ESCAPE

European Science Cluster of Astronomy & Particle physics ESFRI research Infrastructures

Testing Frameworks for the Datalake

Rohini Joshi Rizart Dona

SKA CERN

December 9, 2020 - 2nd ESCAPE WP2/DIOS Workshop

ESCAPE - The European Science Cluster of Astronomy & Particle Physics ESFRI Research Infrastructures has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement n° 824064.

Overview

- The Datalake Stack
- Continuous Testing
- Continuous Testing - GFAL
- Continuous Testing - FTS
- Continuous Testing - Rucio
- Future Work
- References

Funded by the European Union's Horizon 2020 - Grant N° 824064

The Datalake Stack

- The Datalake currently employs the following software stack that deals with data transfer/access

  RUCIO $\longrightarrow$ FTS $\longrightarrow$ GFAL

- **GFAL** (Grid File Access Library), a multi-protocol data management library providing an abstraction layer of the grid storage system complexity (supports protocols like GridFTP, Http & Root)
- **FTS** (File Transfer Service), open source software to transfer data reliably and at large scale between storage systems
- **Rucio,** the data orchestration service, a scalable policy-driven scientific data management system that can work with large amounts of data, this is the service that users basically interact with

---

- Rucio uses both FTS and GFAL to perform data **transfer**, data **injection** and data **download** operations
- FTS uses GFAL to perform the actual transfer on the storage level, it uses **TPC (**Third Party Copy**)** to transfer data between two storages that support the same protocol by using a direct link between the two, parallel transfers **optimization** is also achieved based on the network state
- GFAL is the lowest level component which interacts directly with the **storage** (I/O operations on the actual filesystem)

Funded by the European Union's Horizon 2020 - Grant N° 824064

Continuous Testing

- In order to make sure that all three components are functioning properly we have **continuous testing** in place
- Separate tests target each component individually and explore scenarios that involve both **functional testing** as well as **stress testing**
- Configurable software has been developed and deployed to make the process **automatic**
- All test results are pushed and **visualized** in the equivalent Grafana dashboards that consist the monitoring of the Datalake

Funded by the European Union's Horizon 2020 - Grant N° 824064

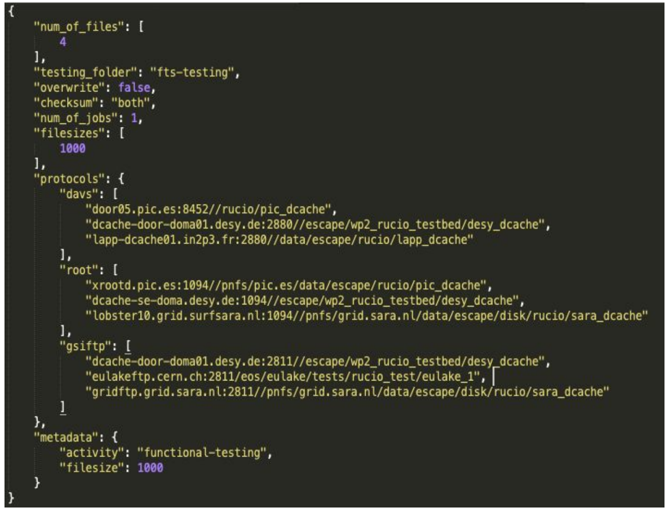
Continuous testing - GFAL

- All RSES (Rucio) consist of one or more endpoints that are associated with a supported protocol
- There are three types of operations that are being done concerning GFAL **functional testing**
  - **Upload** of a file that is a few bytes long to all the endpoints of all RSEs
  - **Download** of the file that was uploaded in the previous step
  - **Deletion** of the file that was uploaded in the first step
- This flow examines the **basic data operations** one can perform on the lowest level of the Datalake
  - Per RSE per endpoint results
  - Automatically pushing results on an Elasticsearch datasource, custom code
- Integrated with **CRIC**
  - Automatically fetching the RSEs configuration before each run
- Python code, deployed inside a container in a Kubernetes cluster @ CERN
  - Testing frequency → every 2 minutes

Funded by the European Union's Horizon 2020 - Grant N° 824064

Continuous testing - FTS (1/2)

- In this case, the same endpoints as in the GFAL testing are examined
- Goal is to trigger **TPC** transfers between all possible endpoint pairs that participate in the Datalake
- The flow is the following
  - The toolkit reads from a **configuration file** all endpoint pairs that are to be tested and information like number of jobs, filesizes, number of files per job, checksum verification and other. Custom **metadata** attributes are also supported
  - Testing folders are being setup if needed, automatic detection of **problematic** endpoints that will be excluded from the testing
  - For each pair of compatible endpoints an FTS transfer will be triggered
    1) Check source for existing files, generate & upload ad-hoc if needed
    2) Trigger FTS job, **asynchronous** action
    3) Keep in a dictionary all triggered job ids
  - Wait for all jobs to finish, when done **delete** the files that were transferred on the destination endpoint (ensures no quota is exceeded)



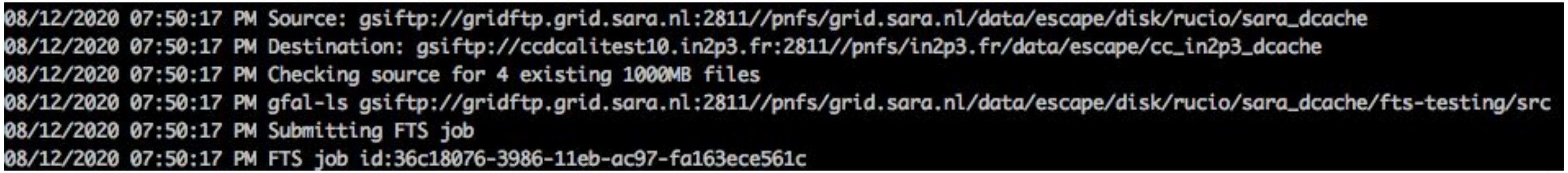
```
{
    "num_of_files": [
        4
    ],
    "testing_folder": "fts-testing",
    "overwrite": false,
    "checksum": "both",
    "num_of_jobs": 1,
    "filesizes": [
        1000
    ],
    "protocols": {
        "davs": [
            "door05.pic.es:8452//rucio/pic_dcache",
            "dcache-door-doma01.desy.de:2880//escape/wp2_rucio_testbed/desy_dcache",
            "lapp-dcache01.in2p3.fr:2880//data/escape/rucio/lapp_dcache"
        ],
        "root": [
            "xrootd.pic.es:1094//pnfs/pic.es/data/escape/rucio/pic_dcache",
            "dcache-se-doma.desy.de:1094//escape/wp2_rucio_testbed/desy_dcache",
            "lobster10.grid.surfsara.nl:1094//pnfs/grid.sara.nl/data/escape/disk/rucio/sara_dcache"
        ],
        "gsiftp": [
            "dcache-door-doma01.desy.de:2811//escape/wp2_rucio_testbed/desy_dcache",
            "eulakeftp.cern.ch:2811/eos/eulake/tests/rucio_test/eulake_1",
            "gridftp.grid.sara.nl:2811//pnfs/grid.sara.nl/data/escape/disk/rucio/sara_dcache"
        ]
    },
    "metadata": {
        "activity": "functional-testing",
        "filesize": 1000
    }
}
```

- Extensive **error handling**, flow will continue even if endpoints fail mid-test
- fts-analysis-datalake repo, python code

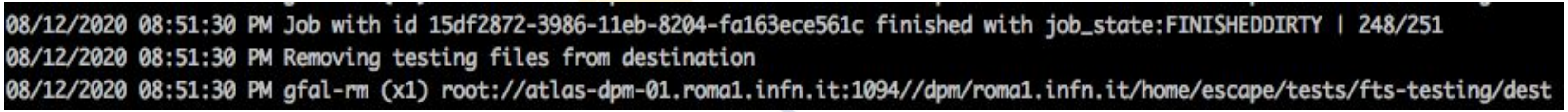
Funded by the European Union's Horizon 2020 - Grant N° 824064

Continuous testing - FTS (2/2)

- Current configuration includes two tests
  - **1MB** files, 4 files per job, all Datalake endpoints participate
  - **1GB** files, 4 files per job, all Datalake endpoints participate except for some testing ones with low quota
- Testing results are automatically pushed from the FTS server to an **InfluxDB** & **Elasticsearch** datasource (all systems maintained by the relevant teams @ CERN)
- Deployed inside a container in a Kubernetes cluster @ CERN
  - Testing frequency ⟶   every 30 minutes



```
08/12/2020 07:50:17 PM Source: gsiftp://gridftp.grid.sara.nl:2811//pnfs/grid.sara.nl/data/escape/disk/rucio/sara_dcache
08/12/2020 07:50:17 PM Destination: gsiftp://ccdcalitest10.in2p3.fr:2811//pnfs/in2p3.fr/data/escape/cc_in2p3_dcache
08/12/2020 07:50:17 PM Checking source for 4 existing 1000MB files
08/12/2020 07:50:17 PM gfal-ls gsiftp://gridftp.grid.sara.nl:2811//pnfs/grid.sara.nl/data/escape/disk/rucio/sara_dcache/fts-testing/src
08/12/2020 07:50:17 PM Submitting FTS job
08/12/2020 07:50:17 PM FTS job id:36c18076-3986-11eb-ac97-fa163ece561c
```



```
08/12/2020 08:51:30 PM Job with id 15df2872-3986-11eb-8204-fa163ece561c finished with job_state:FINISHEDDIRTY | 248/251
08/12/2020 08:51:30 PM Removing testing files from destination
08/12/2020 08:51:30 PM gfal-rm (x1) root://atlas-dpm-01.roma1.infn.it:1094//dpm/roma1.infn.it/home/escape/tests/fts-testing/dest
```

Funded by the European Union's Horizon 2020 - Grant N° 824064

Continuous testing - Rucio (1/2)

- rucio-analysis: extensible python3 framework for repeated (hopefully reproducible) yaml-based data lake tests
- Docker image can be built locally, or pulled from Dockerhub here (master branch updates only)
- Crons and wrapper scripts for running the tests are stored in the repo here
  - Hourly tests running for upload and replication across all RSEs* with 100 KB files, 1 hour lifetimes in scope SKA_SKAO_BARNSLEY-testing
  - ES database sync running every 5 mins
  - Daily report running a slack webhook
- Contributions welcome! stub test module and yaml file to help get started with adding new tests

---

- Simple bash script also used for fast ad-hoc functional testing, uploads to RSES, replica creation between all pairs
  - Deployed inside a container in a Kubernetes cluster @ CERN
  - Testing frequency → every 2 minutes
  - Filesizes → 1GB

Funded by the European Union's Horizon 2020 - Grant N° 824064

Continuous testing - Rucio (2/2)

- Sample run of a test shown on the right
- Corresponding yaml file below

Rucio account

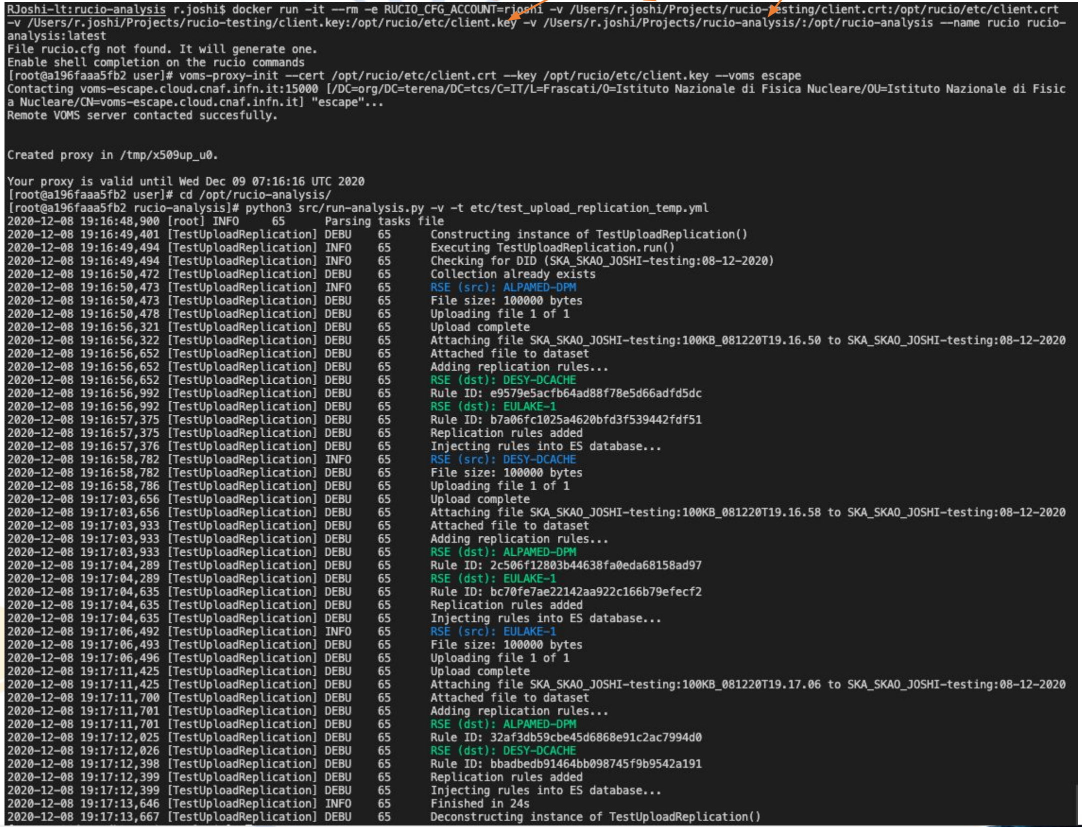
Bind credentials in the container

Bind a copy of the repo to see/persist yaml files

Start docker container

Initialise voms proxy

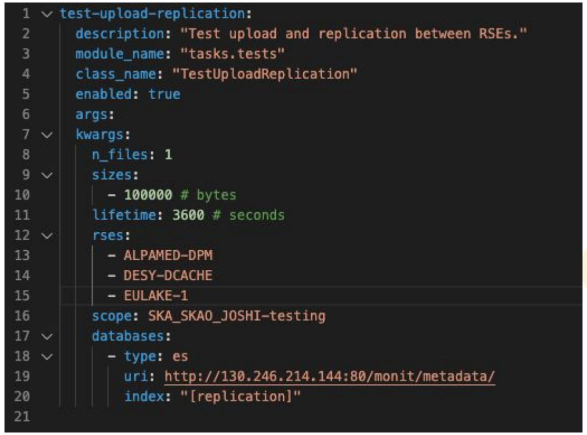
Run test



```
RJoshi-lt:rucio-analysis r.joshi$ docker run -it --rm -e RUCIO_CFG_ACCOUNT=rjoshi -v /Users/r.joshi/Projects/rucio-testing/client.crt:/opt/rucio/etc/client.crt
-v /Users/r.joshi/Projects/rucio-testing/client.key:/opt/rucio/etc/client.key -v /Users/r.joshi/Projects/rucio-analysis/:/opt/rucio-analysis --name rucio rucio-
analysis:latest
File rucio.cfg not found. It will generate one.
Enable shell completion on the rucio commands
[root@a196faaa5fb2 user]# voms-proxy-init --cert /opt/rucio/etc/client.crt --key /opt/rucio/etc/client.key --voms escape
Contacting voms-escape.cloud.cnaf.infn.it:15000 [/DC=org/DC=terena/DC=tcs/C=IT/L=Frascati/O=Istituto Nazionale di Fisica Nucleare/OU=Istituto Nazionale di Fisic
a Nucleare/CN=voms-escape.cloud.cnaf.infn.it] "escape"...
Remote VOMS server contacted succesfully.


Created proxy in /tmp/x509up_u0.

Your proxy is valid until Wed Dec 09 07:16:16 UTC 2020
[root@a196faaa5fb2 user]# cd /opt/rucio-analysis/
[root@a196faaa5fb2 rucio-analysis]# python3 src/run-analysis.py -v -t etc/test_upload_replication_temp.yml
2020-12-08 19:16:48,900 [root] INFO     65      Parsing tasks file
2020-12-08 19:16:49,401 [TestUploadReplication] DEBU    65      Constructing instance of TestUploadReplication()
2020-12-08 19:16:49,494 [TestUploadReplication] INFO    65      Executing TestUploadReplication.run()
2020-12-08 19:16:49,494 [TestUploadReplication] INFO    65      Checking for DID (SKA_SKAO_JOSHI-testing:08-12-2020)
2020-12-08 19:16:50,472 [TestUploadReplication] DEBU    65      Collection already exists
2020-12-08 19:16:50,473 [TestUploadReplication] INFO    65      RSE (src): ALPAMED-DPM
2020-12-08 19:16:50,473 [TestUploadReplication] DEBU    65      File size: 100000 bytes
2020-12-08 19:16:50,478 [TestUploadReplication] DEBU    65      Uploading file 1 of 1
2020-12-08 19:16:56,321 [TestUploadReplication] DEBU    65      Upload complete
2020-12-08 19:16:56,322 [TestUploadReplication] DEBU    65      Attaching file SKA_SKAO_JOSHI-testing:100KB_081220T19.16.50 to SKA_SKAO_JOSHI-testing:08-12-2020
2020-12-08 19:16:56,652 [TestUploadReplication] DEBU    65      Attached file to dataset
2020-12-08 19:16:56,652 [TestUploadReplication] DEBU    65      Adding replication rules...
2020-12-08 19:16:56,652 [TestUploadReplication] DEBU    65      RSE (dst): DESY-DCACHE
2020-12-08 19:16:56,992 [TestUploadReplication] DEBU    65      Rule ID: e9579e5acfb64ad88f78e5d66adfd5dc
2020-12-08 19:16:56,992 [TestUploadReplication] DEBU    65      RSE (dst): EULAKE-1
2020-12-08 19:16:57,375 [TestUploadReplication] DEBU    65      Rule ID: b7a06fc1025a4620bfd3f539442fdf51
2020-12-08 19:16:57,375 [TestUploadReplication] DEBU    65      Replication rules added
2020-12-08 19:16:57,376 [TestUploadReplication] DEBU    65      Injecting rules into ES database...
2020-12-08 19:16:58,782 [TestUploadReplication] INFO    65      RSE (src): DESY-DCACHE
2020-12-08 19:16:58,782 [TestUploadReplication] DEBU    65      File size: 100000 bytes
2020-12-08 19:16:58,786 [TestUploadReplication] DEBU    65      Uploading file 1 of 1
2020-12-08 19:17:03,656 [TestUploadReplication] DEBU    65      Upload complete
2020-12-08 19:17:03,656 [TestUploadReplication] DEBU    65      Attaching file SKA_SKAO_JOSHI-testing:100KB_081220T19.16.58 to SKA_SKAO_JOSHI-testing:08-12-2020
2020-12-08 19:17:03,933 [TestUploadReplication] DEBU    65      Attached file to dataset
2020-12-08 19:17:03,933 [TestUploadReplication] DEBU    65      Adding replication rules...
2020-12-08 19:17:03,933 [TestUploadReplication] DEBU    65      RSE (dst): ALPAMED-DPM
2020-12-08 19:17:04,289 [TestUploadReplication] DEBU    65      Rule ID: 2c506f12803b44638fa0eda68158ad97
2020-12-08 19:17:04,289 [TestUploadReplication] DEBU    65      RSE (dst): EULAKE-1
2020-12-08 19:17:04,635 [TestUploadReplication] DEBU    65      Rule ID: bc70fe7ae22142aa922c166b79efecf2
2020-12-08 19:17:04,635 [TestUploadReplication] DEBU    65      Replication rules added
2020-12-08 19:17:04,635 [TestUploadReplication] DEBU    65      Injecting rules into ES database...
2020-12-08 19:17:06,492 [TestUploadReplication] INFO    65      RSE (src): EULAKE-1
2020-12-08 19:17:06,493 [TestUploadReplication] DEBU    65      File size: 100000 bytes
2020-12-08 19:17:06,496 [TestUploadReplication] DEBU    65      Uploading file 1 of 1
2020-12-08 19:17:11,425 [TestUploadReplication] DEBU    65      Upload complete
2020-12-08 19:17:11,425 [TestUploadReplication] DEBU    65      Attaching file SKA_SKAO_JOSHI-testing:100KB_081220T19.17.06 to SKA_SKAO_JOSHI-testing:08-12-2020
2020-12-08 19:17:11,700 [TestUploadReplication] DEBU    65      Attached file to dataset
2020-12-08 19:17:11,701 [TestUploadReplication] DEBU    65      Adding replication rules...
2020-12-08 19:17:11,701 [TestUploadReplication] DEBU    65      RSE (dst): ALPAMED-DPM
2020-12-08 19:17:12,025 [TestUploadReplication] DEBU    65      Rule ID: 32af3db59cbe45d6868e91c2ac7994d0
2020-12-08 19:17:12,026 [TestUploadReplication] DEBU    65      RSE (dst): DESY-DCACHE
2020-12-08 19:17:12,398 [TestUploadReplication] DEBU    65      Rule ID: bbadbedb91464bb098745f9b9542a191
2020-12-08 19:17:12,399 [TestUploadReplication] DEBU    65      Replication rules added
2020-12-08 19:17:12,399 [TestUploadReplication] DEBU    65      Injecting rules into ES database...
2020-12-08 19:17:13,646 [TestUploadReplication] INFO    65      Finished in 24s
2020-12-08 19:17:13,667 [TestUploadReplication] DEBU    65      Deconstructing instance of TestUploadReplication()
```



```yaml
test-upload-replication:
  description: "Test upload and replication between RSEs."
  module_name: "tasks.tests"
  class_name: "TestUploadReplication"
  enabled: true
  args:
  kwargs:
    n_files: 1
    sizes:
      - 100000 # bytes
    lifetime: 3600 # seconds
    rses:
      - ALPAMED-DPM
      - DESY-DCACHE
      - EULAKE-1
    scope: SKA_SKAO_JOSHI-testing
    databases:
      - type: es
        uri: http://130.246.214.144:80/monit/metadata/
        index: "[replication]"

```

Funded by the European Union's Horizon 2020 - Grant N° 824064

Future work

- FTS toolkit
  - Integration with **CRIC**, automatic creation of endpoint pairs in the configuration file
  - More flexible & **granular** configuration
  - Increase the current configuration to test **bigger** filesizes (e.g. 5GB+)
- Rucio-analysis
  - Some technical debt, code clean-up
  - Stable master branch, that reflects the live config deployed on the test instance

Funded by the European Union's Horizon 2020 - Grant N° 824064

References

- FTS, https://fts.web.cern.ch/fts/
- GFAL, https://dmc-docs.web.cern.ch/dmc-docs/gfal2/gfal2.html
- Rucio, https://rucio.cern.ch/
- GFAL testing software, https://github.com/ESCAPE-WP2/Utilities-and-Operations-Scripts/tree/master/gfal-sam-testing
- FTS testing software, https://github.com/ESCAPE-WP2/fts-analysis-datalake
- Rucio testing software, https://github.com/ESCAPE-WP2/rucio-analysis
- Previous presentation on the rucio-analysis framework

  https://www.dropbox.com/s/x8twzelnyydnkwn/ESCAPE%20WP2%20Meeting%20-%20211020.pdf?dl=0

Funded by the European Union's Horizon 2020 - Grant N° 824064

Thank you!

Questions?

r.joshi@skatelescope.org

Funded by the European Union's
Horizon 2020 - Grant N° 824064