



W&B + run:
ai

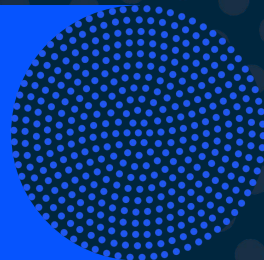


Weight & Biases
with
Run:ai

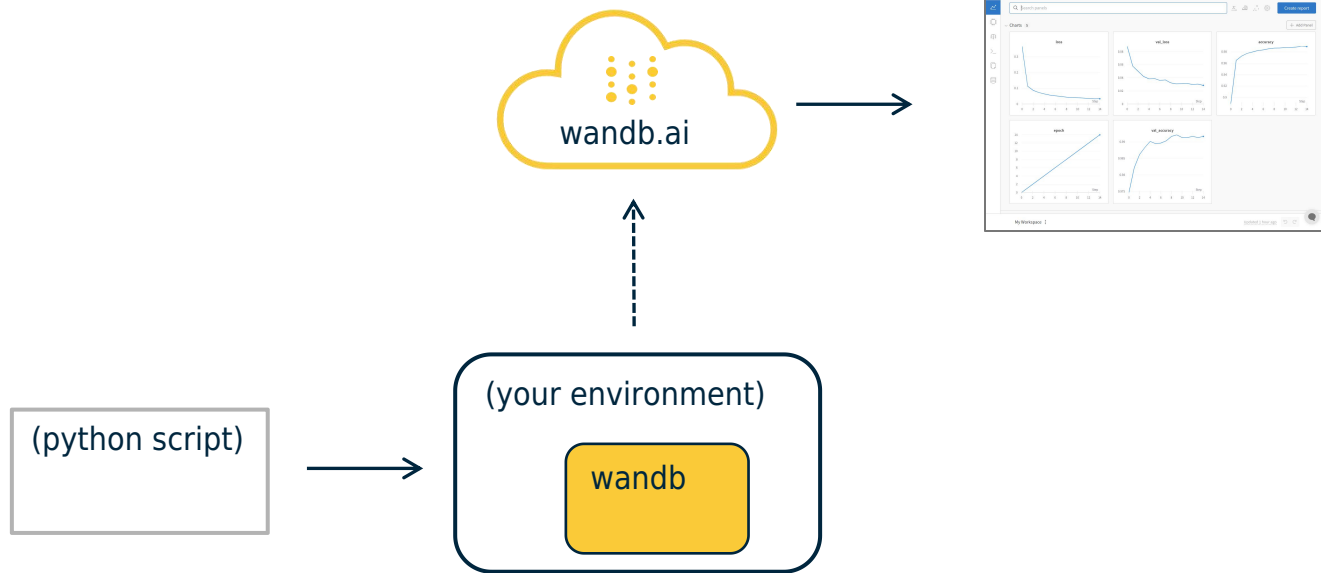
Weights & Biases background



W&B



How does W&B work?

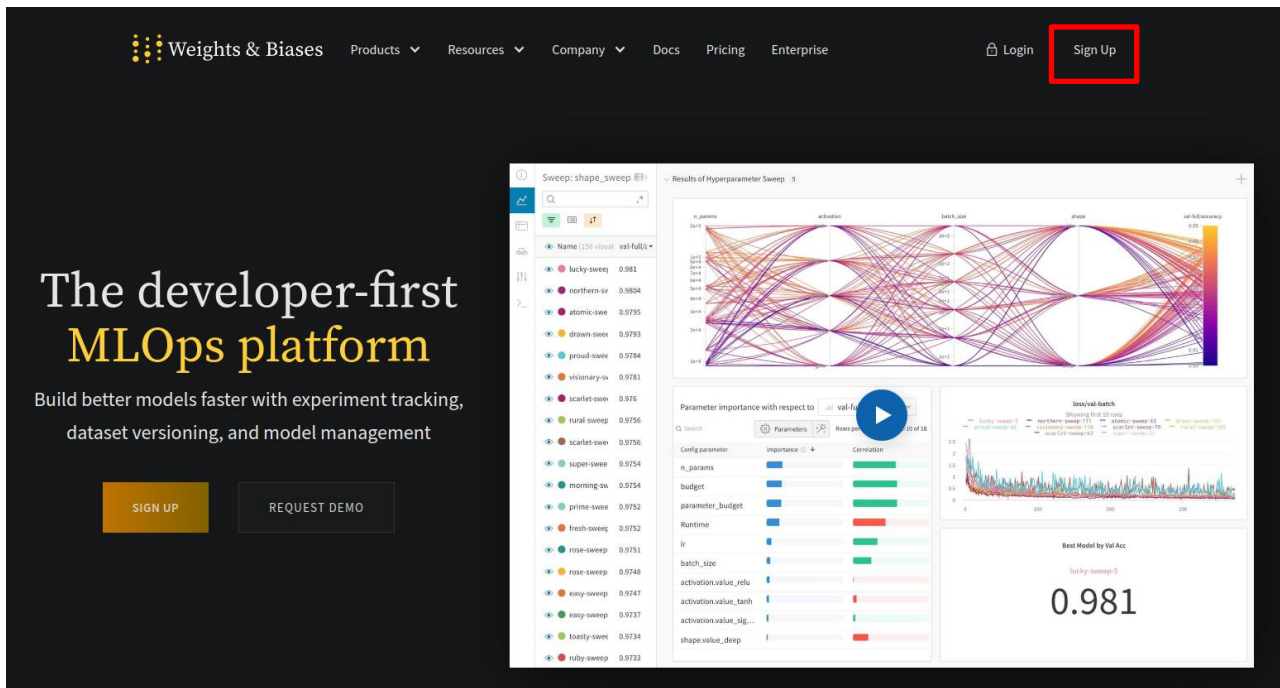


What is needed for W&B to run?

- 1) An **account** with W&B
- 2) The “**wandb**” package installed
- 3) A **.netrc file** (create by running “wandb login” command)
- 4) **wandb code** within your python script

Step 1: Create account with W&B

Navigate to “wandb.ai” on your browser and select “Sign Up”



The screenshot shows the Weights & Biases (W&B) website interface. The top navigation bar includes the W&B logo, links for Products, Resources, Company, Docs, Pricing, and Enterprise, and buttons for Login and Sign Up. The Sign Up button is highlighted with a red rectangular box. Below the navigation bar, the main content area features the text "The developer-first MLOps platform" in a large, bold font, followed by the subtitle "Build better models faster with experiment tracking, dataset versioning, and model management". Below this text are two buttons: "SIGN UP" and "REQUEST DEMO". On the right side of the main content area, there is a preview of the W&B dashboard interface, which displays a list of experiments, a parameter importance chart, and a validation accuracy plot.

The developer-first MLOps platform

Build better models faster with experiment tracking, dataset versioning, and model management

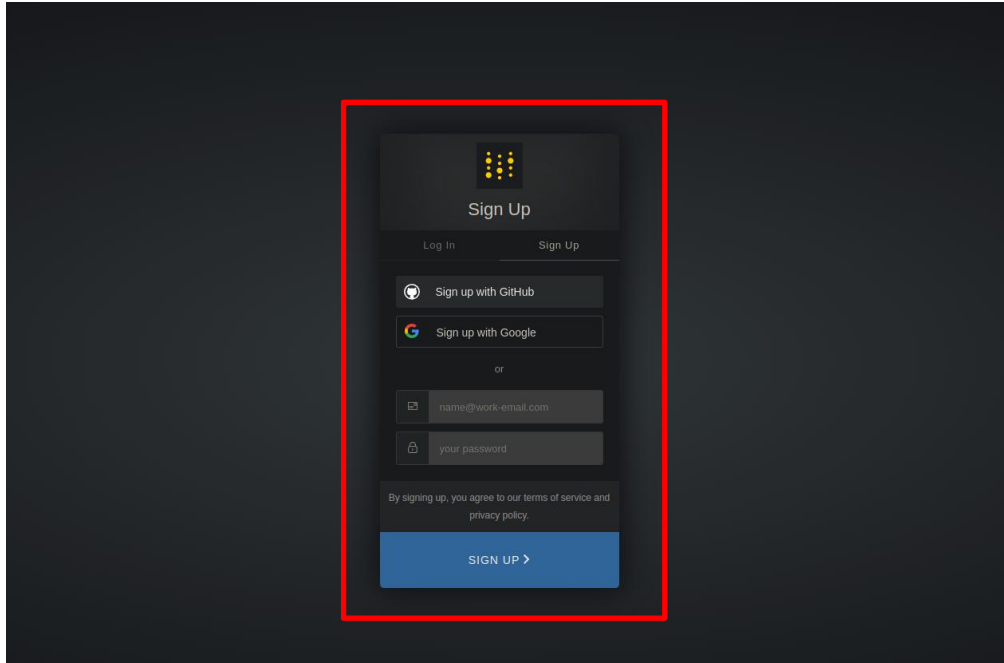
[SIGN UP](#) [REQUEST DEMO](#)

Dashboard Preview:

- Experiments List:** A table showing various experiments with columns for Name, Visual, and Val-Full.
- Parameter Importance:** A chart showing the importance of different parameters (n_params, budget, parameter_budget, Runtime, lr, batch_size, activation_value_relu, activation_value_tanh, activation_value_sigmoid, shape_value_deep) with respect to the validation accuracy.
- Validation Accuracy Plot:** A line graph showing the validation accuracy over time for different experiments.
- Best Model by Val Acc:** A section highlighting the best model, "lucky-sweep-0", with a validation accuracy of 0.981.

Step 1: Create account with W&B

Follow steps to register



Step 1: Create account with W&B

After registering, click on the W&B icon on the top left. You'll notice and API key. This will be used in step 3

1

Home

Applications

Model Registry →

Projects

+ Create new project

Profile

runai-jcosme

Teams

+ Create new team

Get Started with Weights & Biases

Add W&B's lightweight integration to your existing ML code and quickly get live metrics, t
streamed to the project workspace.

- 1 Integrate W&B in your own ML Code with the [Quickstart guide](#).
[Go To Quickstart](#)
- 2 **Authorize with your API key.**
804a212a9...
- 3 See live metrics, terminal logs, and system stats in your project.

Step 2: Install “wandb” package

We can install wandb with conda.

Here is an example command where we create an environment called “wandb-env” and install the “wandb” package along with Tensorflow

```
mamba create -n wandb-env -c conda-forge wandb tensorflow tensorflow-gpu -y
```


Step 3: .netrc file

The .netrc file is created after running “wandb login” command.

The .netrc file is always **created** in the **home** directory (**~/.netrc**)

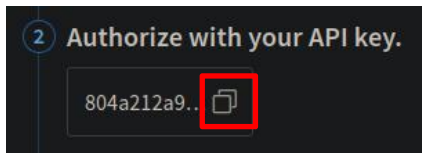
When running wandb within your code, it will always **look** for the .netrc file in the home directory (**~/.netrc**)

Step 3: .netrc file

Run the “wandb login” command

wandb login

Copy the API key from your W&B account page...



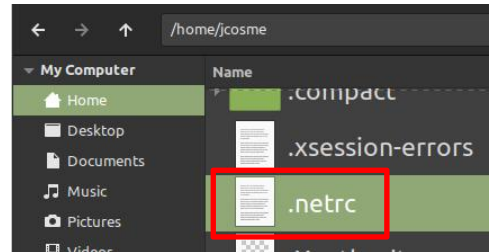
...then paste it into the terminal (and hit enter)

```
File Edit View Search Terminal Help
jcosme@jane:~$ conda activate wandb-env
(wandb-env) jcosme@jane:~$ wandb login
wandb: Logging into wandb.ai. (Learn how to deploy a W&B server locally: https://wandb.me/wandb-server)
wandb: You can find your API key in your browser here: https://wandb.ai/authorize
wandb: Paste an API key from your profile and hit enter, or press ctrl+c to quit: 
```

Step 3: .netrc file

A file will be created in your home directory (~/) called ".netrc" (~/.netrc)

```
jcosme@jane:~$ conda activate wandb-env
(wandb-env) jcosme@jane:~$ wandb login
wandb: Logging into wandb.ai. (Learn how to deploy a W&B server locally: https://wandb.ai/wandb-server)
wandb: You can find your API key in your browser here: https://wandb.ai/authorize
wandb: Paste an API key from your profile and hit enter, or press ctrl+c to quit:
wandb: Appending key for api.wandb.ai to your netrc file: /home/jcosme/.netrc
(wandb-env) jcosme@jane:~$
```



The .netrc file is a text file containing your API key

```
.netrc
1 machine api.wandb.ai
2   login user
3   password 804a212a9
4
```

W&B references this file in order to interface with your wandb.ai account.
You will no longer need to run the "wandb login" command.

Step 4: wandb code in your script

Here are the highlights of the wandb code within our python script

```
import wandb
from wandb.keras import WandbCallback
from tensorflow import keras
from tensorflow.keras import layers

wandb.init(project="wandb-local")

wandb.config = {
    "learning_rate": 0.001,
    "epochs": 15,
    "batch_size": 128
}

# (load data here)
# (build model)

# (continued on next block)...
```

```
# ... (continued from previous block)

opt = keras.optimizers.Adam(
    learning_rate=wandb.config['learning_rate']
)

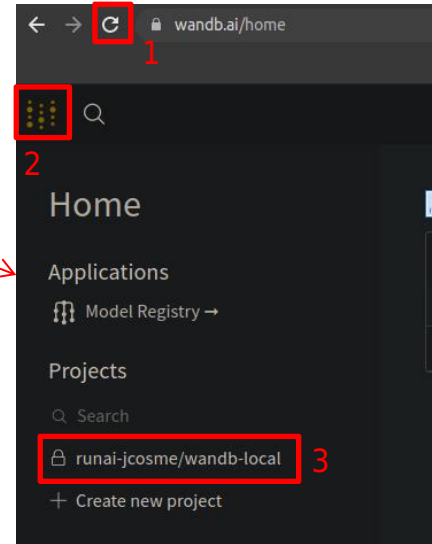
model.compile(
    loss="categorical_crossentropy",
    optimizer=opt,
    metrics=["accuracy"]
)

model.fit(
    x_train,
    y_train,
    validation_split=0.1,
    batch_size=wandb.config['batch_size'],
    epochs=wandb.config['epochs'],
    callbacks=[WandbCallback()],
)
```

For more info on how to use wandb code in your scripts, visit docs.wandb.ai

Step 4: wandb code in your script

After the script finishes running, refresh your web browser, select the W&B icon on the top left, and you will be able to see the project

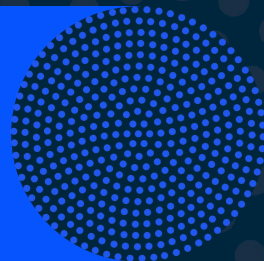


Setting up
W&B with **run:ai**



W&B + **run:**

ai



What is needed for **run:ai** to work with W&B?

- 1) An **account** with W&B
- 2) A **docker image** with **wandb** installed
- 3) A **persistent .netrc** file
- 4) **wandb code** within your python script

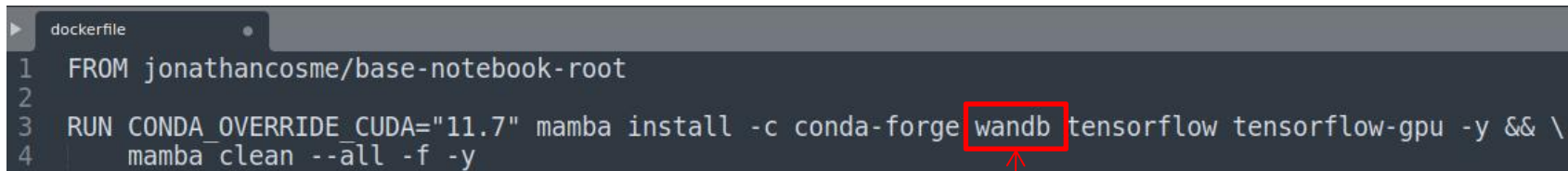
Step 2: Docker image with wandb installed

The docker image we will be using is “**jonathancosme/base-root-wandb-tf**”*

It is publicly available here:

hub.docker.com/r/jonathancosme/base-root-wandb-tf

This is the dockerfile



```
1 FROM jonathancosme/base-notebook-root
2
3 RUN CONDA_OVERRIDE_CUDA="11.7" mamba install -c conda-forge wandb tensorflow tensorflow-gpu -y && \
4     mamba clean --all -f -y
```

This is where we the installation of wandb is specified

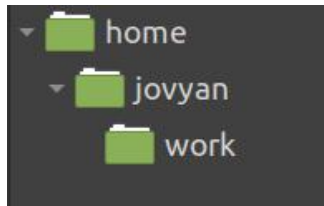
*This image uses “jonathancosme/base-notebook-root” as a base image, which is a slight modification of the official “jupyter/base-notebook” image. More information can be found here: github.com/jonathancosme/jupyter-base-notebook-root

Step 3: Persistent netrc file

about wandb login command:

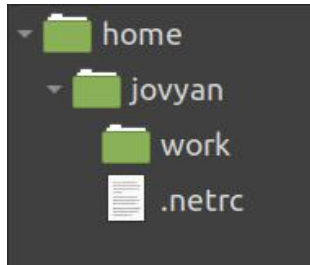
The home directory in our image is
/home/jovyan

(before we run wandb login)



Therefore, the .netrc file will be created as
/home/jovyan/.netrc

(after we run wandb login)

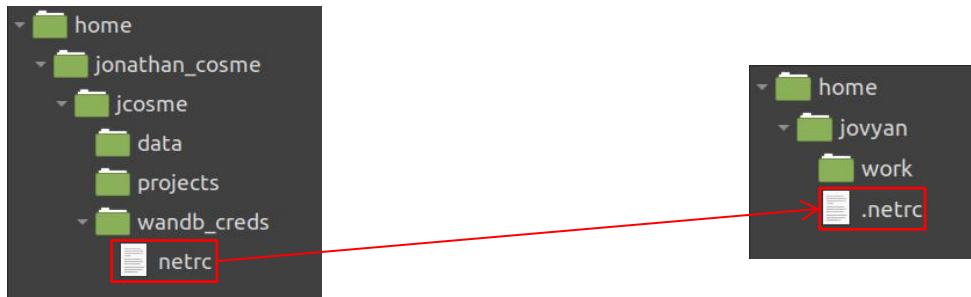
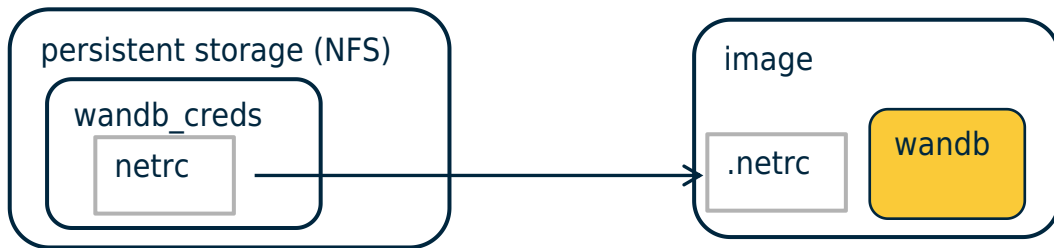


wandb will always look for that file in that location

Step 3: Persistent netrc file

we want to **create a persistent copy of the netrc file** on our NFS...

...so that we can mount that file in our image, when we run jobs

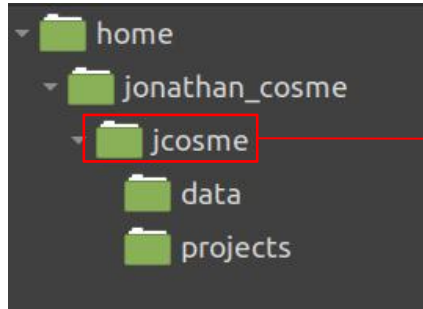


Step 3: Persistent netrc file

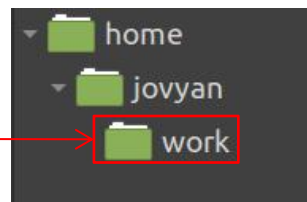
how do we create a persistent netrc file?

1. mount our NFS to the jupyter work directory

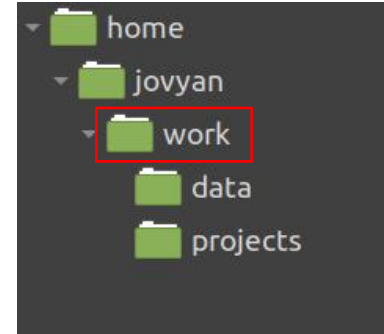
(NFS)



(jupyter)



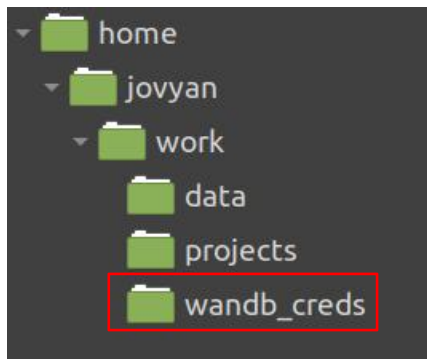
=



Step 3: Persistent netrc file

how do we create a persistent netrc file?

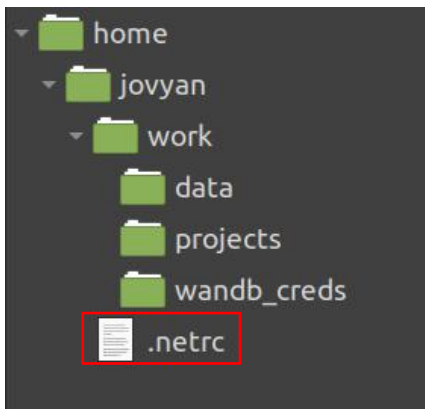
2. Create the wandb_creds folder in the mounted directory



Step 3: Persistent netrc file

how do we create a persistent netrc file?

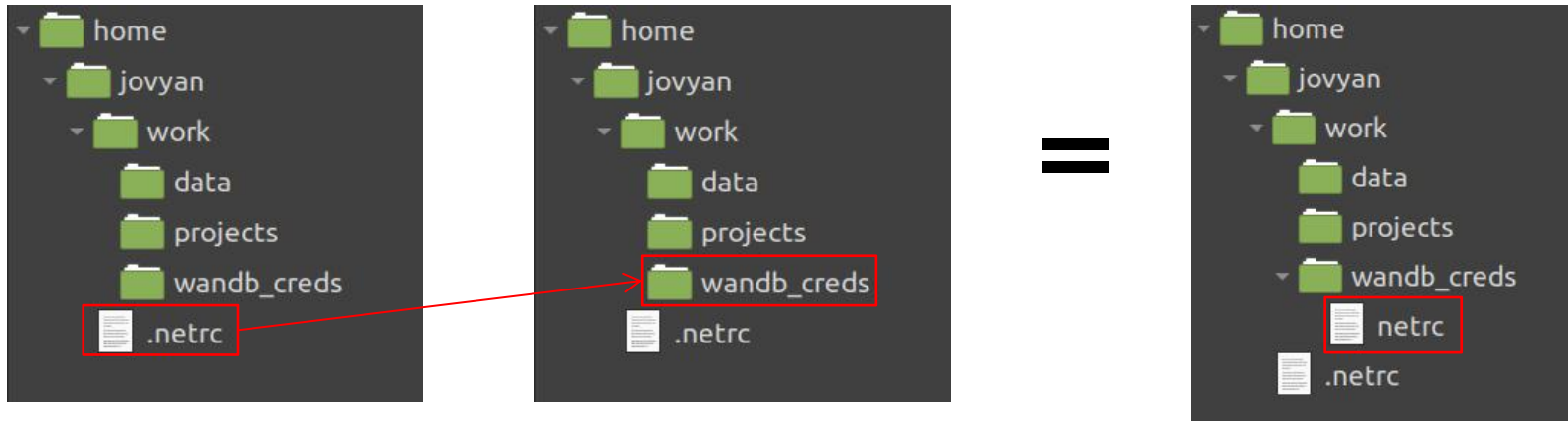
3. Create the .netrc file (with wandb login command)



Step 3: Persistent netrc file

how do we create a persistent netrc file?

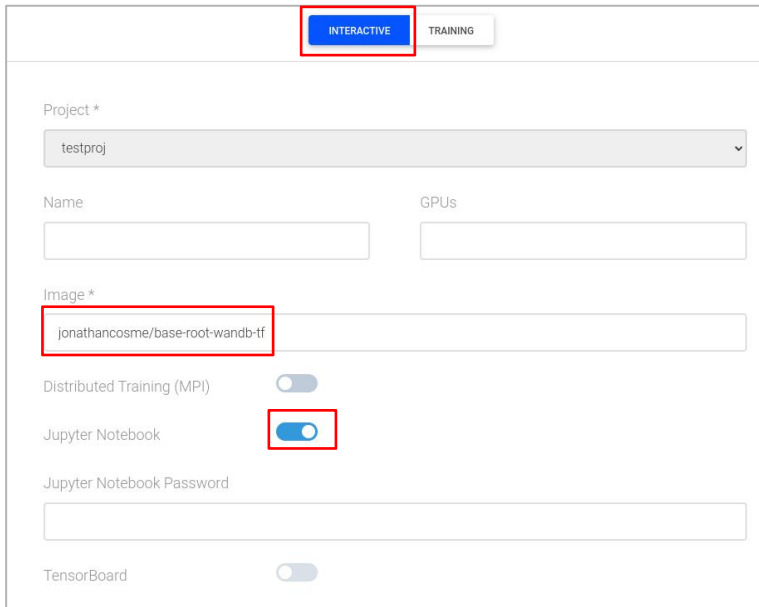
3. Copy .netrc file to created wandb_creds folder



Step 3: Persistent .netrc file

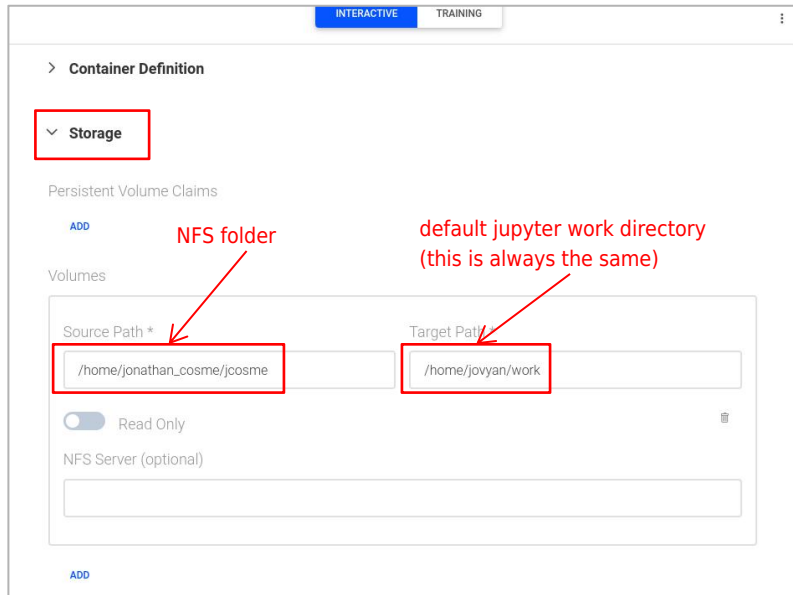
First, we'll need to mount our NFS directory into the /home/jovyan/work directory of our image

On the UI, create a new interactive job with the jonathancosme/base-root-wandb-tf image...



The screenshot shows the 'INTERACTIVE' tab selected. The 'Project' dropdown is set to 'testproj'. The 'Image' field contains 'jonathancosme/base-root-wandb-tf'. The 'Jupyter Notebook' toggle is turned on. The 'TensorBoard' toggle is turned off.

...and mount our NFS to the default work directory



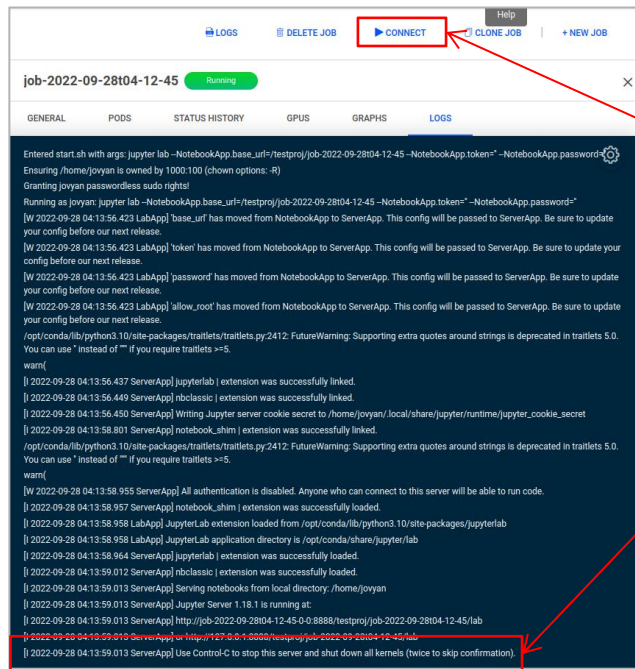
The screenshot shows the 'Storage' section under 'Container Definition'. A volume is configured with 'Source Path' set to '/home/jonathan_cosme/cosme' and 'Target Path' set to '/home/jovyan/work'. Red arrows point from the text 'NFS folder' to the source path and 'default jupyter work directory (this is always the same)' to the target path. The 'Read Only' toggle is turned off, and the 'NFS Server (optional)' field is empty.

Step 3: Persistent .netrc file

1. Even though the job says “running,” seeing this line indicates that the notebook is still starting up

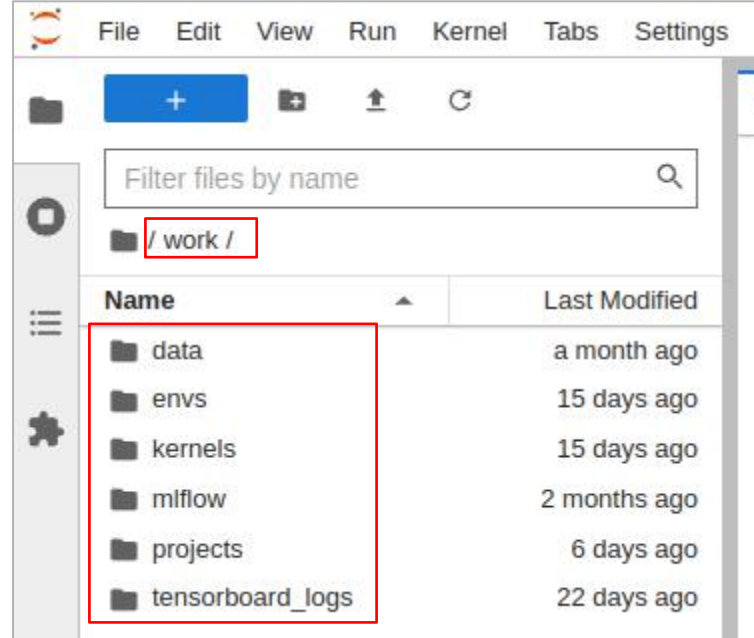


2. Wait until you see a line that says “Use Control-C to stop...” before clicking the connect button



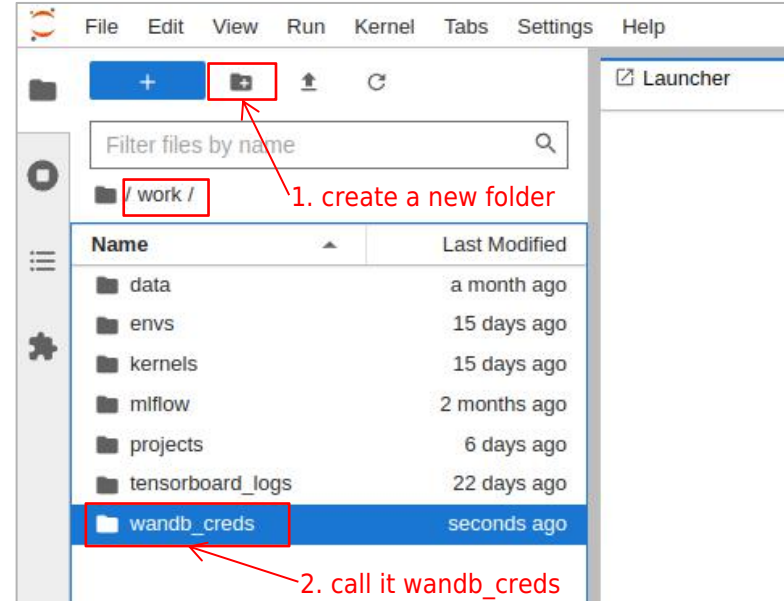
Step 3: Persistent .netrc file

After connecting, we should see all our files and folders within the work directory



Step 3: Persistent .netrc file

Within the work directory, create a new folder called wandb_creds



Step 3: Persistent .netrc file

run “wandb login” command to make the .netrc file

The diagram illustrates the process of creating a persistent .netrc file using the wandb login command. It consists of several interconnected components:

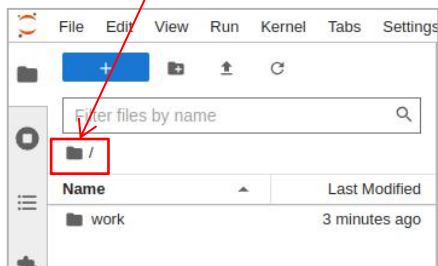
- Terminal Window Selection:** A window titled "Other" contains a "Terminal" icon. A red box highlights this icon, with an arrow pointing to it labeled "1. open a terminal window".
- wandb login Command:** A dark button labeled "wandb login" has an arrow pointing to it labeled "2. run command".
- API Key Authorization:** A dark box titled "2 Authorize with your API key." shows a partial API key "804a212a9..." and a copy icon. A red box highlights the copy icon, with an arrow pointing to it labeled "3. copy wandb API key".
- Terminal Execution:** A terminal window shows the command "wandb login" being executed. The output includes instructions to paste the API key. A red box highlights the prompt "Paste an API key from your profile and hit enter, or press ctrl+c to quit:", with an arrow pointing to it labeled "4. paste API key into terminal when prompted".
- Exit Command:** A dark button labeled "exit" has an arrow pointing to it labeled "5. exit the terminal".

```
(base) jovyana@job-2022-09-28t04-12-45-0-0:~/works wandb login
wandb: Logging into wandb.ai. (Learn how to deploy a W&B server locally: https://wandb.me/wandb-server)
wandb: You can find your API key in your browser here: https://wandb.ai/authorize
wandb: Paste an API key from your profile and hit enter, or press ctrl+c to quit:
wandb: Appending key for api.wandb.ai to your netrc file: /home/jovyana/.netrc
(base) jovyana@job-2022-09-28t04-12-45-0-0:~/works exit
```

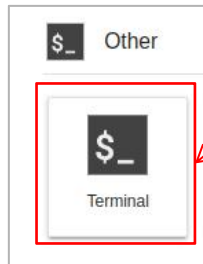
Step 3: Persistent .netrc file

confirm .netrc file was created

1. return to home directory



2. open a new terminal window



3. run command

ls -a



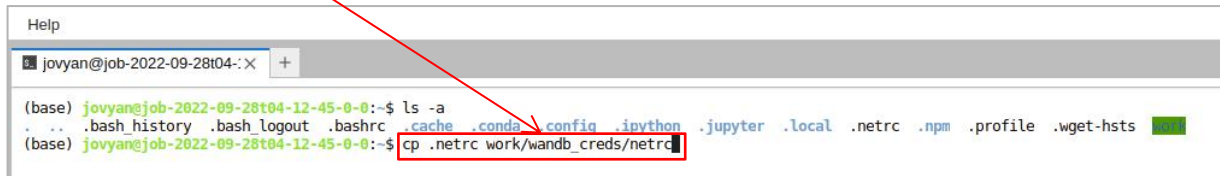
4. confirm .netrc file exists

Step 3: Persistent .netrc file

copy the .netrc file to the persistent wandb_creds folder.
(We will not put a period in front of the file name so that it will be visible to us)*

In the same terminal, run command

```
cp .netrc work/wandb_creds/netrc
```

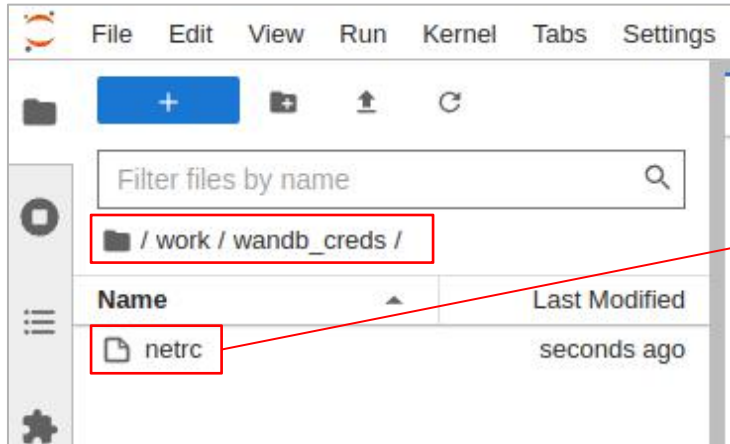


A terminal window screenshot showing the command `cp .netrc work/wandb_creds/netrc` being executed. The terminal title is `jovyan@job-2022-09-28t04-: X`. The prompt is `(base) jovyan@job-2022-09-28t04-12-45-0-0:~$`. The command `ls -a` has been run, showing a list of files including `.bash_history`, `.bash_logout`, `.bashrc`, `.cache`, `.conda`, `.config`, `.ipython`, `.jupyter`, `.local`, `.netrc`, `.npm`, `.profile`, `.wget-hsts`, and `.zshrc`. The command `cp .netrc work/wandb_creds/netrc` is being entered and is highlighted with a red box. A red arrow points from the text "In the same terminal, run command" to the command input line.

*The jupyter lab UI does not show hidden files (files preceeded by a "."). We copy it as a visible file (named ".netrc" instead of ".netrc") so that we do not mistake the wandb_creds folder for an empty folder, in the future.

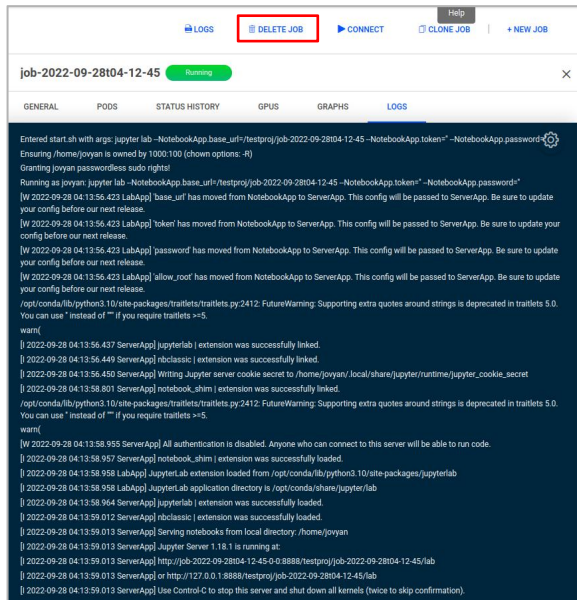
Step 3: Persistent .netrc file

confirm the copied file exists in our wandb_creds folder



Step 3: Persistent .netrc file

Now we can delete the job



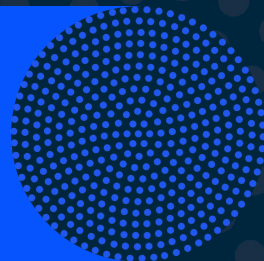
The screenshot shows the RunAI web interface. At the top, there are navigation links: LOGS, DELETE JOB (highlighted with a red box), CONNECT, CLONE JOB, and NEW JOB. Below this is a job card for 'job-2022-09-28t04-12-45' with a green 'Running' status indicator. The job card has tabs for GENERAL, PODS, STATUS HISTORY, GPU, GRAPHS, and LOGS. The LOGS tab is selected, displaying a log of system messages and commands. The log includes messages about ensuring /home/jovyan is owned by 1000:100, granting jovyan passwordless sudo rights, and running as jovyan. It also shows the movement of various configuration files (base_url, token, password, allow_root) from NotebookApp to ServerApp. A warning message from traitlets.py is visible, followed by a successful link for the JupyterLab extension. The log concludes with the Jupyter Server 1.18.1 running at the specified URL.

Running
W&B with **run:ai**



W&B + **run:**

ai



Running wandb jobs with **run:ai**

From now on, it is VERY important that **we mount the persistent netrc file** in our NFS, to the expected location of the **netrc file in our image**
in our case, the persistent netrc file is at:
/home/jonathan_cosme/jcosme/wandb_creds/netrc
and it should be mounted to:
/home/jovyan/.netrc

Use the jonathancosme/base-root-wandb-tf image, and mount our NFS directory to the work directory as usual

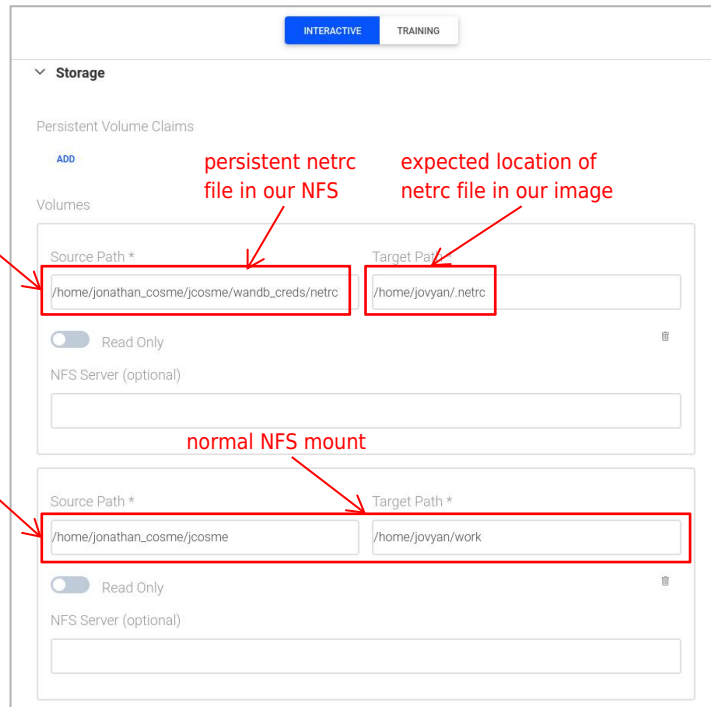


Image *

jonathancosme/base-root-wandb-tf

Distributed Training (MPI) ☐

Jupyter Notebook ☒



INTERACTIVE TRAINING

Storage

Persistent Volume Claims

ADD

Volumes

Source Path * Target Path *

/home/jonathan_cosme/jcosme/wandb_creds/netrc /home/jovyan/.netrc

☐ Read Only

NFS Server (optional)

normal NFS mount

Source Path * Target Path *

/home/jonathan_cosme/jcosme /home/jovyan/work

☐ Read Only

NFS Server (optional)

Jupyter notebook example

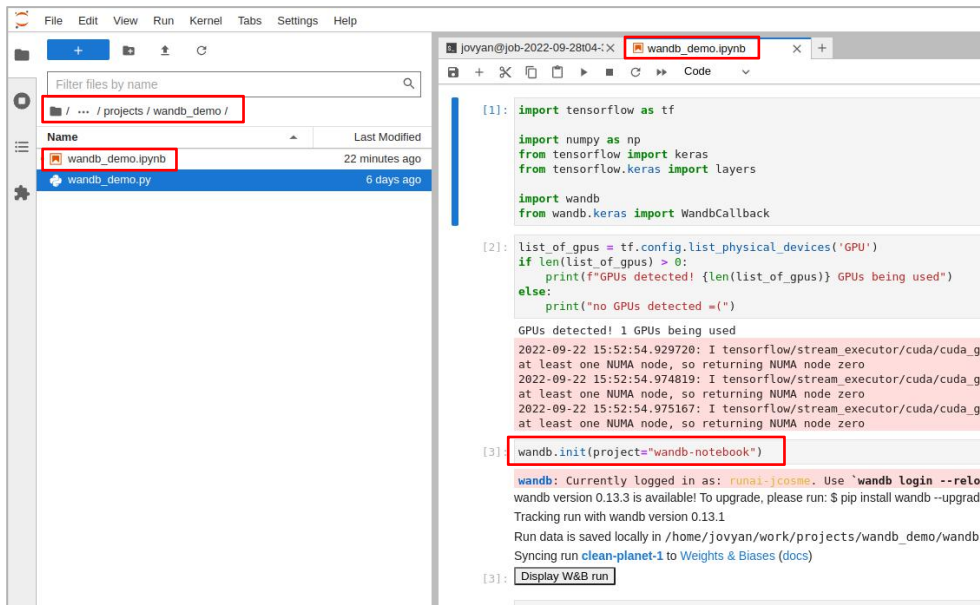
within our directory:

/home/jovyan/work/projects/wandb_demo

we have a **wandb_demo.ipynb** notebook.

Within the notebook, we create a wandb project called

wandb-notebook



```
File Edit View Run Kernel Tabs Settings Help
+ [wandb_demo.ipynb]
Filter files by name
/ ... / projects / wandb_demo /
Name Last Modified
wandb_demo.ipynb 22 minutes ago
wandb_demo.py 6 days ago

[1]: import tensorflow as tf
import numpy as np
from tensorflow import keras
from tensorflow.keras import layers

import wandb
from wandb.keras import WandbCallback

[2]: list of gpus = tf.config.list_physical_devices('GPU')
if len(list of gpus) > 0:
    print(f"GPUs detected! {len(list_of_gpus)} GPUs being used")
else:
    print("no GPUs detected =(")

GPUs detected! 1 GPUs being used
2022-09-22 15:52:54.929720: I tensorflow/stream_executor/cuda/cuda_g
at least one NUMA node, so returning NUMA node zero
2022-09-22 15:52:54.974819: I tensorflow/stream_executor/cuda/cuda_g
at least one NUMA node, so returning NUMA node zero
2022-09-22 15:52:54.975167: I tensorflow/stream_executor/cuda/cuda_g
at least one NUMA node, so returning NUMA node zero

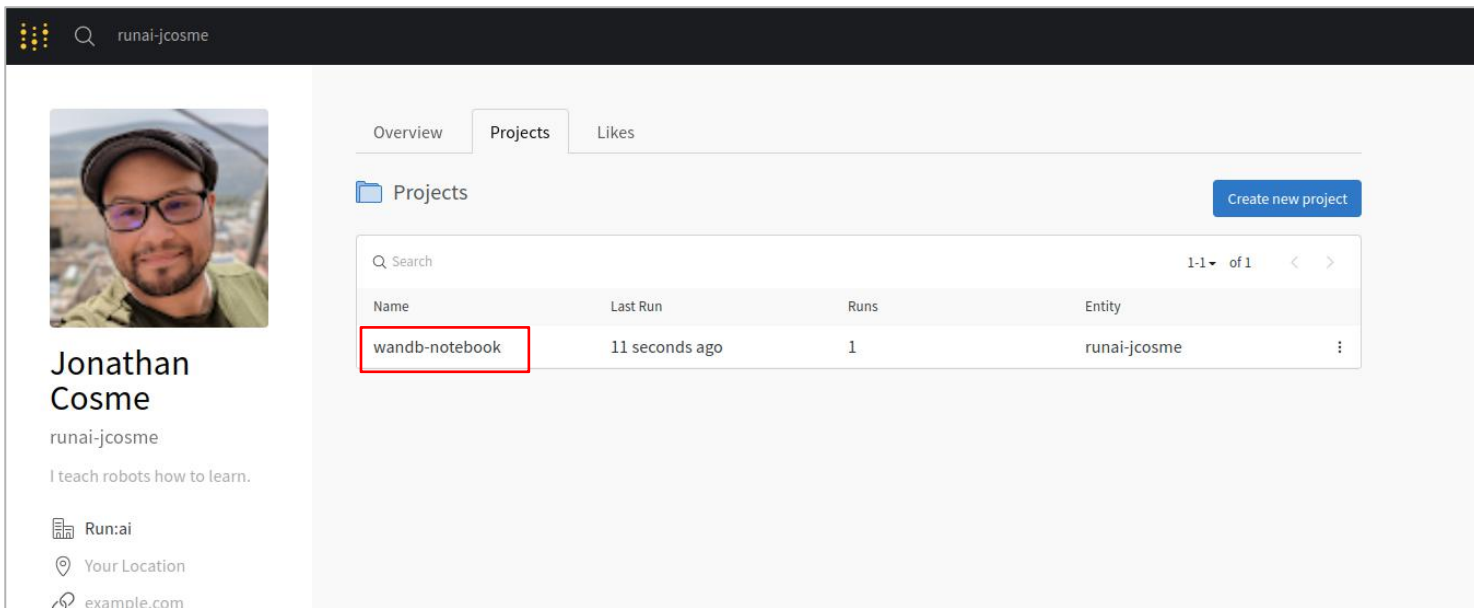
[3]: wandb.init(project="wandb-notebook")

wandb: Currently logged in as: runai-jcosme. Use `wandb login --relo
wandb version 0.13.3 is available! To upgrade, please run: $ pip install wandb --upgrad
Tracking run with wandb version 0.13.1
Run data is saved locally in /home/jovyan/work/projects/wandb_demo/wandb
Syncing run clean-planet-1 to Weights & Biases (docs)

[3]: Display W&B run
```

Jupyter notebook example

If we run the **wandb_demo.ipynb** notebook, we will see the updates on our wandb.ai account



The screenshot displays the wandb.ai interface for user Jonathan Cosme (runai-jcosme). The 'Projects' tab is active, showing a table of runs. The first run, 'wandb-notebook', is highlighted with a red box. The table shows it was run 11 seconds ago with 1 run by the entity 'runai-jcosme'.

Name	Last Run	Runs	Entity
wandb-notebook	11 seconds ago	1	runai-jcosme

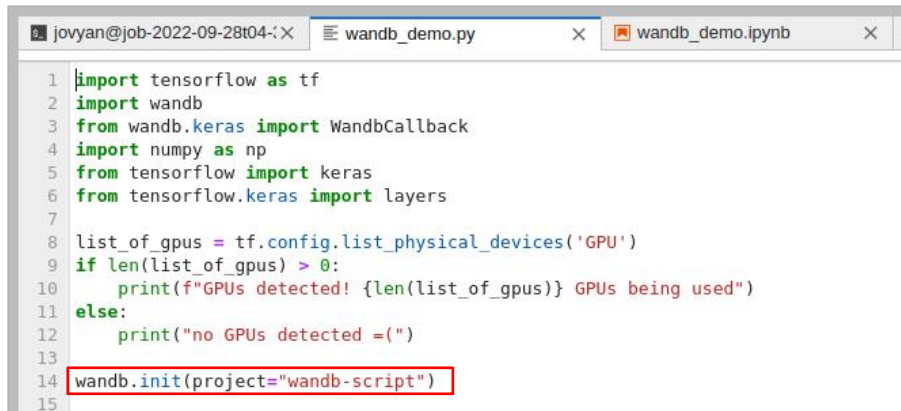
CLI example

within our directory:

/home/jovyan/work/projects/wandb_demo

we also have a **wandb_demo.py** python script.

Within the script, we create a wandb project called **wandb-script**



```
1 import tensorflow as tf
2 import wandb
3 from wandb.keras import WandbCallback
4 import numpy as np
5 from tensorflow import keras
6 from tensorflow.keras import layers
7
8 list_of_gpus = tf.config.list_physical_devices('GPU')
9 if len(list_of_gpus) > 0:
10     print(f"GPUs detected! {len(list_of_gpus)} GPUs being used")
11 else:
12     print("no GPUs detected =(")
13
14 wandb.init(project="wandb-script")
15
```

CLI example

we can run **wandb_demo.py** via the CLI with this command

```
runai submit \  
  --project testproj \  
  --gpu 1 \  
  --job-name-prefix wandb-demo \  
  --image jonathancosme/base-root-wandb-tf \  
  --volume /home/jonathan_cosme/jcosme:/home/jovyan/work \  
  --volume /home/jonathan_cosme/jcosme/wandb_creds/netrc:/home/jovyan/.netrc \  
  -- conda run -n base python work/projects/wandb_demo/wandb_demo.py
```

here are the key highlights

use image with wandb installed

mount NFS to work directory

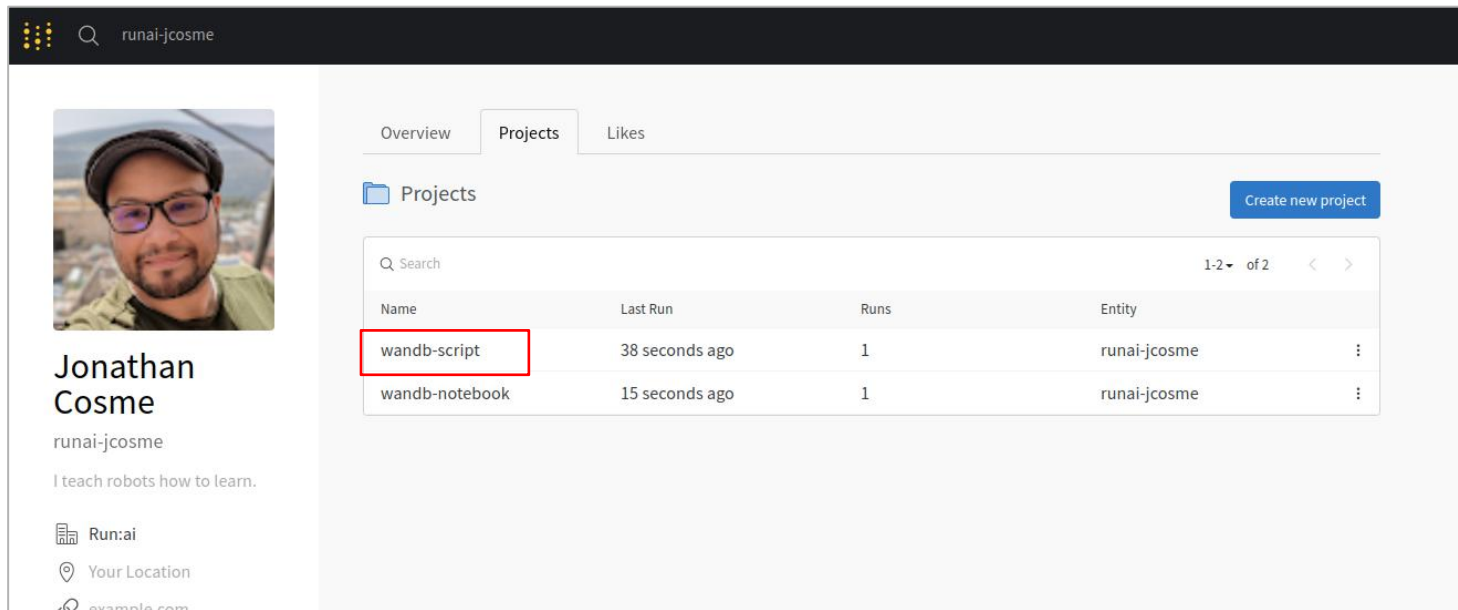
mount netrc file

command to run wandb_demo.py script

```
runai submit \  
  --project testproj \  
  --gpu 1 \  
  --job-name-prefix wandb-demo \  
  --image jonathancosme/base-root-wandb-tf \  
  --volume /home/jonathan cosme/jcosme:/home/jovyan/work \  
  --volume /home/jonathan cosme/jcosme/wandb_creds/netrc:/home/jovyan/.netrc \  
  -- conda run -n base python work/projects/wandb_demo/wandb_demo.py
```

CLI example

After submitting the CLI command, we will see the updates on our wandb.ai account



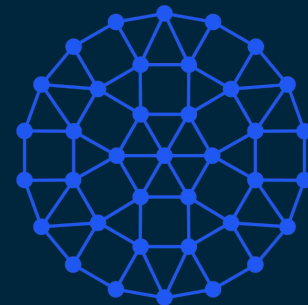
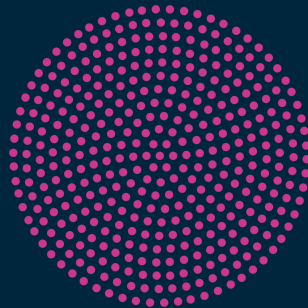
The screenshot shows the wandb.ai profile page for Jonathan Cosme. The profile includes a photo, name, and bio. The 'Projects' tab is selected, showing a table of projects. The 'wandb-script' project is highlighted with a red box.

Jonathan Cosme
runai-jcosme
I teach robots how to learn.

Projects

Name	Last Run	Runs	Entity
wandb-script	38 seconds ago	1	runai-jcosme
wandb-notebook	15 seconds ago	1	runai-jcosme

Thank you!



info@run.ai | www.run.ai

contact@wandb.ai | www.wandb.ai