**Exercise 4**

*Spark SQL on AWS*

**Prior Knowledge**

Unix Command Line Shell

Simple Python

**Learning Objectives**

Understanding how to run Spark on AWS using Flintrock

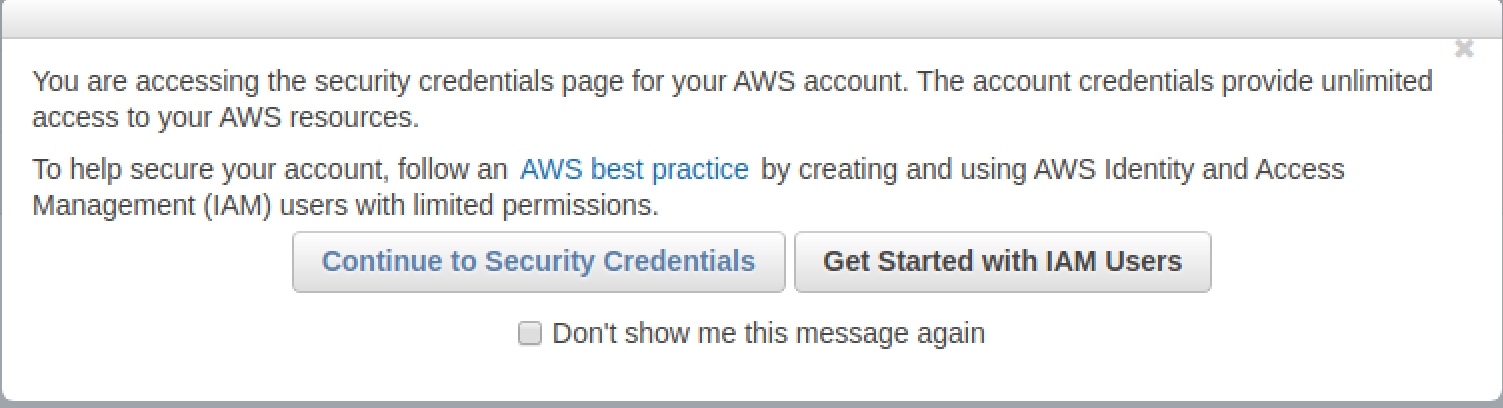
**Software Requirements**

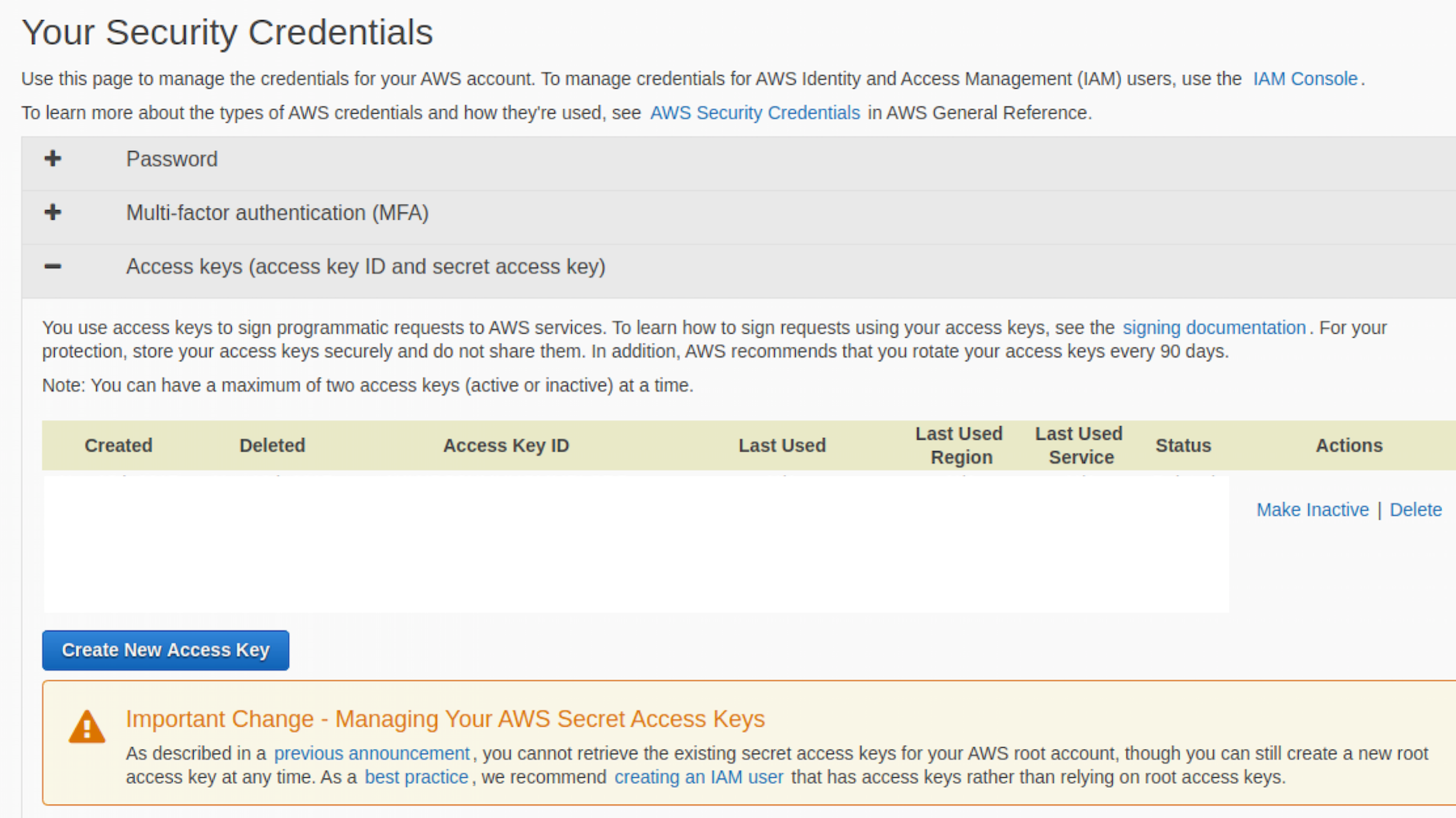
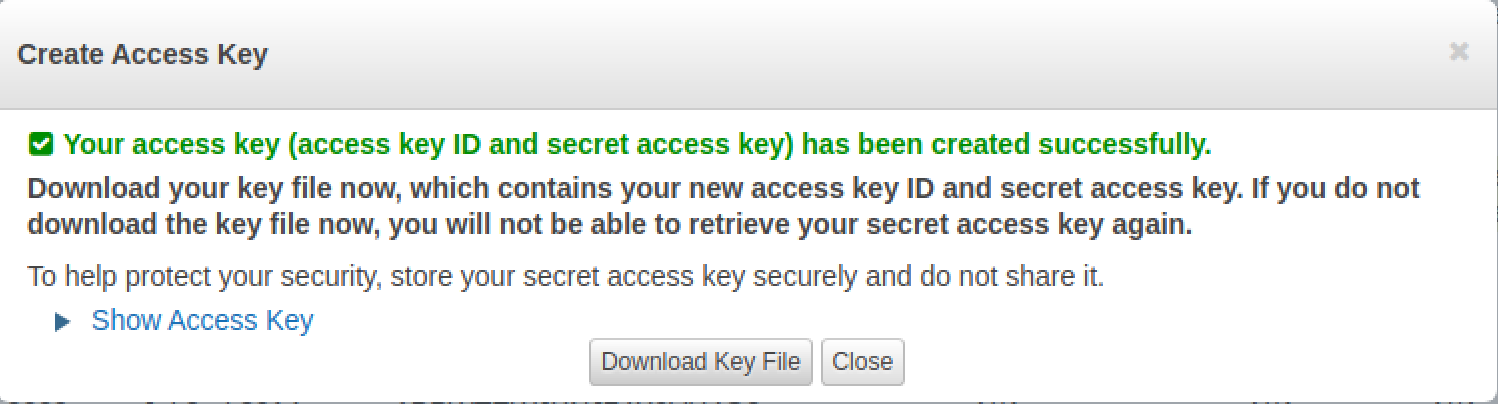
(see separate document for installation of these)

* EC2 credentials
* Flintrock

**Part A. Starting Spark in EC2**

1. Log into your AWS account on the browser, and go to **Security Credentials**. You may get a warning like this:



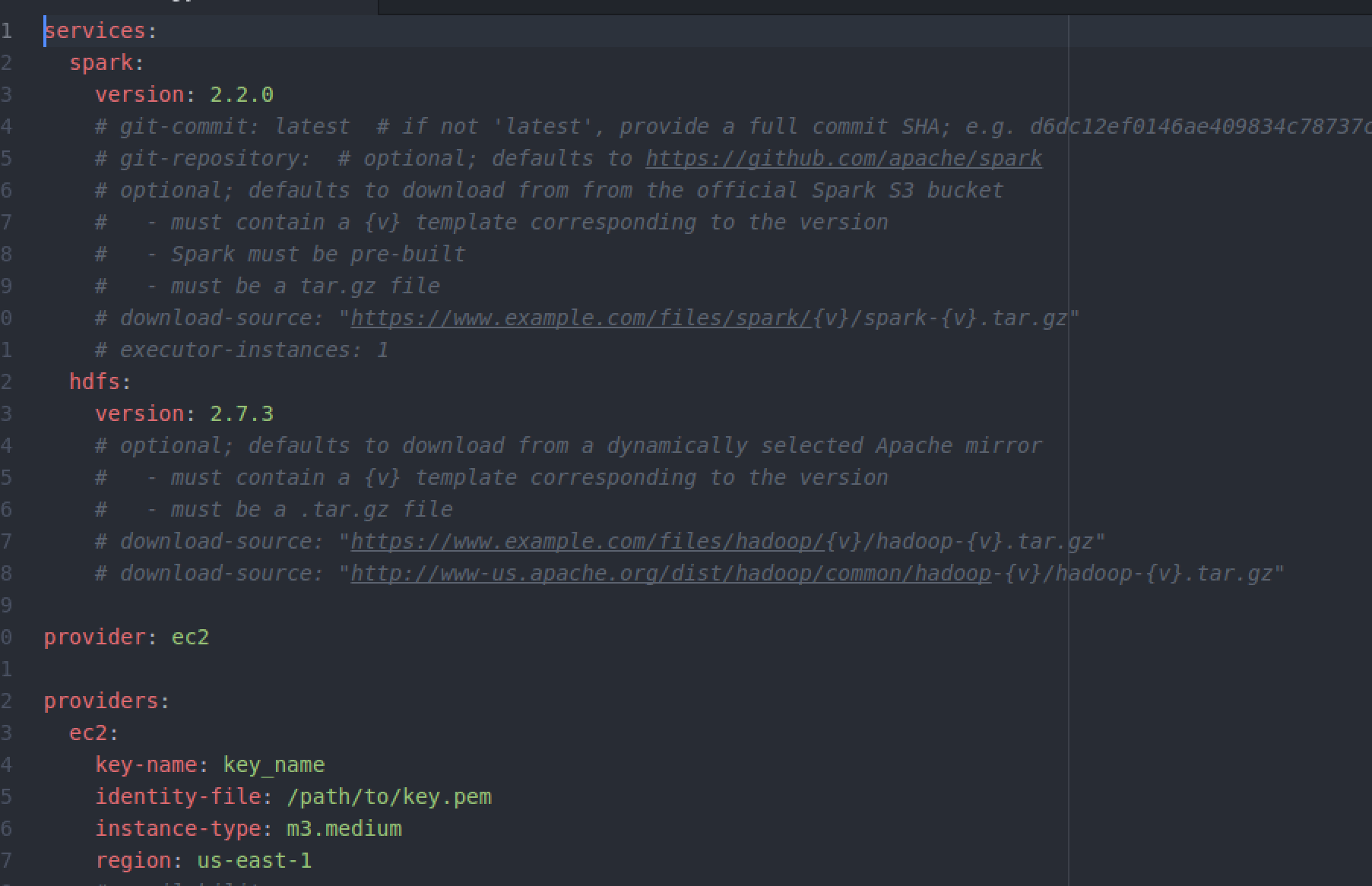
1. Expand the section labeled **Access Keys**(I have blanked out key details in the screen shot)  
   
2. Click **Create New Access Key  
   **
3. **Download the key file**
4. Display the keyfile (e.g. edit it with Atom)
5. On a command line type:  
   aws configure  
     
   You should see:  
   AWS Access Key ID [\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*J3EA]:
6. Copy the Access Key ID from the keyfile, and then hit Enter
7. Do the same for the Secret Access Key
8. Set the default region to **eu-west-1**
9. Set the output to **json**
10. It should like this (but with your keys):

AWS Access Key ID [\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*J3EA]: AKIASF22343434UNM33UVIA   
Secret Access Key [\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*JXb7]: 8z1rtTbU3Ur/llksdafkjhd398u34msndHnGaDY   
Default region name [eu-west-1]: eu-west-1

Default output format [json]: json

1. These keys allow applications to interact with Amazon AWS on your behalf.
2. In addition to these “Access Keys”, we also need an SSH key to continue.   
   If you successfully completed the pre-course Amazon lab,   
   you should have a file ~/keys/bigkp.pem . If not, grab one of the instructors to help you create one.
3. There is a project from the creators of Spark to run it in EC2, but it is not very good! Instead we will use a tool called **flintrock,** which can configure Spark clusters in AWS for you.
4. Before we can use flintrock, you need to modify the config file for flintrock so that it uses your own keys. Edit the flintrock config file:

atom ~/.config/flintrock/config.yaml

It will look something like:  


We need to replace the contents of this file, with one that will work for us.

The source for this is here: <https://freo.me/big-flintrock>

It should look like:

This is modified in a couple of ways. Firstly, it gives the Ireland region and AMI files. Secondly, there is an “instance-profile-name”. This is a AWS feature that gives the running VM access to other APIs - in this case S3.

Finally, I’ve changed the key name and identity file to match your key name and identity file.

1. Copy and paste from the web version to your local version, replacing the existing text.
2. Save the file
3. You should now be able to launch a cluster in Amazon. From a new terminal window  
     
   flintrock launch big
4. Now you should see something like (except with more lines):

Launching 3 instances...

[54.154.17.100] SSH online.

[54.154.17.100] Configuring ephemeral storage...

[54.154.17.100] Installing Java 1.8...

[34.253.201.139] SSH online.

[34.253.201.139] Configuring ephemeral storage...

[34.253.201.139] Installing Java 1.8...

[54.154.17.100] Installing Spark...

[34.253.201.139] Installing Spark...

[34.253.201.139] Configuring Spark master...

Spark Health Report:

\* Master: ALIVE

\* Workers: 1

\* Cores: 1

\* Memory: 2.7 GB

launch finished in 0:03:49.

1. In the meantime, you could start yet another terminal window and prepare your code to run on AWS.  
     
   cd sql  
   cp wind.py wind-s3.py
2. Change the URL so that instead of loading the data from the local filesystem, it reaches out to S3 to do it:   
     
   atom wind-s3.py  
     
   Instead of reading from ‘/home/big/sql/\*.csv’ change it to read from:  
     
   's3a://oxclo-wind/2015/\*'
3. Delete the first two lines (import findspark and findspark.init()).
4. Save the file
5. Once the launch of your cluster has finished, we need to copy the code into the cluster:  
     
   flintrock copy-file big wind-s3.py /home/ec2-user/wind-s3.py

1. Let’s login to the master (all one line):  
     
   flintrock login big  
   You see something like:

Warning: Permanently added '34.253.201.139' (ECDSA) to the list of known hosts.

Last login: Mon Jul 10 18:55:35 2017 from host109-156-251-208.range109-156.btcentralplus.com

\_\_| \_\_|\_ )

\_| ( / Amazon Linux AMI

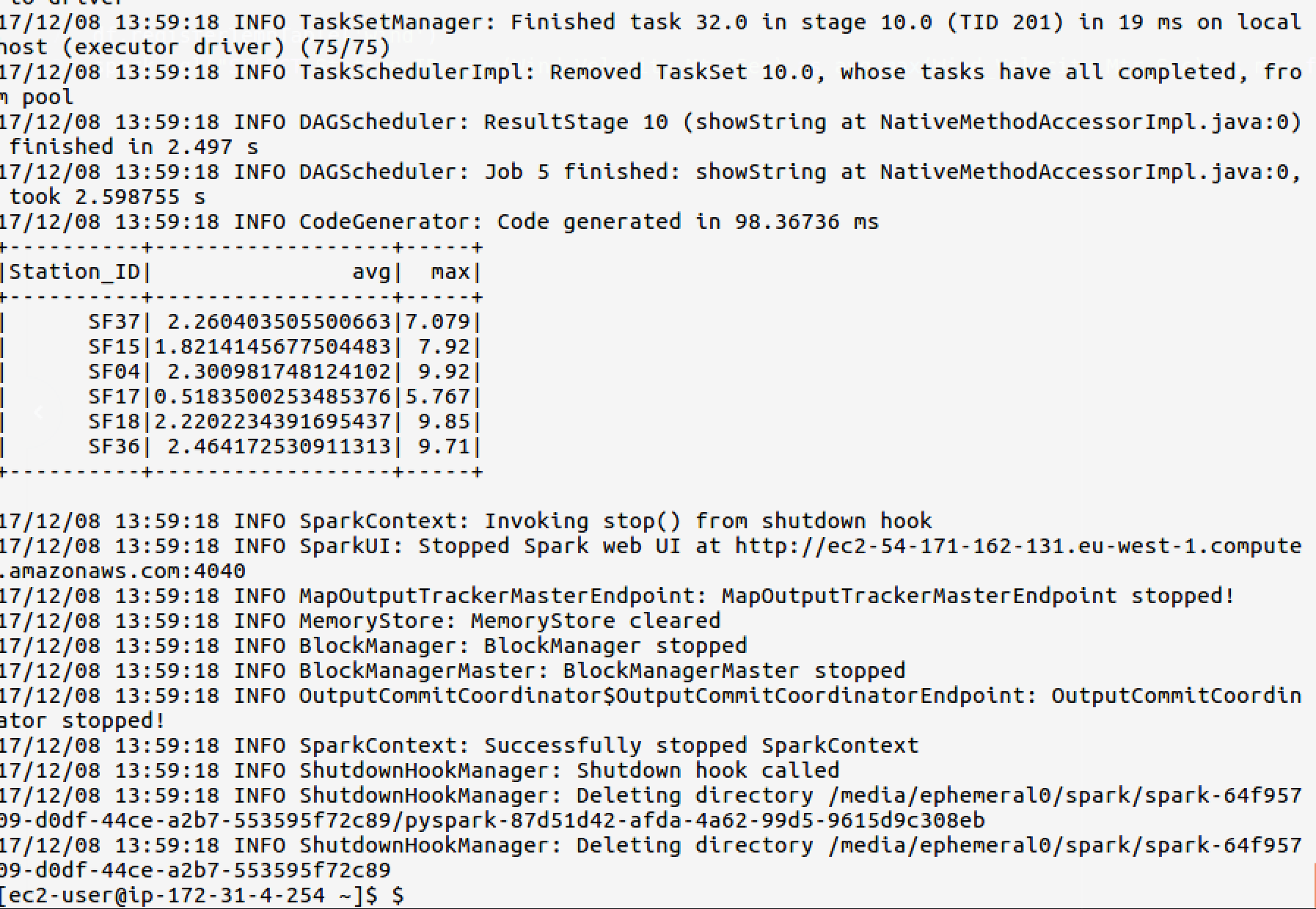
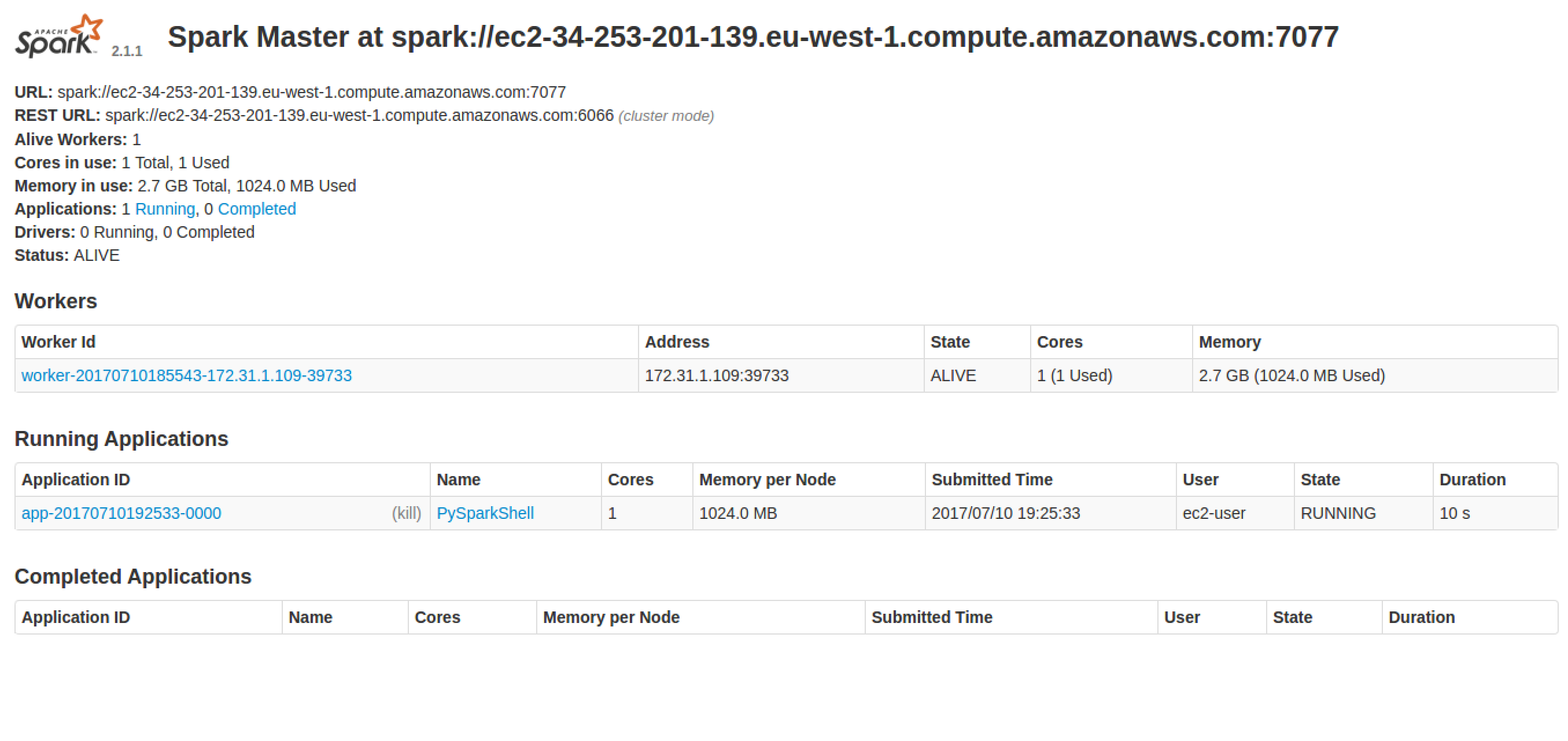
\_\_\_|\\_\_\_|\_\_\_|

https://aws.amazon.com/amazon-linux-ami/2017.03-release-notes/

1 package(s) needed for security, out of 1 available

Run "sudo yum update" to apply all updates.

[ec2-user@ip-172-31-6-32 ~]$

1. This basically just SSH’s you into the master.
2. Now launch your code:  
     
   ~/spark/bin/spark-submit wind-s3.py  
     
   You should see a lot of logging, eventually ending with:  
   
3. In this case, it actually took us longer to run on the cluster than on our local machines. However, you will note that this would speed up for bigger problems where the parallelization would add benefits.
4. It is perfectly possible to get Jupyter to talk to Spark on our cluster, but it is slightly complex, so we will just use the normal Python command-line for the moment.
5. Find the IP address of the Spark Master. There are two ways. Firstly, it showed up in the console when you first launched the flintrock cluster:  
   [34.253.201.139] Configuring Spark master...  
      
   Alternatively, you can find it by typing   
   flintrock describe big
6. Go to <http://xx.xx.xx.xx:8080> using the master’s IP address.  
   You should see something like:
7. You can explore your Spark cluster here.
8. Exit the SSH session:  
   exit
9. If you are planning to do the **Jupyter on EC2 exercise** straight away, then you can start it now and use your existing flintrock/EC2 cluster. Otherwise please follow the next instruction to shut down the EC2 instances.
10. We must remember to stop our cluster as well (its costing money…)  
    From Ubuntu terminal where you started the Spark cluster  
      
    flintrock destroy big  
    Type y when prompted.
11. Congratulations, this lab is complete.