

Data Injection: CTA

Authors: Frederic Gillardo and Berkay Turk

All codes are pushed in the following git repository: <https://gitlab.in2p3.fr/CTA-LAPP/cta-rucio-client>

Step 1 (30/09/2020)

Set up a rucio-client with X.509 authentication using singularity image

We use the following document: https://docs.google.com/document/d/11uCGqrP_TlofAsbm_Kylz2sjsvV4kEv5dLeTbOsKfxl/edit#heading=h.qasaz2piu2fu

Steps performed:

1. Remove passphrase from the key :

```
openssl pkcs12 -in mycert.p12 -clcerts -nokeys -out client.crt
openssl pkcs12 -in mycert.p12 -nocerts -nodes -out client.key
chmod 400 client.crt
chmod 400 client.key
```

2. Edit Rucio.cfg (note: Rucio.cfg is not populated with environment variables, therefore we edited it manually)

```
[client]
rucio_host = https://escape-rucio.cern.ch:32300/
auth_host = https://escape-rucio.cern.ch:32301
auth_type = userpass
ca_cert = /opt/rucio/etc/CERNCertificationAuthority.pem
account = gillardo
request_retries = 3
auth_type = x509
client_cert = /opt/rucio/etc/client.crt
client_key = /opt/rucio/etc/client.key
vo = escape
```

Note: for lapp-esc02, the webdav server does not support checksum, therefore, we disabled the checksum for this specific RSE within RUCIO.

```
rucio-admin rse set-attribute --rse LAPP-WEBDAV --key verify_checksum --value False
rucio-admin rse info LAPP-WEBDAV
```

3. Execute singularity image

```
singularity run -B ${HOME}/.rucio:/opt/rucio/etc -B ${HOME}/.globus/
client.crt:/opt/rucio/etc/client.crt -B ${HOME}/.globus/client.key:/opt/
rucio/etc/client.key rucio-cli.simg
```

4. initialize the voms proxy

```
voms-proxy-init -voms escape
```

Upload a file, even mock if real data is not available for the time being:

1. Create the scope

```
rucio-admin scope add --account gillardo --scope CTA_LAPP_FREDERIC
```

2. Upload a file

```
rucio -v upload --rse LAPP-WEBDAV --scope CTA_LAPP_FREDERIC  
cta_data1.hdf5
```

3. add replication rule

```
rucio add-rule CTA_LAPP_FREDERIC:cta-data1.hdf5 1 LAPP-DCACHE
```

4. Check rule status

```
rucio list-rules CTA_LAPP_FREDERIC:cta-data1.hdf5
```

ID	REPL/STUCK]	RSE_EXPRESSION	ACCOUNT	SCOPE:NAME	COPIES	EXPIRES (UTC)	CREATED (UTC)	STATE[OK/
be7ab08aa67b4b409297f33c34cf6895		LAPP-WEBDAV	gillardo	CTA_LAPP_FREDERIC:cta-data1.hdf5	1	2020-10-05 09:41:49		OK[1/0/0]
50e81b6d15e844b0bccd052456df2785		LAPP-DCACHE	gillardo	CTA_LAPP_FREDERIC:cta-data1.hdf5	1	2020-10-05 09:46:06		

ID	REPL/STUCK]	RSE_EXPRESSION	ACCOUNT	SCOPE:NAME	COPIES	EXPIRES (UTC)	CREATED (UTC)	STATE[OK/
be7ab08aa67b4b409297f33c34cf6895		LAPP-WEBDAV	gillardo	CTA_LAPP_FREDERIC:cta-data1.hdf5	1	2020-10-05 09:41:49		OK[1/0/0]
50e81b6d15e844b0bccd052456df2785		LAPP-DCACHE	gillardo	CTA_LAPP_FREDERIC:cta-data1.hdf5	1	2020-10-05 09:46:06		OK[1/0/0]

5. Download the file locally:

We specify from which rse (lapp-dcache) we want to perform the download:

```
rucio download --rse LAPP-WEBDAV CTA_LAPP_FREDERIC:cta-data1.hdf5 --rse LAPP-DCACHE
```

Step 2 (14/10/2020)

Prerequisite:

1. Set a QOS SAFE for LAPP-DCACHE

```
rucio-admin rse set-attribute --rse LAPP-DCACHE --key QOS --value SAFE
```

2. Set a QOS FAST for LAPP-WEBDAV

```
rucio-admin rse set-attribute --rse LAPP-WEBDAV --key QOS --value FAST
```

3. Greedy option for LAPP-WEBDAV (needs to be reviewed)

```
rucio-admin rse set-attribute --key greedyDeletion --value True --rse LAPP-WEBDAV
```

CTA R0 data management

1. Create dataset

```
rucio add-dataset CTA_LAPP_FREDERIC:CTA_OBSID_1
```

2. Upload 2 files

```
rucio -v upload --lifetime 86400 --rse LAPP-WEBDAV --scope CTA_LAPP_FREDERIC cta_data_DL0_ObsId1_TelId1.hdf5
```

```
rucio -v upload --lifetime 86400 --rse LAPP-WEBDAV --scope CTA_LAPP_FREDERIC cta_data_ObsId1_TelId2.hdf5
```

3. attach files to dataset

```
rucio attach CTA_LAPP_FREDERIC:CTA_OBSID_1 CTA_LAPP_FREDERIC:cta_data_ObsId1_TelId1.hdf5
```

```
rucio attach CTA_LAPP_FREDERIC:CTA_OBSID_1 CTA_LAPP_FREDERIC:cta_data_ObsId1_TelId2.hdf5
```

4. add replication rule for 2 sites

```
rucio add-rule --lifetime 86400 CTA_LAPP_FREDERIC:CTA_OBSID_1 1 'SITE=LAPP&QOS=SAFE'
```

```
rucio add-rule --lifetime 86400 CTA_LAPP_FREDERIC:CTA_OBSID_1 1 'SITE=IN2P3&QOS=CHEAP-ANALYSIS'
```

5. Check status of the rules

```
rucio list-rules CTA_LAPP_FREDERIC:CTA_OBSID_1
```

```
rucio list-file-replicas CTA_LAPP_FREDERIC:cta_data_ObsId1_TelId1.hdf5
```

6. When rule is executed, delete file on LAPP-WEBDAV

```
rucio add-rule --lifetime 86400 CTA_LAPP_FREDERIC:CTA_OBSID_1 1 LAPP-WEBDAV
```

```
rucio delete-rule CTA_LAPP_FREDERIC:CTA_OBSID_1 --rse LAPP-WEBDAV --purge-replicas
```

Bash implementation of Ingest:

```
#!/bin/bash

ObsID=$1
RSE_onsite=LAPP-WEBDAV
scope=CTA_LAPP_FREDERIC

echo "generating files for ObsID $ObsID"
file1=./generatedFiles/cta_data_DL0_ObsId${ObsID}_TelId1.hdf5
file2=./generatedFiles/cta_data_DL0_ObsId${ObsID}_TelId2.hdf5

head -c 1M </dev/urandom > $file1
head -c 1M </dev/urandom > $file2

datasetName=${scope}:CTA_OBSID_${ObsID}
rucio add-dataset $datasetName

rucio -v upload --lifetime 86400 --rse $RSE_onsite --scope $scope $file1
rucio -v upload --lifetime 86400 --rse $RSE_onsite --scope $scope $file2

fqdnFile1=${scope}:cta_data_DL0_ObsId${ObsID}_TelId1.hdf5
fqdnFile2=${scope}:cta_data_DL0_ObsId${ObsID}_TelId2.hdf5

rucio attach $datasetName $fqdnFile1
rucio attach $datasetName $fqdnFile2

rucio add-rule --lifetime 86400 $datasetName 1 'SITE=LAPP&QOS=SAFE'
rucio add-rule --lifetime 86400 $datasetName 1 'SITE=IN2P3&QOS=CHEAP-ANALYSIS'

#rucio list-rules $datasetName
#rucio list-file-replicas $fqdnFile1
#rucio list-file-replicas $fqdnFile2

function listRuleAndWaitUntilItIsFinished () {
    #ret = `rucio list-rules $datasetName`
    ret=REPLICATING
    while [[ $ret == *"REPLICATING"* ]]; do
        echo "$datasetName is still replicating"
        sleep 1s
        ret=$(rucio list-rules ${datasetName})
        echo "rucio list return : ${ret}"
    done
    return 0
}
```

Step 3 (28/10/2020)

Note : the retrieve part is packaged as an singularity container, and is triggered within htCondor Job manager

1. Download all files from a given dataset

```
rucio get --dir ./downloadedFiles/ CTA_LAPP_FREDERIC:CTA_OBSID_2
```

2. perform computation

CTA specific code ...

3. Upload result back to the local rse node from the dataLake

```
rucio -v upload --lifetime 86400 --rse LAPP-WEBDCACHE --scope
CTA_LAPP_FREDERIC cta_data_DL3_ObsId1.hdf5
```

Bash implementation of Retrieve:

```
#!/bin/bash
```

```
ObsID=$1
scope=CTA_LAPP_FREDERIC
downloadedDir=downloaded-files
datasetName=${scope}:CTA_OBSID_${ObsID}
fileDL3=./generatedFiles/cta_data_DL3_ObsId${ObsID}.hdf5
RSE_scienceArchive=LAPP-DCACHE
```

```
mkdir $downloadedDir
rucio download --dir $downloadedDir $datasetName
```

```
for file in ./${downloadedDir}/CTA_OBSID_${ObsID}/*
do
    echo "    reading headers from $file"
    head -c 16 $file
    echo "
    starting (sleep) process "
    sleep 10s
done
```

```
head -c 100K </dev/urandom > $fileDL3
rucio -v upload --lifetime 86400 --rse $RSE_scienceArchive --scope $scope
$fileDL3
```

MISSING steps for demo of November 2020

- HtCondor implementation

##not working

```
rucio --verbose add-did-meta --did CTA_LAPP_FREDERIC:CTA_OBSID_1--key ObsID --value A
Rucio get-did-meta CTA_LAPP_FREDERIC:CTA_OBSID_1
```

CTA use case blocker

- OpenID support
- Metadata management:

- Set any type of metadata during the upload
- Query the file thanks to a query mechanism using metadata values.