Analysis Grand Challenge at REANA

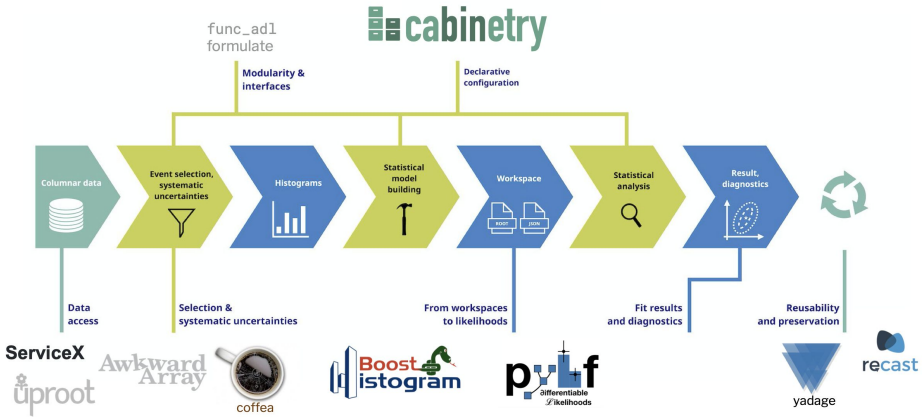
Andrii Povsten
Mentors: Alex Held, Matthew Feickert (University of Wisconsin- Madison), Oksana Shadura (University Nebraska-Lincoln), Tibor Simko (CERN)

FAIR and Open Science in High Energy Physics, OAC-2226378, OAC-2226379 and OAC-2226380

Analysis Grand Challenge

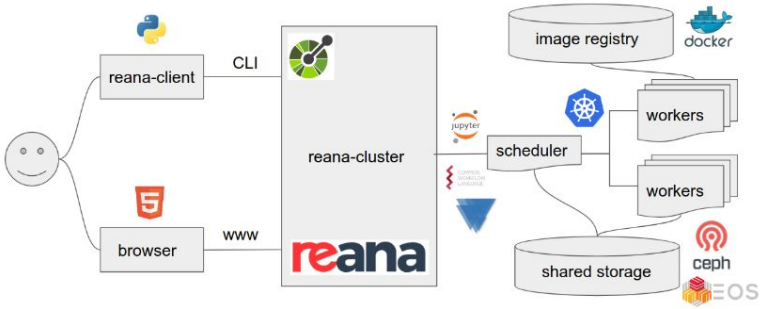
- Columnar data extraction from large dataset
- Processing of that data (event filtering, construction of observables, evaluation of systematic uncertainties) into histograms
- Statistical model construction and statistical inference
- Relevant visualisation for this steps

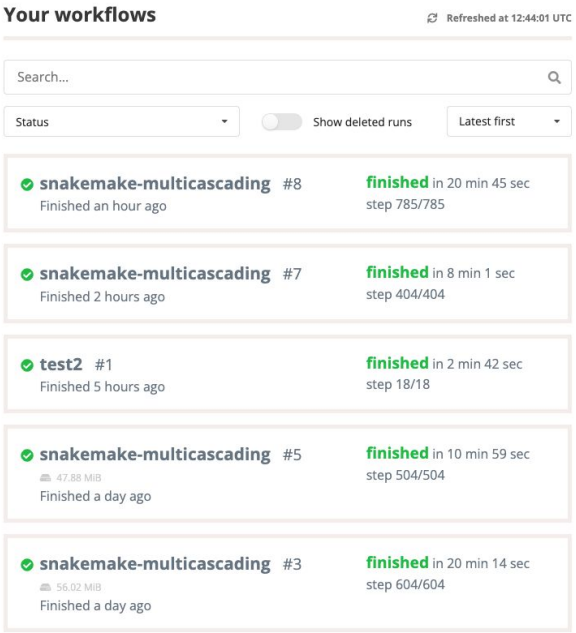
\+ **Adding analysis preservation step to AGC pipeline**



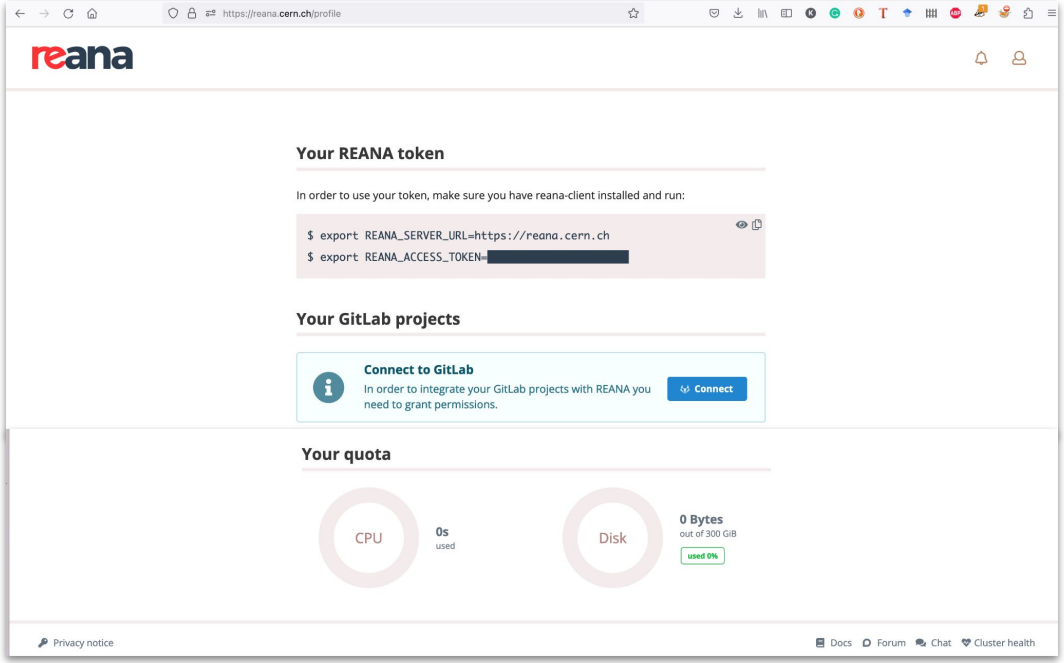
REANA

- **REANA is a reproducible analysis platform** allowing researchers to run containerised data analyses and data simulation pipelines on remote compute clouds.
- **Provides platform for reusable analysis**: *containerised once, allows user to reuse anytime (as well elsewhere)*
    - Supports several **different container technologies** (Docker, Singularity), **compute clouds** (Kubernetes/OpenShift), shared **storage systems** (Ceph, EOS) and **workflow specifications** (CWL, Yadage, Snakemake)





REANA instance at CERN - https://reana.cern.ch



- Using custom user tokens
- CERN gitlab native integration - allows to integrate REANA into your GitLab pipelines.
- Excellent documentation https://docs.reana.io/
- User support: https://forum.reana.io/

For setting REANA client:
pip install reana-client

```
(reana) andrewpovsten@pucomphep03 cms-open-data-ttbar % reana-client ping
REANA server: https://reana.cern.ch
REANA server version: 0.9.2
REANA client version: 0.9.1
Authenticated as: Andrii Povsten <andrii.povsten@cern.ch>
Status: Connected
(reana) andrewpovsten@pucomphep03 cms-open-data-ttbar %
```

Next step: REANA specification file

The REANA reproducible analysis platform requires to have `reana.yaml` file present in your analysis source code (REANA specification file).

Its purpose is to answer the **Four Questions**:

1. What is your input data? (e.g. dataset samples)

2. What is your analysis code? (e.g. python notebook, compiled executable, script)

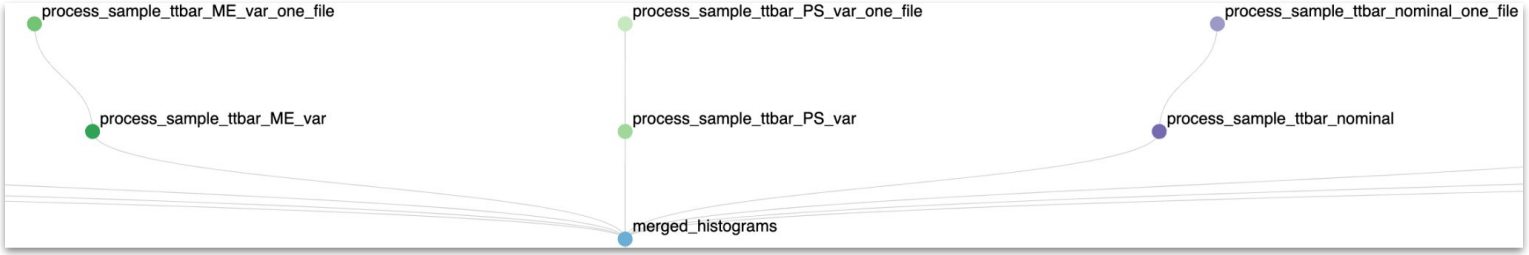
3. What is your computing environment? (e.g. docker image)

4. Which computational steps do you take to arrive at results? (e.g. data processing or statistical model construction and statistical inference)

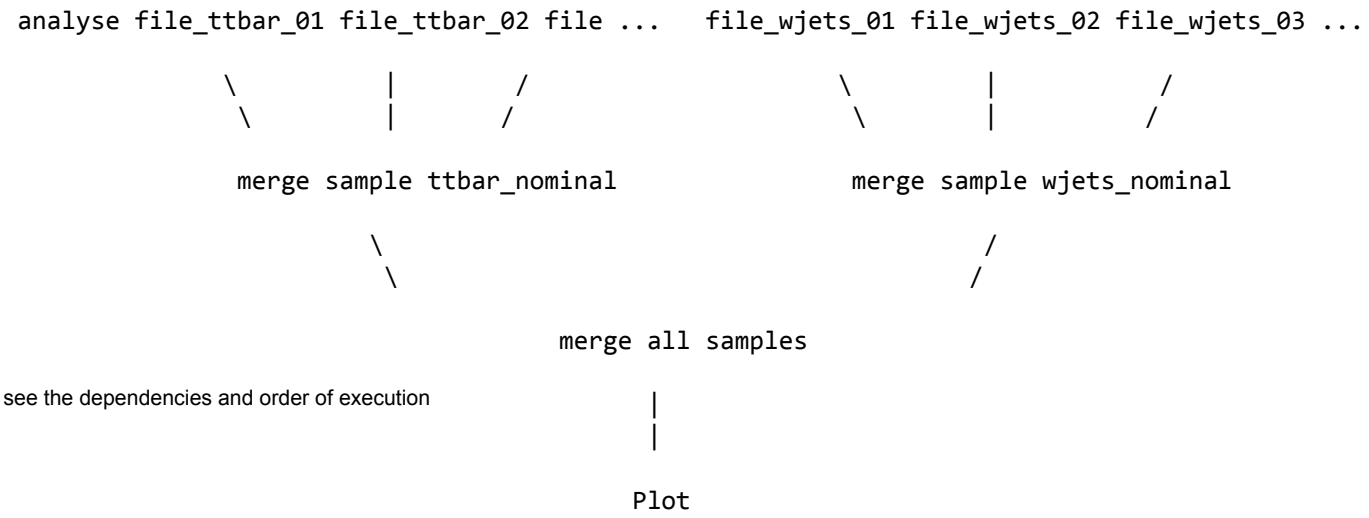
Next step: Workflow management engine choice

- Our choice was to use **Snakemake workflow management system (integrated in REANA)**
  - Help to keep a record of used scripts and their input dependencies
  - Run multiple steps in sequence, parallelising where possible
  - Automatically detect if something changes and then reprocess data if need
- **Snakemake key feature is a “rule” description, which enables the parallelisation within REANA**, running each rule in a separate virtual node.
- **Snakemake allows you to create a set of rules, each one defining a “step” of your analysis.**
- The rules need to be written in a file called Snakefile:
  - *The input:* Data files, scripts, executables or any other files.
  - *The expected output:* It’s not required to list all possible outputs. Just those that you want to monitor or that are used by a subsequent step as inputs.
  - *Shell:* A command to run to process the input and create the output.

Analysis Grand Challenge pipeline: Adapting to Snakemake



Each rule REANA sends to the Kubernetes cluster as separate node

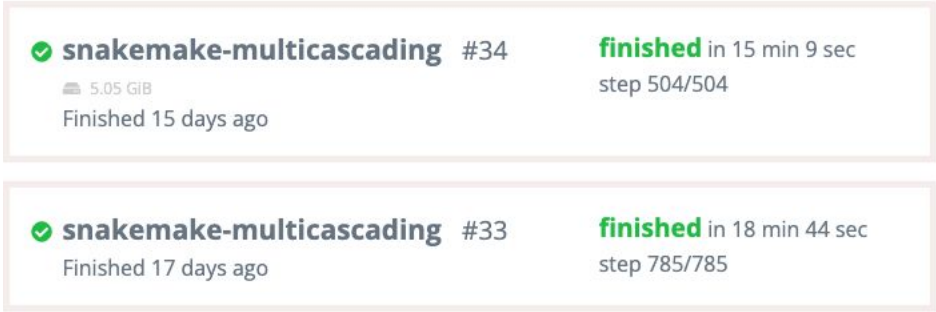


Snakemake checks the inputs and outputs in the rules to see the dependencies and order of execution

Comparison table for single_top_s_chan of coffea vs REANA

Processing single top s channel	*REANA processing*	*Coffea processing*
File 1 1.1 Gb	3.13	2.96
File 2 168 Mb	3.06	2.73
File 3 1.6 Gb	3.16	2.66
File 4 1.1 Gb	3.11	2.58
File 5 900 Mb	3.37	2.66
File 6 140 Mb	2.85	2.58

The current time execution for a huge dataset



Conclusion

- Successfully implement the AGC at REANA
- Get parameterized AGC notebook and execute it with papermill tool

Future Tasks

- Add the recasting step for a new sample which can be merged.
- Further optimisation of AGC processing time in REANA
- Updating to use eos/public instead of UNL url.
- Testing AGC ServiceX and machine learning pipelines in REANA