

Object Storage Usage

An Object Storage is a collection of objects that could be shared and secure among a network storage. Using this model, the Data Space is allocated inside the object store instead of a traditional piece of software like a File System.

One of the most important parts of an Object store is the metadata...

At IFCA, the Object Storage interface is implemented through [Ceph Rados Gateway](#), in contrast with the Object Storage Swift service available OpenStack. However, the Ceph Object Storage is "fully" compatible with the Swift and S3 API. Following this line, the data is stored inside the Ceph Storage Cluster that is accessible through the Ceph Object Gateway (rados gw) using HTTP requests or from the Ceph Block Device Client (cinder).

[blocked URL](#)

Manage the Object Storage using Swift API

Help for interacting with the swift API using HTTP requests

1. Create token in OpenStack using this script.

```
export OS_AUTH_URL=https://keystone.cloud.ifca.es:5000/v3

# With the addition of Keystone we have standardized on the term **tenant**
# as the entity that owns the resources.
export OS_TENANT_ID=dfd23dc772fe4f76ad34d0013e44a745
export OS_TENANT_NAME=ifca.es:mods
export OS_USER_DOMAIN_NAME=IFCA
export OS_IDENTITY_API_VERSION=3

# In addition to the owning entity (tenant), openstack stores the entity
# performing the action as the **user**.
export OS_USERNAME=aidaph

# With Keystone you pass the keystone password.
read -s -p Password: pwd
export OS_PASSWORD=$pwd

openstack --os-username=$OS_USERNAME --os-user-domain-name=$OS_USER_DOMAIN_NAME --os-
password="$OS_PASSWORD" --os-project-name=$OS_TENANT_NAME --os-project-id=$OS_TENANT_ID token
issue
```

2. [Object Storage API](#)

- a. Show object metadata using the HEAD sentence

```
@nova:~$ curl -s --head -H "X-Auth-Token: $TOKEN" https://api.cloud.ifca.es:8080/swift/v1
/SparkTest/country_vaccinations.csv
HTTP/1.1 200 OK
Content-Length: 1075352
Accept-Ranges: bytes
Last-Modified: Wed, 30 Mar 2022 10:17:13 GMT
X-Timestamp: 1648635433.45941
etag: 391086e2094ca2c92106fd1cf39765b6
X-Object-Meta-Orig-Filename: country_vaccinations.csv
X-Trans-Id: tx00000000000000000000f4c-006246d490-44232c6-RegionOne
X-Openstack-Request-Id: tx00000000000000000000f4c-006246d490-44232c6-RegionOne
Content-Type: text/csv
Date: Fri, 01 Apr 2022 10:31:45 GMT
```

- b. create or update an object or its metadata using POST

```
@nova:~$ curl -X POST -H "X-Auth-Token: $TOKEN" -H "X-Object-Meta-Orig-Filename: contry-
vaccinations-test.csv" https://api.cloud.ifca.es:8080/swift/v1/SparkTest/country_vaccinations.csv
@nova:~$ curl -s --head -H "X-Auth-Token: $TOKEN " https://api.cloud.ifca.es:8080/swift/v1
/SparkTest/country_vaccinations.csv
HTTP/1.1 200 OK
Content-Length: 1075352
X-Timestamp: 1648810529.55442
X-Object-Meta-Orig-Filename: contry-vaccinations-test.csv
[...]
Date: Fri, 01 Apr 2022 10:55:35 GMT
```

To create or update custom metadata, use the X-Object-Meta-name header, where name is the name of the metadata item. The **X-Object-Meta-Orig-Filename** is an example of metadata field where **Orig-Filename** is the name of the metadata object. In addition to the custom metadata, you can update the Content-Type, Content-Encoding, Content-Disposition, and X-Delete-At system metadata items. However you cannot update other system metadata, such as Content-Length or Last-Modified.

Ceph Object Gateway Swift integration with Delta Lake and Apache Hadoop/Spark

This current implementation is under testing at IFCA. An issue was open at [delta#950](#)

For using the Delta Lake storage under Ceph Rados Gateway, an implementation of ceph rados gw for delta using OpenStack Swift API will be included in the next releases of Delta. This code is officially [under testing](#), and we check if it works at IFCA.

The test can be launched through a Spark interactive session. First of all, the dependencies must be download before starting the session. These includes the **Delta-core** and the implementation of ceph with hadoop called **hadoop-cephrgw**. Apart from that, some variables are mandatory in the hadoop configuration. The variables can be set in the *core-site.xml* inside the etc/hadoop path of your current \$HOME of Hadoop or you can pass it directly from the terminal. This is an example of the command:

```
hadoopuser@mods-hadoop:~$ export _JAVA_OPTIONS='-Djdk.tls.maxCertificateChainLength=15'; \
spark-shell --packages io.delta:delta-core_2.12:1.1.0,io.github.nanhu-lab:hadoop-cephrgw:1.0.1,org.apache.spark:
spark-hadoop-cloud_2.12:3.2.1\
--conf spark.hadoop.fs.ceph.username=tester \
--conf spark.hadoop.fs.ceph.password=testXXX \
--conf spark.hadoop.fs.ceph.uri=https://api.cloud.ifca.es:8080/swift/v1/
--conf spark.hadoop.fs.s3a.connection.ssl.enabled=false
--conf spark.delta.logStore.class=org.apache.spark.sql.delta.storage.S3SingleDriverLogStore
--conf spark.hadoop.fs.ceph.impl=org.apache.hadoop.fs.ceph.rgw.CephStoreSystem
```

S3 Configuration

TODO

- [Two-Factor Authentication \(2FA\)](#)
- [Object Storage Usage](#)