CENTER FOR URBAN SCIENCE+PROGRESS

NEW YORK UNIVERSITY

CUSP-GX-5008.001: Big Data Management & Analysis

SPRING 2016

Lab 5b – Hadoop @ AWS

In this lab, we're going to setup an Amazon Elastic MapReduce cluster and run a Hadoop Streaming job on top of it. You will need an Amazon Web Services (AWS) account before proceed. If you have not requested your AWS account, please do so at http://aws.amazon.com/, and selecting the AWS Educate Student account. Please also make sure that you have a Terminal with SSH capability on your machine (e.g. a Terminal on Linux/Mac OS X or Putty on Windows).

TASK 1 – Setting up a Development Environment in AWS

First, we need to setup a proper AWS Access Key, Security Group, Key Pair and Role for our development account. This will be used to enable us logging into future AWS instances. For setting up Access Key, please follow this tutorial:
http://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSGettingStartedGuide/AWSCredentials.html

> Please pay close attention to Step 6: *Choose **Download Credentials**, and store the keys in a secure location. Your secret key will no longer be available through the AWS Management Console; you will have the only copy. Keep it confidential in order to protect your account, and never email it. Do not share it outside your organization, even if an inquiry appears to come from AWS or Amazon.com. No one who legitimately represents Amazon will ever ask you for your secret key.*

After that, please follow the following tutorial by Amazon on "*Deploying a Development Environment in Amazon EC2 Using the AWS Command Line Interface*":
http://docs.aws.amazon.com/cli/latest/userguide/tutorial-ec2-ubuntu.html
In Step 1, you can use the region "us-east-1" instead of "us-west-1".
(remember to attach policies: AdminstratorAccess *and* AmazonAPIGatewayAdministrator*)*
In Step 3, please replace the image id "ami-29ebb519" with "**ami-2051294a**", a RHEL7 instance.
(the username is "ec2-user" *instead of* "ubuntu"*)*
Please be sure to stop your instance after the last step. Otherwise, you credit will be charged for the instance.

```
aws ec2 stop-instances --instance-ids <YOUR_INSTANCE_ID>
```

TASK 2 – Creating an S3 Storage Bucket

Please follow the following tutorial to create an S3 storage bucket:
http://docs.aws.amazon.com/AmazonS3/latest/UG/CreatingaBucket.html

After creating a bucket of the name of your choice. Please create a subfolder in that bucket and upload all the Lab 5 materials (please see Lab 5) to your bucket. You can do that through the AWS Web Console at:
https://console.aws.amazon.com/s3/

TASK 3 – Creating an Elastic MapReduce (EMR) Cluster

We can use the following tutorial as a guideline for creating an EMR Cluster on AWS:
http://docs.aws.amazon.com/ElasticMapReduce/latest/ManagementGuide/emr-gs.html

Please note that you only need to follow Step 1 and 2 on that page, and when creating your cluster, let's use only **m1.medium** instance to avoid unnecessary charge. Make sure that you can SSH into the master node of the cluster following these instructions (**and open SSH Inbound port for your master's security group**):
http://docs.aws.amazon.com/ElasticMapReduce/latest/ManagementGuide/emr-connect-master-node-ssh.html

TASK 4 – Running the WordCount example with Hadoop Streaming

Please follow these steps to run the WordCount example using Hadoop Streaming.

1. SSH into your cluster master node

2. Create a lab5 folder, and synchronize that with your bucket S3 lab5 folder

```
mkdir lab5
aws s3 sync s3://YOUR_BUCKET/lab5 lab5
```

3. Upload data onto HDFS and run the Word Count Example:

```
cd lab5
hadoop fs -put book.txt .
./run_emr.sh mapper.py reducer.py book.txt output 2 2 counts.txt
```

4. Verify your counts.txt to contain the list of words and counts.

5. We could also run the example above but with fetching data directly from our s3 buckets

```
./run_emr.sh s3://BUCKET/lab5/mapper.py \
s3://YOUR_BUCKET/lab5/reducer.py s3://YOUR_BUCKET/lab5/book.txt \
s3://YOUR_BUCKET/lab5/output 2 2 counts.txt
```

6. counts.txt is still being copied to local disk since we would like to inspect it from the console.

7. In Step 5, since all of our files and input/output are stored on S3, there is no need for us to login to the master node and submit our job. Indeed, we can add a step to do this through the Console. Add the following streaming step to your cluster with the following information:

a.	Mapper:	s3://YOUR_BUCKET/lab5/mapper.py
b.	Reducer:	s3://YOUR_BUCKET/lab5/reducer.py
c.	Input:	s3://YOUR_BUCKET/lab5/book.txt
d.	Output:	s3://YOUR_BUCKET/lab5/output_new

8. Wait for finish, then download and merge all output into one file called output.txt

IMPORTANT: always TERMINATE your cluster after your use to avoid additional charges.