

SPARK STANALONE APPLICATION ON AWS USING COMMAND LINE METHOD

Note : We are using 'sbt package' to compile and create jar file as explained in the previous document.

Below is the template code (Word count).

```
arunkumar@hadoop: ~/spark
arunkumar@hadoop:~/spark$ cat src/main/scala/WCArg.scala
import org.apache.spark.SparkContext
import org.apache.spark.SparkContext._
import org.apache.spark.SparkConf

object WCArg {
  def main(args: Array[String]) {
    println("Arguments:" +args(0) +"," + args(1));
    if (args.length !=2) {
      println("Arguments should be in the below format")
      println("Usage : WCArg input output")
    }
    val logFile = args(0)
    val conf = new SparkConf().setAppName("Word Count")
    val sc = new SparkContext(conf)
    val logData = sc.textFile(logFile)
    val counts = logData.flatMap(line => line.split(" "))
                        .map(word => (word, 1))
                        .reduceByKey(_ + _)
    counts.saveAsTextFile(args(1))
  }
}
```

1. Run sbt package and create jar file

```
arunkumar@hadoop: ~/spark
arunkumar@hadoop:~/spark$ ls target/scala-2.10/simple-project_2.10-1.0.jar
target/scala-2.10/simple-project_2.10-1.0.jar
arunkumar@hadoop:~/spark$
```

2. We will be using Amazon CLI for interacting with AWS. To install CLI, please go through the below link to install CLI:

<http://aws.amazon.com/cli/>

3. Copy the required files to S3 including the jar file using the below command

```
aws s3 cp ~/spark/target/scala-2.10/simple-project_2.10-1.0.jar s3://sparknpu/Jar/spark.jar
```

4. Verify the above

```
aws s3 ls s3://sparknpu/Jar/
```

```
arunkumar@hadoop: ~/spark
arunkumar@hadoop:~/spark$ aws s3 cp ~/spark/target/scala-2.10/simple-project_2.10-1.0.jar s3://sparknpu/Jar/spark.jar
upload: target/scala-2.10/simple-project_2.10-1.0.jar to s3://sparknpu/Jar/spark.jar
arunkumar@hadoop:~/spark$ aws s3 ls s3://sparknpu/Jar/
2014-10-05 15:58:08          0
2014-11-14 13:50:41      4375 simple-project_2.10-1.0.jar
2014-10-05 15:59:33   9135337 spark-example-project-0.2.0.jar
2014-11-14 16:32:00    17412 spark.jar
arunkumar@hadoop:~/spark$
```

5. Export the AWS credentials

export AWS_ACCESS_KEY_ID = your key

export AWS_SECRET_ACCESS_KEY= your key

6. Create a EC2 spark cluster by using the below command

ec2/spark-ec2 -k mykeypair -i ~/mykeypair.pem -s 2 --region us-west-1 --wait 240 launch mycluster

```
arunkumar@hadoop: ~/spark
arunkumar@hadoop:~/spark$ ec2/spark-ec2 -k mykeypair -i ~/mykeypair.pem -s 2 --region us-west-1 --wait 240 launch mycluster
Setting up security groups...
Searching for existing cluster mycluster...
Spark AMI: ami-7a320f3f
Launching instances...
Launched 2 slaves in us-west-1b, regid = r-11181d4f
Launched master in us-west-1b, regid = r-be1e1be0
Waiting for instances to start up...
█
```

Note : It takes a while to complete.

7. Copy down the spark master URL from the messages. The master URL should look like

<http://ec2-54-67-46-155.us-west-1.compute.amazonaws.com:8080/>


Open the browser and familiarize with workers/jobs/cores etc.

```
arunkumar@hadoop: ~/spark
Starting GANGLIA gmond: [ OK ]
Connection to ec2-54-67-34-43.us-west-1.compute.amazonaws.com closed.
Shutting down GANGLIA gmond: [ FAILED ]
Starting GANGLIA gmond: [ OK ]
Connection to ec2-54-67-46-146.us-west-1.compute.amazonaws.com closed.
Shutting down GANGLIA gmetad: [ FAILED ]
Starting GANGLIA gmetad: [ OK ]
Stopping httpd: [ FAILED ]
Starting httpd: [ OK ]
Connection to ec2-54-67-46-155.us-west-1.compute.amazonaws.com closed.
Spark standalone cluster started at http://ec2-54-67-46-155.us-west-1.compute.amazonaws.com:8080
Ganglia started at http://ec2-54-67-46-155.us-west-1.compute.amazonaws.com:5080/ganglia
Done!
arunkumar@hadoop:~/spark$ █
```

Download: xSpark Stan xRunning Sp xQuick Star xHow to acc xInbox (1) xGetting He xls AWS C xSpark Mas x

ec2-54-67-46-155.us-west-1.compute.amazonaws.com:8080

AppsLog OnAmazon StudentMyChart - HomeSorting AlgorithmMedtronic CarelCourse ListPatient Portal LoOther bookmarks

 Spark Master at spark://ec2-54-67-46-155.us-west-1.compute.amazonaws.com:7077

URL: spark://ec2-54-67-46-155.us-west-1.compute.amazonaws.com:7077

Workers: 2

Cores: 4 Total, 0 Used

Memory: 12.6 GB Total, 0.0 B Used

Applications: 0 Running, 0 Completed

Drivers: 0 Running, 0 Completed

Status: ALIVE

Workers

Id	Address	State	Cores	Memory
worker-20141115004952-ip-172-31-23-146.us-west-1.compute.internal-36676	ip-172-31-23-146.us-west-1.compute.internal:36676	ALIVE	2 (0 Used)	6.3 GB (0.0 B Used)
worker-20141115004952-ip-172-31-23-147.us-west-1.compute.internal-42608	ip-172-31-23-147.us-west-1.compute.internal:42608	ALIVE	2 (0 Used)	6.3 GB (0.0 B Used)

Running Applications

ID	Name	Cores	Memory per Node	Submitted Time	User	State	Duration
----	------	-------	-----------------	----------------	------	-------	----------

Completed Applications

ID	Name	Cores	Memory per Node	Submitted Time	User	State	Duration
----	------	-------	-----------------	----------------	------	-------	----------

8. Login to the master node by the below command

```
ec2/spark-ec2 -k mykeypair -i ~/mykeypair.pem --login mycluster
```

```
arunkumar@hadoop: ~/spark
arunkumar@hadoop:~/spark$ ec2/spark-ec2 -k mykeypair -i ~/mykeypair.pem --
region us-west-1 login mycluster
Searching for existing cluster mycluster...
Found 1 master(s), 2 slaves
Logging into master ec2-54-67-46-155.us-west-1.compute.amazonaws.com...
Last login: Sat Nov 15 00:46:00 2014 from 67.170.253.177

  _|_  _|_  )
  _|_ (  _|_ /   Amazon Linux AMI
  _|_ \_|_  _|_

https://aws.amazon.com/amazon-linux-ami/2013.03-release-notes/
There are 67 security update(s) out of 275 total update(s) available
Run "sudo yum update" to apply all updates.
Amazon Linux version 2014.09 is available.
root@ip-172-31-30-37 ~]$
```

9. Now we are ready to submit the Standalone Application to the cluster. Below is the command.

```
bin/spark-submit --class WCAArg
--master spark://ec2-54-67-46-155.us-west-1.compute.amazonaws.com:7077
--deploy-mode cluster
s3n://AccessKey:Secretkey@sparknpu/Jar/spark.jar
s3n://AccessKey:Secretkey@sparknpu/data/spark.txt
s3n://AccessKey:Secretkey@sparknpu/output/sparkwordcount
```

```
arunkumar@hadoop: ~/spark
root@ip-172-31-30-37 spark]$ bin/spark-submit --class WCAArg
--master spark://ec2-54-67-46-155.us-west-1.compute.amazonaws.com:7077
--deploy-mode cluster
s3n://AKIAI22X776278CEXOMA:MoZ1761kEUA1Bov1dLCOwG1eVtOvUibukmmu8vw9M@sparknpu/Jar/spark.jar
s3n://AKIAI22X776278CEXOMA:MoZ1761kEUA1Bov1dLCOwG1eVtOvUibukmmu8vw9M@sparknpu/data/spark.txt
s3n://AKIAI22X776278CEXOMA:MoZ1761kEUA1Bov1dLCOwG1eVtOvUibukmmu8vw9M@sparknpu/output/sparkwordcount

Spark assembly has been built with Hive, including Datanucleus jars on classpath
Sending launch command to spark://ec2-54-67-46-155.us-west-1.compute.amazonaws.com:7077
Driver successfully submitted as driver-20141115011009-0000
... waiting before polling master for driver state
... polling master for driver state
State of driver-20141115011009-0000 is RUNNING
Driver running on ip-172-31-23-146.us-west-1.compute.internal:36676 (worker-20141115004952-ip-172-31-23-146.
us-west-1.compute.internal-36676)
root@ip-172-31-30-37 spark]$
```

10. Copy the output files from S3 to local and view the result

Useful Commands :

```
aws s3 cp s3://sparknpu/output/sparkwordcount ~/SparkWorkSpace/awssparkoutput --recursive
```

```
arunkumar@hadoop: ~  
arunkumar@hadoop:~$ aws s3 ls s3://sparknpu/output/sparkwordcount/  
2014-11-14 17:10:25      0 _SUCCESS  
2014-11-14 17:10:24     15 part-00000  
2014-11-14 17:10:24     12 part-00001  
arunkumar@hadoop:~$
```

```
arunkumar@hadoop: ~  
arunkumar@hadoop:~$ aws s3 cp s3://sparknpu/output/sparkwordcount ~/SparkWorkSpace/awssparkoutput --recursive  
download: s3://sparknpu/output/sparkwordcount/_SUCCESS to SparkWorkSpace/awssparkoutput/_SUCCESS  
download: s3://sparknpu/output/sparkwordcount/part-00000 to SparkWorkSpace/awssparkoutput/part-00000  
download: s3://sparknpu/output/sparkwordcount/part-00001 to SparkWorkSpace/awssparkoutput/part-00001  
arunkumar@hadoop:~$
```

```
arunkumar@hadoop: ~  
arunkumar@hadoop:~$ ls SparkWorkSpace/awssparkoutput/  
part-00000 part-00001 _SUCCESS  
arunkumar@hadoop:~$ cat SparkWorkSpace/awssparkoutput/part-00000  
(2,1)  
(Line,3)  
arunkumar@hadoop:~$ cat SparkWorkSpace/awssparkoutput/part-00001  
(3,1)  
(1,1)  
arunkumar@hadoop:~$
```

11. Terminating Cluster (Very Important)

Note : Double check by browsing through EC2 instances in AWS management console.

ec2/spark-ec2 --region us-west-1 destroy mycluster

```
arunkumar@hadoop: ~/spark
(Line,3)
arunkumar@hadoop:~$ cat SparkWorkSpace/awssparkoutput/part-00001
(3,1)
(1,1)
arunkumar@hadoop:~$ cd spark
arunkumar@hadoop:~/spark$ ec2/spark-ec2 --region us-west-1 destroy mycluster
Are you sure you want to destroy the cluster mycluster?
The following instances will be terminated:
Searching for existing cluster mycluster...
Found 1 master(s), 2 slaves
> ec2-54-67-46-155.us-west-1.compute.amazonaws.com
> ec2-54-67-34-43.us-west-1.compute.amazonaws.com
> ec2-54-67-46-146.us-west-1.compute.amazonaws.com
ALL DATA ON ALL NODES WILL BE LOST!!
Destroy cluster mycluster (y/N): y
Terminating master...
Terminating slaves...
arunkumar@hadoop:~/spark$
```

