Spark Streaming Demo

Overview

This demo illustrates how to execute a pyspark (Python) spark streaming job. The job accepts a sequence of lines that the user types in onto one terminal window over a 10 second interval and then counts the number of distinct words in those lines and outputs the word count results to a second terminal window. This continues every 10 seconds. To do this we will set up a Spark EMR cluster and connect two terminal windows to it. In the first we will run the Linux ‘nc’ (Netcat) command. It will open a TCP socket on port 3333. After it does so, any line you then type will be sent out on that port. In another terminal window we will execute a pyspark word count program that will set up the spark streaming pipeline using DStreams. Our initial DStream will be connected to and read the lines from port 3333 and then go on to perform the word count process.

So on one terminal (connected to the EMR master node) you might see:

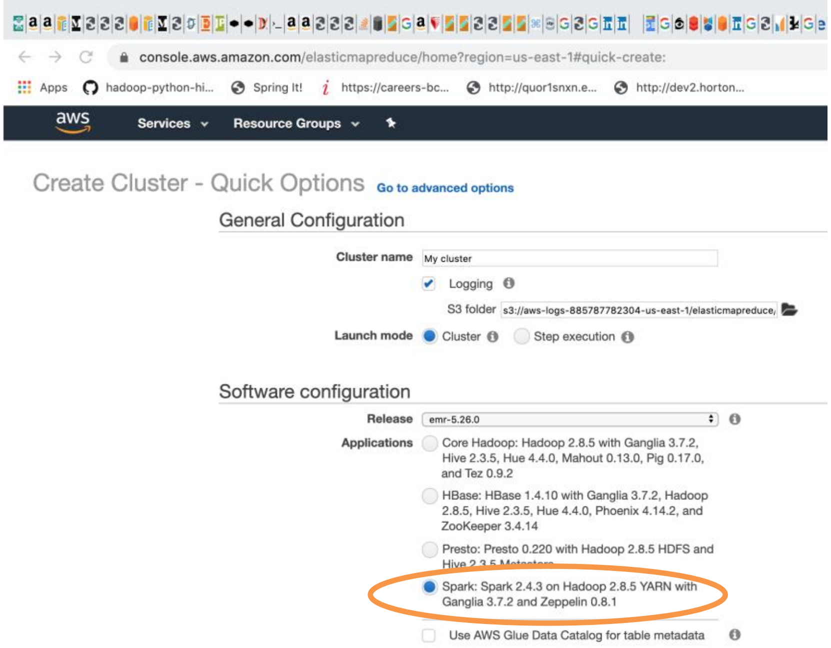
```
[hadoop@ip-172-31-19-223 ~]$ nc -lk ec2-3-91-10-18.compute-1.amazonaws.com 3333

this is a test of the the system            <- note
```

And output from the word count program running in the other terminal should look something like:

```
-------------------------------------------

Time: 2019-11-05 21:25:00

-------------------------------------------

(u'a', 1)

(u'this', 1)

(u'is', 1)

(u'test', 1)

(u'the', 2)

(u'of', 1)

(u'system', 1)
```

Running the Demo

1) Start up a Hadoop cluster as previously, but instead of choosing the “Core Hadoop” configuration chose the “Spark” configuration (see below), otherwise proceed as before.



console.aws.amazon.com/elasticmapreduce/home?region=us-east-1#quick-create:

Create Cluster - Quick Options **Go to advanced options**

General Configuration

Cluster name My cluster

☑ Logging

S3 folder s3://aws-logs-885787782304-us-east-1/elasticmapreduce/

Launch mode ● Cluster ○ Step execution

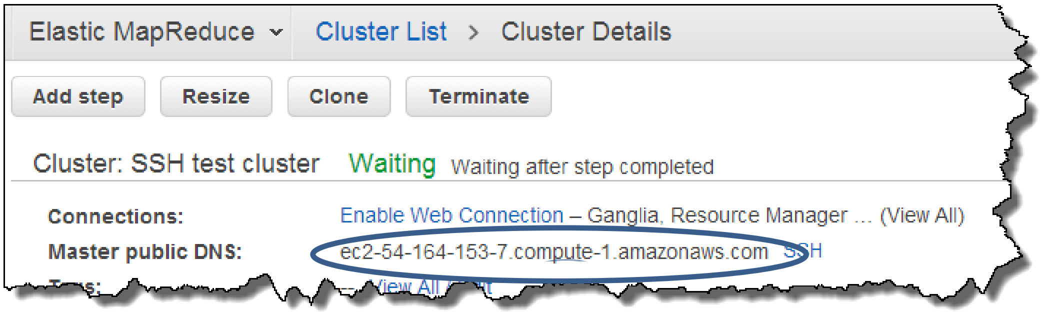
Software configuration

Release emr-5.26.0

Applications
○ Core Hadoop: Hadoop 2.8.5 with Ganglia 3.7.2, Hive 2.3.5, Hue 4.4.0, Mahout 0.13.0, Pig 0.17.0, and Tez 0.9.2
○ HBase: HBase 1.4.10 with Ganglia 3.7.2, Hadoop 2.8.5, Hive 2.3.5, Hue 4.4.0, Phoenix 4.14.2, and ZooKeeper 3.4.14
○ Presto: Presto 0.220 with Hadoop 2.8.5 HDFS and Hive 2.3.5 Metastore
● Spark: Spark 2.4.3 on Hadoop 2.8.5 YARN with Ganglia 3.7.2 and Zeppelin 0.8.1

□ Use AWS Glue Data Catalog for table metadata

2) At a later point in these instructions you will need to use the public DNS name of the master node of your EMR cluster. To retrieve it using the Amazon EMR console
   a) Find the EMR service page.
   b) On the **Cluster List** page, select the link for your cluster.
   c) Note the **Master public DNS** value that appears at the top of the **Cluster Details** page.



Cluster: SSH test cluster Waiting Waiting after step completed

Connections: Enable Web Connection – Ganglia, Resource Manager ... (View All)

Master public DNS: ec2-54-164-153-7.compute-1.amazonaws.com SSH

3) Download consume.py and log4j.properties files from the assignment to your local PC or MAC

4) There is one item you must change in consume.py. In the following line you must replace <Master public DNS> with your own public DNS name (found as described above)

```
lines = ssc.socketTextStream("<Master public DNS>", 3333)
```

For example:

```
lines = ssc.socketTextStream("ec2-54-164-153-7.compute-1.amazonaws.com”, 3333)
```

5) scp this modified consume.py file to your EMR cluster master node. You may need to answer a security question with “Y/y” or “Yes”.
6) Then scp the file log4j.properties to your EMR cluster master node.
7) Open two terminal sessions to the EMR master node. We will call one the EC2-1 window and the other the EC2-2 window.
8) In the EC2-1 window enter the following:

```
sudo cp ./log4j.properties /etc/spark/conf/log4j.properties
```

This changes the logging properties to turn off “INFO” messages to allow easier viewing of the results of the stream processing job. But it is not something you always want to disable.

9) In the EC2-1 window enter the following command to open a TCP (socket) connection on port 3333

```
nc -lk 3333
```

10) In the EC2-2 window enter the following command:

```
spark-submit consume.py
```

This takes a while to start up. So, wait for some messages issued to the console before continuing. Note, when you do this you might see a message beginning with “WARN StreamingContext:...” which you can ignore.

11) Now in the EC2-1 window enter one or more lines of text and press Enter/Return after each one including the last. You should see the word count results scroll by in the EC2-2 window
12) Remember to terminate your EMR instance after you are done!