# Useful commands

## **Start the Virtual Machine on the server.**

* Install the Thunder-Client extension on Visual Studio
* Start the Virtual Machine using Thunder-Client
  + HTPP Post Request by clicking on New Request button.
  + Copy and paste the URL into the address bar : <https://prod-29.australiasoutheast.logic.azure.com:443/workflows/3acf899c939d4485af4d9bba59249262/triggers/manual/paths/invoke?api-version=2016-10-01&sp=%2Ftriggers%2Fmanual%2Frun&sv=1.0&sig=AHJ-zHVooCYUB4NPyBtP0m5qDq6P6bVIcQG0q9yKTSs>
  + Edit the Body {"action":"start","vm":"the-diego"}
  + Click on the send button and wait for a response: “the-diego” started.

## **Start a SSH section.**

* Obtain the private key (SSH) from the administrator.
* Save the private key at: C:\Users\DiegoLozano\.ssh\id\_rsa
* Create a configuration file:

Host the-diego

HostName 20.53.242.136

User azureuser

IdentityFile C:\Users\DiegoLozano\.ssh\id\_rsa\the-diego\_key.pem

* Open the Remote Explorer VS extension.
* New remote
* Search and load for the configuration file
* Click on connect
* The remote explorer opens a new window
* Check the bottom-left conner SHH:the-diego
* The bottom of the new window must pull up to see the terminal interface

## **Ubuntu / Linux environment.**

ls - List files and folders: anaconda anaconda3 composetest testing torchradio

cd - Go to a directory, for example, cd torchradio

echo “$PWD” 🡪 /home/azureuser/torchradio

## **Install Pyhton distribution and libraries using Conda / Anaconda.**

* Install Anaconda (python3)
* Create an environment name “torchradio” using setup.cfg file from GitHub
* Install Black and Ruff Python libraries.
* conda activate torchradio

## **Black**

Black is the uncompromising Python code formatter.

Black . 🡪 Check and formats all python files in a directory.

## **Ruff: Code Linter**

Linters are programs that advise about code quality by displaying warnings and errors. They can detect your Python code mistakes, notice invalid code patterns and find elements that do not follow your conventions.

Ruff . 🡪 Check

ruff . --fix 🡪Fix all error

ruff path/to/code/to/check.py

ruff /home/azureuser/torchradio/examples/dsss.py --watch

ruff /home/azureuser/torchradio/examples/dsss.py --fix

ruff /home/azureuser/torchradio/torchradio/rx.py --fix

ruff /home/azureuser/torchradio/torchradio/sim.py --fix

## **Git hub**

ssh -T [git@github.com](mailto:git@github.com) 🡪 Testing the SSH connection

Make sure that you are in the right directory

* echo “$PWD” 🡪 /home/azureuser/torchradio

|  |  |  |  |
| --- | --- | --- | --- |
| Step | Command | Description | Example |
| 1 | Git status | It gives the information about the branches, new files and changes. | **git status**  On branch add-dsss  Your branch is up to date with 'origin/add-dsss'. |
| 2 | Git branch | List all branches (Local and Remote) and the pointed branch (\*). | **git branch**  DSSS  \* add-dsss  main  origin/main |
| 3 | Git checkout | Switch/point a branch | **git checkout DSSS** |
| 4 | Git add | Adds a change in the working directory to the staging area. | **git add torchradio/sim.py** |
| 5 | Git commit | Captures a snapshot of the project's currently staged changes. | **git commit -m “Rewrite current DSSS blocks using PyTorch”** |
| 6 | Git push | To upload local repository content to a remote repository | **git push origin add-dsss** |

Git commit saves repository changes on local but not remote repository. Contrarily, Git push then updates your git commit changes and sends it to remote repository.

git init - Create empty Git repo

git clone: git clone https://github.com/dilozcl/testing

git config user.name:

git remote update

git fetch

git checkout --track origin/<BRANCH-NAME>

git checkout DSSS

code sim.py

Commit change

Push the Local changes to Remote branch

Make sure the right environment

Run before committ

black . Formatting

ruff . Check, debugging

Python programming

Create class for