

Using Jupyterhub

Signing Up

A JupyterHub admin will have to add you to the user list. Once your name has been included, you can login from the “sign in” page. Whatever password you give in the “sign in” box will be set as your account password.

A screenshot of the JupyterHub 'Sign in' page. The page has an orange header with the text 'Sign in'. Below the header, there are two input fields: 'Username:' with the value 'taylor' and 'Password:' with a masked password represented by dots. At the bottom of the form is an orange button labeled 'Sign in'.

Setting up a conda environment

The default python environment has only a few basic packages installed. It's recommended that you setup your own conda environment for any code you want to run. To enable conda for your account, go to the homepage for your jupyterhub and click “New” and under “Other”, select a “Terminal”.

Type “conda init” in the terminal. You should see the conda environment added to “.bashrc”. Exit out of the terminal (this can be done by simply typing “exit”) and re-open a new terminal.

```
jupyter-temp@cyberpower:~$ conda init
no change      /opt/tljh/user/condabin/conda
no change      /opt/tljh/user/bin/conda
no change      /opt/tljh/user/bin/conda-env
no change      /opt/tljh/user/bin/activate
no change      /opt/tljh/user/bin/deactivate
no change      /opt/tljh/user/etc/profile.d/conda.sh
no change      /opt/tljh/user/etc/fish/conf.d/conda.fish
no change      /opt/tljh/user/shell/condabin/Conda.ps1
no change      /opt/tljh/user/shell/condabin/conda-hook.ps1
no change      /opt/tljh/user/lib/python3.9/site-packages/xontrib/conda.xsh
no change      /opt/tljh/user/etc/profile.d/conda.csh
modified       /home/jupyter-temp/.bashrc

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

jupyter-temp@cyberpower:~$
```

You should now see a (base) conda environment when you open your terminal.

```
(base) jupyter-temp@cyberpower:~$
```

To create a custom environment for an individual project, use the command

```
conda create --name tensorflow
conda activate tensorflow
```

If you want to use the GPU with tensorflow, you only need to install tensorflow through conda with the command

```
conda install tensorflow
```

If you want to use tensorflow from a jupyter notebook

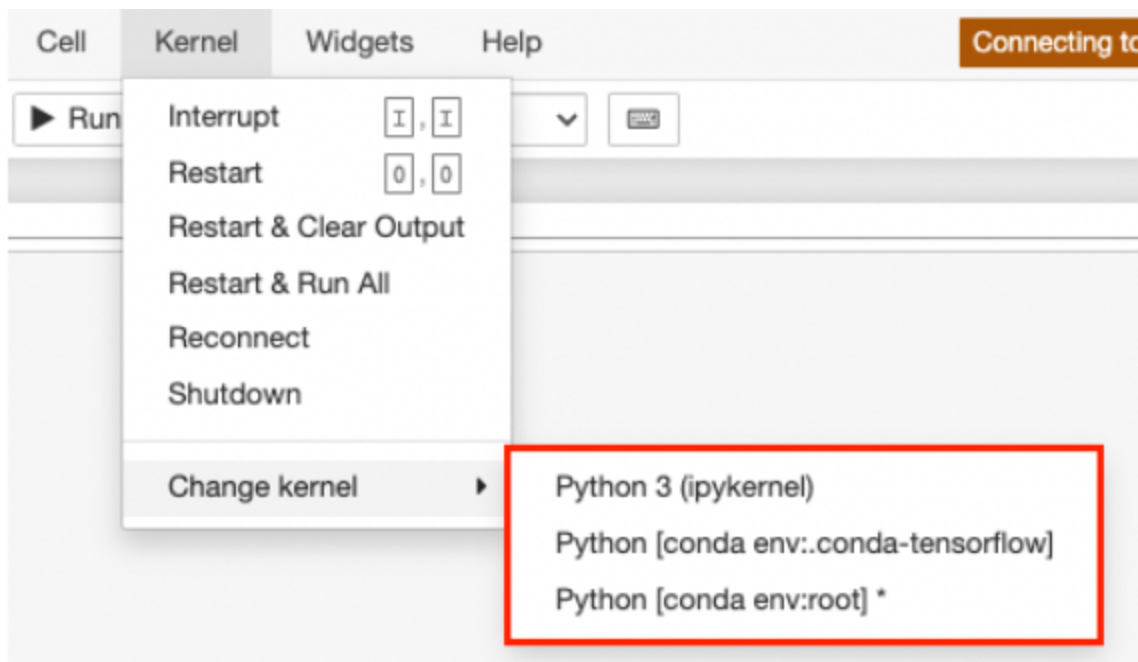
```
conda activate tensorflow
```

Adding your environment to jupyter notebooks

By default, your conda environment will only be available via the terminal. If you want to use these packages from a jupyter notebook you need to install the ipykernel package via conda. This can be done by activating the package you want to use and using conda install with ipykernel

```
conda activate tensorflow
conda install ipykernel
```

Now, when opening a Jupyter notebook, you should see your conda environment in the list of kernels. Note, you may need to restart your jupyter server for this change to take effect. This can be done by going to “Control Panel” at the top right, selecting “Stop My Server” and then “Start My Server”



In the case of tensorflow, you can check that your environment has access to the GPU with the commands

```
import tensorflow as tf  
gpu_available = tf.test.is_gpu_available()
```