Setting up your AWS account

Amazon will ask you for your credit card information during the setup process. You will be charged for using their services. You should not have to spend more than 5-10 dollars.

1. Go to http://aws.amazon.com/ and sign up:
   a. You may sign in using your existing Amazon account or you can create a new account by selecting "Create a free account" using the button at the right, then selecting "I am a new user."
   b. Enter your contact information and confirm your acceptance of the AWS Customer Agreement.
   c. Once you have created an Amazon Web Services Account, you may need to accept a telephone call to verify your identity. Some students have used Google Voice successfully if you don't have or don't want to give a mobile number.
2. Once you have an account, go to http://aws.amazon.com/ and sign in. You will work primarily from the Amazon Management Console.
3. Create Security Credentials. Go to the AWS security credentials page . If you are asked about IAM users, close the message. Expand "Access Keys" and click "Create New Access Key." You will see a message **Your access key (access key ID and secret access key) has been created successfully.** Click "Download Key File" and make note of where you saved the file.

Setting up an EC2 key pair

To connect to the Amazon EC2 instances you will be creating, you need to create an SSH key pair.

1. After setting up your account, follow Amazon's instructions to create a key pair. Follow the instructions in section "Creating Your Key Pair Using Amazon EC2." (We have reports that Internet Explorer could make it impossible to download the .pem private key file; you may want to use a different browser.)
2. Download and save the .pem private key file to disk. We will reference the .pem file as `</path/to/saved/keypair/file.pem>` in the following instructions.
3. Make sure only you can access the .pem file. If you do not change the permissions, you will get an error message at a later step. Change the permissions using this command:

   ```
   $ chmod 600 </path/to/saved/keypair/file.pem>
   ```

4. Note: This step will NOT work on Windows 7 with cygwin. Windows 7 does not allow file permissions to be changed through this mechanism, and they must be changed for ssh to work. So if you must use Windows, you should use PuTTY as your ssh client. In this case, you will further have to transform this key file into PuTTY format. For more information go

to http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/putty.html and look under "Private Key Format."

Starting an AWS Cluster and running Pig Interactively

To run a Pig job on AWS, you need to start up an AWS cluster using the AWS Management Console and connect to the Hadoop *master node.* Follow the steps below. You may also find Amazon's interactive Pig tutorial useful, but note that the screenshots are slightly out of date.

To set up and connect to a pig cluster, perform the following steps:

1. Go to http://console.aws.amazon.com/elasticmapreduce/home signing in if necessary.
2. Click the "Create Cluster".
3. Enter "Data Science Assignment Cluster" or anything you wish as the Cluster Name.
4. Uncheck the Logging box so that it is disabled.
5. Scroll down to Software Configuration and select **AMI Version 2.4.2** (We will be using Hadoop 1.x for compatibility with the version of Pig)
6. Scroll down to "Security and Access" and select the Key Pair you created above.
7. Scroll to the bottom and select "Create Cluster"
8. On the next page, information about your cluster will be displayed. It will begin in the "Starting" state and may take several minutes to startup completely.
9. Once the cluster has started, you will see a "Master Public DNS" name of the form ec2-XX-XXX-X-XXX.compute-1.amazonaws.com. Make a note of this; we will refer to it as **<master.public-dns-name.amazonaws.com>**.
10. Now you are ready to connect to your cluster and run Pig jobs. From a terminal, use the following command:

    ```
    $ ssh -o "ServerAliveInterval 10" -i </path/to/saved/keypair/file.pem>
    hadoop@<master.public-dns-name.amazonaws.com>
    ```
11. Once you connect successfully, just type

    ```
    $ pig
    ```
12. Now you should have a pig prompt

    ```
    grunt>
    ```
13. This is the interactive mode where you type in pig queries. In this quiz we will use pig only interactively. (The alternative is to have pig read the program from a file.) Here you will cut and paste `example.pig`. You are now ready to return to the quiz.

    Other useful information:
    - For the first job you run, Hadoop will create the output directory for you automatically. But Hadoop refuses to overwrite existing results. So you will need to move your prior results to a different directory before re-running your script, specify a different output directory in the script, or delete the prior results altogether.

To see how to perform these tasks and more, see "Managing the results of your Pig queries" below.

- To exit pig, type `quit` at the `grunt>` promt. To terminate the ssh session, type `exit` at the unix prompt: after that you must terminate the AWS cluster (see next).
- To kill a pig job type CTRL/C while pig is running.This kills pig only: after that you need to kill the hadoop job. We show you how to do this below.

Monitoring Hadoop jobs

By far the easiest way to do this from linux or a mac is to use ssh tunneling.

1. On your local machine, run this command

```
ssh -L 9100:localhost:9100 -L 9101:localhost:9101  -i ~/.ssh/</path/to/saved/keypair/file.pem> hadoop@<master.public-dns-name.amazonaws.com>
```

2. Open your browser to http://localhost:9101

From there, you can monitor your jobs' progress using the UI.

Another way to do this is to use lynx, a text-based web browser.

1. Using LYNX. Very easy, you don't need to download anything. Open a separate `ssh` connection to the AWS master node and type:

   ```
   % lynx http://localhost:9101/
   ```

   Lynx is a text browswer. Navigate as follows: `up/down arrows` = move through the links (current link is highlighted); `enter` = follows a link; `left arrow` = return to previous page.

   Examine the webpage carefully while your pig program is running. You should find information about the map tasks, the reduce tasks, you should be able to drill down into each map task (for example to monitor its progress); you should be able to look at the log files of the map tasks (if there are runtime errors, you will see them only in these log files).

Killing a Hadoop Job

Later, in the assignment, we will show you how to launch MapReduce jobs through Pig. You will basically write Pig Latin scripts that will be translated into MapReduce jobs (see lecture notes).

Some of these jobs can take a long time to run. If you decide that you need to interrupt a job before it completes, here is the way to do it:

If you want to kill pig, you first type CTRL/C, which kills pig only. Next, kill the hadoop job, as follows. From the job tracker interface find the hadoop `job_id`, then type:

```
% hadoop job -kill job_id
```

You do not need to kill any jobs at this point.

However, you can now exit pig (just type "quit") and exit your ssh session. You can also kill the SSH SOCKS tunnel to the master node.

Terminating an AWS cluster

When you are done running Pig scripts, make sure to **ALSO** terminate your job flow. This is a step that you need to do **in addition to** stopping pig and Hadoop (if necessary) above.

This step shuts down your AWS cluster:

1. Go to the Management Console.
2. Select the job in the list.
3. Click the Terminate button at the top.
4. You may need to turn Termination protection off before you are allowed to terminate -- you can do so from within the dialog.
5. Wait for a while (may take minutes) and recheck until the job state becomes TERMINATED.

Pay attention to this step. If you fail to terminate your job and only close the browser, or log off AWS, your AWS will continue to run, and AWS will continue to charge you: for hours, days, weeks, and when your credit is exhausted, it chages your creditcard. Make sure you don't leave the console until you have confirmation that the job is terminated.

You can now shut down your cluster.

Checking your Balance

Please check your balance regularly!!!

1. Go to the Management Console.
2. Click on your name in the top right corner and select "Account Activity".
3. Now click on "detail" to see any charges < $1.

To avoid unnecessary charges, terminate your job flows when you are not using them.

USEFUL: AWS customers can now use **billing alerts** to help monitor the charges on their AWS bill. You can get started today by visiting your Account Activity page to enable monitoring of your charges. Then, you can set up a billing alert by simply specifying a bill threshold and an e-mail address to be notified as soon as your estimated charges reach the threshold.

Managing the results of your Pig queries

For the next step, you need to restart a new cluster as follows. Hopefully, it should now go very quickly:

- Start a new cluster with one instance.
- Start a new interactive Pig session (through grunt)
- Connect to the monitoring interface using an ssh tunnel or via lynx

We will now get into more details about running Pig scripts.

Your pig program stores the results in several files in a directory. You have two options: (1) store these files in the Hadoop File System, or (2) store these files in S3. In either case you will need to copy them to your local machine.

1. Storing Files in the Hadoop File System

This is done through the following pig command (used in `example.pig`):

```
 store count_by_object_ordered into '/user/hadoop/example-results' us
ing PigStorage();
```

Before you run the pig query, you need to (A) create the /user/hadoop directory. After you run the query you need to (B) copy this directory to the local directory of the AWS master node, then (C) copy this directory from the AWS master node to your local machine.

1.A. Create the "/user/hadoop" Directory in the Hadoop Filesystem

You will need to do this for each new job flow that you create.

To create a `/user/hadoop` directory on the AWS cluster's HDFS file system run this from within grunt

```
hadoop fs -mkdir /user/hadoop
```

Check that the directory was created by listing it with this command:

```
hadoop fs -ls /user/hadoop
```

You may see some output from either command, but you should not see any errors.

You can also do this directly from grunt with the following command.

```
grunt> fs -mkdir /user/hadoop
```

Now you are ready to run your first sample program. Take a look at the starter code that we provided in the course materials repo. Copy and paste the content of `example.pig.`

Note: The program may appear to hang with a 0% completion time... go check the job tracker. Scroll down. You should see a MapReduce job running with some non-zero progress.

Note 2: Once the first MapReduce job gets to 100%... if your grunt terminal still appears to be suspended... go back to the job tracker and make sure that **the reduce phase is also 100% complete**. It can take some time for the reducers to start making any progress.

Note 3: The example generates more than 1 MapReduce job... so be patient.

1.B. Copying files from the Hadoop Filesystem

The result of a pig script is stored in the hadoop directory specified by the `store` command. That is, for `example.pig`, the output will be stored at `/user/hadoop/example-results`, as specified in the script. HDFS is separate from the master node's file system, so before you can copy this to your local machine, you must copy the directory from HDFS to the master node's Linux file system:

```
% hadoop fs -copyToLocal /user/hadoop/example-results example-results
```

This will create a directory `example-results` with `part-*` files in it, which you can copy to your local machine with `scp`. You can then concatenate all the `part-*` files to get a single results file, perhaps sorting the results if you like.

An easier option may be to use

```
% hadoop fs -getmerge  /user/hadoop/example-results example-results
```

This command takes a source directory and a destination file as input and concatenates files in src into the destination local file.

Use `hadoop dfs -help` or see the `hadoop dfs` guide to learn how to manipulate HDFS. (Note that `hadoop fs` is the same as `hadoop dfs`.)

1.C. Copying files to or from the AWS master node

- To copy one file from the master node back to your computer, run this command *on the local computer:*

```
$ scp -o "ServerAliveInterval 10" -i </path/to/saved/keypair/file.pem>
hadoop@<master.public-dns-name.amazonaws.com>:<file_path> .
```

where `<file_path>` can be absolute or relative to the AWS master node's home folder. The file should be copied onto your current directory ('.') on your local computer.

- Better: copy an entire directory, recursively. Suppose your files are in the directory `example-results`. They type the following *on your loal computer*:

```
$ scp -o "ServerAliveInterval 10" -i </path/to/saved/keypair/file.pem>
-r hadoop@<master.public-dns-name.amazonaws.com>:example-results .
```

- As an alternative, you may run the scp command on the AWS master node, and connect to your local machine. For that, you need to know your local machine's domain name, or IP address, and your local machine needs to accept ssh connections.

2. Storing Files in S3

To use this approach, go to your AWS Management Console, click on Create Bucket, and create a new bucket (=directory). Give it a name that may be a public name. Do not use any special chatacters, including underscore. Let's say you call it `superman`. Click on Actions, Properties, Permissions. Make sure you have all the permissions.

Modify the store command of `example.pig` to:

```
store count_by_object_ordered into 's3n://superman/example-results';
```

Run your pig program. When it terminates, then in your S3 console you should see the new directory `example-results`. Click on individual files to download. The number of files depends on the number of reduce tasks, and may vary from one to a few dozens. The only disadvantage of using S3 is that you have to click on each file separately to download.

Note that S3 is permanent storage, and you are charged for it. You can safely store all your query answers for several weeks without exceeding your credit; at some point in the future remember to delete them.

Created Thu 23 May 2013 11:02 PM PDT

Last Modified Tue 22 Jul 2014 1:04 AM PDT