- `token` is the authentication token obtained earlier from Oracle Cloud Infrastructure Object Storage Classic. See Authenticating Access When Using the REST API.
- For the syntax of `accountURL`, see About REST URLs for Oracle Cloud Infrastructure Object Storage Classic Resources.
- `containerName` is the name of the container to be created.

> **Note:**
>
> Ensure that the container name complies with the input restrictions mentioned in Character Restrictions.

> **Note:**
>
> When you send a REST API request to Oracle Cloud Infrastructure Object Storage Classic, all non-ASCII characters in container names, object names and metadata values must be URL-encoded. For example, `my container` should be encoded as `my%20container`, where `%20` is the HTML encoding for the space character. Similarly, `my Über Container` should be encoded as `my %20%C3%9Cber%20Container`, where `%20` represents the space character and `%C3%9C` is the Ü character.

HTTP Response Codes

- Success: `201 Created`

- Failure: See Error Code Reference for Oracle Cloud Infrastructure Object Storage Classic

cURL Command Examples

Sample Cloud account with the following details:

- Account name: `acme`
- REST Endpoint URL: `https://acme.storage.oraclecloud.com/v1/Storage-acme`
- REST Endpoint (Permanent) URL: `https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/Storage-7b16fede61e1417ab83eb52e06f0e365`

> **Note:**
>
> The REST Endpoint (Permanent) URL is displayed for the accounts created after November 2017.

- Using the REST Endpoint URL obtained from the **REST Endpoint** field:

```
curl -v -X PUT \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     https://acme.storage.oraclecloud.com/v1/Storage-acme/FirstContainer
```

The following is an example of the output of this command:

```
> PUT /v1/Storage-acme/FirstContainer HTTP/1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.14.0.0
zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: acme.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b
>
< HTTP/1.1 201 Created
< Date: Fri, 06 Mar 2015 10:34:20 GMT
< Content-Length: 0
< Content-Type: text/html; charset=UTF-8
< X-Trans-Id: tx23a1084b8c674fdeae8d4-0054f982ac
< Cache-Control: no-cache
< Pragma: no-cache
< Content-Language: en
```

- Using the Service Permanent REST Endpoint URL obtained from the **REST Endpoint (Permanent)** field:

> **Note:**
>
> This cURL command example applies to the accounts created after November 2017.

```
curl -v -X PUT \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/
Storage-7b16fede61e1417ab83eb52e06f0e365/FirstContainer
```

The following is an example of the output of this command:

```
> PUT /v1/Storage-7b16fede61e1417ab83eb52e06f0e365/FirstContainer HTTP/1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.14.0.0
zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b
>
< HTTP/1.1 201 Created
< Date: Fri, 06 Mar 2015 10:34:20 GMT
< Content-Length: 0
< Content-Type: text/html; charset=UTF-8
< X-Trans-Id: tx23a1084b8c674fdeae8d4-0054f982ac
< Cache-Control: no-cache
< Pragma: no-cache
< Content-Language: en
```

For setting the replication policy in the container, see Setting a Container-Specific Policy Using the REST API.

For information about getting details of a container, see Getting Container Metadata.

Creating Archive Containers


This topic does not apply to Oracle Cloud at Customer.

A container for which the `X-Storage-Class` metadata field is set to `Archive` is called an `Archive` container. You can use `Archive` containers to store data that won't be accessed for a while or that will be accessed infrequently. Any user with the Service Administrator role can create `Archive` containers.

You can create `Archive` containers by using the following interfaces:

Interface	Resources
Web Console	See Creating an Archive Container Using the Web Console.
RESTful API	See: • Creating an Archive Container Using the REST API • Create Container in *REST API for Archive Storage in Oracle Cloud Infrastructure Object Storage Classic*
File Transfer Manager API	See `createContainer` in *Java API Reference for Oracle Cloud Infrastructure Object Storage Classic File Transfer Manager*.

Note:

- You can set `X-Storage-Class` only when creating a new container. After you set `X-Storage-Class`, you cannot change it.
- You cannot create an `Archive` container by using the Java API.

Features Not Supported for Archive Containers

The following features are *not* supported for an `Archive` container:

- Bulk-creating objects
- Bulk-deleting objects
- Scheduling automatic deletion of objects by using the `X-Delete-At` and `X-Delete-After` headers
- Georeplication
- Setting a container-specific replication policy

Creating an Archive Container Using the Web Console

1. Sign in to the Oracle Cloud Infrastructure Object Storage Classic console.
2. Click **Create Container**.
   The **Create Container** dialog is displayed.
3. Enter a name for the container.

> **Note:**
>
> Ensure that the container name complies with the input restrictions mentioned in Character Restrictions.

4. Select `Archive` in the **Storage Class** field.
5. Click **Create**.
   The container is created and displayed in the console.

Creating an Archive Container Using the REST API

cURL Command Syntax

```
curl -v -X PUT \
     -H "X-Auth-Token: token" \
     -H "X-Storage-Class: Archive" \
     accountURL/containerName
```

- `token` is the authentication token obtained earlier from Oracle Cloud Infrastructure Object Storage Classic. See Authenticating Access When Using the REST API.
- For the syntax of `accountURL`, see About REST URLs for Oracle Cloud Infrastructure Object Storage Classic Resources.
- `containerName` is the name of the container to be created.

> **Note:**
>
> Ensure that the container name complies with the input restrictions mentioned in Character Restrictions.

> **Note:**
>
> When you send a REST API request to Oracle Cloud Infrastructure Object Storage Classic, all non-ASCII characters in container names, object names and metadata values must be URL-encoded. For example, `my container` should be encoded as `my%20container`, where `%20` is the HTML encoding for the space character. Similarly, `my Über Container` should be encoded as `my %20%C3%9Cber%20Container`, where `%20` represents the space character and `%C3%9C` is the Ü character.

HTTP Response Codes

- Success: `201 Created`
- Failure: See Error Code Reference for Oracle Cloud Infrastructure Object Storage Classic

cURL Command Examples

Sample Cloud account with the following details:

- Account name: `acme`
- REST Endpoint URL: `https://acme.storage.oraclecloud.com/v1/Storage-acme`
- REST Endpoint (Permanent) URL: `https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/Storage-7b16fede61e1417ab83eb52e06f0e365`

> **Note:**
>
> The REST Endpoint (Permanent) URL is displayed for the accounts created after November 2017.

- Using the REST Endpoint URL obtained from the **REST Endpoint** field:

```
curl -v -X PUT \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     -H "X-Storage-Class: Archive"
     https://acme.storage.oraclecloud.com/v1/Storage-acme/firstArchiveContainer
```

The following is an example of the output of this command:

```
> PUT /v1/Storage-acme/firstArchiveContainer HTTP/1.1
> User-Agent: curl/7.29.0
> Host: acme.storage.oraclecloud.com:443
> Accept: */*
> X-Auth-Token: AUTH_tk1ff0554c1fefff9209696d63553722fd
> X-Storage-Class: Archive
>
< HTTP/1.1 201 Created
< X-Trans-Id: tx1d8e9739df4a47bb847e0-005582875bga
< Date: Thu, 18 Jun 2015 08:54:51 GMT
< Content-Type: text/html;charset=UTF-8
< Content-Length: 0
<
```

- Using the Service Permanent REST Endpoint URL obtained from the **REST Endpoint (Permanent)** field:

  > **Note:**
  >
  > This cURL command example applies to the accounts created after November 2017.

  ```
  curl -v -X PUT \
       -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
       -H "X-Storage-Class: Archive"
       https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/
  Storage-7b16fede61e1417ab83eb52e06f0e365/firstArchiveContainer
  ```

  The following is an example of the output of this command:

  ```
  > PUT /v1/Storage-7b16fede61e1417ab83eb52e06f0e365/firstArchiveContainer HTTP/1.1
  > User-Agent: curl/7.29.0
  > Host: storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com:443
  > Accept: */*
  > X-Auth-Token: AUTH_tk1ff0554c1fefff9209696d63553722fd
  > X-Storage-Class: Archive
  >
  < HTTP/1.1 201 Created
  < X-Trans-Id: tx1d8e9739df4a47bb847e0-005582875bga
  < Date: Thu, 18 Jun 2015 08:54:51 GMT
  < Content-Type: text/html;charset=UTF-8
  < Content-Length: 0
  <
  ```

For information about getting details of an `Archive` container, see Getting Container Metadata.

For creating objects in an `Archive` container, see Creating Objects.

Listing Containers

All containers within an account can be listed.

Any user within the identity domain can perform this task.

You can list containers by using the following interfaces:

Interface	Resources
Web Console (Not available on Oracle Cloud at Customer)	See Listing Containers Using the Web Console.
RESTful API	See: • Listing a Container Using the REST API • List Endpoints in *REST API for Standard Storage in Oracle Cloud Infrastructure Object Storage Classic*.
Java Library	See `listContainer` in *Java API Reference for Oracle Cloud Infrastructure Object Storage Classic*

Interface	Resources
File Transfer Manager API	See `listContainers` in *Java API Reference for Oracle Cloud Infrastructure Object Storage Classic File Transfer Manager*.
File Transfer Manager CLI	See Listing Containers in *Command-Line Reference for Oracle Cloud Infrastructure Object Storage Classic*.

Listing Containers Using the Web Console

(Not available on Oracle Cloud at Customer)
Sign in to the Oracle Cloud Infrastructure Object Storage Classic console. The list of containers is displayed.

Listing a Container Using the REST API

Containers are sorted lexicographically using `memcmp()`. All containers, up to 10000 by default, will be returned in the list, unless you filter the list by using any of the following query parameters:

- `limit`: Limit the number of containers listed to the specified value. The default and maximum value is 10000.
- `marker`: Return containers with names greater than the specified string.
- `end_marker`: Return containers with names less than the specified string.
- `format`: Return extended information about each returned container in either `xml` or `json` format (REST API only).

cURL Command Syntax

```
curl -v -X GET \
     -H "X-Auth-Token: token" \
     accountURL[?query_parameter=value]
```

- `token` is the authentication token obtained earlier from Oracle Cloud Infrastructure Object Storage Classic. See Authenticating Access When Using the REST API.
- For the syntax of `accountURL`, see About REST URLs for Oracle Cloud Infrastructure Object Storage Classic Resources.
- `query_parameter=value` is the optional filtering parameter.

> **Note:**
>
> When you send a REST API request to Oracle Cloud Infrastructure Object Storage Classic, all non-ASCII characters in container names, object names and metadata values must be URL-encoded. For example, `my container` should be encoded as `my%20container`, where `%20` is the HTML encoding for the space character. Similarly, `my Über Container` should be encoded as `my %20%C3%9Cber%20Container`, where `%20` represents the space character and `%C3%9C` is the Ü character.

HTTP Response Codes

- Success: `200 OK`

> **Note:**
>
> If there are no containers, the HTTP response code would be `204 OK`.

- Failure: See Error Code Reference for Oracle Cloud Infrastructure Object Storage Classic

cURL Command Examples

Sample Cloud account with the following details:

- Account name: `acme`
- REST Endpoint URL: `https://acme.storage.oraclecloud.com/v1/Storage-acme`
- REST Endpoint (Permanent) URL:`https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/Storage-7b16fede61e1417ab83eb52e06f0e365`

> **Note:**
>
> The REST Endpoint (Permanent) is displayed for the accounts created after November 2017.

- Using the REST Endpoint URL obtained from the **REST Endpoint** field:

```
curl -v -X GET \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     https://acme.storage.oraclecloud.com/v1/Storage-acme?limit=15
```

This command lists the first 15 containers, lexicographically sorted, in the specified account.

The following is an example of the output of this command:

```
> GET /v1/Storage-acme HTTP/1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.14.0.0
zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: acme.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tk6403794c218a709d1c6c5a76444d01f6
>
< HTTP/1.1 200 OK
< Date: Fri, 06 Mar 2015 10:38:15 GMT
< Content-Length: 109
< X-Account-Container-Count: 3
< Accept-Ranges: bytes
< X-Account-Object-Count: 843
< X-Account-Bytes-Used: 10304761355
< X-Timestamp: 1412823447.62495
< X-Account-Meta-Test5: test1
< X-Account-Meta-Quota-Bytes: 107374182400
< Content-Type: text/plain; charset=utf-8
< X-Account-Meta-Test: test
< X-Account-Meta-Test1: test1
< X-Trans-Id: tx29052c64fe384fc690ccc-0054f98397
< Cache-Control: no-cache
< Pragma: no-cache
< Content-Language: en
```

```
<
FirstContainer
hello
lab
```

- Using the Service Permanent REST Endpoint URL obtained from the **REST Endpoint (Permanent)** field:

  > **Note:**
  >
  > This cURL command example applies to the accounts created after November 2017.

- ```
  curl -v -X GET \
       -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
       https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/
  Storage-7b16fede61e1417ab83eb52e06f0e365?limit=15
  ```

  This command lists the first 15 containers, lexicographically sorted, in the specified account.

  The following is an example of the output of this command:

  ```
  > GET /v1/Storage-7b16fede61e1417ab83eb52e06f0e365 HTTP/1.1
  > User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.14.0.0
  zlib/1.2.3 libidn/1.18 libssh2/1.4.2
  > Host: storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com
  > Accept: */*
  > X-Auth-Token: AUTH_tk6403794c218a709d1c6c5a76444d01f6
  >
  < HTTP/1.1 200 OK
  < Date: Fri, 06 Mar 2015 10:38:15 GMT
  < Content-Length: 109
  < X-Account-Container-Count: 3
  < Accept-Ranges: bytes
  < X-Account-Object-Count: 843
  < X-Account-Bytes-Used: 10304761355
  < X-Timestamp: 1412823447.62495
  < X-Account-Meta-Test5: test1
  < X-Account-Meta-Quota-Bytes: 107374182400
  < Content-Type: text/plain; charset=utf-8
  < X-Account-Meta-Test: test
  < X-Account-Meta-Test1: test1
  < X-Trans-Id: tx29052c64fe384fc690ccc-0054f98397
  < Cache-Control: no-cache
  < Pragma: no-cache
  < Content-Language: en
  <
  FirstContainer
  hello
  lab
  ```

Setting a Container-Specific Policy


This topic does not apply to Oracle Cloud at Customer.
  ```

You can specify a different replication policy for each container other than the account-level replication policy. The container-specific policy enables you to control, at a more granular level, what data gets replicated to a geographical distant data center (DC).

You can set a container-specific replication policy by using the following interfaces:

Interface	Resources
Web Console	See Setting a Container-Specific Policy Using the Web Console.
RESTful API	See: • Setting a Container-Specific Policy in a Data Center Within the Same Region Using the REST API • Setting a Container-Specific Policy in a Different Region Using the REST API
File Transfer Manager CLI	See Setting a Container-Specific Replication Policy in *Command-Line Reference for Oracle Cloud Infrastructure Object Storage Classic*.
File Transfer Manager API	See `ContainerReplicationPolicy` in *Java API Reference for Oracle Cloud Infrastructure Object Storage Classic File Transfer Manager*.

Topics:

- Setting a Container-Specific Policy Using the Web Console
- Setting a Container-Specific Policy Using the REST API

Setting a Container-Specific Policy Using the Web Console

> **Important:**
>
> This topic applies to the accounts created **after** March 2018, wherein, for these accounts, the account-level replication policy has been set to `any`, by default.

1. Sign in to the Oracle Cloud Infrastructure Object Storage Classic console.
2. Click **Create Container**.
   The **Create Container** dialog box is displayed.
3. Enter a name for the container.

   > **Note:**
   >
   > Ensure that the container name complies with the input restrictions mentioned in Character Restrictions.

4. Select `Standard` in the **Storage Class** field.
5. Select the data center in the **Data Center** field, in which the container must be created.
   The data center in which the container is created is the **source** data center.

6. Select **Replicate to Data Center(s)** under **Advanced** to replicate the container in the desired data center.
   The data center selected to replicate the container is the **target** data center. You can select multiple target DCs for a container.

7. Click **Create**.
   The container is created in the source data center and displayed in the web console.

Viewing the Container Replicated in the Target Data Center

1. Sign in to the Oracle Cloud Infrastructure Object Storage Classic console.
2. Select the target data center in the **Containers** pane.
   The list of containers created in your account is displayed.
3. Identify the container that you created in the source data center.
4. Select the container.
   The following message is displayed:

   ```
   You are currently viewing a READ ONLY copy of this container. Switch
   to source environment to edit this container.
   ```

Setting a Container-Specific Policy Using the REST API

 This topic does not apply to Oracle Cloud at Customer.

You can specify a different replication policy for each container other than the account-level policy. The container-level policy overrides the account-level policy. This enables you to control, at a more granular level, what data gets replicated to a geographical distant data center (DC).

Any user with the Service Administrator role can perform this task.

> **Note:**
>
> - You cannot set replication policies for `archive` containers.
> - You can set the container-level replication policy when you create a container or when the container is empty.

Setting a Container-Specific Policy in a Data Center Within the Same Region Using the REST API

You can set your container's source DC and georeplication DC to be a subset of that of the service instance. For example, if the primary DC of your service instance is `us2` and the georeplication DCs are `us6` and `uscom-central-1`, then the primary DC of your container can be `us6`, `us2`, or `uscom-central-1`. One or both of the remaining DCs can be selected as the replication DC of the container. Specify your container's primary DC URL and replication DC URLs in the `sourceRegion` and `targetRegions` parameters respectively in the request body JSON file.

Write requests to a container for which you've set a replication policy must be sent to the DC-specific REST endpoint URL, and not to the global namespace URL.

DC-specific REST endpoint URLs are in the format:

`https://dataCenterCode.storage.oraclecloud.com/v1/Storage-identityDomainID`

Example: `https://us2.storage.oraclecloud.com/Storage-myDomain`

Setting a Container-Specific Policy in a Different Region Using the REST API

You can specify a replication policy for your container by selecting a data center outside the region where the primary DC and georeplication DC of your service instance are located. For example, if the primary DC and georeplication DC of your service instance are `us2` and `us6` in the US region, then you can select a data center, say `em2` that's located in a non-US region to replicate your container. Specify the external container to which the objects from your container must be replicated by specifying the external container's URL in the `externalTargetRegions` parameter of the request body JSON file. You can specify multiple external replication DCs for your container.

When your container is the destination for the replication of objects from an external container, specify the external container's URL in the `externalSourceRegions` parameter of the request body JSON file. Your container can be the destination for replication of objects from multiple source containers.

> **Note:**
>
> While setting the container's replication policy in a different region, first set the replication policy of the target region with the `externalSourceRegions` parameter, and next set the replication of the source region with the `externalTargetRegions` parameter.

> **Important:**
>
> To set the container's replication policy in a different region, ensure that the REST Endpoint URLs specified in the request body JSON file are in the **GUID** format. For example: `https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/Storage-7b16fede61e1417ab83eb52e06f0e365/myContainer`.
> **Do not** use the friendly name format. For example: `https://acme.storage.oraclecloud.com/v1/Storage-acme/myContainer`. The friendly name REST Endpoint URLs may change and lead to invalid replication policy.

Creating the Request Body JSON File

Create a request body JSON file using the following template and store it on your host:

```
{
  "sourceRegion": {
    "name": "primary_DC_code",
```

```
    "url": "primary_DC_url"},
  "targetRegions": [{
    "name": "replication_DC_code",
    "url": "replication_DC_url"}],
  "externalSourceRegions": [{
    "name": "external",
    "url": "external_source_url"}],
  "externalTargetRegions": [{
    "name": "external",
    "url": "external_target_url"}]
}
```

- `primary_DC_code` is the data center code of the primary DC of your container.
- `primary_DC_url` is the URL of your container located in the primary DC of your container. This is an optional parameter.
- `replication_DC_code` is the data center code of the replication DC of your container.
- `replication_DC_url` is the URL of the replication of your container located in the replication DC of your container. This is an optional parameter.
- `external_source_url` is the URL of the container in an external region from which the objects would be replicated into your container.
- `external_target_url` is the URL of the container in an external region into which the objects in your container must be replicated.

> **Note:**
>
> The parameters `primary_DC_url` and `replication_DC_url` are optional. Providing the data center code is sufficient for the normal working of the command.

- Example request body JSON file for your container `myContainer` in the following scenario:
  - Your service instance has `us2` primary DC.
  - Your service instance has `us6` and `uscom-central-1` replication DCs.
  - Primary DC for your container is `us6`.
  - Replication DCs for your container are`us2` and `uscom-central-1` within the same region.

  Note that the examples demonstrate the use of GUID-based URLs to set the source and target regions. If the GUID-based URL for the container is `https://`**`storage-7b16fede61e1417ab83eb52e06f0e365`**`.storage.oraclecloud.com/v1/Storage-`**`7b16fede61e1417ab83eb52e06f0e365`**`/myContainer`, then <*fooN*> is `storage-7b16fede61e1417ab83eb52e06f0e365` and <*barN*> is `7b16fede61e1417ab83eb52e06f0e365`.

  ```
  {
    "sourceRegion": {
      "name": "us6",
      "url": "https://<foo1>.storage.oraclecloud.com/v1/Storage-<bar1>/
  myContainer"},
    "targetRegions": [{
  ```

```
    "name": "us2",
    "url": "https://<foo2>.storage.oraclecloud.com/v1/Storage-<bar2>/
myContainer"}, {
    "name": "uscom-central-1",
    "url": "https://<foo3>.storage.oraclecloud.com/v1/Storage-<bar3>/
myContainer"}],
  "externalSourceRegions": [],
  "externalTargetRegions": []
}
```

- Example request body JSON file for your container `mySecondContainer` in the following scenario:
  - Your service instance has `us2` primary DC and `us6` replication DC.
  - Primary DC for your container is `us2`.
  - Replication DC for your container is `us6` within the same region.
  - Your container is the destination for replication from a container `externalSourceContainer1` and `externalSourceContainer2` that reside in different regions.
  - The objects in your container are replicated to the container `externalTargetContainer` in a data center of a different region.

  Note that the examples demonstrate the use of GUID-based URLs to set the source and target regions. If the GUID-based URL for the container is `https://` **`storage-7b16fede61e1417ab83eb52e06f0e365`**`.storage.oraclecloud.com/v1/Storage-` **`7b16fede61e1417ab83eb52e06f0e365`**`/myContainer`, then <*fooN*> is `storage-7b16fede61e1417ab83eb52e06f0e365` and <*barN*> is `7b16fede61e1417ab83eb52e06f0e365`.

```
{
  "sourceRegion": {
    "name": "us2",
    "url": "https://<foo4>.storage.oraclecloud.com/v1/Storage-<bar4>/
mySecondContainer"},
  "targetRegions": [{
    "name": "us6",
    "url": "https://<foo5>.storage.oraclecloud.com/v1/Storage-<bar5>/
mySecondContainer"}],
  "externalSourceRegions": [{
    "name": "external",
    "url": "https://<foo6>.storage.oraclecloud.com/v1/Storage-<bar6>/
externalSourceContainer1"}, {
    "name": "external",
    "url": "https://<foo7>.storage.oraclecloud.com/v1/Storage-<bar7>/
externalSourceContainer2"}],
  "externalTargetRegions": [{
    "name": "external",
    "url": "https://<foo8>.storage.oraclecloud.com/v1/Storage-<bar8>/
externalTargetContainer"}]
}
```

cURL Command Syntax to Specify the Replication Policy for a Container

- To specify a replication policy for an empty container:

```
curl -v -X POST \
     -H "X-Auth-Token: token" \
     -H "Content-Type: application/json" \
```

```
     -d "@file" \
     accountURL/containerName?repPolicy
```

- To specify a replication policy while creating a container:

```
curl -v -X PUT \
     -H "X-Auth-Token: token" \
     -d "@file" \
     accountURL/containerName?repPolicy
```

- To read a container's replication policy:

```
curl -v -X GET \
     -H "X-Auth-Token: token" \
     accountURL/containerName?repPolicy
```

- `token` is the authentication token obtained earlier from Oracle Cloud Infrastructure Object Storage Classic. See Authenticating Access When Using the REST API.
- `file` is the full path and name of the file that contains the required container-level policy, in JSON format.
- For the syntax of `accountURL`, see About REST URLs for Oracle Cloud Infrastructure Object Storage Classic Resources.
- `containerName` is the container for which the replication policy must be set.

> **Note:**
>
> When you send a REST API request to Oracle Cloud Infrastructure Object Storage Classic, all non-ASCII characters in container names, object names and metadata values must be URL-encoded. For example, `my container` should be encoded as `my%20container`, where `%20` is the HTML encoding for the space character. Similarly, `my Über Container` should be encoded as `my %20%C3%9Cber%20Container`, where `%20` represents the space character and `%C3%9C` is the `Ü` character.

HTTP Response Codes

- Success:
  - When updating a container with the replication policy: `204 No Content`
  - When creating a container and specifying the replication policy: `201 Created`
  - When reading a container's replication policy: `200 OK`
- Failure: See Error Code Reference for Object Storage Classic

cURL Command Examples

1. This command sets the replication policy for the container `FirstContainer`:

```
curl -v -X POST \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     -H "Content-Type: application/json" \
     -d "@requestbody.json" \
     https://acme.storage.oraclecloud.com/v1/Storage-acme/FirstContainer?
repPolicy
```

The following is an example of the output of this command:

```
> POST /v1/Storage-acme/FirstContainer?repPolicy HTTP/1.1
> User-Agent: curl/7.49.1
> Host: acme.storage.oraclecloud.com
> Accept: */*
> Content-Type: application/json
> Content-Length: 489
> X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b
>
* upload completely sent off: 489 out of 489 bytes
< HTTP/1.1 204 No Content
< Date: Fri, 10 Nov 2017 06:32:55 GMT
< Content-Type: text/html;charset=UTF-8
< X-Trans-Id: txe8869b3edea348e5b49eb-0054f99743
```

2. This command specifies the replication policy while creating the container `SecondContainer`:

```
curl -v -X PUT \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     -d "@requestbody.json" \
     https://acme.storage.oraclecloud.com/v1/Storage-acme/SecondContainer?
repPolicy
```

The following is an example of the output of this command:

```
> PUT /v1/Storage-acme/SecondContainer?repPolicy HTTP/1.1
> User-Agent: curl/7.49.1
> Host: acme.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b
>
< HTTP/1.1 201 Created
< Date: Thu, 09 Nov 2017 10:03:18 GMT
< Content-Type: text/html; charset=UTF-8
< X-Trans-Id: txe8869b3edea348e5b49eb-0054f99678
< X-Last-Modified-Timestamp: 1510221797.90473
< Cache-Control: no-cache
< Pragma: no-cache
< Content-Language: en
```

3. This command displays the replication policy of the container `FirstContainer`:

```
curl -v -X GET \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     https://acme.storage.oraclecloud.com/v1/Storage-acme/FirstContainer?
repPolicy
```

The following is an example of the output of this command:

```
> GET /v1/Storage-acme/FirstContainer?repPolicy HTTP/1.1
> User-Agent: curl/7.49.1
> Host: acme.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b
>
< HTTP/1.1 200 OK
< < Date: Fri, 10 Nov 2017 06:17:44 GMT
< Content-Type: application/json;charset=UTF-8
< X-Trans-Id: txe8869b3edea348e5b49eb-0054f99838
{
  "mode": "ACTIVE_PASSIVE",
```

```
  "sourceRegion": {
    "name": "us2",
    "url": "https://<foo9>.storage.oraclecloud.com/v1/Storage-<bar9>/
FirstContainer"
  },
  "targetRegions": [{
    "name": "us6",
    "url": "https://<foo10>.storage.oraclecloud.com/v1/Storage-<bar10>/
FirstContainer"
  }],
  "externalSourceRegions": [{
    "name": "external",
    "url": "https://<foo11>.storage.oraclecloud.com/v1/Storage-<bar11>/
ExternalSourceContainer"
  }],
  "externalTargetRegions": [{
    "name": "external",
    "url": "https://<foo12>.storage.oraclecloud.com/v1/Storage-<bar12>/
ExternalTargetContainer"
  }]
* STATE: PERFORM => DONE handle 0x600057870; line 1955 (connection #0)
* multi_done
* Connection #0 to host left intact
}
```

Note that the example output displays the GUID-based URLs that are used to set the source and target regions. If the GUID-based URL for the container is `https://` **`storage-7b16fede61e1417ab83eb52e06f0e365`**`.storage.oraclecloud.com/v1/Storage-` **`7b16fede61e1417ab83eb52e06f0e365`**`/myContainer`, then <*fooN*> is `storage-7b16fede61e1417ab83eb52e06f0e365` and <*barN*> is `7b16fede61e1417ab83eb52e06f0e365`.

Deleting Containers

All objects within a container must first be deleted before the container can be deleted.

Any user with the Service Administrator role can perform this task.

You can delete containers by using the following interfaces:

Interface	Resources
Web Console (Not available on Oracle Cloud at Customer)	See Deleting a Container Using the Web Console.
RESTful API	See Delete Container in *REST API for Standard Storage in Oracle Cloud Infrastructure Object Storage Classic*.
Java Library	See `deleteContainer` in *Java API Reference for Oracle Cloud Infrastructure Object Storage Classic*.
File Transfer Manager CLI	See Deleting Containers in *Command-Line Reference for Oracle Cloud Infrastructure Object Storage Classic*.
File Transfer Manager API	See `deleteContainer` in *Java API Reference for Oracle Cloud Infrastructure Object Storage Classic File Transfer Manager*.

For information about deleting multiple objects in a container in a single operation, see Bulk-Deleting Objects.

Deleting a Container Using the Web Console

(Not available on Oracle Cloud at Customer)

1. Sign in to the Oracle Cloud Infrastructure Object Storage Classic console.
   The list of containers is displayed.
2. Identify the container that you want to delete.
3. Click **delete** on the left side of the container name.
   The **delete** button is displayed only for empty containers.

   The following message appears:

   ```
   Are you sure want to delete this container?
   ```
4. Click **OK**.
   The container is deleted.

Deleting a Container Using the REST API

All objects within a container must first be deleted before the container can be deleted. To find out whether a container contains any objects, send a `HEAD` request to the container URL.

cURL Command Syntax

```
curl -v -X DELETE \
     -H "X-Auth-Token: token" \
     accountURL/containerName
```

- `token` is the authentication token obtained earlier from Oracle Cloud Infrastructure Object Storage Classic. See Authenticating Access When Using the REST API.
- For the syntax of `accountURL`, see About REST URLs for Oracle Cloud Infrastructure Object Storage Classic Resources.
- `containerName` is the name of the container to be deleted.

> **Note:**
>
> When you send a REST API request to Oracle Cloud Infrastructure Object Storage Classic, all non-ASCII characters in container names, object names and metadata values must be URL-encoded. For example, `my container` should be encoded as `my%20container`, where `%20` is the HTML encoding for the space character. Similarly, `my Über Container` should be encoded as `my %20%C3%9Cber%20Container`, where `%20` represents the space character and `%C3%9C` is the Ü character.

HTTP Response Codes

- Success: `204 No Content`
- Failure: See Error Code Reference for Oracle Cloud Infrastructure Object Storage Classic

cURL Command Examples

Sample Cloud account with the following details:

- Account name: `acme`
- REST Endpoint URL: `https://acme.storage.oraclecloud.com/v1/Storage-acme`
- REST Endpoint (Permanent) URL: `https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/Storage-7b16fede61e1417ab83eb52e06f0e365`

> **Note:**
>
> The REST Endpoint (Permanent) is displayed for the accounts created after November 2017.

- Using the REST Endpoint URL obtained from the **REST Endpoint** field:

```
curl -v -X DELETE \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     https://acme.storage.oraclecloud.com/v1/Storage-acme/FirstContainer
```

The following is an example of the output of this command:

```
> DELETE /v1/Storage-acme/FirstContainer HTTP/1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.14.0.0
zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: acme.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b
>
< HTTP/1.1 204 No Content
< Date: Fri, 06 Mar 2015 10:43:38 GMT
< Content-Length: 0
< Content-Type: text/html; charset=UTF-8
< X-Trans-Id: txc100a7408d564f82916fb-0054f984da
< Cache-Control: no-cache
< Pragma: no-cache
< Content-Language: en
<
```

- Using the Service Permanent REST Endpoint URL obtained from the **REST Endpoint (Permanent)** field:

> **Note:**
>
> This cURL command example applies to the accounts created after November 2017.

```
curl -v -X DELETE \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/
Storage-7b16fede61e1417ab83eb52e06f0e365/FirstContainer
```

The following is an example of the output of this command:

```
> DELETE /v1/Storage-7b16fede61e1417ab83eb52e06f0e365/FirstContainer HTTP/1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.14.0.0
zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b
>
< HTTP/1.1 204 No Content
< Date: Fri, 06 Mar 2015 10:43:38 GMT
< Content-Length: 0
< Content-Type: text/html; charset=UTF-8
< X-Trans-Id: txc100a7408d564f82916fb-0054f984da
< Cache-Control: no-cache
< Pragma: no-cache
< Content-Language: en
<
```

Enabling Server-Side Encryption

Topics

- About Server-Side Encryption
- Enabling Server-Side Encryption Using the Web Console
- Enabling Server-Side Encryption Using an Oracle-Provided Key
- Enabling Server-Side Encryption Using a Customer-Provided Key
- Verifying Server-Side Encryption

About Server-Side Encryption

You can configure containers in your service instance to store all the data uploaded to them in an encrypted form. The encryption and decryption occur entirely on the server. When you download objects from such containers, the object is decrypted on the server and then delivered to you. Server-side encryption uses AES-256 encryption algorithm.

You can enable encryption for a container by adding the `X-Server-Side-Encryption` metadata header. This header can have one of the following values:

- `BASE_ENCRYPTION`: Indicates that data in the container must be encrypted.
- `NONE`: Indicates that server-side encryption is not enabled for the container. This is equivalent to omitting this header entirely.

Provide your own Base64-encoded AES-256 master encryption key for a specific container. You must include the `X-Server-Side-Encryption-Container-Key` metadata header and provide your master key, in addition to the `X-Server-Side-Encryption` metadata header.

If you do not specify the `X-Server-Side-Encryption-Container-Key` header, the container is encrypted using an Oracle-provided, randomly generated master key.

> **Note:**
>
> You can enable server-side encryption only while creating a container.
>
> Server-side encryption is not available in all Oracle data regions. If you don't see the **Enable Encryption** option under **Advanced** in the **Create a Container** dialog box in the web console, then this feature is not available in your data region.
>
> You can't enable or disable this feature for an existing container. Once server-side encryption is set, the feature is immutable.

Enabling Server-Side Encryption Using the Web Console


This topic does not apply to Oracle Cloud at Customer.

> **Note:**
>
> Server-side encryption can be enabled only while creating a container, not later.

1. Sign in to the Oracle Cloud Infrastructure Object Storage Classic console.
2. Click **Create Container**.
   The **Create Container** dialog box is displayed.
3. Enter a name for the container.
   For the list of character restrictions while creating a container, see Character Restrictions.
4. Select `Standard` or `Archive` in the **Storage Class** field, based on your choice of the storage class for the container.
5. To enable server-side encryption of the data stored in the container, select **Enable Encryption**.
   All the data stored in the container is encrypted using an Oracle-provided key.
6. To enable server-side encryption using your master encryption key:
   - Generate an AES 256 encryption key using the OpenSSL toolkit.
     `openssl enc -aes256 -k password -P`

     Example:

     ```
     openssl enc -aes256 -k mypassword -P
     ```

     Output:

     ```
     salt=ADCA338FB4594CC6
     key=08A8C5A2C81EB0508AAA1EAB7C81BC7AC9747E3E752E04FECAEE8D09E83A8C09
     iv=4586BFB397A9CE3FFB7F90D14BF6B506
     ```

   - Encode the key in Base64 format.
     `echo 'value_from_key_field' | xxd -r -p | base64`

     Example:

```
echo
"08A8C5A2C81EB0508AAA1EAB7C81BC7AC9747E3E752E04FECAEE8D09E83A8C09"
| xxd -r -p | base64
```

Output:

```
CKjFosgesFCKqh6rfIG8esl0fj51LgT+yu6NCeg6jAk=
```

- Enter the Base64-encoded AES-256 encryption key in the **Server Side Encryption w/ Customer Master Keys** field in the console.

7. Click **Create**.
   The container is created and displayed in the console.

Enabling Server-Side Encryption Using an Oracle-Provided Key

To enable server-side encryption, create a container and assign it the `X-Server-Side-Encryption` metadata header. All objects written to a container with this header will be encrypted.

```
curl -v -X PUT \
     -H "X-Auth-Token: token" \
     -H "X-Server-Side-Encryption: BASE_ENCRYPTION" \
     accountURL/containerName
```

- `token` is the authentication token obtained earlier from Oracle Cloud Infrastructure Object Storage Classic. See Authenticating Access When Using the REST API.
- For the syntax of `accountURL`, see About REST URLs for Oracle Cloud Infrastructure Object Storage Classic Resources.
- `containerName` is the name of the container to be created.
  For the list of character restrictions while creating a container, see Character Restrictions.
- `X-Server-Side-Encryption` is the header to enable server-side encryption.

Example:

Create a container `MyEncryptedCont` and assign the server-side encryption metadata header.

```
curl -v -X PUT \
     -H "X-Auth-Token: AUTH_tk57dee23521d6e4d809deeffa6bd23cd6"
     -H "X-Server-Side-Encryption: BASE_ENCRYPTION"
     https://acme.storage.oraclecloud.com/v1/Storage-acme/MyEncryptedCont
```

The following is an example of the output of this command:

```
PUT /v1/Storage-acme/MyEncryptedCont HTTP/1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.14.0.0 zlib/
1.2.3 libidn/1.18 libssh2/1.4.2
> Host: acme.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b
> X-Server-Side-Encryption: BASE_ENCRYPTION
< HTTP/1.1 201 Created
< Date: Wed, 06 Dec 2016 10:34:20 GMT
< Content-Length: 0
< Content-Type: text/html; charset=UTF-8
< X-Trans-Id: tx23a1084b8c674fdeae8d4-0054f982ac
< Cache-Control: no-cache
```

```
< Pragma: no-cache
< Content-Language: en
```

Enabling Server-Side Encryption Using a Customer-Provided Key

To enable server-side encryption using your master encryption key, generate a Base64-encoded AES-256 encryption key and assign the following headers when you create the container:

- `X-Server-Side-Encryption`
- `X-Server-Side-Encryption-Container-Key`

1. Generate an AES 256 encryption key.
   `openssl enc -aes256 -k password -P`

   Example:

   ```
   openssl enc -aes256 -k mypassword -P
   ```

   Output:

   ```
   salt=ADCA338FB4594CC6
   key=08A8C5A2C81EB0508AAA1EAB7C81BC7AC9747E3E752E04FECAEE8D09E83A8C09
   iv=4586BFB397A9CE3FFB7F90D14BF6B506
   ```

2. Encode the key in Base64 format.
   `echo 'value_from_key_field' | xxd -r -p | base64`

   Example:

   ```
   echo
   "08A8C5A2C81EB0508AAA1EAB7C81BC7AC9747E3E752E04FECAEE8D09E83A8C09" |
   xxd -r -p | base64
   ```

   Output:

   ```
   CKjFosgesFCKqh6rfIG8esl0fj51LgT+yu6NCeg6jAk=
   ```

3. Create a container and assign the server-side encryption metadata headers.

   ```
   curl -v -X PUT \
        -H "X-Auth-Token: token" \
        -H "X-Server-Side-Encryption: BASE_ENCRYPTION" \
        -H "X-Server-Side-Encryption-Container-Key: master_encryption_key"
        accountURL/containerName
   ```

   - `token` is the authentication token obtained earlier from Oracle Cloud Infrastructure Object Storage Classic. See Authenticating Access When Using the REST API.
   - For the syntax of `accountURL`, see About REST URLs for Oracle Cloud Infrastructure Object Storage Classic Resources.
   - `containerName` is the name of the container to be created.
     For the list of character restrictions while creating a container, see Character Restrictions.
   - `X-Server-Side-Encryption` is the header to enable server-side encryption.
   - `master_encryption_key` is the Base64–encoded AES-256 encryption key generated in the previous step.

   Example:

Create a container `MyEncryptedCont` and assign the server-side encryption metadata headers.

```
curl -v -X PUT \
     -H "X-Auth-Token: AUTH_tk57dee23521d6e4d809deeffa6bd23cd6"
     -H "X-Server-Side-Encryption: BASE_ENCRYPTION"
     -H "X-Server-Side-Encryption-Container-Key: CKjFosgesFCKqh6rfIG8esl0fj51LgT
+yu6NCeg6jAk="
     https://acme.storage.oraclecloud.com/v1/Storage-acme/MyEncryptedCont
```

> **Note:**
>
> The URL of the account in this example is `https://acme.storage.oraclecloud.com/v1/Storage-acme`. Replace this URL with the URL for your account. For the steps to find out your account URL, see About REST URLs for Oracle Cloud Infrastructure Object Storage Classic Resources.

The following is an example of the output of this command:

```
PUT /v1/Storage-acme/MyEncryptedCont HTTP/1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.14.0.0
zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: acme.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b
> X-Server-Side-Encryption: BASE_ENCRYPTION
> X-Server-Side-Encryption-Container-Key: CKjFosgesFCKqh6rfIG8esl0fj51LgT
+yu6NCeg6jAk=
< HTTP/1.1 201 Created
< Date: Wed, 07 Dec 2016 10:34:20 GMT
< Content-Length: 0
< Content-Type: text/html; charset=UTF-8
< X-Trans-Id: tx23a1084b8c674fdeae8d4-0054f982ac
< Cache-Control: no-cache
< Pragma: no-cache
< Content-Language: en
```

Verifying Server-Side Encryption

Verify the container metadata to check if server-side encryption is enabled for the container.

```
curl -v -X HEAD \
     -H "X-Auth-Token:token" \
     accountURL/containerName
```

- `token` is the authentication token obtained earlier from Oracle Cloud Infrastructure Object Storage Classic. See Authenticating Access When Using the REST API.
- For the syntax of `accountURL`, see About REST URLs for Oracle Cloud Infrastructure Object Storage Classic Resources.
- `containerName` is the name of the container to be created.

Example:

```
curl -v -X HEAD \
     -H "X-Auth-Token: AUTH_tk64e7143df33fcbf2f20047c3c37984db" \
      https://acme.storage.oraclecloud.com/v1/Storage-acme/MyEncryptedCont
```

The following is an example of the output of this command:

```
> HEAD /v1/Storage-acme/MyEncryptedCont HTTP/1.1
> User-Agent: curl/7.29.0
> Host: acme.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tk64e7143df33fcbf2f20047c3c37984db
<
< HTTP/1.1 204 No Content
< Date: Wed, 7 Dec 2016 11:40:11 GMT
< X-Container-Object-Count: 0
< X-Container-Write: Storage-acme.Storage.Storage_ReadWriteGroup,Storage-
acme.myCustomRole
< X-Container-Meta-Category: Books
< Accept-Ranges: bytes
< X-Container-Meta-Quota-Count: 100
< X-Timestamp: 1425639066.56315
< X-Container-Read: .r:*,.rlistings
< X-Server-Side-Encryption: BASE_ENCRYPTION
< X-Container-Bytes-Used: 0
< Content-Type: text/plain; charset=utf-8
< X-Container-Meta-Quota-Bytes: 10737418240
< X-Trans-Id: txb0b9882eceba45b287266-0054f9921b
< Cache-Control: no-cache
< Pragma: no-cache
< Content-Language: en
```

The `X-Server-Side-Encryption` metadata header is displayed in the output with the value `BASE_ENCRYPTION`, confirming that server-side encryption is enabled in the container.

Getting Container Metadata

Any user with the Service Administrator role or a role that is specified in the `X-Container-Read` ACL of the container can perform this task.

You can retrieve information about an object in a container by sending a `HEAD` request, which returns the following information:

- Container ACLs (`X-Container-Read` and `X-Container-Write`)
- Container quotas: (`X-Container-Meta-Quota-Count` and `X-Container-Meta-Quota-Bytes`)
- Number of objects in the container (`X-Container-Object-Count`)
- Storage space used by all objects in the container, in bytes (`X-Container-Bytes-Used`)
- Custom metadata (`X-Container-Meta-`*`Name`*)
- Storage class of the container (`X-Storage-Class`), returned only for `Archive` containers

You can view the container metadata by using the following interfaces:

Interface	Resources
Oracle Cloud Infrastructure Object Storage Classic Console (Not available on Oracle Cloud at Customer)	See Getting Container Metadata Using the Web Console.
RESTful API	See Show Container Metadata in *REST API for Standard Storage in Oracle Cloud Infrastructure Object Storage Classic*.
File Transfer Manager API	See `getContainer` in *Java API Reference for Oracle Cloud Infrastructure Object Storage Classic File Transfer Manager*.
File Transfer Manager CLI	See Getting Container Metadata in *Command-Line Reference for Oracle Cloud Infrastructure Object Storage Classic*.

Getting Container Metadata Using the Web Console

(Not available on Oracle Cloud at Customer)

1. Sign in to the Oracle Cloud Infrastructure Object Storage Classic console.
2. Select a container from the list. Expand **Container Information** pane for the container.
   The details of the container are displayed in the **Container Information** pane.

Getting Container Metadata Using the REST API

cURL Command Syntax

```
curl -v -X HEAD \
     -H "X-Auth-Token: token" \
     accountURL/containerName
```

- `token` is the authentication token obtained earlier from Oracle Cloud Infrastructure Object Storage Classic. See Authenticating Access When Using the REST API.
- For the syntax of `accountURL`, see About REST URLs for Oracle Cloud Infrastructure Object Storage Classic Resources.
- `containerName` is the name of the container for which metadata should be retrieved.

> **Note:**
>
> When you send a REST API request to Oracle Cloud Infrastructure Object Storage Classic, all non-ASCII characters in container names, object names and metadata values must be URL-encoded. For example, `my container` should be encoded as `my%20container`, where `%20` is the HTML encoding for the space character. Similarly, `my Über Container` should be encoded as `my %20%C3%9Cber%20Container`, where `%20` represents the space character and `%C3%9C` is the Ü character.

HTTP Response Codes

- Success: `204 No Content`
- Failure: See Error Code Reference for Oracle Cloud Infrastructure Object Storage Classic

cURL Command Examples

Sample Cloud account with the following details:

- Account name: `acme`
- REST Endpoint URL: `https://acme.storage.oraclecloud.com/v1/Storage-acme`
- REST Endpoint (Permanent) URL: `https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/Storage-7b16fede61e1417ab83eb52e06f0e365`

> **Note:**
>
> The REST Endpoint (Permanent) URL is displayed for the accounts created after November 2017.

- Using the REST Endpoint URL obtained from the **REST Endpoint** field:

```
curl -v -X HEAD \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     https://acme.storage.oraclecloud.com/v1/Storage-acme/FirstContainer
```

The following is an example of the output of this command for a `Standard` container:

```
> HEAD /v1/Storage-acme/FirstContainer HTTP/1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.14.0.0
zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: acme.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b
>
< HTTP/1.1 204 No Content
< Date: Fri, 06 Mar 2015 11:40:11 GMT
< X-Container-Object-Count: 0
< X-Container-Write: Storage-acme.Storage.Storage_ReadWriteGroup,Storage-
acme.myCustomRole
< X-Container-Meta-Category: Books
< Accept-Ranges: bytes
< X-Container-Meta-Quota-Count: 100
< X-Timestamp: 1425639066.56315
< X-Container-Read: .r:*,.rlistings
< X-Container-Bytes-Used: 0
< Content-Type: text/plain; charset=utf-8
< X-Container-Meta-Quota-Bytes: 10737418240
< X-Trans-Id: txb0b9882eceba45b287266-0054f9921b
< Cache-Control: no-cache
< Pragma: no-cache
< Content-Language: en
```

For an `Archive` container, the header `X-Storage-Class` displays the value `Archive` in the metadata. The following is an example of this command:

```
> HEAD /v1/Storage-acme/firstArchiveContainer1 HTTP/1.1
> User-Agent: curl/7.29.0
> Host: acme.storage.oraclecloud.com:443
> Accept: */*
> X-Auth-Token: AUTH_tk32486a5bbed54ee74213a30d6d128f9d
>
< HTTP/1.1 204 No Content
< X-Container-Object-Count: 0
< X-Container-Write: cloudua001.Storage.Storage_ReadWriteGroup
< Accept-Ranges: bytes
< X-Timestamp: 1434733473.12286
< X-Container-Read:
cloudua001.Storage.Storage_ReadOnlyGroup,cloudua001.Storage.Storage_ReadWriteGrou
p
< X-Container-Bytes-Used: 0
< X-Trans-Id: txfcffc30b5a0c44cb91491-005584544ega
< Date: Fri, 19 Jun 2015 17:41:34 GMT
< X-Storage-Class: Archive
< Content-Type: text/plain;charset=utf-8
```

- Using the Service Permanent REST Endpoint URL obtained from the **REST Endpoint (Permanent)** field:

> **Note:**
>
> This cURL command example applies to the accounts created after November 2017.

```
curl -v -X HEAD \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/
Storage-7b16fede61e1417ab83eb52e06f0e365/FirstContainer
```

The following is an example of the output of this command for a `Standard` container:

```
> HEAD /v1/Storage-7b16fede61e1417ab83eb52e06f0e365/FirstContainer HTTP/1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.14.0.0
zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b
>
< HTTP/1.1 204 No Content
< Date: Fri, 06 Mar 2015 11:40:11 GMT
< X-Container-Object-Count: 0
< X-Container-Write:
Storage-7b16fede61e1417ab83eb52e06f0e365.Storage.Storage_ReadWriteGroup,Storage-7
b16fede61e1417ab83eb52e06f0e365.myCustomRole
< X-Container-Meta-Category: Books
< Accept-Ranges: bytes
< X-Container-Meta-Quota-Count: 100
< X-Timestamp: 1425639066.56315
< X-Container-Read: .r:*,.rlistings
< X-Container-Bytes-Used: 0
< Content-Type: text/plain; charset=utf-8
< X-Container-Meta-Quota-Bytes: 10737418240
< X-Trans-Id: txb0b9882eceba45b287266-0054f9921b
< Cache-Control: no-cache
```

```
< Pragma: no-cache
< Content-Language: en
```

For an `Archive` container, the header `X-Storage-Class` displays the value `Archive` in the metadata. The following is an example of this command:

```
> HEAD /v1/Storage-7b16fede61e1417ab83eb52e06f0e365/firstArchiveContainer1 HTTP/
1.1
> User-Agent: curl/7.29.0
> Host: storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com:443
> Accept: */*
> X-Auth-Token: AUTH_tk32486a5bbed54ee74213a30d6d128f9d
>
< HTTP/1.1 204 No Content
< X-Container-Object-Count: 0
< X-Container-Write: cloudua001.Storage.Storage_ReadWriteGroup
< Accept-Ranges: bytes
< X-Timestamp: 1434733473.12286
< X-Container-Read:
cloudua001.Storage.Storage_ReadOnlyGroup,cloudua001.Storage.Storage_ReadWriteGrou
p
< X-Container-Bytes-Used: 0
< X-Trans-Id: txfcffc30b5a0c44cb91491-005584544ega
< Date: Fri, 19 Jun 2015 17:41:34 GMT
< X-Storage-Class: Archive
< Content-Type: text/plain;charset=utf-8
```

Deleting Container Metadata

You can delete container metadata by using the following interfaces:

Interface	Resources
Oracle Cloud Infrastructure Object Storage Classic Console (Not available on Oracle Cloud at Customer)	See Deleting Container Metadata Using the Web Console.
RESTful API	See: • Deleting Container Metadata Using the REST API • Create, update, or delete container metadata in *REST API for Standard Storage in Oracle Cloud Infrastructure Object Storage Classic*
Java Library	See `deleteContainerMetadata` in *Java API Reference for Oracle Cloud Infrastructure Object Storage Classic*.

Deleting Container Metadata Using the Web Console

(Not available on Oracle Cloud at Customer)

1. Sign in to the Oracle Cloud Infrastructure Object Storage Classic console.
2. Select the container from which you would like to delete the metadata.
3. Expand **Container Information**.
   The details of the container are displayed.

4. Click **Edit**.
   Look for the **Custom Metadata** field.
5. Identify the metadata name and value that you want to delete.
6. Click  on the right side of the metadata value.
7. Click **Save**.
   The metadata name and value are deleted.

Deleting Container Metadata Using the REST API

cURL Command Syntax

```
curl -v -X POST \
     -H "X-Auth-Token: token" \
     -H "X-Remove-Container-Meta-Name: any_arbitrary_string" \
     accountURL/containerName
```

- `token` is the authentication token obtained earlier from Oracle Cloud Infrastructure Object Storage Classic. See Authenticating Access When Using the REST API.
- `Name` and `value` are the metadata key and value to be deleted.
- For the syntax of `accountURL`, see About REST URLs for Oracle Cloud Infrastructure Object Storage Classic Resources.
- `containerName` is the name of the container from which the metadata should be deleted.

> **Note:**
>
> When you send a REST API request to Oracle Cloud Infrastructure Object Storage Classic, all non-ASCII characters in container names, object names and metadata values must be URL-encoded. For example, `my container` should be encoded as `my%20container`, where `%20` is the HTML encoding for the space character. Similarly, `my Über Container` should be encoded as `my %20%C3%9Cber%20Container`, where `%20` represents the space character and `%C3%9C` is the Ü character.

HTTP Response Codes

- Success: `204 No Content`
- Failure: See Error Code Reference for Oracle Cloud Infrastructure Object Storage Classic

cURL Command Examples

Sample Cloud account with the following details:

- Account name: `acme`
- REST Endpoint URL: `https://acme.storage.oraclecloud.com/v1/Storage-acme`
- REST Endpoint (Permanent) URL: `https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/Storage-7b16fede61e1417ab83eb52e06f0e365`

> **Note:**
>
> The REST Endpoint (Permanent) URL is displayed for the accounts created after November 2017.

- Using the REST Endpoint URL obtained from the **REST Endpoint** field:

```
curl -v -X POST \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     -H "X-Remove-Container-Meta-Category: Books" \
     https://acme.storage.oraclecloud.com/v1/Storage-acme/FirstContainer
```

The following is an example of the output of this command:

```
> POST /v1/Storage-acme/FirstContainer HTTP/1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.14.0.0
zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: acme.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b
> X-Remove-Container-Meta-Category: Books
>
< HTTP/1.1 204 No Content
< X-Trans-Id: tx30d406ea72b340378476a-00585b78c7ga
< Date: Thu, 22 Dec 2016 06:55:04 GMT
< Connection: keep-alive
< X-Last-Modified-Timestamp: 1482389703.89669
< Content-Type: text/html;charset=UTF-8
<
```

- Using the Service Permanent REST Endpoint URL obtained from the **REST Endpoint (Permanent)** field:

> **Note:**
>
> This cURL command example applies to the accounts created after November 2017.

```
curl -v -X POST \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     -H "X-Remove-Container-Meta-Category: Books" \
     https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/
Storage-7b16fede61e1417ab83eb52e06f0e365/FirstContainer
```

The following is an example of the output of this command:

```
> POST /v1/Storage-7b16fede61e1417ab83eb52e06f0e365/FirstContainer HTTP/1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.14.0.0
zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b
> X-Remove-Container-Meta-Category: Books
>
< HTTP/1.1 204 No Content
< X-Trans-Id: tx30d406ea72b340378476a-00585b78c7ga
< Date: Thu, 22 Dec 2016 06:55:04 GMT
```

```
< Connection: keep-alive
< X-Last-Modified-Timestamp: 1482389703.89669
< Content-Type: text/html;charset=UTF-8
<
```

Setting Container Metadata

Topics:

- Setting Container ACLs
- Setting Container Quotas
- Setting Custom Metadata for Containers
- Enabling CORS for a Container
- Making Objects in a Container Immutable

Setting Container ACLs

The ability to read and write objects in a container is governed by the Access Control Lists (ACLs) assigned to the container. Any user with the Service Administrator role can perform this task.

A container has two ACLs, `X-Container-Read` and `X-Container-Write`.

The `X-Container-Read` ACL consists of a comma-separated list of roles or *referrer designations*. The `X-Container-Write` ACL consists of a comma-separated list of roles.

- The roles can be built-in roles or custom roles. Custom roles are defined in the **Users** page in the Infrastructure Classic Console.
  - For a role that was provisioned as part of another service instance, the format is *`domainName.serviceName.roleName`*
  - For a custom role, the format is *`domainName.roleName`*
- A referrer designation indicates the host (or hosts) for which read access to the container should be allowed or denied. When the server receives a request for the container, it compares the referrer designations specified in the `X-Container-Read` ACL with the value of the `Referer` header in the request, and determines whether access should be allowed or denied. The syntax of the referrer designation is: `.r:`*`value`*
  - `value` indicates the host for which access to the container should be allowed. It can be a specific host name (example: `.r:www.example.com`), a domain (example: `.r:.example.com`), or an asterisk (`.r:*`) to indicate all hosts. Note that if `.r:*` is specified, objects in the container will be publicly readable without authentication.
  - A minus sign (`-`) before `value` (example: `.r:-temp.example.com`) indicates that the host specified in the `value` field must be denied access to the container.
  - By default, read access to a container does not include permission to list the objects in the container. To allow listing of objects as well, include the `.rlistings` directive in the ACL (example: `.r:*,.rlistings`).

For creating custom roles for a traditional Cloud account. See Adding a Custom Role in *Managing and Monitoring Oracle Cloud*.

For creating custom roles for accounts with Identity Cloud Service, see Create a Custom Role for Cloud Accounts with Identity Cloud Service.

You can set container ACLs by using the following interfaces:

Interface	Resources
Web Console (Not available on Oracle Cloud at Customer)	See Setting Container ACLs Using the Web Console.
RESTful API	See Create, update, or delete container metadata in *REST API for Standard Storage in Oracle Cloud Infrastructure Object Storage Classic*.
Java Library	See `setContainerAcl` in *Java API Reference for Oracle Cloud Infrastructure Object Storage Classic*.
File Transfer Manager API	See setWriteAcl() in *Java API Reference for Oracle Cloud Infrastructure Object Storage Classic File Transfer Manager*.

Setting Container ACLs Using the Web Console

(Not available on Oracle Cloud at Customer)

1. Sign in to the Oracle Cloud Infrastructure Object Storage Classic console.
   The list of containers is displayed.
2. Select the container for which you would like to set the read/write access.
3. Expand **Container Information**.
   The details of the container are displayed.
4. Click **Edit**.
   Look for **Permissions** under **Container Properties**.
5. Click **Add Role**.
6. Enter the desired value in the empty field.
   By default, the following ACLs are set:
   - `Storage_ReadWriteGroup` to provide both read and write access to a user.
   - `Storage_ReadOnlyGroup` to provide read only access to a user
7. Click **Save**.

Setting Container ACLs Using the REST API

cURL Command Syntax

```
curl -v -X POST \
     -H "X-Auth-Token: token" \
     -H "X-Container-Read: item[,item...]" \
     -H "X-Container-Write: item[,item...]" accountURL/containerName
```

> **Note:**
>
> When you send a REST API request to Oracle Cloud Infrastructure Object Storage Classic, all non-ASCII characters in container names, object names and metadata values must be URL-encoded. For example, `my container` should be encoded as `my%20container`, where `%20` is the HTML encoding for the space character. Similarly, `my Über Container` should be encoded as `my%20%C3%9Cber%20Container`, where `%20` represents the space character and `%C3%9C` is the Ü character.

HTTP Response Codes

- Success: `204 No Content`
- Failure: See Error Code Reference for Oracle Cloud Infrastructure Object Storage Classic

cURL Command Examples

Sample Cloud account with the following details:

- IDCS GUID: `idcs-b75f75ed2528447fb59a798c1f08a38d` (for a sample IDCS account)
- Account name: `acme` (for a sample IDCS or traditional account)
- REST Endpoint URL for the sample account: `https://acme.storage.oraclecloud.com/v1/Storage-acme`
- REST Endpoint (Permanent) URL for the sample account: `https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/Storage-7b16fede61e1417ab83eb52e06f0e365`

> **Note:**
>
> The REST Endpoint (Permanent) URL is displayed for the accounts created after November 2017.

The following commands set up ACLs for the container named `FirstContainer`:

- Provide write access for any user with the predefined role, `Storage_ReadWriteGroup` and the custom role, `myCustomRole`:
  - Using the REST Endpoint URL obtained from the **REST Endpoint** field:

```
curl -v -X POST \
        -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
        -H "X-Container-Write:
myDomain.Storage.Storage_ReadWriteGroup,myDomain.myCustomRole" \
        https://acme.storage.oraclecloud.com/v1/Storage-acme/FirstContainer
```

    The following is an example of the output of this command:

```
> POST /v1/Storage-acme/FirstContainer HTTP/1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/
3.14.0.0 zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: acme.storage.oraclecloud.com
```

```
> Accept: */*
> X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b
> X-Container-Write:Storage-acme.Storage.Storage_ReadWriteGroup,Storage-
acme.myCustomRole
>
< HTTP/1.1 204 No Content
< Date: Fri, 06 Mar 2015 11:19:21 GMT
< Content-Length: 0
< Content-Type: text/html; charset=UTF-8
< X-Trans-Id: txbf2c736d57494bf88e76a-0054f98d39
< Cache-Control: no-cache
< Pragma: no-cache
< Content-Language: en
```

- Using the Service Permanent REST Endpoint URL obtained from the **REST Endpoint (Permanent)** field:

  > **Note:**
  >
  > This cURL command example applies to the accounts created after November 2017.

  ```
  curl -v -X POST \
          -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
          -H "X-Container-Write:
  myDomain.Storage.Storage_ReadWriteGroup,myDomain.myCustomRole" \
          https://
  storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/
  Storage-7b16fede61e1417ab83eb52e06f0e365/FirstContainer
  ```

  The following is an example of the output of this command:

  ```
  > POST /v1/Storage-7b16fede61e1417ab83eb52e06f0e365/FirstContainer HTTP/1.1
  > User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.14.0.0
  zlib/1.2.3 libidn/1.18 libssh2/1.4.2
  > Host: acme.storage.oraclecloud.com
  > Accept: */*
  > X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b
  > X-Container-
  Write:Storage-7b16fede61e1417ab83eb52e06f0e365.Storage.Storage_ReadWriteGroup,Sto
  rage-7b16fede61e1417ab83eb52e06f0e365.myCustomRole
  >
  < HTTP/1.1 204 No Content
  < Date: Fri, 06 Mar 2015 11:19:21 GMT
  < Content-Length: 0
  < Content-Type: text/html; charset=UTF-8
  < X-Trans-Id: txbf2c736d57494bf88e76a-0054f98d39
  < Cache-Control: no-cache
  < Pragma: no-cache
  < Content-Language: en
  ```

- Provide read access for all hosts and also allow listing of the objects in the container:
  - Using the REST Endpoint URL obtained from the **REST Endpoint** field:

    ```
    curl -v -X POST \
         -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
         -H "X-Container-Read: .r:*,.rlistings" \
         https://acme.storage.oraclecloud.com/v1/Storage-acme/FirstContainer
    ```

The following is an example of the output of this command:

```
> POST /v1/Storage-acme/FirstContainer HTTP/1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/
3.14.0.0 zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: acme.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b
> X-Container-Read: .r:*,.rlistings
>
< HTTP/1.1 204 No Content
< Date: Fri, 06 Mar 2015 11:23:16 GMT
< Content-Length: 0
< Content-Type: text/html; charset=UTF-8
< X-Trans-Id: tx9127a70f18144c17afce5-0054f98e24
< Cache-Control: no-cache
< Pragma: no-cache
< Content-Language: en
0
```

– Using the Service Permanent REST Endpoint URL obtained from the **REST Endpoint (Permanent)** field:

> **Note:**
>
> This cURL command example applies to the accounts created after November 2017.

```
curl -v -X POST \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     -H "X-Container-Read: .r:*,.rlistings" \
     https://
storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/
Storage-7b16fede61e1417ab83eb52e06f0e365/FirstContainer
```

The following is an example of the output of this command:

```
> POST /v1/Storage-7b16fede61e1417ab83eb52e06f0e365/FirstContainer HTTP/1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/
3.14.0.0 zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: acme.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b
> X-Container-Read: .r:*,.rlistings
>
< HTTP/1.1 204 No Content
< Date: Fri, 06 Mar 2015 11:23:16 GMT
< Content-Length: 0
< Content-Type: text/html; charset=UTF-8
< X-Trans-Id: tx9127a70f18144c17afce5-0054f98e24
< Cache-Control: no-cache
< Pragma: no-cache
< Content-Language: en
0
```

Setting Container Quotas

For each container, you can set quotas for the maximum number of bytes the container can contain (`X-Container-Meta-Quota-Bytes`) and the maximum number of objects the container can contain (`X-Container-Meta-Quota-Count`).

Any user with the Service Administrator role can perform this task.

You can set container quotas by using the following interfaces:

Interface	Resources
RESTful API	See Create, update, or delete container metadata in *REST API for Standard Storage in Oracle Cloud Infrastructure Object Storage Classic*.
File Transfer Manager CLI	See setQuotaBytes()in *Command-Line Reference for Oracle Cloud Infrastructure Object Storage Classic*.

cURL Command Syntax

```
curl -v -X POST \
     -H "X-Auth-Token: token" \
     -H "X-Container-Meta-Quota-Bytes: maxBytes" \
     -H "X-Container-Meta-Quota-Count: maxObjects" accountURL/containerName
```

- `token` is the authentication token obtained earlier from Oracle Cloud Infrastructure Object Storage Classic. See Authenticating Access When Using the REST API.
- `maxBytes` is the maximum number of bytes of data that can be stored in the container.
- `maxObjects` is the maximum number of objects that can be created in the container.
- For the syntax of `accountURL`, see About REST URLs for Oracle Cloud Infrastructure Object Storage Classic Resources.
- `containerName` is the name of the container for which the quota should be set.

> **Note:**
>
> When you send a REST API request to Oracle Cloud Infrastructure Object Storage Classic, all non-ASCII characters in container names, object names and metadata values must be URL-encoded. For example, `my container` should be encoded as `my%20container`, where `%20` is the HTML encoding for the space character. Similarly, `my Über Container` should be encoded as `my %20%C3%9Cber%20Container`, where `%20` represents the space character and `%C3%9C` is the Ü character.

HTTP Response Codes

- Success: `204 No Content`
- Failure: See Error Code Reference for Oracle Cloud Infrastructure Object Storage Classic

For information about getting container quota, see Getting Container Metadata.

cURL Command Examples

Sample Cloud account with the following details:

- Account name: `acme`
- REST Endpoint URL: `https://acme.storage.oraclecloud.com/v1/Storage-acme`
- REST Endpoint (Permanent) URL: `https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/Storage-7b16fede61e1417ab83eb52e06f0e365`

> **Note:**
>
> The REST Endpoint (Permanent) URL is displayed for the accounts created after November 2017.

- Using the REST Endpoint URL obtained from the **REST Endpoint** field:

```
curl -v -X POST \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     -H "X-Container-Meta-Quota-Bytes: 10737418240" \
     -H "X-Container-Meta-Quota-Count: 100" \
     https://acme.storage.oraclecloud.com/v1/Storage-acme/FirstContainer
```

This command sets a quota of 10737418240 bytes (10 GB) and 100 objects for the container named `FirstContainer`. The following is an example of the output of this command:

```
> POST /v1/Storage-acme/FirstContainer HTTP/1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.14.0.0
zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: acme.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b
> X-Container-Meta-Quota-Bytes: 10737418240
> X-Container-Meta-Quota-Count: 100
>
< HTTP/1.1 204 No Content
< Date: Fri, 06 Mar 2015 11:32:19 GMT
< Content-Length: 0
< Content-Type: text/html; charset=UTF-8
< X-Trans-Id: txe8869b3edea348e5b49eb-0054f99043
< Cache-Control: no-cache
< Pragma: no-cache
< Content-Language: en
```

- Using the Service Permanent REST Endpoint URL obtained from the **REST Endpoint (Permanent)** field:

```
curl -v -X POST \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     -H "X-Container-Meta-Quota-Bytes: 10737418240" \
     -H "X-Container-Meta-Quota-Count: 100" \
     https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/
Storage-7b16fede61e1417ab83eb52e06f0e365/FirstContainer
```

This command sets a quota of 10737418240 bytes (10 GB) and 100 objects for the container named `FirstContainer`. The following is an example of the output of this command:

```
> POST /v1/Storage-7b16fede61e1417ab83eb52e06f0e365/FirstContainer HTTP/1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.14.0.0
zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b
> X-Container-Meta-Quota-Bytes: 10737418240
> X-Container-Meta-Quota-Count: 100
>
< HTTP/1.1 204 No Content
< Date: Fri, 06 Mar 2015 11:32:19 GMT
< Content-Length: 0
< Content-Type: text/html; charset=UTF-8
< X-Trans-Id: txe8869b3edea348e5b49eb-0054f99043
< Cache-Control: no-cache
< Pragma: no-cache
< Content-Language: en
```

Setting Custom Metadata for Containers

Custom metadata are arbitrary key-value pairs associated with a container. You may create any custom or arbitrary metadata you need.

Any user with the Service Administrator role can perform this task.

You can set custom metadata for containers by using the following interfaces:

Interface	Resources
Web Console (Not available on Oracle Cloud at Customer)	See Setting Custom Metadata Using the Web Console.
RESTful API	See Create, update, or delete container metadata in *REST API for Standard Storage in Oracle Cloud Infrastructure Object Storage Classic*.
Java Library	See `updateContainerMetadata` in *Java API Reference for Oracle Cloud Infrastructure Object Storage Classic*.
File Transfer Manager CLI	See setCustomMetadata() in *Command-Line Reference for Oracle Cloud Infrastructure Object Storage Classic*.

Setting Custom Metadata Using the Web Console

(Not available on Oracle Cloud at Customer)

1. Sign in to the Oracle Cloud Infrastructure Object Storage Classic console.
2. Select the container for which you would like to set the custom metadata.
3. Expand **Container Information**.
   The details of the container are displayed.
4. Click **Edit**.
   Look for the **Custom Metadata** field.

5. Click **Add Metadata**.
6. Enter the metadata name and value in the fields under **Add Metadata**.

> **Note:**
>
> Ensure that the custom metadata name and value comply with the input restrictions mentioned in Character Restrictions.

7. Click **Save**.
   The metadata name and value are displayed in the **Container Information** pane.

Setting Custom Metadata Using the REST API

cURL Command Syntax

```
curl -v -X POST \
     -H "X-Auth-Token: token" \
     -H "X-Container-Meta-Name: value" \
     accountURL/containerName
```

- `token` is the authentication token obtained earlier from Oracle Cloud Infrastructure Object Storage Classic. See Authenticating Access When Using the REST API.
- `Name` and `value` are the metadata name and value to be created.

> **Note:**
>
> Ensure that the custom metadata name and value comply with the input restrictions mentioned in Character Restrictions.

- For the syntax of `accountURL`, see About REST URLs for Oracle Cloud Infrastructure Object Storage Classic Resources.
- `containerName` is the name of the container for which custom metadata should be created.

> **Note:**
>
> When you send a REST API request to Oracle Cloud Infrastructure Object Storage Classic, all non-ASCII characters in container names, object names and metadata values must be URL-encoded. For example, `my container` should be encoded as `my%20container`, where `%20` is the HTML encoding for the space character. Similarly, `my Über Container` should be encoded as `my %20%C3%9Cber%20Container`, where `%20` represents the space character and `%C3%9C` is the Ü character.

HTTP Response Codes

- Success: `204 No Content`
- Failure: See Error Code Reference for Oracle Cloud Infrastructure Object Storage Classic

cURL Command Examples

Sample Cloud account with the following details:

- Account name: `acme`
- REST Endpoint URL: `https://acme.storage.oraclecloud.com/v1/Storage-acme`
- REST Endpoint (Permanent) URL: `https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/Storage-7b16fede61e1417ab83eb52e06f0e365`

> **Note:**
>
> The REST Endpoint (Permanent) URL is displayed for the accounts created after November 2017.

- Using the REST Endpoint URL obtained from the **REST Endpoint** field:

```
curl -v -X POST \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     -H "X-Container-Meta-Category: Books" \
     https://acme.storage.oraclecloud.com/v1/Storage-acme/FirstContainer
```

The following is an example of the output of this command:

```
> POST /v1/Storage-acme/FirstContainer HTTP/1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.14.0.0
zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: acme.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b
> X-Container-Meta-Category: Books
>
< HTTP/1.1 204 No Content
< Date: Fri, 06 Mar 2015 11:35:35 GMT
< Content-Length: 0
< Content-Type: text/html; charset=UTF-8
< X-Trans-Id: tx3e77b77de39f4097a5a49-0054f99107
< Cache-Control: no-cache
< Pragma: no-cache
< Content-Language: en
```

- Using the Service Permanent REST Endpoint URL obtained from the **REST Endpoint (Permanent)** field:

> **Note:**
>
> This cURL command example applies to the accounts created after November 2017.

```
curl -v -X POST \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     -H "X-Container-Meta-Category: Books" \
     https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/
Storage-7b16fede61e1417ab83eb52e06f0e365/FirstContainer
```

The following is an example of the output of this command:

```
> POST /v1/Storage-7b16fede61e1417ab83eb52e06f0e365/FirstContainer HTTP/1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.14.0.0
zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b
> X-Container-Meta-Category: Books
>
< HTTP/1.1 204 No Content
< Date: Fri, 06 Mar 2015 11:35:35 GMT
< Content-Length: 0
< Content-Type: text/html; charset=UTF-8
< X-Trans-Id: tx3e77b77de39f4097a5a49-0054f99107
< Cache-Control: no-cache
< Pragma: no-cache
< Content-Language: en
```

For information about getting custom container metadata, see Getting Container Metadata.

For information about deleting custom container metadata, see Deleting Container Metadata.

Enabling CORS for a Container

Cross-Origin Resource Sharing (CORS) allows browser-based programs (like JavaScript) to access resources in another domain. This enables web applications to access Oracle Cloud Infrastructure Object Storage Classic, overcoming the **Same-Origin** policy that's used by browsers to prevent access to resources in other domains.

To enable CORS access for a container, complete the following steps:

1. Specify the origins from which requests are allowed, by setting the `X-Container-Meta-Access-Control-Allow-Origin` metadata header.

   cURL Command Syntax

   ```
   curl -i -XPOST \
        -H "X-Auth-Token: token" \
        -H "X-Container-Meta-Access-Control-Allow-Origin: origins" \
        accountURL/containerName
   ```

   - `token` is the authentication token obtained earlier from Oracle Cloud Infrastructure Object Storage Classic. See Authenticating Access When Using the REST API
   - `origins` is a space-separated list of the origins from which you want to allow CORS requests to the container
   - `containerName` is the name of the container for which custom metadata should be created.

   HTTP Response Codes

   - Success: `204 No Content`
   - Failure: See Error Code Reference for Oracle Cloud Infrastructure Object Storage Classic

Sample Cloud account with the following details:

- Account name: `acme`
- REST Endpoint URL: `https://acme.storage.oraclecloud.com/v1/Storage-acme`
- REST Endpoint (Permanent) URL: `https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/Storage-7b16fede61e1417ab83eb52e06f0e365`

> **Note:**
>
> The REST Endpoint (Permanent) URL is displayed for the accounts created after November 2017.

The REST Endpoint URL for the sample Cloud account is used in all the cURL command examples in this section. To use the REST Endpoint (Permanent) URL, replace `https://acme.storage.oraclecloud.com/v1/Storage-acme` with `https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/Storage-7b16fede61e1417ab83eb52e06f0e365`.

cURL Command Example

This example shows how to allow CORS requests from two origins (`http://acme-admin.example.com` and `http://acme-app.example.com`) to a container named *myContainer*.

```
curl -i -XPOST \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     -H "X-Container-Meta-Access-Control-Allow-Origin: http://acme-admin.example.com/ http://acme-app.example.com/" \
     https://acme.storage.oraclecloud.com/v1/Storage-acme/myContainer
```

> **Note:**
>
> To allow access to the container from **any** domain, set the `X-Container-Meta-Access-Control-Allow-Origin` header value to "*".

The following is an example of the output of this command:

```
< HTTP/1.1 204 No Content
< X-Trans-Id: tx3e77b77de39f4097a5a49-0054f99107
< Date: Fri, 06 Mar 2015 11:35:35 GMT
< Connection: keep-alive
< X-Last-Modified-Timestamp: 1469879583.18732
< Content-Type: text/html; charset=UTF-8
```

2. (Optional) Set the duration that the origins can hold the results of an `OPTIONS` request, by setting the `X-Container-Meta-Access-Control-Max-Age` header. Browsers typically send an `OPTIONS` request first to check whether the origin is allowed to send requests.

cURL Command Syntax

```
curl -i -XPOST \
     -H "X-Auth-Token: token" \
     -H "X-Container-Meta-Access-Control-Max-Age: maxAge" \
     accountURL/containerName
```

- maxAge is the time (in seconds) for which the results of the OPTIONS request must be valid.

HTTP Response Codes

- Success: `204 No Content`
- Failure: See Error Code Reference for Oracle Cloud Infrastructure Object Storage Classic

cURL Command Example

```
curl -i -XPOST \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     -H "X-Container-Meta-Access-Control-Max-Age: 10000" \
     https://acme.storage.oraclecloud.com/v1/Storage-acme/myContainer
```

The following is an example of the output of this command:

```
< HTTP/1.1 204 No Content
< X-Trans-Id: txd0af08f8298140c599348-00579c95bdga
< Date: Fri, 06 Mar 2015 11:35:36 GMT
< Connection: keep-alive
< X-Last-Modified-Timestamp: 1469879741.41535
< Content-Type: text/html; charset=UTF-8
```

3. (Optional) Specify the response headers that the service should return in response to CORS requests to this container, by setting the `X-Container-Meta-Access-Control-Expose-Headers` metadata header. If this header is not set explicitly, the response to CORS requests will return the container's standard metadata along with any CORS metadata headers that're set for the container. To view the list of container's standard metadata, see Getting Container Metadata.

cURL Command Syntax

```
curl -i -XPOST \
     -H "X-Auth-Token: token" \
     -H "X-Container-Meta-Access-Control-Expose-Headers: headers" \
     accountURL/containerName
```

- `headers` is a space-separated list of the headers that the service must return in response to CORS requests to the container.

HTTP Response Codes

- Success: `204 No Content`
- Failure: See Error Code Reference for Oracle Cloud Infrastructure Object Storage Classic

cURL Command Example

This example shows how to set up the service to return two headers (`X-Container-Object-Count` and `X-Container-Bytes-Used`) in response to CORS requests to the container named *myContainer*.

```
curl -i -XPOST \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     -H "X-Container-Meta-Access-Control-Expose-Headers: X-Container-Object-
Count X-Container-Bytes-Used" \
     https://acme.storage.oraclecloud.com/v1/Storage-acme/myContainer
```

The following is an example of the output of this command:

```
< HTTP/1.1 204 No Content
< X-Trans-Id: tx3e708ed6834d4ba4bf1cd-00579c95bdga
< Date: Fri, 06 Mar 2015 11:35:37 GMT
< Connection: keep-alive
< X-Last-Modified-Timestamp: 1469879793.38545
< Content-Type: text/html; charset=UTF-8
```

4. To verify CORS access, do the following:

   a. Send an `OPTIONS` request to find out whether requests from your origin are allowed and to get a list of the allowed methods.

      cURL Command Syntax

      ```
      curl -i -XOPTIONS \
           -H "X-Auth-Token: token" \
           -H "Origin: myURL" \
           -H "Access-Control-Request-Method: method" \
           accountURL/containerName/objectName
      ```

      - `myURL` is the origin of the request.
      - `method` is type of request you want to make, such as `HEAD`, `GET`, `PUT`, `POST` or `DELETE`.
      - `objectName` is the name of the object you want to access or update.

      HTTP Response Codes

      - Success: `200 OK`
      - Failure: See Error Code Reference for Oracle Cloud Infrastructure Object Storage Classic

      cURL Command Example

      This example shows how to send `OPTIONS` request from the origin `http://www.example.com` for the `GET` method to an object named *myObject* in the container named *myContainer*.

      ```
      curl -i -XOPTIONS \
           -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
           -H "Origin: http://www.example.com" \
           -H "Access-Control-Request-Method: GET" \
           https://acme.storage.oraclecloud.com/v1/Storage-acme/myContainer/
      myObject
      ```

      The following is an example of the output of this command:

      ```
      < HTTP/1.1 200 OK
      < Allow: HEAD, GET, PUT, POST, OPTIONS, DELETE
      < Content-Length: 0
      < Date: Fri, 06 Mar 2015 11:35:38 GMT
      ```

   b. Send the actual request such as `HEAD`, `GET`, `PUT`, `POST` or `DELETE`.

cURL Command Syntax

```
curl -i -Xmethod \
     -H "X-Auth-Token: token" \
     -H "Origin: myURL" \
     accountURL/containerName/objectName
```

- `method` is the actual request such as `HEAD`, `GET`, `PUT`, `POST` or `DELETE`.
- `objectName` is the name of the object which has to be accessed or updated.

HTTP Response Codes

- Success:
  - For `HEAD`, `POST` and `DELETE`: `204 No Content`
  - For `GET`: `200 OK`
  - For `PUT`: `201 Created`
- Failure: See Error Code Reference for Oracle Cloud Infrastructure Object Storage Classic

cURL Command Example

This example shows how to send `GET` request from the origin `http://www.example.com` to an object named *myObject* in the container named *myContainer*.

```
curl -i -XGET \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     -H "Origin: http://www.example.com" \
     https://acme.storage.oraclecloud.com/v1/Storage-acme/myContainer/
myObject
```

The following is an example of the output of this command:

```
< HTTP/1.1 200 OK
< Accept-Ranges: bytes
< Last-Modified: Fri, 06 Mar 2015 10:35:00 GMT
< Etag: d41d8cd98f00b204e9800998ecf8427e
< X-Timestamp: 1469879406.30001
< X-Trans-Id: tx2a649bebc5d64bdfa8cc6-00579c96daga
< Date: Fri, 06 Mar 2015 11:35:38 GMT
< Connection: keep-alive
< X-Last-Modified-Timestamp: 1469879406.30001
< Content-Type: application/octet-stream;charset=UTF-8
< Content-Length: 0
```

Making Objects in a Container Immutable

You can make the objects in your container immutable by setting the Write-Once-Read-Many (WORM) policy for your container to prevent the users from modifying and deleting the objects in the container for a specified duration. The container-level WORM policy applies to all the objects that're uploaded to the container, unless an object has it's own object-level WORM policy set during upload.

Once you've set a container's WORM policy, you cannot change it. When the WORM policy expires, you can delete the objects in the container but can't modify the objects

or object metadata. To set the WORM policy for a specific object, see Making an Object Immutable.

You must have the `Service Administrator` role to set a container's WORM policy header `X-Worm-Expiration-Days`.

For information about using the REST API to set the container metadata, see Create, Update, or Delete Container Metadata in *REST API for Standard Storage in Oracle Cloud Infrastructure Object Storage Classic*.

cURL Command Syntax

To set the WORM policy for an empty container:

```
curl -v -X POST \
     -H "X-Auth-Token: token" \
     -H "X-Worm-Expiration-Days: period" \
     accountURL/containerName
```

To set the WORM policy while creating a container:

```
curl -v -X PUT \
     -H "X-Auth-Token: token" \
     -H "X-Worm-Expiration-Days: period" \
     accountURL/containerName
```

- `token` is the authentication token obtained earlier from Oracle Cloud Infrastructure Object Storage Classic. See Authenticating Access When Using the REST API.
- `period` is the duration, in days, for which the WORM policy is set for the container.
- For the syntax of `accountURL`, see About REST URLs for Oracle Cloud Infrastructure Object Storage Classic Resources.
- `containerName` is the name of the container for which the WORM policy must be set.

> **Note:**
>
> When you send a REST API request to Oracle Cloud Infrastructure Object Storage Classic, all non-ASCII characters in container names, object names and metadata values must be URL-encoded. For example, `my container` should be encoded as `my%20container`, where `%20` is the HTML encoding for the space character. Similarly, `my Über Container` should be encoded as `my %20%C3%9Cber%20Container`, where `%20` represents the space character and `%C3%9C` is the `Ü` character.

HTTP Response Codes

- Success:
  - When updating a container with the WORM policy: `204 No Content`
  - When creating a container with the WORM policy: `201 Created`
- Failure: See Error Code Reference for Object Storage Classic

For information about getting container metadata, see Getting Container Metadata.

cURL Command Examples

Sample Cloud account with the following details:

- Account name: `acme`
- REST Endpoint URL: `https://acme.storage.oraclecloud.com/v1/Storage-acme`
- REST Endpoint (Permanent) URL: `https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/Storage-7b16fede61e1417ab83eb52e06f0e365`

> **Note:**
>
> The REST Endpoint (Permanent) URL is displayed for the accounts created after November 2017.

The REST Endpoint URL for the sample Cloud account is used in all the cURL command examples in this section. To use the REST Endpoint (Permanent) URL, replace `https://acme.storage.oraclecloud.com/v1/Storage-acme` with `https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/Storage-7b16fede61e1417ab83eb52e06f0e365`.

- This command sets the WORM policy of `4` days for the container `FirstContainer`:

```
curl -v -X POST \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     -H "X-Worm-Expiration-Days: 4" \
     https://acme.storage.oraclecloud.com/v1/Storage-acme/FirstContainer
```

  The following is an example of the output of this command:

```
> POST /v1/Storage-acme/FirstContainer HTTP/1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.14.0.0
zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: acme.storage.oraclecloud.com
> Accept: */*
> X-Worm-Expiration-Days: 4
> X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b
>
< HTTP/1.1 204 No Content
< Date: Tue, 06 Dec 2016 11:32:19 GMT
< Content-Type: text/html; charset=UTF-8
< X-Trans-Id: txe8869b3edea348e5b49eb-0054f99043
< Cache-Control: no-cache
< Pragma: no-cache
< Content-Language: en
```

- This command creates the container `SecondContainer` and sets the WORM policy of `2` days:

```
curl -v -X PUT \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     -H "X-Worm-Expiration-Days: 2" \
     https://acme.storage.oraclecloud.com/v1/Storage-acme/SecondContainer
```

  The following is an example of the output of this command:

```
> PUT /v1/Storage-acme/SecondContainer HTTP/1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.14.0.0
zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: acme.storage.oraclecloud.com
> Accept: */*
> X-Worm-Expiration-Days: 2
> X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b
>
< HTTP/1.1 201 Created
< Date: Tue, 06 Dec 2016 11:36:24 GMT
< Content-Type: text/html; charset=UTF-8
< X-Trans-Id: txe8869b3edea348e5b49eb-0054f99078
< Cache-Control: no-cache
< Pragma: no-cache
< Content-Language: en
```

- This command gets the header values of the object `ObjectA` in the container `FirstContainer` that already has a WORM policy of `4` days set:

  ```
  curl -v -s -X HEAD \
       -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
       https://acme.storage.oraclecloud.com/v1/Storage-acme/FirstContainer/ObjectA
  ```

  In the following example of the output of this command, the object header `X-Worm-Expiration-At` is indicating the Epoch expiration time of the WORM policy that is set on the container:

  ```
  > HEAD /v1/Storage-acme/FirstContainer/ObjectA HTTP/1.1
  > User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.14.0.0
  zlib/1.2.3 libidn/1.18 libssh2/1.4.2
  > Host: acme.storage.oraclecloud.com
  > Accept: */*
  >
  > X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b
  >
  < HTTP/1.1 200 OK
  < Date: Tue, 06 Dec 2016 11:37:09 GMT
  < X-Worm-Expiration-At: 1481366700
  < Content-Type: text/html; charset=UTF-8
  < X-Timestamp: 1481024229
  < X-Trans-Id: txe8869b3edea348e5b49eb-005417894
  < Cache-Control: no-cache
  < Pragma: no-cache
  < Content-Language: en
  ```

  Here, the Epoch expiration time `1481366700` corresponds to the date `Sat, 10 Dec 2016 11:45:00 GMT` which is 4 days from the date `Tue, 06 Dec 2016 11:45:00 GMT` when the object was uploaded. See http://epochconverter.com.

3

Managing Objects in Object Storage Classic

This section provides documentation about managing objects in Object Storage Classic.

Topics:

- Typical Workflow for Managing Objects
- Roles Required for Managing Objects in Object Storage Classic
- Listing Objects in a Container
- Creating Objects
- Getting Object Metadata
- Finding Out the Status of Objects in an Archive Container
- Restoring Archived Objects
- Tracking Restoration of an Object in an Archive Container
- Downloading Objects
- Deleting Objects
- Updating Object Metadata
- Copying Objects
- Encrypting Objects

Typical Workflow for Managing Objects

Use the following table as a guide to the workflow for managing objects. If you have not yet subscribed to or set up your service, see Workflow for Getting Started with Oracle Cloud Infrastructure Object Storage Classic.

Task	Description	More Information
Access the service	Access the service through the REST API. To know more about accessing the service using other interfaces, see Interfaces to Object Storage Classic.	Accessing Object Storage Classic
Generate an authentication token	Generate an authentication token if you are accessing the service through the REST API interface. If you are accessing the service through Java library, the authentication token would be automatically generated and applied.	Authenticating Access When Using the REST API
List containers	List containers in an account.	Listing Containers
List objects within a container	List objects within a specified container.	Listing Objects in a Container

Task	Description	More Information
Create objects	Create a single object, bulk create objects and upload files larger than 5GB.	Creating a Single Object
Get object metadata	Retrieve information about an object in a container.	Getting Object Metadata
Restore archived objects	To download an archived object, the object must first be restored.	Restoring Archived Objects
Download objects	Download an object's metadata and data.	Downloading Objects
Delete objects	Delete and bulk delete objects.	Deleting Objects
Update object metadata	Update custom metadata and special metadata.	Updating Object Metadata
Copy objects	Copy an object to another object.	Copying Objects
Encrypt objects	Transparently encrypt objects with the Java library before uploading.	Encrypting Objects

Roles Required for Managing Objects in Object Storage Classic

Users with the Storage Administrator role can create, read, update, and delete all containers and objects for the service instance.

- Tasks that cause changes to containers and objects require the `Storage_ReadWriteGroup` role or a custom role in the associated container's `X-Container-Write` Access Control List (ACL).
- Tasks that do not cause changes to containers and objects require the `Storage_ReadOnlyGroup` role or a custom role in the associated container's `X-Container-Read` ACL.

For more information about roles and ACLs, see About Oracle Cloud Infrastructure Object Storage Classic Roles and Users.

Listing Objects in a Container

All objects within a container can be listed.

Any user with the Service Administrator role or a role that is specified in the `X-Container-Read` ACL of the container can perform this task.

You can list objects in a container by using the following interfaces:

Interface	Resources
Web Console (Not available on Oracle Cloud at Customer)	See Listing Objects in a Container Using the Web Console.
RESTful API	See Show container details and List Objects in *REST API for Standard Storage in Oracle Cloud Infrastructure Object Storage Classic*.

Interface	Resources
Java Library	See `listObjects` in *Java API Reference for Oracle Cloud Infrastructure Object Storage Classic*.
File Transfer Manager API	See `listObjects` in *Java API Reference for Oracle Cloud Infrastructure Object Storage Classic File Transfer Manager*.
File Transfer Manager CLI	See Listing Objects in *Command-Line Reference for Oracle Cloud Infrastructure Object Storage Classic*.

Listing Objects in a Container Using the Web Console

(Not available on Oracle Cloud at Customer)

1. Sign in to the Oracle Cloud Infrastructure Object Storage Classic console. The list of containers is displayed.
2. Select the container for which you want to list objects. The objects in the container are listed.

Listing Objects in a Container Using the REST API

Objects are sorted by their names lexicographically, using `memcmp()`. All objects, up to 10000 by default, will be returned in the list, unless you filter the list by using any of the following parameters:

- `limit`: Limit the number of objects listed to the specified value. The default and maximum value is 10000.
- `marker`: Return objects with names greater than the specified string.
- `end_marker`: Return objects with names less than the specified string.
- `format`: Return extended information about each returned object in either `xml` or `json` format (REST API only).
- `prefix`: Return objects with names that start with the specified string.
- `delimiter`: Return objects with names that include the specified character. Only the substring of object names before the specified character are returned. Only unique substrings are returned.
  - If the `prefix` parameter is also used, any matches of the specified delimiter character are ignored.
  - Used to emulate directory structures within a container (that is, with a forward slash (/) as the delimiter).

cURL Command Syntax

```
curl -v -X GET \
     -H "X-Auth-Token: token" \
     accountURL/containerName[?query_parameter=value]
```

- `token` is the authentication token obtained earlier from Oracle Cloud Infrastructure Object Storage Classic. See Authenticating Access When Using the REST API.
- For the syntax of `accountURL`, see About REST URLs for Oracle Cloud Infrastructure Object Storage Classic Resources.
- `containerName` is the name of the container for which objects should be listed.

- `query_parameter=value` is the optional filtering parameter.

> **Note:**
>
> When you send a REST API request to Oracle Cloud Infrastructure Object Storage Classic, all non-ASCII characters in container names, object names and metadata values must be URL-encoded. For example, `my container` should be encoded as `my%20container`, where `%20` is the HTML encoding for the space character. Similarly, `my Über Container` should be encoded as `my %20%C3%9Cber%20Container`, where `%20` represents the space character and `%C3%9C` is the Ü character.

HTTP Response Codes

- Success: `200 OK`

> **Note:**
>
> If there are no objects, the HTTP response code would be `204 No Content`.

- Failure: See Error Code Reference for Oracle Cloud Infrastructure Object Storage Classic

cURL Command Examples

Sample Cloud account with the following details:

- Account name: `acme`
- REST Endpoint URL: `https://acme.storage.oraclecloud.com/v1/Storage-acme`
- REST Endpoint (Permanent) URL: `https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/Storage-7b16fede61e1417ab83eb52e06f0e365`

> **Note:**
>
> The REST Endpoint (Permanent) URL is displayed for the accounts created after November 2017.

- Using the REST Endpoint URL obtained from the **REST Endpoint** field:

```
curl -v -X GET \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     https://acme.storage.oraclecloud.com/v1/Storage-acme/myContainer?limit=15
```

> **Note:**
>
> The URL of the account in this example is `https://acme.storage.oraclecloud.com/v1/Storage-acme`. Replace this URL with the URL for your account. For the steps to find out your account URL, see About REST URLs for Oracle Cloud Infrastructure Object Storage Classic Resources.

The following is an example of the output of this command:

```
> GET /v1/Storage-acme/FirstContainer HTTP/1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.14.0.0
zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: acme.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tk4af5123f84d2e3ffb9e77ba657ac8edf
>
< HTTP/1.1 200 OK
< Date: Mon, 09 Mar 2015 11:15:50 GMT
< Content-Length: 63
< X-Container-Object-Count: 4
< X-Container-Write: myIdentityDomainID.Storage.Storage_ReadWriteGroup
< Accept-Ranges: bytes
< X-Timestamp: 1425033529.95392
< X-Container-Read:
myIdentityDomainID.Storage.Storage_ReadOnlyGroup,myIdentityDomainID.Storage.Stora
ge_ReadWriteGroup
< X-Container-Bytes-Used: 92095
< Content-Type: text/plain; charset=utf-8
< X-Trans-Id: tx23ba568df8864b45bc443-0054fd80e6
< Cache-Control: no-cache
< Pragma: no-cache
< Content-Language: en
<
Backup-2-0_24680
Backup-3-0_32872
MetadataLog-0_32872
test.key
```

- Using the Service Permanent REST Endpoint URL obtained from the **REST Endpoint (Permanent)** field:

> **Note:**
>
> This cURL command example applies to the accounts created after November 2017.

```
curl -v -X GET \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/
Storage-7b16fede61e1417ab83eb52e06f0e365/myContainer?limit=15
```

> **Note:**
>
> The URL of the account in this example is `https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/Storage-7b16fede61e1417ab83eb52e06f0e365`. Replace this URL with the URL for your account. For the steps to find out your account URL, see About REST URLs for Oracle Cloud Infrastructure Object Storage Classic Resources.

The following is an example of the output of this command:

```
> GET /v1/Storage-7b16fede61e1417ab83eb52e06f0e365/FirstContainer HTTP/1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.14.0.0
zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tk4af5123f84d2e3ffb9e77ba657ac8edf
>
< HTTP/1.1 200 OK
< Date: Mon, 09 Mar 2015 11:15:50 GMT
< Content-Length: 63
< X-Container-Object-Count: 4
< X-Container-Write: myIdentityDomainID.Storage.Storage_ReadWriteGroup
< Accept-Ranges: bytes
< X-Timestamp: 1425033529.95392
< X-Container-Read:
myIdentityDomainID.Storage.Storage_ReadOnlyGroup,myIdentityDomainID.Storage.Stora
ge_ReadWriteGroup
< X-Container-Bytes-Used: 92095
< Content-Type: text/plain; charset=utf-8
< X-Trans-Id: tx23ba568df8864b45bc443-0054fd80e6
< Cache-Control: no-cache
< Pragma: no-cache
< Content-Language: en
<
Backup-2-0_24680
Backup-3-0_32872
MetadataLog-0_32872
test.key
```

Creating Objects

Objects must be created within a container. You can create a single object or upload multiple objects to a container.

Topics:

- Creating a Single Object
- Uploading Multiple Objects in a Single Operation
- Uploading Large Objects

Creating a Single Object

Objects must be created within a container. Objects can be created by uploading files and/or specifying metadata.

Any user with the Service Administrator role or a role that is specified in the `X-Container-Write` ACL of the container can perform this task. You can create objects in a `standard` or `archive` container.

You can create objects in a container by using the following interfaces:

Interface	Resources
Web Console (Not available on Oracle Cloud at Customer)	See Creating a Single Object Using the Web Console.
RESTful API	See • Creating a Single Object Using the REST API • Create or Replace Object in *REST API for Standard Storage in Oracle Cloud Infrastructure Object Storage Classic*.
Java Library	See `storeObject` in *Java API Reference for Oracle Cloud Infrastructure Object Storage Classic*.
File Transfer Manager API	See upload in *Java API Reference for Oracle Cloud Infrastructure Object Storage Classic File Transfer Manager*.
File Transfer Manager CLI	See Uploading an Object in *Command-Line Reference for Oracle Cloud Infrastructure Object Storage Classic*.

For the steps to upload multiple objects in a single operation, see Uploading Multiple Objects in a Single Operation.

Creating a Single Object Using the Web Console

(Not available on Oracle Cloud at Customer)

1. Sign in to the Oracle Cloud Infrastructure Object Storage Classic console.
2. Select the container in which you would like to upload an object.
3. Select **Enable Upload and Download** in the **Container Information** pane.
   The following message appears:

   ```
   In order to upload and download objects to the current container, this
   application must enable Cross-Origin Resource Sharing (CORS) for this
   domain domainName.
   You can disable this setting at anytime.
   Would you like to continue?
   ```

4. Click **OK** to enable Cross-Origin Resource Sharing (CORS).
   The **Upload Objects** button is enabled.

> **Note:**
>
> You can disable CORS in the container at any time. Select **Disable Upload and Download** in the **Container Information** pane to disable CORS. You can't upload objects to the container if CORS is disabled.

5. Click **Upload Objects** and select the object to be uploaded.

> **Note:**
>
> Ensure that the object name complies with the input restrictions mentioned in Character Restrictions.

The upload progress and upload status of the object is displayed.

After the object is uploaded, the object details (**Last Modified** and **Size**) are displayed.

Creating a Single Object Using the REST API

cURL Command Syntax

```
curl -v -X PUT \
     -H "X-Auth-Token: token" \
     -T file \
     accountURL/containerName/objectName
```

- `token` is the authentication token obtained earlier from Oracle Cloud Infrastructure Object Storage Classic. See Authenticating Access When Using the REST API.
- `file` is the full path and name of the file to be uploaded.
- For the syntax of `accountURL`, see About REST URLs for Oracle Cloud Infrastructure Object Storage Classic Resources.
- `containerName` is the name of the container in which the object should be created.
- `objectName` is the name of the object to be created.

> **Note:**
>
> Ensure that the object name complies with the input restrictions mentioned in Character Restrictions.

> **Note:**
>
> When you send a REST API request to Oracle Cloud Infrastructure Object Storage Classic, all non-ASCII characters in container names, object names and metadata values must be URL-encoded. For example, `my container` should be encoded as `my%20container`, where `%20` is the HTML encoding for the space character. Similarly, `my Über Container` should be encoded as `my%20%C3%9Cber%20Container`, where `%20` represents the space character and `%C3%9C` is the Ü character.

To schedule automatic deletion of objects, include the `X-Delete-After` or `X-Delete-At` header. See Scheduling Automatic Deletion of Objects.

HTTP Response Codes

- Success: `201 Created`
- Failure: See Error Code Reference for Oracle Cloud Infrastructure Object Storage Classic

cURL Command Examples

Sample Cloud account with the following details:

- Account name: `acme`
- REST Endpoint URL: `https://acme.storage.oraclecloud.com/v1/Storage-acme`
- REST Endpoint (Permanent) URL: `https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/Storage-7b16fede61e1417ab83eb52e06f0e365`

> **Note:**
>
> The REST Endpoint (Permanent) URL is displayed for the accounts created after November 2017.

- Using the REST Endpoint URL obtained from the **REST Endpoint** field:

```
curl -v -X PUT \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     -T myFile.txt \
     https://acme.storage.oraclecloud.com/v1/Storage-acme/FirstContainer/myObject
```

The following is an example of the output of this command:

```
> PUT /v1/Storage-acme/FirstContainer/myObject HTTP/1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.14.0.0
zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: acme.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b
> Content-Length: 23
> Expect: 100-continue
>
* Done waiting for 100-continue
< HTTP/1.1 201 Created
```

```
< Date: Mon, 09 Mar 2015 11:26:57 GMT
< Last-Modified: Mon, 09 Mar 2015 11:26:58 GMT
< Content-Length: 0
< Etag: 846fa9d298be05e5f598703f0c3d6f51
< Content-Type: text/html; charset=UTF-8
< X-Trans-Id: tx2a97f34acb7048679ae3b-0054fd8381
< Cache-Control: no-cache
< Pragma: no-cache
< Content-Language: en
```

- Using the Service Permanent REST Endpoint URL obtained from the **REST Endpoint (Permanent)** field:

> **Note:**
>
> This cURL command example applies to the accounts created after November 2017.

```
curl -v -X PUT \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     -T myFile.txt \
     https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/
Storage-7b16fede61e1417ab83eb52e06f0e365/FirstContainer/myObject
```

The following is an example of the output of this command:

```
> PUT /v1/Storage-7b16fede61e1417ab83eb52e06f0e365/FirstContainer/myObject HTTP/
1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.14.0.0
zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b
> Content-Length: 23
> Expect: 100-continue
>
* Done waiting for 100-continue
< HTTP/1.1 201 Created
< Date: Mon, 09 Mar 2015 11:26:57 GMT
< Last-Modified: Mon, 09 Mar 2015 11:26:58 GMT
< Content-Length: 0
< Etag: 846fa9d298be05e5f598703f0c3d6f51
< Content-Type: text/html; charset=UTF-8
< X-Trans-Id: tx2a97f34acb7048679ae3b-0054fd8381
< Cache-Control: no-cache
< Pragma: no-cache
< Content-Language: en
```

Uploading Multiple Objects in a Single Operation

You can create multiple objects in a single operation by uploading an archive file containing multiple files and directories.

> **Note:**
>
> The term **archive** in this topic refers to files in the `.tar`, `.tar.gz`, or `tar.bz2` format. It does not refer to the `Archive` storage class. Bulk-creating objects in an `Archive` container is not supported.

Any user with the Service Administrator role or a role that is specified in the `X-Container-Write` ACL of the container can perform this task.

To bulk-create objects:

1. Create a local archive (`.tar`, `.tar.gz`, or `tar.bz2`) of the files and directories to be stored.
   - The top-level directory will be stored as a container; nested directories will be represented in object names.
   - Files will be stored as objects.

   Example archive file contents:

   ```
   $ tar -tzf myfiles.tar.gz

   bulktest-files/
   bulktest-files/file1.txt
   bulktest-files/file2.txt
   bulktest-files/file3.txt
   ```

2. Upload the archive.
   - Upload the file to the account URL. Specify the archive type by using the `?extract-archive` request parameter.
   - The response indicates the number of files created and any errors, if any.

   cURL Command Examples

   Sample Cloud account with the following details:

   - Account name: `acme`
   - REST Endpoint URL: `https://acme.storage.oraclecloud.com/v1/Storage-acme`
   - REST Endpoint (Permanent) URL: `https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/Storage-7b16fede61e1417ab83eb52e06f0e365`

> **Note:**
>
> The REST Endpoint (Permanent) URL is displayed for the accounts created after November 2017.

- Using the REST Endpoint URL obtained from the **REST Endpoint** field:

```
$ curl -v -X PUT
       -H "X-Auth-Token: AUTH_tkcc14bdc52d571f70991b68907098f831" \
       "https://acme.storage.oraclecloud.com/v1/Storage-acme/?extract-
archive=tar.gz" \
       --data-binary @myfiles.tar.gz

> PUT /v1/Storage-acme/?extract-archive=tar.gz HTTP/1.1
> Host: acme.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tkcc14bdc52d571f70991b68907098f831
> Content-Length: 214
> Content-Type: application/x-www-form-urlencoded

< HTTP/1.1 200 OK
< Date: Thu, 12 Dec 2013 14:56:18 GMT
< Content-Type: text/plain
< Transfer-Encoding: chunked

Number Files Created: 3
Response Body:
Response Status: 201 Created
```

> **Note:**
>
> For the result of the operation, look at the response body (not the returned status code).

- Using the Service Permanent REST Endpoint URL obtained from the **REST Endpoint (Permanent)** field:
  This cURL command example applies to the accounts created after November 2017.

```
$ curl -v -X PUT
       -H "X-Auth-Token: AUTH_tkcc14bdc52d571f70991b68907098f831" \
       "https://
storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/
Storage-7b16fede61e1417ab83eb52e06f0e365/?extract-archive=tar.gz" \
       --data-binary @myfiles.tar.gz

> PUT /v1/Storage-7b16fede61e1417ab83eb52e06f0e365/?extract-archive=tar.gz
HTTP/1.1
> Host: storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tkcc14bdc52d571f70991b68907098f831
> Content-Length: 214
> Content-Type: application/x-www-form-urlencoded

< HTTP/1.1 200 OK
< Date: Thu, 12 Dec 2013 14:56:18 GMT
< Content-Type: text/plain
```

```
< Transfer-Encoding: chunked

Number Files Created: 3
Response Body:
Response Status: 201 Created
```

> **Note:**
>
> For the result of the operation, look at the response body (not the returned status code).

You can create multiple objects in a single operation by using the following different interfaces:

Interface	Resources
RESTful API	See Bulk Operations in *OpenStack Object Storage Service API Reference*.
File Transfer Manager API	See `uploadMultipleFiles` in *Java API Reference for Oracle Cloud Infrastructure Object Storage Classic File Transfer Manager*.
File Transfer Manager CLI	See Uploading an Object in *Command-Line Reference for Oracle Cloud Infrastructure Object Storage Classic*.

Uploading Large Objects

A single object can hold up to 5 GB of data, but multiple objects can be linked together to hold more than 5 GB of contiguous data. You can create small objects as segments and upload them as one large object by using a manifest object.

> **Note:**
>
> A large object can have a maximum of 2048 segments. Each segment can be up to 5 GB. So the maximum size of a file that you can upload to Oracle Cloud Infrastructure Object Storage Classic as a large object is 10 TB.

Any user with the Service Administrator role or a role that is specified in the `X-Container-Write` ACL of the container can perform this task.

You can upload large objects in a container by using the following interfaces:

Interface	Resources
Web Console (Not available on Oracle Cloud at Customer)	See Uploading a Large Object Using the Web Console.
RESTful API	• See Uploading a Large Object Using the REST API. • See Create or Replace Object in *REST API for Standard Storage in Oracle Cloud Infrastructure Object Storage Classic*.
File Transfer Manager API	See upload in *Java API Reference for Oracle Cloud Infrastructure Object Storage Classic File Transfer Manager*.

Interface	Resources
File Transfer Manager CLI	See Uploading Files in *Command-Line Reference for Oracle Cloud Infrastructure Object Storage Classic*.

Uploading a Large Object Using the Web Console

(Not available on Oracle Cloud at Customer)

1. Sign in to the Oracle Cloud Infrastructure Object Storage Classic console.
2. Select the container in which you would like to upload a large object.
3. Select **Enable Upload and Download** in the **Container Information** pane.
   The following message appears:

   ```
   In order to upload and download objects to the current container, this
   application must enable Cross-Origin Resource Sharing (CORS) for this
   domain domainName.
   You can disable this setting at anytime.
   Would you like to continue?
   ```

4. Click **OK** to enable Cross-Origin Resource Sharing (CORS).
   The **Large Object Upload** button is enabled.

   > **Note:**
   >
   > You can disable CORS in the container at any time. Select **Disable Upload and Download** in the **Container Information** pane to disable CORS. You can't upload objects to the container if CORS is disabled.

5. Click **Large Object Upload** and select the large object to be uploaded.
   The upload progress of all the segment objects is displayed.

   After the uploads are complete, the details (**Last Modified** and **Size**) are displayed for the manifest object and segment objects.

Uploading a Large Object Using the REST API

To upload a large object:

1. Segment the large file locally into multiple sequential segment files, each smaller than 5 GB.
   On Linux, for example, you can use the following command:

   ```
   split -b 10m file_name segment_name
   ```

2. List all the segment files.
   ```
   ls –al segment_name*
   ```

3. Create objects from each segment file. Upload all the objects in the same container.

   ```
   curl -v -X PUT \
           -H "X-Auth-Token:token" \
           -T segmentName \
           accountURL/containerName/objectName
   ```

- `token` is the authentication token obtained earlier from Oracle Cloud Infrastructure Object Storage Classic. See Authenticating Access When Using the REST API.
- `segmentName` is the full path and name of the segment file to be uploaded.
- For the syntax of `accountURL`, see About REST URLs for Oracle Cloud Infrastructure Object Storage Classic Resources.
- `containerName` is the name of the container in which the object should be created.
- `objectName` is the name of the object to be created. Ensure that you name the object with the corresponding segment file name.

4. Create a manifest file in JSON format, where each element contains the following attributes:

   - `path`: The container and object name in the format: *`containerName/segmentObjectName`*
   - `etag`: MD5 checksum of the segment object.
     You can find the value from the `Etag` header of the segment object.
   - `size_bytes`: Size of the segment object.
     You can find the value from the `Content-Length` header of the segment object.

   Ensure that the manifest file contains these three attributes for each segment object.

   Sample manifest file:

```
[
    {
        "path": "FirstContainer/segment_aa",
        "etag": "f1c9645dbc14efddc7d8a322685f26eb",
        "size_bytes": 10485760
    },
    {
        "path": "FirstContainer/segment_ab",
        "etag": "f1c9645dbc14efddc7d8a322685f26eb",
        "size_bytes": 10485760
    },
    {
        "path": "FirstContainer/segment_ac",
        "etag": "f1c9645dbc14efddc7d8a322685f26eb",
        "size_bytes": 10485760
    },
    {
        "path": "FirstContainer/segment_ad",
        "etag": "f1c9645dbc14efddc7d8a322685f26eb",
        "size_bytes": 10485760
    },

    ...
    {
        "path": "FirstContainer/segment_aj",
        "etag": "f1c9645dbc14efddc7d8a322685f26eb",
        "size_bytes": 10485760
    }
]
```

5. Upload the manifest file that you just created. In the URI, include the `?multipart-manifest=put` query parameter.

```
curl -v -X PUT \
        -H "X-Auth-Token:token" \
        "accountURL/containerName/LargeFileName?multipart-manifest=put" \
        -T ./fileName.json
```

   - `LargeFileName` is the name of the large object
   - `fileName.json` is the name of the manifest file
   - `?multipart-manifest=put` is the query parameter to upload the manifest file

6. Check the size of the large object.

```
curl -v -X HEAD \
        -H "X-Auth-Token:token" \
        accountURL/containerName/LargeObjectName
```

   The size of the large object is the total size of all the segment objects. To download a large object, see Downloading Large Objects.

Example:

Sample Cloud account with the following details:

- Account name: `acme`
- REST Endpoint URL: `https://acme.storage.oraclecloud.com/v1/Storage-acme`
- REST Endpoint (Permanent) URL: `https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/Storage-7b16fede61e1417ab83eb52e06f0e365`

> **Note:**
>
> The REST Endpoint (Permanent) URL is displayed for the accounts created after November 2017.

The REST Endpoint URL for the sample Cloud account is used in the steps below. To use the REST Endpoint (Permanent) URL, replace `https://acme.storage.oraclecloud.com/v1/Storage-acme` with `https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/Storage-7b16fede61e1417ab83eb52e06f0e365`.

1. Segment the large file locally into multiple sequential segment files, each smaller than 5 GB:
   `split -b 10m myLargeFile.zip segment_`
2. List all the segment files:
   `ls –al segment_*`

```
segment_aa
segment_ab
segment_ac
segment_ad
segment_ae
segment_af
segment_ag
```

```
segment_ah
segment_ai
segment_aj
```

3. Create objects from each segment file (*segment_aa*, *segment_ab*...*segment_aj*), preserving the segment file names. Upload all the objects in the same container `FirstContainer`. Here's an example for one of the segment files:

```
curl -v -X PUT \
         -H "X-Auth-Token: AUTH_tk5a58b7a8c34bb7b662523a59a5272650" \
         -T segment_aa
         https://acme.storage.oraclecloud.com/v1/Storage-acme/FirstContainer/
segment_aa
```

The following is the output of this command:

```
> User-Agent: curl/7.29.0
> Host: acme.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tkc9305a46ebaa0585c4c7ae063c844f0b
> Content-Length: 10485760
> Expect: 100-continue
>
< HTTP/1.1 100 Continue
* We are completely uploaded and fine
< HTTP/1.1 201 Created
< Date: Tue, 15 Dec 2015 10:18:26 GMT
< Last-Modified: Tue, 15 Dec 2015 10:17:21 GMT
< X-Trans-Id: tx85da332ec5ae4852b7d8c-00566fe8b0ga
< Etag: f1c9645dbc14efddc7d8a322685f26eb
< Connection: keep-alive
< X-Last-Modified-Timestamp: 1450174640.10123
< Content-Type: text/html;charset=UTF-8
< Content-Length: 0
<
```

4. Create a manifest file in JSON format.
Sample manifest file:

```
[
    {
        "path": "FirstContainer/segment_aa",
        "etag": "f1c9645dbc14efddc7d8a322685f26eb",
        "size_bytes": 10485760
    },
    {
        "path": "FirstContainer/segment_ab",
        "etag": "f1c9645dbc14efddc7d8a322685f26eb",
        "size_bytes": 10485760
    },
    {
        "path": "FirstContainer/segment_ac",
        "etag": "f1c9645dbc14efddc7d8a322685f26eb",
        "size_bytes": 10485760
    },
    {
        "path": "FirstContainer/segment_ad",
        "etag": "f1c9645dbc14efddc7d8a322685f26eb",
        "size_bytes": 10485760
    },

    ...
```

```
    {
        "path": "FirstContainer/segment_aj",
        "etag": "f1c9645dbc14efddc7d8a322685f26eb",
        "size_bytes": 10485760
    }
]
```

5. Upload the manifest file. Add the query parameter `?multipart-manifest=put` to upload the manifest file.

```
curl -v -X PUT \
        -H "X-Auth-Token: AUTH_tk5a58b7a8c34bb7b662523a59a5272650"
        "https://acme.storage.oraclecloud.com/v1/Storage-acme/FirstContainer/
myLargeFile.manifest?multipart-manifest=put" \
        -T ./manifest.json
```

6. Download the large object by sending a `GET` request. All the segment objects are concatenated and downloaded as one large object.

```
curl -v -X GET \
        -H "X-Auth-Token: AUTH_tk5a58b7a8c34bb7b662523a59a5272650"
        https://acme.storage.oraclecloud.com/v1/Storage-acme/FirstContainer/
myLargeFile.manifest \
        -o ./myLargeFile
```

   Download the manifest object by sending a `GET` request and add the query parameter `?multipart-manifest=get`.

```
curl -v -X GET \
        -H "X-Auth-Token: AUTH_tk5a58b7a8c34bb7b662523a59a5272650"
        "https://acme.storage.oraclecloud.com/v1/Storage-acme/FirstContainer/
myLargeFile.manifest?multipart-manifest=get" \
        -o ./manifestFile
```

7. Run a `HEAD` request to view the size of the large object (`myLargeFile`) that you created:

```
curl -v -X HEAD \
     -H "X-Auth-Token: AUTH_tkbaebb60dfa5b80d84e62b0d5d07031e5"
         https://acme.storage.oraclecloud.com/v1/Storage-acme/FirstContainer/
myLargeFile
```

   The following is the output of this command:

```
> HEAD /v1/Storage-acme/FirstContainer/myLargeFile HTTP/1.1
> User-Agent: curl/7.29.0
> Host: acme.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tkc9305a46ebaa0585c4c7ae063c844f0b
< Etag: "e6da53c20abee5c471fe8bf796abb1a4"
< Accept-Ranges: bytes
< Last-Modified: Tue, 15 Dec 2015 10:07:53 GMT
< X-Timestamp: 1455012472.56679
< X-Trans-Id: txcab964b91ba8474ca9193-0056b9bb6fga
< Date: Tue, 15 Dec 2015 10:12:00 GMT
< Connection: keep-alive
< X-Last-Modified-Timestamp: 1455012472.56679
< Content-Type: application/octet-stream;charset=UTF-8
< Content-Length: 104857600
curl: (18) transfer closed with 52428800 bytes remaining to read
```

You can view the size of the large object in the header `Content-Length`. The size of the large object is the sum total of the sizes of the segment objects. To download the large object, see Downloading Large Objects.

Making an Object Immutable

You can make an object immutable by setting it's Write-Once-Read-Many (WORM) policy when uploading it to the container to prevent the users from modifying or deleting it for a specified duration.

Once you've set an object's WORM policy, you cannot change it. When the WORM policy expires, you can delete the object but can't modify it or it's metadata. If the container to which you intend to upload the object has a WORM policy, then the duration of the object-level WORM policy must be equal to or higher than the duration of the container-level policy. To learn more about setting a container's WORM policy, see Making Objects in a Container Immutable.

Any user with the `Service Administrator` role or a role that is specified in the `X-Container-Write` ACL of the container can set the object's WORM policy headers `X-Worm-Expiration-Days` and `X-Worm-Expiration-At`.

For information about using the REST API to set the object metadata, see Create or Replace Object in *REST API for Standard Storage in Oracle Cloud Infrastructure Object Storage Classic*.

cURL Command Syntax

To set the WORM policy of an object by specifying the duration of validity of the policy:

```
curl -v -X PUT \
     -H "X-Auth-Token: token" \
     -H "X-Worm-Expiration-Days: period" \
     -H "Content-Length: length"
     accountURL/containerName/objectName
```

To set the WORM policy of an object by setting the expiration time:

```
curl -v -X PUT \
     -H "X-Auth-Token: token" \
     -H "X-Worm-Expiration-At: time" \
     -H "Content-Length: length"
     accountURL/containerName/objectName
```

- `token` is the authentication token obtained earlier from Oracle Cloud Infrastructure Object Storage Classic. See Authenticating Access When Using the REST API.
- `period` is the duration, in days, for which the WORM policy is set for the container.
- `time` is the UNIX Epoch timestamp representing the date and time at which the WORM policy must expire. For example, 1481364600 represents December 10, 2016 10:10:00 GMT. See http://epochconverter.com.
  This value must be greater than the current Epoch time.
- `length` is the size of the object.
- For the syntax of `accountURL`, see About REST URLs for Oracle Cloud Infrastructure Object Storage Classic Resources.
- `containerName` is the name of the container to which the object must be uploaded.

- `objectName` is the name of the object for which the WORM policy must be set.

> **Note:**
>
> When you send a REST API request to Oracle Cloud Infrastructure Object Storage Classic, all non-ASCII characters in container names, object names and metadata values must be URL-encoded. For example, `my container` should be encoded as `my%20container`, where `%20` is the HTML encoding for the space character. Similarly, `my Über Container` should be encoded as `my %20%C3%9Cber%20Container`, where `%20` represents the space character and `%C3%9C` is the Ü character.

HTTP Response Codes

- Success: `201 Created`
- Failure: See Error Code Reference for Object Storage Classic

For information about getting object metadata, see Getting Object Metadata.

cURL Command Examples

Sample Cloud account with the following details:

- Account name: `acme`
- REST Endpoint URL: `https://acme.storage.oraclecloud.com/v1/Storage-acme`
- REST Endpoint (Permanent) URL: `https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/Storage-7b16fede61e1417ab83eb52e06f0e365`

> **Note:**
>
> The REST Endpoint (Permanent) URL is displayed for the accounts created after November 2017.

The REST Endpoint URL for the sample Cloud account is used in all the cURL command examples in this section. To use the REST Endpoint (Permanent) URL, replace `https://acme.storage.oraclecloud.com/v1/Storage-acme` with `https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/Storage-7b16fede61e1417ab83eb52e06f0e365`.

- This command sets the WORM policy of `4` days for the object `FirstObject` uploaded to the container `FirstContainer`:

```
curl -v -X PUT \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     -H "X-Worm-Expiration-Days: 4" \
     -H "Content-Length: 0"
     https://acme.storage.oraclecloud.com/v1/Storage-acme/FirstContainer/
FirstObject
```

  The following is an example of the output of this command:

```
> PUT /v1/Storage-acme/FirstContainer/FirstObject HTTP/1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.14.0.0
zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: acme.storage.oraclecloud.com
> Accept: */*
> X-Worm-Expiration-Days: 4
> X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b
> Content-Length: 0

< HTTP/1.1 201 Created
< Date: Fri, 06 Dec 2016 11:41:20 GMT
< Content-Type: text/html; charset=UTF-8
< X-Trans-Id: txe8869b3edea348e5b49eb-0054f99088
< Cache-Control: no-cache
< Pragma: no-cache
< Content-Language: en
```

- This command sets the WORM policy to expire at `Thu, 08 Dec 2016 22:00:00 GMT` represented by the Epoch timestamp `1481234400` for the object `SecondObject` uploaded to the container `FirstContainer`:

  ```
  curl -v -X PUT \
       -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
       -H "X-Worm-Expiration-At: 1481234400" \
       -H "Content-Length: 12"
       https://acme.storage.oraclecloud.com/v1/Storage-acme/FirstContainer/
  SecondObject
  ```

  The following is an example of the output of this command:

  ```
  > PUT /v1/Storage-acme/FirstContainer/SecondObject HTTP/1.1
  > User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.14.0.0
  zlib/1.2.3 libidn/1.18 libssh2/1.4.2
  > Host: acme.storage.oraclecloud.com
  > Accept: */*
  > X-Worm-Expiration-At: 1481234400
  > X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b
  > Content-Length: 12

  < HTTP/1.1 201 Created
  < Date: Fri, 06 Dec 2016 11:43:50 GMT
  < Content-Type: text/html; charset=UTF-8
  < X-Trans-Id: txe8869b3edea348e5b49eb-0054f99095
  < Cache-Control: no-cache
  < Pragma: no-cache
  < Content-Language: en
  ```

Getting Object Metadata

Any user with the Service Administrator role or a role that is specified in the `X-Container-Read` ACL of the container can perform this task.

You can retrieve information about an object in a container by sending a `HEAD` request, which returns the following information:

- Object type (`Content-Type`)
- Object size (`Content-Length`)
- MD5 Checksum value (`Etag`)

You can retrieve object metadata by using the following interfaces:

Interface	Resources
RESTful API	See Show Object Metadata in *REST API for Standard Storage in Oracle Cloud Infrastructure Object Storage Classic*.
File Transfer Manager API	See class: ObjectMetadata in *Java API Reference for Oracle Cloud Infrastructure Object Storage Classic File Transfer Manager*.
File Transfer Manager CLI	See setQuotaBytes() in *Command-Line Reference for Oracle Cloud Infrastructure Object Storage Classic*.

cURL Command Syntax

```
curl -v -X HEAD \
     -H "X-Auth-Token: token" \
     accountURL/containerName/objectName
```

- `token` is the authentication token obtained earlier from Oracle Cloud Infrastructure Object Storage Classic. See Authenticating Access When Using the REST API.
- For the syntax of `accountURL`, see About REST URLs for Oracle Cloud Infrastructure Object Storage Classic Resources.
- `containerName` is the name of the container.
- `objectName` is the name of the object for which metadata should be retrieved.

> **Note:**
>
> When you send a REST API request to Oracle Cloud Infrastructure Object Storage Classic, all non-ASCII characters in container names, object names and metadata values must be URL-encoded. For example, `my container` should be encoded as `my%20container`, where `%20` is the HTML encoding for the space character. Similarly, `my Über Container` should be encoded as `my %20%C3%9Cber%20Container`, where `%20` represents the space character and `%C3%9C` is the Ü character.

HTTP Response Codes

- Success: `200 OK`
- Failure: See Error Code Reference for Oracle Cloud Infrastructure Object Storage Classic

cURL Command Examples

Sample Cloud account with the following details:

- Account name: `acme`
- REST Endpoint URL: `https://acme.storage.oraclecloud.com/v1/Storage-acme`
- REST Endpoint (Permanent) URL: `https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/Storage-7b16fede61e1417ab83eb52e06f0e365`

> **Note:**
>
> The REST Endpoint (Permanent) URL is displayed for the accounts created after November 2017.

- Using the REST Endpoint URL obtained from the **REST Endpoint** field:

```
curl -v -X HEAD \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     https://acme.storage.oraclecloud.com/v1/Storage-acme/FirstContainer/myObject
```

The following is an example of the output of this command:

```
> HEAD /v1/acme.storage.oraclecloud.com/v1/Storage-acme/FirstContainer/myObject
HTTP/1.1
> User-Agent: curl/7.29.0
> Host: acme.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tk4863bdbfb5eee0d6d452eed5348d21ed
>
< HTTP/1.1 200 OK
< Accept-Ranges: bytes
< Last-Modified: Wed, 16 Dec 2015 08:14:17 GMT
< Etag: f1c9645dbc14efddc7d8a322685f26eb
< X-Timestamp: 1450253656.45313
< X-Trans-Id: txb3d80329b3ec4915971c0-0056a9d7c0ga
< Date: Thu, 28 Jan 2016 08:56:32 GMT
< Connection: keep-alive
< X-Last-Modified-Timestamp: 1450253656.45313
< Content-Type: application/octet-stream;charset=UTF-8
< Content-Length: 10485760
```

- Using the Service Permanent REST Endpoint URL obtained from the **REST Endpoint (Permanent)** field:
  This cURL command example applies to the accounts created after November 2017.

```
curl -v -X HEAD \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/
Storage-7b16fede61e1417ab83eb52e06f0e365/FirstContainer/myObject
```

The following is an example of the output of this command:

```
> HEAD v1/storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/
Storage-7b16fede61e1417ab83eb52e06f0e365/FirstContainer/myObject HTTP/1.1
> User-Agent: curl/7.29.0
> Host: storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tk4863bdbfb5eee0d6d452eed5348d21ed
>
< HTTP/1.1 200 OK
< Accept-Ranges: bytes
< Last-Modified: Wed, 16 Dec 2015 08:14:17 GMT
< Etag: f1c9645dbc14efddc7d8a322685f26eb
< X-Timestamp: 1450253656.45313
< X-Trans-Id: txb3d80329b3ec4915971c0-0056a9d7c0ga
< Date: Thu, 28 Jan 2016 08:56:32 GMT
< Connection: keep-alive
< X-Last-Modified-Timestamp: 1450253656.45313
```

```
< Content-Type: application/octet-stream;charset=UTF-8
< Content-Length: 10485760
```

Getting Object Metadata for Large Objects

Any user with the Service Administrator role or a role that is specified in the `X-Container-Read` ACL of the container can perform this task.

You can retrieve information about a large object in a container by sending a `HEAD` request, which returns the following information:

- Object type (`Content-Type`)
- Object size (`Content-Length`)
- MD5 Checksum value (`Etag`)

For information about retrieving large object metadata by using the REST API, see Get Object Metadata in *OpenStack Object Storage Service API Reference*. The Java library does not support this task.

cURL Command Syntax

```
curl -v -X HEAD \
     -H "X-Auth-Token: token" \
     accountURL/containerName/manifestObjectName
```

- `token` is the authentication token obtained earlier from Oracle Cloud Infrastructure Object Storage Classic. See Authenticating Access When Using the REST API.
- For the syntax of `accountURL`, see About REST URLs for Oracle Cloud Infrastructure Object Storage Classic Resources.
- `containerName` is the name of the container.
- `manifestObjectName` is the name of the large object for which metadata should be retrieved.

HTTP Response Codes

- Success: `200 OK`
- Failure: See Error Code Reference for Oracle Cloud Infrastructure Object Storage Classic

cURL Command Examples

Sample Cloud account with the following details:

- Account name: `acme`
- REST Endpoint URL: `https://acme.storage.oraclecloud.com/v1/Storage-acme`
- REST Endpoint (Permanent) URL: `https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/Storage-7b16fede61e1417ab83eb52e06f0e365`

> **Note:**
>
> The REST Endpoint (Permanent) URL is displayed for the accounts created after November 2017.

- Using the REST Endpoint URL obtained from the **REST Endpoint** field:

```
curl -v -X HEAD \
     -H "X-Auth-Token: AUTH_tka6b18380f0e22c57d721b9101d425759" \
     https://acme.storage.oraclecloud.com/v1/Storage-acme/FirstContainer/
LargeFile.manifest
```

  The following is an example of the output of this command:

```
> HEAD /v1/acme.storage.oraclecloud.com/v1/Storage-acme/FirstContainer/
LargeFile.manifest HTTP/1.1
> User-Agent: curl/7.29.0
> Host: acme.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tka6b18380f0e22c57d721b9101d425759
>
< HTTP/1.1 200 OK
< Etag: "6e9647a0cd268b9299e647d28f7027f7"
< Accept-Ranges: bytes
< Last-Modified: Mon, 25 Jan 2016 18:48:22 GMT
< X-Object-Manifest: FirstContainer/segment_
< X-Timestamp: 1453747701.91311
< X-Trans-Id: txbfb035e9ddf24d96bf602-0056ab5bf2ga
< Date: Fri, 29 Jan 2016 12:32:50 GMT
< Connection: keep-alive
< X-Last-Modified-Timestamp: 1453747701.91311
< Content-Type: application/zip;charset=UTF-8
< Content-Length: 52428800
```

- Using the Service Permanent REST Endpoint URL obtained from the **REST Endpoint (Permanent)** field:
  This cURL command example applies to the accounts created after November 2017.

```
curl -v -X HEAD \
     -H "X-Auth-Token: AUTH_tka6b18380f0e22c57d721b9101d425759" \
     https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/
Storage-7b16fede61e1417ab83eb52e06f0e365/FirstContainer/LargeFile.manifest
```

  The following is an example of the output of this command:

```
> HEAD /v1/storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/
Storage-7b16fede61e1417ab83eb52e06f0e365/FirstContainer/LargeFile.manifest HTTP/
1.1
> User-Agent: curl/7.29.0
> Host: storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tka6b18380f0e22c57d721b9101d425759
>
< HTTP/1.1 200 OK
< Etag: "6e9647a0cd268b9299e647d28f7027f7"
< Accept-Ranges: bytes
< Last-Modified: Mon, 25 Jan 2016 18:48:22 GMT
< X-Object-Manifest: FirstContainer/segment_
< X-Timestamp: 1453747701.91311
```

```
< X-Trans-Id: txbfb035e9ddf24d96bf602-0056ab5bf2ga
< Date: Fri, 29 Jan 2016 12:32:50 GMT
< Connection: keep-alive
< X-Last-Modified-Timestamp: 1453747701.91311
< Content-Type: application/zip;charset=UTF-8
< Content-Length: 52428800
```

Finding Out the Status of Objects in an Archive Container


This topic does not apply to Oracle Cloud at Customer.

Any user with a role that is specified in the `X-Container-Read` ACL of a container can find out the status of an object in an `Archive` container.

Finding Out the Status of Objects Using the Web Console

1. Sign in to the Oracle Cloud Infrastructure Object Storage Classic console.
2. Select the `Archive` container from which you would like to find out the status of an object.
   The objects in the container are listed.
3. Identify the object for which you want to find out the status.
4. Select **Check Status**.
   - If the object is archived, the **Checking current status of archive object** dialog box appears with the following message:

     `Currently the object is archived.`

     If you would like to restore the object, select **Restore**. See Restoring Archived Objects.

     Else, click **Cancel**.
   - If the archived object is being restored, the **Checking current status of archive object** dialog box appears with the following message:

     `Restoration job is in progress.`
5. Click **OK**.

Finding Out the Status of Objects Using the REST API

After creating or restoring an object in an `Archive` container, you can find out the status of the object from its header `X-Archive-Restore-Status`, which can have one of the following values:

- `archived`: Indicates that the object is archived
- `restored`: Indicates that the object is restored
- `inprogress`: Indicates that object restoration is in progress

cURL Command Syntax

```
curl -v -X HEAD \
     -H "X-Auth-Token: token" \
     accountURL/containerName/objectName
```

- `token` is the authentication token obtained earlier from Oracle Cloud Infrastructure Object Storage Classic. See Authenticating Access When Using the REST API.
- For the syntax of `accountURL`, see About REST URLs for Oracle Cloud Infrastructure Object Storage Classic Resources.
- `containerName` is the name of the container that contains the object whose archival or restoration status needs to be identified.
- `objectName` is the name of the object for which you want to find out the restoration status.

> **Note:**
>
> When you send a REST API request to Oracle Cloud Infrastructure Object Storage Classic, all non-ASCII characters in container names, object names and metadata values must be URL-encoded. For example, `my container` should be encoded as `my%20container`, where `%20` is the HTML encoding for the space character. Similarly, `my Über Container` should be encoded as `my %20%C3%9Cber%20Container`, where `%20` represents the space character and `%C3%9C` is the `Ü` character.

HTTP Response Codes

- Success: `200 OK`
- Failure: See Error Code Reference for Oracle Cloud Infrastructure Object Storage Classic

cURL Command Examples

Sample Cloud account with the following details:

- Account name: `acme`
- REST Endpoint URL: `https://acme.storage.oraclecloud.com/v1/Storage-acme`
- REST Endpoint (Permanent) URL: `https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/Storage-7b16fede61e1417ab83eb52e06f0e365`

> **Note:**
>
> The REST Endpoint (Permanent) URL is displayed for the accounts created after November 2017.

- Using the REST Endpoint URL obtained from the **REST Endpoint** field:

```
curl -v -X HEAD \
     -H "X-Auth-Token: AUTH_tkb237a55e17b772a7579f433d8b4f6e05" \
     https://acme.storage.oraclecloud.com/v1/Storage-acme/firstArchiveContainer/
obj1.txt
```

The following is an example of the output of this command with the object status as `archived`:

```
> HEAD /v1/Storage-acme/firstArchiveContainer/obj1.txt HTTP/1.1
> User-Agent: curl/7.29.0
> Host: acme.storage.oraclecloud.com:443
> Accept: */*
> X-Auth-Token: AUTH_tkb237a55e17b772a7579f433d8b4f6e05
>
< HTTP/1.1 200 OK
< X-Archive-Restore-Status: archived
< Accept-Ranges: bytes
< Last-Modified: Thu, 18 Jun 2015 19:45:26 GMT
< Etag: 3db2050fcf84bb631dcae417d3db518c
< X-Timestamp: 1434656725.77022
< X-Trans-Id: tx37fb896a0a8144f3860da-0055832016ga
< Date: Thu, 18 Jun 2015 19:46:30 GMT
< Content-Type: text/plain;charset=UTF-8
< Content-Length: 12
<
```

- Using the Service Permanent REST Endpoint URL obtained from the **REST Endpoint (Permanent)** field:

  > **Note:**
  >
  > This cURL command example applies to the accounts created after November 2017.

  ```
  curl -v -X HEAD \
       -H "X-Auth-Token: AUTH_tkb237a55e17b772a7579f433d8b4f6e05" \
       https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/
  Storage-7b16fede61e1417ab83eb52e06f0e365/firstArchiveContainer/obj1.txt
  ```

  The following is an example of the output of this command with the object status as `archived`:

  ```
  > HEAD /v1/Storage-7b16fede61e1417ab83eb52e06f0e365/firstArchiveContainer/
  obj1.txt HTTP/1.1
  > User-Agent: curl/7.29.0
  > Host: storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com:443
  > Accept: */*
  > X-Auth-Token: AUTH_tkb237a55e17b772a7579f433d8b4f6e05
  >
  < HTTP/1.1 200 OK
  < X-Archive-Restore-Status: archived
  < Accept-Ranges: bytes
  < Last-Modified: Thu, 18 Jun 2015 19:45:26 GMT
  < Etag: 3db2050fcf84bb631dcae417d3db518c
  < X-Timestamp: 1434656725.77022
  < X-Trans-Id: tx37fb896a0a8144f3860da-0055832016ga
  < Date: Thu, 18 Jun 2015 19:46:30 GMT
  < Content-Type: text/plain;charset=UTF-8
  < Content-Length: 12
  <
  ```

Restoring Archived Objects


This topic does not apply to Oracle Cloud at Customer.

Any user with the Service Administrator role or a role that is specified in the `X-Container-Read` ACL of the container can perform this task.

To download an archived object, the object must first be restored. Restoring archived objects is an asynchronous operation. It can take up to four hours to restore an object. When you have requested to restore an object, a Job ID and a tracking URL are returned. This information can be used to monitor the object's restoration progress.

You can restore archived objects in a container by using the following interfaces:

Interface	Resources
Web Console	See Restoring an Archived Object Using the Web Console.
RESTful API	See Trigger the retrieval of an archived object in *REST API for Archive Storage in Oracle Cloud Infrastructure Object Storage Classic*.
File Transfer Manager CLI	See Restoring an Object in *Command-Line Reference for Oracle Cloud Infrastructure Object Storage Classic*.
File Transfer Manager API	See restoreObject in *Java API Reference for Oracle Cloud Infrastructure Object Storage Classic File Transfer Manager*.

Restoring an Archived Object Using the Web Console

1. Sign in to the Oracle Cloud Infrastructure Object Storage Classic console.
2. Select the `Archive` container from which the archived object must be restored. The objects in the container are listed.
3. Identify the archived object that you want to restore.
4. Select **Check Status**.
   The **Checking current status of archive object** dialog box appears with the following message:

   ```
   Currently the object is archived.
   ```
5. Select **Restore** to restore the archived object.
   The following message appears:

   ```
   Are you sure you want to restore the object objectName?
   ```
6. Click **OK**.
   The following message appears:

   ```
   Restore job for objectName initiated successfully. You can check the
   Restore Status by clicking the Check Status button.
   ```

   By default, the object remains restored for one day, after which you must restore it again to be able to download it.
7. Click **OK**.

Restoring an Archived Object Using the REST API

cURL Command Syntax

```
curl -v -X POST \
     -H "X-Auth-Token: token" \
     "accountURL/containerName/objectName?restore"
```

- `token` is the authentication token obtained earlier from Oracle Cloud Infrastructure Object Storage Classic. See Authenticating Access When Using the REST API.
- For the syntax of `accountURL`, see About REST URLs for Oracle Cloud Infrastructure Object Storage Classic Resources. Use `v0` in the REST API URL.
- `containerName` is the name of the container in which the object should be restored.
- `objectName` is the name of the object to be restored.
- `restore` is the query parameter to restore the object.

> **Note:**
>
> When you send a REST API request to Oracle Cloud Infrastructure Object Storage Classic, all non-ASCII characters in container names, object names and metadata values must be URL-encoded. For example, `my container` should be encoded as `my%20container`, where `%20` is the HTML encoding for the space character. Similarly, `my Über Container` should be encoded as `my%20%C3%9Cber%20Container`, where `%20` represents the space character and `%C3%9C` is the Ü character.

HTTP Response Codes

- Success: `202 Accepted`
- Failure: See Error Code Reference for Oracle Cloud Infrastructure Object Storage Classic

cURL Command Example

Sample Cloud account with the following details:

- Account name: `acme`
- REST Endpoint URL: `https://acme.storage.oraclecloud.com/v1/Storage-acme`
- REST Endpoint (Permanent) URL: `https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/Storage-7b16fede61e1417ab83eb52e06f0e365`

> **Note:**
>
> The REST Endpoint (Permanent) URL is displayed for the accounts created after November 2017.

- Using the REST Endpoint URL obtained from the **REST Endpoint** field:

```
curl -v -X POST \
     -H "X-Auth-Token: AUTH_tk1ff0554c1fefff9209696d63553722fd" \
     "https://acme.storage.oraclecloud.com/v0/Storage-acme/firstArchiveContainer/
file.txt?restore"
```

The following is an example of the output of this command:

```
> HEAD /v1/Storage-acme/firstArchiveContainer/file.txt HTTP/1.1
> User-Agent: curl/7.29.0
> Host: acme.storage.oraclecloud.com:443
> Accept: */*
> X-Auth-Token: AUTH_tk1ff0554c1fefff9209696d63553722fd
>
> POST /v0/Storage-acme/firstArchiveContainer/file.txt?restore HTTP/1.1
> User-Agent: curl/7.29.0
> Host: acme.storage.oraclecloud.com:443
> Accept: */*
> X-Auth-Token: AUTH_tk1ff0554c1fefff9209696d63553722fd
< HTTP/1.1 202 Accepted
< Location: acme.storage.oraclecloud.com:443/v0/Storage-acme/
firstArchiveContainer?jobs&jobid=a75c8bbf53224a88738e68d6628acd83a4b300e4
< X-Archive-Restore-Tracking: http://acme.storage.oraclecloud.com:443/v0/Storage-
acme/TestArch?jobs&jobid=a75c8bbf53224a88738e68d6628acd83a4b300e4
< X-Archive-Restore-JobId: a75c8bbf53224a88738e68d6628acd83a4b300e4
< Content-Length: 0
< Date: Thu, 18 Jun 2015 17:53:41 GMT
<
```

> **Note:**
>
> By default, the object remains restored for one day, after which you must restore it again to be able to download it. You can change the period (in days) that an object remains restored, by specifying that period in the `X-Archive-Restore-Expiration` header, as shown in the following example.
>
> ```
> curl -v -X POST \
>      -H "X-Auth-Token: AUTH_tk1ff0554c1fefff9209696d63553722fd" \
>      -H "X-Archive-Restore-Expiration: 3" \
>      "https://acme.storage.oraclecloud.com/v0/Storage-acme/
> firstArchiveContainer/file.txt?restore
> ```

- Using the Service Permanent REST Endpoint URL obtained from the **REST Endpoint (Permanent)** field:
  This cURL command example applies to the accounts created after November 2017.

  ```
  curl -v -X POST \
       -H "X-Auth-Token: AUTH_tk1ff0554c1fefff9209696d63553722fd" \
       "https://
  storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/
  Storage-7b16fede61e1417ab83eb52e06f0e365/firstArchiveContainer/file.txt?restore"
  ```

  The following is an example of the output of this command:

  ```
  > HEAD /v1/Storage-7b16fede61e1417ab83eb52e06f0e365/firstArchiveContainer/
  file.txt HTTP/1.1
  > User-Agent: curl/7.29.0
  > Host: storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com:443
  > Accept: */*
  > X-Auth-Token: AUTH_tk1ff0554c1fefff9209696d63553722fd
  >
  > POST /v0/Storage-7b16fede61e1417ab83eb52e06f0e365/firstArchiveContainer/
  ```

```
file.txt?restore HTTP/1.1
> User-Agent: curl/7.29.0
> Host: storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com:443
> Accept: */*
> X-Auth-Token: AUTH_tk1ff0554c1fefff9209696d63553722fd
< HTTP/1.1 202 Accepted
< Location: https://
storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/
Storage-7b16fede61e1417ab83eb52e06f0e365/firstArchiveContainer?
jobs&jobid=a75c8bbf53224a88738e68d6628acd83a4b300e4
< X-Archive-Restore-Tracking: http://
storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com:443/v0/
Storage-7b16fede61e1417ab83eb52e06f0e365/TestArch?
jobs&jobid=a75c8bbf53224a88738e68d6628acd83a4b300e4
< X-Archive-Restore-JobId: a75c8bbf53224a88738e68d6628acd83a4b300e4
< Content-Length: 0
< Date: Thu, 18 Jun 2015 17:53:41 GMT
<
```

Note:

By default, the object remains restored for one day, after which you must restore it again to be able to download it. You can change the period (in days) that an object remains restored, by specifying that period in the `X-Archive-Restore-Expiration` header, as shown in the following example.

```
curl -v -X POST \
     -H "X-Auth-Token: AUTH_tk1ff0554c1fefff9209696d63553722fd" \
     -H "X-Archive-Restore-Expiration: 3" \
     "https://
storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v0/
Storage-7b16fede61e1417ab83eb52e06f0e365/firstArchiveContainer/
file.txt?restore
```

Next Step

You can now track the restoration progress of the object in the `Archive` container. To track the object's restoration progress, make a note of the URL in the `X-Archive-Restore-Tracking` header and the job ID in the header `X-Archive-Restore-JobId` when you restore an object. For more information, see Tracking Restoration of an Object in an Archive Container.

Tracking Restoration of an Object in an Archive Container


This topic does not apply to Oracle Cloud at Customer.

Any user with a role that is specified in the `X-Container-Read` ACL of the container can track the restoration of an object in an `Archive` container.

It might take up to four hours for an object to be restored and accessible.

You can track the restoration progress of the object from the URL in the `X-Archive-Restore-Tracking` status header in the object metadata.

The following details are displayed when you track the restoration progress:

- Restoration Start Time
- Restoration End Time (if the restoration is complete)
- Restoration Progress
- Percentage of Restoration Completion
- Job Details
- Object Size
- Object Etag
- Restored Object Expiration time
- Link to archived version of the object
- Job Type
- Job ID

For information about tracking restoration of an object in an `archive` container by using the REST API, see Return status of restore job in *REST API for Archive Storage in Oracle Cloud Infrastructure Object Storage Classic*.

cURL Command Syntax

```
curl -v -X GET \
     -H "X-Auth-Token: token" \
     "accountURL/containerName?jobs&jobID"
```

- `token` is the authentication token obtained earlier from Oracle Cloud Infrastructure Object Storage Classic. See Authenticating Access When Using the REST API.
- For the syntax of `accountURL`, see About REST URLs for Oracle Cloud Infrastructure Object Storage Classic Resources. Use `v0` in the REST API URL.
- `containerName` is the name of the container in which the object should be restored.
- `jobID` is the job ID returned in the header `X-Archive-Restore-JobId` when you restore an object. See Restoring Archived Objects.

> **Note:**
>
> When you send a REST API request to Oracle Cloud Infrastructure Object Storage Classic, all non-ASCII characters in container names, object names and metadata values must be URL-encoded. For example, `my container` should be encoded as `my%20container`, where `%20` is the HTML encoding for the space character. Similarly, `my Über Container` should be encoded as `my %20%C3%9Cber%20Container`, where `%20` represents the space character and `%C3%9C` is the Ü character.

HTTP Response Codes

- Success: `200 OK`

- Failure: See Error Code Reference for Oracle Cloud Infrastructure Object Storage Classic

cURL Command Examples

- Account name: `acme`
- REST Endpoint URL: `https://acme.storage.oraclecloud.com/v1/Storage-acme`
- REST Endpoint (Permanent) URL: `https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/Storage-7b16fede61e1417ab83eb52e06f0e365`

> **Note:**
>
> The REST Endpoint (Permanent) URL is displayed for the accounts created after November 2017.

- Using the REST Endpoint URL obtained from the **REST Endpoint** field:

```
curl -v -X GET \
     -H "X-Auth-Token: AUTH_tk1ff0554c1fefff9209696d63553722fd"
     "https://acme.storage.oraclecloud.com/v0/Storage-acme/firstArchiveContainer?
jobs&jobid=a75c8bbf53224a88738e68d6628acd83a4b300e4"
```

The following is an example of the output of this command:

```
> GET /v0/Storage-acme/firstArchiveContainer?
jobs&jobid=a75c8bbf53224a88738e68d6628acd83a4b300e4 HTTP/1.1
> User-Agent: curl/7.29.0
> Host: acme.storage.oraclecloud.com:443
> Accept: */*
> X-Auth-Token: AUTH_tk1ff0554c1fefff9209696d63553722fd
>
< HTTP/1.1 200 OK
< Transfer-Encoding: chunked
< Date: Thu, 18 Jun 2015 17:56:14 GMT
<
* Connection #0 to host acme.storage.oraclecloud.com left intact
 {
 "endTime":"2015-06-18,17:54",
 "progress":"success","completedPercentage":100,
 "intervalToPoll":10,
 "jobDetails":{
    "objectSizeInBytes":"20",
    "objectEtag":"4221d002ceb5d3c9e9137e495ceaa647",
    "objectExpiration":"1434736481424"
    },
 "links":
  [
    {"rel":"self",
     "href":"http://acme.storage.oraclecloud.com:443/v0/Storage-acme/TestArch?
jobs&jobid=a75c8bbf53224a88738e68d6628acd83a4b300e4"},
    {"rel":"original","href":"http://acme.storage.oraclecloud.com:443/v1/Storage-
acme/TestArch/file.txt"},
    {"rel":"canonical","href":"http://acme.storage.oraclecloud.com:443/v1/
Storage-acme/TestArch/file.txt"}
    ],
 "startTime":"2015-06-18,17:53",
```

```
 "completed":true,
 "jobType":"RestoreArchivedObjectJob",
 "jobId":"a75c8bbf53224a88738e68d6628acd83a4b300e4"
 }
```

The `objectExpiration` field in the response body shows the UNIX Epoch time stamp representing the date and time when the restored object will expire. For example, `1434736481424` represents June 19, 2015 17:54:41 GMT. See http://www.epochconverter.com/. The standard expiry duration is 24 hours.

- Using the Service Permanent REST Endpoint URL obtained from the **REST Endpoint (Permanent)** field:
  This cURL command example applies to the accounts created after November 2017.

```
curl -v -X GET \
     -H "X-Auth-Token: AUTH_tk1ff0554c1fefff9209696d63553722fd"
     "https://
storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v0/
Storage-7b16fede61e1417ab83eb52e06f0e365/firstArchiveContainer?
jobs&jobid=a75c8bbf53224a88738e68d6628acd83a4b300e4"
```

The following is an example of the output of this command:

```
> GET /v0/Storage-7b16fede61e1417ab83eb52e06f0e365/firstArchiveContainer?
jobs&jobid=a75c8bbf53224a88738e68d6628acd83a4b300e4 HTTP/1.1
> User-Agent: curl/7.29.0
> Host: storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com:443
> Accept: */*
> X-Auth-Token: AUTH_tk1ff0554c1fefff9209696d63553722fd
>
< HTTP/1.1 200 OK
< Transfer-Encoding: chunked
< Date: Thu, 18 Jun 2015 17:56:14 GMT
<
* Connection #0 to host
storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com left intact
 {
 "endTime":"2015-06-18,17:54",
 "progress":"success","completedPercentage":100,
 "intervalToPoll":10,
 "jobDetails":{
    "objectSizeInBytes":"20",
    "objectEtag":"4221d002ceb5d3c9e9137e495ceaa647",
    "objectExpiration":"1434736481424"
    },
 "links":
  [
    {"rel":"self",
     "href":"http://
storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com:443/v0/
Storage-7b16fede61e1417ab83eb52e06f0e365/TestArch?
jobs&jobid=a75c8bbf53224a88738e68d6628acd83a4b300e4"},
    {"rel":"original","href":"http://
storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com:443/v1/
Storage-7b16fede61e1417ab83eb52e06f0e365/TestArch/file.txt"},
    {"rel":"canonical","href":"http://
storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com:443/v1/
Storage-7b16fede61e1417ab83eb52e06f0e365/TestArch/file.txt"}
    ],
 "startTime":"2015-06-18,17:53",
```

```
"completed":true,
"jobType":"RestoreArchivedObjectJob",
"jobId":"a75c8bbf53224a88738e68d6628acd83a4b300e4"
}
```

The `objectExpiration` field in the response body shows the UNIX Epoch time stamp representing the date and time when the restored object will expire. For example, `1434736481424` represents June 19, 2015 17:54:41 GMT. See http://www.epochconverter.com/. The standard expiry duration is 24 hours.

You can now download the restored object before it expires. For more information, see Downloading Objects.

Downloading Objects

When you download an object, the object's metadata and data are downloaded. You can download a single object or a large object.

Topics:

- Downloading an Object
- Downloading a Large Object
- Downloading an Object Using a Temporary URL

Downloading an Object

When you download an object, the object's metadata and data are downloaded.

Any user with the Service Administrator role or a role that is specified in the `X-Container-Read` ACL of the container can perform this task.

You can download objects from a container by using the following interfaces:

Interface	Resources
Web Console (Not available on Oracle Cloud at Customer)	See Downloading an Object Using the Web Console.
RESTful API	See Get object content and metadata in *REST API for Standard Storage in Oracle Cloud Infrastructure Object Storage Classic*.
Java Library	See `retrieveObject` in *Java API Reference for Oracle Cloud Infrastructure Object Storage Classic*.
File Transfer Manager API	See `download` in *Java API Reference for Oracle Cloud Infrastructure Object Storage Classic File Transfer Manager*.
File Transfer Manager CLI	See Downloading an Object in *Command-Line Reference for Oracle Cloud Infrastructure Object Storage Classic*.

Downloading an Object Using the Web Console

(Not available on Oracle Cloud at Customer)

1. Sign in to the Oracle Cloud Infrastructure Object Storage Classic console.

2. Select the container from which the object must be downloaded.
   The objects in the container are listed.
3. If you see the **Enable Upload and Download** button in the **Container Information** pane, you can't download the object.
   a. Click **Enable Upload and Download**.

      The following message appears:

      ```
      In order to upload and download objects to the current container,
      this application must enable Cross-Origin Resource Sharing (CORS)
      for this domain domainName.
      You can disable this setting at anytime.
      Would you like to continue?
      ```

   b. Click **OK** to enable Cross-Origin Resource Sharing (CORS).

      > **Note:**
      >
      > You can disable CORS in the container at any time. Select **Disable Upload and Download** in the **Container Information** pane to disable CORS.

4. Identify the object that you want to download.

   > **Note:**
   >
   > You can't download objects that are larger than 10 MB using the web console. To download such objects, use the CLI or REST API.

5. From the **Actions** drop-down list for the object to be downloaded, select **Download**.
   The object is downloaded to the desired location.

Downloading an Object Using the REST API

cURL Command Syntax

```
curl -v -X GET \
     -H "X-Auth-Token: token" \
     -o file \
     accountURL/containerName/objectName
```

- `token` is the authentication token obtained earlier from Oracle Cloud Infrastructure Object Storage Classic. See Authenticating Access When Using the REST API.
- `file` is the full path and name of the file to which the object should be downloaded.
- For the syntax of `accountURL`, see About REST URLs for Oracle Cloud Infrastructure Object Storage Classic Resources.
- `containerName` is the name of the container that contains the object to be downloaded.
- `objectName` is the name of the object to be downloaded.

> **Note:**
>
> When you send a REST API request to Oracle Cloud Infrastructure Object Storage Classic, all non-ASCII characters in container names, object names and metadata values must be URL-encoded. For example, `my container` should be encoded as `my%20container`, where `%20` is the HTML encoding for the space character. Similarly, `my Über Container` should be encoded as `my %20%C3%9Cber%20Container`, where `%20` represents the space character and `%C3%9C` is the Ü character.

HTTP Response Codes

- Success: `200 OK`
- Failure: See Error Code Reference for Oracle Cloud Infrastructure Object Storage Classic

cURL Command Examples

Sample Cloud account with the following details:

- Account name: `acme`
- REST Endpoint URL: `https://acme.storage.oraclecloud.com/v1/Storage-acme`
- REST Endpoint (Permanent) URL: `https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/Storage-7b16fede61e1417ab83eb52e06f0e365`

> **Note:**
>
> The REST Endpoint (Permanent) URL is displayed for the accounts created after November 2017.

- Using the REST Endpoint URL obtained from the **REST Endpoint** field:

```
curl -v -X GET \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     -o myFile.txt \
     https://acme.storage.oraclecloud.com/v1/Storage-acme/myContainer/myObject
```

The following is an example of the output of this command:

```
> GET /v1/Storage-acme/FirstContainer/myObject HTTP/1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.14.0.0
zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: acme.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b
>
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
<
< HTTP/1.1 200 OK
< Date: Mon, 09 Mar 2015 11:34:33 GMT
< Content-Length: 23
```

```
< Accept-Ranges: bytes
< Last-Modified: Mon, 09 Mar 2015 11:26:58 GMT
< Etag: 846fa9d298be05e5f598703f0c3d6f51
< X-Timestamp: 1425900417.95553
< Content-Type: application/octet-stream
< X-Trans-Id: txf0b592c7e49b4475944f8-0054fd8549
< Cache-Control: no-cache
< Pragma: no-cache
< Content-Language: en
<
{ [data not shown]
  0    23    0    23    0     0     53      0 --:--:-- --:--:-- --:--:--   234*
Connection #0 to host acme.storage.oraclecloud.com left intact
```

- Using the Service Permanent REST Endpoint URL obtained from the **REST Endpoint (Permanent)** field:

> **Note:**
>
> This cURL command example applies to the accounts created after November 2017.

```
curl -v -X GET \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     -o myFile.txt \
     https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/
Storage-7b16fede61e1417ab83eb52e06f0e365/myContainer/myObject
```

The following is an example of the output of this command:

```
> GET /v1/Storage-7b16fede61e1417ab83eb52e06f0e365/FirstContainer/myObject HTTP/
1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.14.0.0
zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b
>
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
<
< HTTP/1.1 200 OK
< Date: Mon, 09 Mar 2015 11:34:33 GMT
< Content-Length: 23
< Accept-Ranges: bytes
< Last-Modified: Mon, 09 Mar 2015 11:26:58 GMT
< Etag: 846fa9d298be05e5f598703f0c3d6f51
< X-Timestamp: 1425900417.95553
< Content-Type: application/octet-stream
< X-Trans-Id: txf0b592c7e49b4475944f8-0054fd8549
< Cache-Control: no-cache
< Pragma: no-cache
< Content-Language: en
<
{ [data not shown]
  0    23    0    23    0     0     53      0 --:--:-- --:--:-- --:--:--   234*
Connection #0 to host
```

```
storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com left intact
```

Downloading a Large Object

Large objects contain data more than 5 GB. To download a large object, you must download the manifest object which returns all the segments concatenated as a single large object.

You can download a large object by using the following interface:

Interface	Resources
File Transfer Manager CLI	See Downloading an Object in *Command-Line Reference for Oracle Cloud Infrastructure Object Storage Classic*.

cURL Command Syntax

To download a static large object:

```
curl -v -X GET \
     -H "X-Auth-Token: token" \
     -o largeObject \
     accountURL/containerName/manifestFile \
```

- `token` is the authentication token obtained earlier from Oracle Cloud Infrastructure Object Storage Classic. See Authenticating Access When Using the REST API.
- `largeObject` is the full path and name of the file to which the object should be downloaded.
- For the syntax of `accountURL`, see About REST URLs for Oracle Cloud Infrastructure Object Storage Classic Resources.
- `containerName` is the name of the container that contains the segment objects (that form the large object) to be downloaded.
- `manifestFile` is the name of the manifest file.

> **Note:**
>
> When you send a REST API request to Oracle Cloud Infrastructure Object Storage Classic, all non-ASCII characters in container names, object names and metadata values must be URL-encoded. For example, `my container` should be encoded as `my%20container`, where `%20` is the HTML encoding for the space character. Similarly, `my Über Container` should be encoded as `my %20%C3%9Cber%20Container`, where `%20` represents the space character and `%C3%9C` is the Ü character.

To download just the manifest object (and not the entire large object), append the `?multipart-manifest=get` query parameter to the `GET` request:

```
curl -v -X GET \
     -H "X-Auth-Token: token" \
     -o manifestFile \
```

```
accountURL/containerName/manifestObjectName?multipart-manifest=get \
```

cURL Command Examples

Sample Cloud account with the following details:

- Account name: `acme`
- REST Endpoint URL: `https://acme.storage.oraclecloud.com/v1/Storage-acme`
- REST Endpoint (Permanent) URL: `https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/Storage-7b16fede61e1417ab83eb52e06f0e365`

> **Note:**
>
> The REST Endpoint (Permanent) URL is displayed for the accounts created after November 2017.

- Using the REST Endpoint URL obtained from the **REST Endpoint** field:

```
curl -v -X GET \
       -H "X-Auth-Token: AUTH_tk5a58b7a8c34bb7b662523a59a5272650" \
       -o myLargeObject \
       https://acme.storage.oraclecloud.com/v1/Storage-acme/FirstContainer/
myLargeFile.manifest \
```

The following is an example of the output of this command:

```
> GET /v1/Storage-acme/myLargeFile.manifest HTTP/1.1
> User-Agent: curl/7.29.0
> Host: acme.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tk5a58b7a8c34bb7b662523a59a5272650
>
< HTTP/1.1 200 OK
< Etag: "6e9647a0cd268b9299e647d28f7027f7"
< Accept-Ranges: bytes
< Last-Modified: Mon, 20 Jan 2016 18:48:22 GMT
< X-Object-Manifest: FirstContainer/segment_
< X-Timestamp: 1453747701.91311
< X-Trans-Id: tx9f77e1e8b7b74de18dc53-0056a8a3f5ga
< Date: Wed, 20 Jan 2016 11:03:18 GMT
< Connection: keep-alive
< X-Last-Modified-Timestamp: 1453747701.91311
< Content-Type: application/zip;charset=UTF-8
< Content-Length: 52428800
```

- Using the Service Permanent REST Endpoint URL obtained from the **REST Endpoint (Permanent)** field:

> **Note:**
>
> This cURL command example applies to the accounts created after November 2017.

```
curl -v -X GET \
        -H "X-Auth-Token: AUTH_tk5a58b7a8c34bb7b662523a59a5272650" \
        -o myLargeObject \
        https://
storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/
Storage-7b16fede61e1417ab83eb52e06f0e365/FirstContainer/myLargeFile.manifest \
```

The following is an example of the output of this command:

```
> GET /v1/Storage-7b16fede61e1417ab83eb52e06f0e365/myLargeFile.manifest HTTP/1.1
> User-Agent: curl/7.29.0
> Host: storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tk5a58b7a8c34bb7b662523a59a5272650
>
< HTTP/1.1 200 OK
< Etag: "6e9647a0cd268b9299e647d28f7027f7"
< Accept-Ranges: bytes
< Last-Modified: Mon, 20 Jan 2016 18:48:22 GMT
< X-Object-Manifest: FirstContainer/segment_
< X-Timestamp: 1453747701.91311
< X-Trans-Id: tx9f77e1e8b7b74de18dc53-0056a8a3f5ga
< Date: Wed, 20 Jan 2016 11:03:18 GMT
< Connection: keep-alive
< X-Last-Modified-Timestamp: 1453747701.91311
< Content-Type: application/zip;charset=UTF-8
< Content-Length: 52428800
```

Downloading an Object Using a Temporary URL

You can create time-limited temporary URLs to provide secure, temporary access to download objects from Oracle Cloud Infrastructure Object Storage Classic.

To generate temporary URLs for objects, you must use a secret key that you specify beforehand either at the account level or at a container level. In either case, you can specify up to two keys.

- You can use an account-level key to download objects from any container in your Oracle Cloud Infrastructure Object Storage Classic instance.
- You can use a container-level key to download any object from that container.

Before You Begin

You can generate temporary URLs using the OpenStack Swift CLI or using a script written in any programming language. This document includes an example that uses the OpenStack Swift CLI and another example that uses a Python script.

Install the following software on a local host:

- Python 2.7
- OpenStack Swift CLI client (for installation instructions, see https://docs.openstack.org/pike/user/)

Procedure

1. Set a key to secure the temporary URLs. You can set keys at the account level and for a specific container, and use any one of those keys to generate temporary URLs.

- To set a key for the account, run the following cURL command:

```
curl -X POST \
     -H 'X-Auth-Token: token' \
     -H 'X-Account-Meta-Temp-URL-Key: key' \
     accountURL
```

  - token is the authentication token obtained from Oracle Cloud Infrastructure Object Storage Classic.
  - accountURL is the REST endpoint URL for your service instance.
  - key is any arbitrary string of your choice.

  Example:

```
curl -X POST \
     -H 'X-Auth-Token: AUTH_tk6b7d5b0d94e653217ee0898d43613a07' \
     -H 'X-Account-Meta-Temp-URL-Key: tempkey' \
     https://acme.storage.oraclecloud.com/v1/Storage-acme
```

- To set a key for a container, run the following cURL command:

```
curl -X POST \
     -H 'X-Auth-Token: token' \
     -H 'X-Container-Meta-Temp-URL-Key: key' \
     accountURL/containerName
```

  - token is the authentication token obtained from Oracle Cloud Infrastructure Object Storage Classic.
  - accountURL is the REST endpoint URL for your service instance.
  - key is any arbitrary string of your choice.
  - containerName is the name of the container from which you want to download objects.

  Example:

```
curl -X POST \
     -H 'X-Auth-Token: AUTH_tk6b7d5b0d94e653217ee0898d43613a07' \
     -H 'X-Container-Meta-Temp-URL-Key: tempkey1' \
     https://acme.storage.oraclecloud.com/v1/Storage-acme/myFirstContainer
```

2. You can generate temporary URLs using the OpenStack Swift CLI or using a script written in any programming language.

   To generate a temporary URL using the OpenStack Swift CLI, complete the following steps:

   a. Run the following command:

```
swift tempurl GET seconds /v1/Storage-identityDomainName/containerName/
objectName key
```

      - seconds is the number of seconds that the temporary URL should be valid from the time it is generated.
      - identityDomainName is the identity domain name of your Oracle Cloud Infrastructure Object Storage Classic account.
      - containerName is the name of the container.
      - objectName is the name of the object that needs to be downloaded using the temporary URL.

- `key` is the value that you assigned earlier to the `X-Account-Meta-Temp-URL-Key` header or `X-Container-Meta-Temp-URL-Key` header.

Example:

```
swift tempurl GET 300 /v1/Storage-acme/FirstContainer/tempobject tempkey
```

Output:

```
/v1/Storage-acme/FirstContainer/tempobject?
temp_url_sig=555417815bf1288b46c6c2ae9fc2f90a437a3110&temp_url_expi
res=1467281788
```

The output contains the path of the object (`/v1/Storage-acme/FirstContainer/tempobject`) and the following additional query parameters:

- `temp_url_sig`: A cryptographic signature of the URL
- `temp_url_expires`: The date and time when the URL will expire, in the UNIX Epoch format.

b. (Optional) If you generated a temporary URL for a container-terminated path, the URL so generated would not contain any object name. You must insert the name of the required object between the container name and "?" in the URL, as shown in the following example:

```
/v1/containerName/objectName?
temp_url_sig=value&temp_url_expires=value
```

c. Note that the temporary URL generated by the OpenStack Swift CLI is not the full URL. Construct the full URL by prefixing the account URL to the generated path, as follows:

```
https://acme.storage.oraclecloud.com/v1/Storage-acme/containerName/
objectName?temp_url_sig=value&temp_url_expires=value
```

Example:

```
https://acme.storage.oraclecloud.com/v1/Storage-acme/FirstContainer/
tempobject?
temp_url_sig=555417815bf1288b46c6c2ae9fc2f90a437a3110&temp_url_expires=146728
1788
```

3. Alternatively (instead of using the OpenStack Swift CLI), you can generate a temporary URL for an object by using an HMAC-SHA1 signature created using Python or any other programming language.

The HMAC-SHA1 signature must include the following:

- GET method
- Date and time when the URL will expire
- Key: The value that you assigned earlier to the `X-Account-Meta-Temp-URL-Key` header or `X-Container-Meta-Temp-URL-Key` header.
- Path of the object that needs to be downloaded using the temporary URL: `/v1/Storage-identityDomainName/containerName/objectName`

  Note that you can terminate the path at a container instead of specifying an object name. The resulting temporary URL can be used to download any object in the specified container. Here's the syntax for a container-terminated path: `/v1/Storage-identityDomainName/containerName`

Example:

Sample Cloud account with the following details:

- Account name: `acme`
- REST Endpoint URL: `https://acme.storage.oraclecloud.com/v1/Storage-acme`
- REST Endpoint (Permanent) URL: `https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/Storage-7b16fede61e1417ab83eb52e06f0e365`

> **Note:**
>
> The REST Endpoint (Permanent) URL is displayed for the accounts created after November 2017.

The REST Endpoint URL for the sample Cloud account is used in the examples in this section. To use the REST Endpoint (Permanent) URL, replace `https://acme.storage.oraclecloud.com/v1/Storage-acme` with `https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/Storage-7b16fede61e1417ab83eb52e06f0e365`.

The following sample Python code snippet generates a temporary URL, using input that you provide.

```
# USER INPUT: Specify the following parameters:
serviceInstanceName = 'Storage' # Leave as is unless your service
instance has a different name
identityDomainName = 'acme' # Name of your identity domain
container = 'myContainer' # Container that has the objects you need the
tempURL for
key = 'mykey' # X-Container-Meta-Temp-Url-Key or X-Account-Meta-Temp-
Url-Key value
object = 'myObject' # Object name that you need the tempURL for. This
is optional if a container-level key is used.
urlDuration = 300 # Seconds for which the temp URL should work
serviceRestEndpoint = 'https://acme.storage.oraclecloud.com/v1/Storage-
acme' # REST endpoint URL of your service instance

# CODE TO GENERATE THE TEMPORARY URL: Don't change anything in this
section
import hmac
from hashlib import sha1
from time import time
path = '/v1/' + serviceInstanceName + '-' + identityDomainName + '/' +
container + '/' + object
expires = int(time() + urlDuration)
hmac_body = '%s\n%s\n%s' % ('GET', expires, path)
sig = hmac.new(key, hmac_body, sha1).hexdigest()
url = serviceRestEndpoint + '/' + container + '/' + object +'?
temp_url_sig=' + sig + '&temp_url_expires=' + str(expires)
print(url)
```

a. Copy this Python code snippet to a plain text file.

b. In the `USER INPUT` section of the code, enter appropriate values for all the parameters, as described in the comments in the code.

c. Save the file, and note the file name (say, `tempURL.py`).

d. Run the Python script:
`python tempURL.py`

The script generates and displays the temporary URL for the object or container that you specified.

Example:

- Temporary URL for an object: `https://acme.storage.oraclecloud.com/v1/Storage-acme/containerName/objectName?temp_url_sig=value&temp_url_expires=value`
- Temporary URL for a container: `https://acme.storage.oraclecloud.com/v1/containerName?temp_url_sig=value&temp_url_expires=value`

> **Note:**
>
> To use the container-terminated temporary URL to download an object, you must insert the object name between the container name and "?" in the URL, as shown in the following example: `https://acme.storage.oraclecloud.com/v1/Storage-acme/containerName/objectName?temp_url_sig=value&temp_url_expires=value`

4. Give the temporary URL to the users who need to download the object.

Deleting Objects

When an object is deleted, the object and its metadata will be removed permanently.

Topics:

- Deleting a Single Object
- Deleting a Large Object
- Deleting Multiple Objects in a Single Operation

Deleting a Single Object

Any user with the Service Administrator role or a role that is specified in the `X-Container-Write` ACL of the container can perform this task.

You can delete an object in a container by using the following interfaces:

Interface	Resources
Web Console (Not available on Oracle Cloud at Customer)	See Deleting a Single Object Using the Web Console.
RESTful API	See Delete Object in *REST API for Standard Storage in Oracle Cloud Infrastructure Object Storage Classic*.
Java Library	See `deleteObject` in *Java API Reference for Oracle Cloud Infrastructure Object Storage Classic*.
File Transfer Manager API	See deleteObject in *Java API Reference for Oracle Cloud Infrastructure Object Storage Classic File Transfer Manager*.
File Transfer Manager CLI	See Deleting Objects in *Command-Line Reference for Oracle Cloud Infrastructure Object Storage Classic*.

> **Note:**
>
> You can schedule deletion of objects at a specified time in the future or after a specified period of time has elapsed, by using the `X-Delete-At` or `X-Delete-After` header. See Scheduling Automatic Deletion of Objects.

Deleting a Single Object Using the Web Console

(Not available on Oracle Cloud at Customer)

1. Sign in to the Oracle Cloud Infrastructure Object Storage Classic console.
2. Select the container that has the object you want to delete.
   The objects in the container are displayed.
3. Identify the object that you want to delete.
4. From the **Actions** drop-down list for the object, select **Delete**.
   The following message appears:

   ```
   Are you sure want to delete this object?
   ```

5. Click **OK**.
   The object is deleted from the container.

Deleting a Single Object Using the REST API

cURL Command Syntax

```
curl -v -X DELETE \
     -H "X-Auth-Token: token" \
     accountURL/containerName/objectName
```

- `token` is the authentication token obtained earlier from Oracle Cloud Infrastructure Object Storage Classic. See Authenticating Access When Using the REST API.
- For the syntax of `accountURL`, see About REST URLs for Oracle Cloud Infrastructure Object Storage Classic Resources.
- `containerName` is the name of the container that contains the object to be deleted.
- `objectName` is the name of the object to be deleted.

> **Note:**
>
> When you send a REST API request to Oracle Cloud Infrastructure Object Storage Classic, all non-ASCII characters in container names, object names and metadata values must be URL-encoded. For example, `my container` should be encoded as `my%20container`, where `%20` is the HTML encoding for the space character. Similarly, `my Über Container` should be encoded as `my %20%C3%9Cber%20Container`, where `%20` represents the space character and `%C3%9C` is the Ü character.

HTTP Response Codes

- Success: `204 No Content`
- Failure: See Error Code Reference for Oracle Cloud Infrastructure Object Storage Classic

cURL Command Examples

Sample Cloud account with the following details:

- Account name: `acme`
- REST Endpoint URL: `https://acme.storage.oraclecloud.com/v1/Storage-acme`
- REST Endpoint (Permanent) URL: `https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/Storage-7b16fede61e1417ab83eb52e06f0e365`

> **Note:**
>
> The REST Endpoint (Permanent) URL is displayed for the accounts created after November 2017.

- Using the REST Endpoint URL obtained from the **REST Endpoint** field:

```
curl -v -X DELETE \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     https://acme.storage.oraclecloud.com/v1/Storage-acme/FirstContainer/
myObject2
```

The following is an example of the output of this command:

```
> DELETE /v1/Storage-acme/FirstContainer/myObject2 HTTP/1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.14.0.0
zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: acme.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b
>
< HTTP/1.1 204 No Content
< Date: Mon, 09 Mar 2015 11:40:23 GMT
< Content-Length: 0
< Content-Type: text/html; charset=UTF-8
< X-Trans-Id: tx47aef42f16c44bd9a72cb-0054fd86a7
< Cache-Control: no-cache
```

```
< Pragma: no-cache
< Content-Language: en
```

- Using the Service Permanent REST Endpoint URL obtained from the **REST Endpoint (Permanent)** field:

> **Note:**
>
> This cURL command example applies to the accounts created after November 2017.

```
curl -v -X DELETE \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/
Storage-7b16fede61e1417ab83eb52e06f0e365/FirstContainer/myObject2
```

The following is an example of the output of this command:

```
> DELETE /v1/Storage-7b16fede61e1417ab83eb52e06f0e365/FirstContainer/myObject2
HTTP/1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.14.0.0
zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b
>
< HTTP/1.1 204 No Content
< Date: Mon, 09 Mar 2015 11:40:23 GMT
< Content-Length: 0
< Content-Type: text/html; charset=UTF-8
< X-Trans-Id: tx47aef42f16c44bd9a72cb-0054fd86a7
< Cache-Control: no-cache
< Pragma: no-cache
< Content-Language: en
```

Deleting a Large Object

Large objects contain data more than 5 GB.

cURL Command Syntax

To delete a static large object, send a `DELETE` request and add the query parameter `?multipart-manifest=delete` to delete the manifest object and all the segment files.

> **Note:**
>
> To delete a static large object from an `Archive` container, delete each segment file and the manifest object individually. (Not available on Oracle Cloud at Customer)

```
curl -v -X DELETE \
     -H "X-Auth-Token: token" \
     accountURL/containerName/manifestFile?multipart-manifest=delete \
```

- `token` is the authentication token obtained earlier from Oracle Cloud Infrastructure Object Storage Classic. See Authenticating Access When Using the REST API.
- For the syntax of `accountURL`, see About REST URLs for Oracle Cloud Infrastructure Object Storage Classic Resources.
- `containerName` is the name of the container that contains the manifest file to be deleted.
- `manifestFile` is the name of the manifest file.
- `?multipart-manifest=delete` is the query parameter to delete the static large object.

> **Note:**
>
> When you send a REST API request to Oracle Cloud Infrastructure Object Storage Classic, all non-ASCII characters in container names, object names and metadata values must be URL-encoded. For example, `my container` should be encoded as `my%20container`, where `%20` is the HTML encoding for the space character. Similarly, `my Über Container` should be encoded as `my %20%C3%9Cber%20Container`, where `%20` represents the space character and `%C3%9C` is the Ü character.

cURL Command Examples

The following are cURL command examples to delete a static large object.

Sample Cloud account with the following details:

- Account name: `acme`
- REST Endpoint URL: `https://acme.storage.oraclecloud.com/v1/Storage-acme`
- REST Endpoint (Permanent) URL: `https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/Storage-7b16fede61e1417ab83eb52e06f0e365`

> **Note:**
>
> The REST Endpoint (Permanent) URL is displayed for the accounts created after November 2017.

- Using the REST Endpoint URL obtained from the **REST Endpoint** field:

```
curl -v -X DELETE \
        -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
        https://acme.storage.oraclecloud.com/v1/Storage-acme/FirstContainer/
myLargeFile.manifest?multipart-manifest=delete \
```

The following is an example of the output of this command:

```
> DELETE /v1/Storage-acme/myLargeFile.manifest?multipart-manifest=delete
> User-Agent: curl/7.29.0
> Host: acme.storage.oraclecloud.com
```

```
> Accept: */*
> X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b
>
< HTTP/1.1 204 No Content
< X-Trans-Id: txa590ad2c5ce54317bd02e-0056a8a5d5ga
< Date: Wed, 20 Jan 2016 11:11:17 GMT
< Connection: keep-alive
< X-Last-Modified-Timestamp: 1453893077.47202
< Content-Type: text/html;charset=UTF-8
<
```

- Using the Service Permanent REST Endpoint URL obtained from the **REST Endpoint (Permanent)** field:

  > **Note:**
  >
  > This cURL command example applies to the accounts created after November 2017.

```
curl -v -X DELETE \
        -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
        https://
storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/
Storage-7b16fede61e1417ab83eb52e06f0e365/FirstContainer/myLargeFile.manifest?
multipart-manifest=delete \
```

The following is an example of the output of this command:

```
> DELETE /v1/Storage-7b16fede61e1417ab83eb52e06f0e365/myLargeFile.manifest?
multipart-manifest=delete
> User-Agent: curl/7.29.0
> Host: storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b
>
< HTTP/1.1 204 No Content
< X-Trans-Id: txa590ad2c5ce54317bd02e-0056a8a5d5ga
< Date: Wed, 20 Jan 2016 11:11:17 GMT
< Connection: keep-alive
< X-Last-Modified-Timestamp: 1453893077.47202
< Content-Type: text/html;charset=UTF-8
<
```

Deleting Multiple Objects in a Single Operation

Multiple objects can be deleted in a single operation.

> **Note:**
>
> - This feature is not supported for `Archive` containers.
> - You can delete up to 10,000 objects in a single operation.

Any user with the Service Administrator role or a role that is specified in the `X-Container-Write` ACL of the container can perform this task.
To bulk-delete objects:

Sample Cloud account with the following details –

- Account name: `acme`
- REST Endpoint URL: `https://acme.storage.oraclecloud.com/v1/Storage-acme`
- REST Endpoint (Permanent) URL: `https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/Storage-7b16fede61e1417ab83eb52e06f0e365`

> **Note:**
>
> The REST Endpoint (Permanent) URL is displayed for the accounts created after November 2017.

The REST Endpoint URL for the sample Cloud account is used in the steps below. To use the REST Endpoint (Permanent) URL, replace `https://acme.storage.oraclecloud.com/v1/Storage-acme` with `https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/Storage-7b16fede61e1417ab83eb52e06f0e365`.

1. Create a plain text file that lists each object you want to delete.

   Each line in the plain-text file should be in the format *`container/object`*, as shown in the following example:

   ```
   $ cat objects_to_delete.txt

   myContainer1/myObject1
   myContainer1/myObject2
   myContainer2/myObjectN
   myContainer3/myObjectX
   ```

   > **Note:**
   >
   > The plain-text file that you create can list objects in multiple containers.

   To quickly build such a file, send a `GET` call to the container that has the objects you want to delete, pipe the output of the `GET` call to a `sed` command that prefixes the container name to each object name, and then redirect the edited output to a text file.

   Example command:

   ```
   $ curl -X GET
          -H "X-Auth-Token: AUTH_tkcc14bdc52d571f70991b68907098f831"
          "https://acme.storage.oraclecloud.com/v1/Storage-acme/myContainer"
          | sed 's/^/myContainer\//'
          > objects_to_delete.txt
   ```

   - `myContainer` is the name of the container that has the objects to be deleted. Replace this with the name of your container.

- `objects_to_delete` is the plain text file in which the objects that must be deleted are listed.
- Note the 'escaped' forward slash (`\/`) that's added after the container name. It's necessary to demarcate the container and object name in each line in the text file.

2. Send a `DELETE` request to the account URL.
   - Set `Content-Type` to `text/plain`.
   - Include the `bulk-delete` parameter.

   To delete the objects listed in the file `objects_to_delete.txt` that you created in the previous step:

```
$ curl -v -s -X DELETE
       -H "X-Auth-Token: AUTH_tkcc14bdc52d571f70991b68907098f831"
       -H "Content-Type: text/plain"
       -T objects_to_delete.txt
       "https://acme.storage.oraclecloud.com/v1/Storage-acme/?bulk-delete"
```

   The following is an example of the output of this command

```
> DELETE /v1/Storage-acme/?bulk-delete HTTP/1.1
> Host: acme.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tkcc14bdc52d571f70991b68907098f831
> Content-Type: text/plain
> Content-Length: 75

< HTTP/1.1 200 OK
< Date: Thu, 12 Dec 2013 14:56:53 GMT
< Content-Type: text/plain
< Transfer-Encoding: chunked

Number Deleted: 3
Number Not Found: 0
Response Body:
Response Status: 200 OK
Errors:
```

> **Note:**
>
> - For the result of the operation, look at the response body (not the returned status code).
> - In the response body, if `Number Deleted` doesn't match the count of lines in the text file that you specified, then some objects weren't found or deleted. Check for the objects that remain in the container and delete them as required.

You can delete multiple objects in a single operation by using the following interfaces:

Interface	Resources
RESTful API	See Bulk Operations in *OpenStack Object Storage Service API Reference*.

Interface	Resources
File Transfer Manager API	See deleteObjects in *Java API Reference for Oracle Cloud Infrastructure Object Storage Classic File Transfer Manager*.

Copying Objects

An object can be copied to another object within the same or another container. There is no need to download the object and then upload it again; the copying operation is performed entirely on the server.

A user with any of the following roles can perform this task:

- Service Administrator role
- A role that is specified in the `X-Container-Read` ACL of the source container and in the `X-Container-Write` ACL of the target container

The following restrictions apply:

- You can copy static large objects between containers. You can copy the segments directly, but you must re-create the manifest. You cannot use the existing manifest as it points to the old segments, which may have been deleted.
- You cannot copy objects to other accounts.

You can copy objects by using the following interfaces:

Interface	Resources
RESTful API	See Copy Object in *OpenStack Object Storage Service API Reference*
File Transfer Manager CLI	See Copying an Object in n *Command-Line Reference for Oracle Cloud Infrastructure Object Storage Classic*.
File Transfer Manager API	See `CopyObjectRequestConfig` in *Java API Reference for Oracle Cloud Infrastructure Object Storage Classic File Transfer Manager*.

cURL Command Syntax

```
curl -v -X PUT \
     -H "X-Auth-Token: token" \
     -H "X-Copy-From: /srcContainer/srcObject" \
     -H "Content-Length: 0" \
     accountURL/dstContainer/dstObject
```

or

```
curl -v -X COPY \
     -H "X-Auth-Token: token" \
     -H "Destination: /dstContainer/dstObject" \
     accountURL/srcContainer/srcObject
```

- `token` is the authentication token obtained earlier from Oracle Cloud Infrastructure Object Storage Classic. See Authenticating Access When Using the REST API.
- `srcContainer` is the name of the container that contains the object to be copied.
- `srcObject` is the name of the object to be copied.

- For the syntax of `accountURL`, see About REST URLs for Oracle Cloud Infrastructure Object Storage Classic Resources.
- `dstContainer` is the name of the container to which the object should be copied. This should be an existing container. In other words, the `COPY` operation won't create the destination container.
- `dstObject` is the name of the object to be created as a result of the `COPY` operation.

HTTP Response Codes

- Success: `201 Created`
- Failure: See Error Code Reference for Oracle Cloud Infrastructure Object Storage Classic

cURL Command Examples

The following commands copy `/container1/object1` to `/container2/object1`.

Sample Cloud account with the following details:

- Account name: `acme`
- REST Endpoint URL: `https://acme.storage.oraclecloud.com/v1/Storage-acme`
- REST Endpoint URL (Permanent) URL: `https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/Storage-7b16fede61e1417ab83eb52e06f0e365`

> **Note:**
>
> The REST Endpoint URL (Permanent) URL is displayed for the accounts created after November 2017.

- Using the REST Endpoint URL obtained from the **REST Endpoint** field:

```
curl -v -X PUT \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     -H "X-Copy-From: /container1/object1" \
     -H "Content-Length: 0"
     https://acme.storage.oraclecloud.com/v1/Storage-acme/container2/object1
```

or

```
curl -v -X COPY \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     -H "Destination: /container2/object1" \
     https://acme.storage.oraclecloud.com/v1/Storage-acme/container1/object1
```

The following is an example of the output of this command:

```
> PUT /v1/Storage-acme/Container2/object1 HTTP/1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.14.0.0
zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: acme.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b
> X-Copy-From: /Container1/object1
```

```
> Content-Length: 0
>
< HTTP/1.1 201 Created
< Date: Mon, 09 Mar 2015 11:58:23 GMT
< Content-Length: 0
< X-Object-Meta-Language: english
< X-Copied-From-Last-Modified: Mon, 09 Mar 2015 11:46:34 GMT
< X-Copied-From: Container1/object1
< Last-Modified: Mon, 09 Mar 2015 11:58:24 GMT
< Etag: 846fa9d298be05e5f598703f0c3d6f51
< Content-Type: text/html; charset=UTF-8
< X-Trans-Id: txecc0da22db8542c099ed0-0054fd8ade
< Cache-Control: no-cache
< Pragma: no-cache
< Content-Language: en
```

- Using the Service Permanent REST Endpoint URL obtained from the **REST Endpoint (Permanent)** field:

> **Note:**
>
> This cURL command example applies to the accounts created after November 2017.

```
curl -v -X PUT \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     -H "X-Copy-From: /container1/object1" \
     -H "Content-Length: 0"
     https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/
Storage-7b16fede61e1417ab83eb52e06f0e365/container2/object1
```

or

```
curl -v -X COPY \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     -H "Destination: /container2/object1" \
     https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/
Storage-7b16fede61e1417ab83eb52e06f0e365/container1/object1
```

The following is an example of the output of this command:

```
> PUT /v1/Storage-7b16fede61e1417ab83eb52e06f0e365/Container2/object1 HTTP/1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.14.0.0
zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b
> X-Copy-From: /Container1/object1
> Content-Length: 0
>
< HTTP/1.1 201 Created
< Date: Mon, 09 Mar 2015 11:58:23 GMT
< Content-Length: 0
< X-Object-Meta-Language: english
< X-Copied-From-Last-Modified: Mon, 09 Mar 2015 11:46:34 GMT
< X-Copied-From: Container1/object1
< Last-Modified: Mon, 09 Mar 2015 11:58:24 GMT
< Etag: 846fa9d298be05e5f598703f0c3d6f51
< Content-Type: text/html; charset=UTF-8
```

```
< X-Trans-Id: txecc0da22db8542c099ed0-0054fd8ade
< Cache-Control: no-cache
< Pragma: no-cache
< Content-Language: en
```

Encrypting Objects

You can encrypt objects by using the following interfaces:

Interface	Resources
Java Library	See: • Using the Java Library • `EncryptedCloudStorage` in *Java API Reference for Oracle Cloud Infrastructure Object Storage Classic*.
File Transfer Manager API	See `EncryptedFileTransferManager` in *Java API Reference for Oracle Cloud Infrastructure Object Storage Classic File Transfer Manager*.
File Transfer Manager CLI	See: • Using FTM CLI • About Client-Side File Encryption in *Command-Line Reference for Oracle Cloud Infrastructure Object Storage Classic*.

Using the Java Library

The Java library can encrypt objects as they are being stored to Oracle Cloud Infrastructure Object Storage Classic. All encryption happens within the Java library; no encryption happens within the service. The REST API does not support this.

Any user with the Service Administrator role or a role that is specified in the `X-Container-Write` ACL of the container can perform this task.

When you use the client-side encryption feature of the Java library, for every object that you create in Oracle Cloud Infrastructure Object Storage Classic, a unique symmetric key is generated. The Java library uses the symmetric key to encrypt your data before storing it. In addition, you must provide and manage an asymmetric key pair. After encrypting your data, the Java library encrypts the symmetric key as an envelope key by using the asymmetric key pair. Note that you can rotate a previously used key pair for a new key pair without downloading and re-encrypting each object. The envelope key is stored as metadata alongside the object data.

When you use the Java library to access such encrypted objects, the envelope key is first retrieved and decrypted by using the asymmetric key pair that you provide. The resulting symmetric key is then used to decrypt the object data.

You can restrict access to client-side encrypted objects. For example, if you have write access to a container but not to the asymmetric key pair, you can remove the envelope key metadata from an object. So the object data becomes unrecoverable, even if you have access to the asymmetric key pair.

When using the Java library's encryption feature, note the following:

- Only 2048 bit RSA key pairs are supported.
- Only object data is encrypted, not object metadata.
- Segmented objects cannot be encrypted.

- Nonencrypted objects cannot be downloaded while using the encryption feature.

For more information, see `EncryptedCloudStorage` in *Java API Reference for Oracle Cloud Infrastructure Object Storage Classic*.

Using FTM CLI

The FTM CLI enables client-side encryption of data that you want to store on Oracle Cloud Infrastructure Object Storage Classic. The FTM CLI on the client encrypts files before they are transferred to the cloud service. Files are encrypted when they are transferred to the cloud as well as when they are at rest on the cloud. The FTM CLI allows client-side encryption and decryption of static large objects (SLOs) but not dynamic large objects (DLOs). See About Client-Side File Encryption in *Command-Line Reference for Oracle Cloud Infrastructure Object Storage Classic*.

- To generate a master key, see Generating Master Key.
- To rotate the master key, see Rotating Master Key.
- To encrypt a file, see Uploading Files.
- To decrypt an object, see Downloading an Object.

Updating Object Metadata

Topics:

- Updating Custom Metadata for Objects
- Scheduling Automatic Deletion of Objects

Updating Custom Metadata for Objects

Custom metadata are arbitrary key-value pairs. You may define and update any custom or arbitrary metadata that you need.

Any user with the Service Administrator role or a role that is specified in the `X-Container-Write` ACL of the container can perform this task.

The service transforms custom metadata keys as follows:

- Underscores are converted to hyphens.
- The first character after a hyphen is capitalized. All other letters are converted to lowercase.
- The Java library automatically prefixes each key with `X-Object-Meta-Custom`.

You can update custom metadata for objects by using the following interfaces:

Interface	Resources
RESTful API	See Create or Update Object Metadata in *REST API for Standard Storage in Oracle Cloud Infrastructure Object Storage Classic*.
Java Library	See `updateObjectMetadata` in *Java API Reference for Oracle Cloud Infrastructure Object Storage Classic*.
File Transfer Manager API	See class: ObjectMetadata in *Java API Reference for Oracle Cloud Infrastructure Object Storage Classic File Transfer Manager*.

Interface	Resources
File Transfer Manager CLI	See Setting Object Metadata in *Command-Line Reference for Oracle Cloud Infrastructure Object Storage Classic*.

You can also delete custom metadata for objects by sending a `POST` request. See Deleting Object Metadata.

cURL Command Syntax

```
curl -v -X POST \
     -H "X-Auth-Token: token" \
     -H "X-Object-Meta-Name-1: value-1" \
     -H "X-Object-Meta-Name-2: value-2" \
     accountURL/containerName/objectName
```

- `token` is the authentication token obtained earlier from Oracle Cloud Infrastructure Object Storage Classic. See Authenticating Access When Using the REST API.
- `Name` and `value` are the metadata header and value to be updated.
  You can specify multiple metadata headers and values that you would like to add or update.

> **Note:**
>
> Ensure that the custom metadata name and value comply with the input restrictions mentioned in Character Restrictions.

- For the syntax of `accountURL`, see About REST URLs for Oracle Cloud Infrastructure Object Storage Classic Resources.
- `containerName` is the name of the container that contains the object for which custom metadata should be updated.
- `objectName` is the name of the object for which custom metadata should be updated.

> **Note:**
>
> When you send a REST API request to Oracle Cloud Infrastructure Object Storage Classic, all non-ASCII characters in container names, object names and metadata values must be URL-encoded. For example, `my container` should be encoded as `my%20container`, where `%20` is the HTML encoding for the space character. Similarly, `my Über Container` should be encoded as `my %20%C3%9Cber%20Container`, where `%20` represents the space character and `%C3%9C` is the Ü character.

HTTP Response Codes

- Success: `202 Accepted`
- Failure: See Error Code Reference for Oracle Cloud Infrastructure Object Storage Classic

cURL Command Examples

Sample Cloud account with the following details:

- Account name: `acme`
- REST Endpoint URL: `https://acme.storage.oraclecloud.com/v1/Storage-acme`
- REST Endpoint (Permanent) URL: `https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/Storage-7b16fede61e1417ab83eb52e06f0e365`

> **Note:**
>
> The REST Endpoint (Permanent) URL is displayed for the accounts created after November 2017.

- Using the REST Endpoint URL obtained from the **REST Endpoint** field:

```
curl -v -X POST \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     -H "X-Object-Meta-Language: english" \
     -H "X-Object-Meta-Country: US" \
     https://acme.storage.oraclecloud.com/v1/Storage-acme/FirstContainer/myObject
```

The following is an example of the output of this command. Using the above command, the values of the `X-Object-Meta-Language` and `X-Object-Meta-Country` metadata headers are updated to `english` and `US` respectively.

```
> POST /v1/Storage-acme/FirstContainer/myObject HTTP/1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.14.0.0
zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: acme.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b
> X-Object-Meta-Language: english
> X-Object-Meta-Country: US
>
< HTTP/1.1 202 Accepted
< Date: Mon, 09 Mar 2015 11:46:34 GMT
< Content-Length: 76
< Content-Type: text/html; charset=UTF-8
< X-Trans-Id: txd54813b92dcd46849b009-0054fd8819
< Cache-Control: no-cache
< Pragma: no-cache
< Content-Language: en
<
Accepted. The request is accepted for processing.
```

- Using the Service Permanent REST Endpoint URL obtained from the **REST Endpoint (Permanent)** field:
  This cURL command example applies to the accounts created after November 2017.

```
curl -v -X POST \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     -H "X-Object-Meta-Language: english" \
     -H "X-Object-Meta-Country: US" \
```

```
    https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/
Storage-7b16fede61e1417ab83eb52e06f0e36/FirstContainer/myObject
```

The following is an example of the output of this command. Using the above command, the values of the `X-Object-Meta-Language` and `X-Object-Meta-Country` metadata headers are updated to `english` and `US` respectively.

```
> POST /v1/Storage-7b16fede61e1417ab83eb52e06f0e36/FirstContainer/myObject HTTP/
1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.14.0.0
zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: storage-7b16fede61e1417ab83eb52e06f0e36.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b
> X-Object-Meta-Language: english
> X-Object-Meta-Country: US
>
< HTTP/1.1 202 Accepted
< Date: Mon, 09 Mar 2015 11:46:34 GMT
< Content-Length: 76
< Content-Type: text/html; charset=UTF-8
< X-Trans-Id: txd54813b92dcd46849b009-0054fd8819
< Cache-Control: no-cache
< Pragma: no-cache
< Content-Language: en
<
Accepted. The request is accepted for processing.
```

Deleting Object Metadata

To delete **all** the custom metadata for an object, send a `POST` request without specifying any metadata header. All the custom metadata is deleted. Note that the system metadata is retained.

To delete specific custom metadata, send a `POST` request and specify only the metadata headers and their values that you'd like to retain or update. Do **not** specify the metadata that must be deleted. Any metadata headers that are specified in the request are updated. The headers that aren't specified in the request are deleted.

cURL Command Examples

The following command examples shows how to delete all the custom metadata of the object `myObject`.

Sample Cloud account with the following details:

- Account name: `acme`
- REST Endpoint URL: `https://acme.storage.oraclecloud.com/v1/Storage-acme`
- REST Endpoint (Permanent) URL: `https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/Storage-7b16fede61e1417ab83eb52e06f0e365`

> **Note:**
>
> The REST Endpoint (Permanent) URL is displayed for the accounts created after November 2017.

- Using the REST Endpoint URL obtained from the **REST Endpoint** field:

```
curl -v -X POST \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     https://acme.storage.oraclecloud.com/v1/Storage-acme/FirstContainer/myObject
```

  The following is an example of the output of this command:

```
> POST /v1/Storage-acme/FirstContainer/myObject HTTP/1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.14.0.0
zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: acme.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b
>
< HTTP/1.1 202 Accepted
< Date: Mon, 22 Dec 2016 11:46:34 GMT
< Content-Length: 76
< Content-Type: text/html; charset=UTF-8
< X-Trans-Id: txd54813b92dcd46849b009-0054fd8819
< Cache-Control: no-cache
< Pragma: no-cache
< Content-Language: en
<
Accepted. The request is accepted for processing.
```

- Using the Service Permanent REST Endpoint URL obtained from the **REST Endpoint (Permanent)** field:
  This cURL command example applies to the accounts created after November 2017.

- ```
  curl -v -X POST \
       -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
       https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/
  Storage-7b16fede61e1417ab83eb52e06f0e365/FirstContainer/myObject
  ```

  The following is an example of the output of this command:

```
> POST /v1/Storage-7b16fede61e1417ab83eb52e06f0e365/FirstContainer/myObject HTTP/
1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.14.0.0
zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b
>
< HTTP/1.1 202 Accepted
< Date: Mon, 22 Dec 2016 11:46:34 GMT
< Content-Length: 76
< Content-Type: text/html; charset=UTF-8
< X-Trans-Id: txd54813b92dcd46849b009-0054fd8819
< Cache-Control: no-cache
< Pragma: no-cache
< Content-Language: en
<
Accepted. The request is accepted for processing.
```
```

Scheduling Automatic Deletion of Objects

You can schedule deletion of objects at a specified time in the future or after a specified period of time has elapsed, by using the `X-Delete-After` or `X-Delete-At` header, respectively.

> **Note:**
>
> You cannot schedule automatic deletion of objects for an `Archive` container by using the `X-Delete-After` and `X-Delete-At` headers. (Not available on Oracle Cloud at Customer)

Any user with the Service Administrator role or a role that is specified in the `X-Container-Write` ACL of the container can perform this task.

To schedule automatic deletion while creating an object, include the `X-Delete-After` or `X-Delete-At` header in the `PUT` request. To schedule automatic deletion for an existing object, use the `POST` request. The Java library does not support scheduling automatic deletion of objects.

cURL Command Syntax

To schedule automatic deletion of an existing object after a specified duration:

```
curl -v -X POST \
     -H "X-Auth-Token: token" \
     -H "X-Delete-After: period" \
     accountURL/containerName/objectName
```

To schedule automatic deletion of an existing object at a specified time in the future:

```
curl -v -X POST \
     -H "X-Auth-Token: token" \
     -H "X-Delete-At: time" \
     accountURL/containerName/objectName
```

- `token` is the authentication token obtained earlier from Oracle Cloud Infrastructure Object Storage Classic. See Authenticating Access When Using the REST API.
- `period` is the duration, in seconds, after which the object should be deleted.
- `time` is the UNIX Epoch timestamp representing the date and time at which the object should be deleted. For example, `1416218400` represents November 17, 2014 10:00:00 GMT. See http://www.epochconverter.com/.
- For the syntax of `accountURL`, see About REST URLs for Oracle Cloud Infrastructure Object Storage Classic Resources.
- `containerName` is the name of the container that contains the object for which automatic deletion should be scheduled.
- `objectName` is the name of the object for which automatic deletion should be scheduled.

cURL Command Examples

Sample Cloud account with the following details:

- Account name: `acme`
- REST Endpoint URL: `https://acme.storage.oraclecloud.com/v1/Storage-acme`
- REST Endpoint (Permanent) URL: `https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/Storage-7b16fede61e1417ab83eb52e06f0e365`

> **Note:**
>
> The REST Endpoint (Permanent) URL is displayed for the accounts created after November 2017.

The REST Endpoint URL for the sample Cloud account is used in the cURL command examples below. To use the REST Endpoint (Permanent) URL, replace `https://acme.storage.oraclecloud.com/v1/Storage-acme` with `https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/Storage-7b16fede61e1417ab83eb52e06f0e365`.

- The following command sets the object named `myObject` to be deleted automatically after 86400 seconds:

```
curl -v -X POST \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     -H "X-Delete-After: 86400" \
     https://acme.storage.oraclecloud.com/v1/Storage-acme/myContainer/myObject
```

  The following is an example of the output of this command:

```
> POST /v1/Storage-acme/myContainer/myObject HTTP/1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.14.0.0
zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: acme.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b
> X-Delete-After: 86400
>
< HTTP/1.1 202 Accepted
< Date: Mon, 23 Mar 2015 12:32:39 GMT
< Content-Length: 76
< Content-Type: text/html; charset=UTF-8
< X-Trans-Id: txbb5a2f22164e47aa8116f-00551007e7
< Cache-Control: no-cache
< Pragma: no-cache
< Content-Language: en
<
The request is accepted for processing.
```

  The following command sets the object named `myObject` to be deleted automatically on November 30, 2014 at 10:00:00 GMT, represented by the UNIX Epoch timestamp, 1417341600:

```
curl -v -X POST \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
```

```
    -H "X-Delete-At: 1417341600" \
    https://acme.storage.oraclecloud.com/v1/Storage-acme/myContainer/myObject
```

The following is an example of the output of this command:

```
> POST /v1/Storage-acme/myContainer/myObject HTTP/1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.14.0.0
zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: acme.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b
> X-Delete-At: 1417341600
>
< HTTP/1.1 202 Accepted
< Date: Mon, 23 Mar 2015 12:32:39 GMT
< Content-Length: 76
< Content-Type: text/html; charset=UTF-8
< X-Trans-Id: txbb5a2f22164e47aa8116f-00551007e7
< Cache-Control: no-cache
< Pragma: no-cache
< Content-Language: en
<
The request is accepted for processing.
```

4

Managing Your Object Storage Classic Account

This section provides documentation on how to set and view metadata for your Oracle Cloud Infrastructure Object Storage Classic account.

Topics

- Setting Account Metadata
- Getting Account Metadata
- Deleting Account Metadata
- Enabling Audit Logging

Setting Account Metadata

Custom metadata are arbitrary key-value pairs associated with an account. You may create any custom or arbitrary metadata you need.

Any user with the Service Administrator role can perform this task.

You can set the account metadata by using the following interfaces:

Interface	Resources
Web Console (Not available on Oracle Cloud at Customer)	See Setting Account Metadata Using the Web Console.
RESTful API	See Create, Update or Delete Account Metadata in *REST API for Standard Storage in Oracle Cloud Infrastructure Object Storage Classic*.
File Transfer Manager API	See class: AccountMetadata in *Java API Reference for Oracle Cloud Infrastructure Object Storage Classic File Transfer Manager*.
File Transfer Manager CLI	See Setting Account Metadata in *Command-Line Reference for Oracle Cloud Infrastructure Object Storage Classic*.

Setting Account Metadata Using the Web Console

(Not available on Oracle Cloud at Customer)

1. Sign in to the Oracle Cloud Infrastructure Object Storage Classic console.
2. Expand **Account Information**.
3. Click **Edit**.
   In the **Account Information** pane, look for **Custom Metadata**.
4. Click **Add Metadata**.

5. Enter the metadata name and value.

> **Note:**
>
> Ensure that the metadata name and value comply with the input restrictions mentioned in Character Restrictions.

6. Click **Save**.

Setting Account Metadata Using the REST API

cURL Command Syntax

```
curl -v -X POST \
     -H "X-Auth-Token: token" \
     -H "X-Account-Meta-Name: value" \
     accountURL
```

- `token` is the authentication token obtained earlier from Oracle Cloud Infrastructure Object Storage Classic. See Authenticating Access When Using the REST API.
- `Name` and `value` are the metadata key and value to be created.

> **Note:**
>
> Ensure that the metadata name and value comply with the input restrictions mentioned in Character Restrictions.

- For the syntax of `accountURL`, see About REST URLs for Oracle Cloud Infrastructure Object Storage Classic Resources.

HTTP Response Codes

- Success: `204 No Content`
- Failure: See Error Code Reference for Oracle Cloud Infrastructure Object Storage Classic

cURL Command Examples

Sample Cloud account with the following details:

- Account name: `acme`
- REST Endpoint URL: `https://acme.storage.oraclecloud.com/v1/Storage-acme`
- REST Endpoint (Permanent) URL: `https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/Storage-7b16fede61e1417ab83eb52e06f0e365`

> **Note:**
>
> The REST Endpoint (Permanent) URL is displayed for the accounts created after November 2017.

- Using the REST Endpoint URL obtained from the **REST Endpoint** field:

```
curl -v -X POST \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     -H "X-Account-Meta-Owner: IT" \
     https://acme.storage.oraclecloud.com/v1/Storage-acme
```

The following is an example of the output of this command:

```
> POST /v1/Storage-acme HTTP/1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.14.0.0
zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: acme.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b
> X-Account-Meta-Owner: IT
>
< HTTP/1.1 204 No Content
< Date: Fri, 06 Mar 2015 11:44:29 GMT
< Content-Length: 0
< Content-Type: text/html; charset=UTF-8
< X-Trans-Id: tx0742dd38e3a445529860a-0054f9931d
< Cache-Control: no-cache
< Pragma: no-cache
< Content-Language: en
<
```

- Using the Service Permanent REST Endpoint URL obtained from the **REST Endpoint (Permanent)** field:

> **Note:**
>
> This cURL command example applies to the accounts created after November 2017.

```
curl -v -X POST \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     -H "X-Account-Meta-Owner: IT" \
     https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/
Storage-7b16fede61e1417ab83eb52e06f0e365
```

The following is an example of the output of this command:

```
> POST /v1/Storage-7b16fede61e1417ab83eb52e06f0e365 HTTP/1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.14.0.0
zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b
> X-Account-Meta-Owner: IT
>
< HTTP/1.1 204 No Content
< Date: Fri, 06 Mar 2015 11:44:29 GMT
< Content-Length: 0
< Content-Type: text/html; charset=UTF-8
< X-Trans-Id: tx0742dd38e3a445529860a-0054f9931d
< Cache-Control: no-cache
< Pragma: no-cache
< Content-Language: en
<
```

For information about deleting account metadata, see Deleting Account Metadata.

For information about viewing account metadata, see Getting Account Metadata.

Getting Account Metadata

Any user within the identity domain can perform this task.

You can retrieve the following information about your account:

- Replication Policy (`X-Account-Meta-Policy-Georeplication`)
- Support for `archive` containers (`X-Account-Meta-Policy-Archive`)
- Account quota, in bytes (`X-Account-Meta-Quota-Bytes`)
- Total number of objects created in all the containers in the account (`X-Account-Object-Count`)
- Number of containers in the account (`X-Account-Container-Count`)
- Storage space used, in bytes (`X-Account-Bytes-Used`)
- Custom account metadata (`X-Account-Meta-`*`Name`*)

You can view the account metadata by using the following interfaces:

Interface	Resources
Web Console (Not available on Oracle Cloud at Customer)	See Getting Account Metadata Using the Web Console.
RESTful API	See • Show Account Metadata in *REST API for Standard Storage in Oracle Cloud Infrastructure Object Storage Classic* • Getting Account Metadata Using the REST API
File Transfer Manager API	See `AccountMetadata` in *Java API Reference for Oracle Cloud Infrastructure Object Storage Classic File Transfer Manager*.
File Transfer Manager CLI	See Getting Account Metadata in *Command-Line Reference for Oracle Cloud Infrastructure Object Storage Classic*.

Getting Account Metadata Using the Web Console

(Not available on Oracle Cloud at Customer)

1. Sign in to the Oracle Cloud Infrastructure Object Storage Classic console.
2. Expand **Account Information**.
   The details of the account are displayed in the **Account Information** pane.

Getting Account Metadata Using the REST API

You can retrieve information about your Oracle Cloud Infrastructure Object Storage Classic account by sending a `HEAD` request to the account.

cURL Command Syntax

```
curl -v -X HEAD \
     -H "X-Auth-Token: token" \
     accountURL
```

- `token` is the authentication token obtained earlier from Oracle Cloud Infrastructure Object Storage Classic. See Authenticating Access When Using the REST API.
- For the syntax of `accountURL`, see About REST URLs for Oracle Cloud Infrastructure Object Storage Classic Resources.

HTTP Response Codes

- Success: `204 No Content`
- Failure: See Error Code Reference for Oracle Cloud Infrastructure Object Storage Classic

cURL Command Examples

Sample Cloud account with the following details:

- Account name: `acme`
- REST Endpoint URL: `https://acme.storage.oraclecloud.com/v1/Storage-acme`
- REST Endpoint (Permanent) URL: `https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/Storage-7b16fede61e1417ab83eb52e06f0e365`

> **Note:**
>
> The REST Endpoint (Permanent) URL is displayed for the accounts created after November 2017.

- Using the REST Endpoint URL obtained from the **REST Endpoint** field:

```
curl -v -X HEAD \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     https://acme.storage.oraclecloud.com/v1/Storage-acme
```

The following is an example of the output of this command:

```
> HEAD /v1/Storage-acme HTTP/1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.14.0.0
zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: acme.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b
>
< HTTP/1.1 204 No Content
< Accept-Ranges: bytes
< X-Account-Object-Count: 4510
< X-Account-Bytes-Used: 72691758435
< X-Account-Meta-Policy-Georeplication: us2
< X-Account-Meta-Policy-Archive: arch-us2
< X-Timestamp: 1412823447.62495
< X-Account-Meta-Test5: test1
< X-Account-Container-Count: 40
< X-Account-Meta-Owner: IT
< X-Account-Meta-Test1: test1
< X-Account-Meta-Test: test
```

```
< X-Trans-Id: tx8c2e61b26e684f77975a8-0057578589ga
< Date: Wed, 08 Jun 2016 02:40:09 GMT
< Connection: keep-alive
< X-Last-Modified-Timestamp: 1412823447.62495
< Content-Type: text/plain;charset=UTF-8
```

- Using the Service Permanent REST Endpoint URL obtained from the **REST Endpoint (Permanent)** field:

> **Note:**
>
> This cURL command example applies to the accounts created after November 2017.

```
curl -v -X HEAD \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/
Storage-7b16fede61e1417ab83eb52e06f0e365
```

The following is an example of the output of this command:

```
> HEAD /v1/Storage-7b16fede61e1417ab83eb52e06f0e365 HTTP/1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.14.0.0
zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b
>
< HTTP/1.1 204 No Content
< Accept-Ranges: bytes
< X-Account-Object-Count: 4510
< X-Account-Bytes-Used: 72691758435
< X-Account-Meta-Policy-Georeplication: us2
< X-Account-Meta-Policy-Archive: arch-us2
< X-Timestamp: 1412823447.62495
< X-Account-Meta-Test5: test1
< X-Account-Container-Count: 40
< X-Account-Meta-Owner: IT
< X-Account-Meta-Test1: test1
< X-Account-Meta-Test: test
< X-Trans-Id: tx8c2e61b26e684f77975a8-0057578589ga
< Date: Wed, 08 Jun 2016 02:40:09 GMT
< Connection: keep-alive
< X-Last-Modified-Timestamp: 1412823447.62495
< Content-Type: text/plain;charset=UTF-8
```

Deleting Account Metadata

You can delete account metadata by using the following interfaces:

Interface	Resources
Oracle Cloud Infrastructure Object Storage Classic Console (Not available on Oracle Cloud at Customer)	See Deleting Account Metadata Using the Web Console.
RESTful API	See: • Deleting Account Metadata Using the REST API • Create, update, or delete account metadata; or bulk-delete containers or objects in *REST API for Standard Storage in Oracle Cloud Infrastructure Object Storage Classic*

Deleting Account Metadata Using the Web Console

(Not available on Oracle Cloud at Customer)

1. Sign in to the Oracle Cloud Infrastructure Object Storage Classic console.
2. Expand **Account Information**.
3. Click **Edit**.
   In the **Account Information** pane, look for **Custom Metadata**.
4. Identify the metadata name and value that you want to delete.
5. Click  on the right side of the metadata value.
6. Click **Save**.
   The metadata name and value are deleted.

Deleting Account Metadata Using the REST API

cURL Command Syntax

```
curl -v -X POST \
     -H "X-Auth-Token: token" \
     -H "X-Remove-Account-Meta-Name: any_arbitrary_string" \
     accountURL
```

- `token` is the authentication token obtained earlier from Oracle Cloud Infrastructure Object Storage Classic. See Authenticating Access When Using the REST API.
- `Name` and `value` are the metadata key and value to be deleted.
- For the syntax of `accountURL`, see About REST URLs for Oracle Cloud Infrastructure Object Storage Classic Resources.

> **Note:**
>
> When you send a REST API request to Oracle Cloud Infrastructure Object Storage Classic, all non-ASCII characters in container names, object names and metadata values must be URL-encoded. For example, `my container` should be encoded as `my%20container`, where `%20` is the HTML encoding for the space character. Similarly, `my Über Container` should be encoded as `my %20%C3%9Cber%20Container`, where `%20` represents the space character and `%C3%9C` is the Ü character.

HTTP Response Codes

- Success: `204 No Content`
- Failure: See Error Code Reference for Oracle Cloud Infrastructure Object Storage Classic

cURL Command Examples

Sample Cloud account with the following details:

- Account name: `acme`
- REST Endpoint URL: `https://acme.storage.oraclecloud.com/v1/Storage-acme`
- REST Endpoint (Permanent) URL: `https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/Storage-7b16fede61e1417ab83eb52e06f0e365`

> **Note:**
>
> The REST Endpoint (Permanent) URL is displayed for the accounts created after November 2017.

- Using the REST Endpoint URL obtained from the **REST Endpoint** field:

```
curl -v -X POST \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     -H "X-Remove-Account-Meta-Category: IT" \
     https://acme.storage.oraclecloud.com/v1/Storage-acme
```

The following is an example of the output of this command:

```
> POST /v1/Storage-acme HTTP/1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.14.0.0
zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: acme.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b
> X-Remove-Account-Meta-Category: IT
>
< HTTP/1.1 204 No Content
< Accept-Encoding: identity
< X-Account-Meta-Category:
< Is_ssl: ssl
< User-Agent: proxy-server 39062
< Host: 160.34.16.120
```

```
< Referer: POST http://acme.storage.oraclecloud.com/v1/Storage-acme
< X-Trans-Id: txdad3557bf8694b95a71d3-00585b78f3ga
< X-Timestamp: 1482389747.77487
< Date: Thu, 22 Dec 2016 06:55:47 GMT
< Connection: keep-alive
< X-Last-Modified-Timestamp: 1482389747.77487
< Content-Type: text/html;charset=UTF-8
```

- Using the Service Permanent REST Endpoint URL obtained from the **REST Endpoint (Permanent)** field:

> **Note:**
>
> This cURL command example applies to the accounts created after November 2017.

```
curl -v -X POST \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     -H "X-Remove-Account-Meta-Category: IT" \
     https://
Sstorage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/
Storage-7b16fede61e1417ab83eb52e06f0e365
```

The following is an example of the output of this command:

```
> POST /v1/Storage-7b16fede61e1417ab83eb52e06f0e365 HTTP/1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.14.0.0
zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b
> X-Remove-Account-Meta-Category: IT
>
< HTTP/1.1 204 No Content
< Accept-Encoding: identity
< X-Account-Meta-Category:
< Is_ssl: ssl
< User-Agent: proxy-server 39062
< Host: 160.34.16.120
< Referer: POST http://
storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/
Storage-7b16fede61e1417ab83eb52e06f0e365
< X-Trans-Id: txdad3557bf8694b95a71d3-00585b78f3ga
< X-Timestamp: 1482389747.77487
< Date: Thu, 22 Dec 2016 06:55:47 GMT
< Connection: keep-alive
< X-Last-Modified-Timestamp: 1482389747.77487
< Content-Type: text/html;charset=UTF-8
```

Enabling Audit Logging

Audit logging provides usage accountability of the account and the resources created in the account. To record all the activities performed in an account for compliance reasons, you can use audit logs. For example: You can find out when an object was deleted and by whom.

Any user with the Service Administrator role can perform this task.

1. Create a container to store all the audit log events.
2. Configure the account to enable audit logging by specifying the container you've created to store the audit log events.
   Send a `POST` request to update the account metadata using the following header key: `X-Account-Meta-Audit-Log-Container`.

   The header value is name of the container you want to use for storing the audit log events.
3. List the events stored in the container to view the events.

To disable audit logging, send a `POST` request with the header `X-Remove-Account-Meta-Audit-Log-Container`.

Example:

Sample Cloud account with the following details:

- Account name: `acme`
- REST Endpoint URL: `https://acme.storage.oraclecloud.com/v1/Storage-acme`
- REST Endpoint (Permanent) URL: `https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/Storage-7b16fede61e1417ab83eb52e06f0e365`

> **Note:**
>
> The REST Endpoint (Permanent) URL is displayed for the accounts created after November 2017.

The REST Endpoint URL for the sample Cloud account is used in the steps below. To use the REST Endpoint (Permanent) URL, replace `https://acme.storage.oraclecloud.com/v1/Storage-acme` with `https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/Storage-7b16fede61e1417ab83eb52e06f0e365`.

1. Create a container `auditlog` to store all of the audit log events.

```
curl -v -X PUT \
     -H "X-Auth-Token: AUTH_tk6f5584fffdadc60870c876590efd9e18" \
     https://acme.storage.oraclecloud.com/v1/Storage-acme/auditlog
```

   The following is an example of the output of this command:

```
> PUT /v1/Storage-acme/auditlog HTTP/1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.14.0.0
zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: acme.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b
>
< HTTP/1.1 201 Created
< Date: Mon, 11 Jun 2018 03:59:35 GMT
< Content-Length: 0
< Content-Type: text/html; charset=UTF-8
< X-Trans-Id: tx23a1084b8c674fdeae8d4-0054f982ac
< Cache-Control: no-cache
```

```
< Pragma: no-cache
< Content-Language: en
```

2. Associate the container `auditlog` to the account to enable audit logging.

```
curl -v -X POST \
     -H 'X-Account-Meta-Audit-Log-Container: auditlog'
     -H 'X-Auth-Token: AUTH_tk6f5584fffdadc60870c876590efd9e18'
     https://acme.storage.oraclecloud.com/v1/Storage-acme
```

The following is an example of the output of this command:

```
> User-Agent: curl/7.29.0
> Host: acme.storage.oraclecloud.com
> Accept: */*
> X-Account-Meta-Audit-Log-Container: auditlog
> X-Auth-Token: AUTH_tk6f5584fffdadc60870c876590efd9e18
>
< HTTP/1.1 204 No Content
< Accept-Encoding: identity
< Is_ssl: ssl
< User-Agent: proxy-server 4924
< Host: 10.193.9.53
< X-Timestamp: 1528689575.91230
< X-Trans-Id: txe2f83090f27a4af0b33b9-005b1df3a7ga
< X-Trans-Id: txe2f83090f27a4af0b33b9-005b1df3a7ga
< Referer: POST http://acme.storage.oraclecloud.com/v1/Storage-acme
< Server: Oracle-Storage-Cloud-Service
< X-Account-Meta-Audit-Log-Container: auditlog
< Date: Mon, 11 Jun 2018 03:59:35 GMT
< Connection: keep-alive
< X-Last-Modified-Timestamp: 1528689575.91230
< Content-Type: text/html;charset=UTF-8
```

3. List all the events stored in the container `auditlog` to view the audit log.

```
curl -v -X GET
     -H 'X-Auth-Token: AUTH_tk6f5584fffdadc60870c876590efd9e18'
     https://acme.storage.oraclecloud.com/v1/Storage-acme/auditlog
```

The following is an example of the output of this command:

```
> User-Agent: curl/7.29.0
> Host: acme.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tk6f5584fffdadc60870c876590efd9e18
>
< HTTP/1.1 200 OK
< X-Container-Object-Count: 1
< X-Container-Write: acme.Storage.Storage_ReadWriteGroup
< Accept-Ranges: bytes
< X-Timestamp: 1528689463.04771
< X-Container-Read:
acme.Storage.Storage_ReadOnlyGroup,acme.Storage.Storage_ReadWriteGroup
< X-Container-Bytes-Used: 1171
< X-Trans-Id: txda33e2beffc14a46a8e0f-005b1df3f1ga
< Date: Mon, 11 Jun 2018 04:00:49 GMT
< Connection: keep-alive
< X-Storage-Class: Standard
< X-Container-Meta-Policy-Georeplication: container
< X-Container-Policies-Enabled: replication
< X-Last-Modified-Timestamp: 1528689462.70216
```

```
< Content-Type: text/plain;charset=utf-8
< Content-Length: 24
```

4. (Optional) Disable audit logging in the account.

```
curl -v -X POST \
     -H 'X-Remove-Account-Meta-Audit-Log-Container: d'
     -H 'X-Auth-Token: AUTH_tk6f5584fffdadc60870c876590efd9e18'
     https://acme.storage.oraclecloud.com/v1/Storage-acme
```

The following is an example of the output of this command:

```
> User-Agent: curl/7.29.0
> Host: acme.storage.oraclecloud.com
> Accept: */*
> X-Remove-Account-Meta-Audit-Log-Container: d
> X-Auth-Token: AUTH_tk6f5584fffdadc60870c876590efd9e18
>
< HTTP/1.1 204 No Content
< Accept-Encoding: identity
< Is_ssl: ssl
< User-Agent: proxy-server 27794
< Host: 10.193.9.53
< X-Timestamp: 1528689790.13098
< X-Trans-Id: tx47b76bc5475a452ba0d7b-005b1df47ega
< X-Trans-Id: tx47b76bc5475a452ba0d7b-005b1df47ega
< Referer: POST http://acme.storage.oraclecloud.com/v1/Storage-acme
< Server: Oracle-Storage-Cloud-Service
< X-Account-Meta-Audit-Log-Container:
< Date: Mon, 11 Jun 2018 04:03:10 GMT
< Connection: keep-alive
< X-Last-Modified-Timestamp: 1528689790.13098
< Content-Type: text/html;charset=UTF-8
<
* Connection #0 to host acme.storage.oraclecloud.com left intact
```

5

Frequently Asked Questions for Object Storage Classic

This section provides answers to frequently asked questions (FAQ) for Oracle Cloud Infrastructure Object Storage Classic.

Topics:

- How can I find the REST Endpoint URL of my Oracle Cloud Infrastructure Object Storage Classic instance?
- How can I make objects within a container publicly readable?
- How can I increase the upload speed of my large files?
- How can I use a Content Delivery Network (CDN) to accelerate the downloads of my publicly available content?
- Can I delete a container which contains objects?
- How can I store my Oracle Database backups to Oracle Cloud?
- How can I replicate data to multiple data centers?
- How do I find out the replication policy for my account?
- Are there any charges for the storage space used by replicated data stored in the secondary data center?
- Can I change the replication policy for my account?
- Can I change a replication policy for a container?
- How soon after a successful write operation is data replicated to the secondary data center?
- Is replication to the secondary data center guaranteed for every write operation?
- What happens when the primary site is unavailable?
- What happens when a subscription expires?
- How can I identify if a container is an archive or standard container?
- How long does it take to restore an archived object?
- How can I check whether archival or restoration of an object has been completed?
- How can I limit the capacity of a container?
- How can I find out the storage entitlement for my Oracle Cloud Infrastructure Object Storage Classic account?
- Can I upload dynamic large objects using Oracle Cloud Infrastructure Object Storage Classic?
- How do I contact Oracle for support?

How can I find the REST Endpoint URL of my Oracle Cloud Infrastructure Object Storage Classic instance?

See Finding the REST Endpoint URL for Your Cloud Account.

How can I make objects within a container publicly readable?

By default, all access to objects are private and require credentials. It is possible however, to make all objects within a specific container publicly readable.

To make the objects within a container publicly readable, add the following as an entry to the container's read Access Control List:

```
.r:*
```

Also, see Setting Container ACLs.
Note that the Oracle Cloud Infrastructure Object Storage Classic account is accountable for its monthly bandwidth usage, including traffic which is downloading publicly readable objects. If large amounts of traffic are expected, it is better to cache publicly readable objects in a Content Delivery Network (CDN). See How can I use a Content Delivery Network (CDN) to accelerate the downloads of my publicly available content?

How can I increase the upload speed of my large files?

Depending on your network connection, it may take a long time to upload large files.

To decrease the amount of time it takes to upload large files, try locally segmenting a large file into smaller pieces and then uploading the segments serially. Once all segments are uploaded, create a manifest object to map all the pieces together.

For more information about uploading large files by using the static large object approach, see Uploading Large Objects.

How can I use a Content Delivery Network (CDN) to accelerate the downloads of my publicly available content?

A Content Delivery Network (CDN) can be used to accelerate the download rate of objects from a container. It can also limit the egress bandwidth usage of an Oracle Cloud Infrastructure Object Storage Classic account which is serving publicly readable objects. Oracle does not provide CDN functionality with Oracle Cloud Infrastructure Object Storage Classic but customers can use their own third-party CDNs. Third-party CDNs must support *Origin Pull* and HTTPS in order to work with an Oracle Cloud Infrastructure Object Storage Classic account.

The first step is to make a container publicly readable. See How can I make objects within a container publicly readable?. The next step is to use the URL of the Oracle Cloud Infrastructure Object Storage Classic account as the *Origin Server* for the third-party CDN.

At this time, private objects cannot be accelerated with a CDN. Additionally, Oracle will not share the SSL certificate for the domains which Oracle Cloud Infrastructure Object Storage Classic accounts are hosted on.

How can I store my Oracle Database backups to Oracle Cloud?

You can use Oracle Database Backup Service, a cloud storage solution from Oracle for storing Oracle Database backups. It is built on top of the Oracle Cloud

Infrastructure Object Storage Classic, which provides reliable, secure, and scalable storage. In order to store backups in Oracle Database Backup Service, you have to install the client side Oracle Database Cloud Backup Module from OTN. This module is used with Recovery Manager (RMAN) to ship the backup data to Oracle Database Backup Service.

Note that you cannot use the Oracle Database Cloud Backup Module with Oracle Cloud Infrastructure Object Storage Classic. It can only be used with Oracle Database Backup Service.

Can I delete a container which contains objects?

Before you delete the container, you must first delete the objects in the container. To delete the objects, see Deleting Objects. To delete the container, see Deleting Containers.

How can I replicate data to multiple data centers?

(Not available on Oracle Cloud at Customer)
In Storage Classic accounts created after March 2018, the replication policy is set to `any` for the account, by default. For more information, see About Replication Policy for Your Object Storage Classic Instance.

If your account was created before March 2018, then you can enable geo-replication policy on your account to automatically replicate data to multiple data centers. You can do this by selecting a replication policy for your account. The replication policy defines the site where your data is replicated to and guarantees data consistency across multiple sites. For example, if you select the **Chicago-Ashburn** policy, you configure Chicago, Illinois, U.S.A as the primary site and Ashburn, Virginia, U.S.A. as the secondary site for your Oracle Cloud Infrastructure Object Storage Classic account. For more information, see About Replication Policy for Accounts Created Before March 2018.

How do I find out the replication policy for my account?

(Not available on Oracle Cloud at Customer)
Send a `HEAD` request to your account. Alternatively, you can view the replication policy in the Oracle Cloud Infrastructure Object Storage Classic console.

See Verifying the Replication Policy for Your Service Instance.

Are there any charges for the storage space used by replicated data stored in the secondary data center?

(Not available on Oracle Cloud at Customer)
You'll be charged for storage space used in each site — primary and secondary. You'll also be billed for the data transfer across the primary and secondary data centers.

Can I change the replication policy for my account?

(Not available on Oracle Cloud at Customer)
In Storage Classic accounts created after March 2018, the replication policy is set to `any` for the account, by default.

If your account was created before March 2018, then you **cannot** change the replication policy later.

Can I change the replication policy for a container?

You can set the container-specific replication policy when you create a container. You **cannot** change the container-specific replication policy later.

How soon after a successful write operation is data replicated to the secondary data center?

(Not available on Oracle Cloud at Customer)
After a successful write operation, your data is written only to the primary data center and it is asynchronously replicated to the secondary data center. The primary and secondary sites are always *eventually consistent*.

Is replication to the secondary data center guaranteed for every write operation?

(Not available on Oracle Cloud at Customer)
By default, data is *eventually consistent* across the primary and secondary data centers. For every write operation, when an object or container is created or modified, it is not replicated instantaneously to the secondary data center. Until the replication is completed, a container or an object's data may not be consistent across the primary and secondary sites. Over time, all changes to all objects or containers are replicated, and the data becomes consistent across the sites.

What happens when the primary site is unavailable?

(Not available on Oracle Cloud at Customer)
When the primary site is unavailable, a transparent failover redirects your global namespace to the secondary site and all your read requests are redirected to the secondary site. You cannot write data to primary site during failover. All your write requests generate the `403 – Forbidden` error during failover. When the primary site resumes its operation, a transparent failback redirects your global namespace back to the primary site. You can now write data to the primary site.

This is applicable only for accounts that have selected Chicago-Ashburn or Ashburn-Chicago as their geo-replication policy. The accounts that have selected either Chicago or Ashburn as their geo-replication policy will not have transparent failover.

What happens when a subscription expires?

(Not available on Oracle Cloud at Customer)
Go to Oracle Cloud Services Contracts and Service Descriptions page and look for*Oracle Platform as a Service (PaaS)* and *Infrastructure as a Service (IaaS) (Tech Cloud) - Service Descriptions* document.

How can I identify if a container is an archive or standard container?

(Not available on Oracle Cloud at Customer)
By default, containers are of the `Standard` storage class. For an `Archive` container, the `X-Storage-Class` header in the container metadata displays the value `Archive`. See Getting Container Metadata.

How long does it take to restore an archived object?

(Not available on Oracle Cloud at Customer)
It takes up to four hours to restore an archived object. You can track the restoration progress and determine if the object is restored. For more information, see Tracking Restoration of an Object in an Archive Container.

How can I check whether archival or restoration of an object has been completed?

(Not available on Oracle Cloud at Customer)
Send a `HEAD` request to the object. You can determine the status of the object in the `Archive` container from the `X-Archive-Restore-Status` header in the object metadata. For more information, see Finding Out the Status of Objects in an Archive Container.

How can I limit the capacity of a container?

You can set a container quota – in terms of maximum number of objects or maximum number of bytes. See Setting Container Quotas.

How can I find out the storage entitlement for my account?

To find out the storage entitlement for your account, see Viewing Service Details in My Account for Entitlements in *Managing and Monitoring Oracle Cloud*.

If you have a metered account and would like to know your account balance details, see Viewing Estimated Account Balance Details of Metered Services in *Managing and Monitoring Oracle Cloud*.

Can I upload dynamic large objects using Oracle Cloud Infrastructure Object Storage Classic?

Support for uploading dynamic large objects is not available in Oracle Cloud Infrastructure Object Storage Classic. You can download existing dynamic large objects from your account in Oracle Cloud Infrastructure Object Storage Classic.

How do I contact Oracle for support?

See Contacting Oracle for Support.

6 Troubleshooting Object Storage Classic

This section provides solutions for problems you may encounter while using Oracle Cloud Infrastructure Object Storage Classic.

For information about error codes that Oracle Cloud Infrastructure Object Storage Classic may return, see Error Code Reference for Oracle Cloud Infrastructure Object Storage Classic.

Topics:

- Getting a 403 Forbidden error when creating a container
- I created a container (or object), but a GET request doesn't return it
- I updated a container (or object), but a GET request returns old data
- I deleted a container (or object), but a GET request still returns it
- Contacting Oracle for Support

Getting a 403 Forbidden error when creating a container

See 403 Forbidden.

I created a container (or object), but a GET request doesn't return it

The new container or object that you created has not been replicated across all the nodes. Wait for the container or object to be replicated across all the nodes, and try again.

I updated a container (or object), but a GET request returns old data

The updates you made to the container or object have not been replicated across all the nodes. Wait for the updates to be replicated across all the nodes, and try again.

I deleted a container (or object), but a GET request still returns it

The `DELETE` operation has not been replicated across all the storage nodes. Wait for the replication to be completed, and try again.

Contacting Oracle for Support

1. Go to https://support.oracle.com.
2. In the **Sign In** pane, select **Cloud Support** as the portal and sign in.
3. On the **Dashboard** page, click **Create Service Request**.
4. In the **Create Service Request** wizard, do the following:
    a. In the **Service Type** field, select **Oracle Cloud Infrastructure Object Storage Classic**.
    b. In the **Problem Type** field, select the appropriate problem subtype.
5. Follow the prompts in the wizard to complete the service request.

A

Error Code Reference for Object Storage Classic

Oracle Cloud Infrastructure Object Storage Classic returns standard HTTP responses. Every HTTP response contains a status code that indicates success or failure of the request. This section lists the causes and solutions for the HTTP error codes that Oracle Cloud Infrastructure Object Storage Classic may return. If you see an error that it is not documented here, contact Oracle Support.

Topics:

- 400 Bad Request
- 401 Unauthorized
- 403 Forbidden
- 404 Not Found
- 409 Conflict
- 411 Length Required
- 413 Request Entity Too Large
- 422 Unprocessable Entity

400 Bad Request

(Not available on Oracle Cloud at Customer)
Cause

- You tried restoring an object in a container of the `Standard` storage class.
- You tried downloading an object in a container of the `Archive` storage class without first restoring the object.

Solution

- Only objects in a container with the header `X-Storage-Class: Archive` can be restored.
- To download an object in a container with the header `X-Storage-Class: Archive`, you must first restore it.
  1. Restore the object. See Restoring Archived Objects.
  2. Make a note of the job ID mentioned in the header `X-Archive-Restore-JobId`. Track the restoration progress of the object. See Tracking Restoration of an Object in an Archive Container.
  3. Download the object. See Downloading Objects

401 Unauthorized

Cause

- The request does not include an authentication token.
- The authentication token specified in the request is not valid. It may have expired. Authentication tokens expire after 30 minutes.

Solution

Ensure that a valid authentication token is included in the request. See Authenticating Access When Using the REST API.

403 Forbidden

This error may occur when you:

- Create a container or an object
- Update or delete an existing container or an object
- View the metadata of a container or an object
- Download an object
- List objects within a container
- Restore an object in a container for which the `X-Container-Read` ACL does not have any of your roles
- Attempt an operation that's not supported for containers of the `Archive` storage class. See Features Not Supported for Archive Containers. (Not available on Oracle Cloud at Customer)
- Create an `Archive` container in unmetered Oracle Cloud Infrastructure Object Storage Classic accounts. (Not available on Oracle Cloud at Customer)

Cause

- A replication policy has not been selected for your service in Oracle Cloud Infrastructure Classic Console. (Not available on Oracle Cloud at Customer) This is applicable only for the accounts created before March 2018.
- A replication policy has been selected for your service, and a write request was sent to the secondary data center. (Not available on Oracle Cloud at Customer) This is applicable only for the accounts created before March 2018.
- The request was sent to an incorrect data center. For example, the data center for your service is *Chicago (us2)*, but the request was sent to the URL corresponding to the *Ashburn (us6)* data center.
- You don’t have the required permission to perform the operation on the specified container. For example, there may be a change in the roles assigned to your user or the access privileges defined for the container specified in the request.

Solution

1. Ask your account administrator whether a replication policy has been selected for the service. See About Replication Policy for Your Object Storage Classic Instance.
   This is applicable only for the accounts created before March 2018.
2. Make sure that the request is sent to the primary data center selected for your service or to the global namespace URL. See About REST URLs for Oracle Cloud Infrastructure Object Storage Classic Resources.

- Write requests (`PUT`, `POST`, and `DELETE`) can be sent to either the primary data center or the global namespace URL. See About REST URLs for Oracle Cloud Infrastructure Object Storage Classic Resources.
- Read requests (`GET` and `HEAD`) can be sent to the primary data center, the secondary data center, or the global namespace URL.

3. Ask your service administrator whether there is a change in either the roles assigned to your user or the access privileges to the container specified in your request.
4. If the roles or access privileges have changed, request the service administrator to grant you the required permissions to the container.
5. You can restore an archived object only if you have a role that is specified in the `X-Container-Read` ACL.
6. If your request pertains to a container of the `Archive` storage class, check whether the operation is supported. See Features Not Supported for Archive Containers.
7. You can create an `Archive` container only in a metered account.

404 Not Found

This error may occur when you:

- List objects in a nonexistent container
- View or update the metadata of a nonexistent object or container
- Download or copy a nonexistent object
- Delete a nonexistent object or container

Cause

- The specified container or object doesn't exist in Oracle Cloud Infrastructure Object Storage Classic.
- The specified container or object was just created and has not been replicated across all the nodes.

Solution

- Verify whether the specified container or object exists, by listing the containers or listing the objects within the specified container, as appropriate.
- If the container or object was just created, wait for the container or object to get replicated across all the nodes.

409 Conflict

Cause

The specified container cannot be deleted because it contains one or more objects.

Solution

List the objects within the specified container. Delete all the objects in the container and then try deleting the container.

411 Length Required

Cause

The request doesn't contain the `Content-Type` or `Content-Length` request header. For example, you are trying to create an object without any content.

Solution

Ensure that the `Content-Type` or `Content-Length` request header is included in the HTTP request.

413 Request Entity Too Large

Cause

- The file being uploaded is larger than 5 GB, which is the limit for the size of a single object.
- The unused account quota is less than the size of the objects being created.
- If a container-level quota—maximum bytes or number of objects—was set, the unused container quota , is not sufficient for the objects being created.

Solution

- For information about uploading files larger than 5 GB, see Uploading Large Objects.
- If the unused account quota is not sufficient for the objects being created, delete existing objects that are no longer required or purchase more quota.
  To find out the current unused account quota, send a `HEAD` request for the account, and look for the values of the `X-Account-Bytes-Used` and `X-Account-Meta-Quota-Bytes` headers in the response. `X-Account-Meta-Quota-Bytes` minus `X-Account-Bytes-Used` is the unused account quota.
- If the unused container quota is less than the size of the objects being created, delete existing objects that you no longer require or increase the container quota.
  To find out whether container quotas have been set and to determine the current unused quota, send a `HEAD` request for the container, and look for the values of the following headers in the response:
  - `X-Container-Object-Count` and `X-Container-Meta-Quota-Count`
  - `X-Container-Bytes-Used` and `X-Container-Meta-Quota-Bytes`

  `X-Container-Meta-Quota-Count` minus `X-Container-Object-Count` is the number of further objects that can be stored in the container.

  `X-Container-Meta-Quota-Bytes` minus `X-Container-Bytes-Used` is the unused storage space in the container.

  See Setting Special Metadata on a Container.

422 Unprocessable Entity

Cause

The value of the `ETag` header specified in the upload request doesn't match the MD5 checksum of the HTTP response.

Solution

This error may be due to a problem in data transmission. Delete the specified object and try again.

B

Certified Third-Party Products

 This topic does not apply to Oracle Cloud at Customer.

Third-Party Product	Purpose	Certified for Storage Class
CloudBerry Explorer	Cloud explorer and client	`Archive` and `Standard`
Commvault • v10 (sp11) • v11 (sp4)	Backup and recovery	• v10 (sp11) - `Standard` only • v11 (sp4) - `Archive` and `Standard`
CyberDuck Explorer 5.0	Cloud explorer and client	`Standard` only
iRODS 4.1.8	Cloud gateway	`Standard` only
NetApp AltaVault 4.0.x, 4.2.0	Cloud gateway	`Standard` only
OpenStack Swift Client 2.4, 2.7	Cloud explorer and client For installation instructions, see https://docs.openstack.org/pike/user/.	`Standard` only
Veritas NetBackup 7.7.x	Backup and recovery	`Standard` only

C

Java Code Samples for Using the File Transfer Manager API

This appendix provides code samples for using the File Transfer Manager API. For more information, see Java API Reference for File Transfer Manager API.

Uploading a Single File

Sample Cloud account with the following details:

- Account name: `acme`
  For traditional accounts, account name and identity domain name are the same.
- REST Endpoint URL: `https://acme.storage.oraclecloud.com/v1/Storage-acme`
- REST Endpoint (Permanent) URL: `https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/Storage-7b16fede61e1417ab83eb52e06f0e365`

> **Note:**
>
> The REST Endpoint (Permanent) URL is displayed for the accounts created after November 2017.

- For the REST Endpoint URL obtained from the **REST Endpoint** field:

```
     FileTransferAuth auth = new FileTransferAuth
           (
            "john.doe@example.com", // user name
            "password".toCharArray(), // password
            "Storage", //  service name
            "https://acme.storage.oraclecloud.com", // service URL
            "acme" // identity domain
           );
        FileTransferManager manager = null;
        try {
            manager = FileTransferManager.getDefaultFileTransferManager(auth);
            String containerName = "mycontainer";
            String objectName = "xyz.txt";
            File file = new File("/tmp/xyz.txt");
            UploadConfig uploadConfig = new UploadConfig();
            uploadConfig.setOverwrite(true);
            uploadConfig.setStorageClass(CloudStorageClass.Standard);
            System.out.println("Uploading file " + file.getName() + " to
container " + containerName);
            TransferResult uploadResult = manager.upload(uploadConfig,
containerName, objectName, file);
            System.out.println("Upload completed successfully.");
            System.out.println("Upload result:" + uploadResult.toString());
        } catch (ClientException ce) {
            System.out.println("Upload failed. " + ce.getMessage());
```

```
        } finally {
            if (manager != null) {
                manager.shutdown();
            }
        }
    }
}
```

- For the Service Permanent REST Endpoint URL obtained from the **REST Endpoint (Permanent)** field:
  This cURL command example applies to the accounts created after November 2017.

```
     FileTransferAuth auth = new FileTransferAuth
            (

"Storage-7b16fede61e1417ab83eb52e06f0e365:john.doe@example.com", // user name
              "password".toCharArray(), // password
              "https://
storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com" // service URL
            );
        FileTransferManager manager = null;
        try {
            manager = FileTransferManager.getDefaultFileTransferManager(auth);
            String containerName = "mycontainer";
            String objectName = "xyz.txt";
            File file = new File("/tmp/xyz.txt");
            UploadConfig uploadConfig = new UploadConfig();
            uploadConfig.setOverwrite(true);
            uploadConfig.setStorageClass(CloudStorageClass.Standard);
            System.out.println("Uploading file " + file.getName() + " to
container " + containerName);
            TransferResult uploadResult = manager.upload(uploadConfig,
containerName, objectName, file);
            System.out.println("Upload completed successfully.");
            System.out.println("Upload result:" + uploadResult.toString());
        } catch (ClientException ce) {
            System.out.println("Upload failed. " + ce.getMessage());
        } finally {
            if (manager != null) {
                manager.shutdown();
            }
        }

    }
}
```

D

Character Restrictions

You can view the character restrictions when you create and update resources in Oracle Cloud Infrastructure Object Storage Classic.

Input Parameter	Input Restrictions	Unsupported Characters
• Container name	• Only UTF-8 characters • Maximum of 1061 bytes • Can start with any character • Cannot contain a slash (/) character because this character delimits the container and object name	The following characters are not supported: • Characters: ', ", `, <, and >. For example: jane's_file, "Hello_World".txt, Send>Customers.pdf • Strings: When the name contains `/./` or `/../`; When the name ends with `/.` or `/..`; For example: mymachine/./etc, current/../file, php/., object/..
• Object name	• Only UTF-8 characters • Maximum of 1061 bytes • Can start with any character • Cannot contain a slash (/) character because this character delimits the container and object name	The following characters are not supported: • Characters: ', ", `, <, and >. For example: jane's_file, "Hello_World".txt, Send>Customers.pdf • Strings: When the name contains `/./` or `/../`; When the name ends with `/.` or `/..`; For example: mymachine/./etc, current/../file, php/., object/..
• Account custom metadata name • Container custom metadata name • Object custom metadata name	• Only ASCII characters • Maximum of 128 bytes	The following US-ASCII characters are not supported: • Control characters (octet 0-31) and DEL (127) • Separators (,), <, >, @, ,, ;, :, \, ", /, [,], ?, =, {, }, space, horizontal-tab For more information, see https://www.w3.org/Protocols/rfc2616/rfc2616-sec2.html#sec2.2.

Input Parameter	Input Restrictions	Unsupported Characters
• Account custom metadata value • Container custom metadata value • Object custom metadata value	• Any UTF-8 characters • Maximum of 256 bytes	None

E

About Replication Policy for Accounts Created Before March 2018

> **Important:**
>
> This topic applies to accounts created before March 2018.
> For the accounts created after March 2018, the replication policy for the account is set to `any`, by default.

If your account was created **before** March 2018, then you must **select** a replication policy for your account after activating your Storage Classic Subscription. This replication policy defines your primary data center and also specifies whether your data should be replicated to a geographically distant (secondary) data center. Data is written to the primary data center and replicated asynchronously to the secondary data center. The primary and secondary data centers are eventually consistent. In addition to being billed for storage capacity used at each data center, you will also be billed for bandwidth used during replication between data centers.

> **Note:**
>
> See About Guidelines for Selecting a Replication Policy before you select a replication policy for your account.

You can select a replication policy by using the following interfaces:

Interface	Resources
Web Console (Not available on Oracle Cloud at Customer)	See Selecting a Replication Policy for Your Account Using the Web Console.
RESTful API	See Selecting a Replication Policy for Your Account Using the REST API.

About Guidelines for Selecting a Replication Policy

Read these guidelines before you select a replication policy.

Oracle provides several replication policies. Broadly, they belong to one of the following types:

- **Policies that have no georeplication:** These policies specify only the primary data center (DC) that hosts your account.
  All read and write requests go to the primary DC, always. If the primary DC is unavailable, then the requests fail.

Such a policy may be adequate if you have standard data-durability requirements and if an occasional failure of read requests (when the primary DC is unavailable) is acceptable.

- **Georeplication policies:** These policies specify a primary DC that hosts your account, and a geographically distant, georeplication DC.
  Write requests that you send to the global namespace URL are routed to the primary DC. Data that you write is replicated automatically, but asynchronously, to the georeplication DC. The primary and secondary DCs are eventually consistent.

  If the primary DC is unavailable, then write requests fail with the `403 – Forbidden` error, but read requests are routed to the georeplication DC. When the primary DC is available again, read requests that are sent to the global namespace URL are routed to the primary DC.

  You'll be billed for the sum of the capacities used in both DCs and for the data transfer from the primary to the georeplication DC.

  A policy that specifies a georeplication DC is ideal if you have advanced durability requirements for your data or if read requests must succeed always regardless of the state of the primary DC.

For faster data transfer between your other Oracle Cloud services and Oracle Cloud Infrastructure Object Storage Classic, consider selecting a policy that specifies the primary DC that hosts your other services that use Oracle Cloud Infrastructure Object Storage Classic the most.

For example, if Oracle Java Cloud Service is provisioned in the Chicago data center, then for faster data transfer between your Oracle Java Cloud Service instances and Oracle Cloud Infrastructure Object Storage Classic, select a replication policy that specifies Chicago (us2) as the primary data center.

When you select the replication policy for your account, keep in mind any security, legal, and regulatory requirements that may apply to your data.

Selecting a Replication Policy for Your Account Using the Web Console

Read the guidelines carefully, and pick a replication policy that serves your business requirements.

1. Sign in to your Oracle Cloud account.
   If you see Infrastructure Classic at the top of the page when you sign in to Oracle Cloud, then you are using the Infrastructure Classic Console to access your services and your subscription does not support access to the Infrastructure Console. See Signing In to Your Cloud Account in *Getting Started with Oracle Cloud*.

   If you can access the service from the Infrastructure Console, see Signing In to the Console in Oracle Cloud Infrastructure documentation.

2. Look for **Storage**, and from the **Actions**  menu, select **Open Service Console**.

3. Click **Storage**.
   The **Service Details** page displays the details of your Oracle Cloud Infrastructure Object Storage Classic account.

   - If a replication policy has not yet been selected for your account, then the **Guidelines for Selecting a Replication Policy** dialog box appears.

- If the **Guidelines for Selecting a Replication Policy** dialog box does not appear, a replication policy has already been selected for your account. Skip the remainder of this procedure. You can start creating and managing containers and objects in the service.

> **Note:**
>
> If you try to create containers and objects without first selecting a replication policy, then the requests you send to the service will fail with the `403 Forbidden` error.

4. > **Caution:**
   >
   > After you select a replication policy, you can't change it. Select the policy carefully.

   From the drop-down list at the bottom of the dialog box, select a replication policy.

5. After you select a replication policy, click **Set Policy**.
   The Oracle Cloud Infrastructure Object Storage Classic web console is displayed.
6. Expand the **Account Information** pane.
   The selected georeplication policy is displayed.

   If the policy field is blank, then refresh the browser.

You can start creating containers and objects in Oracle Cloud Infrastructure Object Storage Classic.

Finding the Data Center-Specific URL for Your Account

1. Select the data center to set the replication policy for your account.
2. Determine the data center code:

Region	Location	Code
US Commercial 2	Chicago, Il. US	`us2`
US Commercial 6	Ashburn, Va. US	`us6`
US Commercial Central	Illinois, US	`uscom-central-1`
EMEA Commercial 2	Amsterdam, NL. EMEA	`em2`
EMEA Commercial 3	Slough, UK. EMEA	`em3`
Sydney	Sydney, Australia	`aucom-east-1`
LAD	Sao Paulo, Brazil	`brdc1`

   Example: `em2`

3. Construct the data center-specific URL for your account.
   Format:

   `https://`*`dataCenterCode`*`.storage.oraclecloud.com/v1/Storage-`*`account_name`*

   For traditional accounts, the account name and the identity domain name are the same.

   Example:

```
https://em2.storage.oraclecloud.com/v1/Storage-acme
```

Selecting a Replication Policy for Your Account Using the REST API

1. Find out the data center-specific URL for your account. See Finding the Data Center-Specific URL for Your Account.
2. Request an authentication token. See Authenticating Access When Using the REST API.
3. Pick a suitable replication policy for your account from one of the following tables, based on the region where your account is provisioned:

   For US Accounts

Primary Region	Georeplication Region	Policy
Chicago, Il. US	None	`us2`
Ashburn, Va. US	None	`us6`
US Commercial Central, Illinois, US	None	`uscom-central-1`
Chicago, Il. US	Ashburn, Va. US	`us2-us6` Available only for entitlement-based accounts
Ashburn, Va. US	Chicago, Il. US	`us6-us2` Available only for entitlement-based accounts
Ashburn, Va. US	US Commercial Central, Illinois, US	`us6-uscom-central-1` Available only for entitlement-based accounts
Chicago, Il. US	US Commercial Central, Illinois, US	`us2-uscom-central-1` Available only for entitlement-based accounts
US Commercial Central, Illinois, US	Ashburn, Va. US	`uscom-central-1-us6` Available only for entitlement-based accounts
US Commercial Central, Illinois, US	Chicago, Il. US	`uscom-central-1-us2` Available only for entitlement-based accounts
Chicago, Il. US	Ashburn, Va. US	`us2-uscom-east-1`
Ashburn, Va. US	None	`us6-uscom-east-1`
Ashburn, Va. US	None	`uscom-east-1-us6`
Ashburn, Va. US	Chicago, Il. US	`uscom-east-1-us2`
Ashburn, Va. US	US Commercial Central, Illinois, US	`uscom-east-1-uscom-central-1`
US Commercial Central, Illinois, US	Ashburn, Va. US	`uscom-central-1-uscom-east-1`
Ashburn, Va. US	None	`uscom-east-1`

   For EMEA Accounts

Primary Region	Georeplication Region	Policy
Amsterdam, NL. EMEA	None	`em2`

Primary Region	Georeplication Region	Policy
Slough, UK. EMEA	None	em3
Amsterdam, NL. EMEA	Slough, UK. EMEA	em2–em3 Available only for entitlement-based accounts
Slough, UK. EMEA	Amsterdam, NL. EMEA	em3–em2 Available only for entitlement-based accounts
Amsterdam, NL. AD1 EMEA	Amsterdam, NL. EMEA	eucom-north-1–em2
Slough, UK. EMEA	London, UK. AD1 EMEA	em3-gbcom-south-1
London, UK. AD1 EMEA	None	gbcom-south-1
London, UK. AD1 EMEA	Amsterdam, NL. AD1 EMEA	gbcom-south-1-eucom-north-1
Amsterdam, NL. AD1 EMEA	Slough, UK. EMEA	eucom-north-1-em3
Amsterdam, NL. EMEA	Amsterdam, NL. AD1 EMEA	em2-eucom-north-1
Slough, UK. EMEA	Amsterdam, NL. AD1 EMEA	em3-eucom-north-1
Amsterdam, NL. EMEA	London, UK. AD1 EMEA	em2-gbcom-south-1
Amsterdam, NL. AD1 EMEA	None	eucom-north-1
Amsterdam, NL. AD1 EMEA	London, UK. AD1 EMEA	eucom-north-1-gbcom-south-1
London, UK. AD1 EMEA	Slough, UK. EMEA	gbcom-south-1-em3
London, UK. AD1 EMEA	Amsterdam, NL. EMEA	gbcom-south-1-em2

For APAC Accounts

Primary Region	Georeplication Region	Policy
Sydney, Australia	None	aucom-east-1

For LAD Accounts

Primary Region	Georeplication Region	Policy
Sao Paulo, Brazil	None	brdc1

4. Send a `POST` request to set the policy for your account:

```
curl -v -X POST \
        -H "X-Auth-Token: token \
        -H "X-Account-Meta-Policy-Georeplication: ReplicationPolicy"
        data-center-specific–URL
```

- `token` is the authentication token obtained earlier from Oracle Cloud Infrastructure Object Storage Classic.
- `ReplicationPolicy` is the value of the replication policy to be set for your account.
- `data-center-specific–URL` is the data center-specific URL from Step 1.

> **Caution:**
>
> After you select a replication policy, you can't change it. Select the policy carefully.
> See About Guidelines for Selecting a Replication Policy.

Example:

```
curl -v -X POST \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     -H "X-Account-Meta-Policy-Georeplication: us2" \
     https://us2.storage.oraclecloud.com/v1/Storage-acme
```

The following is an example of the output of this command:

```
> POST /v1/Storage-acme HTTP/1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.14.0.0 zlib/
1.2.3 libidn/1.18 libssh2/1.4.2
> Host: us2.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b
> X-Account-Meta-Policy-Georeplication: us2
>
< HTTP/1.1 204 No Content
< Date: Fri, 06 Nov 2016 11:44:29 GMT
< Content-Length: 0
< Content-Type: text/html; charset=UTF-8
< X-Trans-Id: tx0742dd38e3a445529860a-0054f9931d
< Cache-Control: no-cache
< Pragma: no-cache
< Content-Language: en
<
```

To find out the replication policy that's selected for your Oracle Cloud Infrastructure Object Storage Classic account, Verifying the Replication Policy for Your Service Instance.

You can now start creating containers and uploading objects in your account. See Creating Containers.