



Mentoring session exercises

Analysis Reproducibility workshop Mar 28 2025



Podman / Docker

- **Python3.13 (easy):** You want to try out one of the [new features](#) of python 3.13, but don't have it installed locally. Find a container that contains python3.13 and start python.
 - For example, try out the [improved interactive interpreter](#)
- **Build a docker file (easy):** Write your own image that builds on python3.13 and contains the `uproot` library
 - **Optional:** Most python applications have a `requirements.txt` that states all requirements (such as `uproot`). Create a dummy `requirements.txt`, copy it in your container and install all required packages with `pip3 install -r requirements.txt`.
- **Hard: Use the power of docker (podman) and gitlab-ci (github-ci) to keep your analysis environment up-to-date!**
 - Use [CMS OpenData HTauTau Analysis](#) as an an example analysis. Fork it to your own git repository.
 - Write a Dockerfile for it that setups all the dependencies and has the analysis code built inside it into an executable for skimming, that can be used inside the container
 - Add the `build_image` job to the build stage of the `.gitlab-ci.yml`.
 - Push the `.gitlab-ci.yml` upstream

Apptainer

- **Easy:** Repeat one of the docker exercises on your cluster with Apptainer using an interactive session
- **Medium:** Do the same with a definition file.
 - Option: Use the `%runscript` directive to print out the `uproot` version when the container runs
- **Hard:** Perform the [CMS example analysis](#) in a single execution using a definition file. Save the plots in the execution directory.
 - Hint: keep in mind where to store intermediate files.
- **Hard:** Set a Jupyter Notebook with the environment for running the [analysis example](#), and execute it on the cells.

GitHub actions / Gitlab CI

- **From zero to hero (medium):** Start a new repository, add a main.py with a trivial function, a `test_main.py` with a trivial test and add CI that calls `pytest` to run the test
 - Option 1: Test on multiple python versions
 - Option 2: Create a file in your python script and upload it as an artifact (optionally as a second step)
- **LaTeX documents (GitHub; hard):** Create a simple LaTeX document and compile it using [github-action-for-latex](#)
- **Build a docker container (GitHub; hard):** Pick a docker image and build it in the CI using [build-and-push-docker-images](#)

Reproducible analyses with REANA

- **Easy:** In episode 3 “Developing serial workflows”, we have practiced developing serial workflow by means of producing the same fit plot with a different title (“Fit example” → “RooFit example”). Redo the same exercise using Snakemake.

Hints:

- Basic Snakemake rules for the RooFit workflow can be found in episode 6 “Developing parallel workflows”.
- Snakemake workflow engine is using cache by default. It is sufficient to restart Snakemake workflows always from the start. The workflow engine is will run only those rules for which inputs or code has changed, and will reuse results of past workflow steps.

- **Hard.** If you were already comfortable using Snakemake before you took this training, you may try to write the parallel version of HiggsToTauTau analysis workflow using Snakemake rather than Yadage.

Notes:

- Observe how the Snakemake workflow specification is more concise than its Yadage counterpart.
- Since this exercise can be rather time consuming, you can consult the solution [here](#).