Persistent Environments with Run:ai

(using conda/mamba & Jupyter)

Background

A (specific) frequently asked question

Client

Ok so I think I get Docker and how it works with Run:ai… but… does this mean I have to make a Docker image for EVERY single one of my conda environments??

Run:ai

Well, that would be the preferred way to do things.

Client

Is there a way I can create and save environments on-the-fly, like I do in my normal workflow?

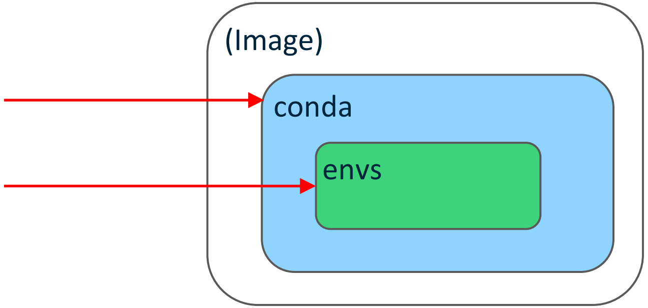
Answer: Yes!

Solution

How do we do this?

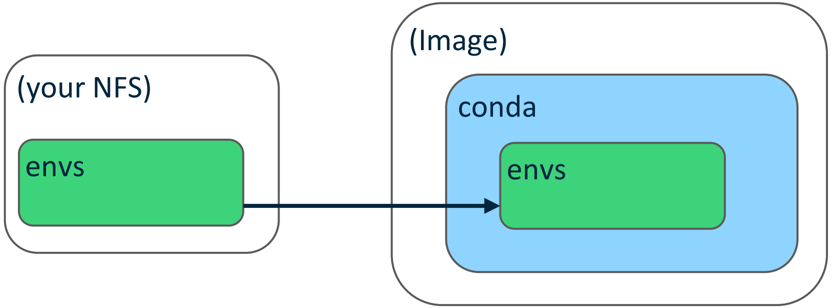
Conda background

1. When conda is installed, it creates a directory to keep its files.
2. Whenever a conda environment is created, it is saved inside a directory called "envs," within the conda directory.



The idea

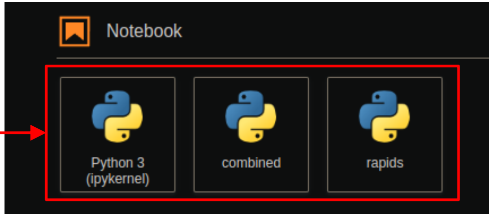
1. **Create** a **folder** called "**envs**" **in** your **NFS**
2. **Mount** the **NFS** "envs" folder, **to** the location of the **conda environments folder** in your image.



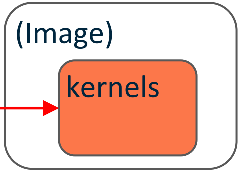
github.com/jonathancosme/runai_persist_envs

A note on Jupyter

Many users like to have **‘tiles’ for their environment** on Jupyter

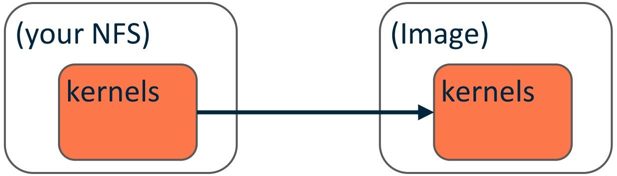


These setting are saved in a folder called ‘kernels’



We can **persist** these ‘tiles’ by:

1. **Creating** a **folder** called “**kernels**” **in** your **NFS**
2. **Mounting** the **NFS** “kernels” folder, **to** the location of the **kernels folder** in your image.



Adding a tile is done with the following sequence of commands, then **refreshing the web browser**

```
mamba create -n {your-env-name} ipykernel {your-env-libs}
```

```
conda activate {your-env-name}
```

```
python -m ipykernel install --user --name={your-env-name}
```

(If you want to remove the ‘tile’, this command is used)

```
jupyter kernelspec uninstall {your-env-name}
```

github.com/jonathancosme/runai_persist_envs

Docker image

The docker image that we will be using throughout this notebook is:
jonathancosme/base-notebook-root*

It is a slightly modified version of the
jupyter/base-notebook
image

You can find more information about jonathancosme/base-notebook-root here:
https://github.com/jonathancosme/jupyter-base-notebook-root

*Please note that all our Jupyter images are based off the official Jupyter Docker images. The official images do not have root privileges by default. In order to successfully mount to the ‘kernels’ and ‘envs’ folders in the Jupyter Lab, root privileges are required. We created a custom image that runs root by default. You can see how our image differs from the official image on the github.

Steps

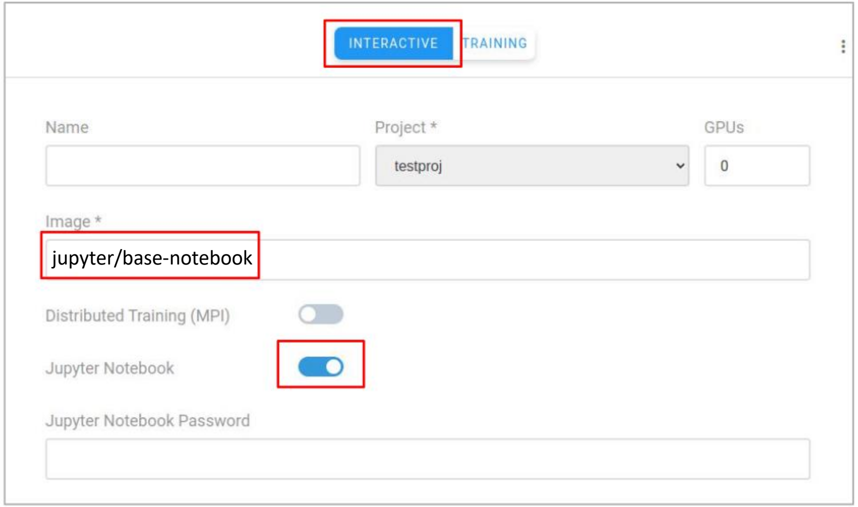
Steps

1. Create ‘envs’ and ‘kernels’ folder in our NFS
2. Mount NFS ‘envs’ and ‘kernels’ folders on an interactive Jupyter lab job
3. Create your environment
4. Test the environment

Step 1

Create ‘envs’ and ‘kernels’ folder in our NFS

1.1 Start a Jupyter Lab interactive session, mounting your NFS folder to /home/jovyan/work



Name:

Project *: testproj

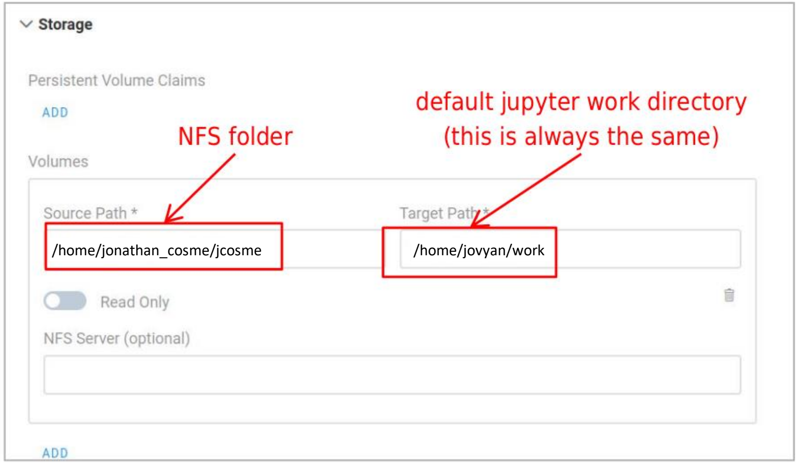
GPUs: 0

Image *: jupyter/base-notebook

Distributed Training (MPI): off

Jupyter Notebook: on

Jupyter Notebook Password:



Source Path *: /home/jonathan_cosme/jcosme

Target Path *: /home/jovyan/work

Read Only: off

NFS Server (optional):

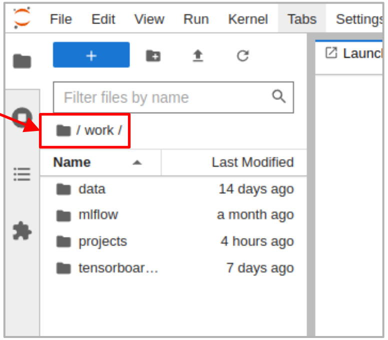
NFS folder

default jupyter work directory (this is always the same)

Step 1

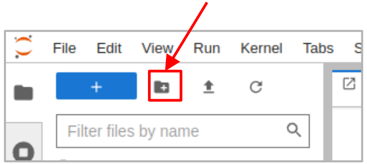
Create ‘envs’ and ‘kernels’ folder in our NFS

1.2 Navigate to the ‘work’ directory. You should see your files/folders inside.

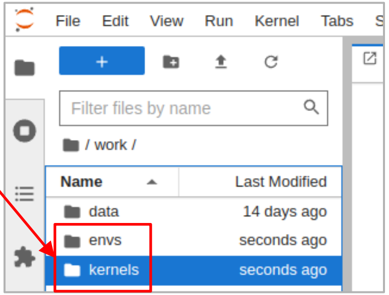


Name	Last Modified
data	14 days ago
mlflow	a month ago
projects	4 hours ago
tensorboar…	7 days ago

1.3 Select the new folder icon, twice (to create two new folders)

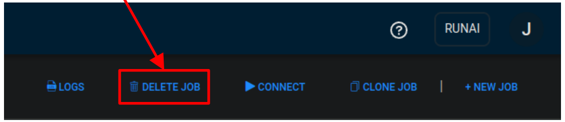


1.4 Rename the folders ‘envs’ and ‘kernels’



Name	Last Modified
data	14 days ago
envs	seconds ago
kernels	seconds ago

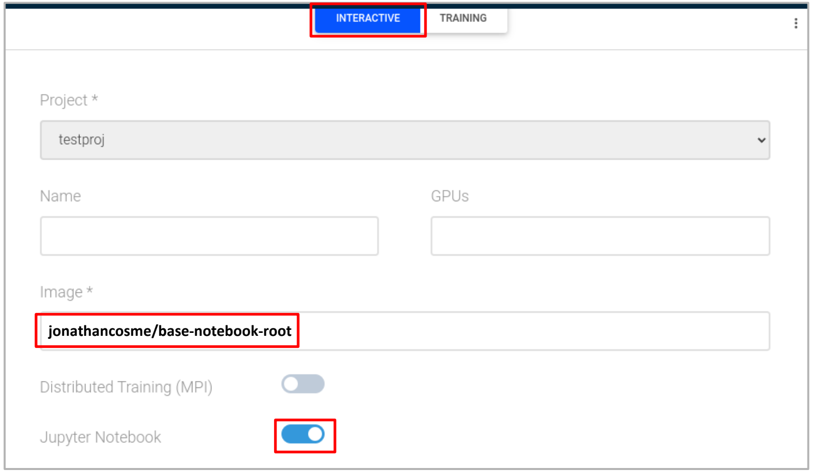
1.5 Delete the job



Step 2

Mount NFS ‘envs’ and ‘kernels’ folders on an interactive Jupyter lab job

2.1 Start a Jupyter Lab interactive session using the image **jonathancosme/base-notebook-root** (do NOT submit job yet)



Project *: testproj

Name:

GPUs:

Image *: jonathancosme/base-notebook-root

Distributed Training (MPI): off

Jupyter Notebook: on

Step 2

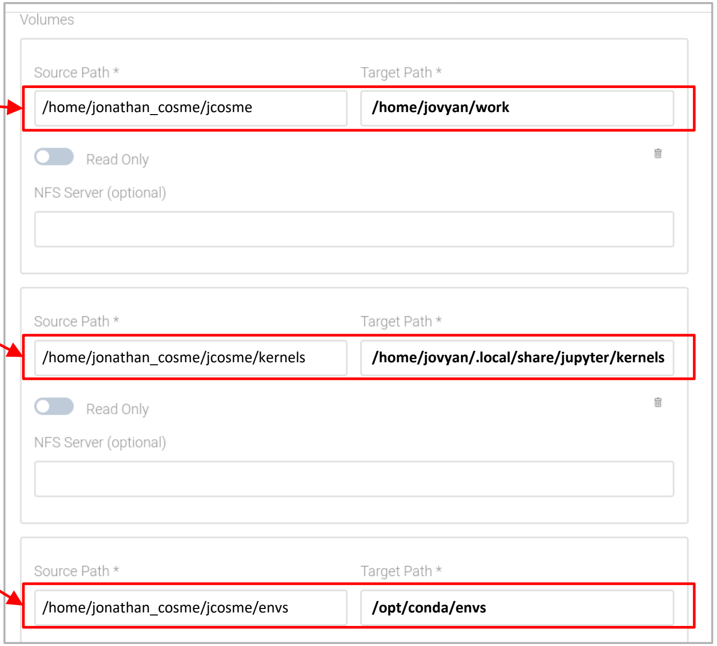
Mount NFS ‘envs’ and ‘kernels’ folders on an interactive Jupyter lab job

2.2 mounts NFS to /home/jovyan/work

2.3 mounts NFS ‘kernels’ folder to /home/jovyan/.local/share/jupyter/kernels

2.4 mounts NFS ‘envs’ folder to /opt/conda/envs

2.5 Submit job



Volumes

Source Path * /home/jonathan_cosme/jcosme

Target Path * /home/jovyan/work

Read Only

NFS Server (optional)

Source Path * /home/jonathan_cosme/jcosme/kernels

Target Path * /home/jovyan/.local/share/jupyter/kernels

Read Only

NFS Server (optional)

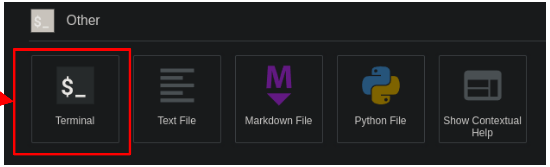
Source Path * /home/jonathan_cosme/jcosme/envs

Target Path * /opt/conda/envs

Step 3

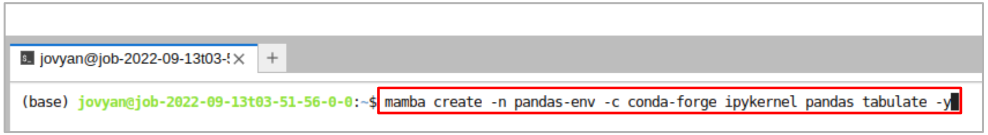
Create your environment

3.1 start a terminal, after connecting to the Jupyter lab interactive job



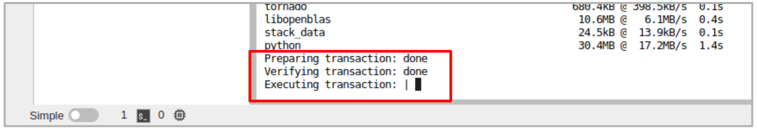
3.2 create an environment using the following format
(ipykernel must always be installed in an environment to create a tile)

```
mamba create -n {your-env-name} -c conda-forge ipykernel {your-env-libraries} -y
```



```
jovyan@job-2022-09-13t03-!
(base) jovyan@job-2022-09-13t03-51-56-0-0:~$ mamba create -n pandas-env -c conda-forge ipykernel pandas tabulate -y
```

3.3 wait for environment to finish creating.
Warning: depending on your NFS speed, the ‘Executing transaction’ step can take a long time (~3 minutes for me)



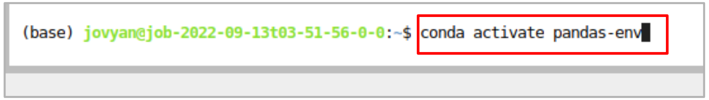
```
tornado                                  680.4kB @ 398.5kB/s  0.1s
libopenblas                               10.6MB @   6.1MB/s  0.4s
stack_data                                24.5kB @  13.9kB/s  0.1s
python                                    30.4MB @  17.2MB/s  1.4s
Preparing transaction: done
Verifying transaction: done
Executing transaction: |
```

Step 3

Create your environment

3.4 activate the newly created environment using the following format

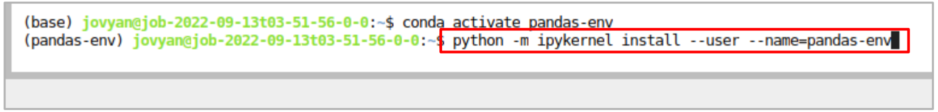
```
conda activate {your-env-name}
```



```
(base) jovyan@job-2022-09-13t03-51-56-0-0:~$ conda activate pandas-env
```

3.5 Create the tile by using the following format

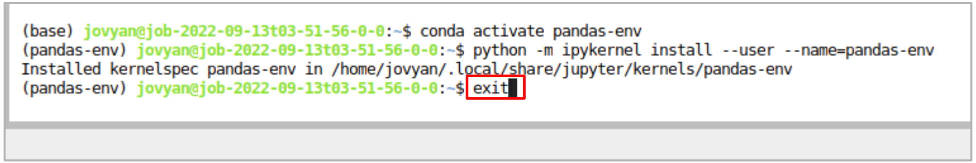
```
Python -m ipykernel install --user --name={your-env-name}
```



```
(base) jovyan@job-2022-09-13t03-51-56-0-0:~$ conda activate pandas-env
(pandas-env) jovyan@job-2022-09-13t03-51-56-0-0:~$ python -m ipykernel install --user --name=pandas-env
```

3.6 exit the terminal with the following command

```
exit
```

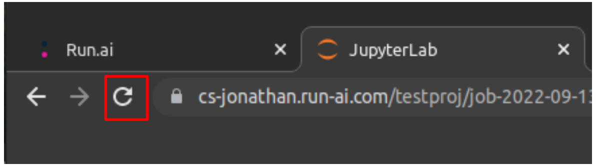


```
(base) jovyan@job-2022-09-13t03-51-56-0-0:~$ conda activate pandas-env
(pandas-env) jovyan@job-2022-09-13t03-51-56-0-0:~$ python -m ipykernel install --user --name=pandas-env
Installed kernelspec pandas-env in /home/jovyan/.local/share/jupyter/kernels/pandas-env
(pandas-env) jovyan@job-2022-09-13t03-51-56-0-0:~$ exit
```

Step 3

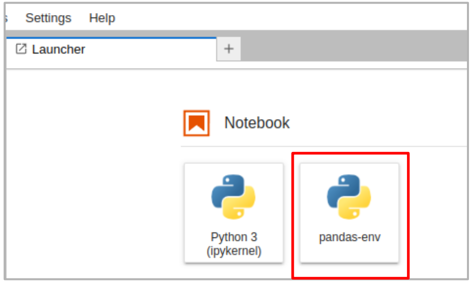
Create your environment

3.7 Refresh the Jupyter Lab tab on your browser

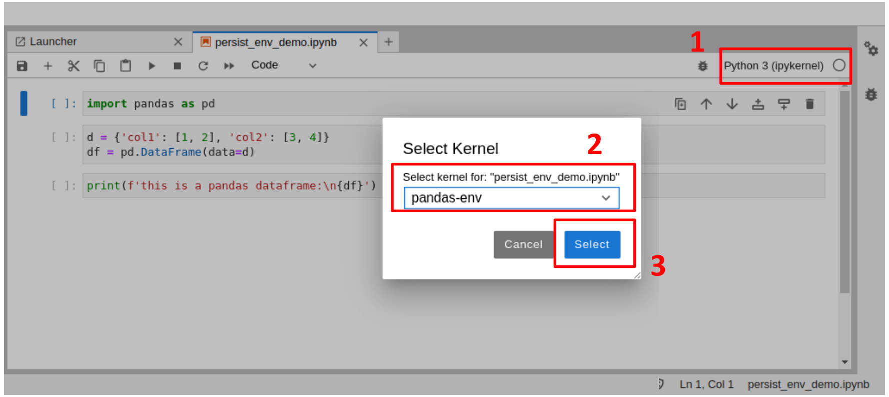


cs-jonathan.run-ai.com/testproj/job-2022-09-1

3.8 A tile for your new environment should now be visible



3.9 You can also select the new env is notebooks



```
[ ]: import pandas as pd

[ ]: d = {'col1': [1, 2], 'col2': [3, 4]}
     df = pd.DataFrame(data=d)

[ ]: print(f'this is a pandas dataframe:\n{df}')
```

Select Kernel

Select kernel for: "persist_env_demo.ipynb"

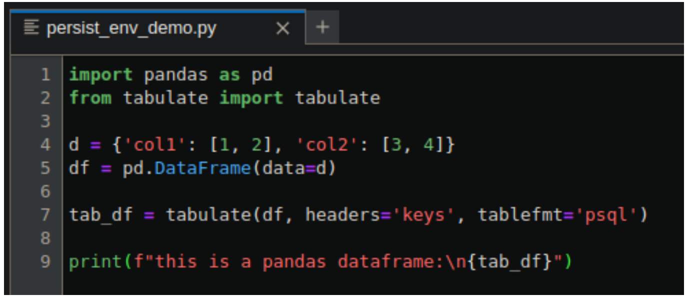
pandas-env

github.com/jonathancosme/runai_persist_envs

Step 4

Test the environment

Let's assume I have this python script….



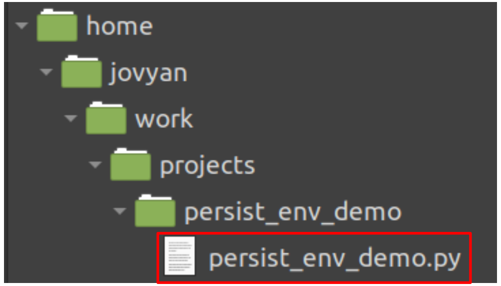
```python
import pandas as pd
from tabulate import tabulate

d = {'col1': [1, 2], 'col2': [3, 4]}
df = pd.DataFrame(data=d)

tab_df = tabulate(df, headers='keys', tablefmt='psql')

print(f"this is a pandas dataframe:\n{tab_df}")
```

…located here (after mounting my NFS to the Jupyter Lab work directory)

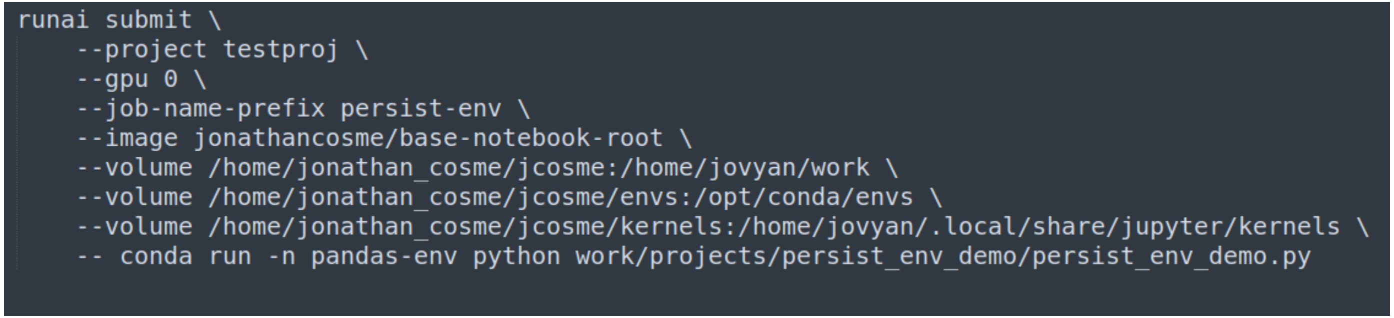


github.com/jonathancosme/runai_persist_envs

Step 4

Test the environment

This is the runai CLI command I want to run

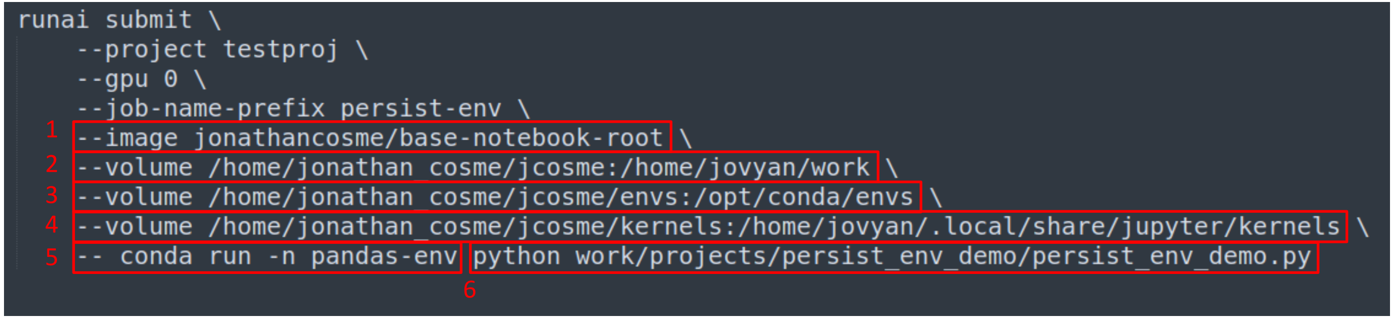


```
runai submit \
    --project testproj \
    --gpu 0 \
    --job-name-prefix persist-env \
    --image jonathancosme/base-notebook-root \
    --volume /home/jonathan_cosme/jcosme:/home/jovyan/work \
    --volume /home/jonathan_cosme/jcosme/envs:/opt/conda/envs \
    --volume /home/jonathan_cosme/jcosme/kernels:/home/jovyan/.local/share/jupyter/kernels \
    -- conda run -n pandas-env python work/projects/persist_env_demo/persist_env_demo.py
```

Step 4

Test the environment

This is the runai CLI command I want to run



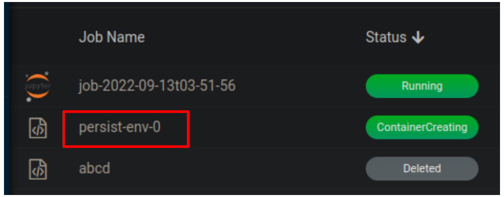
```
runai submit \
    --project testproj \
    --gpu 0 \
    --job-name-prefix persist-env \
    --image jonathancosme/base-notebook-root \
    --volume /home/jonathan_cosme/jcosme:/home/jovyan/work \
    --volume /home/jonathan_cosme/jcosme/envs:/opt/conda/envs \
    --volume /home/jonathan_cosme/jcosme/kernels:/home/jovyan/.local/share/jupyter/kernels \
    -- conda run -n pandas-env python work/projects/persist_env_demo/persist_env_demo.py
```

1. Use the image jonathancosme/base-notebook-root
2. Mount NFS to Jupyter work directory
3. Mount NFS ‘envs’ folder to Jupyter conda envs folder
4. Mount NFS ‘kernels’ folder to Jupyter kernels folder
5. Run our command (to follow) in the conda environment named ‘pandas-env’
6. Run our python script

Step 4

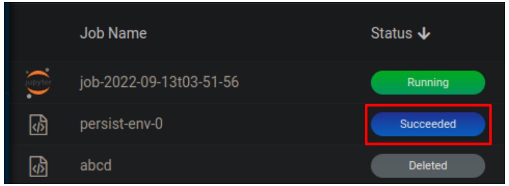
Test the environment

4.1 Run the runai CLI command. You should see a new job in the job list



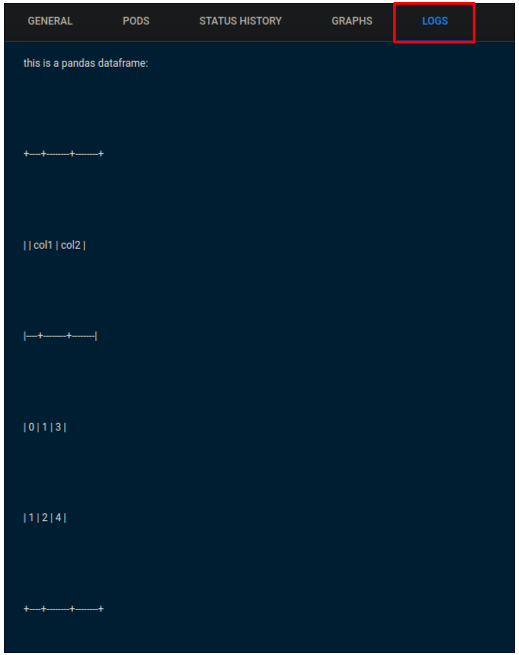
	Job Name	Status ↓
	job-2022-09-13t03-51-56	Running
	persist-env-0	ContainerCreating
	abcd	Deleted

4.2 Wait for the job status to switch to 'Succeeded'



	Job Name	Status ↓
	job-2022-09-13t03-51-56	Running
	persist-env-0	Succeeded
	abcd	Deleted

4.3 View the output of the jobs in the Logs tab



```
this is a pandas dataframe:

+---+--------+--------+

| | col1 | col2 |

|---+--------+--------|

| 0 | 1 | 3 |

| 1 | 2 | 4 |

+---+--------+--------+
```

Thank you!

info@run.ai | www.run.ai