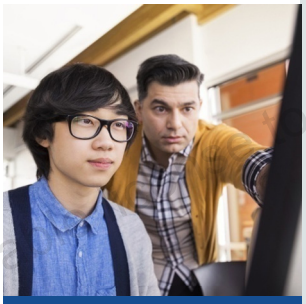
14

Oracle Storage Cloud Service Access and Authentication

Objectives

After completing this lesson, you should be able to:

- Access object storage on Oracle Cloud
- Define Access Control Lists (ACLs)
- Protect data stored in object storage on Oracle Cloud using ACLs



How Do I Authenticate Access to Using REST API?

1. Request an authentication token.
2. Construct the authentication URL for your account.
3. Execute the cURL request commands:
   - `GET`
   - `PUT`

The following are the three main steps involved in accessing object storage on the cloud via the REST API:

1. Request an authentication token.
2. Construct the authentication URL for your account.
3. Execute the cURL commands.
   - `GET`
   - `PUT`

Note: If you do not already have cURL installed, refer to the lesson titled “Introduction to Oracle Storage Cloud Service” for instructions on how to obtain and install it.

How Do I Request an Authentication Token?

- Execute the cURL command:
  - `GET` request

```
curl -v -X GET \
     -H "X-Storage-User: myService-myIdentity3:myUsername" \
     -H "X-Storage-Pass: myPassword" \
     https://foo.storage.oraclecloud.com/auth/v1.0
```


- 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.
- This slide shows an example of a cURL command for requesting an authentication token using `GET`.
- The following slide shows the output.

Output

- Output of the `GET` request:

```
> GET /auth/v1.0 HTTP/1.1
> Host: foo.storage.oraclecloud.com
> Accept: */*
> X-Storage-User: myService-myIdentity3:myUsername
> X-Storage-Pass: myPassword

< HTTP/1.1 200 OK
< X-Storage-Url: https://foo.storage.oraclecloud.com/v1/myService-myIdentity3
< 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.

Now How Do I Construct the Authentication URL for My Account? (1/2)

1. Sign in to the **Oracle Cloud My Services** application.
   - The **My Services** dashboard is displayed. It lists the services that are assigned to your account.
2. Look for **Oracle Storage Cloud Service**.
3. Select **View Details** from the **Actions** menu.
4. Alternatively, click the **Oracle Storage Cloud Service** link on the **Dashboard** page.
   - On the resulting page, the details of your Oracle Storage Cloud Service instance are displayed.

Now How Do I Construct the Authentication URL for My Account? (2/2)

5. Note the REST Endpoint URL, which is displayed in the **REST Endpoint** field under the **Additional Information** section.
   - For example: `https://foo.storage.oraclecloud.com/v1/Storage-myIdentity3`
6. Delete the following portion of the REST Endpoint URL: `v1/Storage-myIdentity3`
   - Now, the edited URL should be: `https://foo.storage.oraclecloud.com/`
7. Append the following to the edited URL: `auth/v1.0`
8. Assuming that the REST endpoint URL for your account is `https://foo.storage.oraclecloud.com/v1/Storage-myIdentity3`, the equivalent authentication URL would be `https://foo.storage.oraclecloud.com/auth/v1.0`

Done. What's Next?

Constructed authentication URL

- When sending the `GET` request you must include user credentials in the following headers:
  - `X-Storage-User`

    Syntax for metered subscriptions:
    ```
    X-Storage-User: Storage-identityDomainID:username
    ```

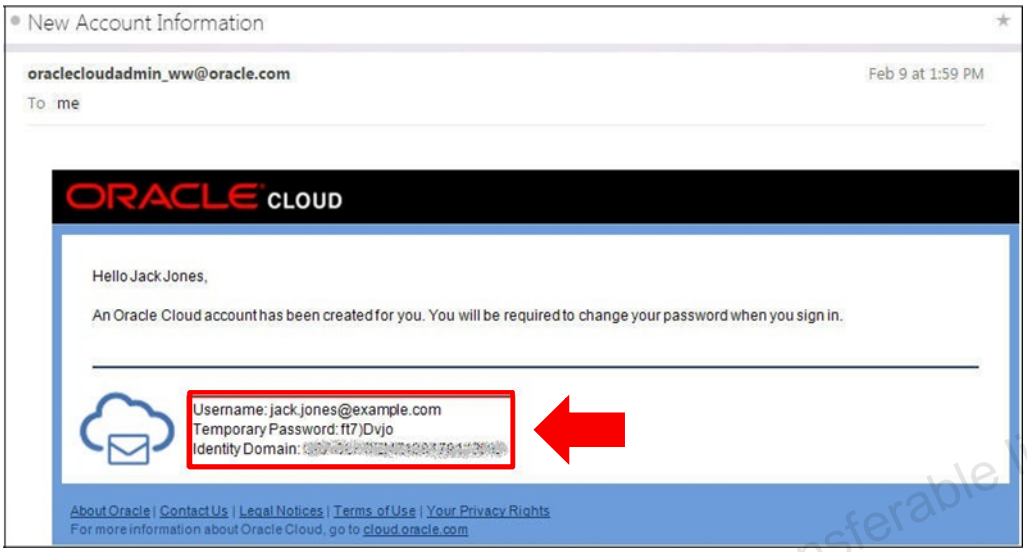
    Syntax for nonmetered subscriptions:
    ```
    X-Storage-User: serviceInstanceName-identityDomainID:userName
    ```

  - `X-Storage-Pass: password`

When you send the `GET` request to the authentication URL that you just constructed, include the user credentials in the following headers.

- `X-Storage-User` has two types of subscribers:
  - **Syntax for metered subscriptions:** `X-Storage-User: Storage-identityDomainID:username`
  - **Syntax for nonmetered subscriptions:** `X-Storage-User: serviceInstanceName-identityDomainID:username`
- `X-Storage-Pass: password`

Where Can I Find My User Information?



- You can find out your username, password, and identity domain from the "*New Account Information*" email that you received from Oracle Cloud when your account was set up, as shown in the slide.
- If you do not have your "*New Account Information*" email, ask your account administrator for your Oracle Cloud username, password, and identity domain.

How Do I Store an Object?

- Storing an object in an account using an authentication token
- Execute the cURL command:
  - `PUT` request

```
curl -v -X PUT \
    -H "X-Auth-Token: AUTH_tk209f7f2ea1265a0d3f29d28a2dc8ced6" \
    -d "Hello, World!" \
    https://foo.storage.oraclecloud.com/v1/myService-
 myIdentity3/myContainer/myObject
```


- The slide shows an example of a cURL command for storing an object in an account using an authentication token using `PUT`.
- The following slide shows the output.

Output

Output of the `PUT` request:

```
> PUT /v1/myService-myIdentity3/myContainer/myObject HTTP/1.1
> Host: foo.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
```


This shows an example of the output of this command.

Using Access Control Lists (ACLs) to Protect Data

- User privileges
  - Permissions based on user roles
    - `X-Container-Read`
    - `X-Container-Write`
- Data protection
  - Implementing Access Control Lists to containers
    - Access to containers and objects can be granted or denied
  - Permission to read and/or write
  - Unique user role: `Storage_Adiministrator`

- 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.
- Data protection is guaranteed because service administrators can grant read or write access to users.
- 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 on the My Services Users page and assign them to the `X-Container-Read` and `X-Container-Write` headers on containers, as required.
- Users with the *Storage_Administrator* role will always have read and write access to all containers in their service instance.

So, What Are the Default Values when Creating a New Container?

- Default ACLs assigned to containers
  1. `X-Container-Read: identity_domain_ID.storage_service.Storage_ReadOnlyGroup,identity_domain_ID.storage_service.Storage_ReadWriteGroup`
  2. `X-Container-Write: identity_domain_ID.storage_service.Storage_ReadWriteGroup`
- All non-administrators are subject to the ACLs for a given container with the exception of the service instance root path.
  - However, only users with the `Storage_Administrator` role can create or delete containers.

- By default, when a container is created in the Oracle Storage Cloud Service, 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`
- 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; however, only users with the *Storage_Administrator* role can create or delete containers.

Example: Creating a Container with Default Values

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`

Quiz

What does ACLs stand for?

a. Access Control Lists
b. Access Computer Lists
c. Admin Control Lists
d. Admin Command Lists

Quiz

An authentication token must be requested in order to authenticate access to Storage Cloud Service.

a. True
b. False

Quiz

What request command do you use to store an object?

a. `POST`

b. `PUT`

c. `GET`

d. `POST` OR `PUT`

Quiz

To better protect data from users, what do you implement to grant or deny access permissions?

a. Admin Control Lists
b. Access Control Levels
c. Access Control Lists
d. Admin Control Lists

Summary

In this lesson, you should have learned how to:

- Access object storage on Oracle Cloud
- Define Access Control Lists (ACLs)
- Protect data stored in object storage on Oracle Cloud using ACLs