# User Manual for Swift+Keystone

## Section 1: Overview

- **Swift:**

OpenStack Swift, also known as OpenStack Object Storage, is open source software designed to manage the storage of large amounts of data cost-effectively on a long-term basis across clusters of standard server hardware. It provides a fully distributed, API-accessible storage platform that can be integrated directly into applications or used for backup, archiving, and data retention.

For the Object Storage service to be secure, some authorization services must be used. The Swift project includes two auth systems:

1) Tempauth
2) Keystone

- **Keystone:**

Openstack Keystone, also known as OpenStack Identity service,  is a set of services developed for the purposes of user authentication and authorization by OpenStack cloud. The service allows to securely check the client’s identity and assign a unique access code (**token**) trusted by internal services.

- **TempAuth:**

Although keystone is the default  and highly used Identity service which provides means of authentication for OpenStack services and users,  tempauth is really light weight and toy approach for the same purpose used in openstack which in times become handy to work with. TempAuth is used primarily in Swift’s functional test environment and can be used in other test environments. It is not recommended to use TempAuth in a production system. For

For further details, Refer the following links:

1) https://docs.openstack.org/swift/latest/
2) https://docs.openstack.org/keystone/latest/
3) https://docs.openstack.org/swift/latest/overview_auth.html

This manual helps the user to use the Object Storage service **Swift** using the Authentication service **Keystone**.

## Section 2: Pre-requisites

- To use swift via keystone the following parameters are important:
    1) Username
    2) Password
    3) User Domain Name / User Domain Id
    4) Project Name / Tenant Name(Consider both to be same)
    5) Project Domain Name / Project Domain Id
    6) Authentication URL
    7) Storage URL

These parameters will be sent via the terminal in form of HTTP request to the swift server and thus will provide the required access to the user.
These parameter are available only for the users who have a swift account which is authenticated by keystone.

- For general basic usage, the following account credentials will be used:
  Username = swift
  Password = swift
  User Domain Name = default
  Project Name = service
  Project Domain Name = default
  Authentication URL = http://10.129.103.86:5000/v3/auth/tokens

  **Note:**
  ***The required Storage URL will be obtained later.**

## **Section 3: Connection Steps**:

1) We need to generate an Auth-Token for security purpose. To generate the token, we send a curl request to the keystone server which is running on the IP 10.129.103.86.

❖ The format of the request is as follows:

```
 curl -i \
  -H "Content-Type: application/json" \
  -d '
{ "auth": {
    "identity": {
      "methods": ["password"],
      "password": {
        "user": {
          "name": "<Username>",
          "domain": { "name": "<User Domain Name>" },
          "password": "<Password>"
        }
      }
    },
    "scope": {
      "project": {
        "name": "<Project Name>",
        "domain": { "name": "<Project Domain Name>" }
      }
    }
  }
}' \
  "<Authentication URL>" ; echo
```

❖ The Response will be as follows:

```
HTTP/1.1 201 Created
Date: Sat, 09 Jun 2018 05:24:24 GMT
Server: Apache/2.4.18 (Ubuntu)
X-Subject-Token: 5da5b479ef0b4cc5a7082ef63f3f1107
Vary: X-Auth-Token
X-Distribution: Ubuntu
x-openstack-request-id: req-2740fd50-03ce-49f8-a504-8cde9b00d746
Content-Length: 1730
Content-Type: application/json
```

```
{"token": {"methods": ["password"], "roles": [{"id":
"b77664f7e2b846b2898a96baf5799777", "name": "admin"}], "expires_at":
"2018-06-09T06:24:24.000000Z", "project": {"domain": {"id":
"b72bcb41252442069816707942b402b5", "name": "default"}, "id":
"b3f70be8acad4ec197e2b5edf48d9e5a", "name": "service"}, "catalog": [{"endpoints":
[{"region_id": "RegionOne", "url": "http://controller:8080/v1", "region": "RegionOne",
"interface": "admin", "id": "4e1e7473f4554afa8ef87f00efcf29f2"}, {"region_id":
"RegionOne", "url":
"http://controller:8080/v1/AUTH_b3f70be8acad4ec197e2b5edf48d9e5a", "region":
"RegionOne", "interface": "internal", "id": "6a74f782e9c247768e3707e7e9178755"},
{"region_id": "RegionOne", "url":
"http://controller:8080/v1/AUTH_b3f70be8acad4ec197e2b5edf48d9e5a", "region":
"RegionOne", "interface": "public", "id": "8a68ec3f836e406ca5b8f1dccda8c406"}],
"type": "object-store", "id": "c7e17a79e8db48e5a670cc115b01c15c", "name": "swift"},
{"endpoints": [{"region_id": "RegionOne", "url": "http://controller:5000/v3", "region":
"RegionOne", "interface": "public", "id": "9fbba597bc0a42b792bd92d2be00364b"},
{"region_id": "RegionOne", "url": "http://controller:35357/v3", "region": "RegionOne",
"interface": "admin", "id": "cfe5bebba97a48c19f551be5c64a0adf"}, {"region_id":
"RegionOne", "url": "http://controller:5000/v3", "region": "RegionOne", "interface":
"internal", "id": "fd3a930c71b34e46ae1b3e6789dec683"}], "type": "identity", "id":
"16b36db62f06483abd8386d839be9ce0", "name": "keystone"}], "user": {"domain": {"id":
"b72bcb41252442069816707942b402b5", "name": "default"}, "id":
"0572c648623a43238d2a416af16661fa", "name": "swift"}, "audit_ids":
["XxUio80FSQiI0bsRUZ0FXQ"], "issued_at": "2018-06-09T05:24:24.000000Z"}}
```

## Section 4: Accessing Swift

To access the swift account, we need to send a curl request which is of the following form:
**curl -v -H “X-AUTH-TOKEN: <token-from-x-auth-token-above>”**
**<url-from-x-storage-url-above>**

Here the user needs to enter the value of X-Auth-Token and X-Storage-URL (obtained above) in the required position.

The request for the above case will be:

```
curl -v -H “X-AUTH-TOKEN:5da5b479ef0b4cc5a7082ef63f3f1107”
http://10.129.103.86:8080/v1/AUTH_b3f70be8acad4ec197e2b5edf48d9e5a
```

The response to the request is of the form:

```
*   Trying 10.129.103.86...
* Connected to 10.129.103.86 (10.129.103.86) port 8080 (#0)
> GET /v1/AUTH_b3f70be8acad4ec197e2b5edf48d9e5a HTTP/1.1
> Host: 10.129.103.86:8080
> User-Agent: curl/7.47.0
> Accept: */*
> X-AUTH-TOKEN:91c6b7f50123439187bf9befacef1ed1
>
< HTTP/1.1 200 OK
< X-Account-Storage-Policy-Gold-Bytes-Used: 679692
< Content-Length: 34
< X-Account-Storage-Policy-Gold-Object-Count: 6
< X-Account-Object-Count: 6
< X-Timestamp: 1528451672.05368
< X-Account-Storage-Policy-Gold-Container-Count: 2
< X-Account-Bytes-Used: 679692
< X-Account-Container-Count: 2
< Content-Type: text/plain; charset=utf-8
< Accept-Ranges: bytes
< x-account-project-domain-id: b72bcb41252442069816707942b402b5
< X-Trans-Id: tx25dac9031d9640598d750-005b1b6a11
< X-Openstack-Request-Id: tx25dac9031d9640598d750-005b1b6a11
< Date: Sat, 09 Jun 2018 05:48:01 GMT
<
swift-container1
swift-container2
* Connection #0 to host 10.129.103.86 left intact
```

This request contains the details of the storage containers and object associated with the **swift** account. It tells about the no of containers and objects present in the account and also displays the list of containers in the account.

In this case we have 2 containers, **swift-container1** and **swift-container2** in our swift account.

## Section 5: Basic Curl Commands to Access Various Features of Swift

(Note: Any curl URL uses GET method by default.)

1) Display List of containers:

```
curl -v -H “X-AUTH-TOKEN:<X-Auth-Token>”  <X-Storage-URL>
```

2) Display List of objects in a container:

```
curl -v -H “X-AUTH-TOKEN:<X-Auth-Token>”  <X-Storage-URL>/<container name>
```

3) Display the contents of the object:

```
curl -v -H “X-AUTH-TOKEN:<X-Auth-Token>”  <X-Storage-URL>/<container name>/<object name>
```

4) Add a new Container to the account:

```
curl -v -H "X-AUTH-TOKEN:<X-Auth-Token>" -X PUT <X-Storage-URL>/< new container’s name>
```

5) Add a new object to a container:

```
curl -v -H “X-AUTH-TOKEN:<X-Auth-Token>” -X PUT <X-Storage-URL>/<container name>/ -T
<file-name>
```

6) Delete a container:

```
curl -v -H “X-AUTH-TOKEN:<X-Auth-Token>” -X DELETE  <X-Storage-URL>/<container name>
```

7) Delete an object from a container:

```
curl -v -H “X-AUTH-TOKEN:<X-Auth-Token>” -X DELETE  <X-Storage-URL>/<container name>/<object
name>
```

8) Download an object from a container:

```
curl -v -H “X-AUTH-TOKEN:<X-Auth-Token>”  <X-Storage-URL>/<container name>/<object name> -o
<name-you-want-to-give-to-file-on-pc>
```

More detailed list of Openstack Object Storage API examples can be found on the following page:
https://developer.openstack.org/api-ref/object-store/

**TROUBLESHOOT:**

If you get authorisation failed error or content access denied error, kindly regenerate the Auth Token and use the new token accordingly.

For further queries, Contact :

Utkarsh Shukla (shukla.utkarsh10@gmail.com) or Jay Modi (jay.m.btechi15@ahduni.edu.in)