CSCE 479/879 Environment Setup

Mrinal Rawool

January 8, 2024

1 Before You Begin...

1. Create an `https://swan.unl.edu` account.
2. Have DUO authentication enabled for HCC/Swan account
3. Have access to the GitHub course repository

2 Steps

1. Download the shell script from environment creation from the GitHub repository folder
2. Log on to Swanś OpenDemand Page
3. On the main menu at the top, choose *Files >Home Directory* and upload the shell script at this location.
4. On the main menu at the top, choose *Clusters >Swan Shell Access* and navigate to the location of the environment creation shell script. Alternatively, you can *ssh* into Swan from your local machine as well.
5. Run the script using the command

```
$ ./create_env.sh
```

3 Confirm Successful Set up

1. Check if you see the newly created Conda environment in your home directory by running the command

```
$ conda env list
```

   You may need to optionally load anaconda by running

```
$ module load anaconda
```

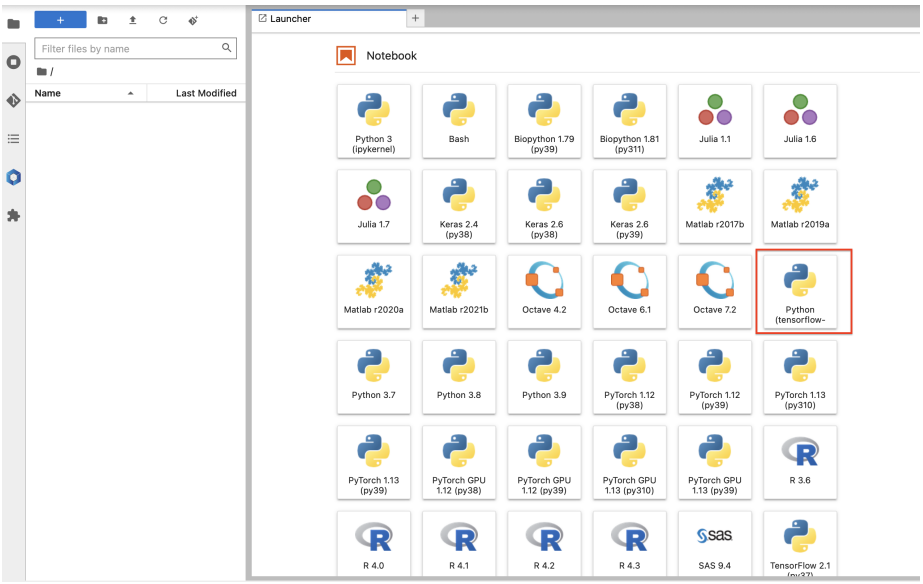

Figure 1: Selecting the correct kernel

2. On the Swan OnDemand Page top menu, choose *Interactive apps* > *Jupyter Lab*. This will open a web form for reserving a Jupyter Lab session. For now, you can use the defaults and scroll down to confirm the selection. Your session will be queued and you will be notified when the session is ready. Click the *Connect to Jupyter Lab* button when it appears. You now have the session reserved for one hour.

3. Create a new Jupyter Notebook and click on the *choose kernel* option on the top right corner of the window.

4. You will see a list of available kernels. If you were successfully able to run 5, you should be able to see the **tensorflow_env** kernel. See figure 1

5. If you're unable to see your kernel, then run the command

```
$ python -m ipykernel install \
--user --name "$CONDA_DEFAULT_ENV" \
--display-name "Python ($CONDA_DEFAULT_ENV)"
```

   from your home directory. To confirm if the command was successful, repeat steps 3 and 4

6. Finally, check if you are able to import installed packages successfully. See fig 2

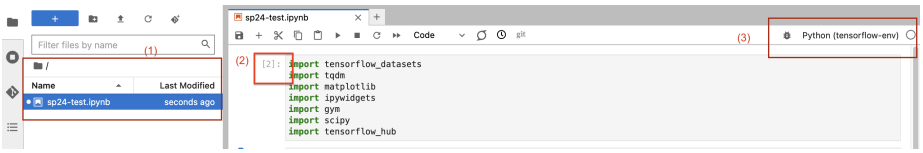

Figure 2: Importing installed packages in the Conda env successfully

4 Explaining the script

This section explains what the script is doing. This will come in handy in case you encounter errors.

1. These commands load existing GPU enabled environment containing the required Tensorflow and Python versions required for this course.

```
module load tensorflow-gpu/py39/2.6
module load anaconda
```

2. This command clones the existing environment and creates a new Conda environment called *tensorflow-env* inside your *Swan* account. This step might take a while to finish so please be patient.

```
conda create -n tensorflow-env \
             --clone $CONDA_DEFAULT_ENV
```

3. Once the envs are cloned, you no longer need the pre-existing Conda env so it needs to be unloaded. *Note: we have only unloaded the Conda env, the anaconda module is still active since we need to run commands to activate the new environment and install some packages*

```
module unload tensorflow-gpu/py39/2.6
```

4. At this stage, if you are running the script line by line, you can check where your *tensorflow-env* has been created using the command

```
conda env list
```

5. Let's activate the new conda environment so that we can install packages

```
conda activate tensorflow-env
```

   If you encounter *CommandNotFound* error similar to the one shown in figure 3, run the command

```
conda init bash
```

   You may need to restart the shell for the changes to be applied as seen in figure 4

```
[mrawool@login1.swan mrawool]$ conda activate tensorflow-env

CommandNotFoundError: Your shell has not been properly configured to use 'conda activate'.
To initialize your shell, run

    $ conda init <SHELL_NAME>

Currently supported shells are:
  - bash
  - fish
  - tcsh
  - xonsh
  - zsh
  - powershell

See 'conda init --help' for more information and options.

IMPORTANT: You may need to close and restart your shell after running 'conda init'.
```

Figure 3: CommandNotFound error

```
IMPORTANT: You may need to close and restart your shell after running 'conda init'.


[mrawool@login1.swan mrawool]$ conda init bash
no change     /util/opt/anaconda/4.9.2/condabin/conda
no change     /util/opt/anaconda/4.9.2/bin/conda
no change     /util/opt/anaconda/4.9.2/bin/conda-env
no change     /util/opt/anaconda/4.9.2/bin/activate
no change     /util/opt/anaconda/4.9.2/bin/deactivate
no change     /util/opt/anaconda/4.9.2/etc/profile.d/conda.sh
no change     /util/opt/anaconda/4.9.2/etc/fish/conf.d/conda.fish
no change     /util/opt/anaconda/4.9.2/shell/condabin/Conda.psm1
no change     /util/opt/anaconda/4.9.2/shell/condabin/conda-hook.ps1
no change     /util/opt/anaconda/4.9.2/lib/python3.8/site-packages/xontrib/conda.xsh
no change     /util/opt/anaconda/4.9.2/etc/profile.d/conda.csh
modified      /home/scott/mrawool/.bashrc

==> For changes to take effect, close and re-open your current shell. <==
```

Figure 4: Initializing shell

6. You can use *pip* as well as *conda* to install and/or upgrade packages. Here, we use *pip* to install some packages. This step might take a while to finish so please be patient.

```
pip install tensorflow-datasets \
            tqdm \
            matplotlib \
            ipywidgets \
            gym \
            scipy \
            tensorflow_hub
```

7. Jupyter provides an interactive Python environment. To set it up, we need to install *iPython kernel*. We also need to make our newly created Conda environment accessible in Jupyter Lab; the application we will be using for our coursework. The following command achieves these objectives.

```
python -m ipykernel install \
          --user --name "$CONDA_DEFAULT_ENV" \
          --display-name "Python ($CONDA_DEFAULT_ENV)"
```

Here's a link to a handy Conda cheat sheet provided by Anaconda.