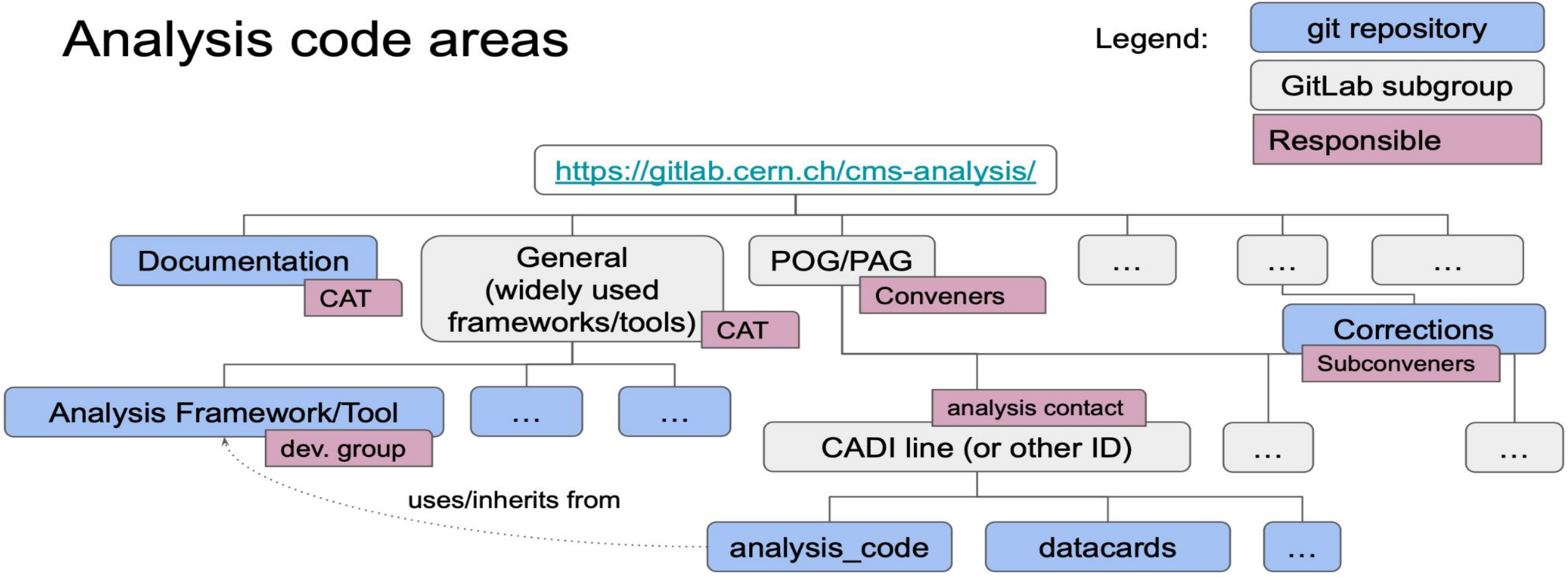
CI/CD on analysis facility

M. Bartolini, P. Lenzi

Workshop on quasi interactive analysis of big data with
high throughput

- Creation of the Common Analysis Tool group in CMS
- Common area where everyone in CMS puts their analysis
- Lives on GitLab
- Keeps the general framework separated from the analysis code
- Frameworks are containerized using Docker



CI-triggered analysis execution

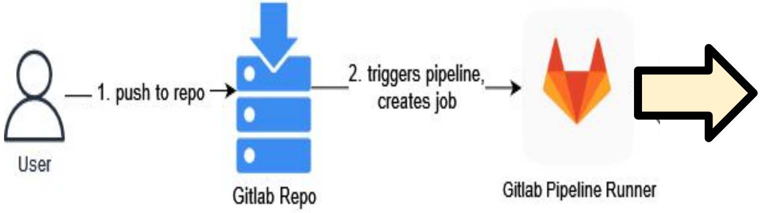
- The idea is to leverage the CI/CD GitLab functionalities to automatically run an analysis workflow whenever a new commit is added
  - Useful to establish bulletproof analysis reproducibility

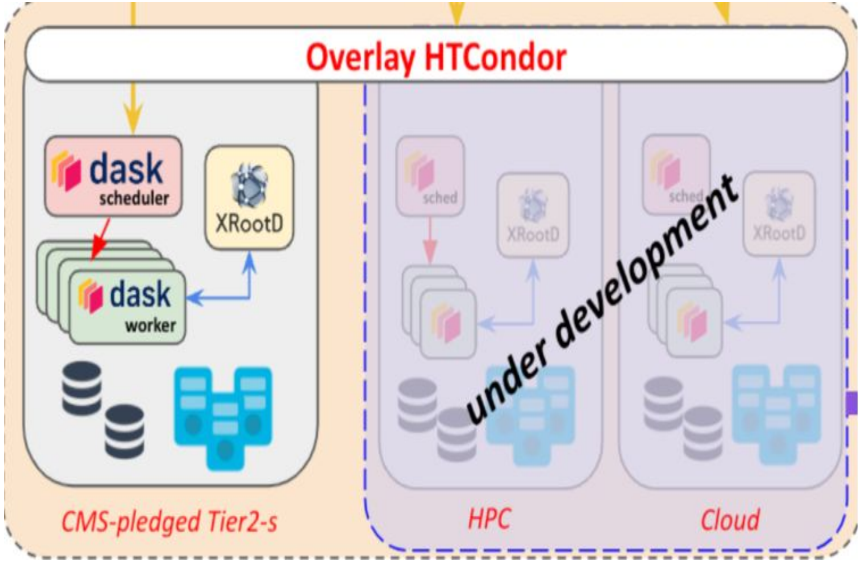
- Complementary use case wrt to the quasi interactive approach presented by T. Tedeschi. The ideas is to run containerized analysis pipeline on remote computing clouds

- Every CMS analysis is supposed to have a repository in cms-analysis:
  - The gitlab-ci.yaml defines the job that is triggered when a new modification is uploaded
  - Proof of concept:
    - Use containerized mkShapesRDF code to execute the analysis

- Since the gitlab runner cannot be used for CPU-heavy tasks, it is possible to offload the work to external HTCondor clusters
  - First test on INFN AF (Analysis Facility) which run on italian T2 resources
  - Easy authentication method via access token

CI-triggered analysis execution





- Offload the work by submitting the jobs in batches to the remote CMS HTCondor cluster
- Can easily extend this to other clusters in the future

Software stack

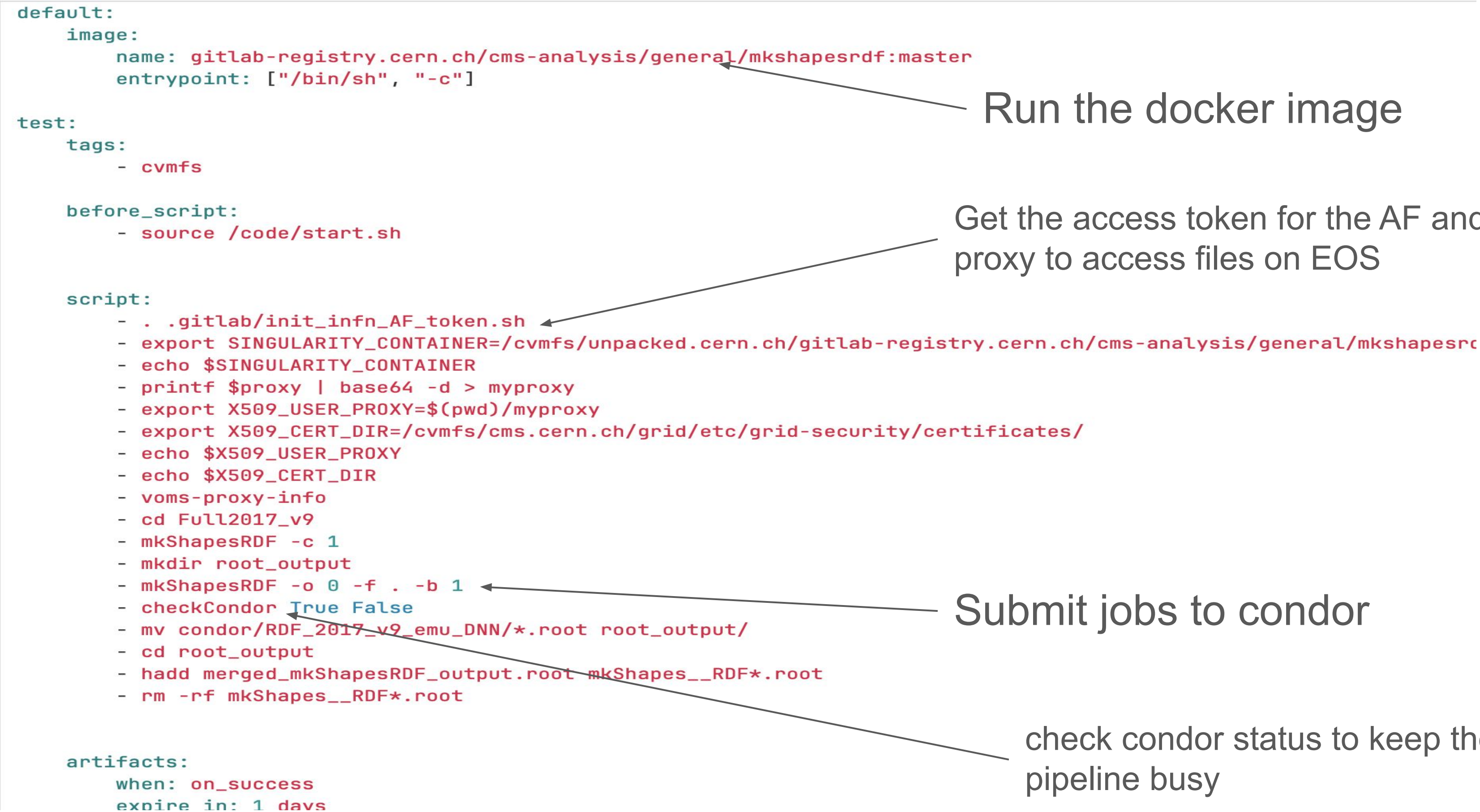
- The software to be used to run the analysis is packaged in a Docker image
  - For the proof-of-concept use case we studied, the Docker image of mkShapesRDF is created and deployed in the mkShapesRDF gitlab-registry and can be downloaded from anywhere (even your laptop if you have docker installed, see instructions):
    - `docker run -ti --user nobody gitlab-registry.cern.ch/cms-analysis/general/mkshapesrdf:master`

- An unpacked version of the image is also deployed to /cvmfs through the sync repository

  ```
  /cvmfs/unpacked.cern.ch/gitlab-registry.cern.ch/cms-analysis/general/mkshapesrdf\:*
  ```

- The image is created (and overwritten) every time a modification is pushed to framework

gitlab-ci.yml



```
default:
    image:
        name: gitlab-registry.cern.ch/cms-analysis/general/mkshapesrdf:master
        entrypoint: ["/bin/sh", "-c"]

test:
    tags:
        - cvmfs

    before_script:
        - source /code/start.sh


    script:
        - . .gitlab/init_infn_AF_token.sh
        - export SINGULARITY_CONTAINER=/cvmfs/unpacked.cern.ch/gitlab-registry.cern.ch/cms-analysis/general/mkshapesrd
        - echo $SINGULARITY_CONTAINER
        - printf $proxy | base64 -d > myproxy
        - export X509_USER_PROXY=$(pwd)/myproxy
        - export X509_CERT_DIR=/cvmfs/cms.cern.ch/grid/etc/grid-security/certificates/
        - echo $X509_USER_PROXY
        - echo $X509_CERT_DIR
        - voms-proxy-info
        - cd Full2017_v9
        - mkShapesRDF -c 1
        - mkdir root_output
        - mkShapesRDF -o 0 -f . -b 1
        - checkCondor True False
        - mv condor/RDF_2017_v9_emu_DNN/*.root root_output/
        - cd root_output
        - hadd merged_mkShapesRDF_output.root mkShapes__RDF*.root
        - rm -rf mkShapes__RDF*.root


    artifacts:
        when: on_success
        expire_in: 1 days
```

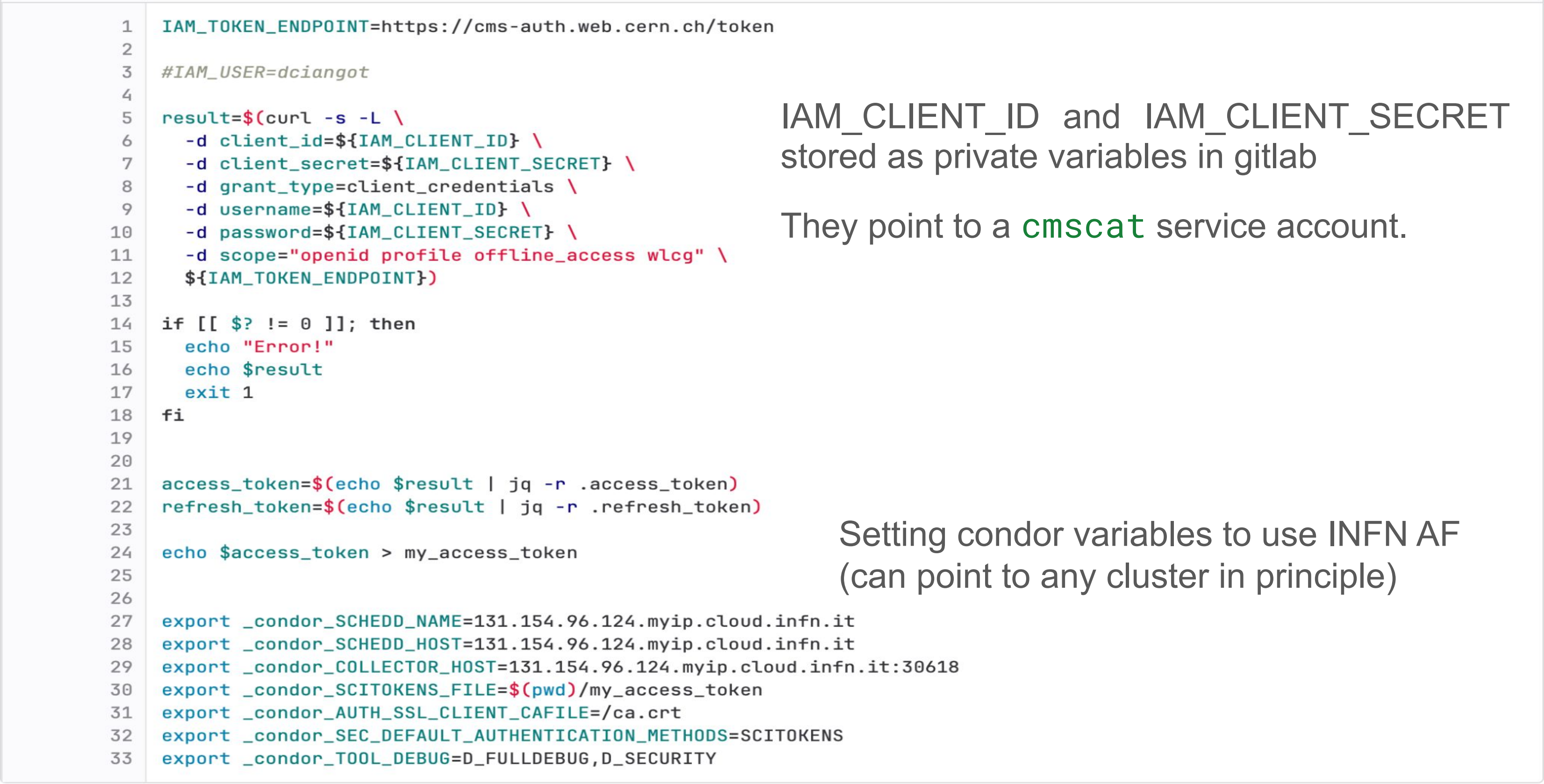
Run the docker image

Get the access token for the AF and the proxy to access files on EOS

Submit jobs to condor

check condor status to keep the pipeline busy

The IAM token



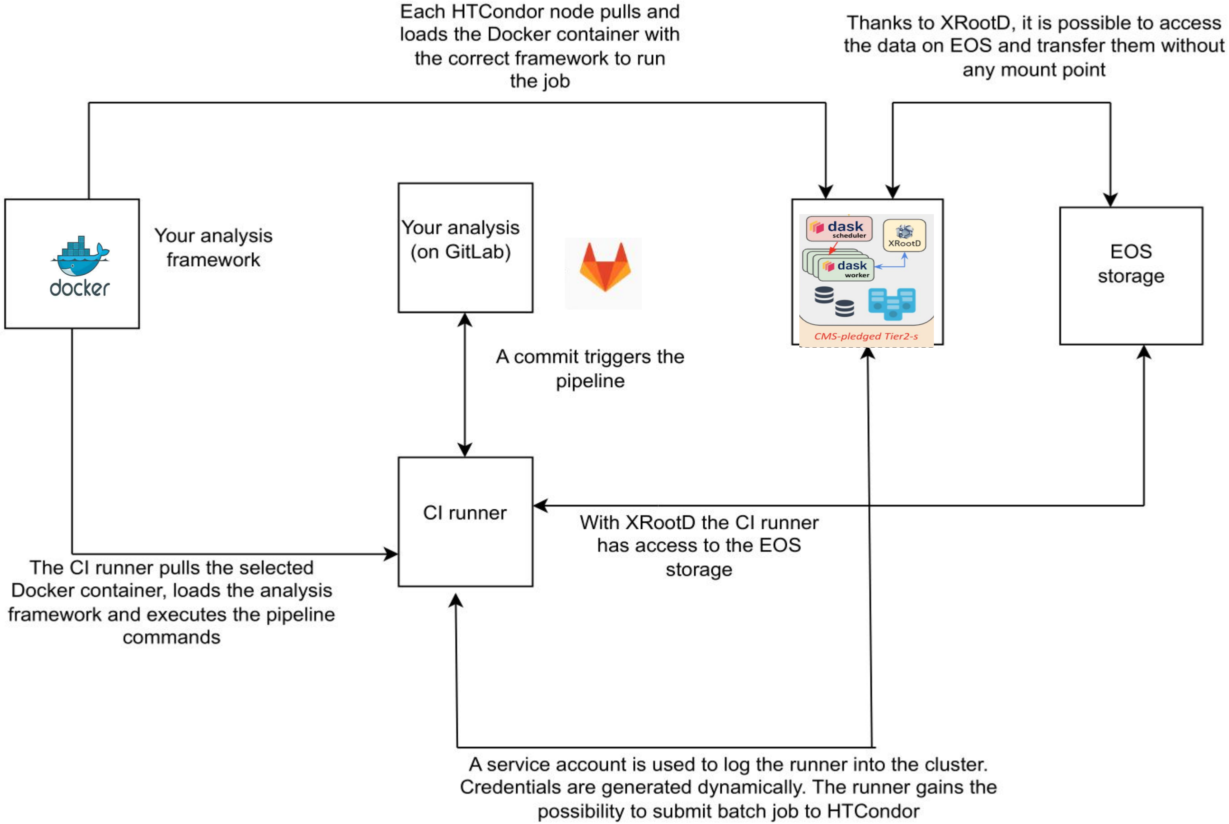
```
IAM_TOKEN_ENDPOINT=https://cms-auth.web.cern.ch/token

#IAM_USER=dciangot

result=$(curl -s -L \
  -d client_id=${IAM_CLIENT_ID} \
  -d client_secret=${IAM_CLIENT_SECRET} \
  -d grant_type=client_credentials \
  -d username=${IAM_CLIENT_ID} \
  -d password=${IAM_CLIENT_SECRET} \
  -d scope="openid profile offline_access wlcg" \
  ${IAM_TOKEN_ENDPOINT})

if [[ $? != 0 ]]; then
  echo "Error!"
  echo $result
  exit 1
fi


access_token=$(echo $result | jq -r .access_token)
refresh_token=$(echo $result | jq -r .refresh_token)

echo $access_token > my_access_token


export _condor_SCHEDD_NAME=131.154.96.124.myip.cloud.infn.it
export _condor_SCHEDD_HOST=131.154.96.124.myip.cloud.infn.it
export _condor_COLLECTOR_HOST=131.154.96.124.myip.cloud.infn.it:30618
export _condor_SCITOKENS_FILE=$(pwd)/my_access_token
export _condor_AUTH_SSL_CLIENT_CAFILE=/ca.crt
export _condor_SEC_DEFAULT_AUTHENTICATION_METHODS=SCITOKENS
export _condor_TOOL_DEBUG=D_FULLDEBUG,D_SECURITY
```

IAM_CLIENT_ID and IAM_CLIENT_SECRET stored as private variables in gitlab

They point to a `cmscat` service account.

Setting condor variables to use INFN AF (can point to any cluster in principle)

The workflow



Each HTCondor node pulls and loads the Docker container with the correct framework to run the job

Thanks to XRootD, it is possible to access the data on EOS and transfer them without any mount point

Your analysis framework

Your analysis (on GitLab)

EOS storage

A commit triggers the pipeline

CI runner

With XRootD the CI runner has access to the EOS storage

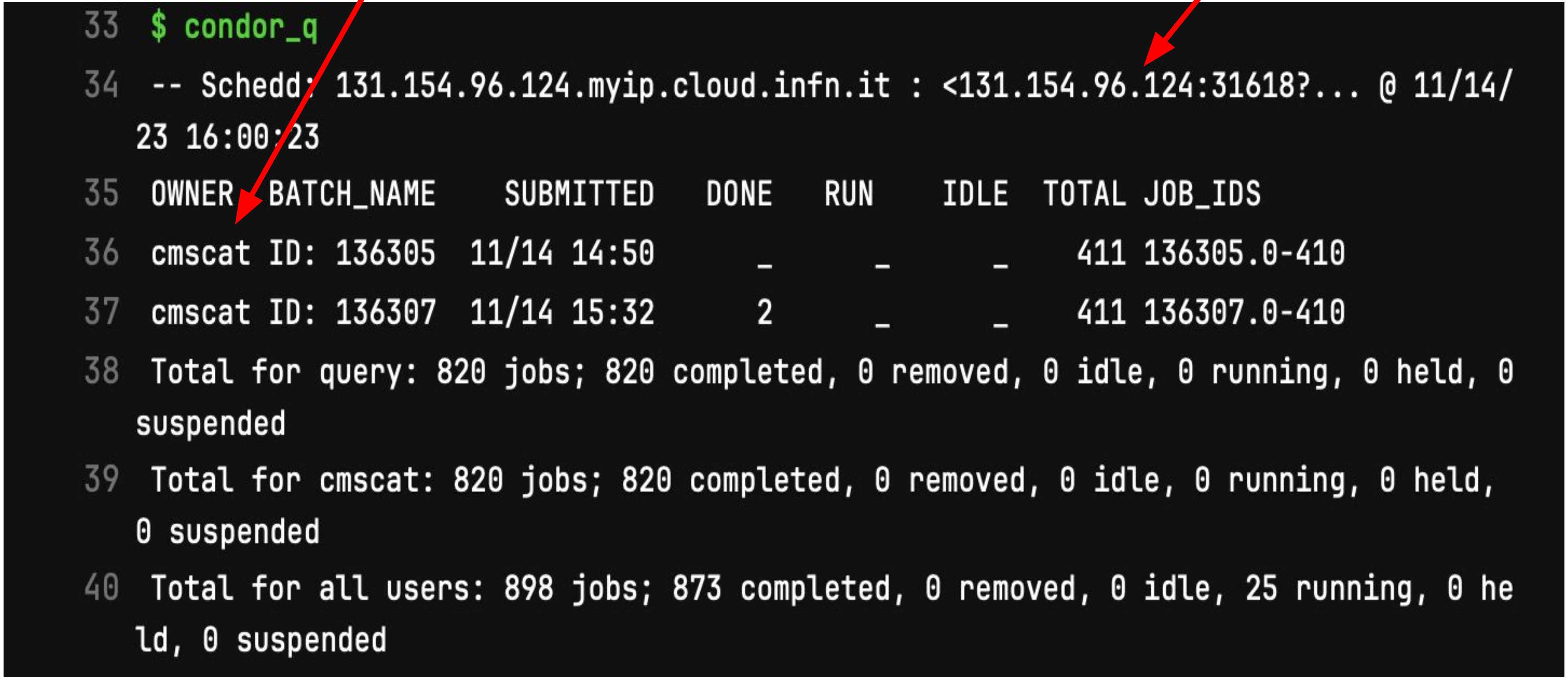
The CI runner pulls the selected Docker container, loads the analysis framework and executes the pipeline commands

A service account is used to log the runner into the cluster. Credentials are generated dynamically. The runner gains the possibility to submit batch job to HTCondor

Submitting jobs to condor as cmscat user

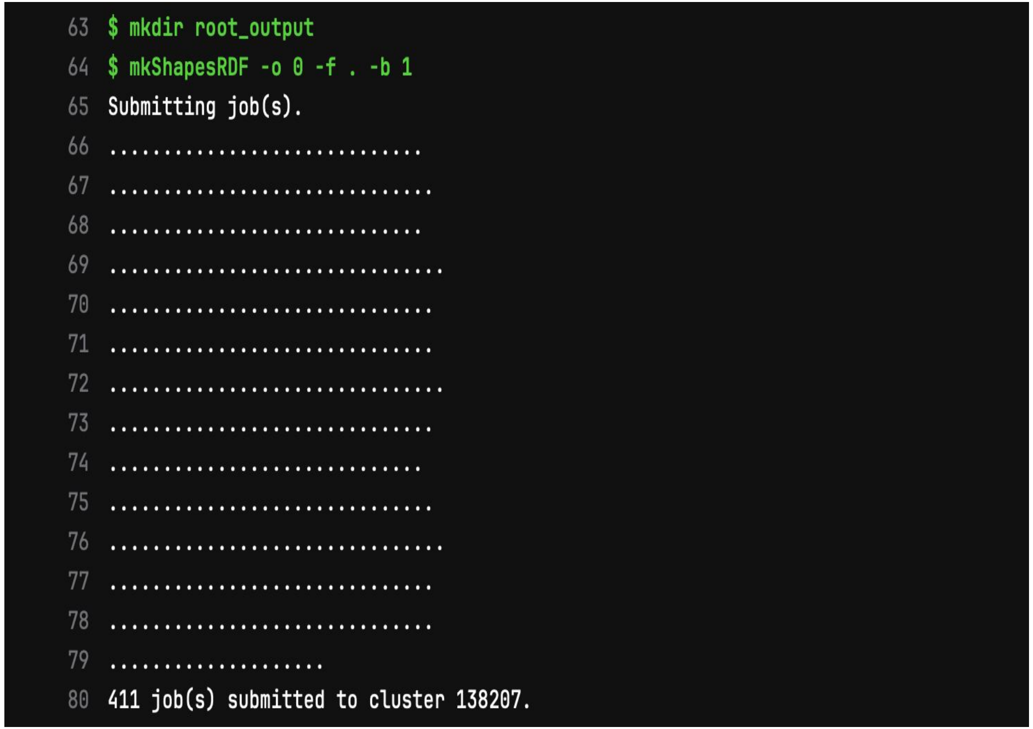
After setting the IAM token, condor will see cmscat as the job owner

The AF scheduler IP should appear



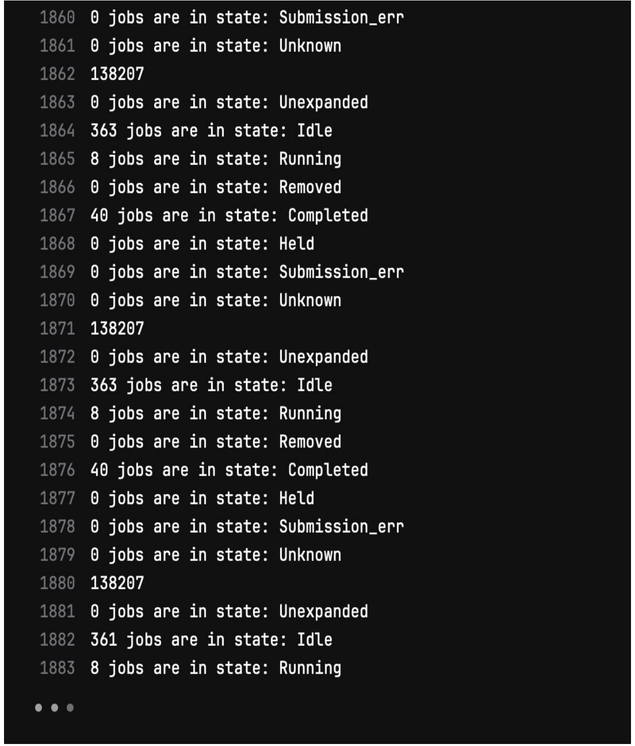
```
$ condor_q
-- Schedd: 131.154.96.124.myip.cloud.infn.it : <131.154.96.124:31618?... @ 11/14/
23 16:00:23
OWNER  BATCH_NAME    SUBMITTED   DONE   RUN    IDLE  TOTAL JOB_IDS
cmscat ID: 136305  11/14 14:50      _      _      _    411 136305.0-410
cmscat ID: 136307  11/14 15:32      2      _      _    411 136307.0-410
Total for query: 820 jobs; 820 completed, 0 removed, 0 idle, 0 running, 0 held, 0
suspended
Total for cmscat: 820 jobs; 820 completed, 0 removed, 0 idle, 0 running, 0 held,
0 suspended
Total for all users: 898 jobs; 873 completed, 0 removed, 0 idle, 25 running, 0 he
ld, 0 suspended
```

Submitting jobs to AF via gitlab CI



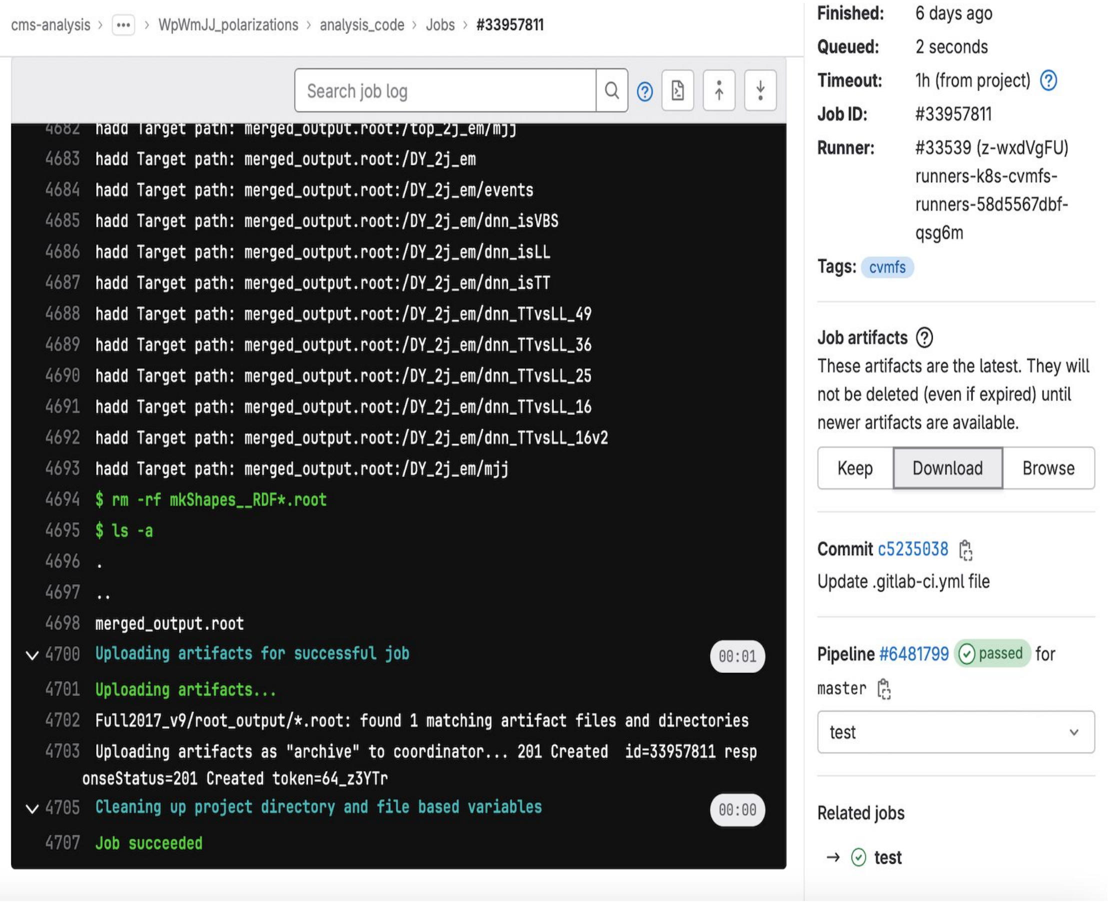
```
$ mkdir root_output
$ mkShapesRDF -o 0 -f . -b 1
Submitting job(s).
.............................
..............................
.............................
...............................
..............................
..............................
...............................
..............................
.............................
..............................
...............................
..............................
..............................
....................
411 job(s) submitted to cluster 138207.
```

New utils to monitor the status of jobs every 10 seconds



```
0 jobs are in state: Submission_err
0 jobs are in state: Unknown
138207
0 jobs are in state: Unexpanded
363 jobs are in state: Idle
8 jobs are in state: Running
0 jobs are in state: Removed
40 jobs are in state: Completed
0 jobs are in state: Held
0 jobs are in state: Submission_err
0 jobs are in state: Unknown
138207
0 jobs are in state: Unexpanded
363 jobs are in state: Idle
8 jobs are in state: Running
0 jobs are in state: Removed
40 jobs are in state: Completed
0 jobs are in state: Held
0 jobs are in state: Submission_err
0 jobs are in state: Unknown
138207
0 jobs are in state: Unexpanded
361 jobs are in state: Idle
8 jobs are in state: Running

• • •
```

Final result



cms-analysis › ••• › WpWmJJ_polarizations › analysis_code › Jobs › #33957811

```
hadd Target path: merged_output.root:/top_2j_em/mjj
hadd Target path: merged_output.root:/DY_2j_em
hadd Target path: merged_output.root:/DY_2j_em/events
hadd Target path: merged_output.root:/DY_2j_em/dnn_isVBS
hadd Target path: merged_output.root:/DY_2j_em/dnn_isLL
hadd Target path: merged_output.root:/DY_2j_em/dnn_isTT
hadd Target path: merged_output.root:/DY_2j_em/dnn_TTvsLL_49
hadd Target path: merged_output.root:/DY_2j_em/dnn_TTvsLL_36
hadd Target path: merged_output.root:/DY_2j_em/dnn_TTvsLL_25
hadd Target path: merged_output.root:/DY_2j_em/dnn_TTvsLL_16
hadd Target path: merged_output.root:/DY_2j_em/dnn_TTvsLL_16v2
hadd Target path: merged_output.root:/DY_2j_em/mjj
$ rm -rf mkShapes__RDF*.root
$ ls -a
.
..
merged_output.root
Uploading artifacts for successful job                                   00:01
Uploading artifacts...
Full2017_v9/root_output/*.root: found 1 matching artifact files and directories
Uploading artifacts as "archive" to coordinator... 201 Created  id=33957811 resp
onseStatus=201 Created token=64_z3YTr
Cleaning up project directory and file based variables                   00:00
Job succeeded
```

Finished: 6 days ago
Queued: 2 seconds
Timeout: 1h (from project)
Job ID: #33957811
Runner: #33539 (z-wxdVgFU) runners-k8s-cvmfs-runners-58d5567dbf-qsg6m
Tags: cvmfs

Job artifacts

These artifacts are the latest. They will not be deleted (even if expired) until newer artifacts are available.

Commit c5235038

Update .gitlab-ci.yml file

Pipeline #6481799 passed for master

test

Related jobs

→ test

- Once the jobs have been submitted a scripts is used to check the status of the condor jobs every n seconds to keep the CI busy
- Once all the jobs are done running the script will exit the loop and all data are transferred back to the CI runner, merged together and added to artifacts
- The complete pipeline

Integration with reana

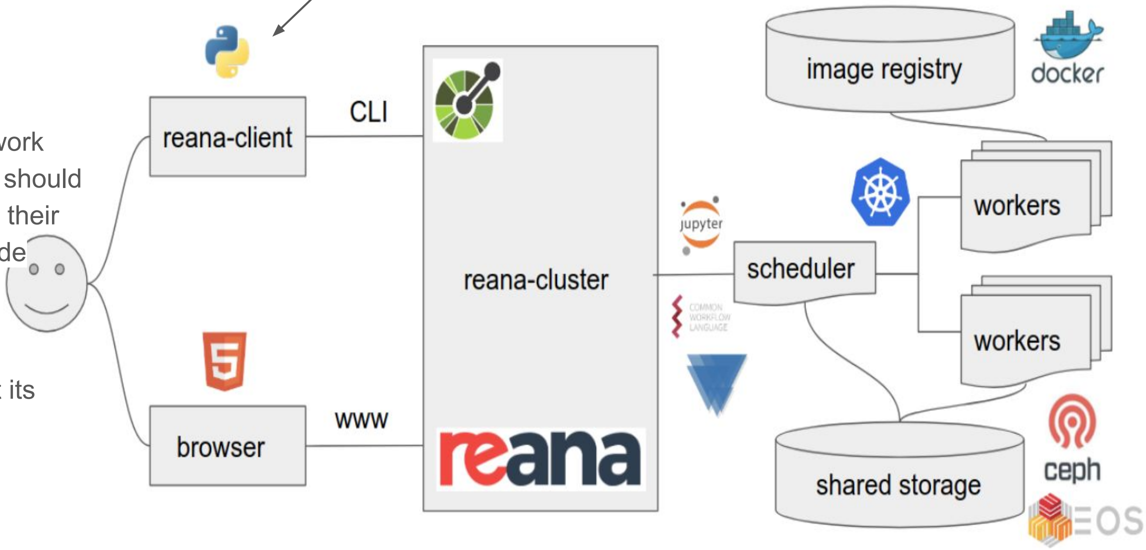
- Our work with the CI/CD was mostly exploratory. Authentication procedure to the computing resources is easy, but it's too specific and managing and controlling jobs submission would require a significant effort to develop custom code

- We need to move towards decoupling the framework from underlying computing resources -> the user should ideally be able to choose which cluster to submit their jobs to and retrieve files without modifying the code

- For this reason we plan to use Reana and exploit its integration with gitlab

- Reana is a reproducible analysis platform allowing scientists to run containerized analysis pipeline on remote computing clouds

The idea is to start the reana-client inside the gitlab CI



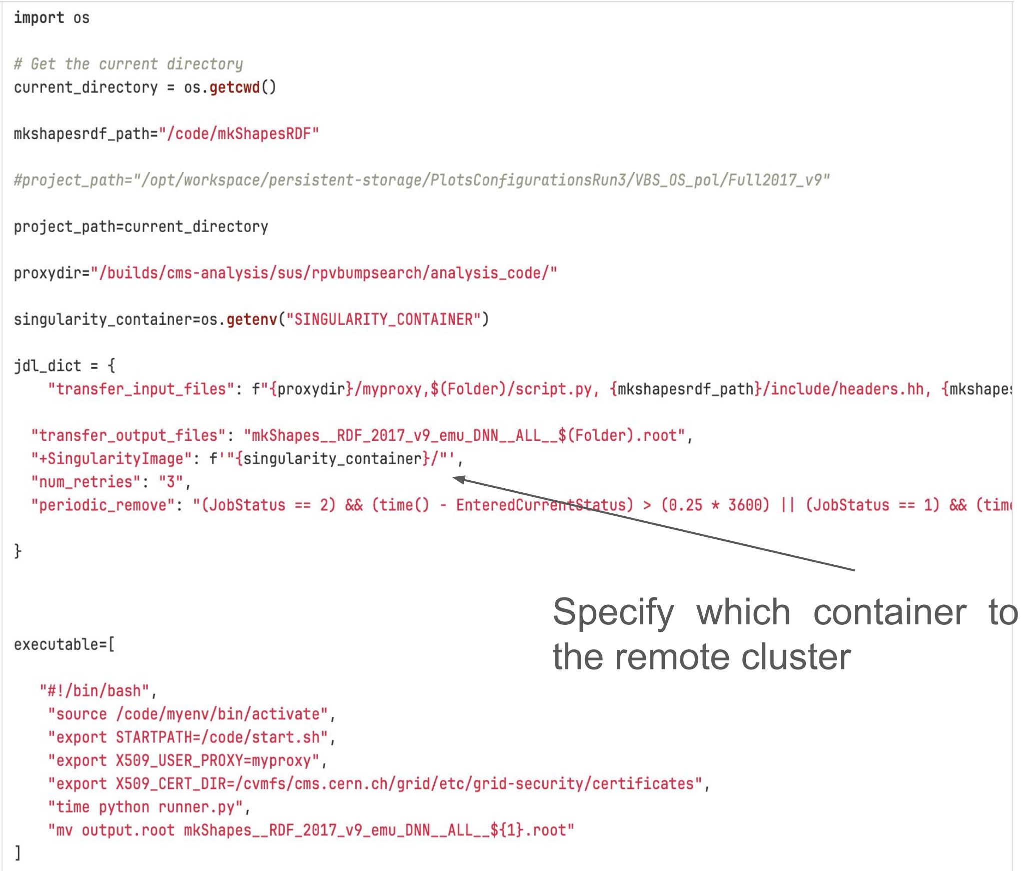
Deploy a reana cluster in the CMS-pledged Tier 2 and use its function to handle submission of jobs

Conclusions

- The proof of concept to run containerized analysis pipeline on remote computing clouds may have potential
- It’s a complementary use case
- We had a discussion with the Perugia’s group about the possibility to deploy a reana cluster in the CMS pledged Tier-2 system
- From our side we plan to start investigating the feasibility of integrating Reana as part of the workflow and check if it can be scaled fairly easily without any significant code overhaul

BACKUP

The condor config file



```
import os

# Get the current directory
current_directory = os.getcwd()

mkshapesrdf_path="/code/mkShapesRDF"

#project_path="/opt/workspace/persistent-storage/PlotsConfigurationsRun3/VBS_OS_pol/Full2017_v9"

project_path=current_directory

proxydir="/builds/cms-analysis/sus/rpvbumpsearch/analysis_code/"

singularity_container=os.getenv("SINGULARITY_CONTAINER")

jdl_dict = {
    "transfer_input_files": f"{proxydir}/myproxy,$(Folder)/script.py, {mkshapesrdf_path}/include/headers.hh, {mkshape

  "transfer_output_files": "mkShapes__RDF_2017_v9_emu_DNN__ALL__$(Folder).root",
  "+SingularityImage": f'"{singularity_container}/"',
  "num_retries": "3",
  "periodic_remove": "(JobStatus == 2) && (time() - EnteredCurrentStatus) > (0.25 * 3600) || (JobStatus == 1) && (tim

}



executable=[

   "#!/bin/bash",
    "source /code/myenv/bin/activate",
    "export STARTPATH=/code/start.sh",
    "export X509_USER_PROXY=myproxy",
    "export X509_CERT_DIR=/cvmfs/cms.cern.ch/grid/etc/grid-security/certificates",
    "time python runner.py",
    "mv output.root mkShapes__RDF_2017_v9_emu_DNN__ALL__${1}.root"
]
```

Specify which container to run on the remote cluster

- Condor specific configuration parameters are specified in a jdl_config.py file now

- This used to be hardcoded before the commit