Datalake data access: CMS experiment use case for open and embargoed data

Diego Ciangottini

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.

Outline

- CMS use case activity
  - Compute provisioning
  - Data access
- AuthN/Z
  - Summary
  - Granularity
- Data Access setup:
  - The XRooTD way
  - The HTTP way
- Data import: RUCIO configuration
- Tests summary

Thanks to AndreaC. and ArisF. respectively for IAM based authN/Z and Rucio support

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

CMS use case activity

- The objective: produce a working prototype for CMS use case in ESCAPE data-lake.
  - Analysis data on a lake endpoint + cache layer + computing facility for data processing
    - we start little with a single storage/lake endpoint at CNAF
      - to then extend at least to CERN
- Embargoed data
  - Integrating **capability based** authN/Z
    - we used ESCAPE IAM to self manage the CMS Groups
  - we uses NANOAOD based analysis
    - as an interesting use case for future scenarios
- Open data
  - dataset imported in the data lake as test

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

Data access: embargoed data

- Lake endpoint: we setup an origin server with XRootD
  - where we imported a NANOAOD dataset with Rucio (see later)
  - exposing XRootD and HTTP on different ports

- Restricting access to CMS people and providing access through xrootd and https
  - only member of /escape/cms group (VOMS) can access it through a xrootd
  - only IAM access token with storage.read:/cms scope can access it through https

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

Data access: XCache layer

- An XCache server serving data to the analysis facility
  - both xrootd and https flow
  - keeping ACLs in sync with the lake endpoint for both used protocols

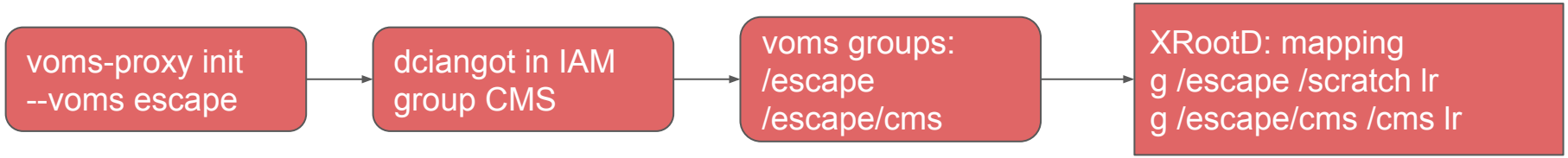
- The XCache server points to the origin above
  - XRootD access:
    - only member of /escape/cms group (VOMS) can access data
    - XCache server has to use its own X509 proxy to contact origin
      - so in a multi group scenario the cache should have a super-user proxy
      - on next release this could be avoided. On hold, waiting for a version to test
  - HTTP access:
    - only IAM access token with storage.read:/cms scope can access it through https

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

Data access: AuthN/Z summary

- custodial sites exposing xrootd endpoint:
  - authN/Z managed by X509 VOMS groups (supported by ESCAPE IAM)
- custodial sites exposing HTTP-WebDAV
  - authN/Z managed directly by IAM groups and JWT

X509/IAM flow



Full JWT scope authz

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

AuthN/Z model: granularity

- The trick stays on IAM mapping between identity and either a VOMS group or token scopes
- The two concepts has been put in sync through a policy
  - group A has its allowed namespace /A
  - any user that is part of group A (managed by group admin):
    - will automatically get group /escape/A when requesting X509 proxy with escape VO
    - will be automatically allowed to request scope storage.read:/A for his access token

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

Data Access setup: The XRooTD way

ESCAPE wiki page

- XRootD 4.11.2
- VomsXrd plugin for group attributes extraction
  - found a bug when multiple groups are specified
  - discussed with devs and tested the patch that worked
    - rpm on its way
- Instruction for setup collect into Dockerfiles:
  - Origin
  - Cache
- Local dev env also available with a docker-compose

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

Data Access setup: the HTTP way

ESCAPE wiki page

- XRootD 4.11.2
- xrootd-scitokens plugin
  - found a bug in parsing scope permission and reported to Brian B.
  - a fix has been proposed and tested → will be included in the next release
- Instruction for setup collect into Dockerfiles:
  - Origin
  - Cache
- Local dev env also available with a docker-compose
- ROOT can read data from HTTP using Davix
  - afaict there is no way to pass a bearer token right away
  - I created a quick patch allowing to pass a bearer token through env variable
  - tested and functioning. Still wait for discussing this with devs.

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

Data import: RUCIO configuration 1/2

- We managed, together with Aris (thanks!), to test a solution for embargoed data in Rucio using account scopes
- Different scopes mapped to different namespace in the data-lake
  - Embargoed data imported with cms_temp scope, ACLs enforced at FS lvl
    - only one RSE at CNAF supports the corresponding namespace so far
  - Open data imported with cms scope
    - every member of ESCAPE can access them
    - replication across lakes to be tested soon

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

Data import: RUCIO configuration 2/2

- Current limitations:
  - CMS logical file names are in the form of /store/mc/…. but Rucio does not accept '/' in the name of DIDs
    - some workaround tested replacing / with `__` . But not very clean
    - in CMS we solved this, but the configuration has to be applied as server level
  - also, we would like to skip the folder prefix hash done by Rucio by default
    - enforced at SE definition
    - e.g. having an entire dataset in the same FS path, not spread on different folders
    - this makes easier the import of data from third SE outside ESCAPE datalake

Aris suggested either to change RSE to the identity algorithm or to have 2 different RSEs per site, but before trying to hack a solution for it, we'd like to have a broader discussion about it

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

Tests summary

- Checked the correct ACLs management
- Tried RUCIO download of a registered embargoed data
- Submitted CMS condor jobs reading through the cache
  - XRootD protocol
  - via WebDAV with DAVIX
- Already visible some latency hiding effect in this simple test setup

Waiting for:

- XRootDs token and scope based authz support
- Rucio full token support
- Rucio multi-VO/group (multiple service proxies/token forwarding)

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

Documentation

XRootD protocol:

- ESCAPE wiki:
  - Bare metal installation
  - Dockerfiles
  - Docker compose example

HTTPs/WebDAV protocol:

- ESCAPE wiki:
  - Bare metal installation
  - Dockerfiles
  - Docker compose example

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

Backup

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

Compute provisioning

- Analysis Facility: we used DODAS to generate a HTCondor pool managed by Kubernetes
  - configuration fully managed through Helm charts
    - no CMS specific
    - cvmfs and squid completely configurable at chart level
    - sft.cern.ch cvmfs on WNs was the only requirement in this case

- Remote access to the analysis facility still granted through GSI
  - automatic configuration provided by INDIGO IAM
  - **Working on the migration toward a full token based model**
    - SciToken auth_method already tested on latest HTCondor release

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