REANA basics and a tutorial

Maxim Potekhin
Nuclear and Particle Physics Software Group (BNL)

Brookhaven National Laboratory

Phenix School 2021

Objectives

- The main objective is to introduce the participants of the School to the basics of REANA operation, and give enough information for further experimentation
- Please see our website for more information: https://www.phenix.bnl.gov/analysis/reana.html
- We'll cover the very basics of Docker just to explain how it's used in REANA, and expand on the brief intro presented on 6/22/2021
- Questions? Contact me at "potekhin at bnl.gov"

Redux: Containers, images, repositories

- “Image” is a read-only template residing in storage
- In essence, a software package - preserved as a archive
- It is used to create a running process - the “container”
- “Registry” is a storage and access system for images (like a repo)
- Example: inspect images on a local machine - “docker image ls”
- Docker Hub is a cloud-based registry provided by Docker Inc

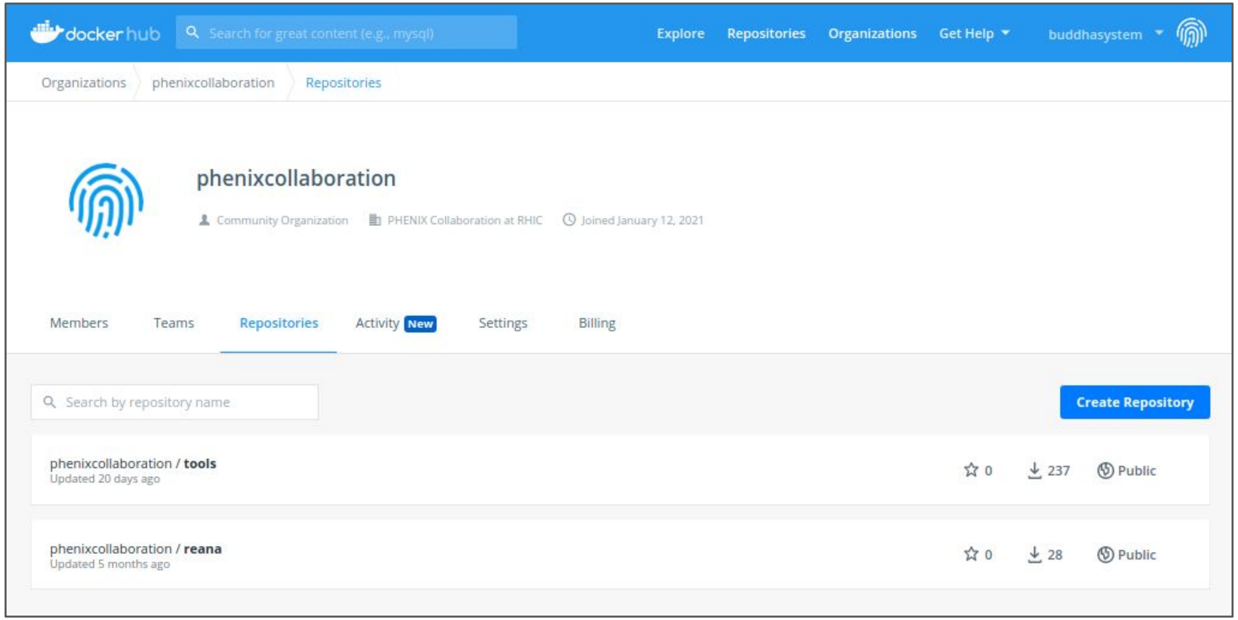
```
REPOSITORY         TAG         IMAGE ID       CREATED          SIZE

simple_server      latest      3d7b269117a7   20 months ago    158MB
django_import      latest      975596a4f73f   20 months ago    207MB
ubuntu             latest      d131e0fa2585   21 months ago    102MB
alpine             latest      cdf98d1859c1   21 months ago    5.53MB
mediawiki          latest      efd68a02fb8a   21 months ago    691MB
mariadb/server     latest      8fe757be2fd3   23 months ago    368MB
nginx              latest      568c4670fa80   2 years ago      109MB
```

Image lifecycle

- Building an image can involve copying files, creating folders, running build procedures, installing packages from third party resources etc.
  - Conceptually identical to building a software stack on an existing machine.
  - Images can be layered i.e. built incrementally, growing the software stack as needed. Example - the user can take a clean OS image and add an IDE of their choice, then perhaps a server.
- The Docker engine maintains a local cache of images on the user's machine
- It is also possible to simply save an image as a tarball if needed for long-term preservation
- The most popular way to store and access images is via "registries", such as Docker Hub
- BNL runs its own registry available internally on site

PHENIX Organization on Docker Hub

- We have presence on Docker Hub - although not large at this point
- https://hub.docker.com/orgs/phenixcollaboration/repositories



phenixcollaboration

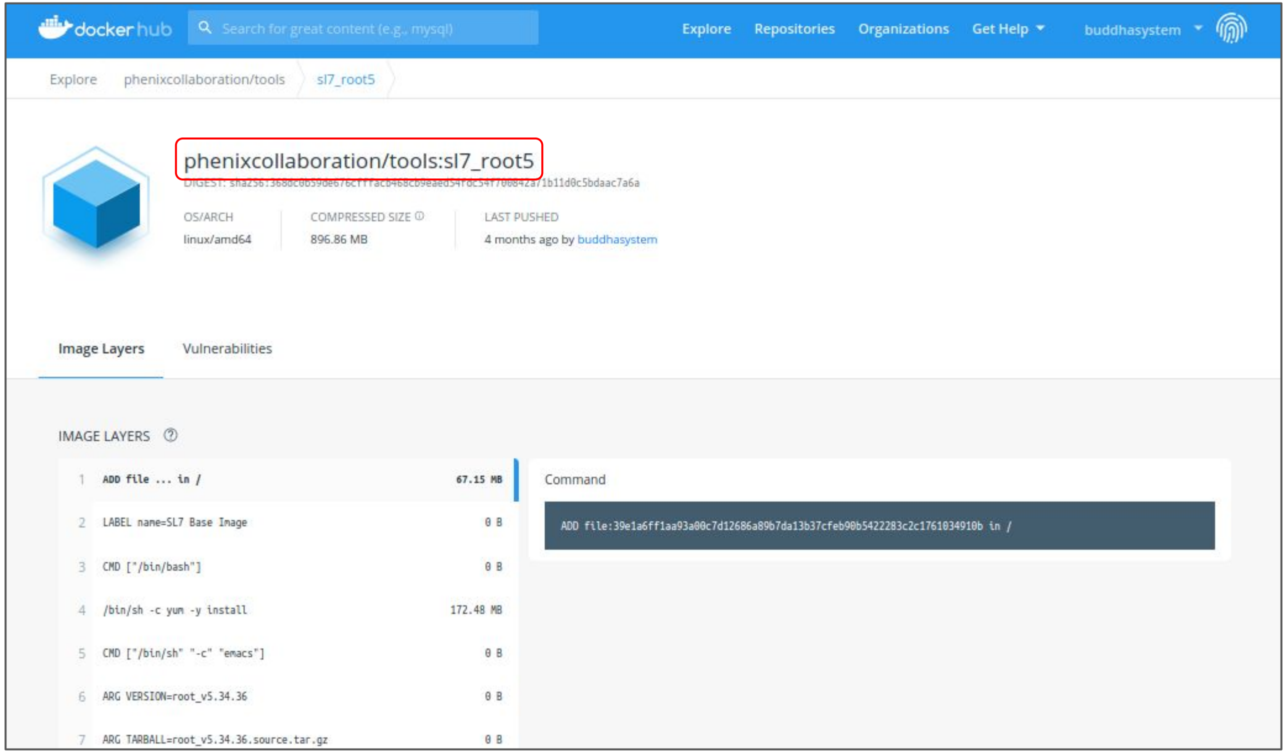
Community Organization   PHENIX Collaboration at RHIC   Joined January 12, 2021

phenixcollaboration / **tools**
Updated 20 days ago — ☆ 0 — 237 — Public

phenixcollaboration / **reana**
Updated 5 months ago — ☆ 0 — 28 — Public

Access to images: an example

- Within a repository the image is identified by a unique tag



phenixcollaboration/tools:sl7_root5

DIGEST: sha256:368dc0b59de676cfffacb468cb9eaed54fdc54f700842a71b11d0c5bdaac7a6a

OS/ARCH	COMPRESSED SIZE	LAST PUSHED
linux/amd64	896.86 MB	4 months ago by buddhasystem

IMAGE LAYERS

```
1  ADD file ... in /                              67.15 MB
2  LABEL name=SL7 Base Image                           0 B
3  CMD ["/bin/bash"]                                   0 B
4  /bin/sh -c yum -y install                     172.48 MB
5  CMD ["/bin/sh" "-c" "emacs"]                        0 B
6  ARG VERSION=root_v5.34.36                           0 B
7  ARG TARBALL=root_v5.34.36.source.tar.gz             0 B
```

Command

```
ADD file:39e1a6ff1aa93a00c7d12686a89b7da13b37cfeb90b5422283c2c1761034910b in /
```

Access to images: an example (cont’d)

If you need ROOT5 or 6, you can always run it on your laptop (you’ll need to install Docker).

ROOT

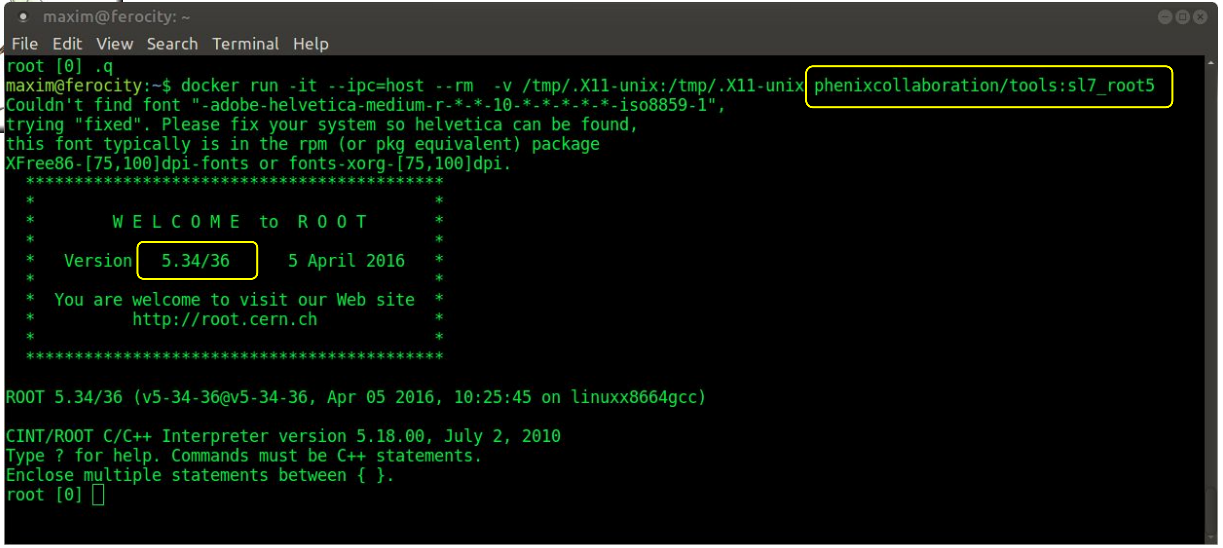
Version 5

Conception: Rene Brun, Fons Rademakers

Lead Developers: Rene Brun, Philippe Canal, Fons Rademakers

Core Engineering: Bertrand Bellenot, Olivier Couet, Gerardo Ganis, Andrei Gheata, Lorenzo Moneta, Axel Naumann, Paul Russo, Matevz Tadel

Version 5.34/36



```
root [0] .q
maxim@ferocity:~$ docker run -it --ipc=host --rm  -v /tmp/.X11-unix:/tmp/.X11-unix phenixcollaboration/tools:sl7_root5
Couldn't find font "-adobe-helvetica-medium-r-*-*-10-*-*-*-*-*-iso8859-1",
trying "fixed". Please fix your system so helvetica can be found,
this font typically is in the rpm (or pkg equivalent) package
XFree86-[75,100]dpi-fonts or fonts-xorg-[75,100]dpi.
  *******************************************
  *                                         *
  *        W E L C O M E  to  R O O T       *
  *                                         *
  *   Version   5.34/36      5 April 2016   *
  *                                         *
  *  You are welcome to visit our Web site  *
  *          http://root.cern.ch            *
  *                                         *
  *******************************************

ROOT 5.34/36 (v5-34-36@v5-34-36, Apr 05 2016, 10:25:45 on linuxx8664gcc)

CINT/ROOT C/C++ Interpreter version 5.18.00, July 2, 2010
Type ? for help. Commands must be C++ statements.
Enclose multiple statements between { }.
root [0]
```

Images in REANA

- When used in analysis, REANA captures
  - The software and its configuration, as defined in an image
  - The workflow
  - The data
- REANA will pull an image of your choice from Docker Hub or other registry
- The user can specify individual images (perhaps very different ones) for each step of the workflow as needed, providing a lot of flexibility of workflow definition

REANA - the workflows

- Very simple, intuitive description of linear workflows in YAML
  - Human-readable
  - Formulated as “steps” and “commands” within each step
  - Parameters (variables) can be used to facilitate modifications (save on typing)
  - Self-documenting (easy to add comments)
  - Often enough to describe final stages of analyses
- A more complex syntax is available to describe arbitrary workflow graphs
  - The “Common Workflow Language” - CWL
  - CWL is not covered today: we’ll save it for the next workshop

REANA - the data

- Data is almost never included in images
- REANA does not share its file systems with conventional machines
- So how do we handle the data?
- REANA has a concept of “workspaces”
  - Similar to scratch space on a batch system, but will persist until deleted
- The input data is “staged in” when the user submits a REANA workflow and the system initiates execution
- What specifically is “staged in” must be defined in the YAML input file
- The outputs can be downloaded (staged out) to the user’s workstation using the client

REANA: a simple example of the YAML input

```
version: 0.0.1
inputs:
  files:
    - ./hello_world_input/hello_world.txt
workflow:
  type: serial
  specification:
    steps:
      - environment: 'ubuntu:18.04'
        commands:
        - echo 'Hello World!' > helloworld1_ouput.txt
        - cat ./hello_world_input/hello_world.txt >> helloworld1_ouput.txt
outputs:
  files:
    - helloworld1_ouput.txt
```

Staging files to the workspace

Pulling image from Docker Hub

Output will be available for download from CLI and GUI

REANA interfaces

- In this exercise we'll access the REANA Web UI by setting up a SSH tunnel during our log-on to the SDCC interactive nodes. With the Web interface, we can
  - Check on the status of workflows running within the system
  - Download output files if necessary (also can use the CLI client)
- We'll use the REANA **CLI client** on interactive SDCC nodes since this way we can avoid the necessity to install anything on our laptops
- The client has a rich interface covering all aspects of REANA functionality

The interactive part

...please don’t hesitate to ask questions...

Log-in with SSH tunnel to access REANA

- You can use utilities like PuTTy, MobaXterm or others capable of ssh tunneling
- Alternatively, on Linux and Linux-like machines you can use straight SSH:

```
ssh -L 30443:kubmaster01.sdcc.bnl.gov:30443 USERNAME@ssh.sdcc.bnl.gov
```

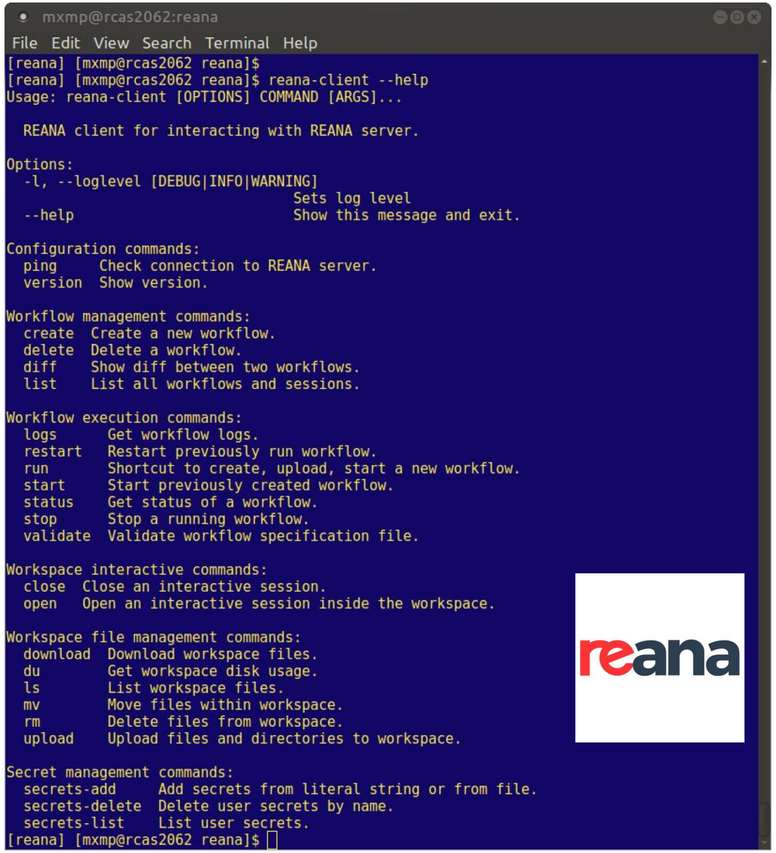
- Once connection is established the REANA Web page will be available at https://localhost:30443/
- You should be able to log into the service with the password you chose when you were applying for an account
- Have you logged on? Can you access your profile in the upper right corner and see the secret token? There is “visibility option” you should used since it’s normally blacked out on that page.

REANA client - the first “ping”

- Log onto an available interactive node (rcas206X machines)
- Use pre-fabricated Python environment to activate the client
  - source /direct/phenix+u/mxmp/.virtualenvs/reana/bin/activate.csh
- Set the environment variables (assuming you run tcsh, change this for bash):
  - setenv REANA_ACCESS_TOKEN *YOUR_PRIVATE_TOKEN*
  - setenv REANA_SERVER_URL https://kubmaster01.sdcc.bnl.gov:30443
- See if the client can run in your account
  - reana-client --help # prints a useful summary of commands, please try this
  - reana-client ping   # try connecting to the server and see the output
- What do you observe?

REANA client - the help screen

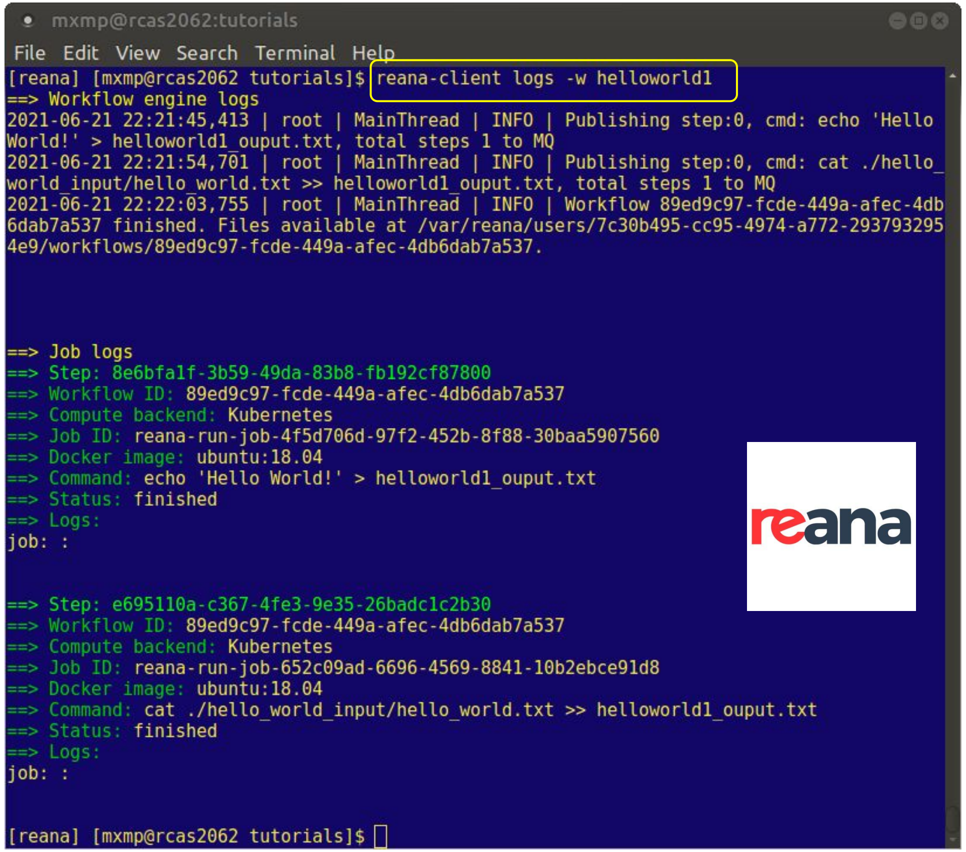
- Each command has its own help screen made available by attaching a “--help” option, e.g. reana-client list --help
- Please take a few minutes studying the available commands. The following are especially important
  - list
  - run
  - upload
  - download
  - delete
- For the “run” command, the following two options are particularly important
  - -w assign a name to your workflow
  - -f read YAML from a file of your choice
- Defaults will be provided but it’s best to name these explicitly



```
[reana] [mxmp@rcas2062 reana]$
[reana] [mxmp@rcas2062 reana]$ reana-client --help
Usage: reana-client [OPTIONS] COMMAND [ARGS]...

  REANA client for interacting with REANA server.

Options:
  -l, --loglevel [DEBUG|INFO|WARNING]
                                  Sets log level
  --help                          Show this message and exit.

Configuration commands:
  ping     Check connection to REANA server.
  version  Show version.

Workflow management commands:
  create  Create a new workflow.
  delete  Delete a workflow.
  diff    Show diff between two workflows.
  list    List all workflows and sessions.

Workflow execution commands:
  logs      Get workflow logs.
  restart   Restart previously run workflow.
  run       Shortcut to create, upload, start a new workflow.
  start     Start previously created workflow.
  status    Get status of a workflow.
  stop      Stop a running workflow.
  validate  Validate workflow specification file.

Workspace interactive commands:
  close  Close an interactive session.
  open   Open an interactive session inside the workspace.

Workspace file management commands:
  download  Download workspace files.
  du        Get workspace disk usage.
  ls        List workspace files.
  mv        Move files within workspace.
  rm        Delete files from workspace.
  upload    Upload files and directories to workspace.

Secret management commands:
  secrets-add     Add secrets from literal string or from file.
  secrets-delete  Delete user secrets by name.
  secrets-list    List user secrets.
[reana] [mxmp@rcas2062 reana]$
```

REANA client - more notes

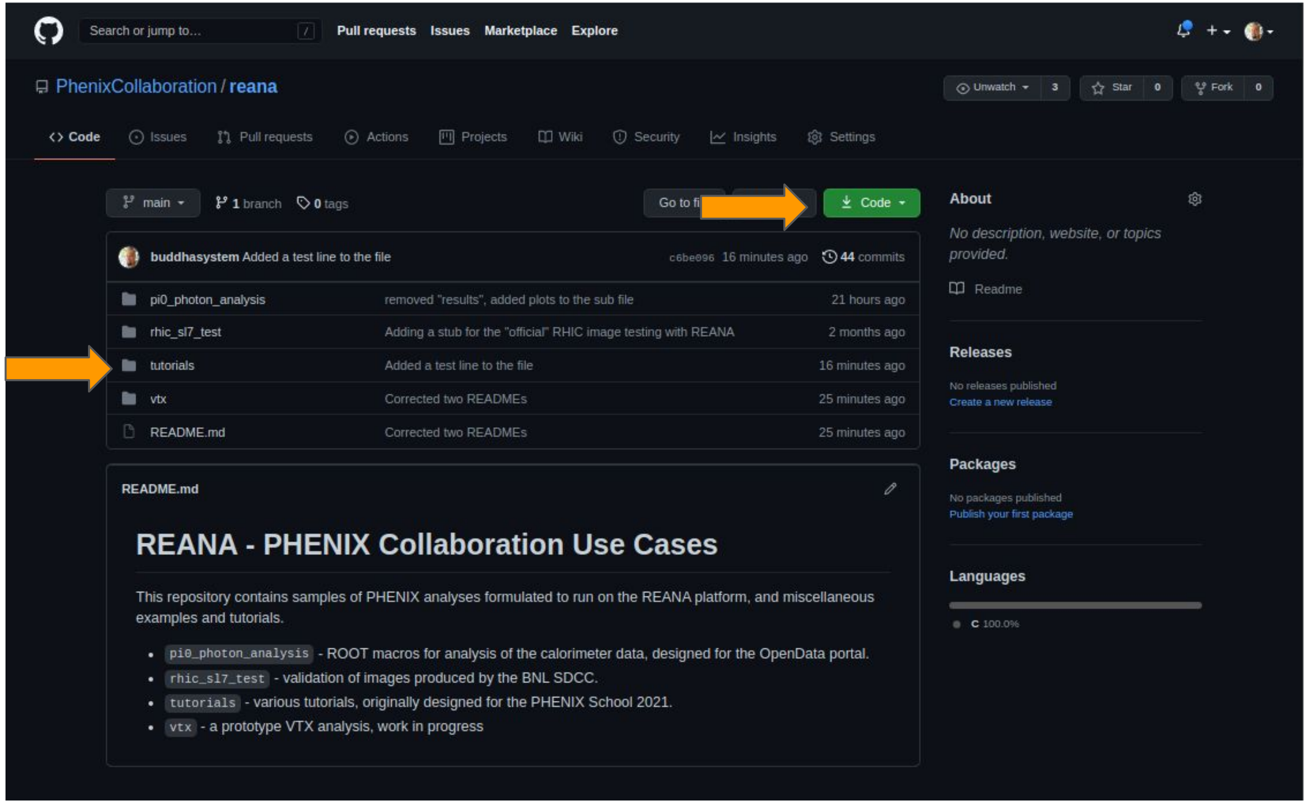
- The “-w” option will give your workflow a unique handle which you will need to use for any operation on that workflow
- Individual multiple runs of the same workflow are indexed automatically by REANA by assigning an integer, so you can have workflows test.1, test.2 etc
- The “logs” option, as shown in the right, will pring log information for each individual step in a workflow



```
[reana] [mxmp@rcas2062 tutorials]$ reana-client logs -w helloworld1
==> Workflow engine logs
2021-06-21 22:21:45,413 | root | MainThread | INFO | Publishing step:0, cmd: echo 'Hello
World!' > helloworld1_ouput.txt, total steps 1 to MQ
2021-06-21 22:21:54,701 | root | MainThread | INFO | Publishing step:0, cmd: cat ./hello_
world_input/hello_world.txt >> helloworld1_ouput.txt, total steps 1 to MQ
2021-06-21 22:22:03,755 | root | MainThread | INFO | Workflow 89ed9c97-fcde-449a-afec-4db
6dab7a537 finished. Files available at /var/reana/users/7c30b495-cc95-4974-a772-293793295
4e9/workflows/89ed9c97-fcde-449a-afec-4db6dab7a537.



==> Job logs
==> Step: 8e6bfa1f-3b59-49da-83b8-fb192cf87800
==> Workflow ID: 89ed9c97-fcde-449a-afec-4db6dab7a537
==> Compute backend: Kubernetes
==> Job ID: reana-run-job-4f5d706d-97f2-452b-8f88-30baa5907560
==> Docker image: ubuntu:18.04
==> Command: echo 'Hello World!' > helloworld1_ouput.txt
==> Status: finished
==> Logs:
job: :


==> Step: e695110a-c367-4fe3-9e35-26badc1c2b30
==> Workflow ID: 89ed9c97-fcde-449a-afec-4db6dab7a537
==> Compute backend: Kubernetes
==> Job ID: reana-run-job-652c09ad-6696-4569-8841-10b2ebce91d8
==> Docker image: ubuntu:18.04
==> Command: cat ./hello_world_input/hello_world.txt >> helloworld1_ouput.txt
==> Status: finished
==> Logs:
job: :

[reana] [mxmp@rcas2062 tutorials]$
```

Get the code from GitHub: https://github.com/PhenixCollaboration/reana

- Use the “code” option
- Clone the repo to a folder in your SDCC account (not your laptop)
- Go to “tutorials”
- Make sure you can access the REANA web page on your laptop - use the SSH tunnel and the URL https://localhost:30443/
- If this is your first time using REANA you’ll only see a welcome message



buddhasystem Added a test line to the file c6be096 16 minutes ago 44 commits

pi0_photon_analysis	removed "results", added plots to the sub file	21 hours ago
rhic_sl7_test	Adding a stub for the "official" RHIC image testing with REANA	2 months ago
tutorials	Added a test line to the file	16 minutes ago
vtx	Corrected two READMEs	25 minutes ago
README.md	Corrected two READMEs	25 minutes ago

No description, website, or topics provided.

No releases published
Create a new release

No packages published
Publish your first package

README.md

REANA - PHENIX Collaboration Use Cases

This repository contains samples of PHENIX analyses formulated to run on the REANA platform, and miscellaneous examples and tutorials.

- `pi0_photon_analysis` - ROOT macros for analysis of the calorimeter data, designed for the OpenData portal.
- `rhic_sl7_test` - validation of images produced by the BNL SDCC.
- `tutorials` - various tutorials, originally designed for the PHENIX School 2021.
- `vtx` - a prototype VTX analysis, work in progress

REANA client - the first run

- Assuming your “ping” attempt worked (try again if it didn’t)
- In the “tutorials” folder, run the command
  - reana-client run -f helloworld1.yaml -w helloworld1
- See if the workflow has been initiated
  - reana-client list
- Also, please reload the REANA web page
- Execution will take some time but should be less than a minute or so
- What do you observe?

REANA client - getting the data back

- Assuming successful completion of the workflow -
- In the “tutorials” folder, run the command
    - reana-client download -w helloworld1
- See if the output file specified in the YAML workflow description was downloaded
    - helloworld1_ouput.txt
- Also, please look the REANA web page and check the “workspace”
- What do you observe?

REANA client - modifying inputs

- Edit the file hello_world_input/hello_world.txt
  - You can add lines, any text etc...

- In the “tutorials” folder, run the command
  - reana-client upload -w helloworld1 hello_world_input/hello_world.txt

- Check the “workspace” in the Web interface (when clicking on the workflow)
- Do you see that the input file has changed?
- Run the workflow again, download the result and see if the output is now different
  - helloworld1_ouput.txt

- What do you observe? Since you ran the workflow under the same name, do you see the serial integer attached to the new instance?

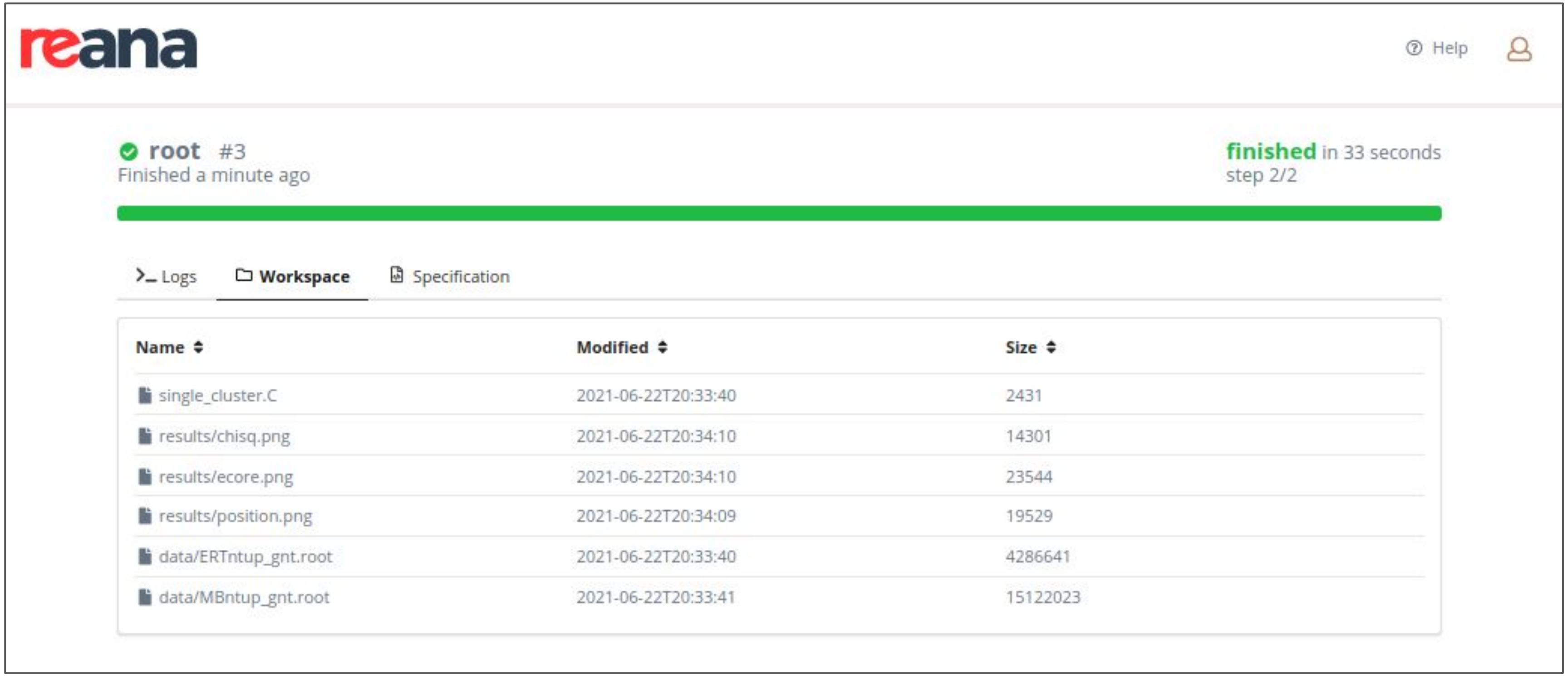
REANA: multiple outputs in one folder

- Please inspect helloworld2.yaml
  - Note the “parameters” and the “results” folder
- In the “tutorials” folder, run the command
  - reana-client run -f helloworld2.yaml -w helloworld2
  - reana-client list # to check on the workflow
- Run the workflow again and download the result
  - reana-client download -w helloworld2
- What do you observe? Do you now see the “results” folder? Inspect the YAML file again to see why this happened.

REANA: an EMCAL analysis example

- In this example, we'll use the data and materials originally developed for the CERN OpenData portal.
- Please go to the "pi0_photon_analysis" folder in the "reana" repository you cloned.
- Please inspect the macro single_cluster.C and the submission file reana.yaml
- This can be run as: reana-client run -w root
- Note the following
  - No need to specify "-f" since the YAML file has the default name "reana.yaml"
  - You can use any name instead of "root" to mark your workflow
  - There are two Ntuple files to be staged in before the execution of the macro
  - We use a standard ROOT6 macro (adapted from the original ROOT5) and a standard ROOT6 image to be pulled from Docker Hub

REANA: running the EMCAL example



root #3
Finished a minute ago

finished in 33 seconds
step 2/2

Name	Modified	Size
single_cluster.C	2021-06-22T20:33:40	2431
results/chisq.png	2021-06-22T20:34:10	14301
results/ecore.png	2021-06-22T20:34:10	23544
results/position.png	2021-06-22T20:34:09	19529
data/ERTntup_gnt.root	2021-06-22T20:33:40	4286641
data/MBntup_gnt.root	2021-06-22T20:33:41	15122023

REANA EMCAL example - getting the data back

- Assuming successful completion of the workflow -
- In the “pi0_photon_analysis” folder, run the command
  - reana-client download -w root
- See if the output files specified in the YAML workflow description was downloaded in the “results” folder

- Also, please look the REANA web page and check the “workspace”
  - You should be able to preview the images in PNG format

- What do you observe?

- For more details on this containerized analysis please see https://opendata.cern.ch/record/15011

Thank you!