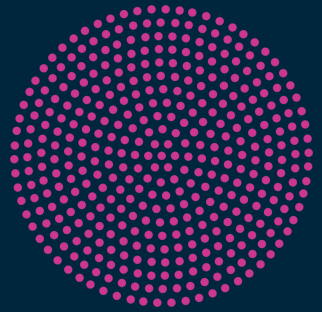
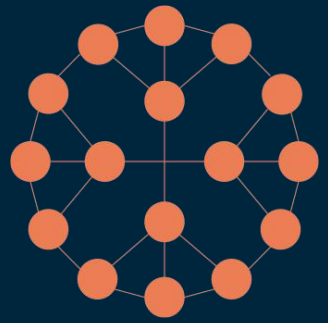
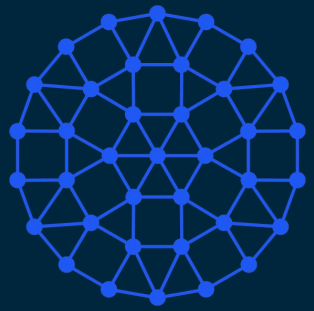
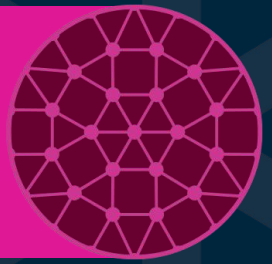



mlflow with Run:ai

mlflow overview

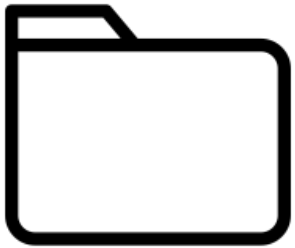


What is needed for mlflow to run?

1) A Database to store information related to experiment runs
2) An Artifacts folder to store objects related to the runs

Database

Artifacts folder



How does mlflow work?

1) First we start the server with a CLI command

running either of these commands will automatically create a databse in the local directory, if one doesn't exist.

```
mlflow ui
```

(or)

```
mlflow server
```

we can also choose to specify the location of the database, and artifact folder, as well as the host IP, and port.

```
mlflow server \
    --backend-store-uri=sqlite:///abs/path/to/db/mlflow.db \
     --default-artifact-root=/abs/path/to/artifacts \
     --host=0.0.0.0 \
     --port=5000
```

How does mlflow work?

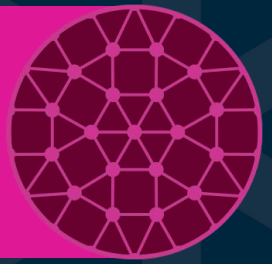
2) call mlflow commands within the python script

you'll want to import mlflow, then set the tracking uri so that mlflow will save everything to the database and artifact folder.

Then you'll want to start your run, and at the end, you'll want to end the run.

```
import mlflow

mlflow.set_tracking_uri('0.0.0.0:5000')
mlflow.start_run()

"""
your code here
"""

mlflow.end_run()
```

mlflow with run:ai



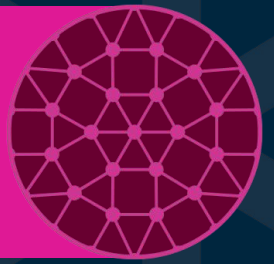
What is needed for mlflow to run on run:ai?

1) A persistent directory to keep
- mlflow database
- mlflow artifacts folder

2) A docker image with the following installed
- mlflow
- jupyterlab*
- jupyter-server-proxy*

*needed in order to access the mlflow UI

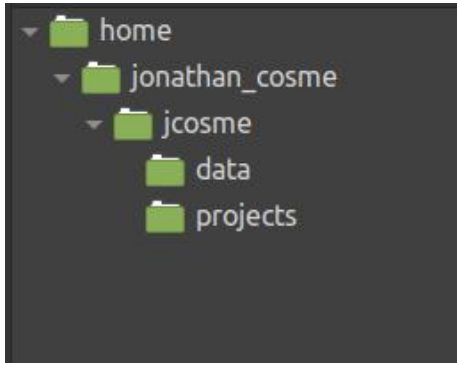
creating persistent directory



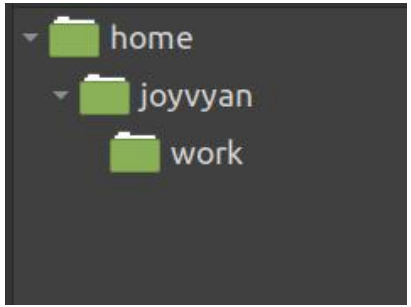
Before we start

We need to create an ‘mlflow’ folder on our NFS.

This is what our NFS folder structure looks like now



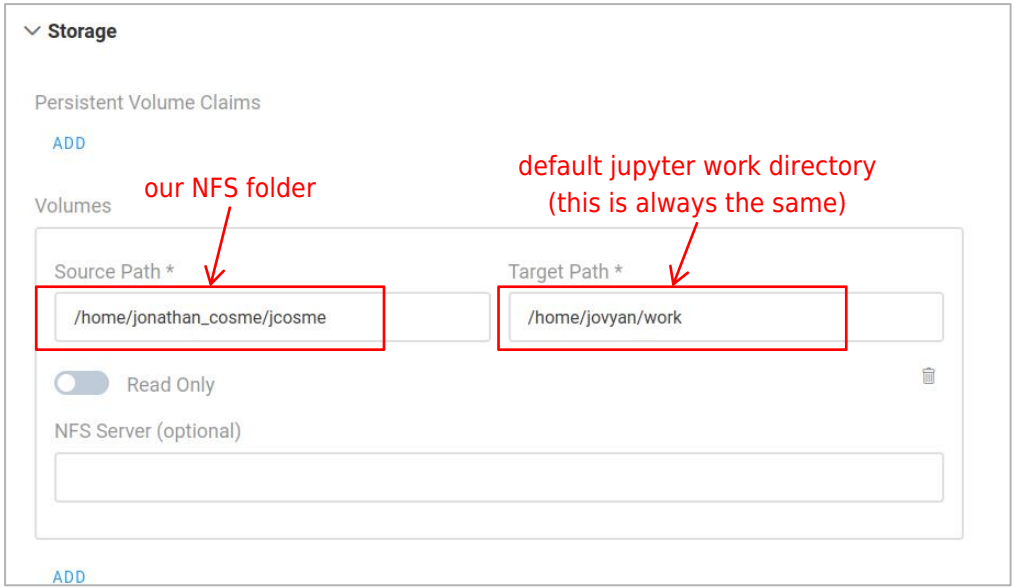
This is the default folder structure for our jupyter lab image



Before we start

We need to create an ‘mlflow’ folder on our NFS.

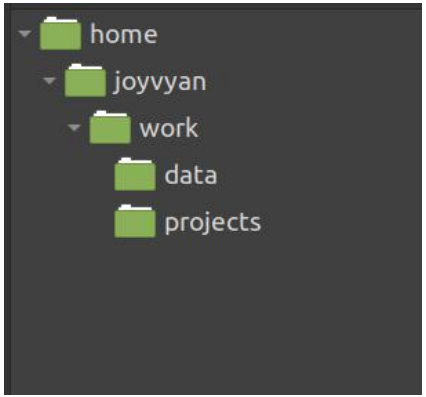
For *our example:*
Whenever we create a job on run:ai, we *must always* mount our NFS to the default jupyter work directory



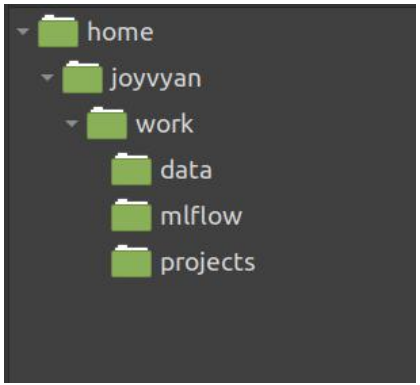
Before we start

We need to create an ‘mlflow’ folder on our NFS.

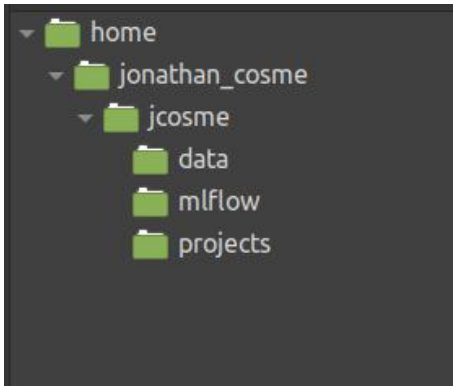
1. After we mount our NFS volume, our work directory will look like this:



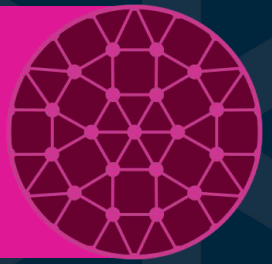
2. Using Jupyter Lab, we create an ‘mlflow’ folder within the work directory:



3. This will cause our NFS directory to automatically look like this:



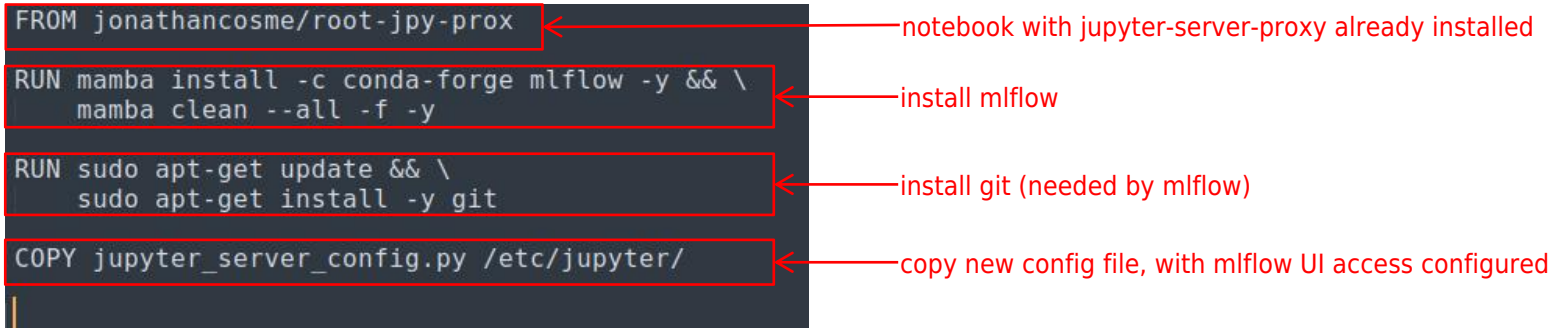
docker image



Docker used in our example

The docker image we will use is:
jonathancosme/mlflow-ui

This is what is in the dockerfile:



Docker used in our example

The docker image we will use is:
jonathancosme/mlflow-ui

in order to access the mlflow UI, we need to add this entry to the jupyter_server_config.py file, and replace the existing file in the image

```
c.ServerProxy.servers = {
    'mlflow-server': {
        'command': [
            'mlflow', 'server',
                '--backend-store-uri=sqlite:///home/jovyan/work/mlflow/mlflow.db',
                '--default-artifact-root=/home/jovyan/work/mlflow/artifacts',
                '--host=0.0.0.0',
                '--port=5000',
                '--serve-artifacts',
            ],
        'timeout': 30,
        'launcher_entry': {
            'title': 'mlflow'
        },
        'port':5000

    }
}
```

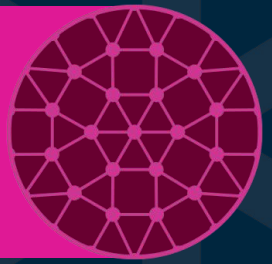
we specify our database and artifact folder locations (this is why we must always mount our NFS directory to the default jupyter work directory)

we must make sure to start the server on this IP and port

This tells jupyter to forward port 5000 to the URL

github.com/jonathancosme/runai_mlflow_demo

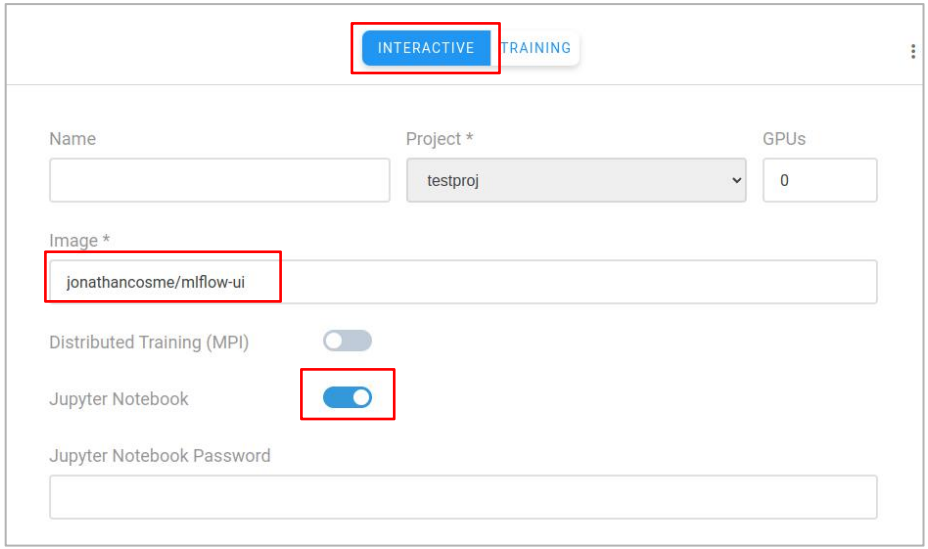
Accessing the mlflow UI

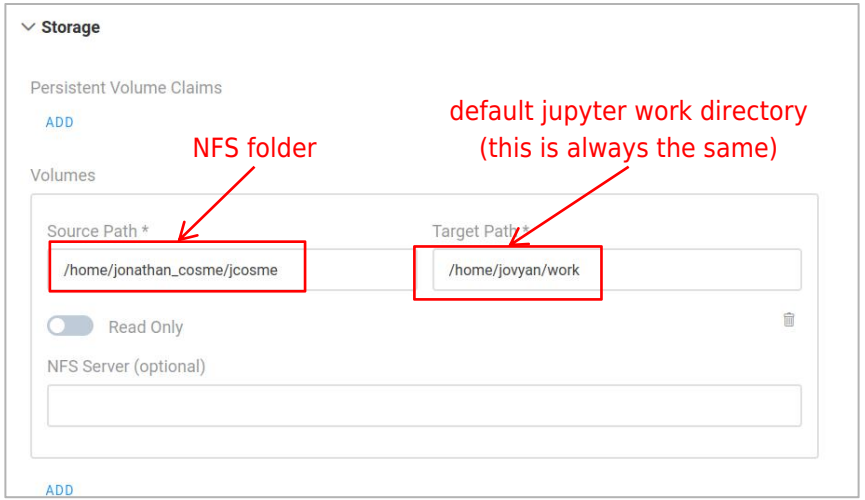


Access mlflow UI

Create a jupyter interactive job with:

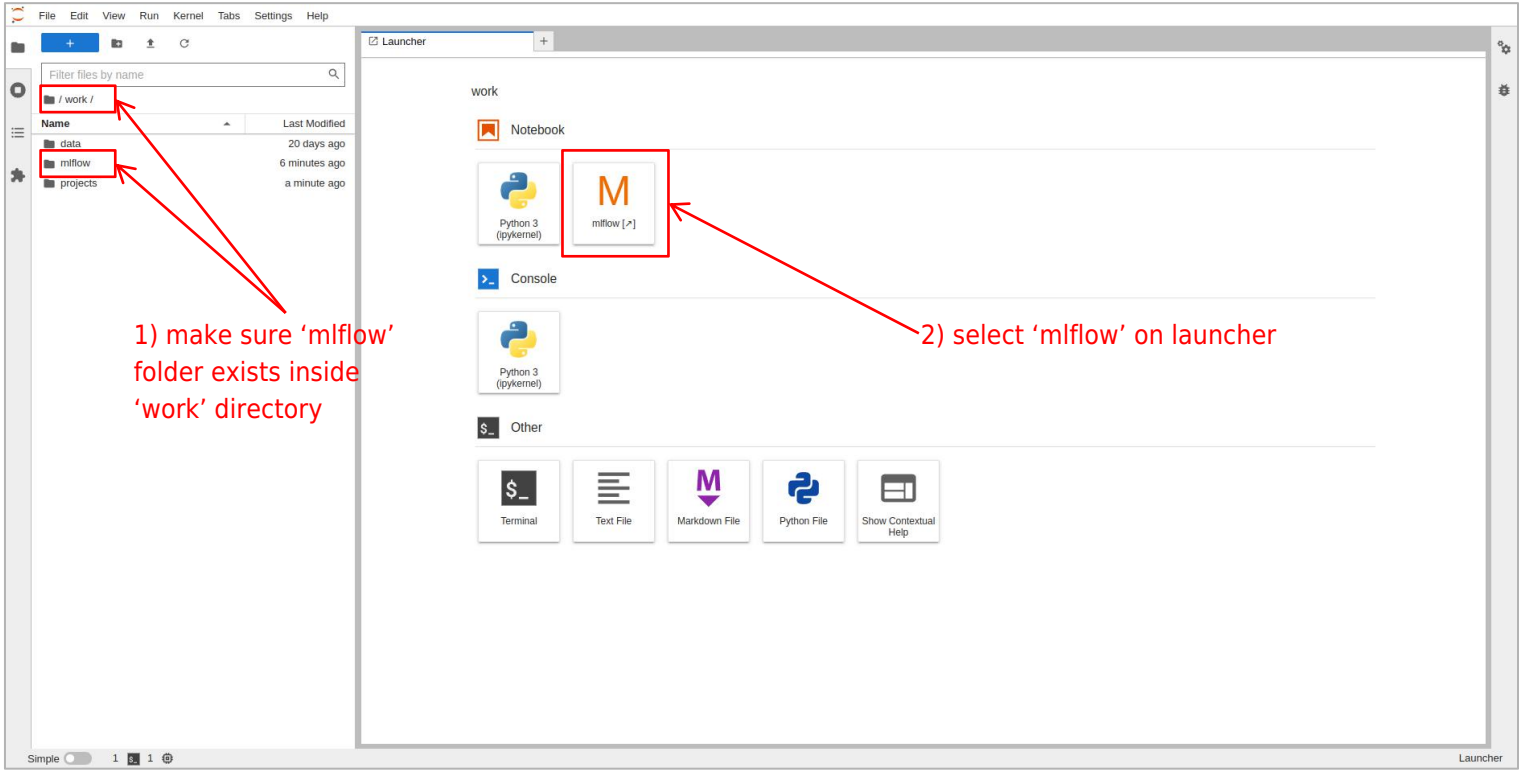
- image jonathancosme/mlflow-iu
- mounted NFS folder (with ‘mlflow’ folder) in default jupyter work directory





github.com/jonathancosme/runai_mlflow_demo

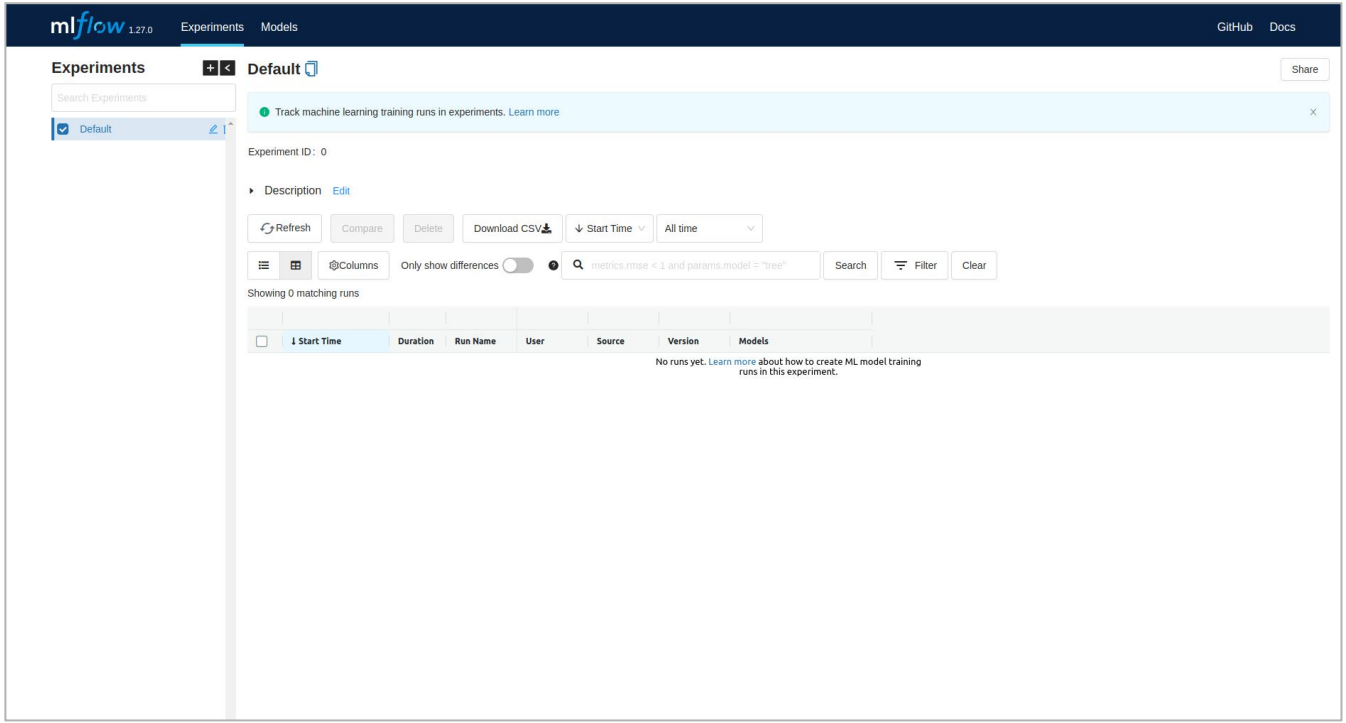
Access mlflow UI



github.com/jonathancosme/runai_mlflow_demo

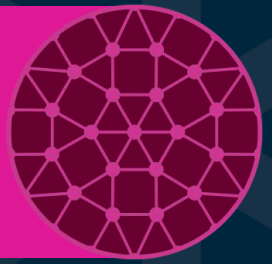
Access mlflow UI

A new tab should appear with the mlflow UI



github.com/jonathancosme/runai_mlflow_demo

running mlflow experiments with run:ai



Python scripts

in your script, you should first launch the mlflow server

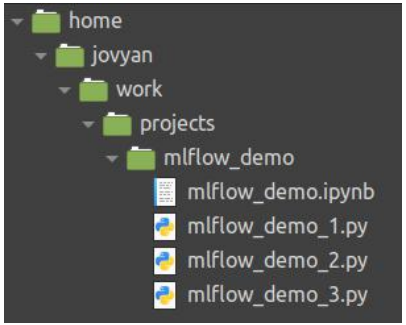
Then be sure to set the tracking uri

```python
import mlflow
import os
import time
from subprocess import Popen

# first we start the mlflow server
mlflow_command = 'mlflow server --backend-store-uri=sqlite:////home/jovyan/work/mlflow/
    mlflow.db --default-artifact-root=/home/jovyan/work/mlflow/artifacts --host=0.0.0.0
    --port=5000 --serve-artifacts'
print(f'running command: {mlflow_command}')
proc = Popen([mlflow_command], shell=True,
             stdin=None, stdout=None, stderr=None, close_fds=True)
print(f'waiting 5 seconds...')
time.sleep(5) # we wait 5 seconds to give the server time to start

# create the tracking uri string
uri = 'http://0.0.0.0:5000' # in our example, this is always the same
project_name = 'test_project'
run_name = 'run_1'

mlflow.set_tracking_uri(uri)
mlflow.set_experiment(experiment_name=project_name)
mlflow.start_run(run_name=run_name)

"""
your code here
"""

mlflow.end_run()
```

github.com/jonathancosme/runai_mlflow_demo

CLI submission

Our example scrips are located here:



so our CLI command would look like this:

```
runai submit \
    --project testproj \
    --gpu 0 \
    --job-name-prefix mlflow-demo \
    --image jonathancosme/mlflow-ui \
    --volume /home/jonathan_cosme/jcosme:/home/jovyan/work \
    -- python work/projects/mlflow_demo/mlflow_demo_1.py
```

make sure you use the mlflow-ui docker image

make sure you mount the NFS to the work directory

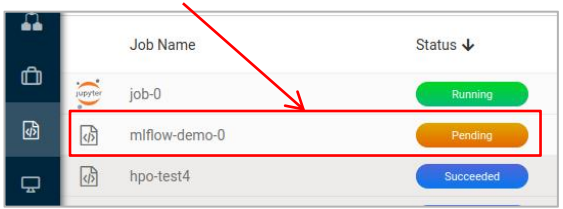
The code to run the job must specify the python script with relative location

Example job submission

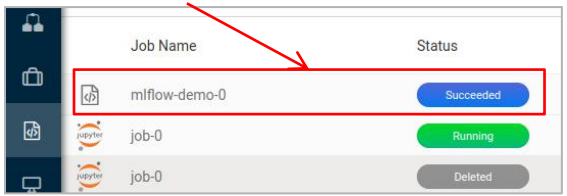
1) submit CLI command

```
jcosme@Jane:~$ runai submit \
>     --project testproj \
>     --gpu 0 \
>     --job-name-prefix mlflow-demo \
>     --image jonathancosme/mlflow-ui \
>     --volume /home/jonathan_cosme/jcosme:/home/jovyan/work \
>     -- python work/projects/mlflow_demo/mlflow_demo_1.py
```

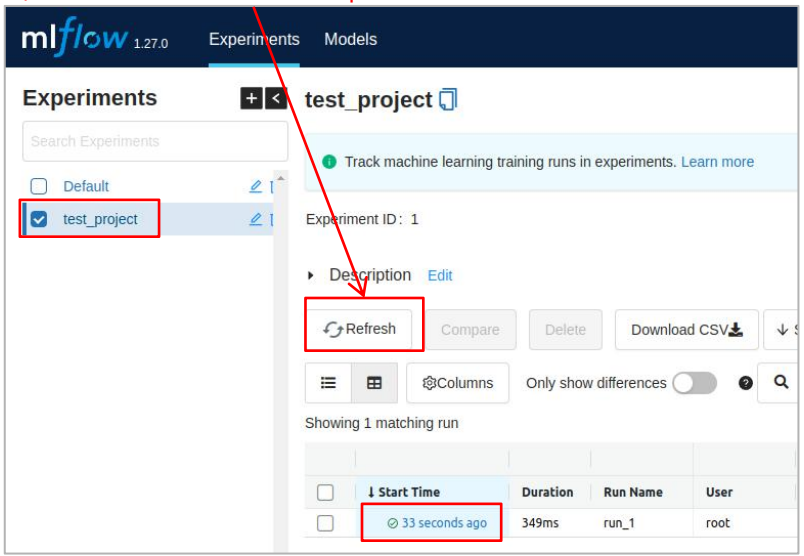
2) a new job should appear



3) wait for job to finish

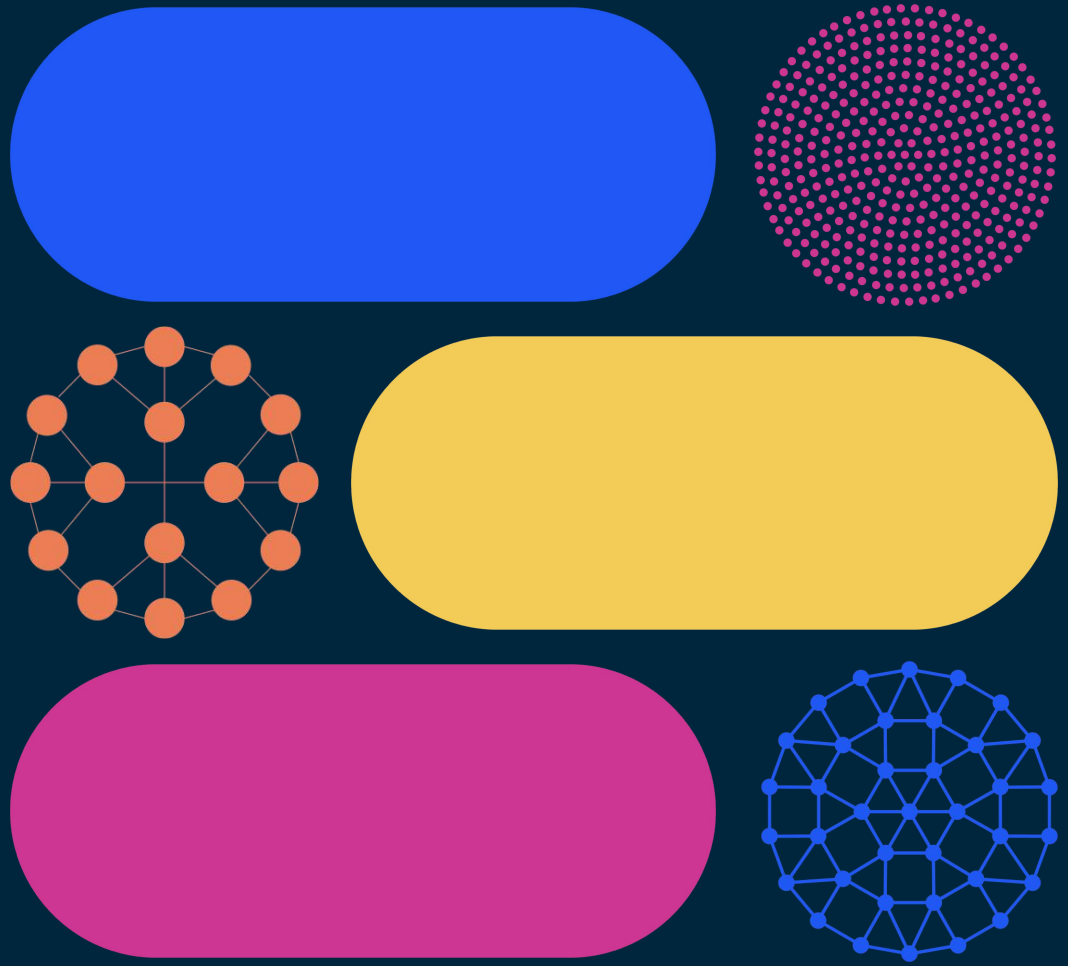


4) refresh mlflow UI to see updates



github.com/jonathancosme/runai_mlflow_demo

Thank you!



info@run.ai | www.run.ai