COMP 330 Lab 1: Using Amazon EMR to Run a Hadoop Job

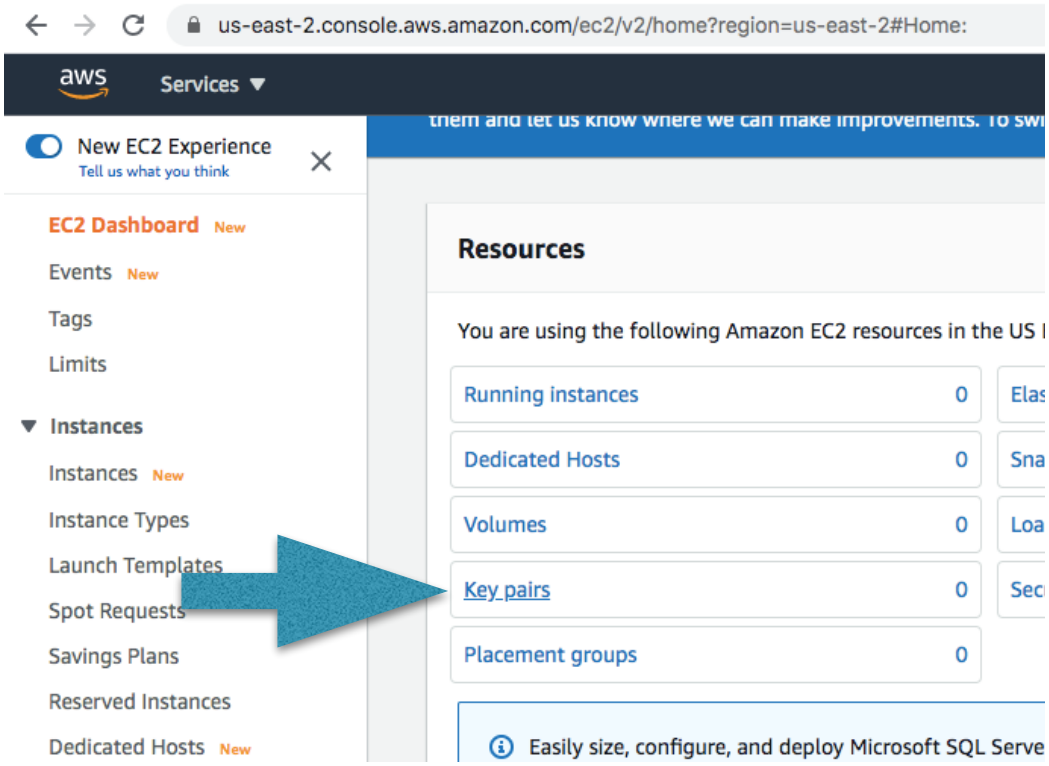
Note: this assumes you have previously signed up for an Amazon account. See Piazza!

Hadoop is a commonly-used, open-source, MapReduce software. In this lab, you will (1) Compile a Hadoop MapReduce program using the Java compiler (Hadoop is a popular open source MapReduce tool). (2) Create a Hadoop cluster using Amazon AWS. (3) Load data into HDFS (this is Hadoop's distributed file system). (4) Run your Hadoop program to process the data.

Task 1: Start Up a Hadoop Cluster

1) Go to Amazon's AWS website (`aws.amazon.com`).

2) Sign in with your user name and password. Click "Services" at the top of the page, and find "EC2" under Compute.

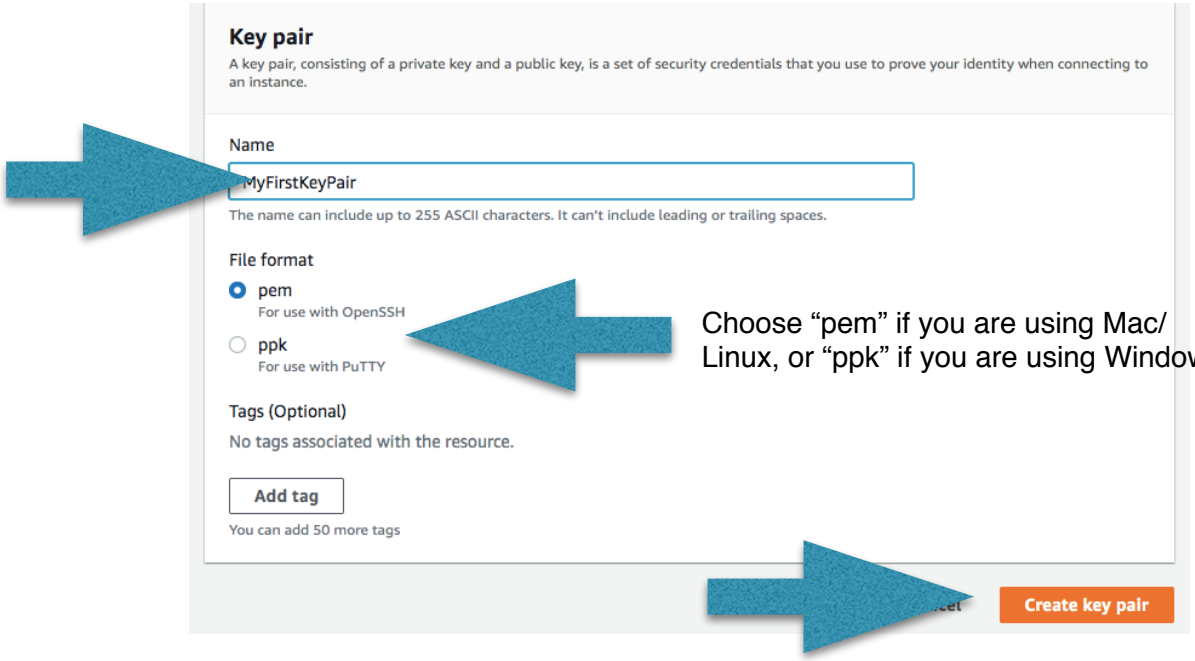


3) Next, you will need to create a "key pair" that will allow you to connect securely to the cluster that you create.

4) Click "key pairs".

5) Click "Create Key Pair".

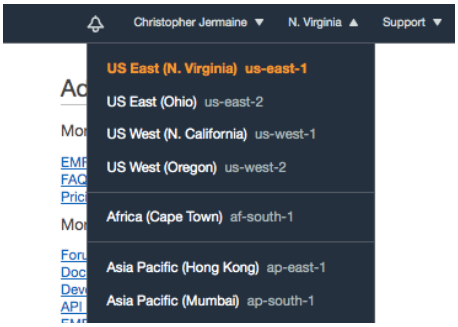
6) Pick a key pair name that is likely unique to you (such as the name of your eldest child, or your last name, so that it is unlikely that you will forget it). Type it in, and click “Create”.



7) This should create a “.pem” file that you can download (or “.ppk” for Windows; see above). You will subsequently use this .pem/.ppk file to connect securely to a machine over the internet. Make sure to save it somewhere that you can find it.

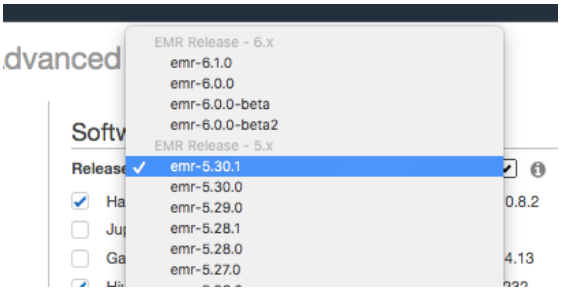
8) Now it is time to create your Hadoop cluster. Again choose “services” at the top of the window and choose “EMR” under “Analytics”. “EMR” stands for “Elastic Map Reduce.”

9) Make sure that “N. Virginia” is selected at the top right.

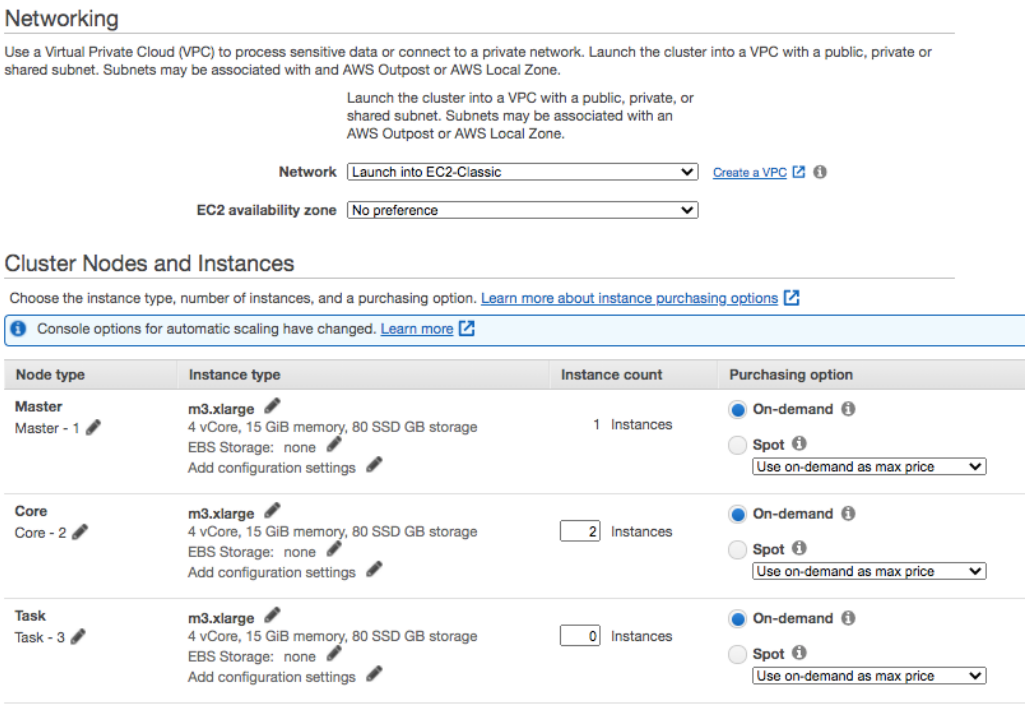


Click “Create cluster”. Choose “Go to advanced options”.

10) Choose emr5.30.1, then click “Next”.

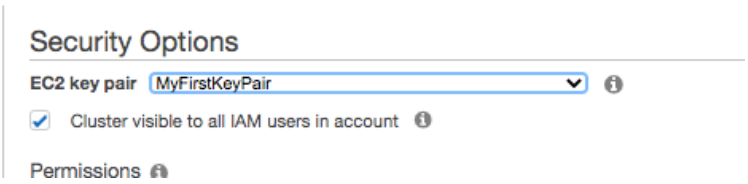


11) On the next page, all of the defaults should be OK. For the Master node, you want an m3.xlarge machine. If you are interested, you can find a list of all instance types at `https://aws.amazon.com/ec2/instance-types/`. Each m3.xlarge machine has 4 CPU cores and 8GB of RAM. For the Core workers, you want 2 m3.xlarge machines. Click next.



12) On the next page (“General options”) everything should be OK, except you can unclick “termination protection”. Click next.

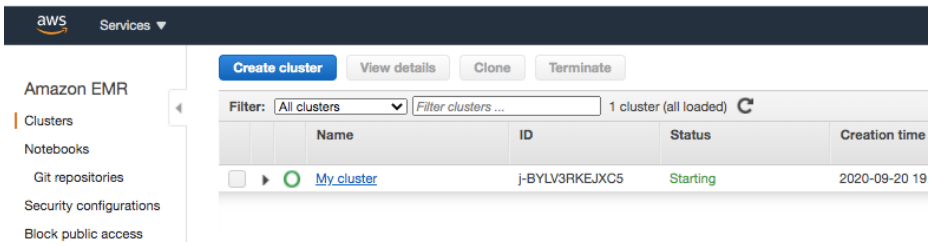
13) On the next page, under EC2 key pair, it is really important to choose the EC2 key pair that you just created. This is important: if you do not do this, you won’t be able to access your cluster.



14) Click “Create Cluster”. Now your machines are being provisioned in the cloud! You will be taken to a page that will display your cluster status. It will take a bit of time for your cluster to come online. You can check the status of your cluster by clicking the little circular update arrow at the top right.

Note: the very first time that create a cluster, it may take 15 minutes or more for the cluster to begin, and Amazon makes sure your account is valid. Take the opportunity to update your Facebook or chat with your neighbor. As soon as your master node changes to “bootstrapping”, you are ready to go.

Note: if you ever want to get back to the page that lists all of your EMR clusters, just click the “AWS” at the top left, then enter or click “EMR”. You will see this:

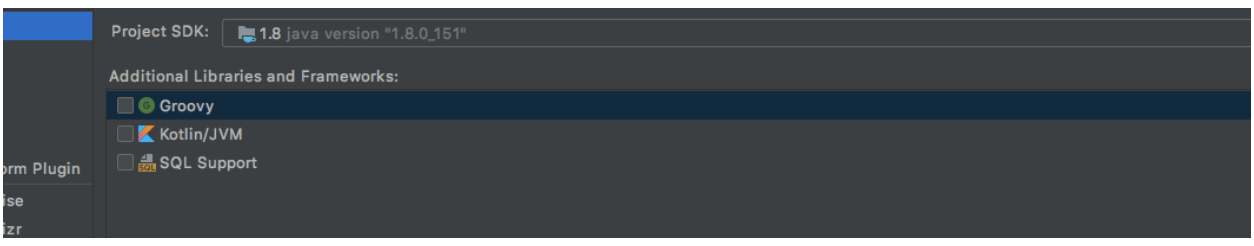


In the meantime, you can start on Task 2.

Task 2: Compile a Hadoop program

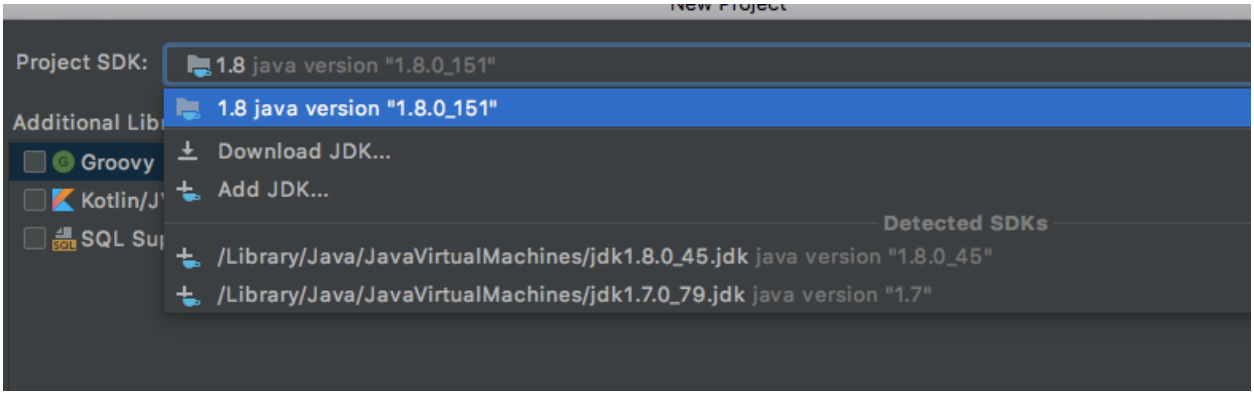
1) Create a new project in IntelliJ.

Make sure you are using Java 8 (that is, version 1.8).



Note that if you don't have version 1.8, you may have to download it. Go here: https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.html

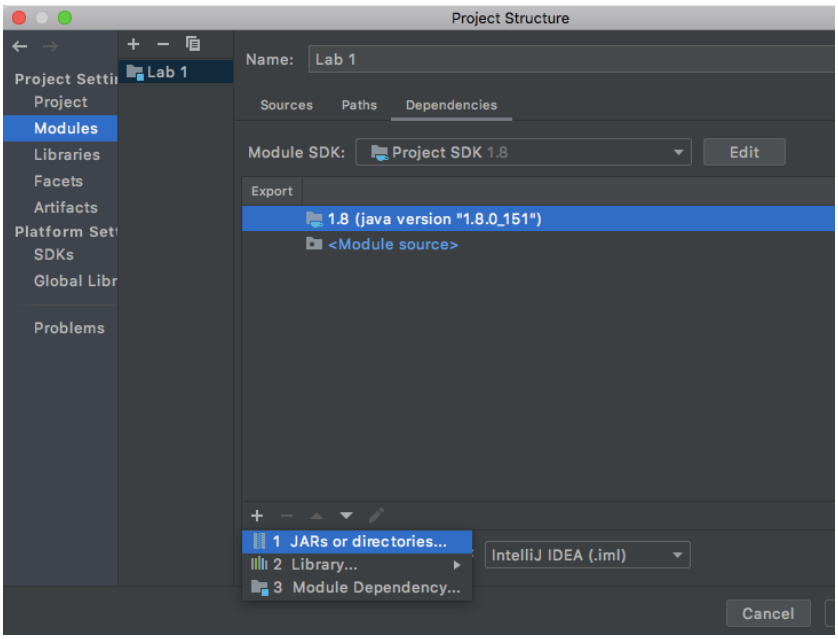
Then you can choose to add that JDK to IntelliJ as follows:



Don't create your project from a template, but do call your project "Lab 1".

2) Go to the directory `http://cmj4.web.rice.edu/HadoopActivity` and download all of the jar files included (make sure to download seven of them!). Put them into a directory on your machine. Jar files like this contain compiled code. These are all of the jars that you will need to compile your Hadoop program on your machine.

3) Now, add those jar's to your project. Choose "File->Project Structure" and then do the following (make sure to press the "+:" to get to the final menu):

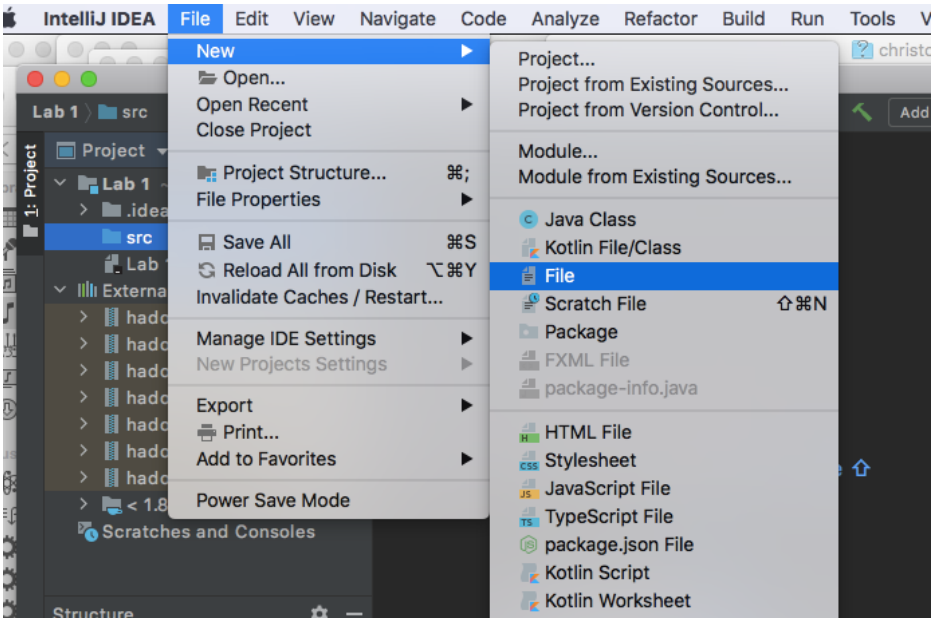


Then choose the seven jars that you should have downloaded:

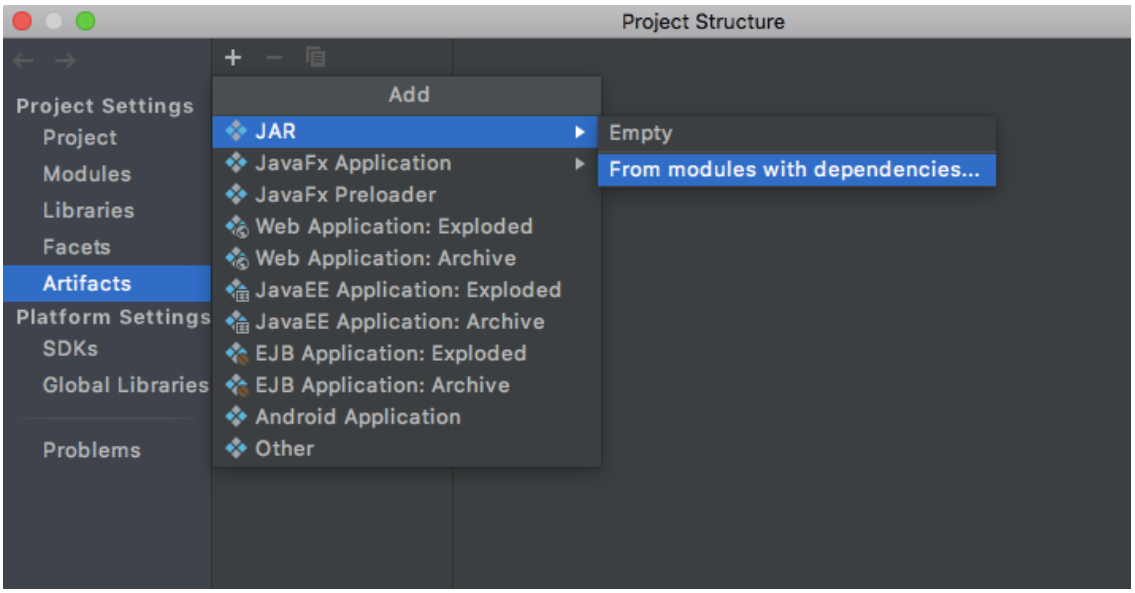
`hadoop-annotations-2.7.3.jar`

```
hadoop-client-2.7.1.jar
hadoop-common-2.7.3.jar
hadoop-hdfs-2.7.3.jar
hadoop-mapreduce-client-app-2.7.3.jar
hadoop-mapreduce-client-core-2.7.3.jar
hadoop-yarn-api-2.7.3.jar
```

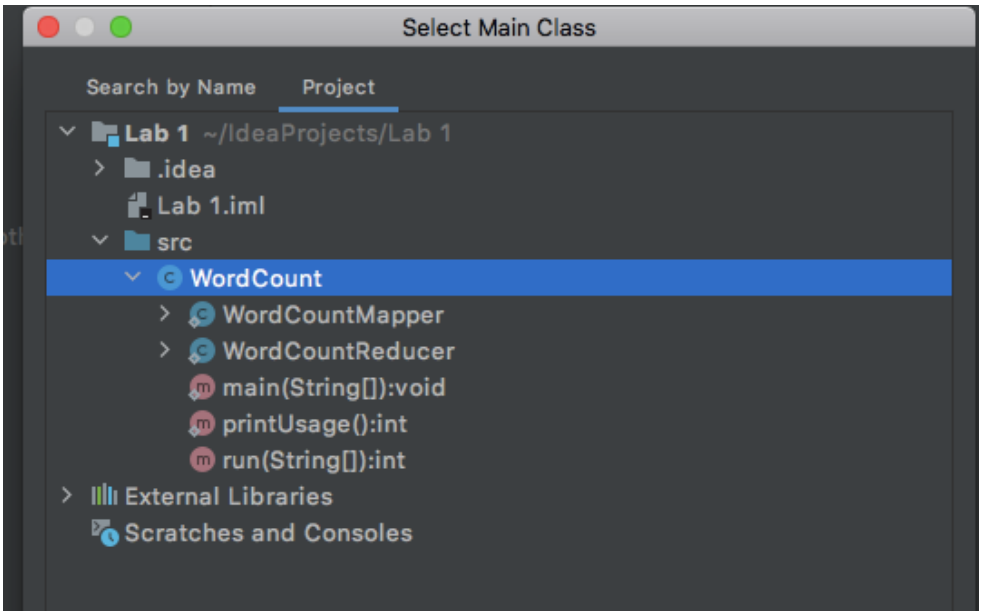
4) Now, highlight “src” and then create a new file in your project called `WordCount.java`:



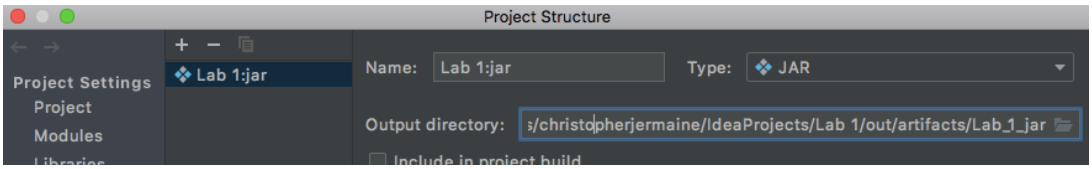
5) From `http://cmj4.web.rice.edu/HadoopActivity`, open `WordCount.java`. Copy and paste the contents of this file into the `WordCount.java` file in your project. Again, make sure to press the “+” to get to the “Add” menu:



Very important: you need to correctly choose your main class correctly; it should be WordCount:

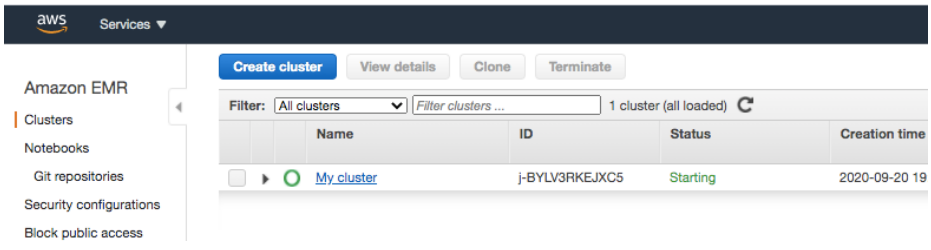


After you choose WordCount, hit “OK” a few times to create the jar. Make sure to pay attention to where you put the jar; you can choose a different location if you want. You’ll need to be able to find it again:



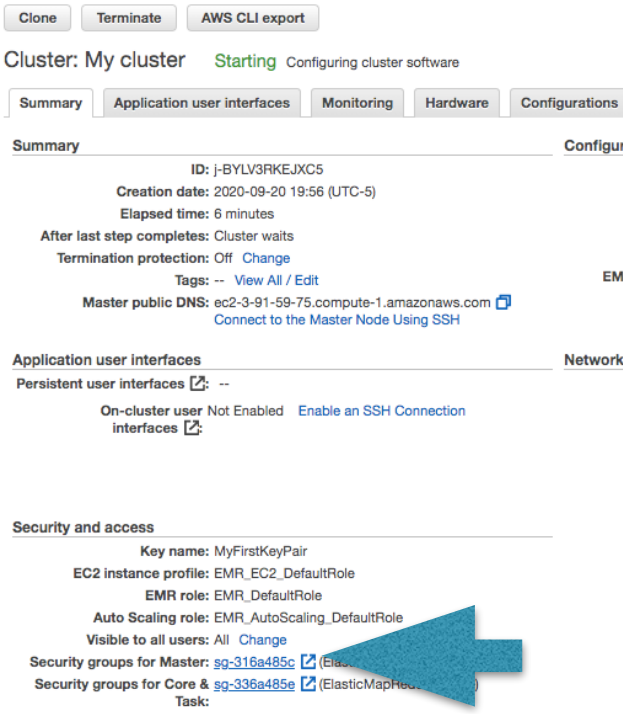
6) Now, choose “Build->Build Artifacts…” then press “Build”. This will actually create Lab_1.jar inside of the directory that you specified above. Note that if you ever change the code and want to make sure that the jar has your updated code, you’ll need to do this again.

7) Time to go back to AWS. Click the “AWS” at the top left, then enter or click “EMR”. You will see this:

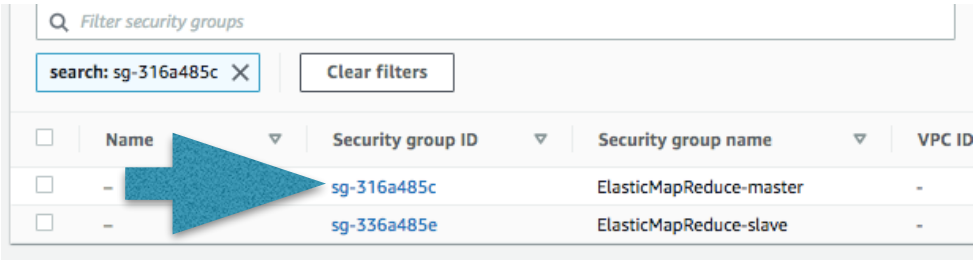


One your cluster is up and running, you will want to connect to the master node so that you run Hadoop jobs on it. Click on your cluster.

8) You need to make it so that you can connect via SSH. Under "Security and Access" (not a tab on the side, just a heading at the lower right), go to "Security groups for Master" and click on the link.

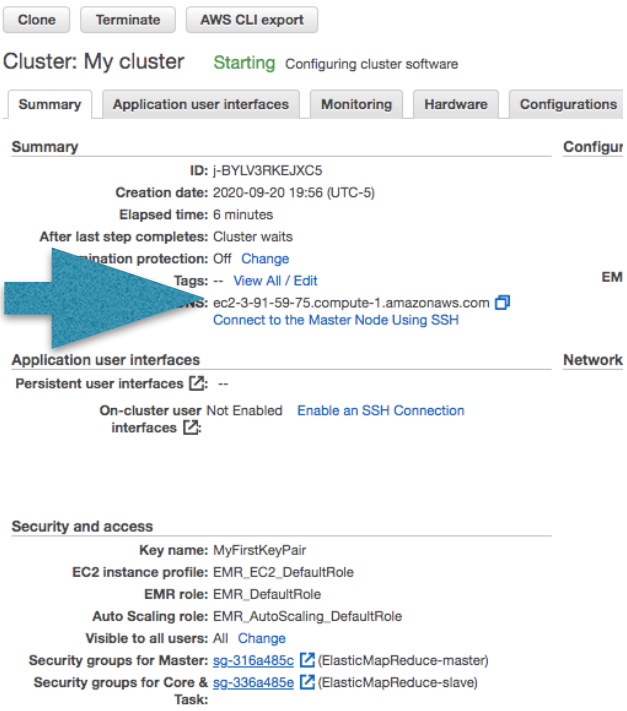


In the new page, click on the row with Group Name = "ElasticMapReduce-master".



At the bottom, click on the Inbound tab. Click on “Edit Inbound Rules”. Click "Add Rule” at the bottom. Then select “SSH” in the first box and “Anywhere” in the second. Click save.

6) Now you need to connect. Again go to your cluster. Locate the address of your master node, which is what you’ll connect to:



7) Now that you have created your cluster, and identified the address of your master node, it is time to connect to the node and run a Hadoop job! To connect to your master node:

Mac/Linux. The following assumes that your .pem file is called `MyFirstKeyPair.pem` and that it is located in your working directory; replace this with the actual name and location of your file, assuming that you called your key pair something else. First, verify that the .pem file is in your directory: Type:

```
$ ls MyFirstKeyPair.pem
```

The result should be:

```
MyFirstKeyPair.pem
```

If you don't see this, there is a problem, as it means you are not in the correct directory, where your .pem file is located, or you forgot to save your .pem file, or something else went wrong. Then type:

```
$ chmod 500 MyFirstKeyPair.pem
```

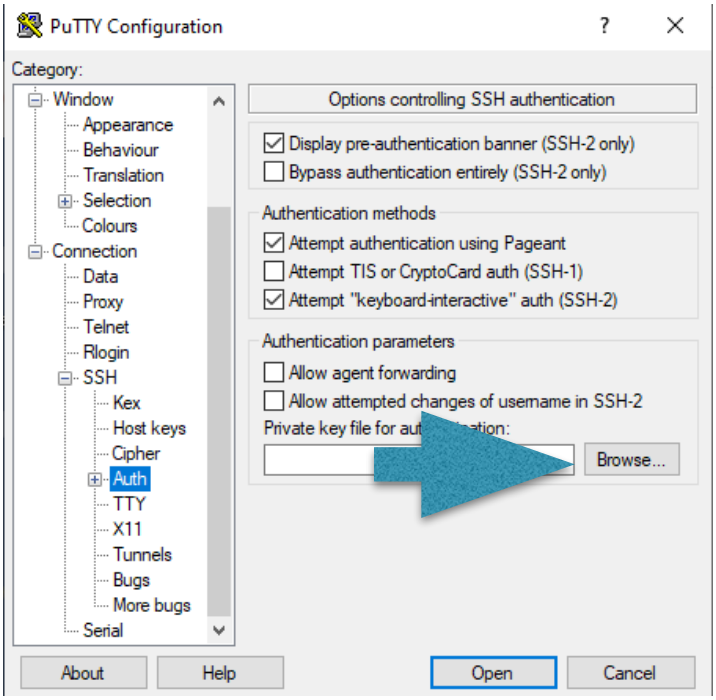
Now, you can connect to your master machine (replace "ec2-3-91-59-75.compute-1.amazonaws.com" with the address of your own master machine):

```
$ ssh -i MyFirstKeyPair.pem
hadoop@ec2-3-91-59-75.compute-1.amazonaws.com
```

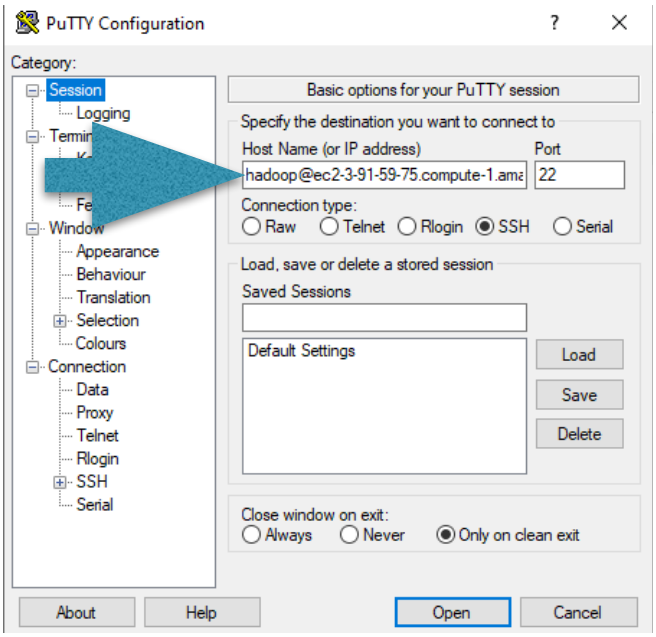
This will give you a Linux prompt; you are connected to your master node. It should look like the following:

```
(base) Christophers-MacBook-Air-2:Downloads christopherjermaine$ ssh -i ~/Downlo
227-100-198.compute-1.amazonaws.com

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
33 package(s) needed for security, out of 90 available
Run "sudo yum update" to apply all updates.

EEEEEEEEEEEEEEEEEEE MMMMMMMM           MMMMMMMM RRRRRRRRRRRRRRR
E::::::::::::::::::E M:::::::M         M:::::::M R::::::::::::::R
EE:::::EEEEEEEEE:::E M::::::::M       M::::::::M R:::::RRRRRR:::::R
  E::::E       EEEEE M:::::::::M     M:::::::::M RR::::R      R::::R
  E::::E             M::::::M:::M   M:::M::::::M   R::::R      R::::R
  E:::::EEEEEEEEEE   M:::::M M:::M M:::M M:::::M   R::::RRRRRR:::::R
  E::::::::::::::E   M:::::M  M:::M:::M  M:::::M   R:::::::::::RR
  E:::::EEEEEEEEEE   M:::::M   M:::::M   M:::::M   R::::RRRRRR::::R
  E::::E             M:::::M    M:::M    M:::::M   R::::R      R::::R
  E::::E       EEEEE M:::::M     MMM     M:::::M   R::::R      R::::R
EE:::::EEEEEEEE:::E M:::::M             M:::::M   R::::R      R::::R
E::::::::::::::::::E M:::::M             M:::::M RR::::R      R::::R
EEEEEEEEEEEEEEEEEEE MMMMMMM             MMMMMMM RRRRRRR      RRRRRR

[hadoop@ip-10-182-151-146 ~]$
```

Windows. In Windows, we'll assume that you are using the PuTTY tools (a Google search on "download putty.exe" is all you need to get it on your laptop. After you download PuTTY, fire it up. In the left-hand side of the dialog that comes up, click "Connection" then "ssh" then "auth" and then click on "Browse" to select the private key file (.ppk file) that you earlier downloaded from Amazon:



Once you have selected your private key, and hit "Open", go back to the "Session" and enter in your machine info (note the "hadoop@…") and again hit "Open":



8) Now, whether or not you are using Windows or Mac/Linux, you will have a Linux prompt to your master node. It is time to run a Hadoop job!

Task 3: Run a Hadoop Program and Get Checked Off

1) Transfer the WordCount jar that you created over to your master node so that you can run it.

Mac/Linux. In Mac or Linux, figure out where your .jar file is located. For me it is at `"ideaProjects/Lab 1/out/artifacts/Lab_1_jar/"`; Wherever yours is located, find it, and then you can verify easily. In my case, I type:

```
$ ls "ideaProjects/Lab 1/out/artifacts/Lab_1_jar/"
```

It should come back with `Lab_1.jar`. Copy your .pem file into this directory. Go to wherever your .pem file is located, and then type:

```
$ cp MyFirstKeyPair.pem "ideaProjects/Lab 1/out/artifacts/Lab_1_jar/"
```

Then go to that directory:

```
$ cd "ideaProjects/Lab 1/out/artifacts/Lab_1_jar/"
```

Type the following to fire up the secure ftp program:

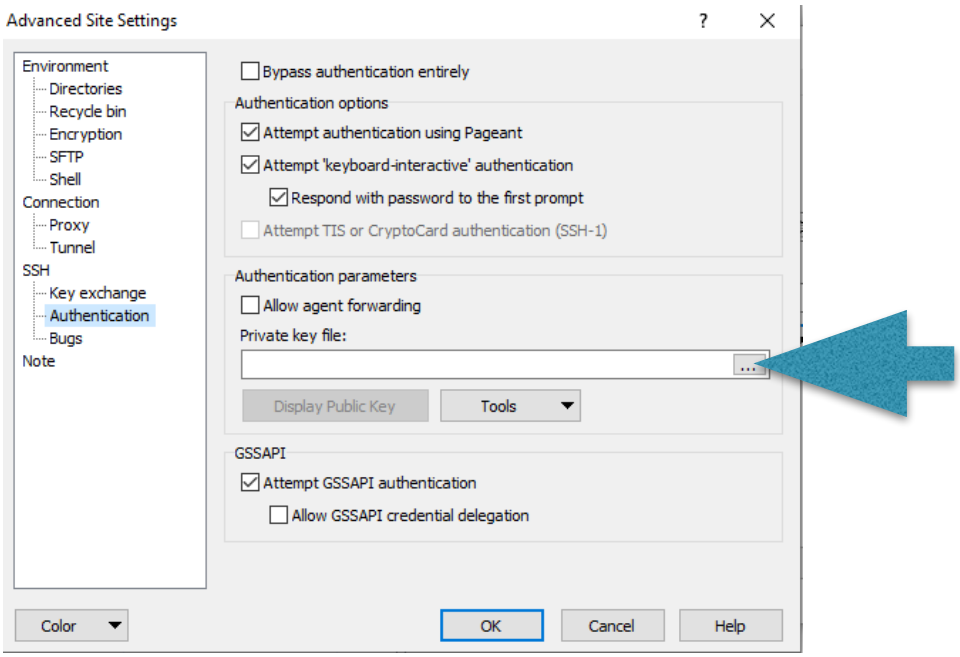
```
sftp -i MyFirstKeyPair.pem
hadoop@ec2-3-91-59-75.compute-1.amazonaws.com
```

Then type:

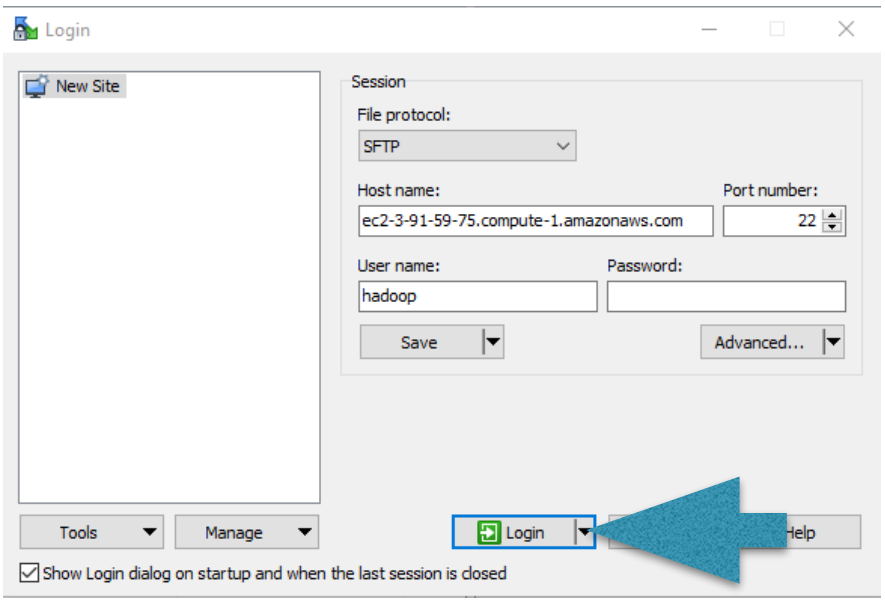
```
put "Lab 1.jar"
```

to upload your jar. Type “exit” to exit the program.

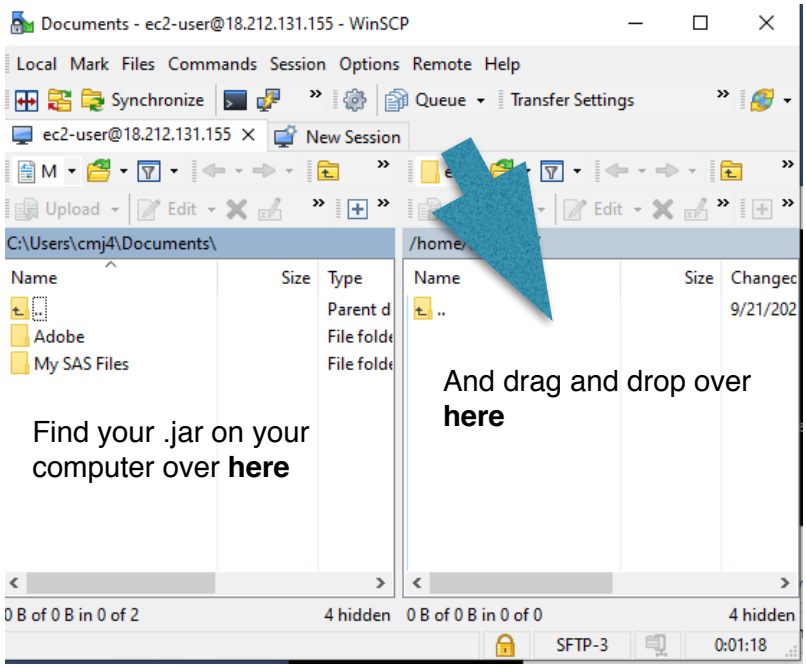
Windows. Download the WinSCP tool from the inernet. Hit the “Advanced” button, and choose your private key (.ppk) file:



Hit “OK” after you choose your private key. Enter in the IP address, and the user name “hadoop”, then press “Login”



WinSCP will connect to the master, and you can use WinSCP's graphical user interface to transfer files to it. Transfer over Lab 1.jar.



2) Load a bunch of data onto your master node. Go back to the command prompt on your master machine (the one opened via SSH). Start by typing:

```
wget https://s3.amazonaws.com/chrisjermainebucket/text/Holmes.txt
wget https://s3.amazonaws.com/chrisjermainebucket/text/dictionary.txt
wget https://s3.amazonaws.com/chrisjermainebucket/text/war.txt
wget https://s3.amazonaws.com/chrisjermainebucket/text/william.txt
```

This will load the four large text files on to your master node. You can look at the contents by (for example) typing “more Holmes.txt” (`control-c` to stop looking, spacebar to see more).

3) Load this data into HDFS. Type:

```
[hadoop@ip-10-147-46-240 ~]$ hadoop fs -mkdir words
[hadoop@ip-10-147-46-240 ~]$ hadoop fs -put *.txt words
```

4) Run word count! From the command line, simply type:

```
[hadoop@ip-10-147-46-240 ~]$ hadoop jar "Lab 1.jar" -r 8 words
wordsOutput
```

This will run the word count with 8 mappers and 8 reducers. A bunch of information will scroll by. After a few seconds, the computation is done.

5) Check out your results. Type:

```
[hadoop@ip-10-147-46-240 ~]$ hadoop fs -ls wordsOutput
```

And you will see something like:

```
Found 9 items
-rw-r--r--   1 hadoop hadoop          0 2016-09-20 16:21 wordsOutput/_SUCCESS
-rw-r--r--   1 hadoop hadoop      19819 2016-09-20 16:21 wordsOutput/part-r-00000
-rw-r--r--   1 hadoop hadoop      20272 2016-09-20 16:21 wordsOutput/part-r-00001
-rw-r--r--   1 hadoop hadoop      18979 2016-09-20 16:21 wordsOutput/part-r-00002
-rw-r--r--   1 hadoop hadoop      20431 2016-09-20 16:21 wordsOutput/part-r-00003
-rw-r--r--   1 hadoop hadoop      19906 2016-09-20 16:21 wordsOutput/part-r-00004
-rw-r--r--   1 hadoop hadoop      19326 2016-09-20 16:21 wordsOutput/part-r-00005
-rw-r--r--   1 hadoop hadoop      20032 2016-09-20 16:21 wordsOutput/part-r-00006
-rw-r--r--   1 hadoop hadoop      19253 2016-09-20 16:21 wordsOutput/part-r-00007
```

6) Note that it's OK if your's looks a bit different. Copy some of the results from HDFS to the master node. Type:

```
[hadoop@ip-10-147-46-240 ~]$ hadoop fs -get wordsOutput/part-r-00001 .
```

Typing "more part-r-00001" will allow you to look at some of the counts! Pressing the space bar allows you to page through this file; typing "q" exits.

7) Copy and paste one of the pages to Canvas to get checked off.

Task 4: **SHUT DOWN YOUR CLUSTER**

Important: never leave your cluster up when you are not using it. **You are being charged!**

1) From the web page for your cluster, click "Terminate".

2) If "Termination Protection" is on, you will have to turn it off before you kill your machines.

3) Note: I've had mixed results actually killing machines in this way. After you kill them, **make sure** that they are dead. Click the cube, click "EC2" and click "Running Instances". There should not be any. If they are still there, click on "Running Instances". Then click the checkbox next to each of your machines, and under "Actions"->"Instance State" choose "terminate". **Only log out after you have verified from the EC2 page that you have no running instance**.