**Virtual Environments**

***Windows Users:*** *You will interact with conda by opening Anaconda Command Prompt that comes installed with the windows distribution of Anaconda.*

***Mac Users:*** *You will interact with conda directly through your terminal.*

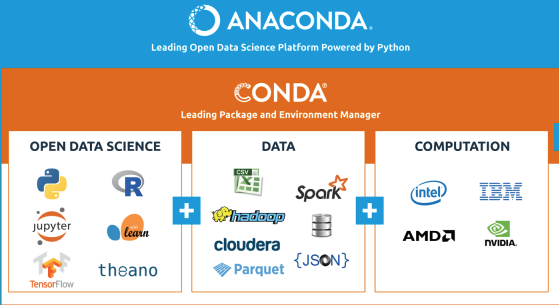
This guide aims to bring a newcomer up to speed with Anaconda and working with virtual environments. Please be sure to read through everything if you have never set up environments before. Also feel free to reference YouTube and google for setting up environments if you are still confused by the end of this guide.

In this bootcamp we are going to be working with many different libraries and as a best practice, when starting on a new project or even exploring new libraries/packages/algorithms, those libraries may have certain dependencies. If you’re installing all your new libraries under your main installation (also known as your ‘base’ installation) things can and will eventually stop working due to dependency issues or conflicts. This is normal.

Professional data scientists tend to always work in a virtual environment. Managing packages and environments are important and can be tedious so, we’ve gone ahead and for sake of simplicity chosen to use the Anaconda platform throughout the duration of this bootcamp. [Anaconda](https://www.anaconda.com/distribution/) is an open source data science platform which has various features. [Conda](https://docs.conda.io/projects/conda/en/latest/index.html) comes preinstalled with Anaconda and acts as your go to package and environment manager.

*You can call on Conda directly from* ***Anaconda Command Prompt in WindowsOS*** *or directly from the* ***terminal on MacOS****.*

*Please note Anacondas Command Prompt only comes installed with the Windows OS distribution of Anaconda*



**INSTRUCTIONS:**

**STEP 1: Install Anaconda**

1. Download and install your operating systems version of [Anaconda](https://www.anaconda.com/distribution/). Please be sure to download the version with **Python 3.7:** If the link above doesn’t work, go to <https://www.anaconda.com/distribution/> to download.
2. If you are not sure whether you want the full installation of Anaconda you could also install miniconda which is a lightweight version of Anaconda. I would suggest if you haven’t ever worked with it, just install the full Anaconda distribution. Here is an additional resource you can read through to understand the differences: <http://deeplearning.lipingyang.org/2018/12/23/anaconda-vs-miniconda-vs-virtualenv/>

Okay you should now have the software you need to get going. From this point forward this guide will show screenshots from the MacOS terminal. If you are using a Windows machine, please search for your Anaconda Command Prompt and run the same commands there. This is essentially your proxy terminal for interacting with Conda. Here are a couple more resources if you aren’t sure about Conda or Virtual Environments.

What is a virtual environment: <https://docs.conda.io/projects/conda/en/latest/user-guide/concepts/environments.html>

Getting started with Conda: <https://docs.conda.io/projects/conda/en/latest/user-guide/getting-started.html>

**STEP 1: Practice setting up a test environment using Conda:**

1. Open up your Terminal (MacOS) or Anaconda Command Prompt (WindowsOS)



Your current environment.

Your current user

Your computer name

type the following command: conda -V (ENTER)

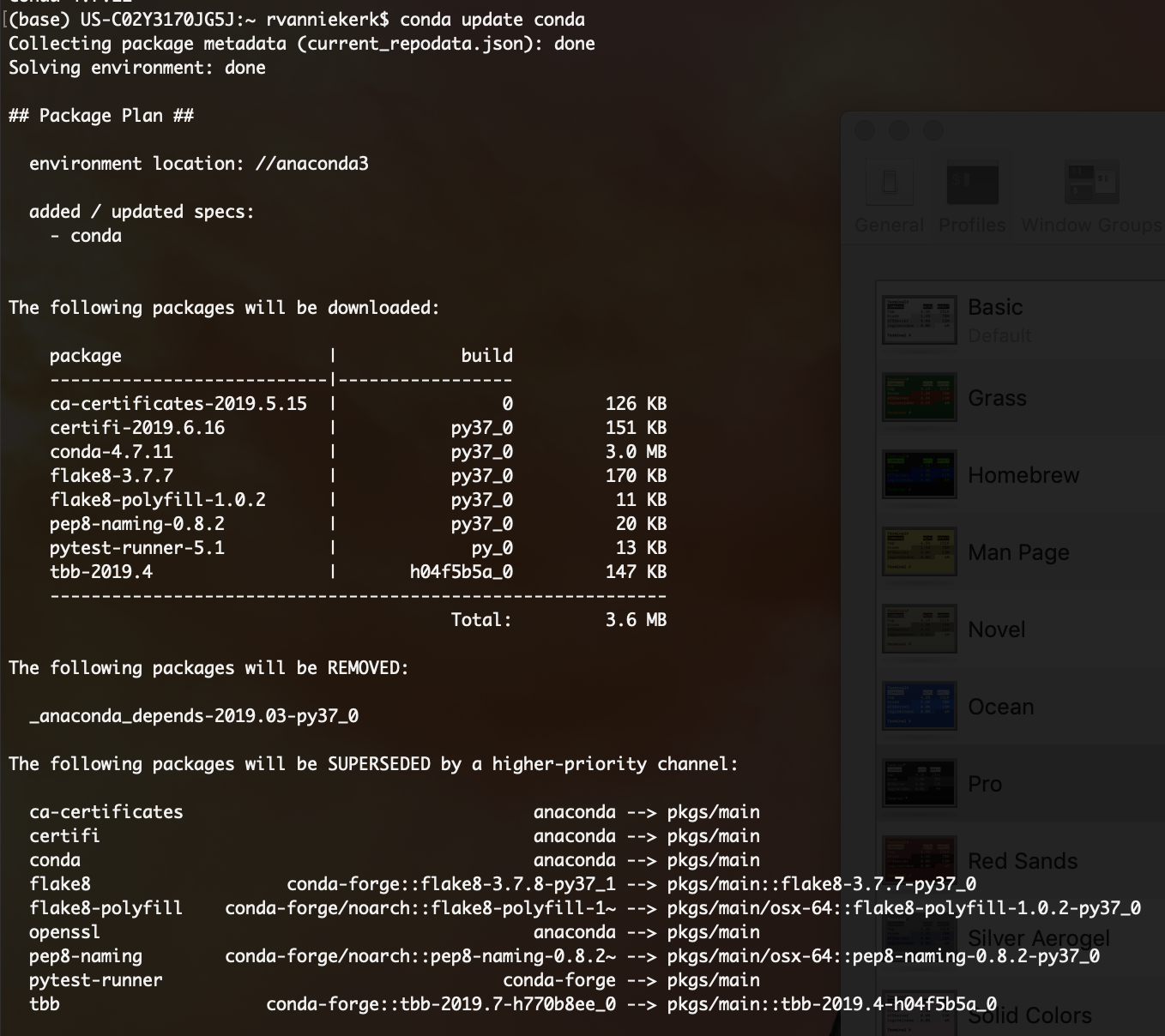
This tells you that conda is installed and returns the version of conda



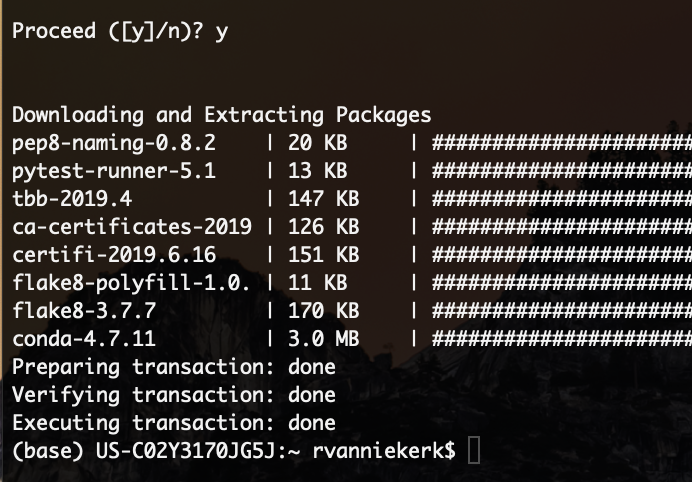
1. type the following command:

conda update conda (ENTER)

This will ensure conda is updated to the latest version.



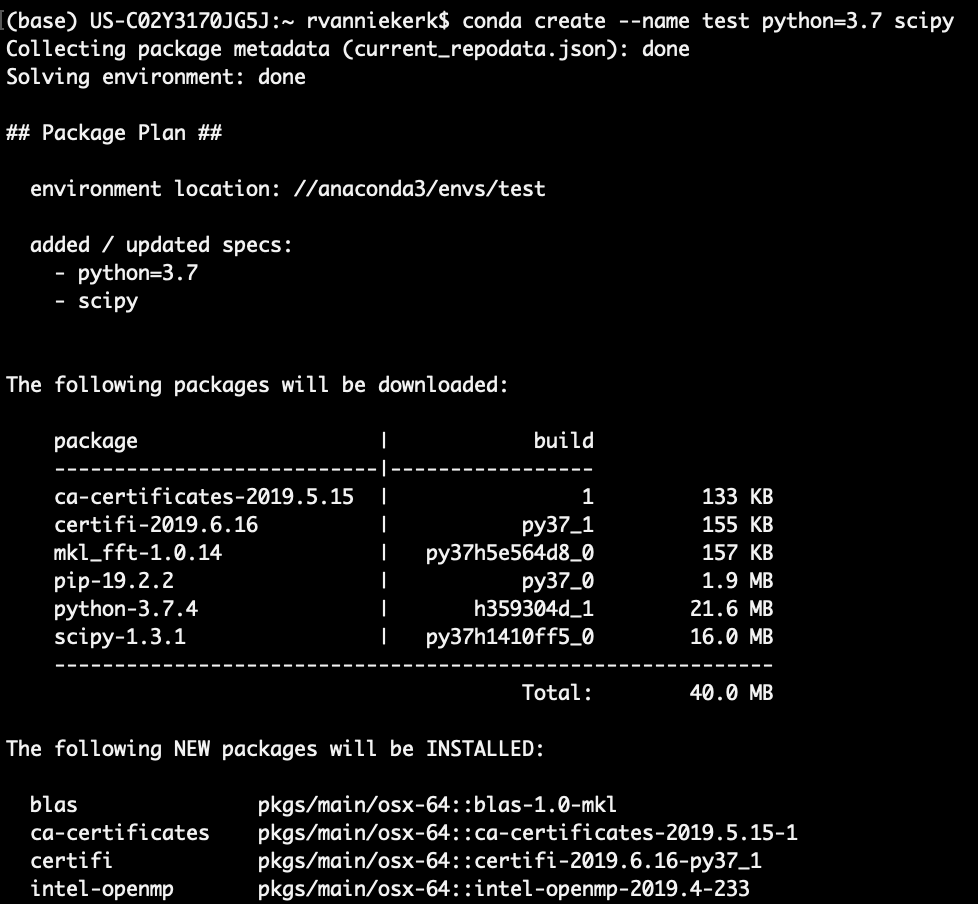
You may be asked to Proceed, if so enter ‘y’ and hit enter.



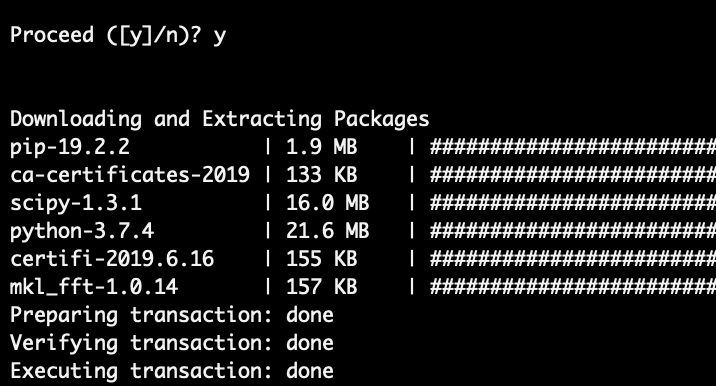
1. Create a new test virtual environment by running the following cmd line code:

conda create --name your\_env\_name python=3.7 scipy=0.15.0 (ENTER)

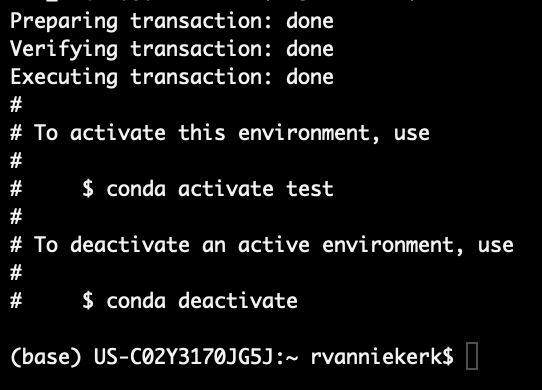
at this point it will give you a printout of packages to be installed, downloaded and any dependency requirements.



Again, you will be prompted to Proceed. His ‘y’ and then enter.



You will also see it explicitly tells you how to activate and deactivate your environment.



1. Type the following CLI cmd:

conda env list (ENTER)

This shows that you now have a new environment called test however the Asterix indicated that you are not ‘in’ that environment. You are still in your base installation.



1. Type the following CLU cmd:

conda activate test (ENTER)

conda env list (ENTER)

Each time you want to run your Jupyter notebook which is built on libraries/packages associated with a Virtual Env, you will need to activate it first.

This ensures you are isolated from your base installation and you can install packages and go wild without the risk of corrupting your base installation.



1. Deactivate and delete your test environment since we aren’t actually going to use this test environment:

conda deactivate (ENTER)

conda env remove -n test (ENTER)

conda env list (ENTER)

You need to conda deactivate your environment before removing it. As can be seen, once I list my environments again, our test environment has been removed.



**STEP 2: Loading a new virtual environment from an existing .yaml file**

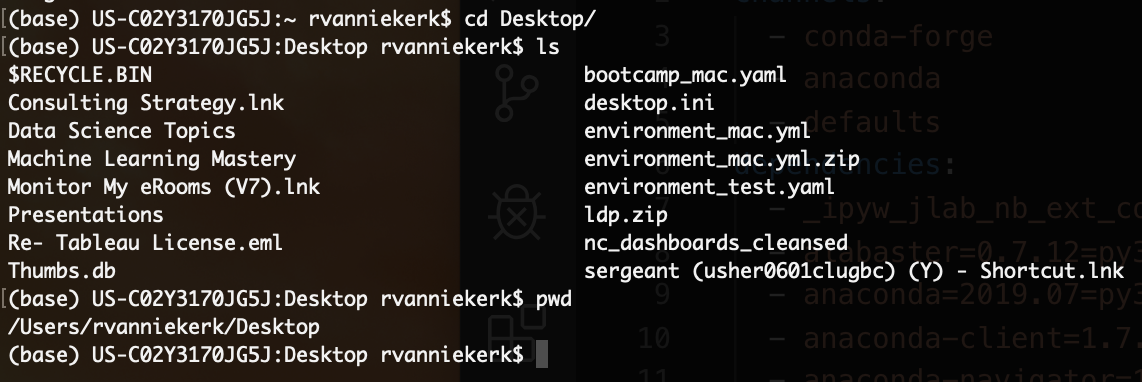
A [.yaml](https://en.wikipedia.org/wiki/YAML) file is a file is commonly used as a config file since it’s human readable. We will use a pre-built .yaml file for building a virtual environment that we will use during the bootcamp.

In short, the .yaml file contains a set of instruction which Conda will interpret and use as a basis for creating a virtual environment. Once downloaded you can open up your respective .yaml file and inspect the contents (please don’t change anything in there). You will notice that there are over 300 packages referenced in that .yaml file. Some of this we won’t use but should provide you with a working environment from which you can run the bootcamp notebooks without experiencing any errors.

***Please note: We have provided separate files for WindowsOS (bootcamp\_win.yml) and MacOS (bootcamp\_mac.yml) in the same directory as the prework directory on GitLab, in the mlg-05-us project folder. Please download your respective operating systems’ .yaml file to somewhere on your desktop as you will need to point to this file when creating your virtual environment.***

1. Download your respective .yaml file from GitLab
   1. Save this file to anywhere on your local system. I’d save mine to my desktop as the file path is short and you’ll need to specify your file path when entering your setup CLI cmd.
   2. Windows users will download the bootcamp\_win.yaml
   3. MacOS users will download the bootcamp\_mac.yaml
2. Open up your terminal (MacOS) or Anaconda Command (WindowsOS) and type the following cmd line code making sure to explicitly type out the relative file path to where you have saved you .yaml file. Please note that you should probably navigate to your desktop via the cmd line to ensure you are in the same directory as your .yaml file when executing the cmd line code below.

Navigating to my desktop…



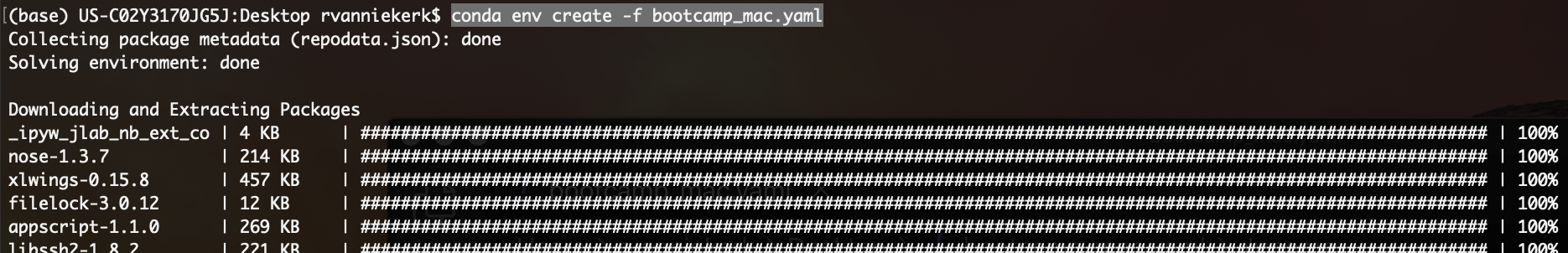
Here is my current working directory

Here is my config file

Type the following to create your environment:

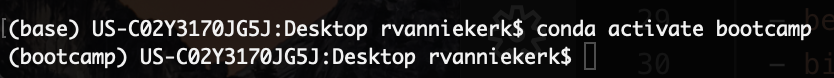
conda env create -f bootcamp\_mac.yaml (ENTER)

You should see somethings similar to what you see in the screengrab below



Since it’s going to be installing over 300 packages, it’s going to take some time… Give it a couple minutes and wait for it to finish installing. Mine took about 5-10 minutes.

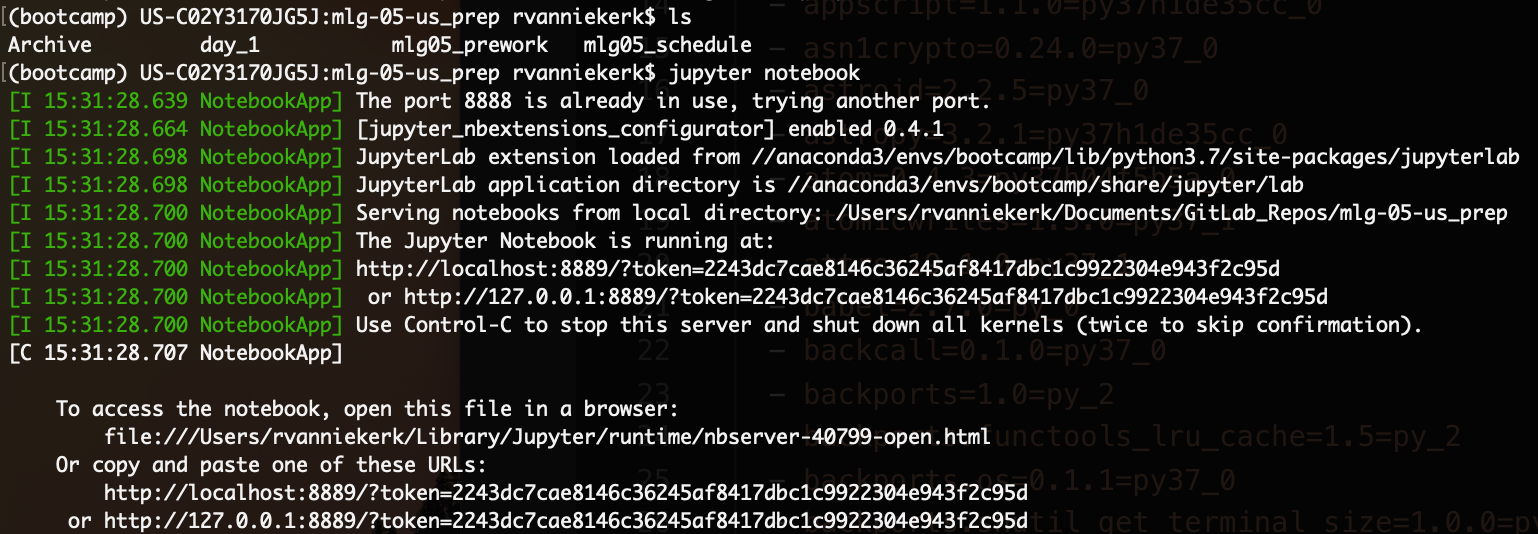
1. Activate your new environment bootcamp environment and ensure its installed



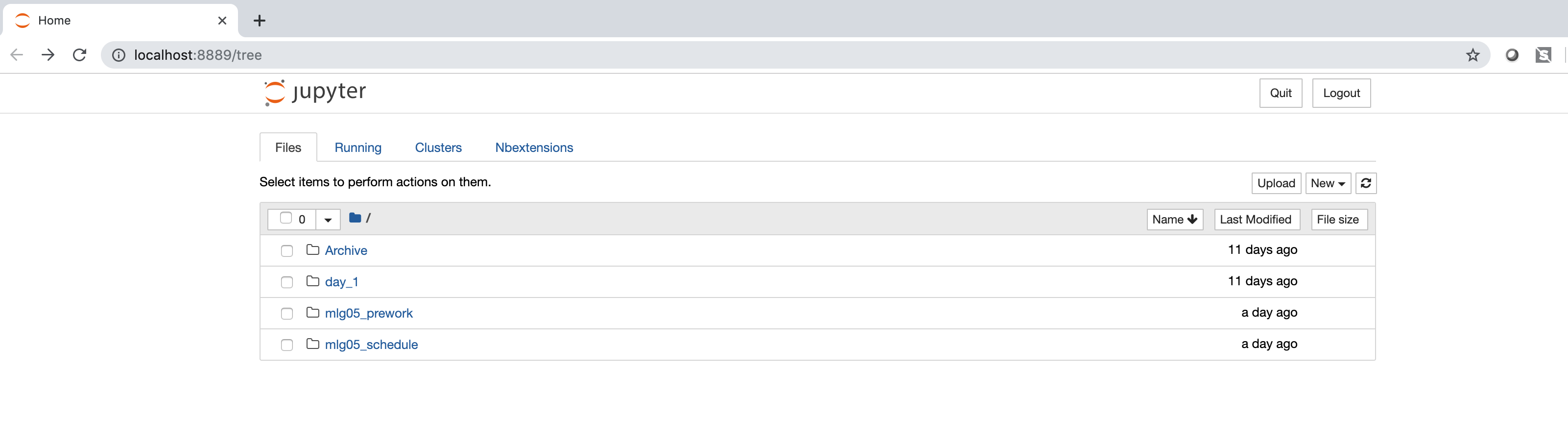
1. Lastly you will need to run the following command to spin up your jupyter notebook from this virtual environment. If you try this code without having jupyter installed, you may either get an error or you will be spinning up a version of jupyter notebook associated with another environment.

Secondly, you should consider navigating to the folder which you’d like to work in for the duration of the bootcamp before spinning up your jupyter notebook. Preferably I’d say the mlg-05-us folder you’ve cloned using GitLab.

jupyter notebook (ENTER)



1. If you’ve done everything right, you should have a local hosted window pop up running you jupyter notebook. If you’ve already cloned and pulled the mlg-05-us repo from GitLab, AND if you’ve navigated to that folder and run the jupyter notebook command from there, you should be set up and ready to go. Here is an example of what is should look like.



Lastly; if you are still having issues, we will set up skype session on the afternoon of Friday, September 13th where instructors will sit down with students who are having issues to assist in getting them set up git Git and GitLab. Please look out for this invite if you are having any trouble.