

Lab 5

In this lab, we will use MapReduce to analyze a dataset of 3-4 million tweets collected over the past weekend. This dataset specifically has English-language tweets from within the United States. You will need to have your AWS account set up (completed in lab 1) to complete this lab.

First, download the files **WordCount.java**, **top.py**, and **twitterdata.txt**. WordCount.java and top.py can be found in the Lab5 resources folder on NYU classes. The twitterdata.txt data can be found at <http://s3.amazonaws.com/bigdataclassescc/twitterdata.txt>

Upload the three files to your Amazon S3 bucket.

Task1: Setup EMR Cluster and run example JAR

Launch an EMR cluster using the public/private key that you generated in Lab1. Wait for the cluster to start up.

In order to connect to your master node via SSH, you will need to first modify your security group. To do this, go to the cluster you just created, and click on the blue link following “Security groups for Master”.

The screenshot shows the AWS Management Console interface for an EMR cluster. At the top, there are buttons for 'Add step', 'Resize', 'Clone', 'Terminate', and 'AWS CLI export'. Below these, the cluster name 'My cluster' is shown with a 'Waiting' status and a note 'Cluster ready after last step completed.'.

The console is divided into several sections:

- Connections:** Includes links for 'Enable Web Connection' and 'Master public DNS'.
- Tags:** Includes a link to 'View All / Edit'.
- Summary:** Displays cluster ID, creation date, elapsed time, auto-terminate setting, and termination protection.
- Configuration Details:** Shows release label, Hadoop distribution, applications, log URI, and EMRFS consistent view.
- Network and Hardware:** Displays availability zone, subnet ID, and instance counts for Master, Core, and Task nodes.
- Security and Access:** Shows key name, EC2 instance profile, EMR role, and security groups. A red circle highlights the 'Security groups' link, which points to 'sg-5d2f9d3a (ElasticMapReduce for Master: master)'.

At the bottom, there is a 'Monitoring' section with a right-pointing arrow.

In the bottom pane, select the Inbound tab and click the Edit button.

The screenshot shows the AWS IAM console interface for a security group. At the top, there's a search bar with 'sg-5d2f9d3a' and a table with columns: Name, Group ID, Group Name, VPC ID, and Description. Below this, the 'Security Group: sg-5d2f9d3a' section is visible. It has tabs for 'Description', 'Inbound', 'Outbound', and 'Tags'. The 'Inbound' tab is selected. Below the tabs, there's an 'Edit' button circled in red. Below the 'Edit' button is a table with columns: Type, Protocol, Port Range, and Source.

Type	Protocol	Port Range	Source
All TCP	TCP	0 - 65535	sg-5c2f9d3b (ElasticMapReduce-slave)
All TCP	TCP	0 - 65535	sg-5d2f9d3a (ElasticMapReduce-master)
Custom TCP Rule	TCP	8443	54.240.230.184/29
Custom TCP Rule	TCP	8443	54.240.230.240/29
Custom TCP Rule	TCP	8443	205.251.233.32/28
Custom TCP Rule	TCP	8443	205.251.234.32/28

Click Add Rule, and select SSH for Type and Anywhere for the Source. Click Save.

The screenshot shows the 'Edit inbound rules' dialog box. It has a title bar 'Edit inbound rules' and a close button 'X'. Below the title bar is a table with columns: Type, Protocol, Port Range, and Source. The table contains several rules. At the bottom left, there's an 'Add Rule' button circled in red. At the bottom right, there are 'Cancel' and 'Save' buttons.

Type	Protocol	Port Range	Source
All TCP	TCP	0 - 65535	Custom IP sg-5c2f9d3b
All TCP	TCP	0 - 65535	Custom IP sg-5d2f9d3a
Custom TCP Rule	TCP	8443	Custom IP 54.240.230.184
Custom TCP Rule	TCP	8443	Custom IP 54.240.230.240
Custom TCP Rule	TCP	8443	Custom IP 205.251.233.32
Custom TCP Rule	TCP	8443	Custom IP 205.251.234.32
Custom TCP Rule	TCP	8443	Custom IP 205.251.233.16
Custom TCP Rule	TCP	8443	Custom IP 205.251.233.16
Custom TCP Rule	TCP	8443	Custom IP 205.251.233.4
Custom TCP Rule	TCP	8443	Custom IP 54.240.230.17
All UDP	UDP	0 - 65535	Custom IP sg-5c2f9d3b
All UDP	UDP	0 - 65535	Custom IP sg-5d2f9d3a
All ICMP	ICMP	0 - 65535	Custom IP sg-5c2f9d3b
All ICMP	ICMP	0 - 65535	Custom IP sg-5d2f9d3a

Type	Protocol	Port Range	Source	Destination	Actions
All TCP	TCP	0 - 65535	Custom IP	sg-5c2f9d3b	✕
All TCP	TCP	0 - 65535	Custom IP	sg-5d2f9d3a	✕
Custom TCP Rule	TCP	8443	Custom IP	54.240.230.18	✕
Custom TCP Rule	TCP	8443	Custom IP	54.240.230.24	✕
Custom TCP Rule	TCP	8443	Custom IP	205.251.233.3	✕
Custom TCP Rule	TCP	8443	Custom IP	205.251.234.3	✕
Custom TCP Rule	TCP	8443	Custom IP	205.251.233.1	✕
Custom TCP Rule	TCP	8443	Custom IP	205.251.233.1	✕
Custom TCP Rule	TCP	8443	Custom IP	205.251.233.4	✕
Custom TCP Rule	TCP	8443	Custom IP	54.240.230.17	✕
All UDP	UDP	0 - 65535	Custom IP	sg-5c2f9d3b	✕
All UDP	UDP	0 - 65535	Custom IP	sg-5d2f9d3a	✕
All ICMP	ICMP	0 - 65535	Custom IP	sg-5c2f9d3b	✕
All ICMP	ICMP	0 - 65535	Custom IP	sg-5d2f9d3a	✕
SSH	TCP	22	Anywhere	0.0.0.0/0	✕

Add Rule Cancel Save

Now you are ready to connect to the master node using SSH. Follow the instructions here:
<http://docs.aws.amazon.com/ElasticMapReduce/latest/DeveloperGuide/emr-connect-master-node-ssh.html>

Once you have a connection to the master, we will compile the java file and package a jar file.

Follow the instructions to build binaries using Amazon EMR:
<http://docs.aws.amazon.com/ElasticMapReduce/latest/DeveloperGuide/emr-build-binaries.html>

The source files are WordCount.java, which can be found under Lab5 Resources on NYU Classes, and twitterdata.txt

For the building binaries step, you should type

```
javac -cp $(hadoop classpath) WordCount.java
jar cvf WordCount.jar *.class
```

After building the binary, copy the jar file to your S3 bucket by typing

```
hadoop fs -put WordCount.jar s3://your_bucket/WordCount.jar
```

Now, go to EMR console and add a step as follows:

Add Step

Step type: Custom JAR

Name*: Custom JAR

JAR location*: s3://your_bucket/WordCount.jar

Arguments: WordCount, s3://your_bucket/twitterdata.txt, s3://your_bucket/output_file

Action on failure: Continue

What to do if the step fails.

Cancel Add

Once the step completes, you can view the output your S3 bucket in the folder your specified.

Task 2: Popular Hashtags

Modify the WordCount.java file such that the output lists hashtags that appear more than 100 times in the data, along with the number of times each distinct hashtag appears.

If you are unfamiliar with Java, you may find the following links helpful:

<https://docs.oracle.com/javase/tutorial/java/nutsandbolts/if.html>

<https://docs.oracle.com/javase/7/docs/api/java/lang/String.html>

You will then need to recompile and repack the binaries on the EMR master, upload the JAR file to your S3 bucket, and add a step as in Task 1.

Task 3: Deliverable: Top 50 Hashtags

Download the python script top.py from NYU Classes and upload it to your S3 bucket.

Move the output file from your WordCount program you created in Task 2 to the EMR master

```
hadoop fs -get s3://your_bucket/output_folder/* output_folder
hadoop fs -get s3://your_bucket/top.py
```

To find the top (most-used) 50 hashtags using the data output by your Task 2, type

```
python top.py 50 output_folder/ top50.txt
```

This outputs the top 50 hashtags into a file called top50.txt. To view the contents of this file, type

```
cat top50.txt
```

Deliverable: Submit your top50.txt file to the Lab5 assignment link in NYU Classes. If you worked with a partner, you may both upload the same file. This must be submitted by Monday, March 7, 2016 at 12:00pm.

IMPORTANT: Remember to terminate your EMR cluster!