**Zach Adair Week 5 Lab 5 Write Up**

**Install Anaconda Python on Vagrant Cluster**

**Regis University – College of Computer & Information Sciences**

**MSDS 610 – Data Engineering**

**February 12, 2019**

# Lab Overview

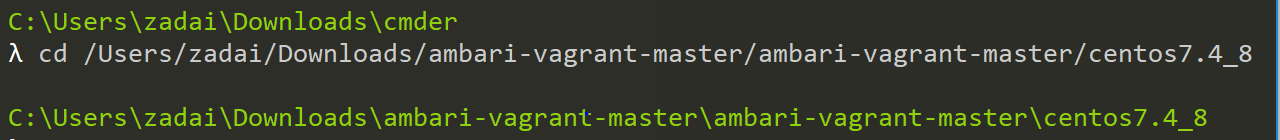
We will be downloading Anaconda Python from Anaconda Website and installing it on each of the nodes. Fortunately, Vagrant gives us a shortcut that helps this process.

After that we’ll learn how to use Jupyter Notebooks to document, run and visualize Python.

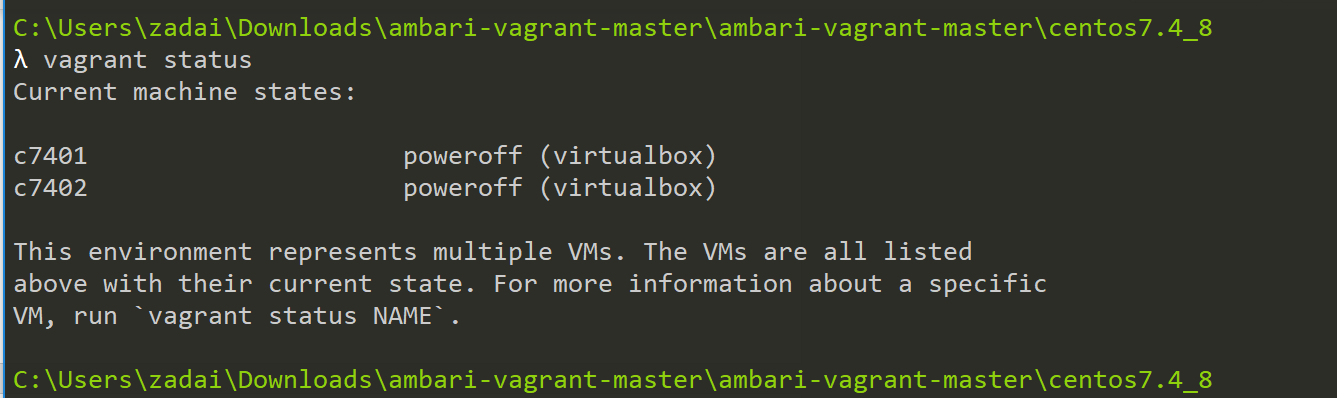
You will clean and process the data using Pandas, Numpy, and maplotlib libraries (Included in the Anaconda Python distribution).

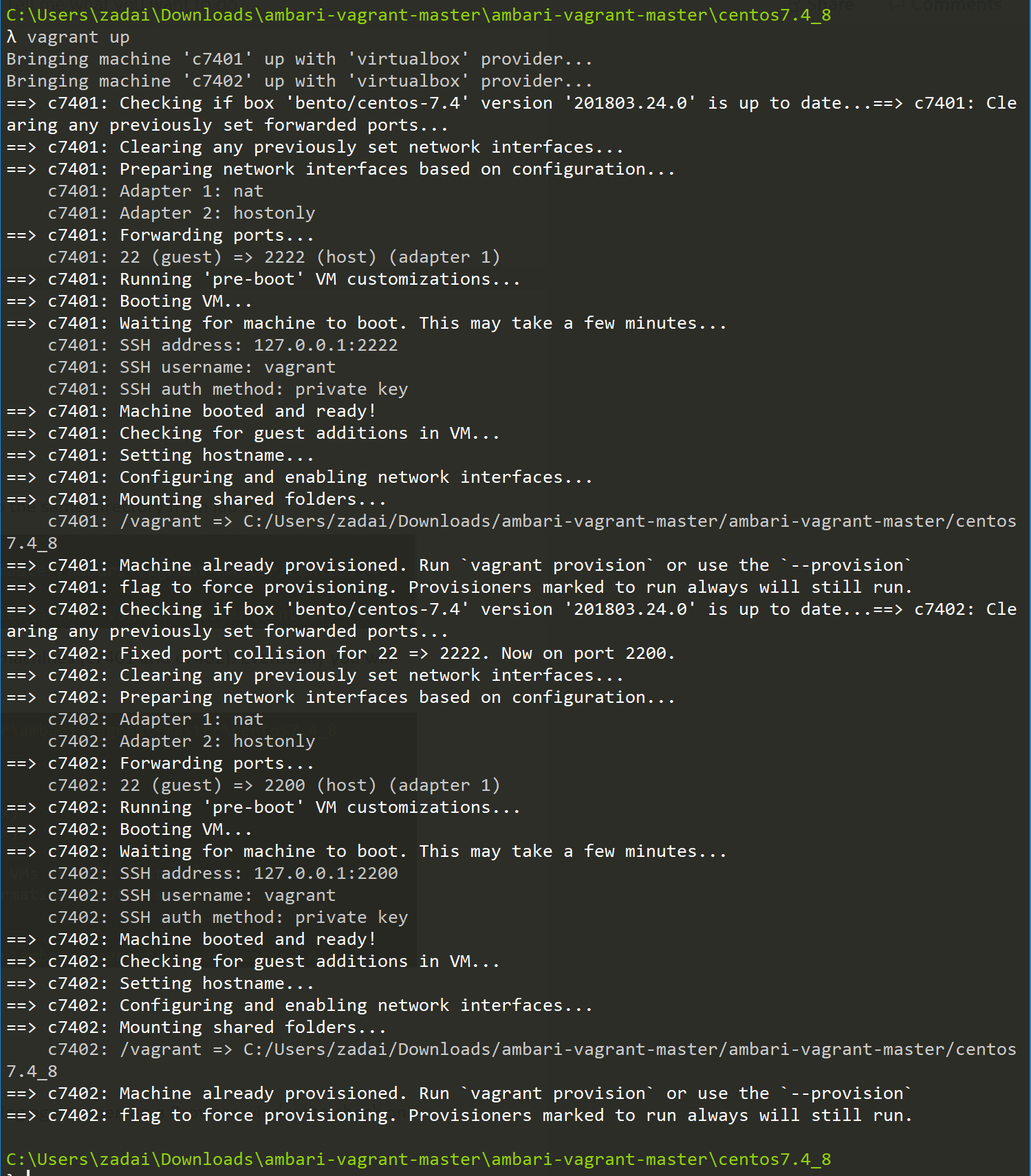
## Resume VMs

Start Cmder, if it isn’t running. Change the directory to the same directory from lab 2:



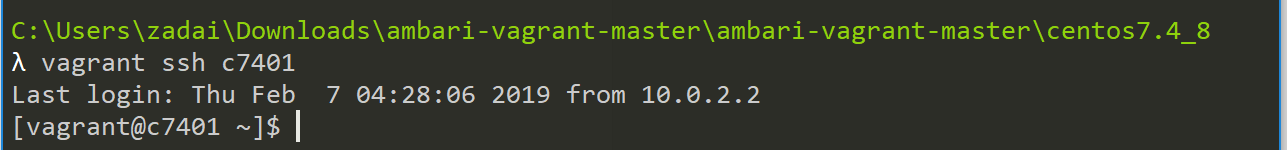
This can take a few minutes to restart the two virtual machines (c7401 and c7402). Eventually you will see they have both restarted:



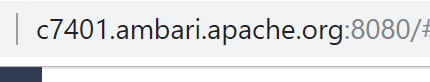


## Ambari Console

Now that the Virtual Machines are restarted and running again, open up a Browser window to check on the status of the Virtual Machines using Ambari.



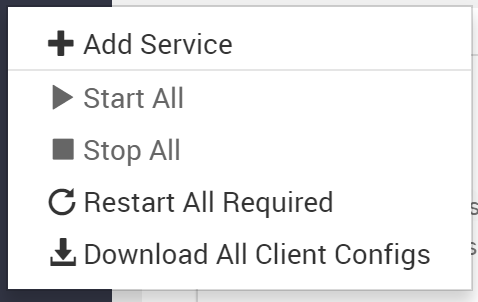
Use the URL: <http://c7401.ambari.apache.org:8080>

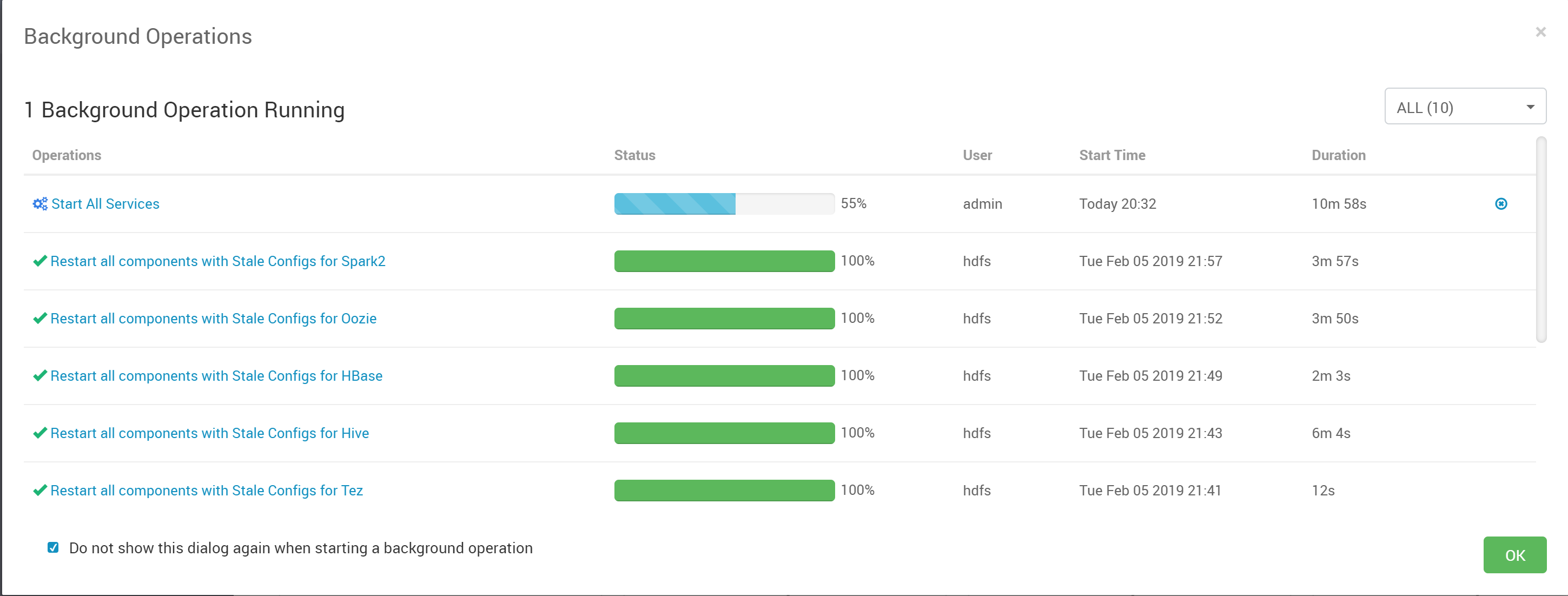


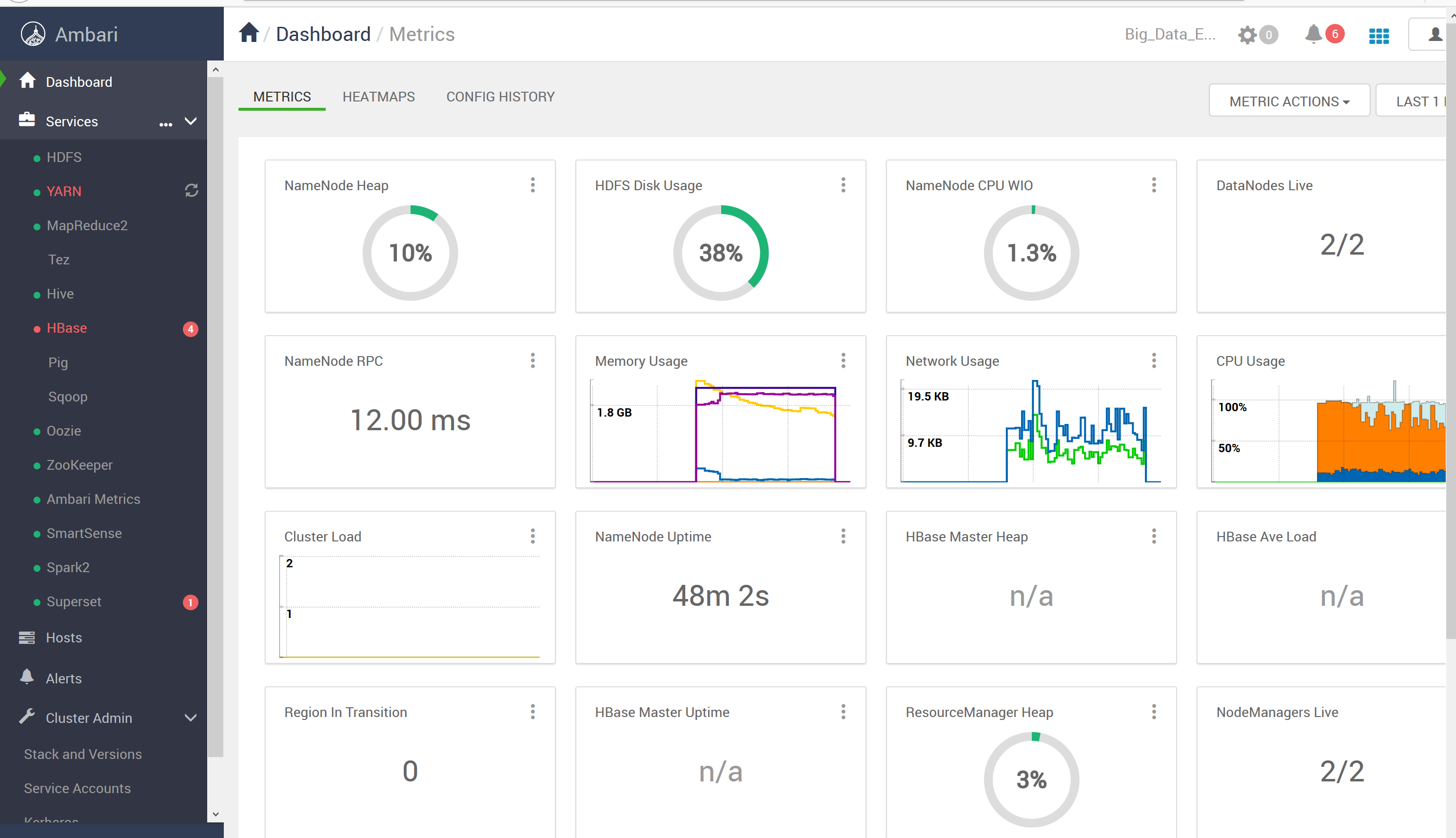
Use the admin ID and password login and click the sign in button

**Once in, your window should show that all datanodes are live or running.**

**The dashboard is not showing all services up and running than perform the Start All function to manually start all the services.**





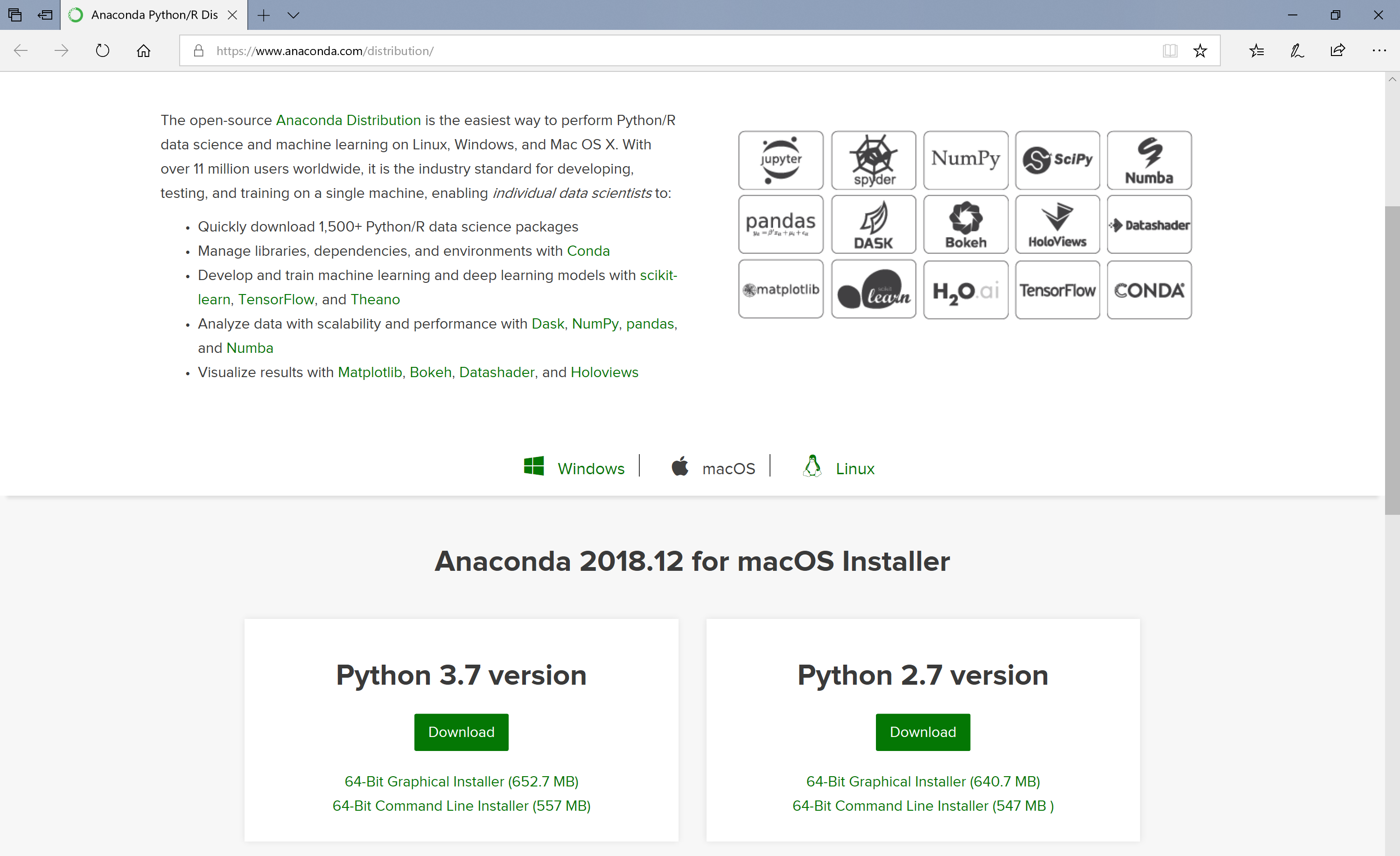


**After running start all I have all the hosts running that I need to have running. Like usual HBase is broken (Still mad it isn’t working) but every other host is so I will move on from here.**

I could start some services that didn’t start after running the Start All option, go in and run each of them manually. Go into Hosts, and then start the services on each hosts manually until everything is running successfully.

## Installing Anaconda

Anaconda can be downloaded from <http://www.anaconda.com/download/>. When downloading from the Ananconda website remember to change the download type to “Linux”

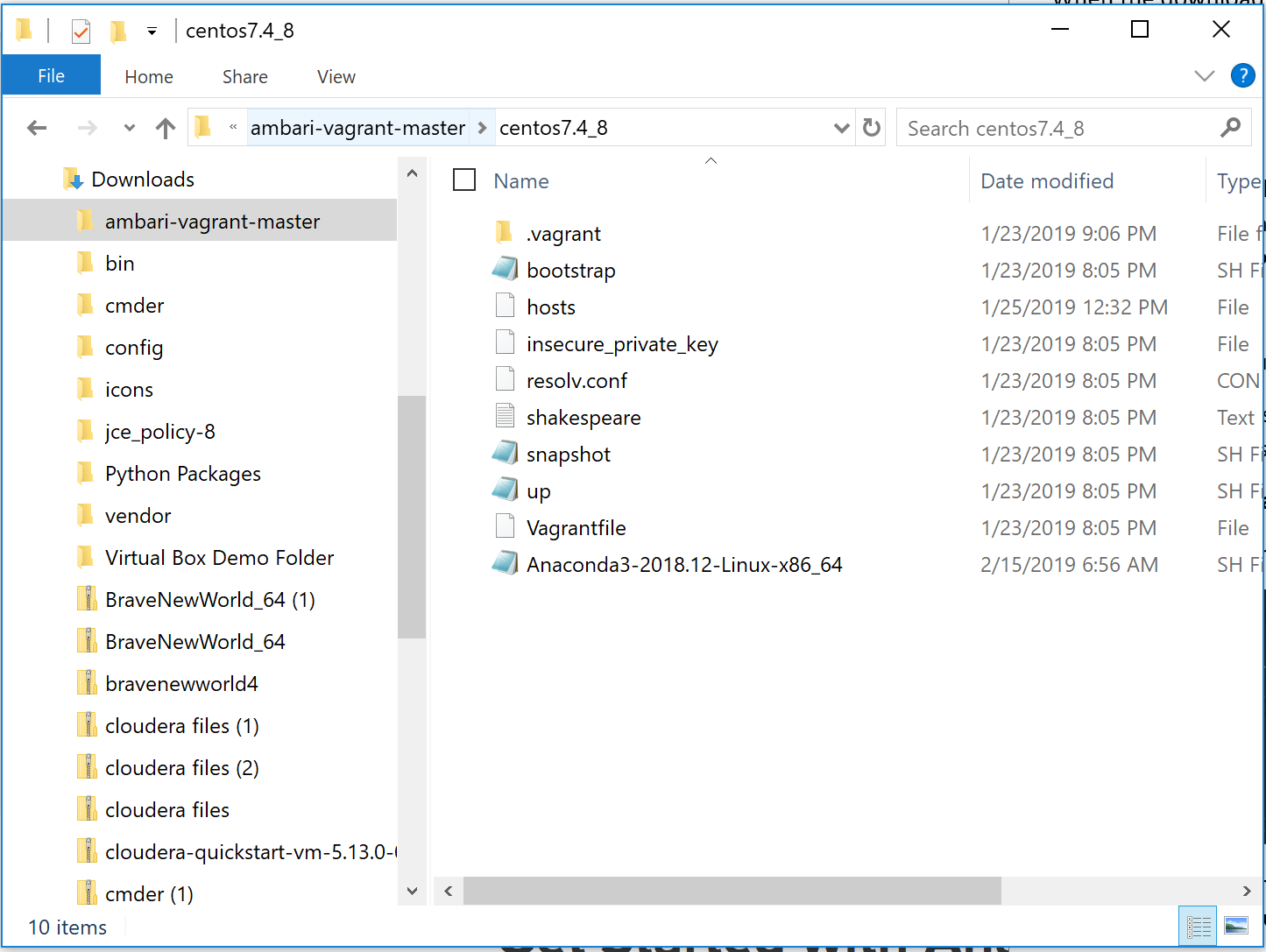


**Note: Make sure you select Linux as the operating system as shown above!**

If it asks you to enter an email address, just close that dialog box, it’s not necessary to enter in an email address.

When the download completes, **copy** the file into your **ambari-vagrant/centos7.4\_8** directory on your local PC

The next screenshot is what it looks like to have the anaconda file in your centos7.4\_8 directory



It is important that the file you downloaded has a “.sh” extension on it. This indicates the file is a Linux shell script and not a Windows executable package file. If your file doesn’t have a .sh extension, something went wrong with the download. Delete all copies and try again.

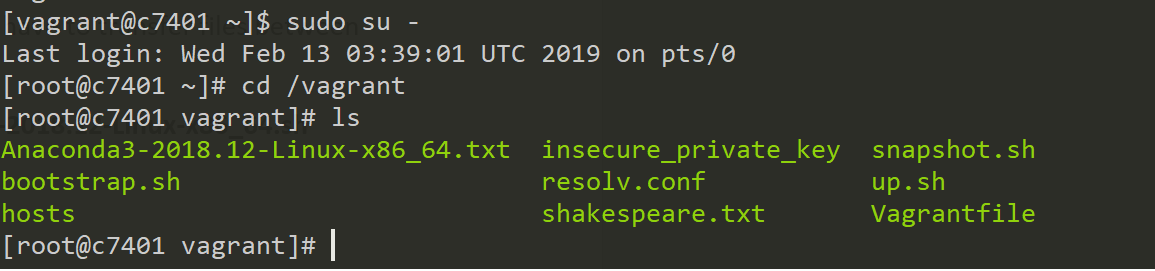
Now ssh to the c7401 machine using the good old: **vagrant ssh c7401**

Change to the root user: **sudo su –**

Now let’s visit a fun shortcut in Vagrant. The centos7.4\_8 directory (on your local PC) is linked to the virtual machines! It shows up as the /vagrant directory!

Change to the **/vagrant** directory: **cd /vagrant**

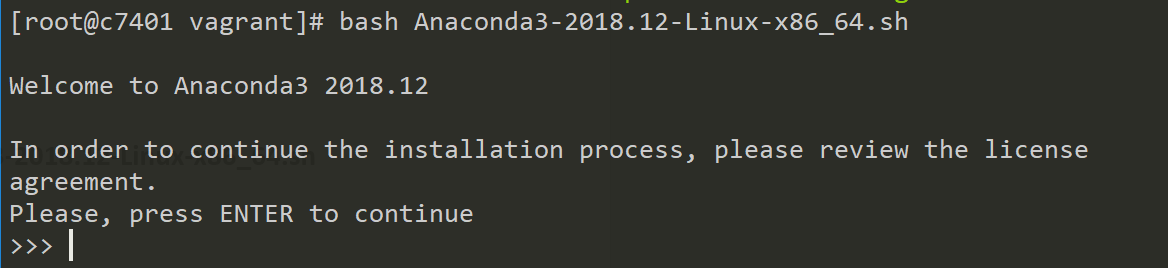
Then enter the ls command as shown below.



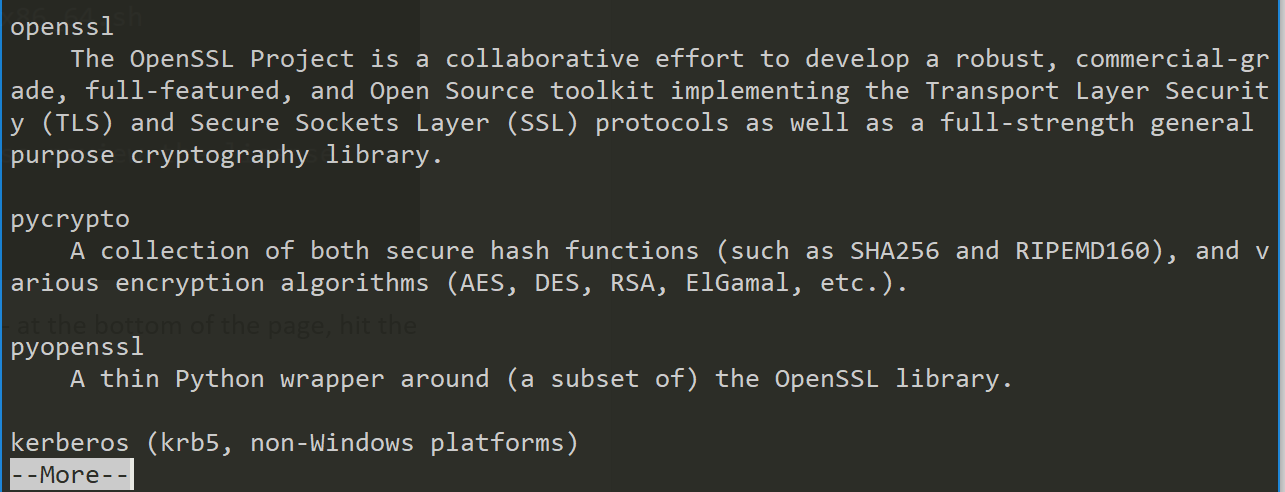
It’s helpful to remember this unique vagrant feature so that you don’t have to transfer files between your local PC and your VMs. It’s a built in link.

Now, run the installer using the following command: **bash Anaconda3-2018.12-Linux-x86\_64.sh**

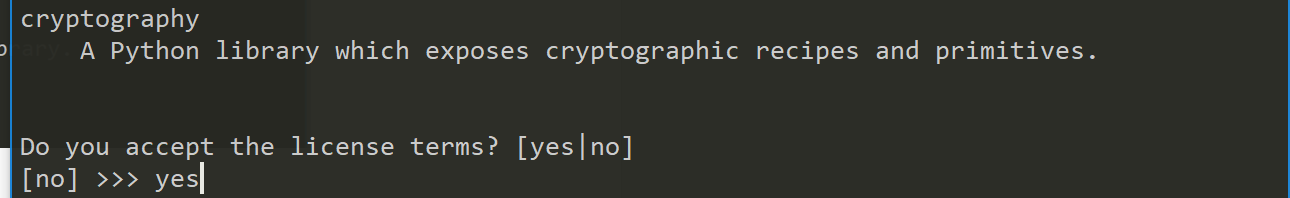
**Note: The version numbers in your file may be different**



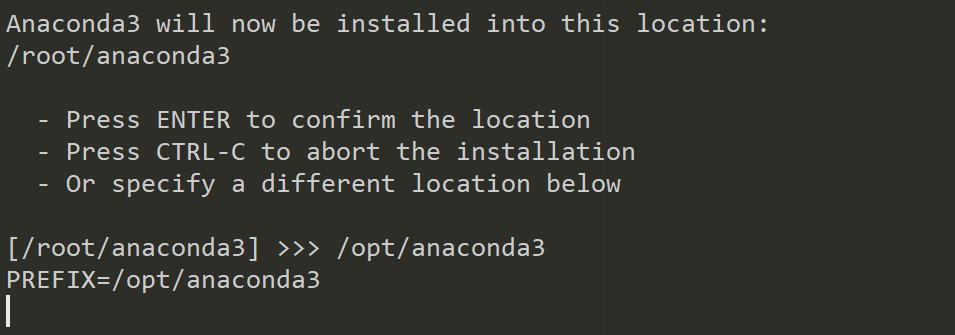
From here on, follow the prompts on the screen. When you see More- at the bottom of the page, hit the spacebar to page through the output.

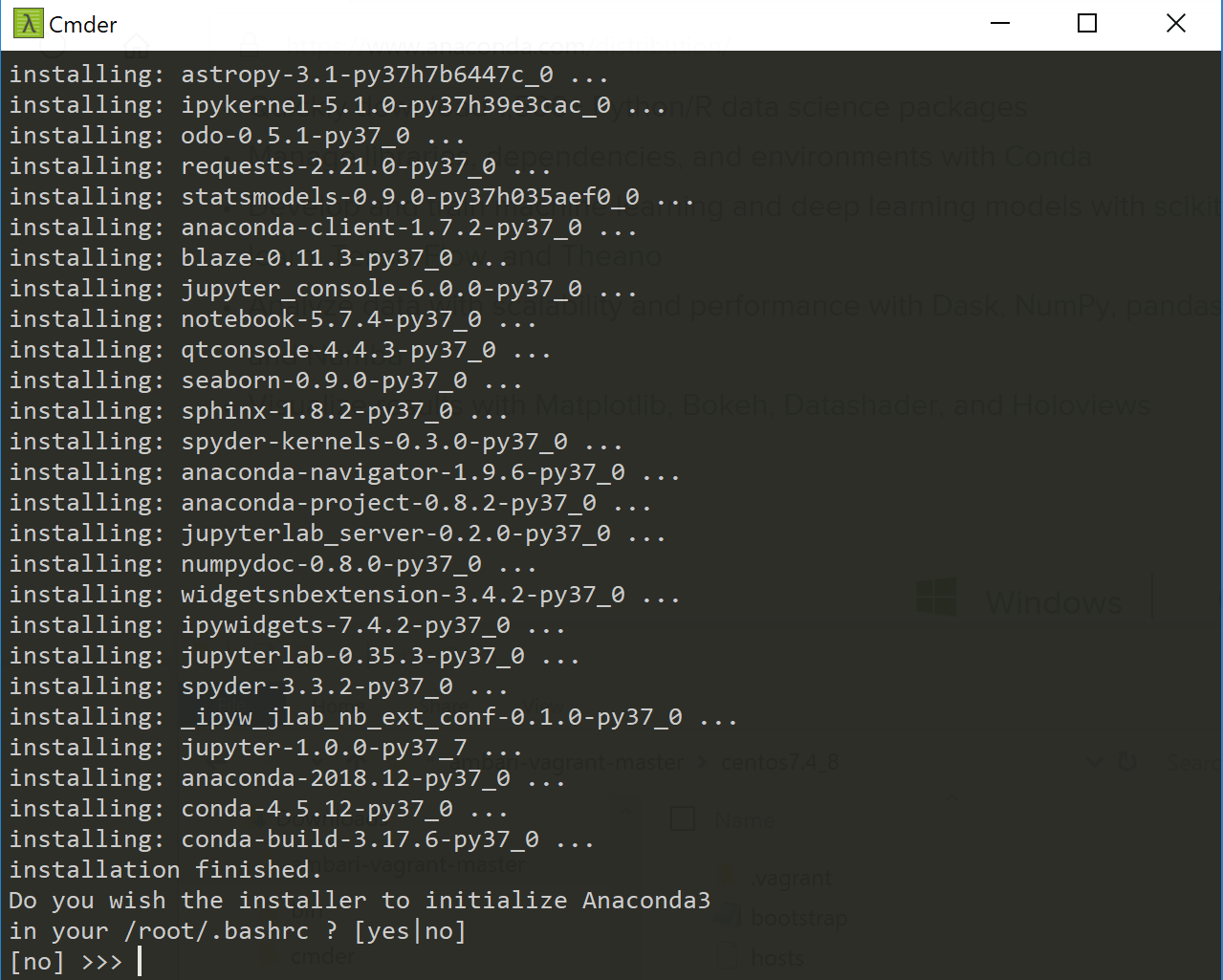


Answer **yes** to accepting the license terms.

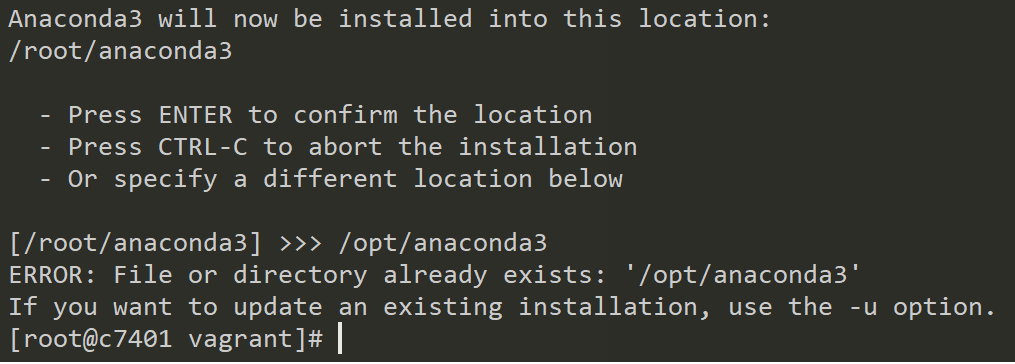


IMPORTANT! When prompted for the install directory, change the directory to /opt/anaconda3 as shown below:



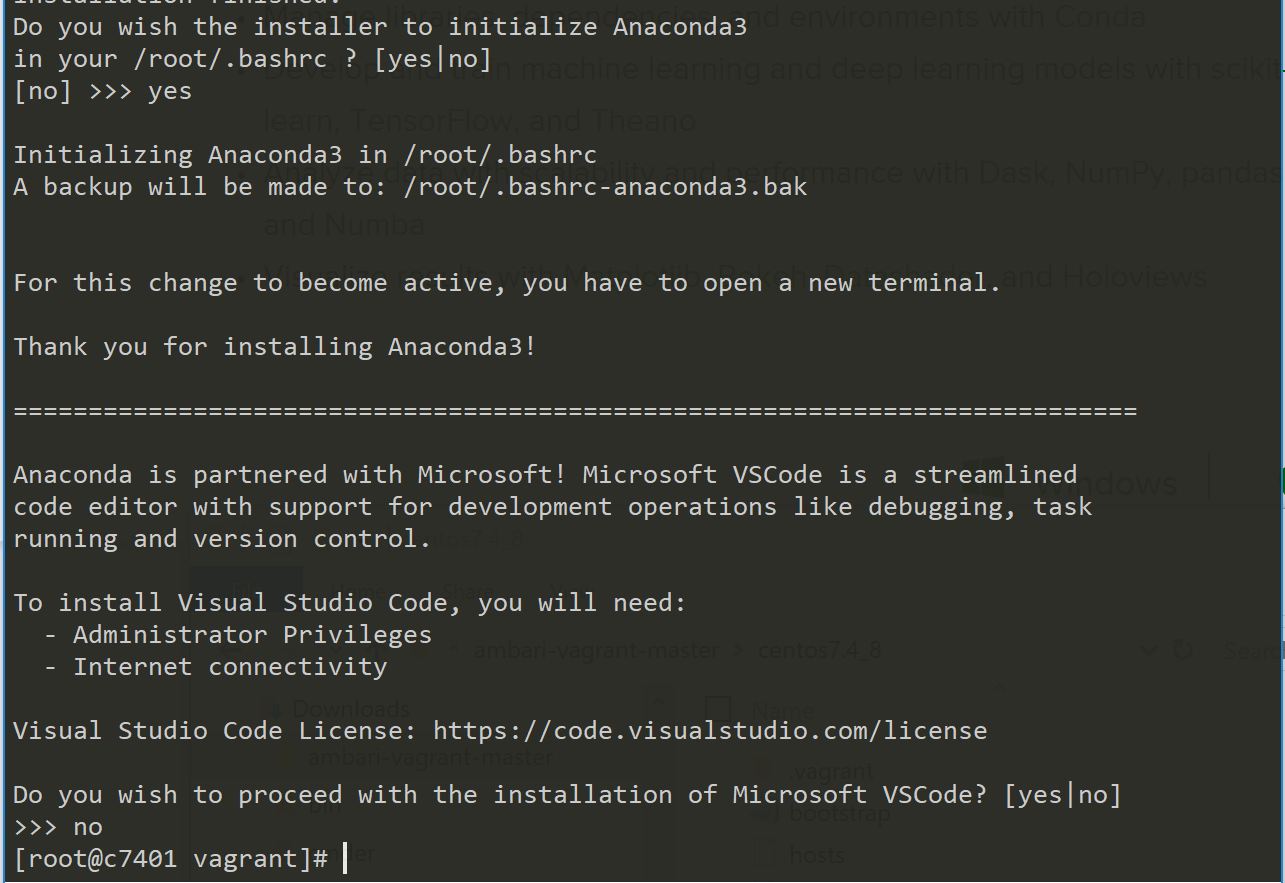


A bunch of “installing:” messages will scroll past on the screen. This can take a long time to install so sit back and relax or go do something while you wait.



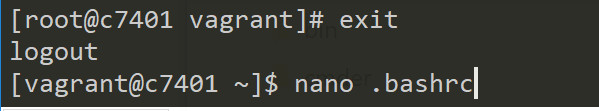
**My first time going through the process I ran into an error and was kicked out. I tried again after not finding anything from research just to see if I made a simple user error and I got an error because my file or directory already existed. So I decided I could keep going from there. My guess is my original error was that the process just timed out and then promptly kicked me out of the Anaconda environment. The resolution to my issue was that I had to remove the /opt/anaconda3 from my directory and start over by redownloading the Anaconda file.**

At the end of the installation, you’ll get a message asking if you’d like to add Anaconda3 to your /root/.bashrc file, **say yes to /root/.bashrc but say no to MS VSCode**



**The install has completed and were now ready to move forward on the lab.**

Now, exit to get back to the vagrant user. Next, nano the .bashrc file in the /home/vagrant directory.



**Nano was not a command option for me but vi was so switch nano with vi before .bashrc**

Add the following export command to the .bashrc file: **export PATH=”opt/anaconda3/bin:$PATH”**

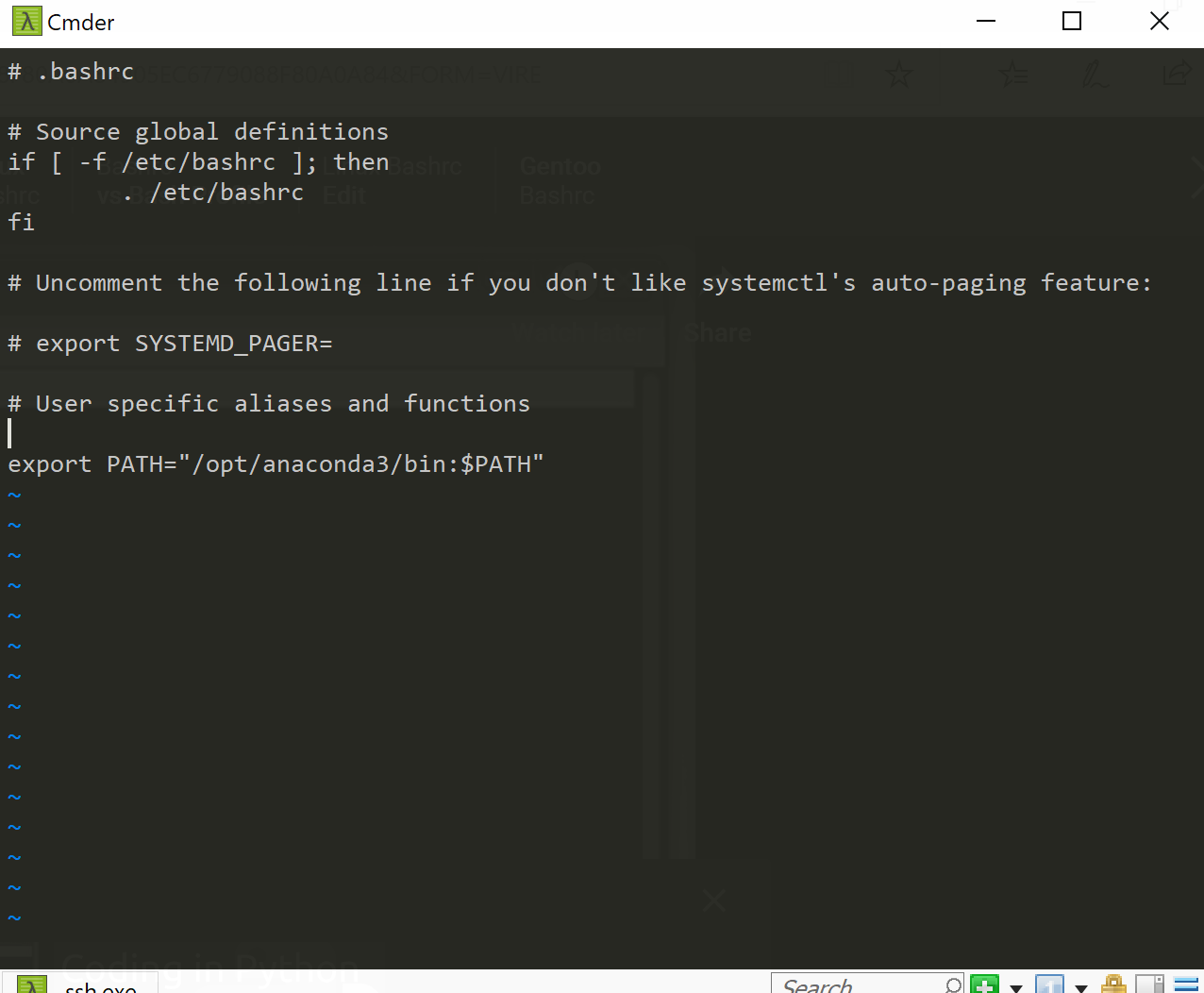
**NOTE: to insert the command above in the .bashrc file hit “i” to insert, to go into EDIT MODE hit “a”, to go into COMMAND MODE: ‘esc’, you need to be in command mode for these next crop of options.**

**To save file - :w + enter**

**To Save and Exit - :x + enter**

**To Exit - :q + enter**

**To Exit without saving - :q! + enter**

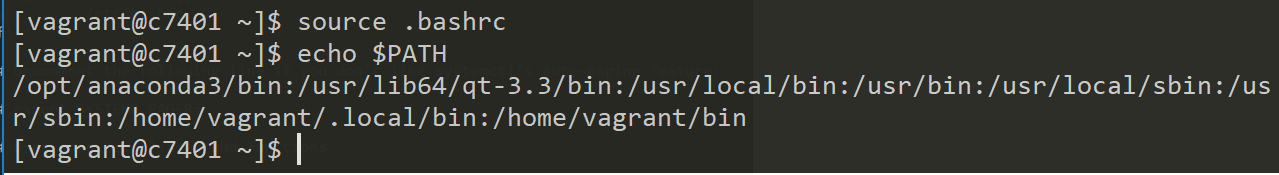


Save and Close the file.

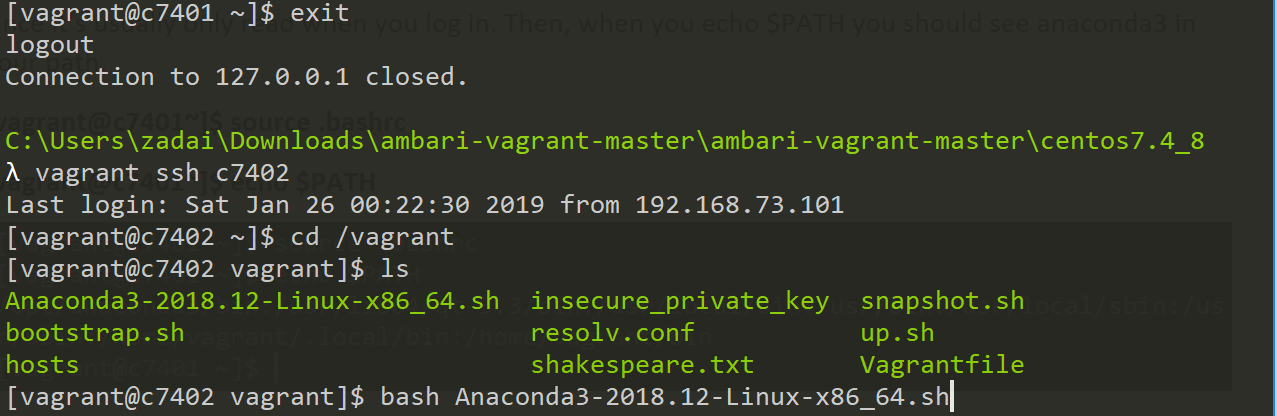
To verify that your PATH variable is set correctly, we need to source the .bashrc file to force it to read it, since it’s usually only read when you log in. Then, when you echo $PATH you should see anaconda3 in your path.

**[vagrant@c7401~]$ source .bashrc**

**[vagrant@c7401~]$ echo $PATH**



To start the install on the other VMs remember that you can use the /vagrant shared directory as you used to c7401, and below is a screenshot to get you started…



And that is the end of the installation for the Anaconda Python on all your VMs!

# Jupyter Notebook

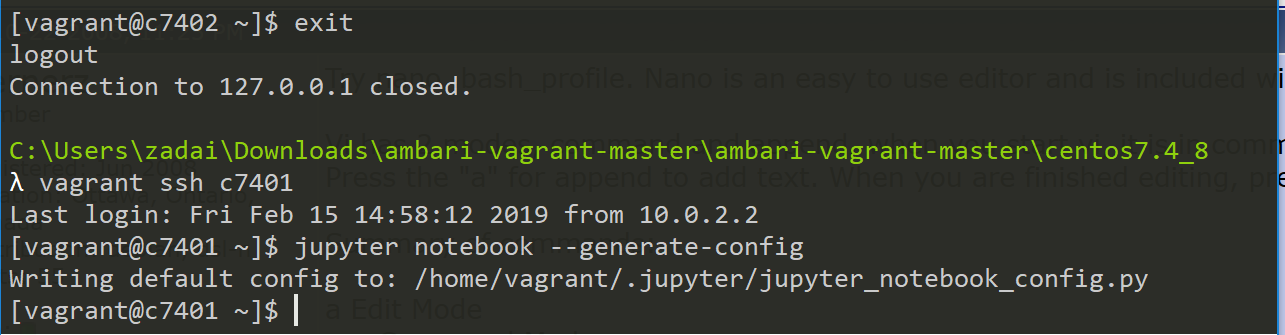
Jupyter notebook (formerly called IPython notebook) is a popular tool in both scientific computing and data science. The Anaconda distribution of Python comes with Jupyter notebook already installed.

The Jupyter notebook interface runs in a web browser and allows you to add markdown text, running code, visualizations, etc.

### Cluster setup

We will be running the notebook from the c7401 machine, so log back into VM. Then, we will start our jupyter notebook configuration.

Jupyter notebook –generate-config



After the command runs you’ll see the path and name of the file written. We will have to edit that file

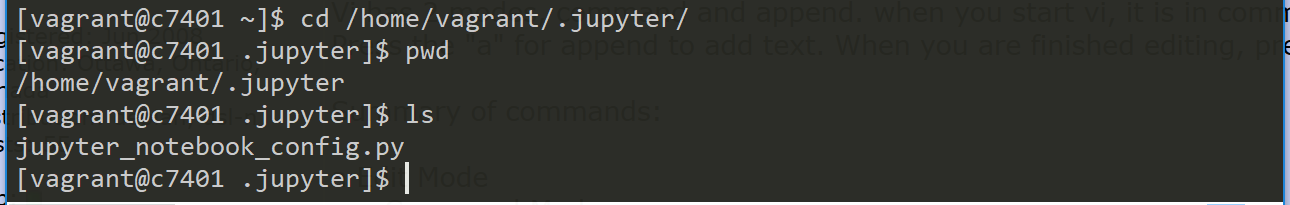
Change to the directory using the cd command as shown below:

**cd /home/vagrant/.jupyter/**

**pwd**

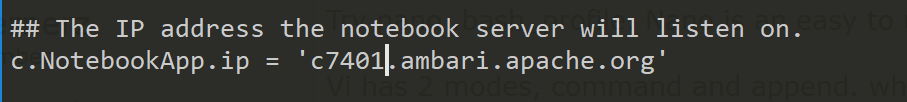
**ls**

You can double check your location with pwd and the ls to list the file

c

Nano jupyter\_notebook\_config.py (you could also use vi here instead of nano)

We are looking for the c.NotebookApp.ip parameter. Ctrl-W searches in nano, so type Ctrl-W (control and shift and W) and enter c.NotebookApp.ip and press enter. **Since I’m using vi instead of nano I don’t have a search command to find it. But I realized #c.NotebookApp went in a sort of alphabetical sequence so I was able to find the ip and change it to the ambari.apache.org**



Change from ‘localhost’ to c7401.ambari.apache.org and remove the # from the line.

Save and exit the file. It should now be possible to connect from your host’s browser to the VM’s Jupyter notebook. And that is what we will test now.

### 1.4.2 Testing Jupyter Notebook

One of the secrets to running Jupyter notebook server is to cd to the directory you have your notebook files in before you start it. In this case we don’t have any of those yet so we’ll just start from our home directory. Type:

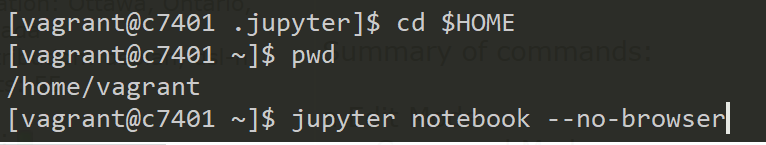
**cd $HOME**

**pwd**

**Note: Make sure you’re in the /home/vagrant directory before you start jupyter as shown below**

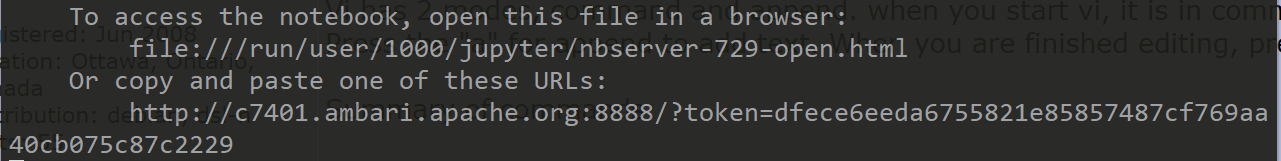
The following command is what you will always use to start Jupyter Notebook.

Jupyter notebook –no-browser



Remember, our VMs don’t have browsers so there is no reason for the server to try to start one. It won’t hurt anything if you forget but I like to save it the trouble.

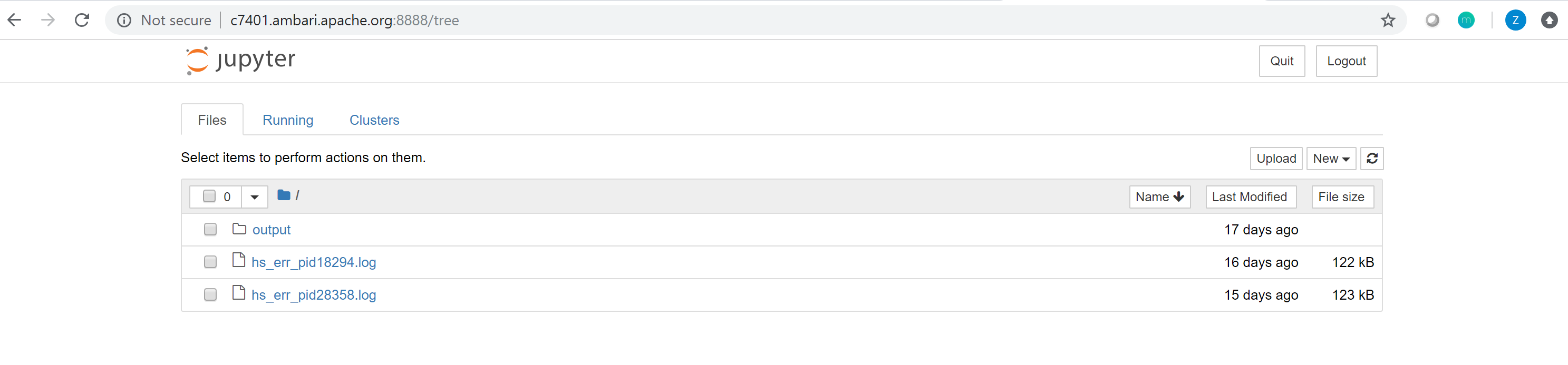
You should get something that looks like this:



At the bottom of the picture you see a URL with a security token.

**Note: that you started the notebook service from the command line, so that window needs to stay active while the notebook is running.**

Now to actually see the notebook you will need to copy that URL from above and use it in a browser. See below. Note that the URL does not show the token after hitting enter.



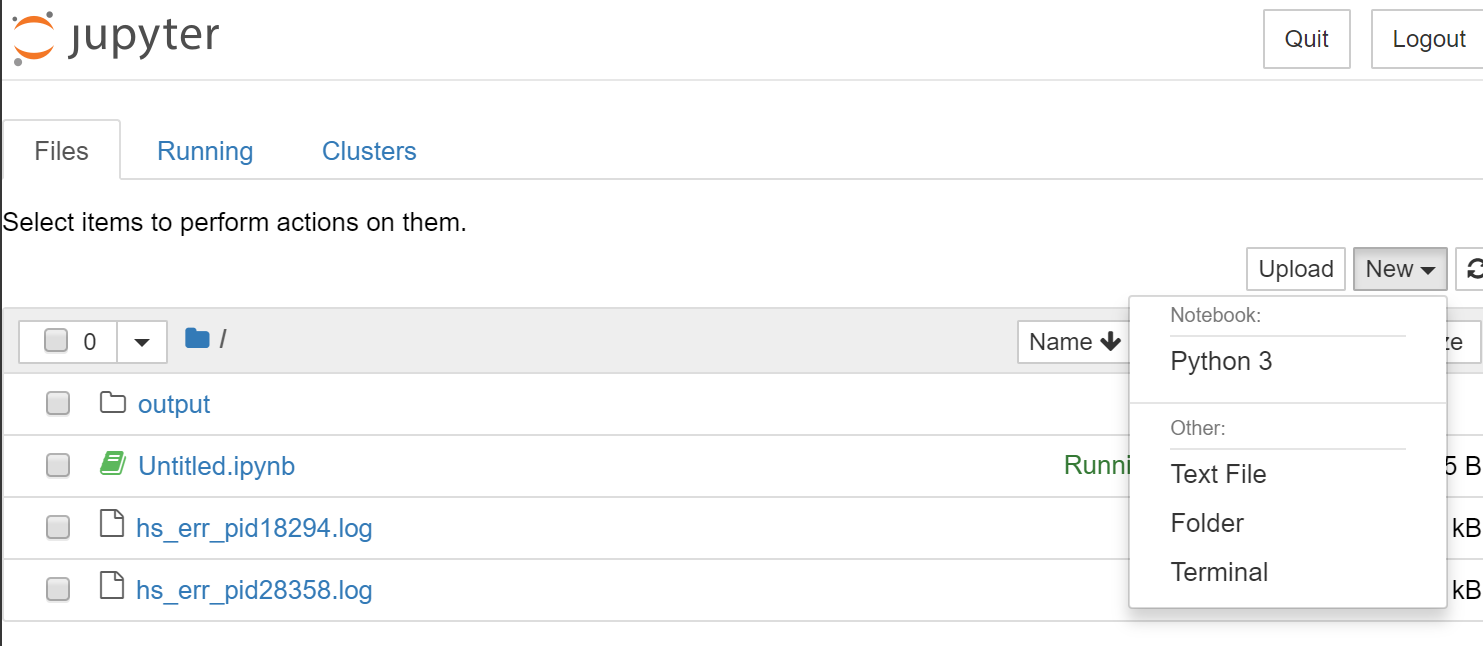
## Next Steps

Watch the Jupyter notebook tutorial video listed in the Week 5 lab and follow along with this new software you installed.

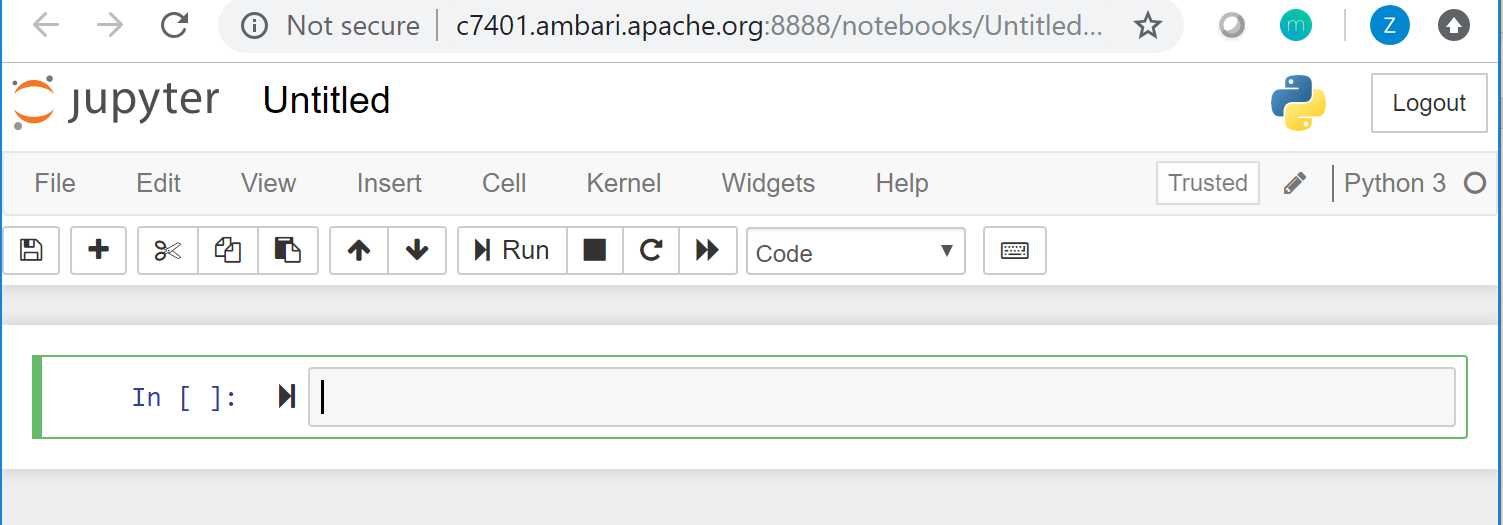
**When finished using Jupyter notebook, use Ctrl + C in your command line window where you started the process, you’ll also have to answer yes to quit the process. Then follow the instructions below to gracefully exit from the ssh, suspend the VM and exit commander**

### 1.5.1 From the Jupyter notebook Tutorial:

**In the home page, start a new page click on ‘New’ and selecting ‘Python 3’**

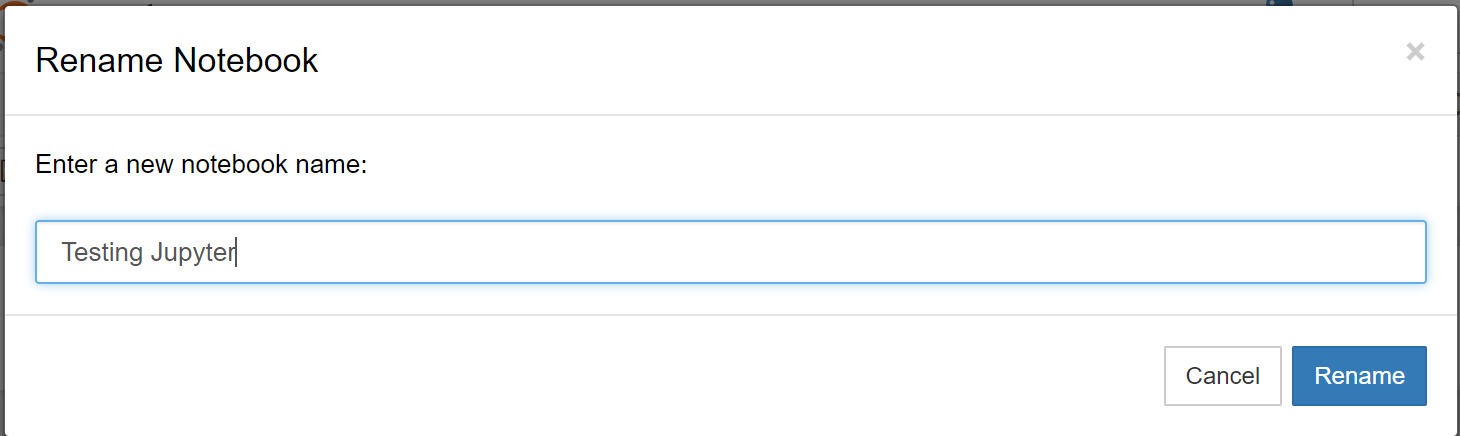


**It will take you to a new page**

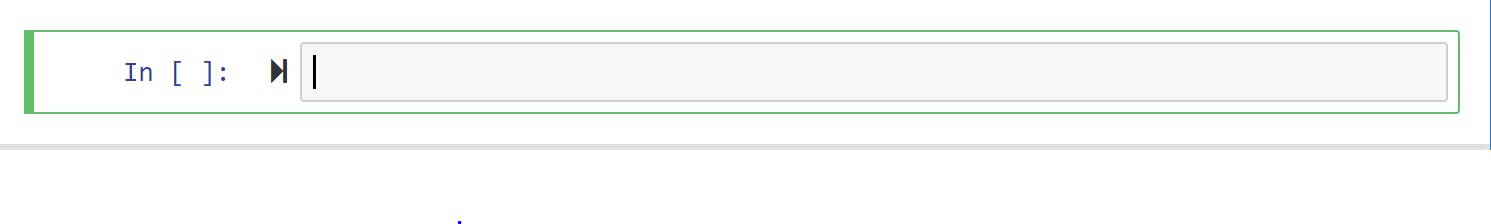


**The help menu button supplies the user with a page of keyboard shortcuts which can be useful for using the tool.**

**Click on the top and change the name of the notebook. I changed mine to ‘Testing Jupyter’**



**There are two types of modes edit mode and command mode. Can get into edit mode by clicking into the line prompt, you will notice the border of the line will change from blue to green indicating you are now in edit mode.**

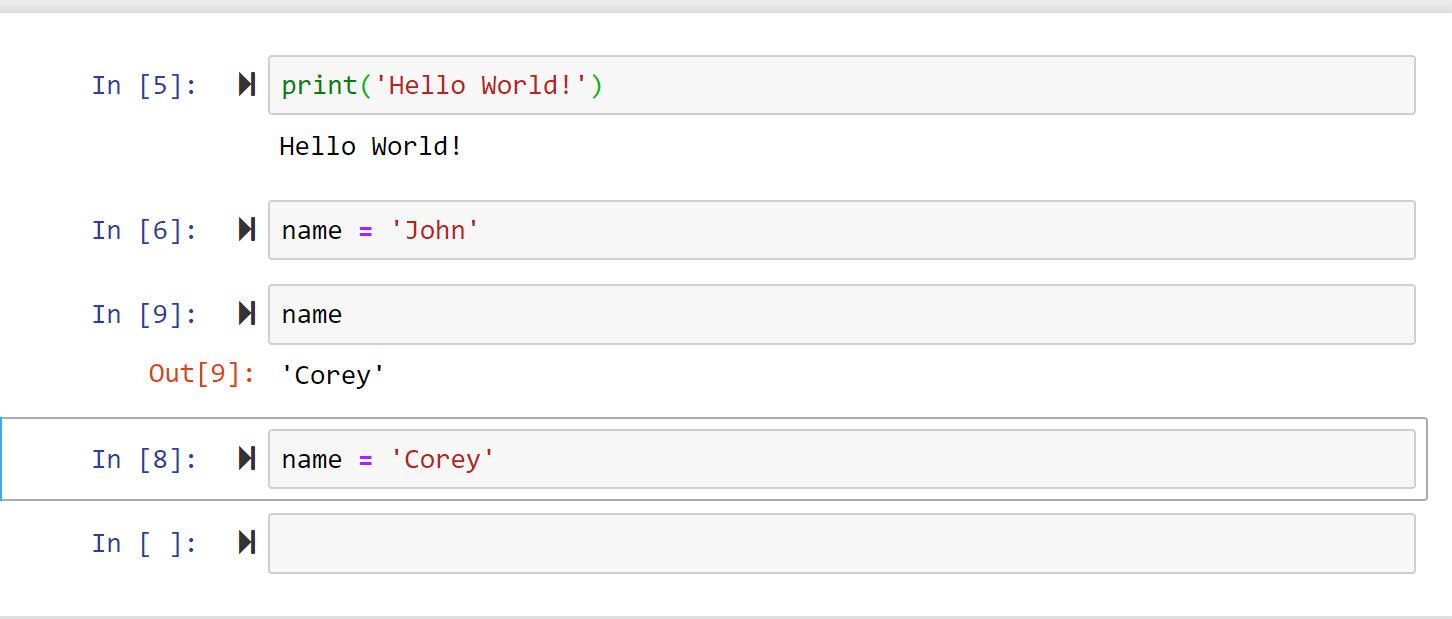


**To execute code, click on the ‘Cell’ option and run one of the options**



**What I got for choosing ‘Run Cells and Select Below’ but also can be done by using the keyboard shortcut Shift + Enter**

**The order of the cells are very important, Jupyter helps keep your lines in order by number your lines and will execute based on order, but if you go back and change a cell you will than see some confusion.**

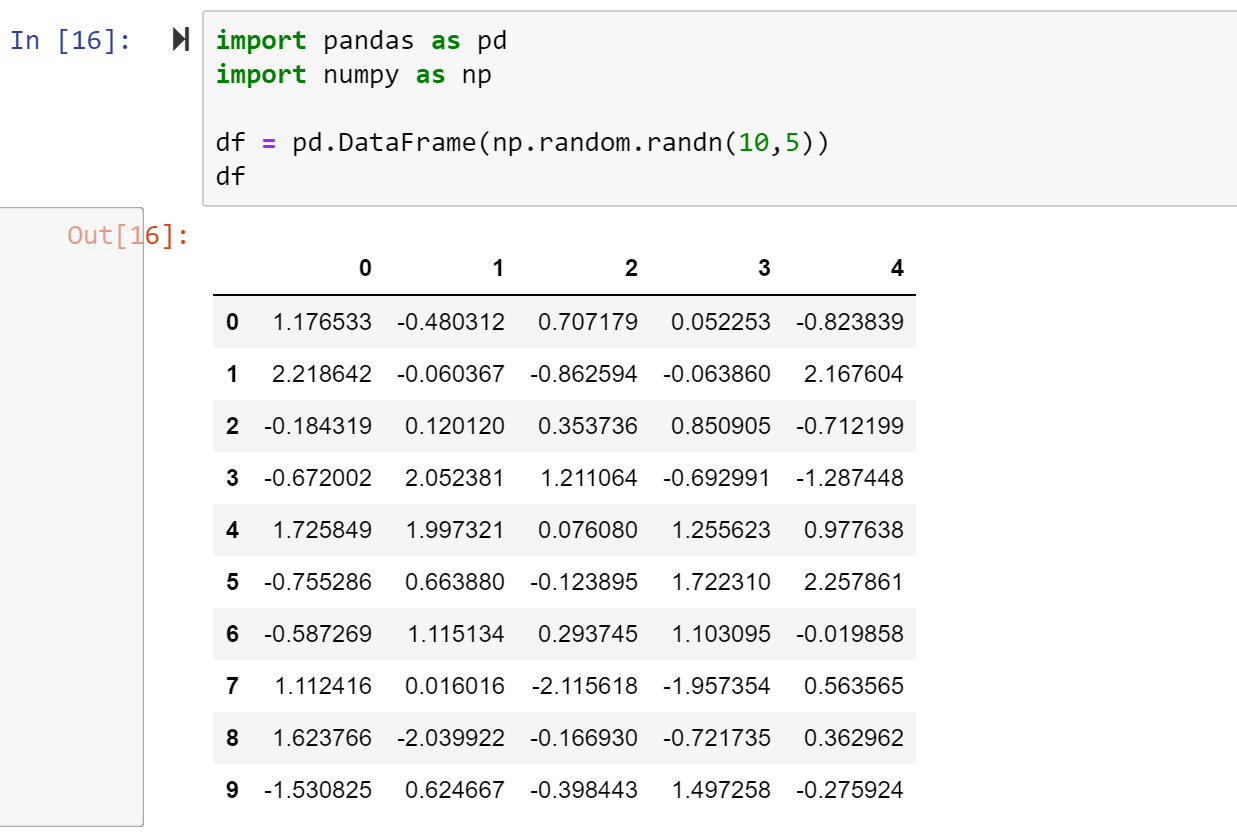


**Started with the name = ‘John’ ran the name command, then changed name = ‘Corey’ and went back and ran name again and now it returns ‘Corey’ because that is what it is currently assigned to.**

**Able to run certain commands that tie back to the file you are connected to, for example: run the %pwd to find the directory the jupyter notebook is attached to**



**Your are able to import different python scripts as well:**

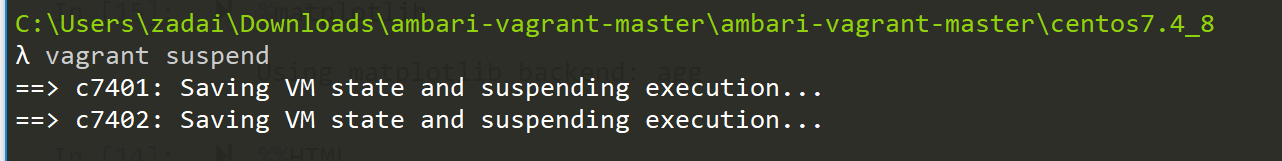


**The Jupyter notebook is a good way to explore your data and a great resource tool as well within their website.**

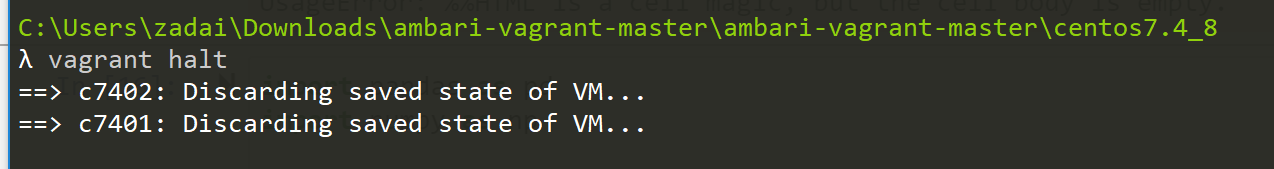
## 1.6 Suspend Cluster

Exit now from the ssh (shell that you’ve been using) to return to the Cmder prompt

**vagrant suspend**



**As an extra measure I like to run the vagrant halt command to my VM’s as well.**



## Reflection

Going over Jupyter notebook’s in Lab 5 was very cool to see. I’ve used Jupyter notebooks on my own in my past to help learn how to code Python but being able to use an Ambari server to create and use a Jupyter notebook was something I’ve never seen or experienced before this lab. I also found the tutorial video informative and it gave me some ideas of stuff to try outside of class.