High Performance Computing Introduction

DS730

In this activity, you will be getting familiar with many of the tools we will use in this course. We will be using a preconfigured Linux machine in the cloud and/or locally that has all of the software we need on it. In order to do some true high performance computing, we will be using a cloud service. Your organization may not have a high performance computing cluster available and may not be willing to spend tens/hundreds of thousands of dollars setting one up. As an added benefit of taking this course, throughout the activities, you will learn how to start up a large cluster of machines, use it to analyze your data and then shut it down when you are done. This solution allows you to only pay for the computing power that you need without paying the huge upfront cost of setting up a cluster.

Many of the activities are long. However, we are assuming that you have no experience with the command line, Linux, SSH keys, Hadoop, Spark, Hortonworks, etc. Many of the commands are explained in detail and there are a lot of pictures. If you are more experienced, please don’t accidentally skip a step or command. Many of the tasks need to be done in order and exactly as written or the entire setup may not work. Because of that, you are encouraged to start task 3 and the optional tasks only if you know you’ll be able to finish them. To give you some time estimate, a person experienced with Linux, SSH keys and the command line can finish task 3 or the optional tasks in 10-15 minutes. I’ve been told that someone with no experience might need 90-120 minutes to finish each of those tasks.

Important:
You are welcome to use newer versions of the software but *you do so at your own risk*. At the time of this writing, the software shown is the most appropriate software available and everything has been tested with this setup. It’s possible a newer version comes out between the modification of this document and when you are taking this course. It’s likely that everything written here will apply to future updates. However, if something goes wrong with newer software, you will be on your own for troubleshooting it.

If you already have access to a cluster/machine with Hadoop/Pig/Hive/Spark/Kafka/Storm installed on it, you can skip tasks 1-3 and the optional tasks. The goal of tasks 1-3 and the 2 optional tasks is to spin up a virtual machine with the Hortonworks sandbox installed on it. **If you are using a cloud solution, always keep a backup of your code locally on your desktop/laptop just in case there are issues**. Here are the benefits and disadvantages of each:

Optional Task 1 - A cloud solution that allows you to connect to Hortonworks anywhere you have the Internet. Allows you to create a machine with several cores and enough memory to run Hortonworks. A disadvantage is that it is cloud based and therefore you are at the mercy of a 3rd party and the Internet. If Microsoft is lagging when you want to work, it is annoying having to wait long periods for things to start up. Please don't let the negatives dissuade you from trying out Microsoft. It works for >90% of people with no problems. Another disadvantage may be the credits. Azure provides $100 in credits for students per year. If you have used your credits in another course or for some other reason, you have no way of getting more credits. Microsoft has been quite strict about giving out more credits if you use up your allocated amount. We cannot give out more credits on Azure and we have yet to hear of a case where Microsoft has given extra credits to someone. You are also on your own with setting up an Azure account. Step-by-step instructions are provided for creating an account and they work well in the vast majority of cases. However, they don't work in about 5% of cases[1]. We do not control any accounts with respect to Microsoft/Azure. Therefore, if your account is locked or you aren't able to get access or your credits aren't showing up, Microsoft support is your only solution[2]. If you use Azure, be very careful to shutdown your virtual machines when you are done using them. The credits are enough to provide 30+ hours of working time per week. But if you forget to shut down your machine, you can burn through credits in a few weeks.

Optional Task 2 - A local Hortonworks solution[3]. You install Hortonworks locally and therefore can use it at any time without being connected to the Internet. Your system will run as fast as your machine will allow. It is recommended that you have about 60GB of free space available and 8GB of memory available (not total memory, but available

[1] This is usually true if you have signed up with or used Azure in the past. If you have exhausted your credits or have used your free trial, there is nothing we can do to reinstate those things.

[2] Microsoft support may claim this is a school issue and ask you to contact your school for help. Unfortunately, this is not us (DS730 instructors). You could try contacting the helpdesk on your local campus if Microsoft points you to back to your campus. The bottom line, if there are account issues with Microsoft, we have no power to help.

[3] Please note that installing Hortonworks locally is only a good idea if you are comfortable with virtual machines and setting them up. The instructors do not have the ability to troubleshoot a broken/failed local install because it is impossible to know what configuration you have and how to fix it.

memory). The disadvantage is being tied to 1 machine. Your machine may be a work machine and may be locked down such that you cannot install software on your machine. Some installs have required minor changes to the BIOS in order to get VirtualBox to work. If your virtual machine breaks or gets into a bad state, you will have to troubleshoot the problem as we cannot connect to your machine to see what's going on.

Task 1 - **We strongly recommend doing this option.** A cloud solution that works on Amazon. Similar advantages and disadvantages to Optional Task 1. The biggest advantage here is that we control the credits and we can easily connect to your machine if there is a problem. If you run out of credits, we have the ability to give you another account such that you can create a new machine and start over. Note that we cannot give your current account more credits. If you run out of credits, you'll have to start over with a brand new account and a brand new virtual machine.

Books & Other Software

Because of the fast paced nature of this area, there is no required book for this course. However, many of the activities are quite long and detailed so one could consider the set of activities the book for this course. The advantage of this kind of "book" is that it can be updated quickly if there is an update that needs to be made. If you come across a portion of text that no longer makes sense or see an outdated image, let your instructor know right away so we can update the document.

One of the most common questions asked in this course is why we don't learn tool XYZ where XYZ is the *best* thing we can learn in this course. As the world of high performance computing changes, the goal of this course is to keep up with the newest technologies and present you the newest versions of each software we use. At the same time, we want to teach the core concepts of these technologies. We do not want you to be pigeonholed into using a specific software tool to solve a problem. If you learn the core concepts of several paradigms, then you should be able to pick up whatever the software-du-jour is when you need it.

We have tried to make entering in commands as painless as possible. Therefore, unless otherwise noted, every command in every activity should be able to be copied and pasted. This also reduces manually copying issues like figuring out the difference between 0 and O and the difference between 1, I and l ← this one is a lowercase L. If you come across a command that does not copy and paste correctly, please inform an instructor so we can fix the issue.

Hortonworks Sandbox

It is important to understand what is going on inside the sandbox. The sandbox is essentially simulating a cluster of machines that run Hadoop, Pig, Hive and other pieces of software. You do not have to install each piece individually. Rather, you can use the software straight out of the box without having to deal with the administrative part of it. If you want to know more about how to install some of the software, see the optional activities for how to setup a Linux machine, install Hadoop, Pig, Hive, etc. **Note that the load times of Hortonworks are generally at least 5 minutes and sometimes closer to 20 minutes.** Do not expect to start up your virtual machine and start working immediately. It takes a while to start all of this software.

In this course, we will focus on two parts of the Hadoop environment but there are many other parts and pieces of software you can learn about.

The first part is the Hadoop Distributed File System (HDFS). HDFS allows for scalable and reliable data storage. It is designed to run on many “cheap” computers. HDFS stores data in large chunks and each chunk is stored on multiple machines (in general) to provide reliable access to the data. For example, consider chunk X is stored on machines A, B and C. Assume you start a job that needs to use chunk X. If A and B are busy doing other work, you can run your job on chunk X on machine C. This runs into the classic space-vs-time problem. You can replicate your data as much as you want across numerous machines. If you replicate a lot, your total storage capacity will go down but your processing time will also go down. If you replicate a little, your total storage capacity will go up but your processing time will also go up. We won’t spend much time looking into the innards of HDFS. A general knowledge that HDFS is distributed storage is sufficient for this course.

One potential confusing part about the sandbox is that you will have 3 filesystems. If you are having trouble remembering what you are connecting to, come back to this part. You can likely skim over this part for now as most of this will not make sense until you are finished with the activity. If you create a file on one filesystem, it will not be accessible on another filesystem[4]. You need to copy/move files from one filesystem to the other in order to use them. There will be ways to copy/move files from filesystem to filesystem shown in the activities. However, there are not always direct connections that can be easily made. You may have to copy/move files from filesystem A to filesystem B

[4] It’s the same concept of having two different machines. If I create a file on my machine, you can’t access it on your machine. In order for you to access the file, I need to send the file to you.

and then from filesystem B to filesystem C in order to get from filesystem A to filesystem C. Here are the filesystems you'll be dealing with:

1. **Linux filesystem** - this is the filesystem that you connect to if you are using port 22 with CyberDuck or PuTTy. After this activity, you shouldn't have to connect to this filesystem.
2. **Hortonworks filesystem** - this is the filesystem that you see what you connect to port 4200 via the browser. You can also connect to this filesystem directly with PuTTy or CyberDuck using port 2222.
3. **HDFS** - the hadoop distributed filesystem. This is the filesystem that you see when you go to the Files View in Ambari. You are able to put files here and download files using the Files View.
4. **Your laptop/desktop's filesystem** - Although not part of the Hadoop ecosystem, your local file system is important too. When you are uploading files to canvas, those files will need to be on your local filesystem before you submit them.

The main part we will concern ourselves with is how to access and manipulate the data. In this course, we will use MapReduce, Pig (or Kafka), Hive and Spark. As of now, you have the option to "open up the black box" a little bit and see how Hadoop actually works. How does Hadoop distribute processing across multiple threads on a single machine? The activity/project associated with threading will show how this is done. If you are not interested in opening up the black box, there is an alternative option learning streaming tools instead. The threading alternative introduces you to other big data software: Storm and Redis. There are many other software titles available inside Hortonworks and if there is enough interest, optional activities can be created. The final project also gives you an opportunity to learn a brand new big data tool and create an activity for it.

Activity Tasks

Task 1: Get Registered with Amazon Web Services

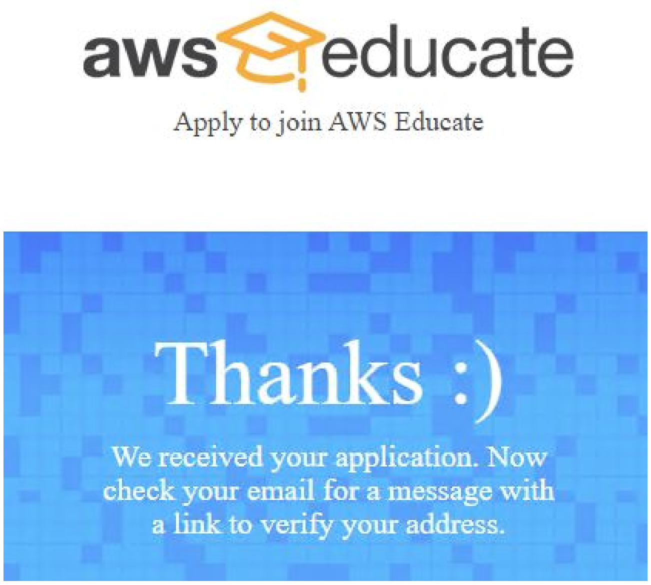
Using the Hortonworks sandbox is a nice way to simulate running your code on many machines. The reality is, your "cluster" is simply 1 machine that is running 4 CPUs and has 16GB of memory. Using Hortonworks on 1 machine is a nice way to test your code. If you want to do true high performance computing to tackle terabytes (or higher) of data, you would want to use a larger cluster. Amazon Web Services (AWS) allows you to spin up as large of a cluster as you need and shut it down whenever you are finished with it. Please note that this task cannot be accomplished until the semester begins[5]. The instructor is not provided with a list of names/emails until that time. If you start this activity during preview week, you will not be able to do tasks 1-3 until the actual first day of the semester. **Do not go to awseducate and create an account on your own. It may cause problems in the future with accounts.** Wait until you receive the email from step 1:

1. You will receive an email from Amazon[6]. The email I received was from AWS Educate Support at support@awseducate.com. Inside the email, there is a sentence like this: *Click here to complete the AWS Educate application process…* Simply click on the "here" link.
2. The link took me to step 2 of 3 of the AWS Educate sign up. If you see step 1, something went wrong and you should stop and inform an instructor. You should never see step 1 following these directions. Step 2 simply has you filling in your details. Put your home campus for the university[7]. If there is a *Promo Code* field, you can leave it blank. Once everything is filled in, click **Next**.
3. Read through the terms and conditions and assuming you accept them, click on the **I Agree** checkbox. Click **Submit**. You will likely be directed to a page that looks like this:

[5] I received the email list early this semester. If you were registered by January 19, your email was uploaded to AWS for an account on January 19. Please check all of your emails and spam folders for the AWS email. If you register for the course on or after January 19, your AWS account will be set up on January 26.

[6] Students at UW-Eau Claire may have the AWS email in quarantine. Since I am at UW-Oshkosh, I do not have an Eau Claire account and I cannot reproduce the error. Just be aware that if you don't see your AWS email, you may have to check your spam folder or release those emails from quarantine (https://www.uwec.edu/kb/article/managed-spam/).

[7] I entered in: **University of Wisconsin - Oshkosh** and had no problems. It wasn't one of the prefilled options but it worked just fine. If you replace Oshkosh with your university's city, you should have no issues.

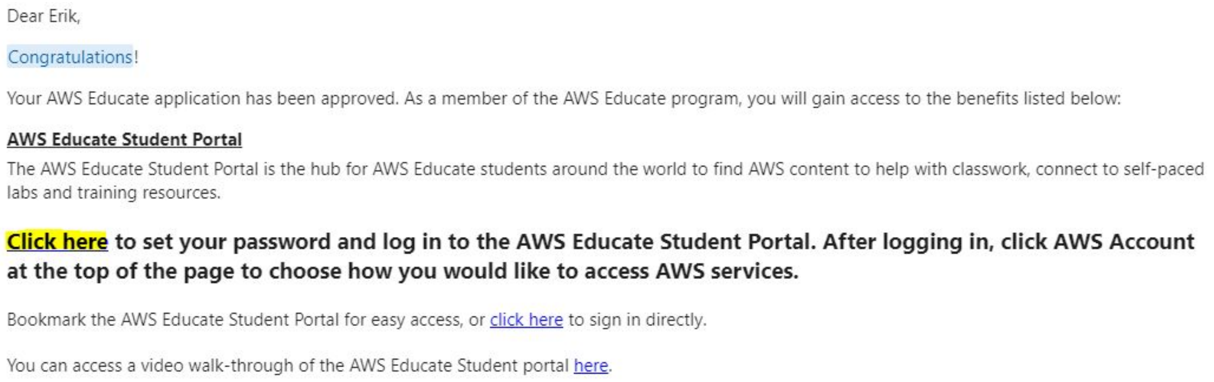


Apply to join AWS Educate

Thanks :)

We received your application. Now check your email for a message with a link to verify your address.

4. Within seconds, I received an email asking me to verify my email address. I clicked on the verification link and it took me to a page saying that my email has been verified.
5. A few minutes later, I received an email telling me that my application was approved. It is possible that your application will remain under review. If this happens, your application will likely be approved in a few hours. Inside that email there is a link to set up your password and login. Click on the “Click here” link to create a password:



Dear Erik,

Congratulations!

Your AWS Educate application has been approved. As a member of the AWS Educate program, you will gain access to the benefits listed below:

AWS Educate Student Portal

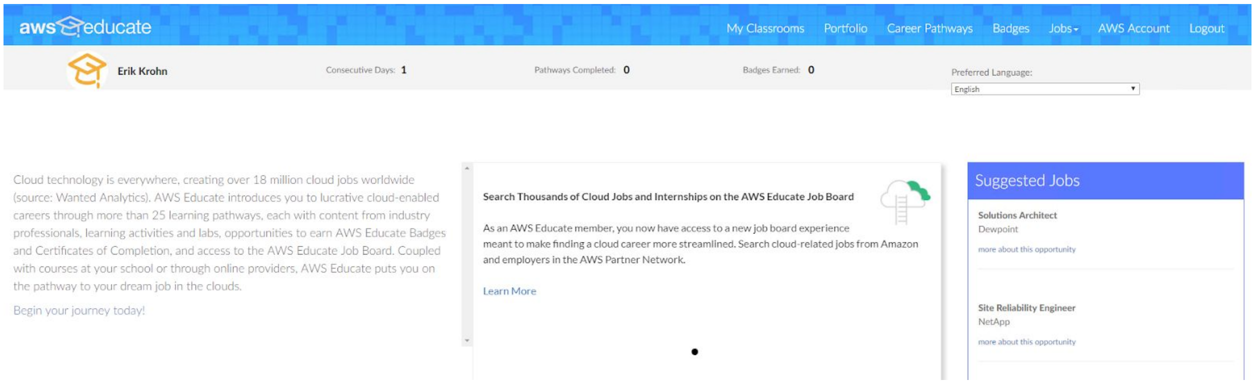
The AWS Educate Student Portal is the hub for AWS Educate students around the world to find AWS content to help with classwork, connect to self-paced labs and training resources.

Click here to set your password and log in to the AWS Educate Student Portal. After logging in, click AWS Account at the top of the page to choose how you would like to access AWS services.

Bookmark the AWS Educate Student Portal for easy access, or click here to sign in directly.

You can access a video walk-through of the AWS Educate Student portal here.

6. Once you have created a password, you should see something like this:



Cloud technology is everywhere, creating over 18 million cloud jobs worldwide (source: Wanted Analytics). AWS Educate introduces you to lucrative cloud-enabled careers through more than 25 learning pathways, each with content from industry professionals, learning activities and labs, opportunities to earn AWS Educate Badges and Certificates of Completion, and access to the AWS Educate Job Board. Coupled with courses at your school or through online providers, AWS Educate puts you on the pathway to your dream job in the clouds.

Begin your journey today!

Search Thousands of Cloud Jobs and Internships on the AWS Educate Job Board

As an AWS Educate member, you now have access to a new job board experience meant to make finding a cloud career more streamlined. Search cloud-related jobs from Amazon and employers in the AWS Partner Network.

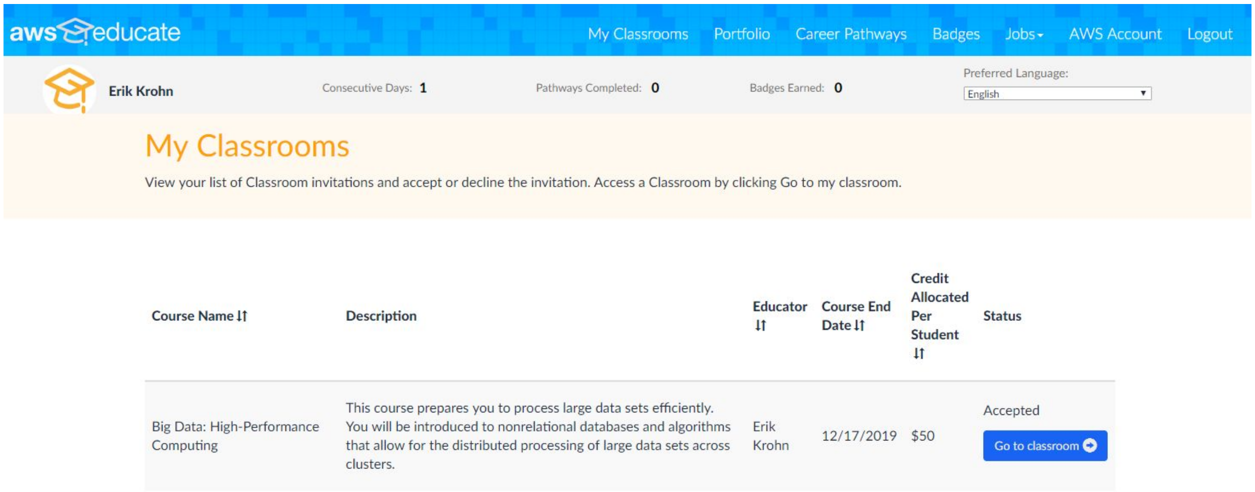
Learn More

Suggested Jobs

Solutions Architect
Dewpoint
more about this opportunity

Site Reliability Engineer
NetApp
more about this opportunity

7. Click on the **My Classrooms** link at the top. This redirected me to a page that looks like this:

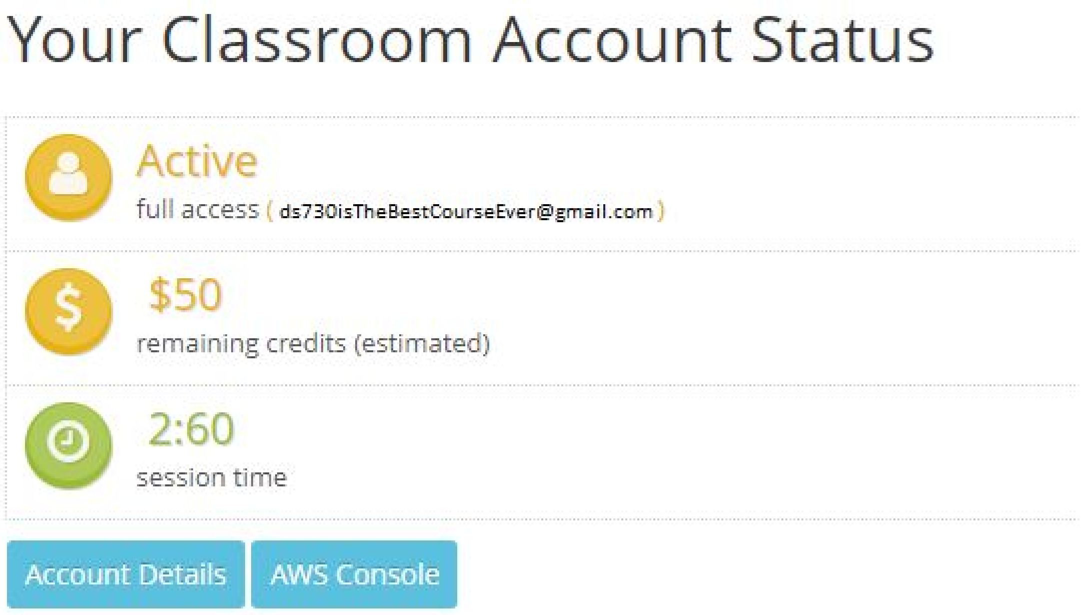


My Classrooms

View your list of Classroom invitations and accept or decline the invitation. Access a Classroom by clicking Go to my classroom.

Course Name ⇅	Description	Educator ⇅	Course End Date ⇅	Credit Allocated Per Student ⇅	Status
Big Data: High-Performance Computing	This course prepares you to process large data sets efficiently. You will be introduced to nonrelational databases and algorithms that allow for the distributed processing of large data sets across clusters.	Erik Krohn	12/17/2019	$50	Accepted Go to classroom

8. Click on the **Go to classroom** link. A pop-up will ask you to confirm you want to go to a new site, read that and click **Continue**. You will be redirected to a terms and conditions page. Read those and assuming you agree, click **I Agree** at the bottom.
9. Click on the **AWS Console** button on the right hand side:



Your Classroom Account Status

Active

full access (ds730isTheBestCourseEver@gmail.com)

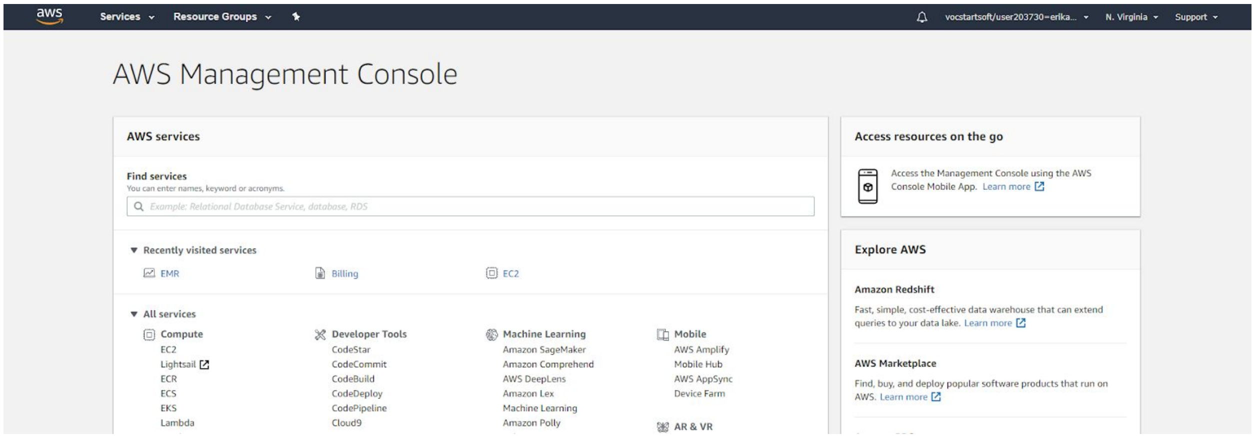
$50

remaining credits (estimated)

2:60

session time

10. The page was a pop-up that my pop-up blocker blocked. Once I allowed the page to load up, this is what I saw:



AWS Management Console

AWS services

Find services

You can enter names, keyword or acronyms.

Example: Relational Database Service, database, RDS

Access resources on the go

Access the Management Console using the AWS Console Mobile App. Learn more

Explore AWS

Amazon Redshift

Fast, simple, cost-effective data warehouse that can extend queries to your data lake. Learn more

AWS Marketplace

Find, buy, and deploy popular software products that run on AWS. Learn more

11. Once you have reached this point, you are done and everything is set up. This is the AWS Console. In order to get back to the AWS Console, you'll have to follow these steps:
    a. Go to https://aws.amazon.com/education/awseducate/ and click on the **Login to AWS Educate** link.
    b. Sign in with your email address and the password you just created.
    c. Repeat steps 7-10 again to get back to your AWS Console.

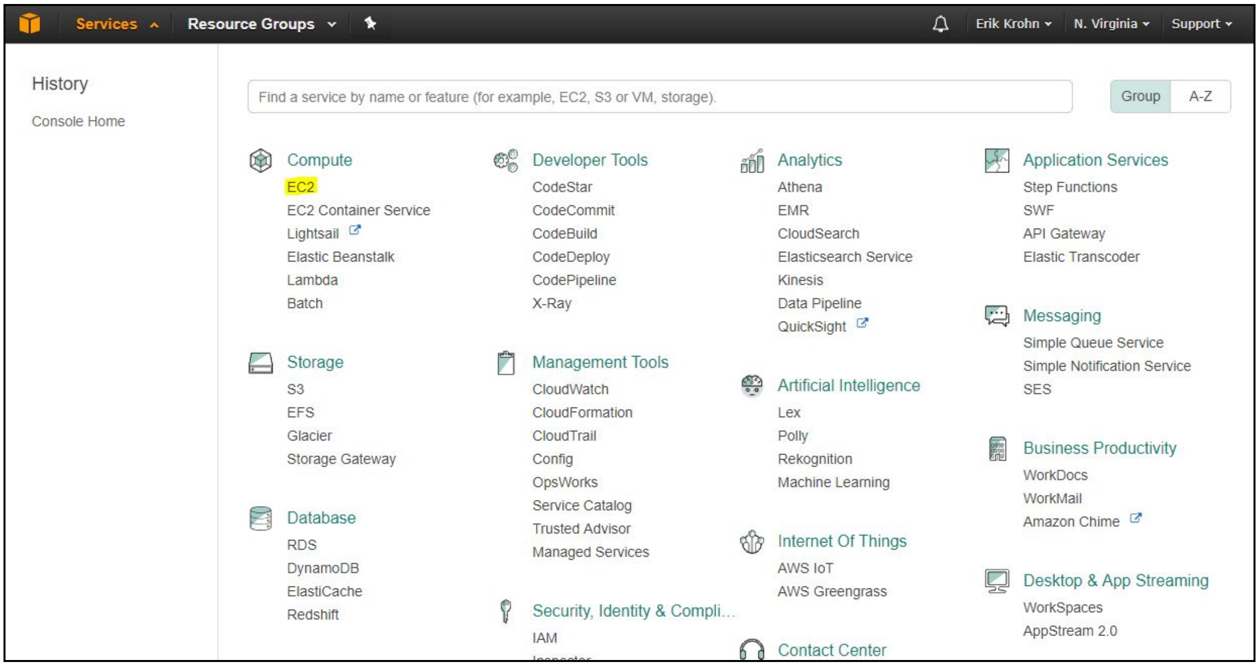
Task 2: Create a Linux Machine on AWS

Connecting to a machine in the cloud is a good start for the cloud based computing we will be using in future activities and projects. We will not connect many times to our Linux filesystem but it is something you should know how to do. Whenever you spin up a cluster on AWS or some other cloud service, you should be aware of how to connect to the cluster to make any tweaks you want.

1. If you use Windows, go here: http://faculty.cs.uwosh.edu/faculty/krohn/ds730/putty.html. If you have a Mac, skip to step 3 as your ssh software is built into your terminal program. If you are using Linux, odds are you don't need instructions on how to connect to a remote server.
2. Download these two files:
   - **putty.exe**
   - **puttygen.exe**

Note:
You can download the Windows installer to install more than you need. However, downloading what is specified above is sufficient. You do not need to install any software to use PuTTy. It is a standalone program that should just run. If you cannot get the software to run locally, you can download PuTTy and PuTTygen and run it on the virtual lab.

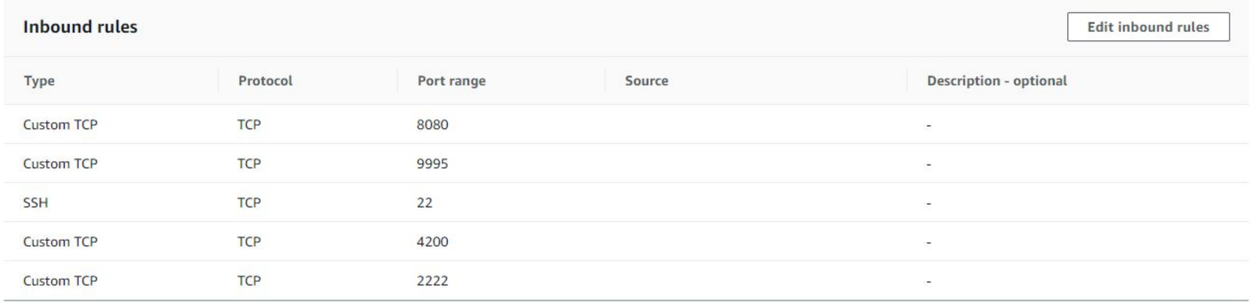
3. Sign into your AWS console.
4. Click **Services**.
5. In the **Compute** section, click **EC2**:



6. Click the **Launch Instance** button.

> **Note:**
> As a comment for the future, if you are looking to save some money, you should look into Spot Requests (Instances). Everything explained in this course is with *On Demand* instances. Basically, when you start up an *On Demand* server, it will run continuously without being interrupted.

7. **Select** the Ubuntu Server… option. As of now, the best option is the 20.04 version.
8. Choose **General Purpose t2.xlarge**. To ensure you don't waste your credits, make sure to stop your instance when you are done using it (details provided later in the task). You can find all of the pricing details here: https://aws.amazon.com/ec2/pricing/.
9. Click **Next: Configure Instance Details**.
10. All of the defaults are good here. click **Next: Add Storage**.
11. Change the GB size from 8GB to 50GB. This can be increased later if you need more space.
12. Click **Next: Add Tags**.
13. The default values on this page can be kept as they are. Click **Next: Configure Security Group**.
14. Make sure **Create a new security** group is checked. Add the following rules to your Security Settings:



Inbound rules

Type	Protocol	Port range	Source	Description - optional
Custom TCP	TCP	8080		-
Custom TCP	TCP	9995		-
SSH	TCP	22		-
Custom TCP	TCP	4200		-
Custom TCP	TCP	2222		-

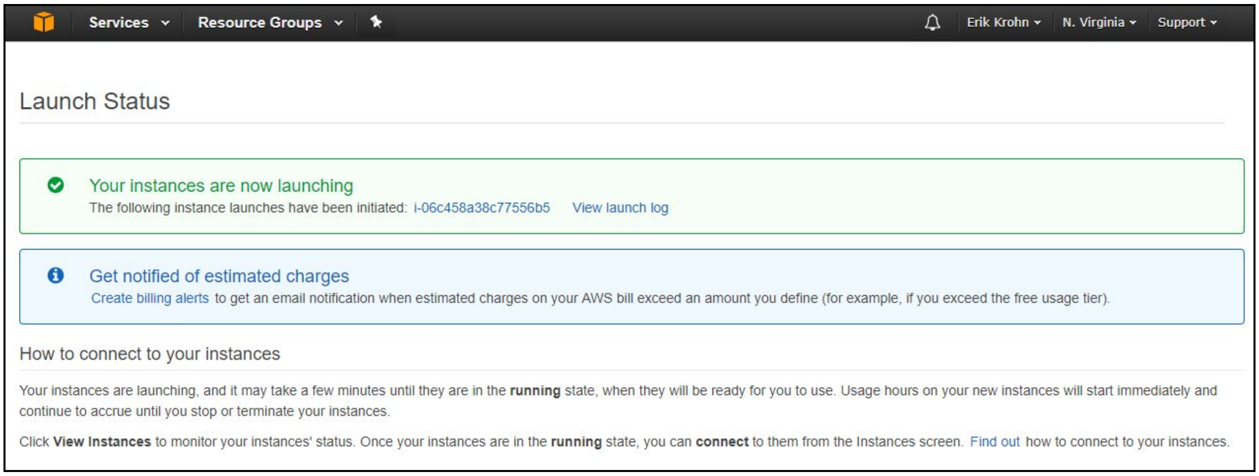
15. The Type for one of the rules should be **SSH** and the others are **Custom TCP Rule**.
16. The Protocol for all rules should be **TCP**.
17. The Port Ranges should be **22, 2222, 4200, 8080** and **9995**.
18. Change the **Source(s)** from **Custom** to **My IP**. The grayed out box next to My IP will display your IP address. This assumes you will be connecting to the instance from your current machine. If you will be connecting from multiple machines, you will need to add new rules for those machines as well. You should note that if you are working from home, it's unlikely your internet provider has given you a static IP address that you will always have. Your IP address may change from day to day, even hour to hour. If you are unable to connect to your EC2 instance in the future, revisit this step and update your security settings so that your current IP can access your EC2 instance[8].
19. Click **Review and Launch**.
20. Click the **Launch** button.
21. Click **Create a new key pair**.
22. Enter a **Key pair name** of whatever you want. I called mine **`UbuntuAWS`**
23. Click **Download Key Pair**.
24. Save that file somewhere to your hard drive and do not lose it![9]

Important:
If you lose your key, you will not be able to access the server.

25. Click **Launch Instances**.
26. Confirm you see something like this:

[8] If you do not know your IP address, you can visit https://www.whatismyip.com/ to find it.

[9] We also recommend storing the key on a flash drive just in case your computer crashes and you lose access to your data.



Launch Status

Your instances are now launching
The following instance launches have been initiated: i-06c458a38c77556b5 View launch log

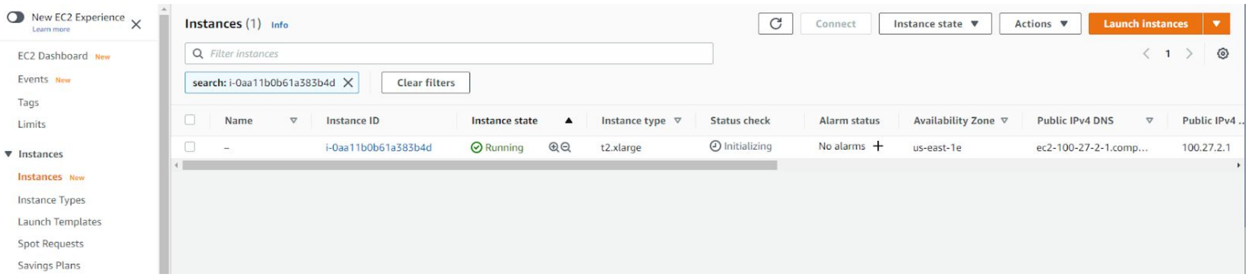
Get notified of estimated charges
Create billing alerts to get an email notification when estimated charges on your AWS bill exceed an amount you define (for example, if you exceed the free usage tier).

How to connect to your instances

Your instances are launching, and it may take a few minutes until they are in the **running** state, when they will be ready for you to use. Usage hours on your new instances will start immediately and continue to accrue until you stop or terminate your instances.

Click **View Instances** to monitor your instances' status. Once your instances are in the **running** state, you can **connect** to them from the Instances screen. Find out how to connect to your instances.

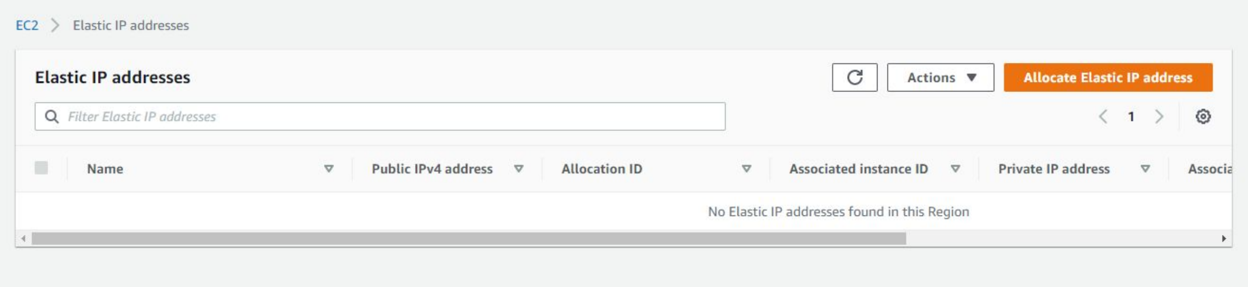
27. Click the link that appears after **The following instance launches have been initiated**.
    - In the previous screen shot, the link appears as **i-06c458a38c77556b5**. Yours will have a different name.
28. Once you click that link, confirm you see something similar to this:



Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ..
–	i-0aa11b0b61a383b4d	Running	t2.xlarge	Initializing	No alarms	us-east-1e	ec2-100-27-2-1.comp...	100.27.2.1

If you return to this page, it may seem that your instance has disappeared. It is likely because of a filter that is in place. Click on the **Clear Filters** button to clear any filters. Clearing the filters will display all of your EC2 instances.

29. In the **Instance State** column, notice it says **running**. If it is still initializing, pending or waiting, then wait for the instance state to be in running mode before continuing. Click on the **Elastic IPs** link on the left hand side. You should see something like this:

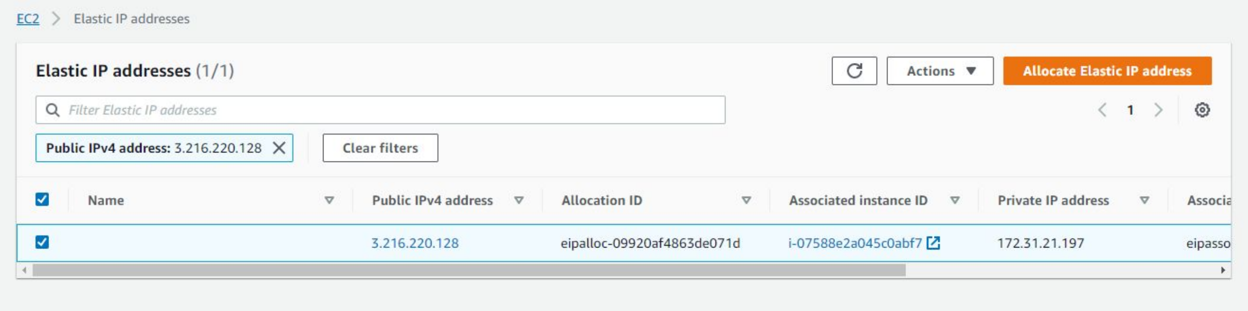


Name	Public IPv4 address	Allocation ID	Associated instance ID	Private IP address	Associa

No Elastic IP addresses found in this Region

- Click on the **Allocate Elastic IP address** button.

- The default option chosen is fine. Click on the **Allocate** button.
- You will now notice an IP address that is assigned to you:



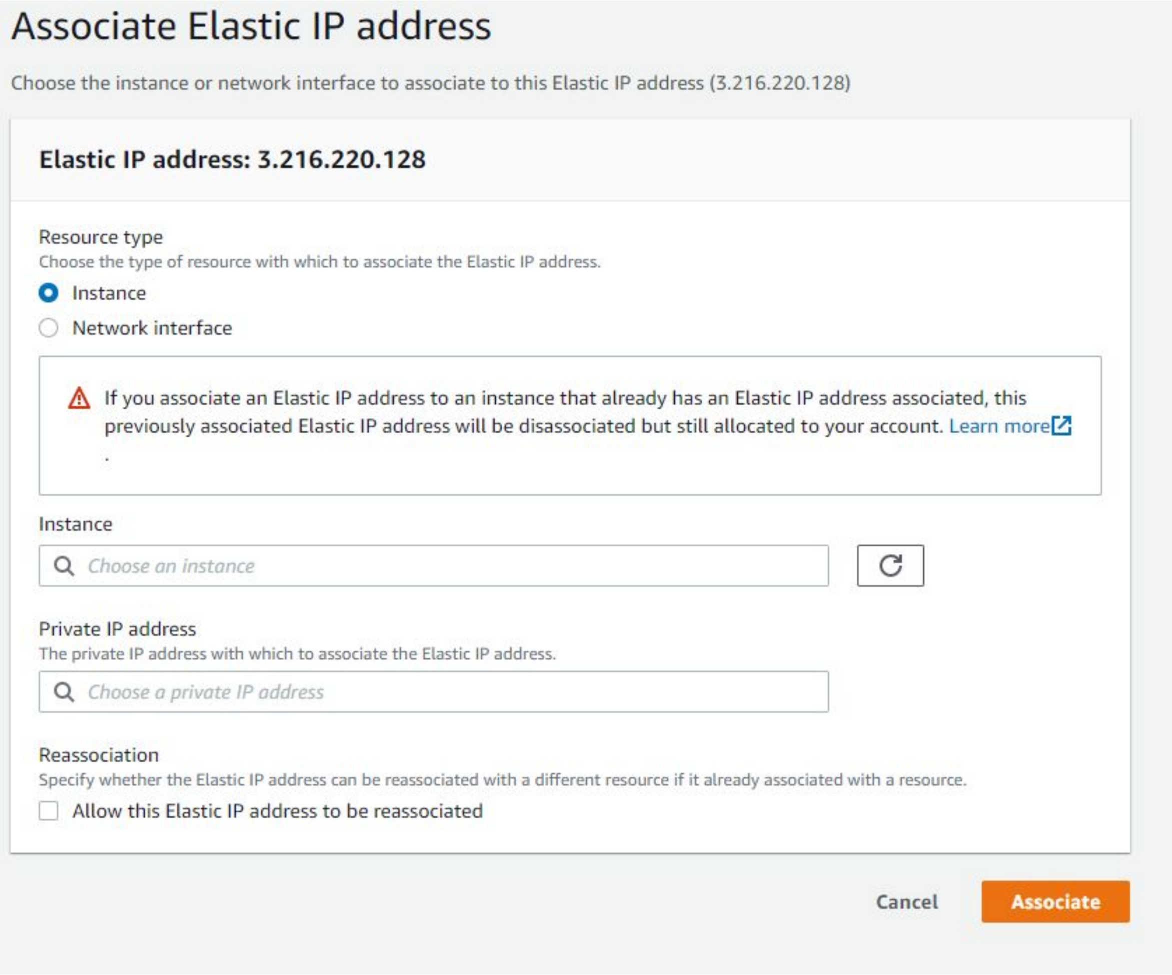
EC2 > Elastic IP addresses

Elastic IP addresses (1/1)

Public IPv4 address: 3.216.220.128 ✕

☑	Name	Public IPv4 address	Allocation ID	Associated instance ID	Private IP address	Associa
☑		3.216.220.128	eipalloc-09920af4863de071d	i-07588e2a045c0abf7	172.31.21.197	eipasso

- **This will be your IP address of your virtual machine for the remainder of the semester.**
- Make sure your IP is selected and click on the **Actions** dropdown box. Click on **Associate Elastic IP address**. You should see something like this:



Associate Elastic IP address

Choose the instance or network interface to associate to this Elastic IP address (3.216.220.128)

Elastic IP address: 3.216.220.128

Resource type

Choose the type of resource with which to associate the Elastic IP address.

● Instance

○ Network interface

⚠ If you associate an Elastic IP address to an instance that already has an Elastic IP address associated, this previously associated Elastic IP address will be disassociated but still allocated to your account. Learn more .

Instance

Choose an instance

Private IP address

The private IP address with which to associate the Elastic IP address.

Choose a private IP address

Reassociation

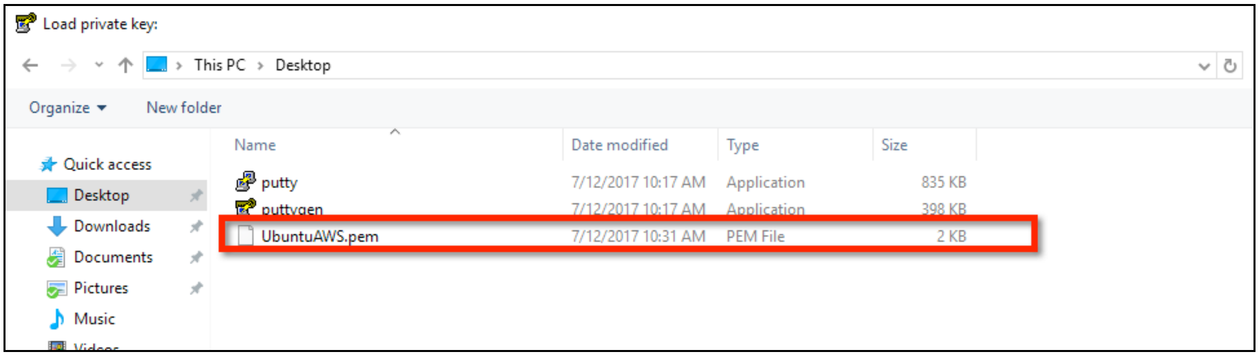
Specify whether the Elastic IP address can be reassociated with a different resource if it already associated with a resource.

□ Allow this Elastic IP address to be reassociated

- Inside the **Instance** box, click on it and you should only have one option. It should be your running instance. Select that instance.

- Click **Associate**. This takes us back to our Elastic IP main page.

30. Write down your IP address as this will be your IP address for the remainder of the semester. From the images, my IP address was 3.216.220.128. Yours will be different.
31. If you are using a Windows machine, use the following steps to connect to your EC2 instance (steps 32-48). If you are using a Mac or a Linux machine, simply open up your Terminal window and use the following instructions to connect and then skip to step 49. In the instructions, be sure to replace ec2-user with ubuntu. In other words, the command should be this:
    ```
    ssh -i /path/my-key-pair.pem ubuntu@public_dns_name
    ```
    https://docs.aws.amazon.com/quickstarts/latest/vmlaunch/step-2-connect-to-instance.html#sshclient
32. The .pem file downloaded in Step 24 is not compatible with PuTTY. Because of this, we need to create a key file that PuTTY can read. Open up the **puttygen.exe** file that you downloaded in Step 2.
33. Click the **Load** button.
34. By default, PuTTYgen only looks for files with extensions of ppk so you must change it to look for all files. Find your .pem file that you downloaded in Step 24, which will look something like this:

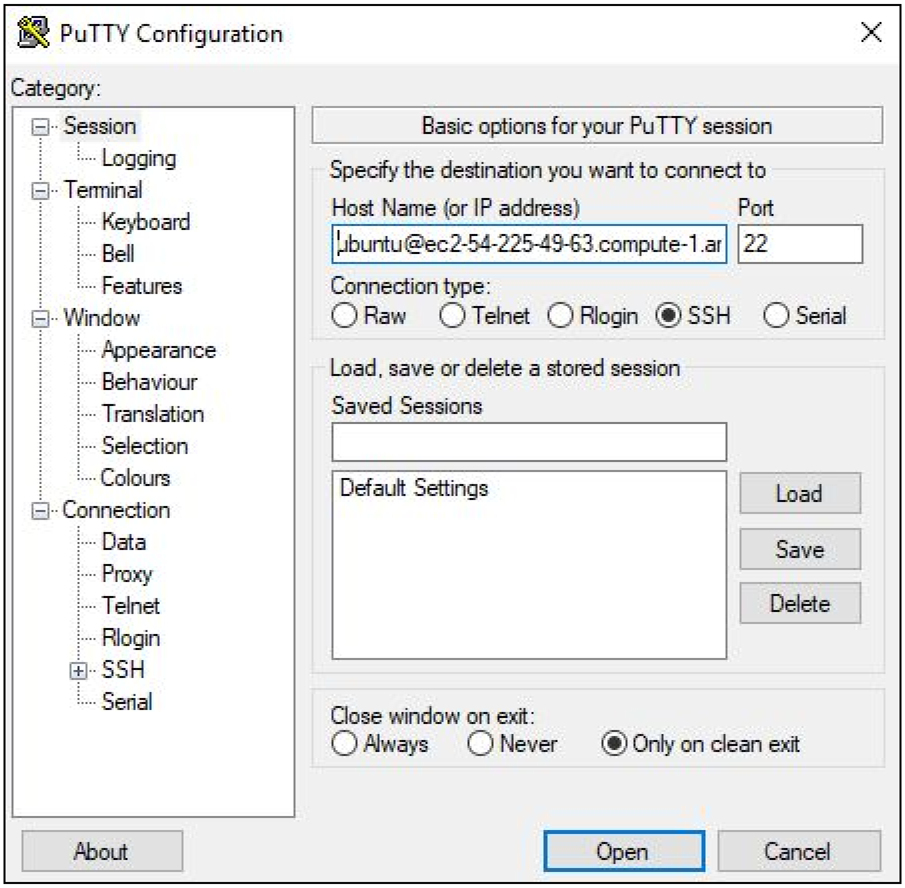


Name	Date modified	Type	Size
putty	7/12/2017 10:17 AM	Application	835 KB
puttygen	7/12/2017 10:17 AM	Application	398 KB
UbuntuAWS.pem	7/12/2017 10:31 AM	PEM File	2 KB

35. Click **open**.
36. PuTTYgen will give you some message about successfully importing the foreign key. Simply click **OK**.
37. In order to save the key that PuTTY can use, click the **Save private key** button.
38. The system will ask you if you want to save it without a passphrase. Click **yes**.
39. Save this **ppk** file somewhere safe. You can name it whatever you want.

Important:
If you lose this file, you will not be able to access your server.

40. Close out of PuTTygen.
41. We are now ready to connect to our server. Open **putty.exe**.
42. In the **Host Name** section, enter **`ubuntu@`** followed by the IP address you wrote down in step 30.
    - As an example, my **Host Name** was: ubuntu@3.216.220.128
43. Enter in a **Port** of 22. You should have something similar to this:



Specify the destination you want to connect to

Host Name (or IP address): ubuntu@ec2-54-225-49-63.compute-1.ar

Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

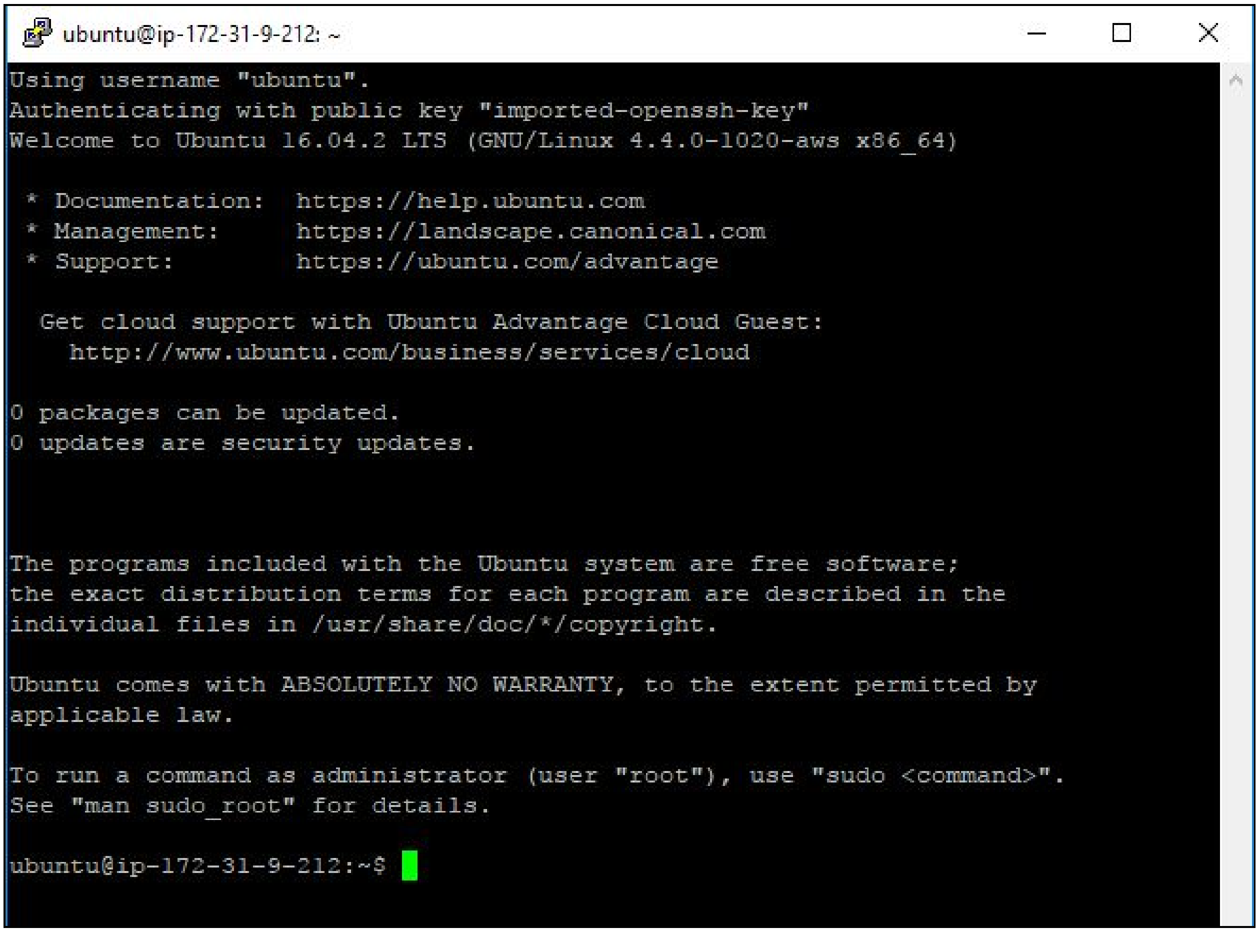
Load, save or delete a stored session

Saved Sessions

Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

44. On the left side, click the plus symbol next to **SSH**.
45. Click the **Auth** option.
46. Click the **Browse** button for the **Private key file for authentication**
47. Find your ppk file that you saved in Step 39 and click on **Open**.
48. Click the **Open** button.
49. When the system asks you about accepting an RSA fingerprint, indicate **Yes**.
50. Assuming everything went correctly, confirm you see something like this. As a reminder, what you are connecting on port 22 is your Linux filesystem.



```
Using username "ubuntu".
Authenticating with public key "imported-openssh-key"
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.4.0-1020-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  Get cloud support with Ubuntu Advantage Cloud Guest:
    http://www.ubuntu.com/business/services/cloud

0 packages can be updated.
0 updates are security updates.



The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-9-212:~$
```

For steps 51-60, all commands will be entered inside the PuTTy window.

51. We will install Java as we will need it later in the course. Before we can install it, we need to update our package list. In the PuTTy terminal window, type in[10]:

    `sudo apt-get update`

52. Make sure your software is up to date by entering:

    `sudo apt-get -y upgrade`

53. If necessary, enter '**Y**' to the upgrade.
54. If you get messages about updating some kind of grub file, choose the default option of **keep the local version currently installed**.

Using the Command Line

We will have to edit files on our Linux filesystems from time to time. If you have a favorite editor, feel free to use that. A simple one explained here is called Vim.

[10] You can also copy and paste the commands. To paste the command into PuTTy, simply right click anywhere in the PuTTy window.

55. To edit a file, enter
    vim nameOfFile
    This will open up a file called nameOfFile in a program called Vim, which you can think of as Notepad. Vim is not installed on Hortonworks by default. To use this in Hortonworks, instead of typing in vim, simply type in vi.
56. Press the letter **i** to enter Insert mode.
    Notice the word **-- INSERT --** on the bottom. Once you are in insert mode, you can type like you normally would.
57. Add the following text to the file:
    Hello, this is a test.
58. After you have added that text to your file, Press the ESC key.
    You should notice the -- INSERT -- disappear from the bottom.
59. Enter the following exactly as it's written:
    :wq
    That's a colon, then w, then q, then the <enter> or <return> key. This will save your file and take you back to the command prompt.
60. To ensure the file was created successfully, use the following command to display it on the screen: **cat nameOfFile**.

Note:
If you are struggling with Vim, search online for Vim tutorials such as http://www.openvim.com/ and read up on the commands. Once you've learned a few of the commands, it becomes very easy to use. Another common editor is called **nano** and instructions for how to use it are in the Linux presentation.

Test Connection and Stop Instance

Important:
You now need to go back to the AWS console page and stop your instance. You don't want to pay for idle time if you are not connected to your machine.

61. In order to stop your instance, go back to your browser and click your instance.
62. Go up to the **Instance State** button.
63. Click the **Stop instance** option.
64. You might see a box that says something about losing ephemeral storage. This is fine; click **Yes**. Wait on this page until the instance is stopped. You may need to refresh the page to see the instance stopped.
65. This is not a step that you should complete right now but it is being provided for future reference. If you want to change your instance type, this is the webpage where you do it (when the instance is stopped). If you so desire, you can use a

t2.2xlarge to better see the power of multithreading near the end of the semester. It's a CPU issue; smaller instances do not provide multiple CPUs. The t2.2xlarge instance provides 8 CPUs and 32GB of memory[11]. The t2.xlarge instance costs about 18 cents an hour. The t2.2xlarge instance, by contrast, is about 37 cents an hour. In order to change your instance type, ensure that your instance is stopped. Right click on the instance you want to change. Choose **Instance Settings** and choose **Change Instance Type**. Choose the instance type you want and hit **Apply**. Be very careful to choose an instance that is available to us with our starter accounts. You can view available options here. If you choose a type and that type is not allowed because of our starter accounts, your instance will terminate and you will lose your virtual machine.

66. At the end of the semester, your AWS account will be terminated. Anything that you've created and used on AWS will be gone and you cannot retrieve it. Therefore, you are encouraged to save any code/work to your local machine before the semester ends if you want to keep it.

Important:
During the semester, I recommended *stopping* as you will be able to restart your instance as it were without having to reconfigure anything. If you *terminate* it each time throughout the semester, you will lose everything and have to redo everything.

In order to restart your instance:

1. Come back to this page.
2. Click your instance.
3. Click **Instance state** and then **Start instance**.

Task 3: Create a Hortonworks Sandbox in the Cloud with AWS

This task creates a sandbox in the cloud with most of the required software installed on it. The main advantage of this is that you do not have to use 1 single machine for this course and can connect to your sandbox from anywhere. If there is a problem, your instructor can connect to your machine and troubleshoot any issues. As a side note, the virtual machine you are creating has nothing to do with the virtual lab that is provided by the program. Do not mistake what you are doing here with the virtual lab that you may have used in previous courses. You should not use the virtual lab at all in this course.

1. Go back to your EC2 instances page and you should have a stopped t2.xlarge instance (the instance you created in the last task). Click on that instance, go up

[11] The current t2.xlarge provides 4 CPUs and 16GB of RAM. It is more than sufficient for this course.

to **Instance State** and click on **Start instance**. Your instance will start up. Your IP address should be the same as it was in Task 2. As a reminder, my Elastic IP was 3.216.220.128. It should be the same. Connect to your EC2 instance using PuTTy and shown in the previous task. As a reminder, when you are connecting with PuTTy to your IP address using **port 22**, you are connecting to your **Linux filesystem**.

2. Once you are connected with PuTTy, enter the following commands. Note that step (c) will take some time as you are downloading and extracting many gigabytes of data:
    a. `wget http://faculty.cs.uwosh.edu/faculty/krohn/ds730/installAll.sh`
    b. **`chmod +x installAll.sh`**
    c. **`sudo ./installAll.sh`**
3. The previous script will take roughly 6 minutes to run. At the end of running the install script, it should say: **Hortonworks is running.** If you see *Hortonworks is not running*, then something went horribly wrong and you need to redo tasks 2 and 3[12]. In the future, you can come back and run this script to see if Hortonworks is running:

    ```
    ./checkHortonworks.sh
    ```

    You shouldn't have to do this again as the Hortonworks ecosystem will restart itself every time you restart the virtual machine.

    As a side note, it is quite common to start working on your work, walk away and forget to shutdown your virtual machine. As of this writing, for every hour you leave your machine on, it will cost you roughly 18 cents in credits. If you discover this the next day, it's generally not a problem. But if you walk away on a Sunday night and don't come back until Friday, you can easily burn through $15-$20 in credits. **With that said, there is a script that automatically shuts your virtual machine down after 5 hours.**

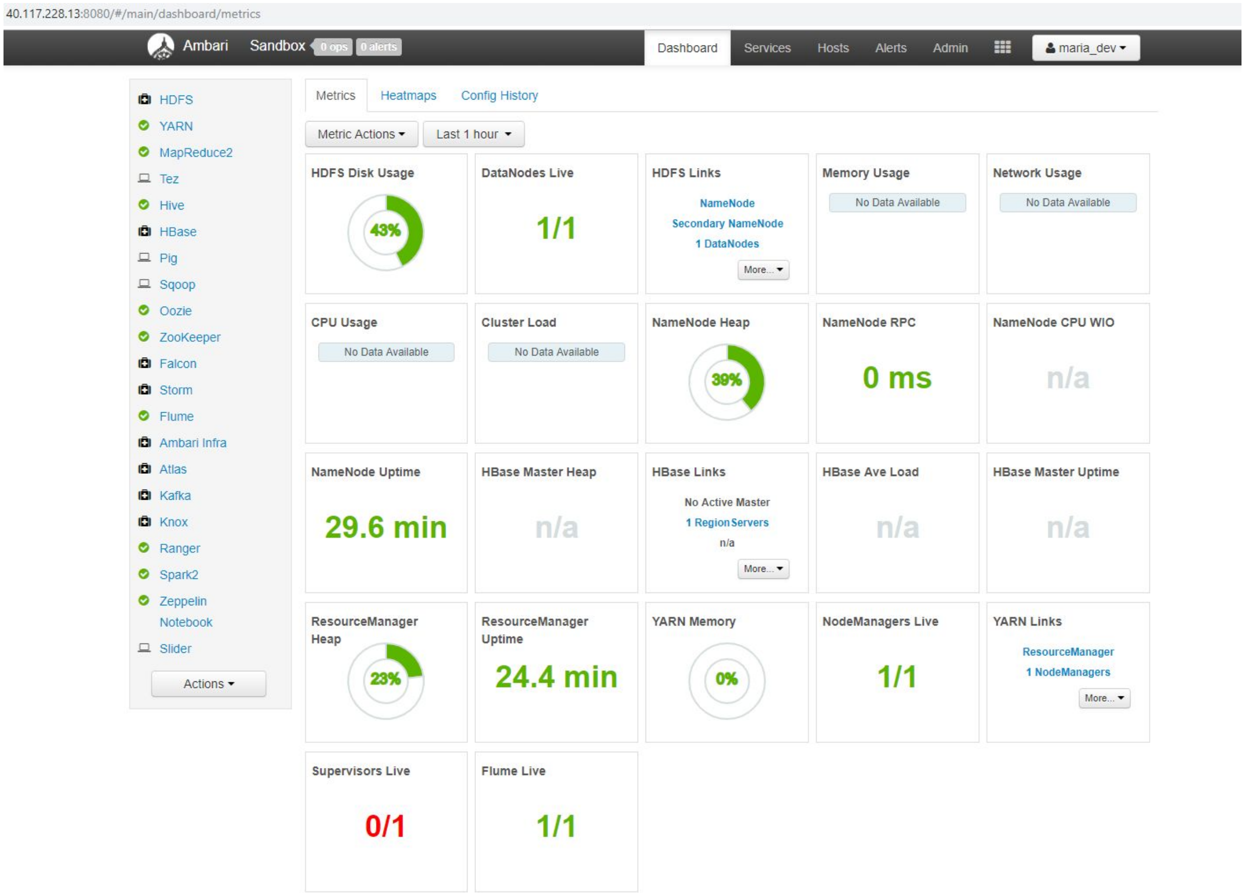
    If you wish to change the 5 hour default to a different amount, log into port 22 of your virtual machine and edit the following file using vim or your favorite editor: **end.sh**. You will notice a line that says: **sleep 298m** which means, wait for 298 minutes before doing anything. Once 298 minutes is over, the machine will start

[12] A script failure is very unlikely to happen. We have tested this script dozens of times and it has succeeded on each instance. Network issues, virtual machine issues are extremely rare (especially on AWS) but not impossible. Therefore, if one of these extremely rare cases occurs, it's best to just start over with task 2 and create a new virtual machine.

its shutdown sequence, which as written, will shutdown in an additional 2 minutes. You can edit the file using vim and change 298m to be whatever you want it to be, ~2 hours (120m), ~8 hours (480m), etc.

You should not rely on the automatic shutdown to stop your virtual machine though. If you are done working after a couple of hours, you should manually go back into AWS and stop your instance. There is no need to waste credits if you know you are done using your machine.

4. Use your browser to navigate to your IP using port 8080. You will go to http://IP:8080 and enter in **maria_dev** as the username and password[13]. You should see something like this:



40.117.228.13:8080/#/main/dashboard/metrics

HDFS Disk Usage: 43%
DataNodes Live: 1/1
HDFS Links: NameNode, Secondary NameNode, 1 DataNodes
Memory Usage: No Data Available
Network Usage: No Data Available
CPU Usage: No Data Available
Cluster Load: No Data Available
NameNode Heap: 39%
NameNode RPC: 0 ms
NameNode CPU WIO: n/a
NameNode Uptime: 29.6 min
HBase Master Heap: n/a
HBase Links: No Active Master, 1 RegionServers, n/a
HBase Ave Load: n/a
HBase Master Uptime: n/a
ResourceManager Heap: 23%
ResourceManager Uptime: 24.4 min
YARN Memory: 0%
NodeManagers Live: 1/1
YARN Links: ResourceManager, 1 NodeManagers
Supervisors Live: 0/1
Flume Live: 1/1

5. **This step is important for all future activities and projects whenever you start up your virtual machine.** You must check this page before beginning any activity/project to ensure the system is ready to use. As long as there are 0 red

[13] There may be a dialog box that pops up and says your password is exposed and you should change your password. Since your IP address is the only one that can access this webpage, you are safe leaving the username and password at the default.

alerts at the top, you are ready to start working. **If you see any red alerts, you must wait for the alerts to clear before starting.** Even though you may be able to load up the 8080 port and the 4200 port, it does not mean the machine is completely started and ready to use. **You must wait for all red alerts to clear before beginning.** Note that an orange alert is fine as this is just a warning. If you only have orange alerts, you can continue. You will likely have 1 orange alert that talks about disk usage. We are only allocating 50GB of space and we will end up using 60%+ of it. If you have red alerts, you must wait. The virtual machine does take a few minutes to load up all of the software. Feel free to explore the dashboard to see all of the tools available for you to use. Once you are done exploring, you can close out of this window. As stated above, be sure to wait until all red alerts are cleared before moving to the next step.

6. Do not use PuTTy for this step! Use your browser and navigate to your IP using port 4200. You will go to http://IP:4200. This step is connecting to your Hortonworks filesystem. Log in using **maria_dev** as the username and password and enter the following commands:
    a. `wget http://faculty.cs.uwosh.edu/faculty/krohn/ds730/installR.sh`
    b. `chmod +x installR.sh`
    c. `sudo ./installR.sh`
7. Wait for R to install. It will take several minutes. Once R is installed, type in **exit** to leave the Hortonworks filesystem. You can close out of this browser.

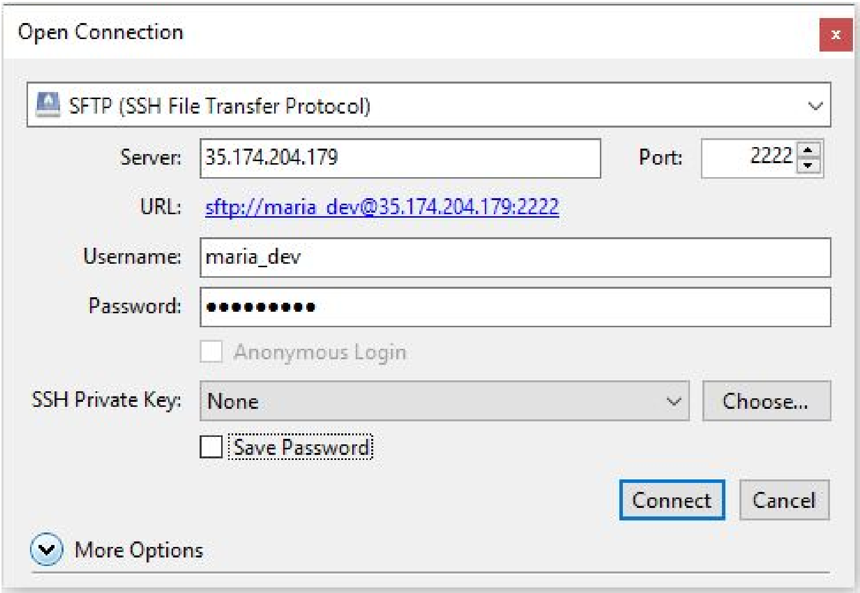
Connect to the Hortonworks Filesystem with Cyberduck

In order to connect to your Hortonworks filesystem using Cyberduck to transfer files back and forth from your local machine to your Hortonworks filesystem, follow these steps[14]:

1. Download Cyberduck from http://faculty.cs.uwosh.edu/faculty/krohn/ds730/cyberduck.html and install it. The download and install should be self-explanatory.
2. Open Cyberduck.
3. Go up to **File > Open Connection**.
4. Change the dropdown box to be **SFTP (SSH File Transfer Protocol)**
5. Enter in your IP address in the **Server**.
6. For the **Port,** leave it at **`2222`**

[14] If you are using a machine that will not let you install software (e.g. a locked down work machine), you can connect to the Virtual Lab and use CyberDuck on the Virtual Lab. CyberDuck will allow you to transfer files from your cloud machine to the Virtual Lab. You'll then be able to transfer those files from the Virtual Lab to your local machine.

7. Change the **Username** to maria_dev.
8. Enter in maria_dev as the **Password**.
9. Confirm that your screen looks something like this:



Open Connection

SFTP (SSH File Transfer Protocol)

Server: 35.174.204.179

Port: 2222

URL: sftp://maria_dev@35.174.204.179:2222

Username: maria_dev

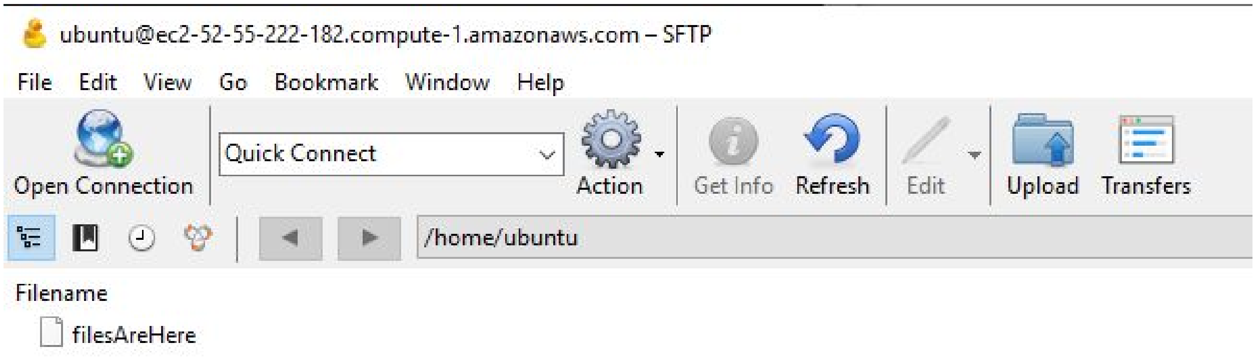
Password: ●●●●●●●●●●

□ Anonymous Login

SSH Private Key: None

□ Save Password

10. Click **Connect**.
11. If the system asks you to trust the key, simply indicate **OK**. You can choose to always accept the key. You should be connected and see something like this:



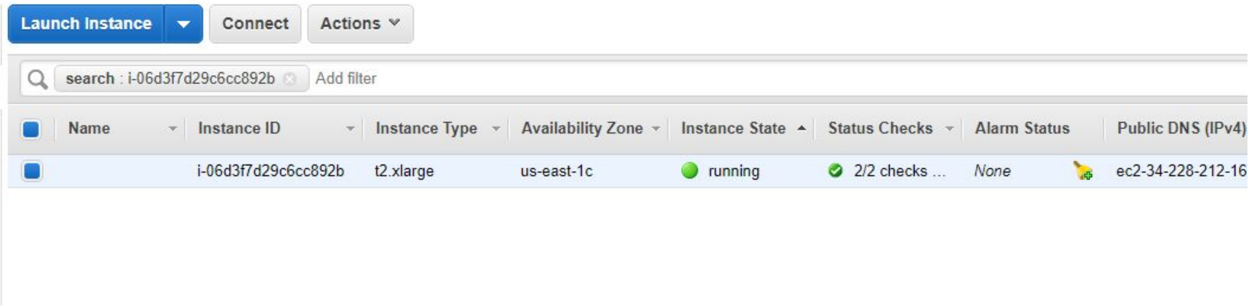
ubuntu@ec2-52-55-222-182.compute-1.amazonaws.com – SFTP

/home/ubuntu

Filename

filesAreHere

12. In order to upload a file, simply click on **Upload**. Find the file you want to upload and click **Choose**.
13. If you want to download a file from the server, double click on the file name and it will go to your **Downloads** folder.
14. If you are continuing with the next tasks, you can skip steps 14 and 15. Be sure to stop your virtual machine whenever you are done using it. If you are not continuing to the next task right now, go to your instances page on AWS and click on the instance you just created:



search : i-06d3f7d29c6cc892b

■	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status		Public DNS (IPv4)
■		i-06d3f7d29c6cc892b	t2.xlarge	us-east-1c	● running	✓ 2/2 checks ...	None		ec2-34-228-212-16

15. Click on **Instance State** → **Stop Instance**. We do not want to terminate this instance because we will be reusing this software throughout the semester.

Task 4: Testing Java

1. Connect to your Hortonworks filesystem using port 4200 in the browser. You can also connect to your Hortonworks filesystem by using port 2222 and PuTTy. If connecting with PuTTy, you will use maria_dev@yourIP as the hostname and 2222 as the port. You do not need to provide a key. When prompted, enter in maria_dev as the password.
2. Create a folder called **JavaExamples**. To do this, type in **mkdir JavaExamples**. To enter that directory, type in **cd JavaExamples**.
3. Create a file called **Test.java** in that folder (see VIM from earlier).
4. Enter the following code into that file:

```
public class Test{
    public static void main(String args[]){
        System.out.println("It works!");
    }
}
```

5. When you are back at the terminal window, compile your program by entering **`javac *.java`**
6. In order to run the file, enter **`java Test`**
    - You should see **It works!** printed to the screen.

Task 5: Test Python

It is assumed that you have some general problem solving skills with programming before starting this course. You should be familiar with the following topics before starting this course:

Selection Statements (e.g. if)
Repetition Statements (e.g. while, for)
Variables
Lists/Arrays
Calling Functions/Methods
Creating Functions/Methods

If you do not know what one of those topics is or do not have a good grasp on how to use each one, you might want to reconsider whether you are prepared for this course. We will write a few short Python programs to test out our Python installation and also to assess your programming ability. You must test your code on your Hortonworks filesystem using the command line. If you find yourself taking many hours to solve these tasks, then it might be best to gain some more experience solving problems with code before taking this course. A good website to work on your programming is https://projecteuler.net/archives. The first 20 or so problems are, relative to most problems in this course, easy to solve. Many of them require the skills above to solve and most of them are very short programs. For example, a solution for problem 1 is located at the end of this document.

Important:

- You are not allowed to use any external libraries (e.g. numpy, sympy, itertools etc.) for these problems as they make the problems trivial. If you are importing something other than sys, then you are not doing this correctly.
- Make sure to follow the directions exactly as my tester code will not work if you don't. For example, in the first problem, if you do not call your file **first.py** or create a function called **fact** instead of **factorial**, the automated tests will fail. Also, be sure to output exactly what the problem asks for and nothing else. For example, for the factorial problem, only output the actual factorial number. If the answer is 720, only output **720**. Do not output something along the lines of: "The factorial is 720." The final Python problem addresses the importance of creating appropriate output.

Run Python program from terminal

Many of you may be unfamiliar with the terminal (command line) and we will be using it for a portion of this course. The following short instructions describe how to create a Python file using only the terminal and how to run your Python code from the terminal.

1. Connect to your Hortonworks filesystem using the browser and port 4200 or PuTTy with port 2222.
2. Create your Python programs using vim as described before[15].
3. Once you have created your **pythonFile.py** program, go back to the terminal so that you can test your code.
4. In order to run your program, enter in the following 2 commands:

   ```
   chmod +x pythonFile.py
   ./pythonFile.py
   ```

   The first command tells the operating system that this file is one that we want to execute. The second command executes the program.

Reading input from redirected file input

In this course, it will be important that we know how to read in from the terminal window. Instead of opening up specific files, it is often handy to simply supply the filenames we want to process at runtime. In order to do this, the following command is used to “send” information from a file to a program:

```
./pythonFile.py < someInputFile
```

The < operator is called the redirect operator. It will take the text in someInputFile and send it to the first.py program. The following is a very simple python program on how to read in that text and simply print it back out:

[15] You are welcome to create your Python files locally. However, you must test them on the Hortonworks filesystem to ensure they work in that environment since all programs will be tested in that environment.

```
#!/usr/bin/python3.6
import sys
line = sys.stdin.readline()
while line:
  print(line)
  line = sys.stdin.readline()
```

As you might have guessed, the above program reads in 1 line at a time and simply prints it out. It isn't doing anything that interesting. You can use all of your standard string manipulations on the line variable, e.g. split, strip, find, etc. Note that the first line of the file is telling the operating system that this is a Python3 file and should be run as such. It is important that the first line of the code is exactly what is written above for the Python files in this activity.

Create a Factorial Calculation Program

You will be creating a program that calculates the factorial of a number.

1. Create a file called **first.py**.
2. Inside that python file, create a function called **factorial** that accepts an integer as an argument and returns the factorial of that integer. Do not print out anything in the factorial function. If the factorial function is called with a negative integer, a -1 is returned from the function. Do not prompt the user to enter a value. This will be taken care of by the tester program. Do not put anything else in this file except the factorial function (i.e. don't put tester code in this file).
3. Make sure your function definition looks like this:

   ```
   def factorial(val):
   ```

Output Average of Integers

Create a program that reads in integers and outputs the average of all of the numbers. Your program will be called using the following command:

```
./second.py < someInputFile
```

Some things to note:

- Only integers will appear in the input file.
- All of the integers in the input file will be separated by a space.
- All integers will be on a single line.
- Create a separate file called second.py to solve this problem.

- Do not read in from a specific file. If you are using the open function, you are doing this problem wrong.

Output Prime Numbers

The goal of this task is to create a Python program that reads in 2 integers that are separated by a space. There will only be 2 integers in the input file. Your Python code must be stored in a file called **third.py**. Your program then prints out all of the prime numbers **strictly** between those 2 numbers in increasing order. If there are no prime numbers strictly between those 2 numbers, then a **No Primes** message is printed out. The order that the numbers were input does not matter. Similar to the previous problem, the program will be called using the following command:

```
./third.py < someInputFile
```

Output: If there are no primes between the two numbers, your code outputs exactly:

No Primes

If there is one prime number, only that 1 number prints out.

If there are multiple prime numbers, the following pattern is used:

firstNum:secondNum!thirdNum&fourthNum:fifthNum!sixthNum&seventhNum

In other words, you separate the first number from the second number with a colon. You separate the second number from the third number with an exclamation point. You separate the third number from the fourth number with an ampersand. This delimiter pattern is repeated in subsequent numbers (colon, exclamation point, ampersand) until there are no more numbers left to be printed. There is no delimiter before the first element nor is there a delimiter after the last element. The numbers in the output should be in numerical order.

Example:
For example, if `5` and `24` were entered in that order, then **7:11!13&17:19!23** would print out.
If `24` and `5` were entered in that order, then **7:11!13&17:19!23** would print out.

Interpret Dirty Data

Throughout this course, your output will be tested using automated testers. Being able to test your code quickly allows us more time to dig into your code and provide better feedback. If we have to spend a lot of time trying to figure out what your output is or how to run your program, then this cuts into the time we have to look at your code. Therefore, it is important that your code is contained in the files we specify and produces the exact output we expect. Having incorrect delimiters, incorrect spacing, etc will cause our automated tests to fail. Be sure you read the output specifications closely so that you are producing the correct output. The output of the previous problem is quite arbitrary (although it assesses good problem solving skills) but this is done to ensure you are prepared for future output requirements. Future output requirements will be much more natural and obvious.

To show you why it is helpful to have all output be the same, consider the following problem. You are a manager for a global weather company and you have asked your regional offices to send you information in the following comma separated format:

Year,Month,Day,TimeCST,TemperatureF,WindMPH

All offices put a header row in each file to explain the data. A key thing is that the temperature should have been in Fahrenheit and located in the second to last column. Another key thing is that the wind should have been located in the last column and should have been in miles per hour. None of the offices sent the data exactly how you asked for it. Your job is to figure out what each of them did and then answer the following 2 questions:

1. Take the average of all of the wind speeds for March, 2006. Which city had the closest wind speed to 8.30 mph?
2. Take the average of all of the temperatures for 2006. Which city had the closest temperature, in fahrenheit, to 49.65.

You can download the 4 files from
http://faculty.cs.uwosh.edu/faculty/krohn/ds730/a1Weather.zip

The 4 cities are ABC, KLM, PQR and XYZ (see filenames). The way you clean the data and obtain the answers is entirely up to you with a couple of caveats. Any wind speed that is negative should be filtered out as this is not possible. Any temperature below -200 should be filtered out. Any wind speed that is listed as calm should be interpreted as 0mph or 0kph depending on the column header. Calm winds should not be filtered out. When filtering out data, only filter out the cell that is invalid, not the entire row. For example, if a row has a temperature of -9999 and a wind of 7.8, filter out the temperature but keep the wind value.

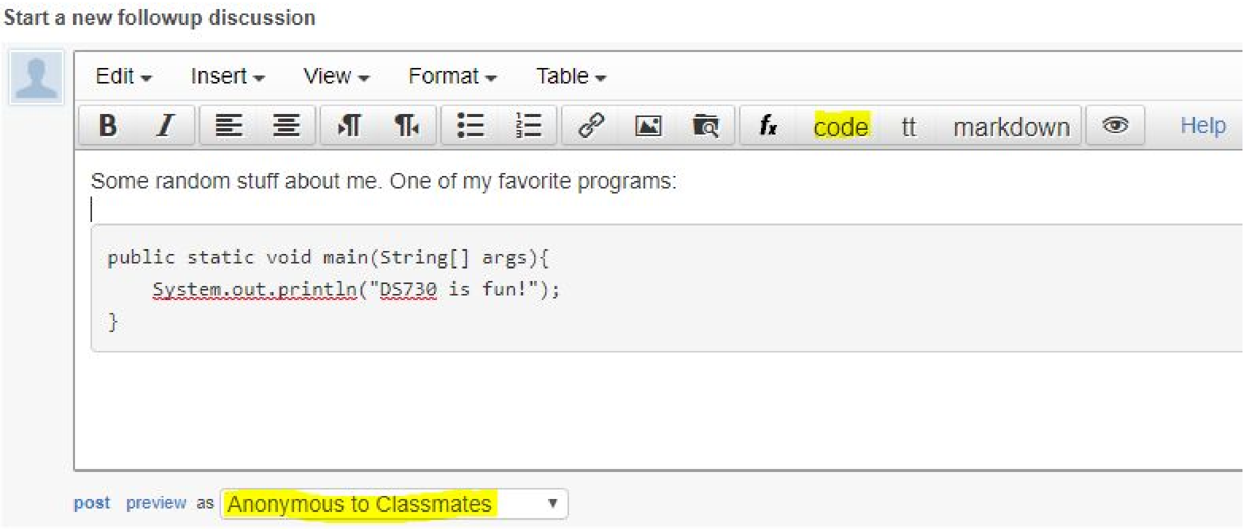
For this dirty data task, you only need to create a text file called **answers.txt** that contains the answer to each of the above questions. Any reasonable formatting of answers.txt is acceptable. You do not need to upload any code or explain how you obtained your answers for this final question.

Task 6: Using Piazza

Piazza is the only way you should be communicating with your instructors in this course. We know that all messages on piazza are class related. Because your piazza messages are class related, they are important. Not all of our emails are important. We will always check and respond to piazza first. On a related note, you should never use the messaging system that is built into canvas. Canvas messaging is not a tool that we use in this course and your message likely won't be seen.

Go to the online discussion board on Piazza (use the Introductions thread that is pinned at the top left of the page) and enter in some information about yourself. Since this is not a face-to-face course, I will likely never meet most of you. However, I would still like to know something about you. You can put whatever you want in that post. Tell me who you are, why you are interested in our data science degree, any interests or hobbies, what you hope to learn from this course, etc. You can write as much or as little as you want. If you have questions for me, feel free to add those and I'll answer them directly. I know you've probably had to do something similar in other courses so feel free to copy and paste that "intro document" so I can get to know you a little bit. **There are a few added requirements for the piazza introduction post that you must adhere to**:

1. **Post a follow-up discussion to the Introductions thread** that has already been posted on piazza. Do not start a new thread. When you are using piazza this semester, if your question is similar/related to an already posted question, please use the follow-up feature. This ensures that all questions related to a task/problem are in 1 location. Please use the search feature on piazza to see if your question has already been answered.
2. **Post your introductory follow-up anonymously**. There may be times where you wish you post anonymously this semester and I want to be sure you know how to do it. See the highlighted dropdown box below. Please note that anonymous posts are anonymous to other students in the course but not for your instructors. We know who is posting it (and therefore you will get credit for doing it).



Some random stuff about me. One of my favorite programs:

```
public static void main(String[] args){
    System.out.println("DS730 is fun!");
}
```

3. **Add your favorite chunk of code to your introduction message.** Click on the **code** button (see highlighted code below). If you are ever posting code in a thread, be sure to use this feature. If you copy and paste code, it will keep your indentation and your code will look much nicer.
4. Once you have posted your follow-up, click on the **Resolved** button above your post. Please use this feature throughout the semester when your follow-up is answered/fixed so we know your issue is resolved.
5. Lastly, **send a private note (not a question) to the instructors on piazza**.

Explore the other features that piazza offers. There is a LaTeX equation editor for those of you who know LaTeX. You can add a table. You can insert a file, an image, etc. We don't care what it says, just post anything. During the semester, if you want to post a lot of code, ask a question or send a comment that is only meant for the instructors, please use a private question/note on piazza instead of sending an email.

Task 7: Submitting your Work

We want to transfer the files to your local machine and zip them up for submitting. Once you have finished writing your Python code, you want to transfer them to your local machine. This can be done with CyberDuck. In order to transfer files:

1. Open Cyberduck.
2. Go up to **File > Open Connection**.
3. Change the dropdown box to be **SFTP (SSH File Transfer Protocol)**
4. Enter in your IP address in the **Server**.

5. Since we created our code on our Hortonworks filesystem, as a reminder, there is another way to access the Hortonworks filesystem. We can do it using PuTTy/Cyberduck using port 2222. For the **Port,** enter **2222**
6. Change the **Username** to maria_dev.
7. Enter in maria_dev as the **Password**.
8. Click the **Connect** button.
9. In order to upload a file, simply click on **Upload**. Find the file you want to upload and click **Choose**.
10. If you want to download a file from the server, double click on the file name and it will go to your **Downloads** folder.
11. Download your Python files from your Hortonworks filesystem.

You should have a total of 3 Python files: **first.py, second.py,** and **third.py**. You should also have an **answers.txt** file. Zip up these 4 files into 1 zip file called **a1.zip** and submit **a1.zip** to the dropbox.

Optional Tasks

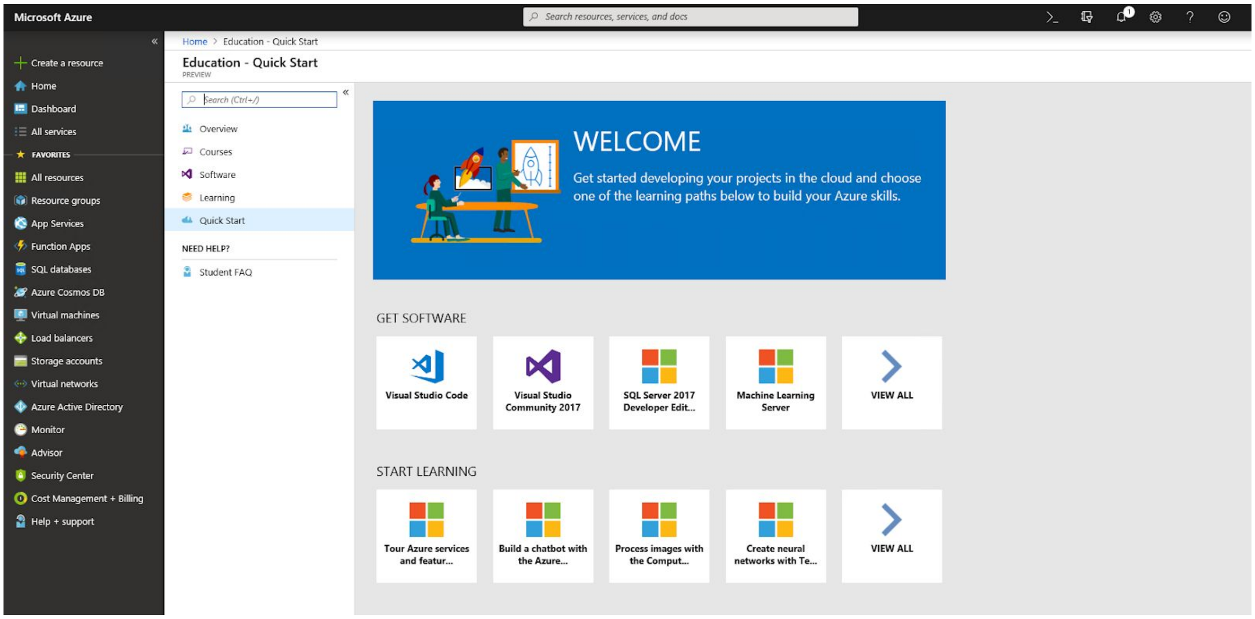
The following are optional tasks that you can do if you wish to set up a Hortonworks machine using Microsoft Azure or locally on your own machine. Since our goal is to have a uniform setup that everyone uses, we will not be providing any support for the following tasks. These steps are simply optional if you want a different cloud experience or if you want to set up your own Hortonworks machine locally.

Optional Task 1: Create a Hortonworks Sandbox on Azure

This task creates a sandbox in the cloud with most of the required software installed on it. The main advantage of this is that you do not have to use 1 single machine for this course and can connect to your sandbox from anywhere. If there is a problem, your instructor can connect to your machine and troubleshoot any issues. As a side note, the virtual machine you are creating has nothing to do with the virtual lab that is provided by the program. Do not mistake what you are doing here with the virtual lab that you may have used in previous courses. You should not use the virtual lab at all in this course.

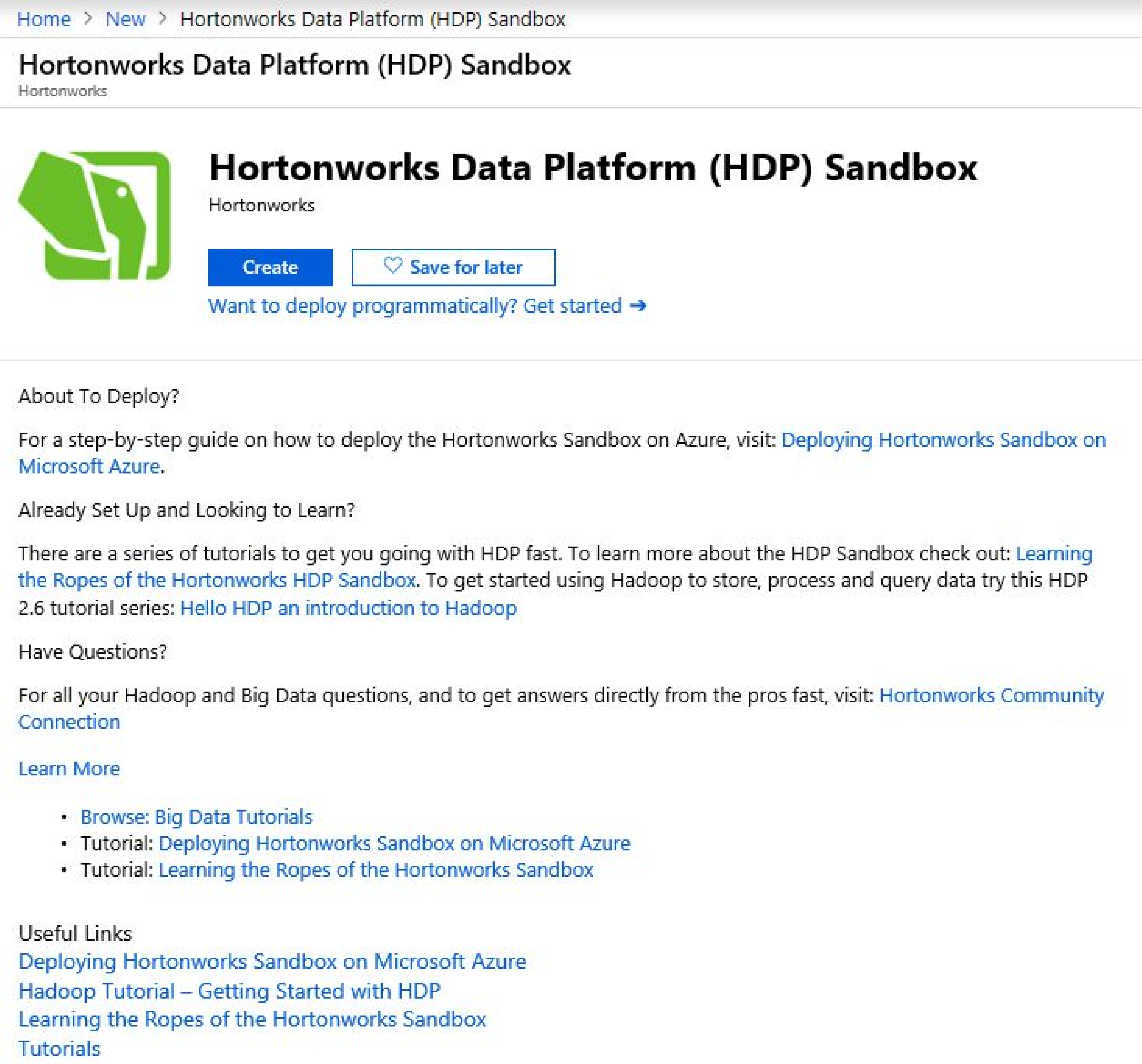
1. Go to https://azure.microsoft.com/en-us/free/students/ and click on the **Activate now** button. If you already have an account with your school email address, then sign in. Otherwise, click on **Create one!** and sign up for an account. Since Microsoft's verification is not universally the same, your prompts and pages may be different from mine. Essentially, do what you need to do to create an account.

This account is how your instructor created an account:
I was not able to create a new account with my .edu address so I simply used an old gmail account. I was eventually able to get signed in. When signed in, it asked me to verify my student status with a school email address. I typed in my **.edu** email account and clicked on **Verify academic status**. After a few seconds, an email showed up in my inbox and I clicked on the verification link. It asked me to verify again by phone. I was not able to verify a phone number from google voice so my guess is other VOIP numbers will also fail. I was able to get a text to my regular cell number so that worked fine. A call to your office number would probably work as well. In either case, my identity was verified and I accepted the agreement and hit the **Sign up** button. You may see a popup about starting a tour. Take the tour if you want or simply hit **Maybe later**. Once I was all signed up, I ended up on this page. Your page may look different from mine. As long as step 2 is possible for you, you are in the right place.



Get started developing your projects in the cloud and choose one of the learning paths below to build your Azure skills.

2. On the left hand side, click on **Create a resource**.
3. Do a search for **Hortonworks** and click on the option for **Hortonworks Data Platform (HDP) Sandbox**. You should see something that looks like this:



Home > New > Hortonworks Data Platform (HDP) Sandbox

Hortonworks Data Platform (HDP) Sandbox
Hortonworks

Hortonworks Data Platform (HDP) Sandbox
Hortonworks

Want to deploy programmatically? Get started ➔

About To Deploy?

For a step-by-step guide on how to deploy the Hortonworks Sandbox on Azure, visit: Deploying Hortonworks Sandbox on Microsoft Azure.

Already Set Up and Looking to Learn?

There are a series of tutorials to get you going with HDP fast. To learn more about the HDP Sandbox check out: Learning the Ropes of the Hortonworks HDP Sandbox. To get started using Hadoop to store, process and query data try this HDP 2.6 tutorial series: Hello HDP an introduction to Hadoop

Have Questions?

For all your Hadoop and Big Data questions, and to get answers directly from the pros fast, visit: Hortonworks Community Connection

Learn More

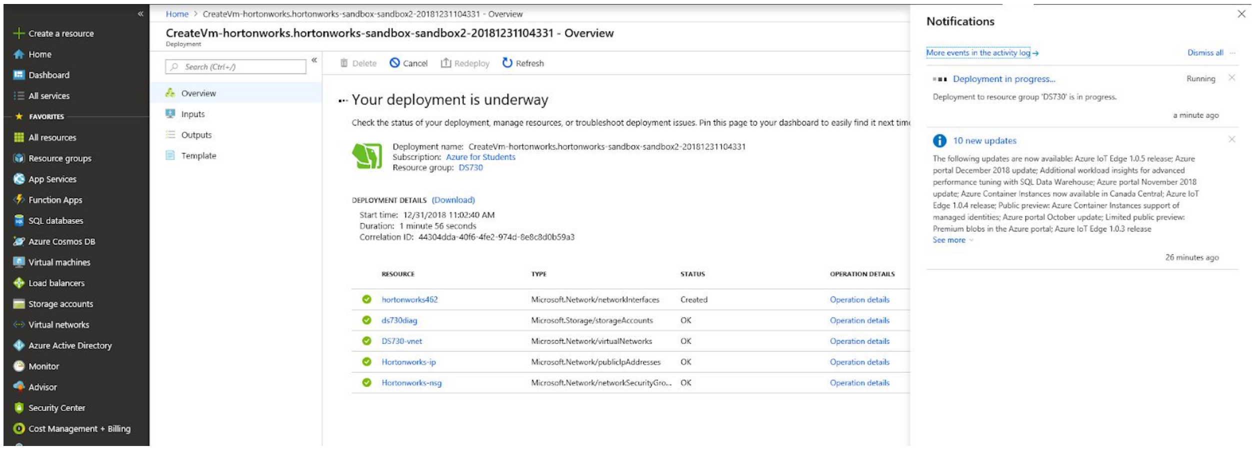
- Browse: Big Data Tutorials
- Tutorial: Deploying Hortonworks Sandbox on Microsoft Azure
- Tutorial: Learning the Ropes of the Hortonworks Sandbox

Useful Links
Deploying Hortonworks Sandbox on Microsoft Azure
Hadoop Tutorial – Getting Started with HDP
Learning the Ropes of the Hortonworks Sandbox
Tutorials

4. Click on the **Create** button. From here we just need to customize our sandbox.
5. On the **Basics** tab, click on **Create new** for the resource group. You can call your resource anything you want, e.g. DS730. The **Virtual machine name** can be called Hortonworks. Under Instance Details, look at the **Size** option. We are only allowed 4 cores with our student account. Therefore, click on the **Change size** link and choose the **Standard B4ms** option. Change the **Administrator Account Authentication type** to be a password. It is less secure than an SSH public key but it is sufficient for this course. Create any username/password you want. You will never need to use this username/password in this course. Feel free to click on the Next buttons if you would like to configure your virtual

machine more. However, the default options are sufficient for this course. Click on the **Review + create** button.

6. Review the options you chose and click on the **Create** button at the bottom. Deployment will take a while so simply wait for it to finish. You should see something like this (click on the bell icon to get the notifications on the right hand side):



Home > CreateVm-hortonworks.hortonworks-sandbox-sandbox2-20181231104331 - Overview

CreateVm-hortonworks.hortonworks-sandbox-sandbox2-20181231104331 - Overview
Deployment

Your deployment is underway

Check the status of your deployment, manage resources, or troubleshoot deployment issues. Pin this page to your dashboard to easily find it next tim

Deployment name: CreateVm-hortonworks.hortonworks-sandbox-sandbox2-20181231104331
Subscription: Azure for Students
Resource group: DS730

DEPLOYMENT DETAILS (Download)

Start time: 12/31/2018 11:02:40 AM
Duration: 1 minute 56 seconds
Correlation ID: 44304dda-40f6-4fe2-974d-8e8c8d0b59a3

RESOURCE	TYPE	STATUS	OPERATION DETAILS
hortonworks462	Microsoft.Network/networkInterfaces	Created	Operation details
ds730diag	Microsoft.Storage/storageAccounts	OK	Operation details
DS730-vnet	Microsoft.Network/virtualNetworks	OK	Operation details
Hortonworks-ip	Microsoft.Network/publicIpAddresses	OK	Operation details
Hortonworks-nsg	Microsoft.Network/networkSecurityGro...	OK	Operation details

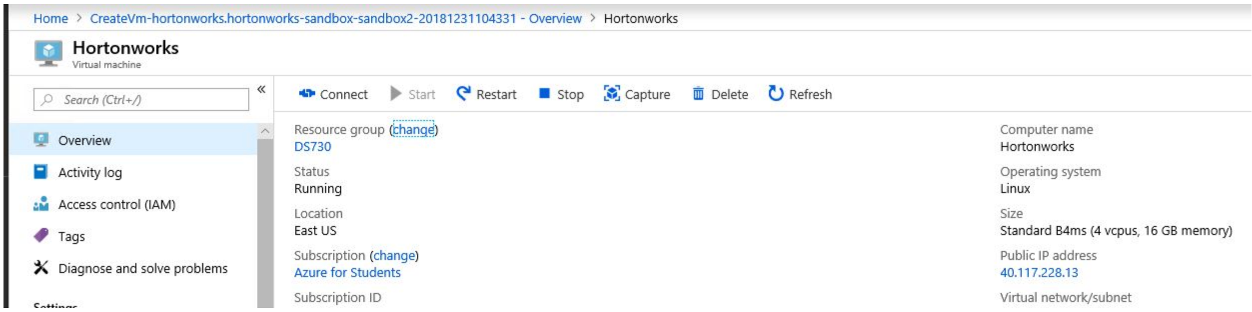
Notifications

More events in the activity log → Dismiss all

Deployment in progress... Running
Deployment to resource group 'DS730' is in progress.
a minute ago

10 new updates
The following updates are now available: Azure IoT Edge 1.0.5 release; Azure portal December 2018 update; Additional workload insights for advanced performance tuning with SQL Data Warehouse; Azure portal November 2018 update; Azure Container Instances now available in Canada Central; Azure IoT Edge 1.0.4 release; Public preview: Azure Container Instances support of managed identities; Azure portal October update; Limited public preview: Premium blobs in the Azure portal; Azure IoT Edge 1.0.3 release
See more
26 minutes ago

7. After about 5 minutes, the deployment should succeed. Yours will likely take longer as you will be doing it at the beginning of the semester when everyone else is trying to do it. Once the deployment has succeeded, click on the **Go to resource** button in the middle of the screen. You will see something like this:



Home > CreateVm-hortonworks.hortonworks-sandbox-sandbox2-20181231104331 - Overview > Hortonworks

Hortonworks
Virtual machine

Resource group (change)
DS730

Status
Running

Location
East US

Subscription (change)
Azure for Students

Subscription ID

Computer name
Hortonworks

Operating system
Linux

Size
Standard B4ms (4 vcpus, 16 GB memory)

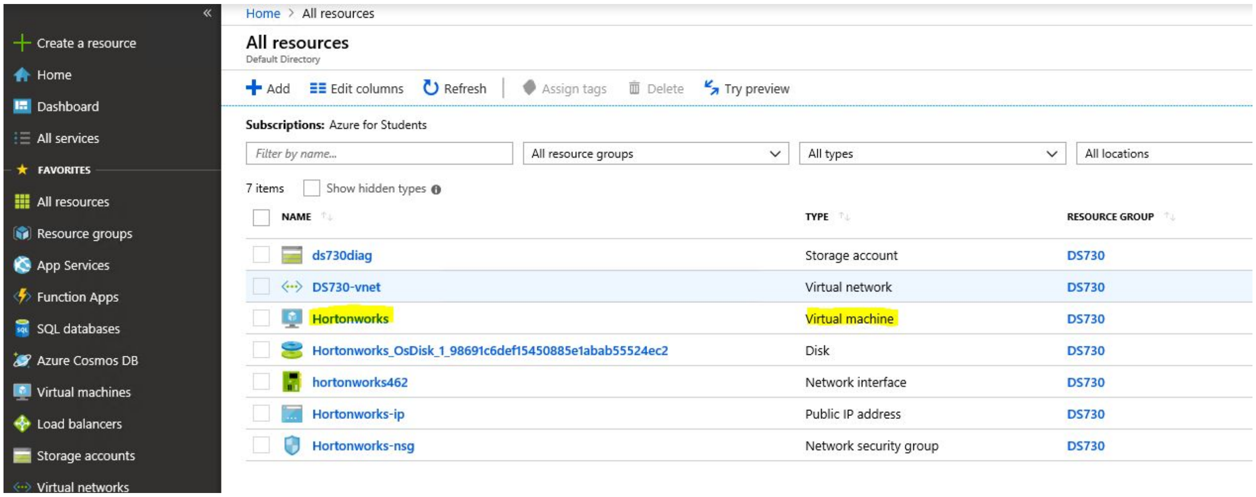
Public IP address
40.117.228.13

Virtual network/subnet

8. Under the **Settings** option, click on **Networking**. By default, essentially all ports are closed for security. However, we need to connect to them so we need to open up some of the ports. Click on the **Add inbound port rule** button. Add the following port using these rule:
    a. **Destination port ranges** - Set it to be 8080. It might default at that port already. If that is the case, then leave it.
    b. **Name** - Not critical but change it to **Ambari** so you know what the port is.
    c. The rest of the options can be left at their defaults. Feel free to change the **Source** if you are super concerned about security.

It will take a minute to create the rule. Add the following Port/Name inbound port rules as well: (4200/ssh, 9995/zeppelin, 2222/hortonworksfs). You may have to change the priority to be a different number for each rule. Change it to any number that is unique.

9. Your Hortonworks Sandbox is setup and is now ready to use. A great guide to learning about your Sandbox is here: https://hortonworks.com/tutorial/learning-the-ropes-of-the-hortonworks-sandbox However, everything you need to know for this course will be shown in this and future activities.
10. Click on the **All resources** option on the left. Click on the **Hortonworks** link where the type is **Virtual machine**:



All resources
Default Directory

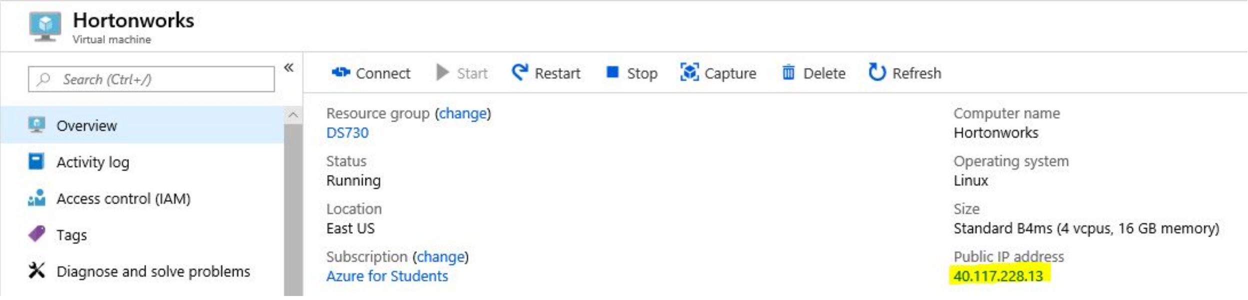
Subscriptions: Azure for Students

Filter by name... | All resource groups | All types | All locations

7 items □ Show hidden types

NAME	TYPE	RESOURCE GROUP
ds730diag	Storage account	DS730
DS730-vnet	Virtual network	DS730
Hortonworks	Virtual machine	DS730
Hortonworks_OsDisk_1_98691c6def15450885e1abab55524ec2	Disk	DS730
hortonworks462	Network interface	DS730
Hortonworks-ip	Public IP address	DS730
Hortonworks-nsg	Network security group	DS730

11. You will find your Public IP address on the right hand side. Mine looks like this:



Hortonworks
Virtual machine

Resource group (change)
DS730

Status
Running

Location
East US

Subscription (change)
Azure for Students

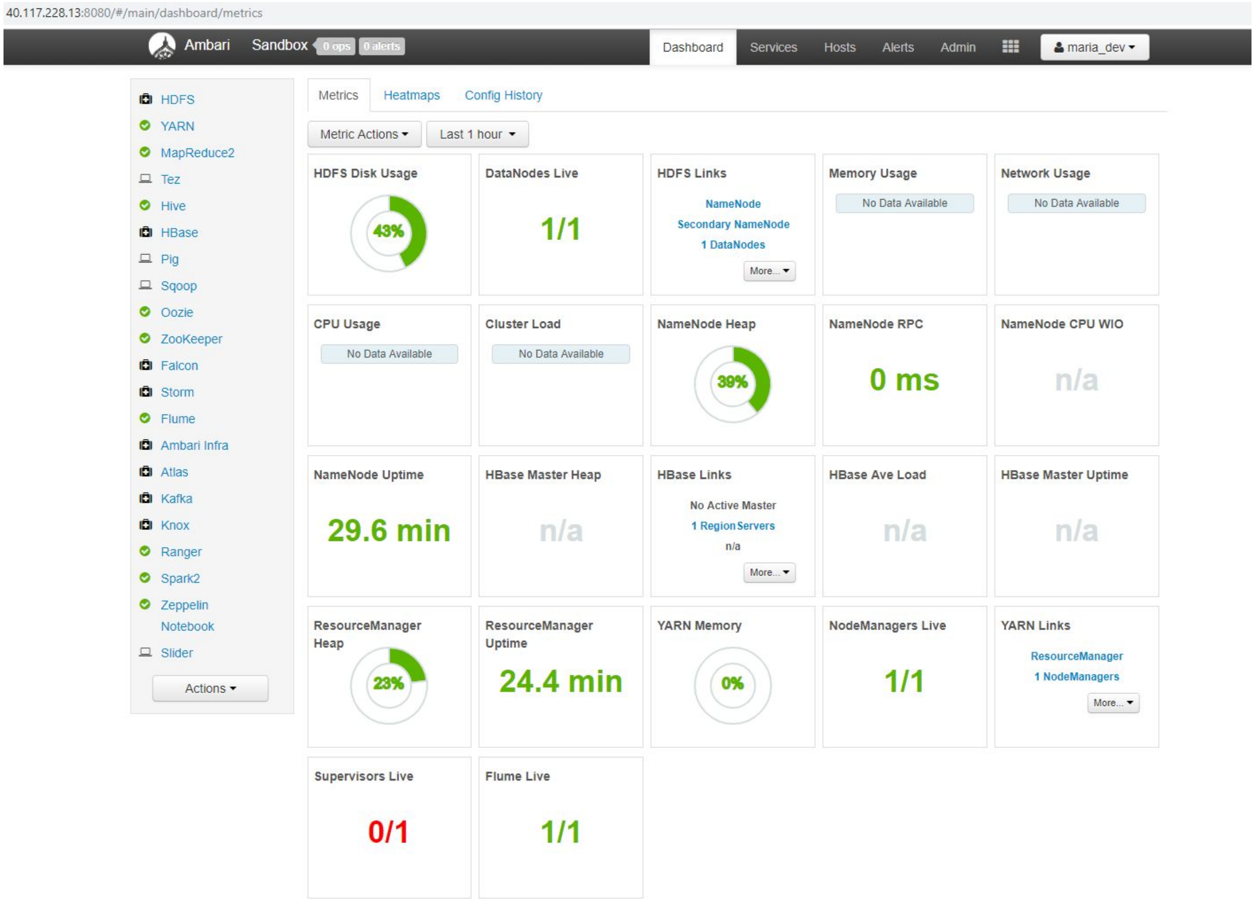
Computer name
Hortonworks

Operating system
Linux

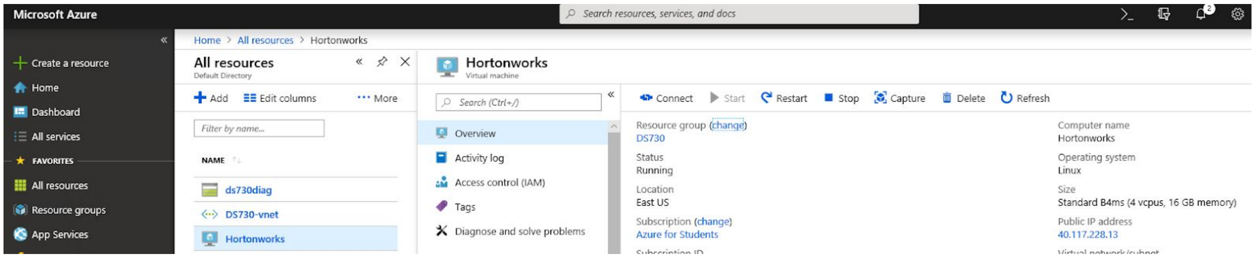
Size
Standard B4ms (4 vcpus, 16 GB memory)

Public IP address
40.117.228.13

12. Note that my IP address is 40.117.228.13 and I will use that IP in the steps below. Replace my IP with yours when you do the following steps.
13. Go to http://40.117.228.13:8080 and enter in **maria_dev** as the username and password. You should see something like this:



14. **This step is important for all future activities and projects whenever you start up your virtual machine.** You must check this page before beginning any activity/project to ensure the system is ready to use. As long as there are 0 alerts at the top, you are ready to start working. **If you see any alerts, you must wait for the alerts to clear before starting.** Even though you may be able to load up the 8080 port and the 4200 port, it does not mean the machine is completely started and ready to use. **You must wait for all alerts to clear before beginning.** Note that an orange alert is fine as this is just a warning. If you only have orange alerts, you can continue. If you have red alerts, you must wait. The virtual machine does take a few minutes to load up all of the software. Feel free to explore the dashboard to see all of the tools available for you to use. Once you are done exploring, you can close out of this window.
15. The next 2 steps are very important to do each time you are done using your virtual machine. If you forget to do these steps, your virtual machine will still be running and you will run out of credits before the semester ends. Go back to your Azure browser window and click on **All resources**. Click on the **Hortonworks Virtual Machine** and you should see something like this:



Resource group (change)
DS730

Status
Running

Location
East US

Subscription (change)
Azure for Students

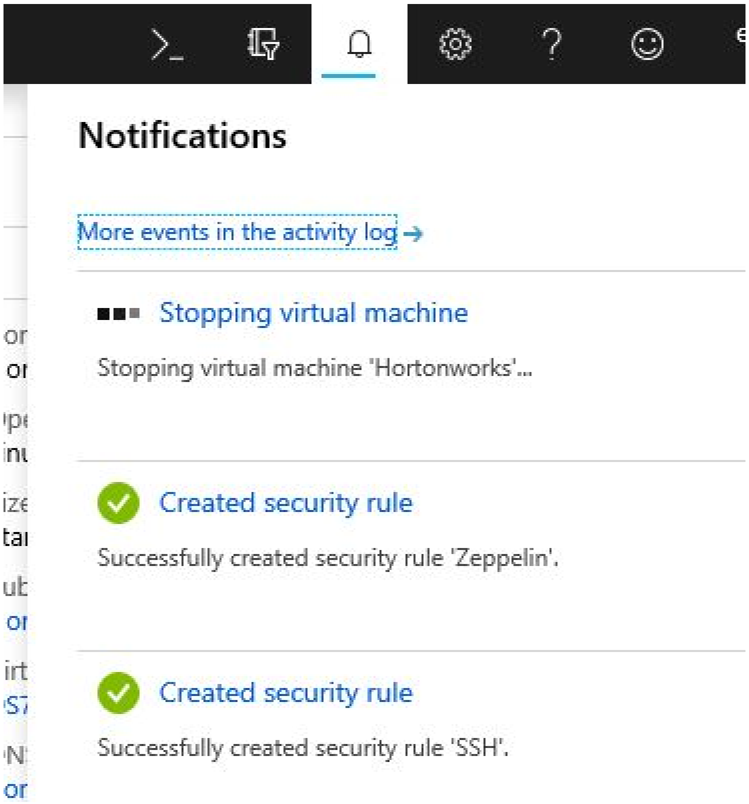
Computer name
Hortonworks

Operating system
Linux

Size
Standard B4ms (4 vcpus, 16 GB memory)

Public IP address
40.117.228.13

16. Click on the **Stop** button in the top-middle of the page. Your virtual machine will shut down and you will no longer be charged for the machine. You are only using your credits when the virtual machine is running. Wait a few minutes and ensure that the virtual machine shuts down. You can click on the bell icon in the upper right hand corner to see what is happening. You should see something like this showing that the virtual machine is being stopped:


Notifications

More events in the activity log →

Stopping virtual machine

Stopping virtual machine 'Hortonworks'...

Created security rule

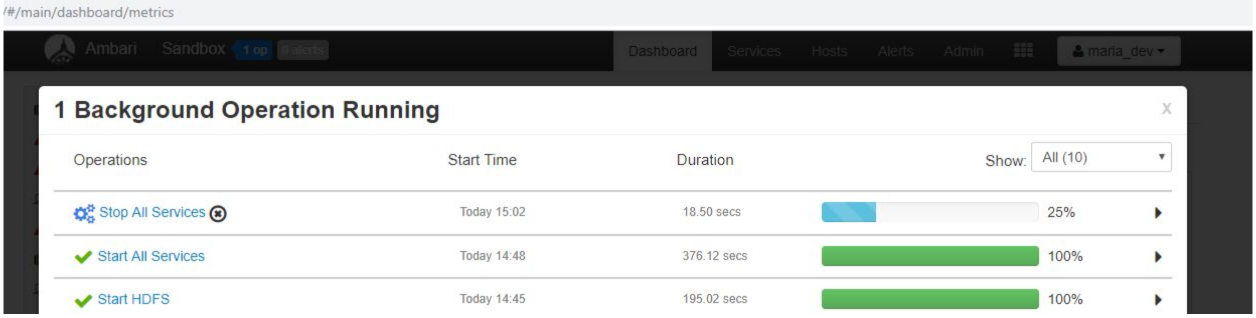
Successfully created security rule 'Zeppelin'.

Created security rule

Successfully created security rule 'SSH'.

17. **Troubleshooting Hortonworks**. If you find your machine in a bad state and your programs are taking a long time to run or simply aren't running at all, these are the best tips to getting back into a good state. You do not have to do these at the beginning but just remember these tips are here for future reference.
    a. If you find there are several alerts or something didn't load up correctly, you are encouraged to debug this by simply restarting the services on the virtual machine. If you find that Hadoop, Hive, Pig, etc. are not functioning properly, restarting all of the services will often help. To do this, click on

the **Actions** button in the lower left hand corner. Click on **Stop All**. Click on the **Confirm Stop** button that pops up. It will likely take a couple of minutes for all services to stop:



1 Background Operation Running

Operations	Start Time	Duration	Show: All (10)
Stop All Services	Today 15:02	18.50 secs	25%
✓ Start All Services	Today 14:48	376.12 secs	100%
✓ Start HDFS	Today 14:45	195.02 secs	100%

Once all services have stopped, click the **OK** button. To start up the services, click on the **Actions** button, click on **Start All** then click on **Confirm Start**. Your services are now restarting.

b. If the previous step does not solve your issue, then stopping the entire machine and restarting it is the next thing to try. Stop your virtual machine by using steps 17 and 18 above. In order to start your virtual machine up again, click on the **Start** button in the top-middle of the screen. After a few minutes, your machine will start up again.

c. If restarting your machine does not solve the problem, then the fastest solution is to delete your virtual machine and create a new one. If some step was skipped or your virtual machine got itself into a bad state, then troubleshooting the issue could take hours. However, deleting your current virtual machine and creating a new one takes minutes. Just be aware that if you delete your virtual machine, any files on that virtual machine will be deleted as well. Be sure to download/copy over anything you want before deleting.

To delete your virtual machine, click on **All resources** on the left hand side. Select all of the resources listed and click on the **Delete** button at the top of the screen. Type **yes** to confirm deletion and click the **Delete** button. Go back to step 2 and create a new virtual machine[16].

[16] Since we are using a student account, we are limited in the number of resources we can use. Therefore, you need to wait until your resources have been deleted before you can create a new virtual machine. Even if you wait, it is possible the system is still updating itself and may not let you create a new machine right away. If you encounter errors creating a new virtual machine (it may say validation failed when you click on review + create), simply wait a few minutes and try again. Eventually, you will be able to create a new virtual machine.

Optional Task 2: Create a Local Hortonworks Sandbox

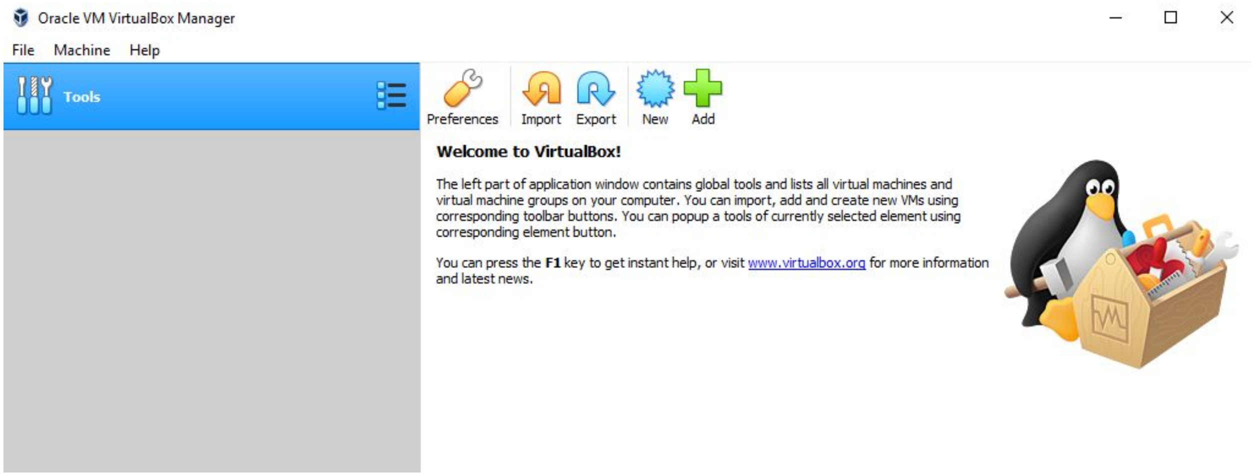
The newest version of Hortonworks is different from the one on Azure and there are considerable differences in the UI. There are also considerable differences in how you run each of the programs. All instructions for the remainder of the course will assume that you are running Hortonworks 2.6.5. Therefore, you are encouraged to download the older version of Hortonworks to be consistent with the one on Azure.

This task creates a Hortonworks virtual machine and installs all of the software we are using in this course. This task takes a couple of hours depending on your Internet connection speed. However, the majority of it is waiting for everything to download. You will need about 100GB of hard drive space available and a machine that has at least 8GB of memory, preferably more[17]. These instructions have worked on a machine running a 64-bit version of Windows 10 and a 64-bit version of Windows 7[18].

1. **Download VirtualBox**. The current version is 6.0.8. You can find the download here: http://faculty.cs.uwosh.edu/faculty/krohn/ds730/virtualbox.html
2. Install VirtualBox on your machine. It is a simple install that requires you to click **Next** and **Install** a few times.
    - If it asks about installing network devices, this is no problem; you can simply click **Install** on those prompts.
3. When the install is complete, click **Finish**.
4. Open VirtualBox up once it is installed. You should see something like this:

[17] You should have a minimum of 8GB of free RAM in order for Hortonworks to work. The more memory the better.

[18] You need to ensure your machine has VT-x (virtualization) enabled. This often needs to be done in the BIOS. If VirtualBox is failing to work, ensure virtualization is enabled. See https://www.howtogeek.com/213795/how-to-enable-intel-vt-x-in-your-computers-bios-or-uefi-firmware/ for more details. If your machine is not working and you are not comfortable modifying BIOS settings or you just don't want to do it, there are several other options to run Hortonworks in the cloud.

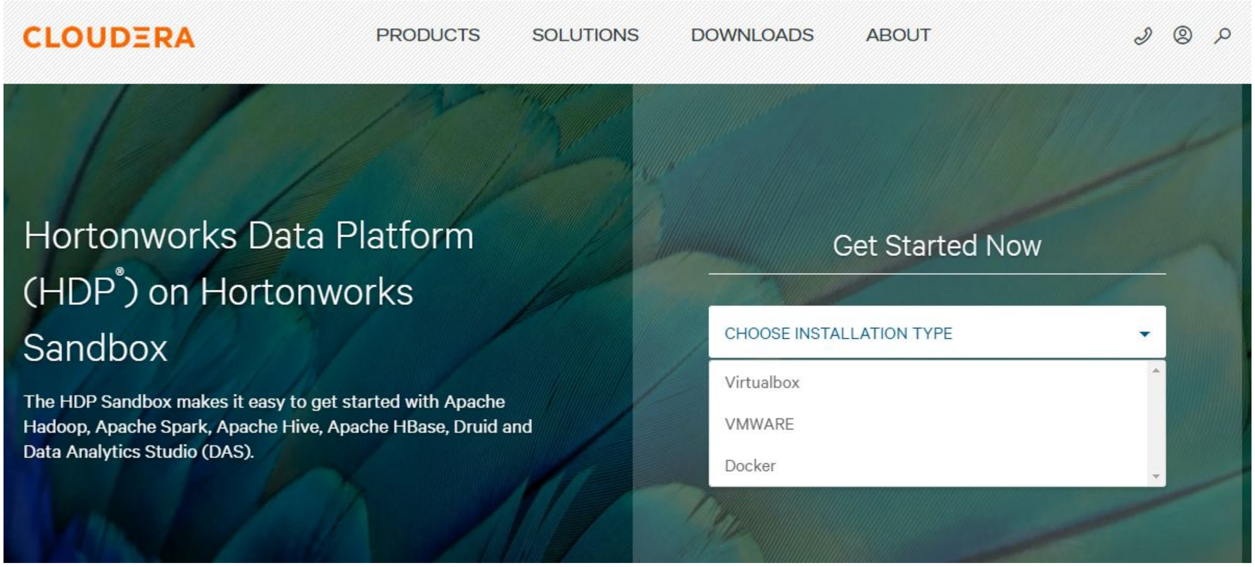


Welcome to VirtualBox!

The left part of application window contains global tools and lists all virtual machines and virtual machine groups on your computer. You can import, add and create new VMs using corresponding toolbar buttons. You can popup a tools of currently selected element using corresponding element button.

You can press the **F1** key to get instant help, or visit www.virtualbox.org for more information and latest news.

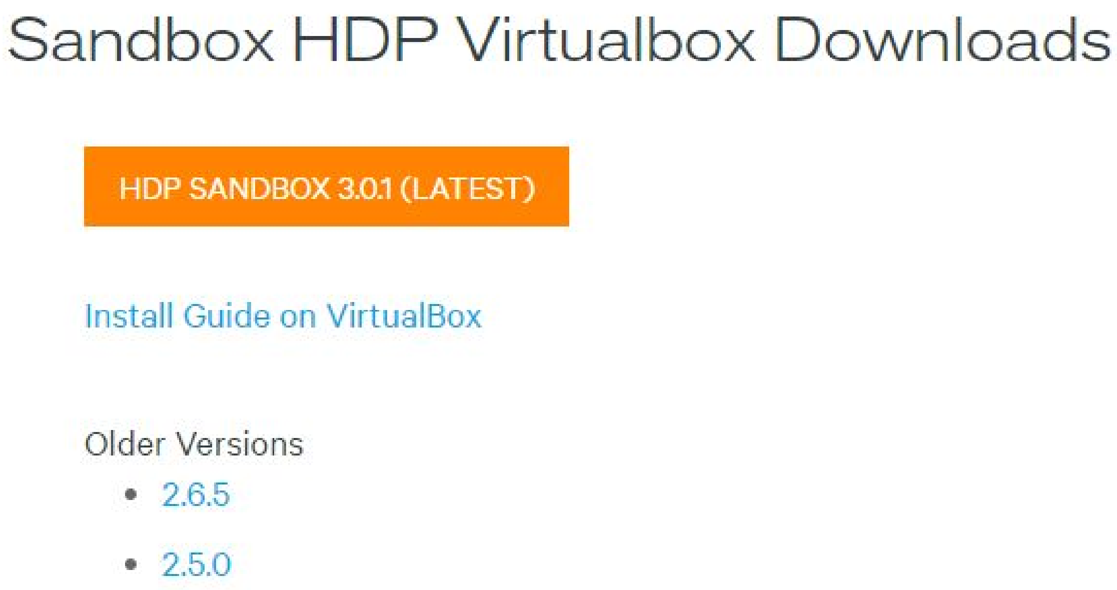
5. **Download Hortonworks Sandbox**. The version of Hortonworks Data Platform (HDP) that is similar to the one on Azure is 2.6.5. You can download it from here: http://faculty.cs.uwosh.edu/faculty/krohn/ds730/hortonworks.html
    - Click on the dropdown box where it says **Choose Installation Type** and select VirtualBox:



Hortonworks Data Platform (HDP®) on Hortonworks Sandbox

The HDP Sandbox makes it easy to get started with Apache Hadoop, Apache Spark, Apache Hive, Apache HBase, Druid and Data Analytics Studio (DAS).

    - It will likely ask you to fill out your name, email address, etc before downloading. Fill out the form, read and accept the terms and conditions and click Submit. On this part of the page, be sure to click on 2.6.5:



Sandbox HDP Virtualbox Downloads

HDP SANDBOX 3.0.1 (LATEST)

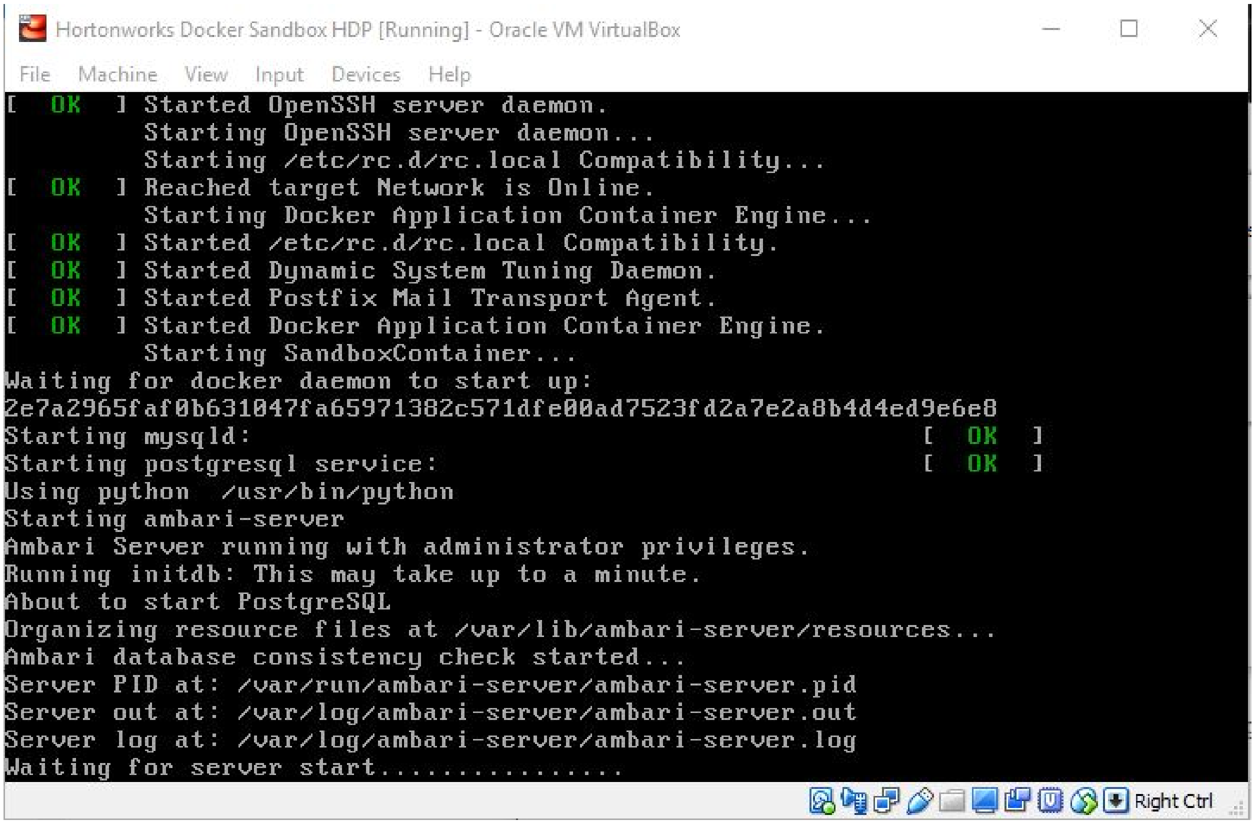
Install Guide on VirtualBox

Older Versions

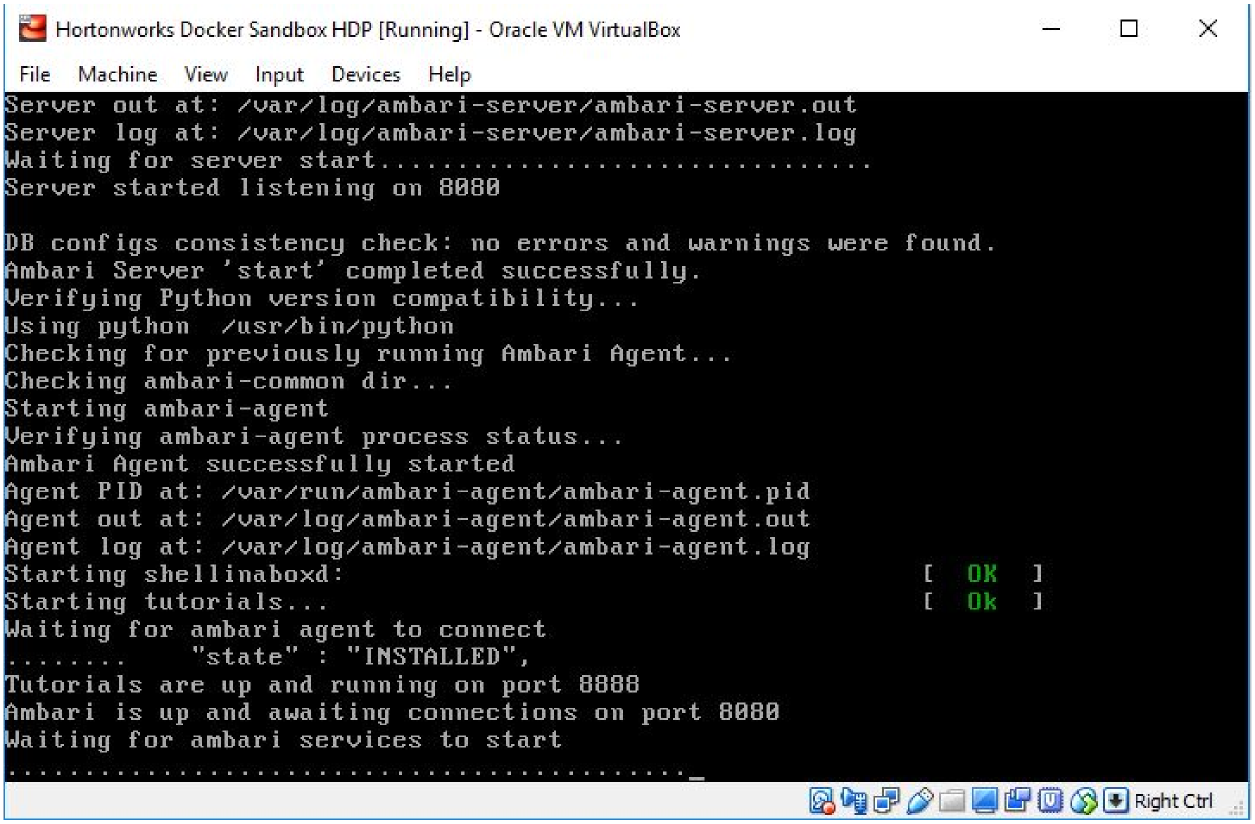
- 2.6.5
- 2.5.0

  - Note that this is a 10GB download so it may take a while if you have a slower home connection.

6. Once the file has downloaded, go to VirtualBox and click on **File** → **Import Appliance**.
7. Click on the folder icon and find the ***.ova** file you just downloaded and click **Open**. For reference, mine was called HDP_2.6.5_virtualbox_180626.ova. Once you have selected it, click **Next**.
8. All of the settings can remain unchanged. If you wish to store your sandbox on an external drive, scroll down and click on the **Virtual Disk Image** location setting at the bottom and change the location. You may also change the **RAM** setting if you would like to use more/less memory. Do not use less than 4GB of memory or your sandbox may not run well. The recommended amount of 8GB is preferred. Click on **Import**.
9. As a reference, my initial *Importing virtual disk image* import took roughly a minute. Once everything is imported, click on the name of the virtual box (probably called Hortonworks Docker Sandbox HDP) and click on **Start**. You should see something like these:


Hortonworks Docker Sandbox HDP [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Right Ctrl

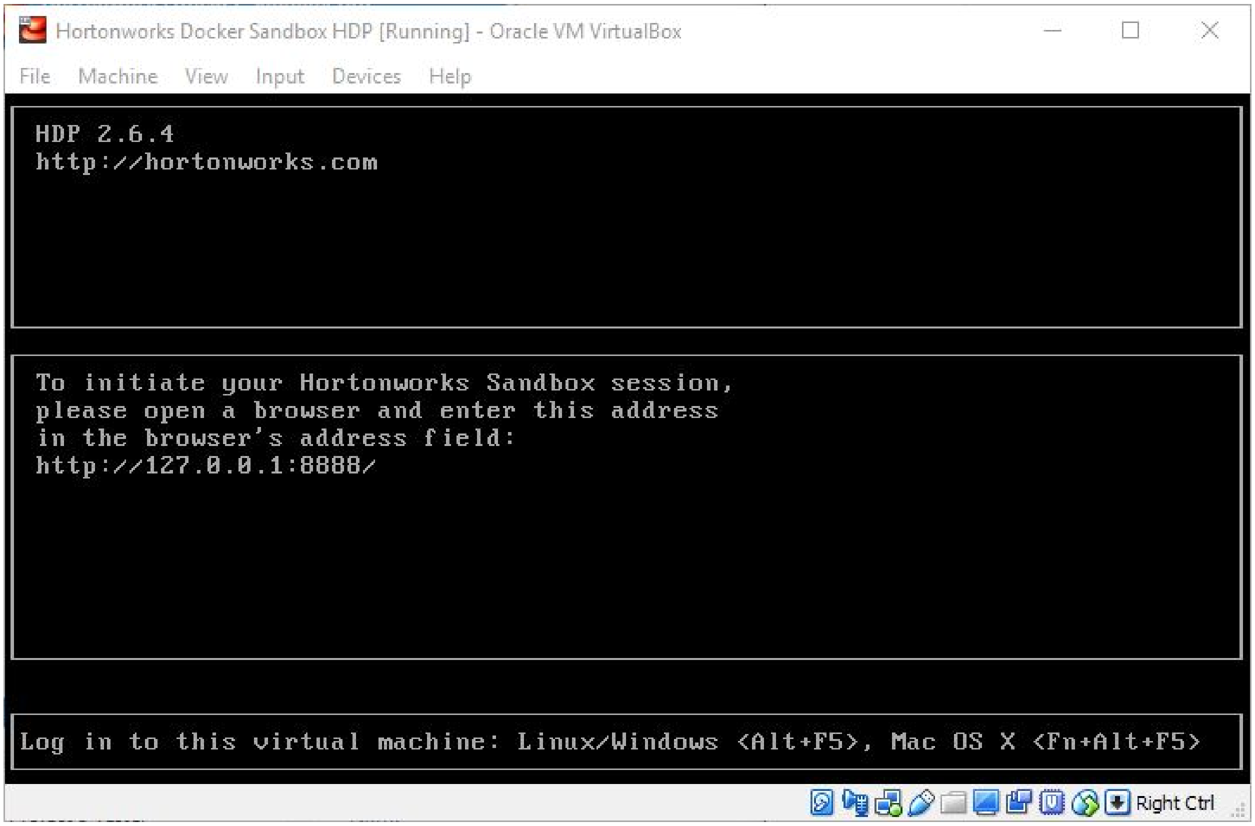
```
[  OK  ] Started OpenSSH server daemon.
         Starting OpenSSH server daemon...
         Starting /etc/rc.d/rc.local Compatibility...
[  OK  ] Reached target Network is Online.
         Starting Docker Application Container Engine...
[  OK  ] Started /etc/rc.d/rc.local Compatibility.
[  OK  ] Started Dynamic System Tuning Daemon.
[  OK  ] Started Postfix Mail Transport Agent.
[  OK  ] Started Docker Application Container Engine.
         Starting SandboxContainer...
Waiting for docker daemon to start up:
2e7a2965faf0b631047fa65971382c571dfe00ad7523fd2a7e2a8b4d4ed9e6e8
Starting mysqld:                                           [  OK  ]
Starting postgresql service:                               [  OK  ]
Using python  /usr/bin/python
Starting ambari-server
Ambari Server running with administrator privileges.
Running initdb: This may take up to a minute.
About to start PostgreSQL
Organizing resource files at /var/lib/ambari-server/resources...
Ambari database consistency check started...
Server PID at: /var/run/ambari-server/ambari-server.pid
Server out at: /var/log/ambari-server/ambari-server.out
Server log at: /var/log/ambari-server/ambari-server.log
Waiting for server start................
```


Hortonworks Docker Sandbox HDP [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Right Ctrl

```
Server out at: /var/log/ambari-server/ambari-server.out
Server log at: /var/log/ambari-server/ambari-server.log
Waiting for server start................................
Server started listening on 8080

DB configs consistency check: no errors and warnings were found.
Ambari Server 'start' completed successfully.
Verifying Python version compatibility...
Using python  /usr/bin/python
Checking for previously running Ambari Agent...
Checking ambari-common dir...
Starting ambari-agent
Verifying ambari-agent process status...
Ambari Agent successfully started
Agent PID at: /var/run/ambari-agent/ambari-agent.pid
Agent out at: /var/log/ambari-agent/ambari-agent.out
Agent log at: /var/log/ambari-agent/ambari-agent.log
Starting shellinaboxd:                                     [  OK  ]
Starting tutorials...                                      [  Ok  ]
Waiting for ambari agent to connect
........    "state" : "INSTALLED",
Tutorials are up and running on port 8888
Ambari is up and awaiting connections on port 8080
Waiting for ambari services to start
............................................_
```

10. Eventually, you will see the following screen. If the screen goes blank, simply hit the space key. Once you see this, your virtual machine is ready to use:


Hortonworks Docker Sandbox HDP [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Right Ctrl

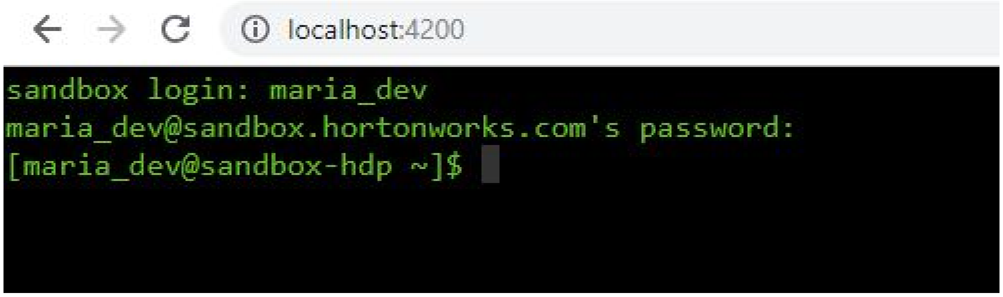
```
HDP 2.6.4
http://hortonworks.com

To initiate your Hortonworks Sandbox session,
please open a browser and enter this address
in the browser's address field:
http://127.0.0.1:8888/

Log in to this virtual machine: Linux/Windows <Alt+F5>, Mac OS X <Fn+Alt+F5>
```

The virtual machine that you setup is like any other machine that you have created. There is an operating system, a filesystem, etc. For those of you familiar with the command line, you may want to use the command line to do certain things. Hortonworks provides a nice web client that allows you to connect to your machine.

11. Open up an Internet browser and go to http://localhost:4200. You should see something like this:



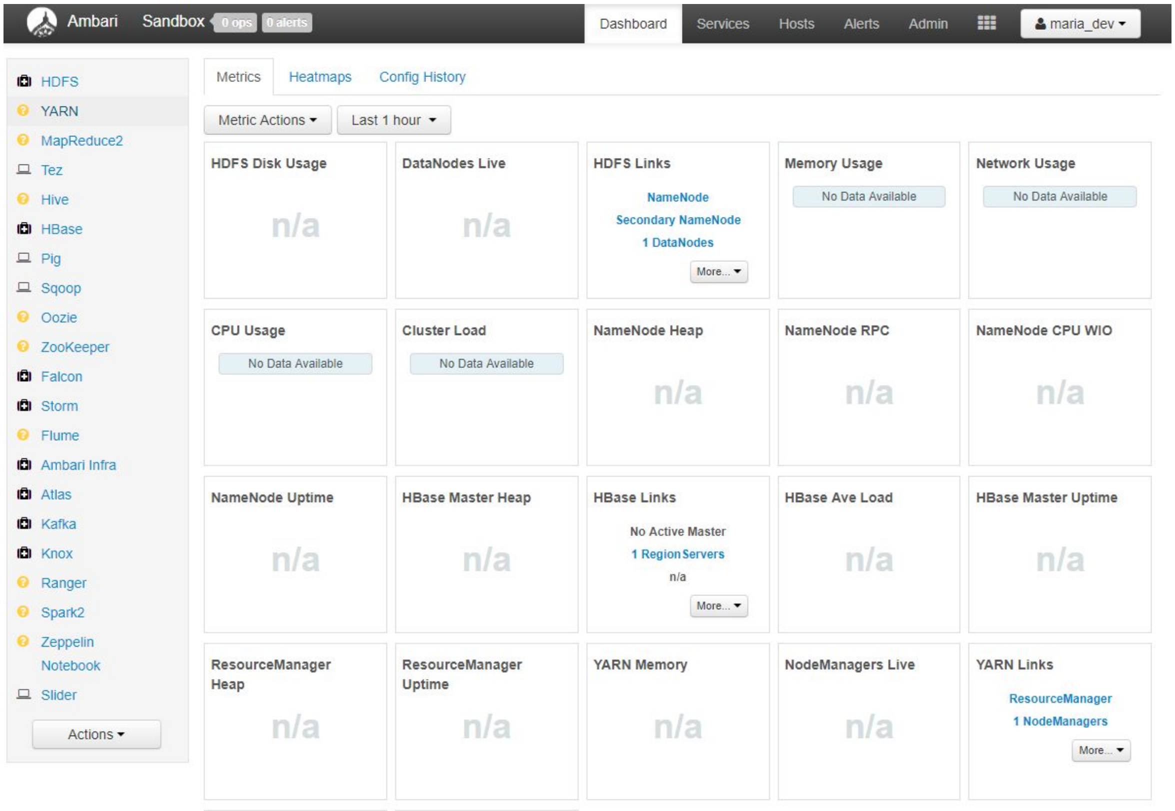
```
sandbox login: maria_dev
maria_dev@sandbox.hortonworks.com's password:
[maria_dev@sandbox-hdp ~]$
```

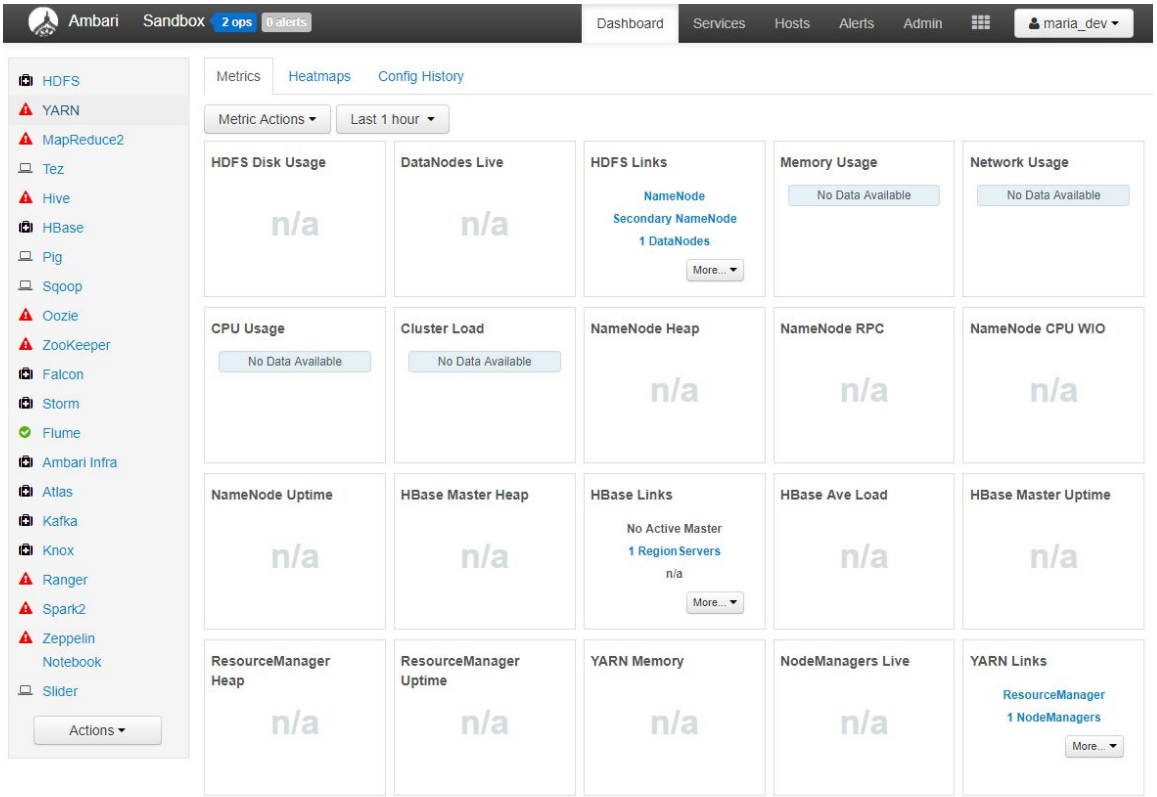
12. Type in **maria_dev** for your username and password.

13. When you are logged in, type in **exit** to close your connection. It will say *Session closed.* and you will see a **Connect** button in the middle of the screen. You can close this window in your browser.

The Ambari dashboard is a nice administrative tool that comes with the sandbox. You will find many applications running but may also find some that are stopped. This is fine as we won't be able to explore everything in the sandbox. Our goal is to view/use some of the main ones.

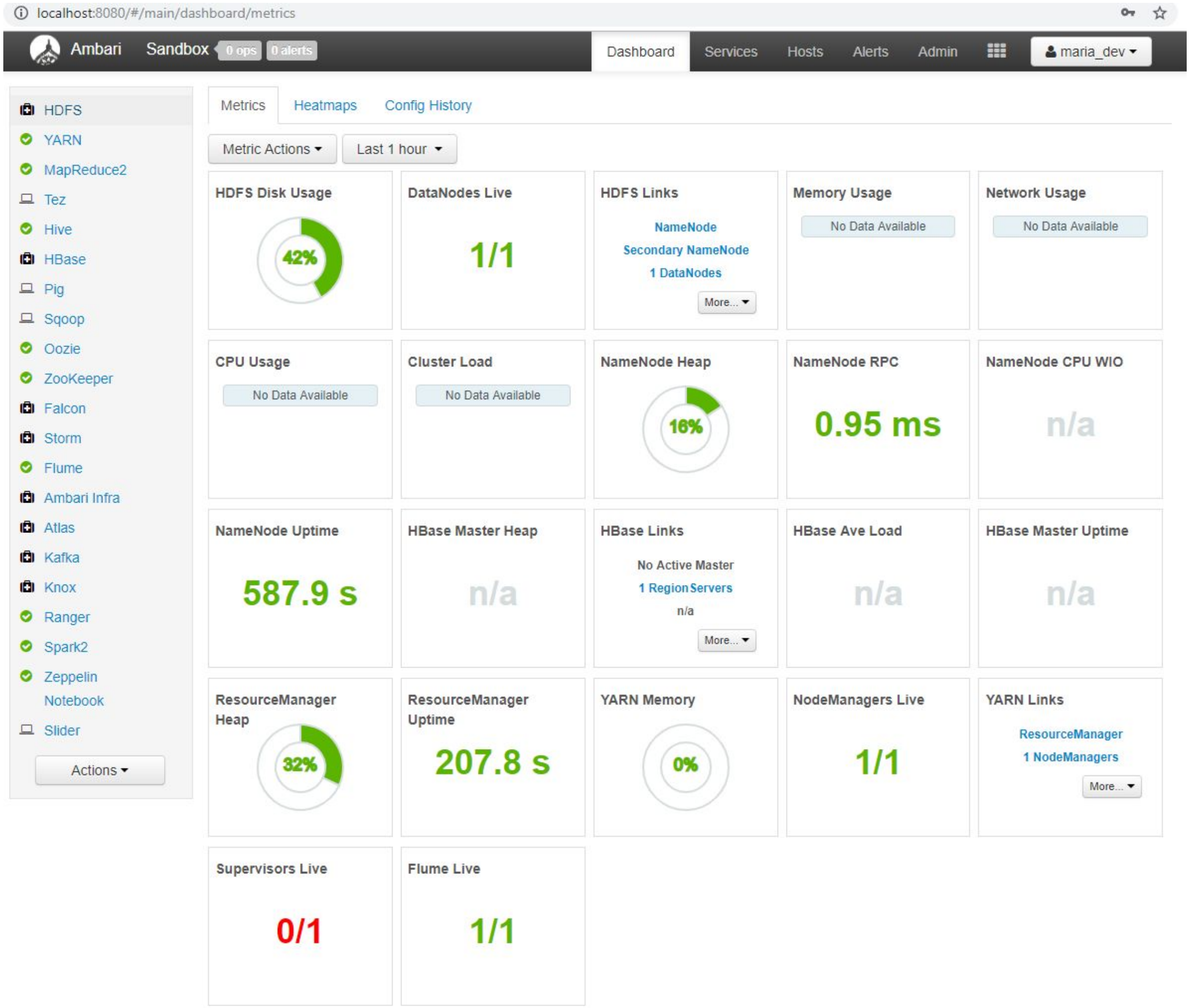
14. If you quickly go to http://localhost:8080, you may get a 502 Bad Gateway error or some other error message. This is normal and it simply means your Hortonworks system is not started completely yet. Wait a few minutes until it is loaded up.
15. Open up an Internet browser and go to http://localhost:8080 and type in **maria_dev** for the username and the password. This username will be our main username throughout the semester. If you log in too quickly, you might see something like these:


Ambari
Sandbox
0 ops
0 alerts
Dashboard
Services
Hosts
Alerts
Admin
maria_dev
HDFS
YARN
MapReduce2
Tez
Hive
HBase
Pig
Sqoop
Oozie
ZooKeeper
Falcon
Storm
Flume
Ambari Infra
Atlas
Kafka
Knox
Ranger
Spark2
Zeppelin Notebook
Slider
Actions
Metrics
Heatmaps
Config History
Metric Actions
Last 1 hour
HDFS Disk Usage
n/a
DataNodes Live
n/a
HDFS Links
NameNode
Secondary NameNode
1 DataNodes
More...
Memory Usage
No Data Available
Network Usage
No Data Available
CPU Usage
No Data Available
Cluster Load
No Data Available
NameNode Heap
n/a
NameNode RPC
n/a
NameNode CPU WIO
n/a
NameNode Uptime
n/a
HBase Master Heap
n/a
HBase Links
No Active Master
1 Region Servers
n/a
More...
HBase Ave Load
n/a
HBase Master Uptime
n/a
ResourceManager Heap
n/a
ResourceManager Uptime
n/a
YARN Memory
n/a
NodeManagers Live
n/a
YARN Links
ResourceManager
1 NodeManagers
More...


Ambari
Sandbox
2 ops
0 alerts
Dashboard
Services
Hosts
Alerts
Admin
maria_dev
HDFS
YARN
MapReduce2
Tez
Hive
HBase
Pig
Sqoop
Oozie
ZooKeeper
Falcon
Storm
Flume
Ambari Infra
Atlas
Kafka
Knox
Ranger
Spark2
Zeppelin Notebook
Slider
Actions
Metrics
Heatmaps
Config History
Metric Actions
Last 1 hour
HDFS Disk Usage
n/a
DataNodes Live
n/a
HDFS Links
NameNode
Secondary NameNode
1 DataNodes
More...
Memory Usage
No Data Available
Network Usage
No Data Available
CPU Usage
No Data Available
Cluster Load
No Data Available
NameNode Heap
n/a
NameNode RPC
n/a
NameNode CPU WIO
n/a
NameNode Uptime
n/a
HBase Master Heap
n/a
HBase Links
No Active Master
1 Region Servers
n/a
More...
HBase Ave Load
n/a
HBase Master Uptime
n/a
ResourceManager Heap
n/a
ResourceManager Uptime
n/a
YARN Memory
n/a
NodeManagers Live
n/a
YARN Links
ResourceManager
1 NodeManagers
More...

This is normal and there is no concern. Just let the system load itself up. It may take several minutes but eventually the yellow/red alerts will disappear and everything will be running. If you still have red alerts to the right of the software titles after 15 minutes, then something is likely wrong. See troubleshooting at the end for possible fixes.

Once everything is loaded up, feel free to poke around the dashboard to see what options you have. There isn't much to do right now but we will be using many of these software tools in this course. Your final running page will look something like this:

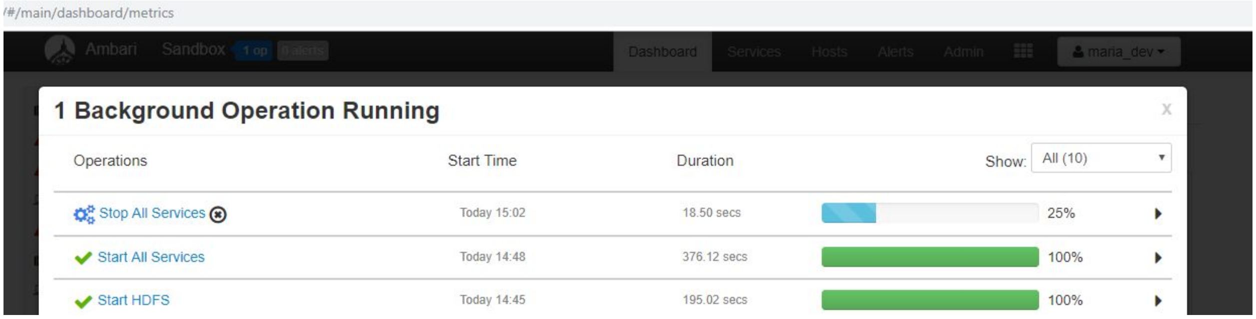


HDFS Disk Usage: 42%
DataNodes Live: 1/1
HDFS Links: NameNode, Secondary NameNode, 1 DataNodes
Memory Usage: No Data Available
Network Usage: No Data Available
CPU Usage: No Data Available
Cluster Load: No Data Available
NameNode Heap: 16%
NameNode RPC: 0.95 ms
NameNode CPU WIO: n/a
NameNode Uptime: 587.9 s
HBase Master Heap: n/a
HBase Links: No Active Master, 1 Region Servers, n/a
HBase Ave Load: n/a
HBase Master Uptime: n/a
ResourceManager Heap: 32%
ResourceManager Uptime: 207.8 s
YARN Memory: 0%
NodeManagers Live: 1/1
YARN Links: ResourceManager, 1 NodeManagers
Supervisors Live: 0/1
Flume Live: 1/1

When you are finished exploring, feel free to shutdown the virtual machine. To do this, go to your VirtualBox Hortonworks window, click **File**, then **Close** and finally choose to **Send the shutdown signal**. If you choose any of the other options, your system may end up in a bad state.

Troubleshooting Hortonworks. If you find your machine in a bad state and your programs are taking a long time to run or simply aren't running at all, these are the best tips to getting back into a good state. These are essentially identical to what you saw in the last task. You do not have to do these at the beginning but just remember these tips are here for future reference.

If you find there are several alerts or something didn't load up correctly, you are encouraged to debug this by simply restarting the services on the virtual machine. If you find that Hadoop, Hive, Pig, etc. are not functioning properly, restarting all of the services will often help. To do this, click on the **Actions** button in the lower left hand corner. Click on **Stop All**. Click on the **Confirm Stop** button that pops up. It will likely take a couple of minutes for all services to stop:



1 Background Operation Running

Operations	Start Time	Duration		
Stop All Services	Today 15:02	18.50 secs		25%
✓ Start All Services	Today 14:48	376.12 secs		100%
✓ Start HDFS	Today 14:45	195.02 secs		100%

Once all services have stopped, click the **OK** button. To start up the services, click on the **Actions** button, click on **Start All** then click on **Confirm Start**. Your services are now restarting.

If the previous step does not solve your issue, then stopping the entire machine and restarting it is the next thing to try. Stop your virtual machine by using the steps on the previous page.

If restarting your machine does not solve the problem, then the fastest solution is to delete your virtual machine and create a new one. If some step was skipped or your virtual machine got itself into a bad state, then troubleshooting the issue could take hours. However, deleting your current virtual machine and creating a new one takes minutes. Just be aware that if you delete your virtual machine, any files on that virtual machine will be deleted as well. Be sure to download/copy over anything you want before deleting.

To delete your virtual machine, go back to step 6 and create a new virtual machine.

Project Euler Problem 1 solution

```
#!/usr/bin/python3.6
sum = 0  #create a sum variable, start it off at 0
current_value = 2  #check all numbers from 2 to 999
while current_value < 1000:
  #if the current number is divisible by 3 or 5, add to sum
  if current_value % 3 == 0 or current_value % 5 == 0:
    sum = sum + current_value
  current_value = current_value + 1
print(sum)
```