Spark AWS lab

1 Description

This section is a high level overview.
In this lab, you will:

1. Create an AWS EMR cluster
2. Upload a PySpark program onto the cluster
3. Run a Spark job
4. Check out the output on HDFS (Hadoop Distributed File System).

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

2 Create your Key Pair

To connect to the cluster you will create later, you need a Key Pair as an identity.

1. Go to Amazons AWS website (aws.amazon.com).
2. Sign in with your user name and password. Go to EC2 (you can reach EC2 from the AWS services search text box).
3. Click "key pairs".
4. Click "Create Key Pair".
5. 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".
6. This should create a ".pem" file that you can download. You will subsequently use this .pem file to connect securely to a machine over the internet.

3 Start Up a Cluster

1. Click the AWS at the upper left of the dashboard to get back to the main menu. Then, search for or click "EMR". This stands for "Elastic Map Reduce."
2. Click "Create cluster".
3. Choose the software configuration that includes Spark 2.4.0, as shown in Figure 1
4. For the Master node, you want an m4.large machine. If you are interested, you can find a list of all instance types at https://aws.amazon.com/ec2/instance-types/. Each m4.large machine has 4 CPU cores and 8GB of RAM. For the Core workers, you want 2 m4.large machines.
5. Under "Security and access", it is really important to choose the EC2 key pair that you just created. **This is important: if you do not do this, you wont be able to access your cluster.**

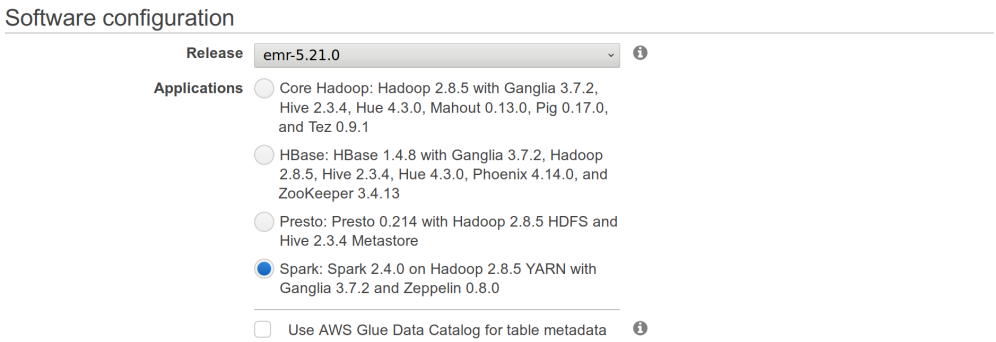

Figure 1: Make sure to ask AWS to load your cluster with Spark!

6. 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.

7. Once your cluster is running, 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 left), 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". Click "Add Rule". Then select "SSH" in the first box and "Anywhere" in the second. Click save.

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".

4 Connect to Master Node

We're going to use a terminal to "ssh" into your spark master. If you are using Mac or Linux you can use your native terminal application. If you are using Windows, you can get to a terminal using Jupyter on ORION.

4.1 Identifying Master Node's Domain

On the "Summary" tab after creating the cluster, locate the section that lists your master node's Public DNS, shown in Figure 2.

4.2 Creating a terminal on Jupyter

To create a terminal on Jupyter, simply launch a server from the `gvacaliuc/db-notebook` image, and select New → Terminal, shown in Figure 3

4.3 Connecting from your terminal

The following assumes that your .pem file is called MyFirstKeyPair.pem; **replace this name with the actual name and location of your file**, assuming that you called your key pair something else. Type:

```
chmod 600 MyFirstKeyPair.pem
```

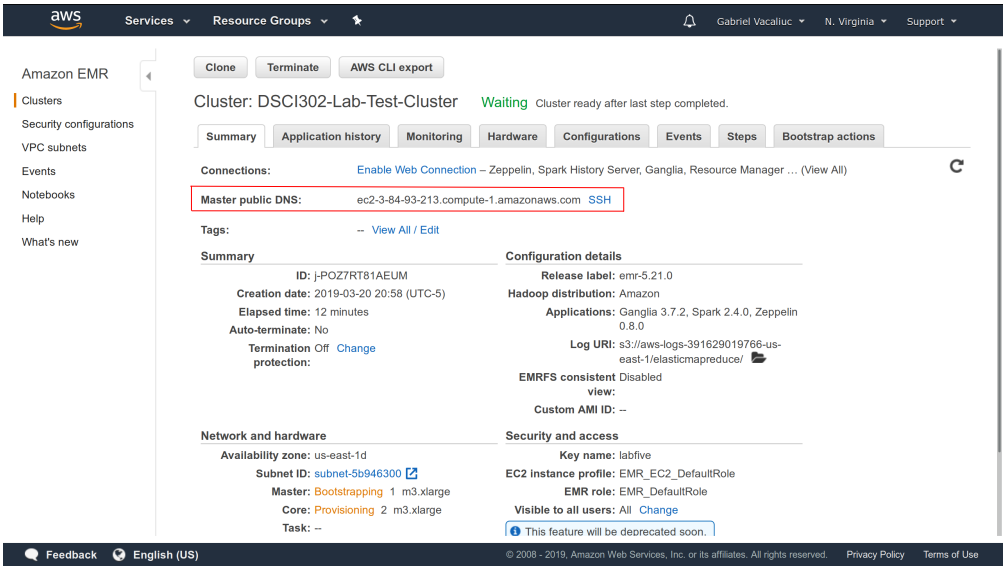

Figure 2: Where to find your master node's Public DNS.

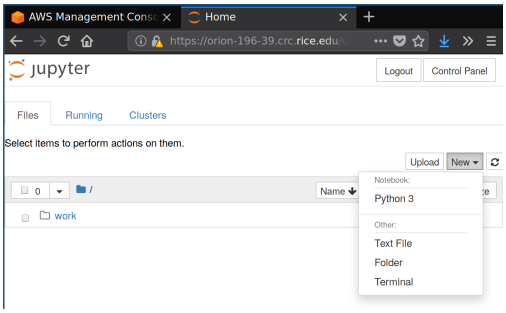

Figure 3: How to create a new terminal from the Jupyter file tree.

Now, you can connect to your master machine. For convenience later, you can set some environment variables, then connect using ssh and your private key:

```
PUBLIC_DNS=ec2-3-84-93-213.compute-1.amazonaws.com # REPLACE THIS WITH YOUR DNS

PRIV_KEY=/path/to/MyFirstKeyPair.pem # REPLACE THIS WITH YOUR PATH

ssh -i $PRIV_KEY hadoop@$PUBLIC_DNS
```

This will give you a Linux prompt on your cluster's master. You can exit the remote machine by typing `exit` at the prompt.

5 Run Spark jobs

It is time to run a Spark job! There are two ways to do this. We will use both.

1. Submit a Spark job

   (a) Download the file "topWords.py" from Canvas and save it in your work directory. If you're working on the ORION terminals, please upload this file there.

   (b) Open a new terminal in the directory you've been working from (either on ORION or your local machine)

   (c) Copy the "topWords.py" file to the remote server:

   ```
   scp -i MyFirstKeyPair.pem /path/to/topWords.py hadoop@$PUBLIC_DNS:~/topWords.py
   ```

   (d) Now run the job! From the command line on the remote machine (follow instructions above to connect if you have since exited), simply type:

   ```
   spark-submit topWords.py
   ```

   This will launch the Spark job. A bunch of information will scroll by. After a few seconds, the computation is done.

   (e) Check out your results. Type:

   ```
   hadoop fs -ls output
   ```

   And you will see something like:

   ```
   Found 5 items
   -rw-r--r--   1 hadoop hadoop          0 2018-10-01 02:39 output/_SUCCESS
   -rw-r--r--   1 hadoop hadoop      90587 2018-10-01 02:39 output/part-00000
   -rw-r--r--   1 hadoop hadoop      95356 2018-10-01 02:39 output/part-00001
   -rw-r--r--   1 hadoop hadoop     101621 2018-10-01 02:39 output/part-00002
   -rw-r--r--   1 hadoop hadoop      92673 2018-10-01 02:39 output/part-00003
   ```

   (f) Copy the results from HDFS to the master node. Type:

   ```
   hadoop fs -get output
   ```

   (g) You can have a look at some of the results by typing:

```
more output/part-00001
```

(h) To download the file, use scp again to retrieve the desired files. Note the "." at the end. This copies the specified file to "here", your current location.

```
scp -i MyFirstKeyPair.pem  hadoop@$PUBLIC_DNS:output/part-00000 .
```

(i) Show one of the graders this file to get checked off.

2. Work with Spark interactively

(a) Copy the `countWords.py` file up to the remote machine. (follow the instructions above we used for uploading `topWords.py`). Note that this should be done from a "local" terminal.

(b) At a terminal on the remote machine (follow instructions above if you have since exited), type `pyspark`. This starts spark and makes the spark context available in your environment as: `sc`.

(c) Now, import the `countWords` method from `countWords.py`:

```
>>> from countWords import countWords
```

(d) Then, run the following command: `countWords(sc, "s3://[DataLocation]/Holmes.txt")`

6 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 instances.

7 FAQ

1. I'm confused. What terminal should I be running these commands on?

   Generally, we'll be working on the remote machine on AWS, but to open a terminal on the remote machine (`ssh`) and / or copy files to it (`scp`), we'll need to use a terminal on your local computer or ORION. So, basically anytime you're running `ssh` or `scp` you should be doing this on a "local" terminal, and AFTER you open a remote terminal using `ssh` to the AWS machine, you should be executing commands (`spark-submit`, `pyspark`, `hadoop fs ...`) on the remote terminal.