

SPARK ON AWS

1. Install Amazon CLI using the below link:

<http://docs.aws.amazon.com/ElasticMapReduce/latest/DeveloperGuide/emr-cli-install.html>

2. Please execute the below commands to verify the installation

`ruby -v`

`gem -v`

`./elastic-mapreduce - version`

```
arunkumar@hadoop: ~/elastic-mapreduce-cli
arunkumar@hadoop:~/elastic-mapreduce-cli$ ruby -v
ruby 1.9.3p484 (2013-11-22 revision 43786) [x86_64-linux]
arunkumar@hadoop:~/elastic-mapreduce-cli$ gem -v
1.8.23
arunkumar@hadoop:~/elastic-mapreduce-cli$
```

```
arunkumar@hadoop: ~/elastic-mapreduce-cli

Uncommon Options
  --debug                Print stack traces when exceptions occur
  --endpoint ENDPOINT    EMR web service host to connect to
  --region REGION        The region to use for the endpoint
  --apps-path APPS_PATH  Specify s3:// path to the base of the emr public bucket to use. e.g s3://us-east-1.elasticmapreduce

Short Options
  -h                    Show help message
  -v                    Turn on verbose logging of program interaction
  -c CRED_FILE          File containing access-id and private-key
  -a ACCESS_ID          AWS Access Id
  -p PRIVATE_KEY        AWS Private Key
  -j JOB_FLOW_ID        The job flow to act on
arunkumar@hadoop:~/elastic-mapreduce-cli$
```

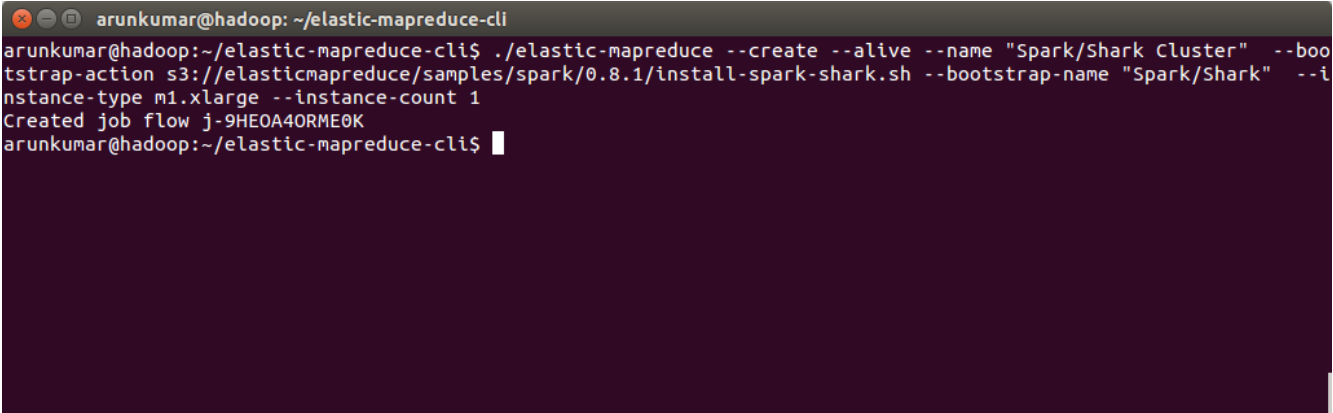
3. Run the below command to create a new spark cluster :

Note: The instance count determines the number of machines will be running in your cluster. Which can low as 1. Please use '1' initially as our sample programs will easily run in that. Also use 'expand all' option in bills to see your usage charges. If you are not using 'expand all' option, always it displays the bill as '0'.

```
elastic-mapreduce --create --alive --name "Spark/Shark Cluster" --bootstrap-action  
s3://elasticmapreduce/samples/spark/0.8.1/install-spark-shark.sh --bootstrap-name  
"Spark/Shark" --instance-type m1.xlarge --instance-count 1
```

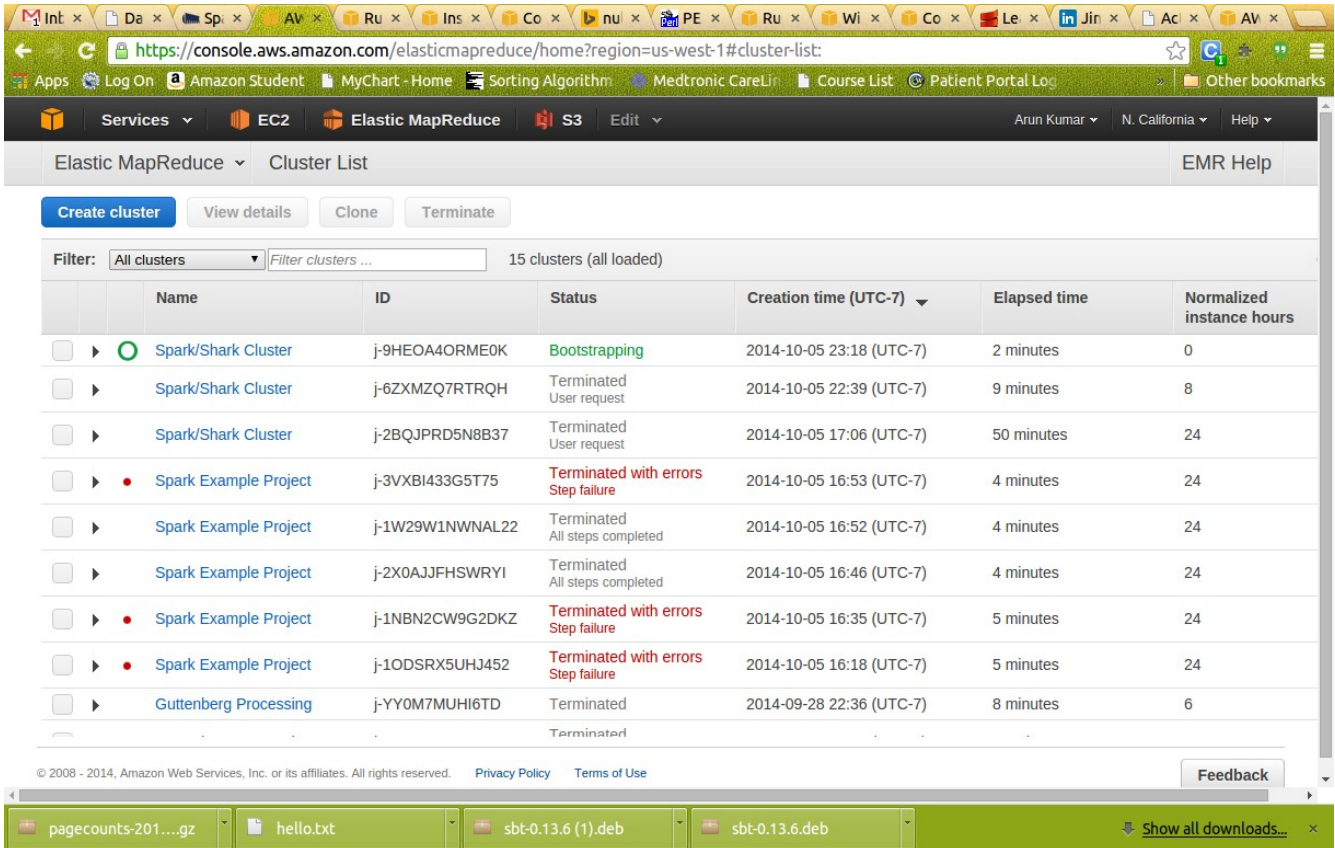
This command will initiate a new cluster and you will see a message like :

Created job flow j-9HE0A40RME0K

A terminal window with a dark background and light text. The title bar shows 'arunkumar@hadoop: ~/elastic-mapreduce-cli'. The command prompt is 'arunkumar@hadoop:~/elastic-mapreduce-cli\$'. The command entered is './elastic-mapreduce --create --alive --name "Spark/Shark Cluster" --bootstrap-action s3://elasticmapreduce/samples/spark/0.8.1/install-spark-shark.sh --bootstrap-name "Spark/Shark" --instance-type m1.xlarge --instance-count 1'. The output is 'Created job flow j-9HE0A40RME0K'. The prompt is now 'arunkumar@hadoop:~/elastic-mapreduce-cli\$' with a cursor.

```
arunkumar@hadoop: ~/elastic-mapreduce-cli  
arunkumar@hadoop:~/elastic-mapreduce-cli$ ./elastic-mapreduce --create --alive --name "Spark/Shark Cluster" --boo  
tstrap-action s3://elasticmapreduce/samples/spark/0.8.1/install-spark-shark.sh --bootstrap-name "Spark/Shark" --i  
nstance-type m1.xlarge --instance-count 1  
Created job flow j-9HE0A40RME0K  
arunkumar@hadoop:~/elastic-mapreduce-cli$
```

4. You can see the same cluster in AWS



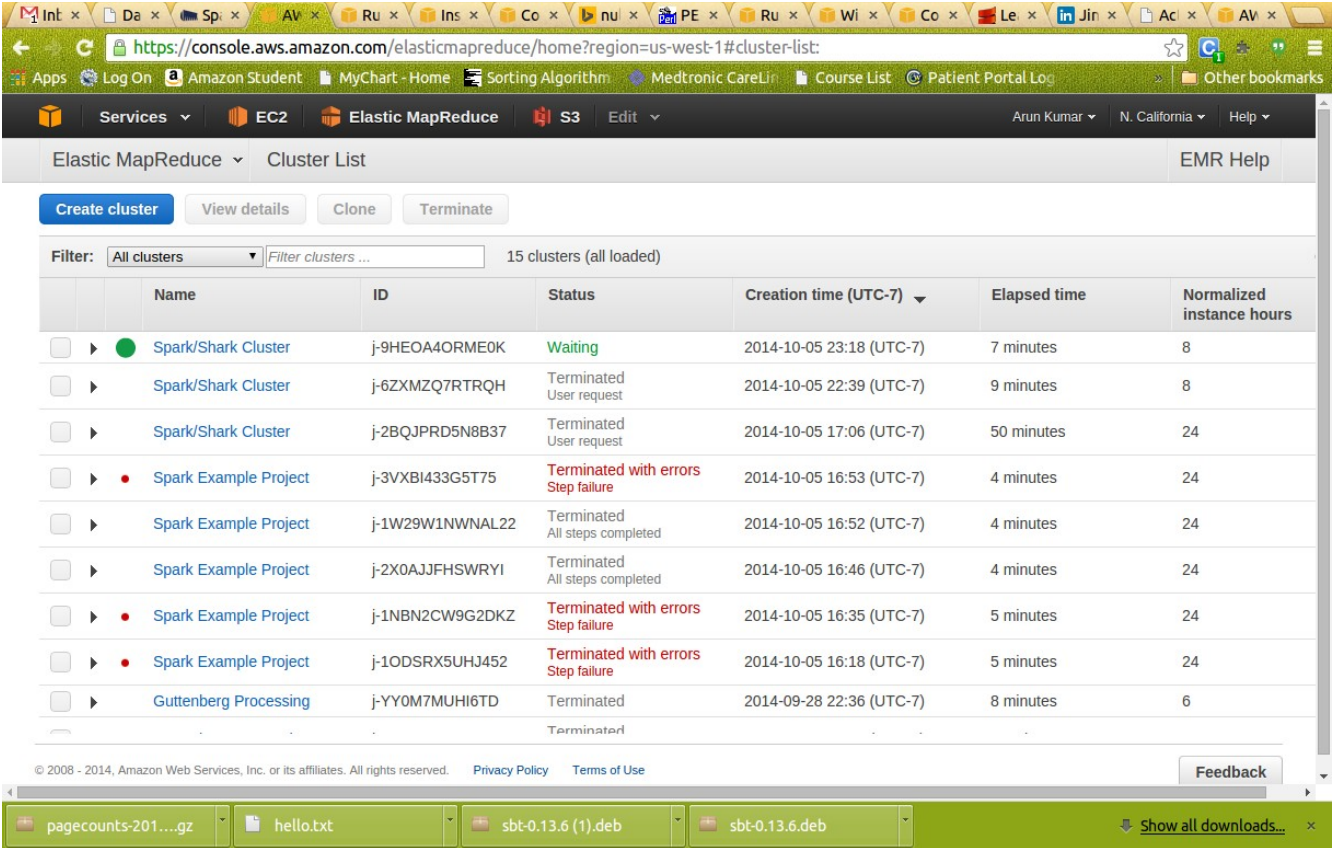
The screenshot shows the AWS Elastic MapReduce console interface. The top navigation bar includes 'Services', 'EC2', 'Elastic MapReduce', 'S3', and 'Edit'. The user is logged in as 'Arun Kumar' in the 'N. California' region. The 'Cluster List' page is displayed, showing a table of 15 clusters. The table columns are: Name, ID, Status, Creation time (UTC-7), Elapsed time, and Normalized instance hours. The clusters are listed in descending order of creation time. The first cluster, 'Spark/Shark Cluster' (ID: j-9HEOA4ORME0K), is in the 'Bootstrapping' status. The second cluster, 'Spark/Shark Cluster' (ID: j-6ZXMZQ7RTRQH), is 'Terminated' due to a 'User request'. The third cluster, 'Spark/Shark Cluster' (ID: j-2BQJPRD5N8B37), is also 'Terminated' due to a 'User request'. The fourth cluster, 'Spark Example Project' (ID: j-3VXBI433G5T75), is 'Terminated with errors' due to a 'Step failure'. The fifth cluster, 'Spark Example Project' (ID: j-1W29W1NWNAL22), is 'Terminated' with 'All steps completed'. The sixth cluster, 'Spark Example Project' (ID: j-2X0AJJFHSWRYI), is 'Terminated' with 'All steps completed'. The seventh cluster, 'Spark Example Project' (ID: j-1NBN2CW9G2DKZ), is 'Terminated with errors' due to a 'Step failure'. The eighth cluster, 'Spark Example Project' (ID: j-1ODSRX5UHJ452), is 'Terminated with errors' due to a 'Step failure'. The ninth cluster, 'Guttenberg Processing' (ID: j-YY0M7MUHI6TD), is 'Terminated'. The bottom of the console shows a file download bar with files like 'pagecounts-201...gz', 'hello.txt', and 'sbt-0.13.6 (1).deb'.

Name	ID	Status	Creation time (UTC-7)	Elapsed time	Normalized instance hours
Spark/Shark Cluster	j-9HEOA4ORME0K	Bootstrapping	2014-10-05 23:18 (UTC-7)	2 minutes	0
Spark/Shark Cluster	j-6ZXMZQ7RTRQH	Terminated User request	2014-10-05 22:39 (UTC-7)	9 minutes	8
Spark/Shark Cluster	j-2BQJPRD5N8B37	Terminated User request	2014-10-05 17:06 (UTC-7)	50 minutes	24
Spark Example Project	j-3VXBI433G5T75	Terminated with errors Step failure	2014-10-05 16:53 (UTC-7)	4 minutes	24
Spark Example Project	j-1W29W1NWNAL22	Terminated All steps completed	2014-10-05 16:52 (UTC-7)	4 minutes	24
Spark Example Project	j-2X0AJJFHSWRYI	Terminated All steps completed	2014-10-05 16:46 (UTC-7)	4 minutes	24
Spark Example Project	j-1NBN2CW9G2DKZ	Terminated with errors Step failure	2014-10-05 16:35 (UTC-7)	5 minutes	24
Spark Example Project	j-1ODSRX5UHJ452	Terminated with errors Step failure	2014-10-05 16:18 (UTC-7)	5 minutes	24
Guttenberg Processing	j-YY0M7MUHI6TD	Terminated	2014-09-28 22:36 (UTC-7)	8 minutes	6

5. When your cluster is in the WAITING status, we will be using the sample data in one of the public S3 bucket located at the below link. We can directly use this data in our spark shell. So you don't have to download it.

<https://s3.amazonaws.com/bigdatademo/sample/wiki/pagecounts-20100212-050000.gz>.

6. Check the status of cluster. It should be in waiting state.



The screenshot displays the AWS Elastic MapReduce console interface. The top navigation bar includes the AWS logo, the text 'Services', and the 'Elastic MapReduce' service selected. The main header shows 'Cluster List' and 'EMR Help'. Below the header, there are buttons for 'Create cluster', 'View details', 'Clone', and 'Terminate'. A filter dropdown is set to 'All clusters', showing '15 clusters (all loaded)'. The main content area is a table with the following columns: Name, ID, Status, Creation time (UTC-7), Elapsed time, and Normalized instance hours. The table lists several clusters, with the first one, 'Spark/Shark Cluster' (ID: j-9HEOA4ORME0K), currently in a 'Waiting' state. Other clusters are in 'Terminated' states, with some having specific error messages like 'Terminated with errors Step failure'.

Name	ID	Status	Creation time (UTC-7)	Elapsed time	Normalized instance hours
Spark/Shark Cluster	j-9HEOA4ORME0K	Waiting	2014-10-05 23:18 (UTC-7)	7 minutes	8
Spark/Shark Cluster	j-6ZXMZQ7RTRQH	Terminated User request	2014-10-05 22:39 (UTC-7)	9 minutes	8
Spark/Shark Cluster	j-2BQJPRD5N8B37	Terminated User request	2014-10-05 17:06 (UTC-7)	50 minutes	24
Spark Example Project	j-3VXBI433G5T75	Terminated with errors Step failure	2014-10-05 16:53 (UTC-7)	4 minutes	24
Spark Example Project	j-1W29W1NWNAL22	Terminated All steps completed	2014-10-05 16:52 (UTC-7)	4 minutes	24
Spark Example Project	j-2X0AJJFHSWRYI	Terminated All steps completed	2014-10-05 16:46 (UTC-7)	4 minutes	24
Spark Example Project	j-1NBN2CW9G2DKZ	Terminated with errors Step failure	2014-10-05 16:35 (UTC-7)	5 minutes	24
Spark Example Project	j-1ODSRX5UHHJ452	Terminated with errors Step failure	2014-10-05 16:18 (UTC-7)	5 minutes	24
Gutenberg Processing	j-YY0M7MUHI6TD	Terminated	2014-09-28 22:36 (UTC-7)	8 minutes	6

© 2008 - 2014, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Feedback

pagecounts-201....gz hello.txt sbt-0.13.6 (1).deb sbt-0.13.6.deb Show all downloads...

7.. Now we should ssh to master node to initialize spark shell. We will be running our scrips in master node. To get the ssh command, click on cluster, then 'cluster details' in Elastic Map reduce console page. Then click on SSH as shown below

The screenshot displays the AWS Elastic MapReduce console interface. The browser address bar shows the URL: `https://console.aws.amazon.com/elasticmapreduce/home?region=us-west-1#cluster-details:j-9HEOA4ORME0K`. The navigation bar includes 'Services', 'EC2', 'Elastic MapReduce', 'S3', and 'Edit'. The breadcrumb trail is 'Elastic MapReduce > Cluster List > Cluster Details'. The cluster name is 'Spark/Shark Cluster' and its state is 'Waiting' with the note 'Waiting for steps to run'. Action buttons include 'Add step', 'Resize', 'Clone', and 'Terminate'. The 'Connections' section shows the 'Master public DNS' as `ec2-54-183-226-77.us-west-1.compute.amazonaws.com` with an 'SSH' link. The 'Summary' tab is active, showing details like ID, creation date, and hardware configuration. The 'Configuration Details' tab shows AMI version, Hadoop distribution, and applications. The 'Security/Network' tab shows availability zone, subnet, and key name. The 'Hardware' section shows the master node is running on `m1.xlarge`. The 'Steps' section at the bottom shows a list of steps including 'pagecounts-201....gz', 'hello.txt', and 'sbt-0.13.6 (1).deb'.

Cluster: Spark/Shark Cluster **Waiting** Waiting for steps to run

Connections: [Enable Web Connection – Job Tracker ... \(View All\)](#)

Master public DNS: `ec2-54-183-226-77.us-west-1.compute.amazonaws.com` [SSH](#)

Tags: -- [View All / Edit](#)

Summary	Configuration Details	Security/Network
ID: j-9HEOA4ORME0K	AMI version: 2.4.2	Availability zone: us-west-1a
Creation date: 2014-10-05 23:18 (UTC-7)	Hadoop distribution: Amazon 1.0.3	Subnet ID: --
Elapsed time: 11 minutes	Applications: --	Key name: mykeypair
Auto-terminate: No	Log URI: <code>s3://sparknpu/log/</code>	EC2 instance profile: --
Termination protection: Off Change	EMRFS: Disabled	EMR role: --
	consistent view:	Visible to all users: None Change

Hardware

Master: **Running** 1 m1.xlarge

Core: --

Task: --

► **Monitoring**

► **Steps**

pagecounts-201....gz | hello.txt | sbt-0.13.6 (1).deb | sbt-0.13.6.deb | [Show all downloads...](#)

Int x Da x Sp x AV x Ru x Ins x Co x nu x PE x Ru x Wi x Co x Le x lin Jin x Ac x AV x

https://console.aws.amazon.com/elasticmapreduce/home?region=us-west-1#cluster-details:j-9HEOA4ORME0K

Apps Log On Amazon Student MyChart - Home Sorting Algorithm Medtronic CareLin Course List Patient Portal Log Other bookmarks

Services EC2 Elastic MapReduce S3 Edit Arun Kumar N. California Help

Elastic MapReduce Cluster List Cluster Details EMR Help

Add step Resize Clone Terminate

Cluster SSH

Connect to the Master Node Using SSH

You can connect to the Amazon EMR master node using SSH to run interactive queries, examine log files, submit Linux commands, and so on.
[Learn more.](#)

Windows Mac / Linux

1. Open a terminal window. On Mac OS X, choose Applications > Utilities > Terminal. On other Linux distributions, terminal is typically found at Applications > Accessories > Terminal.
2. To establish a connection to the master node, type the following command. Replace ~/mykeypair.pem with the location and filename of the private key file (.pem) used to launch the cluster.
`ssh hadoop@ec2-54-183-226-77.us-west-1.compute.amazonaws.com -i ~/mykeypair.pem`
3. Type yes to dismiss the security warning.

Close

Master: Running 1 m1.xlarge
Core: --
Task: --

Monitoring

Steps

pagecounts-201....gz hello.txt sbt-0.13.6 (1).deb sbt-0.13.6.deb

Show all downloads...


```
ssh hadoop@ec2-54-183-226-77.us-west-1.compute.amazonaws.com -i ~/mykeypair.pem
```

```

arunkumar@hadoop: ~/elastic-mapreduce-cli
arunkumar@hadoop:~/elastic-mapreduce-cli$ ssh hadoop@ec2-54-183-226-77.us-west-1.compute.amazonaws.com -i ~/mykeypair.pem
The authenticity of host 'ec2-54-183-226-77.us-west-1.compute.amazonaws.com (54.183.226.77)' can't be established.
RSA key fingerprint is c5:91:15:ec:03:2e:91:9c:d8:b0:e8:92:f3:5a:69:5f.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'ec2-54-183-226-77.us-west-1.compute.amazonaws.com,54.183.226.77' (RSA) to the list of
known hosts.
Linux (none) 3.2.30-49.59.amzn1.x86_64 #1 SMP Wed Oct 3 19:54:33 UTC 2012 x86_64
-----
Welcome to Amazon Elastic MapReduce running Hadoop and Debian/Squeeze.

Hadoop is installed in /home/hadoop. Log files are in /mnt/var/log/hadoop. Check
/mnt/var/log/hadoop/steps for diagnosing step failures.

The Hadoop UI can be accessed via the following commands:

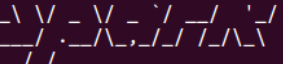
JobTracker      lynx http://localhost:9100/
NameNode        lynx http://localhost:9101/
-----
hadoop@ip-172-31-8-132:~$

```

```
SPARK_MEM="2g" /home/hadoop/spark/spark-shell
```

```
arunkumar@hadoop:~/elastic-mapreduce-cli$ spark-submit --master hadoop://ip-172-18-132:~$ SPARK_MEM="2g" /home/hadoop/spark/spark-shell
```

Welcome to



version 0.8.1

Using Scala version 2.9.3 (Java HotSpot(TM) 64-Bit Server VM, Java 1.7.0_40)
Initializing interpreter...
log4j:WARN No appenders could be found for logger (org.eclipse.jetty.util.log).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.

Creating SparkContext...
Spark context available as sc.
Type in expressions to have them evaluated.
Type :help for more information.

```
scala>
```

You will be treated with the scala prompt.

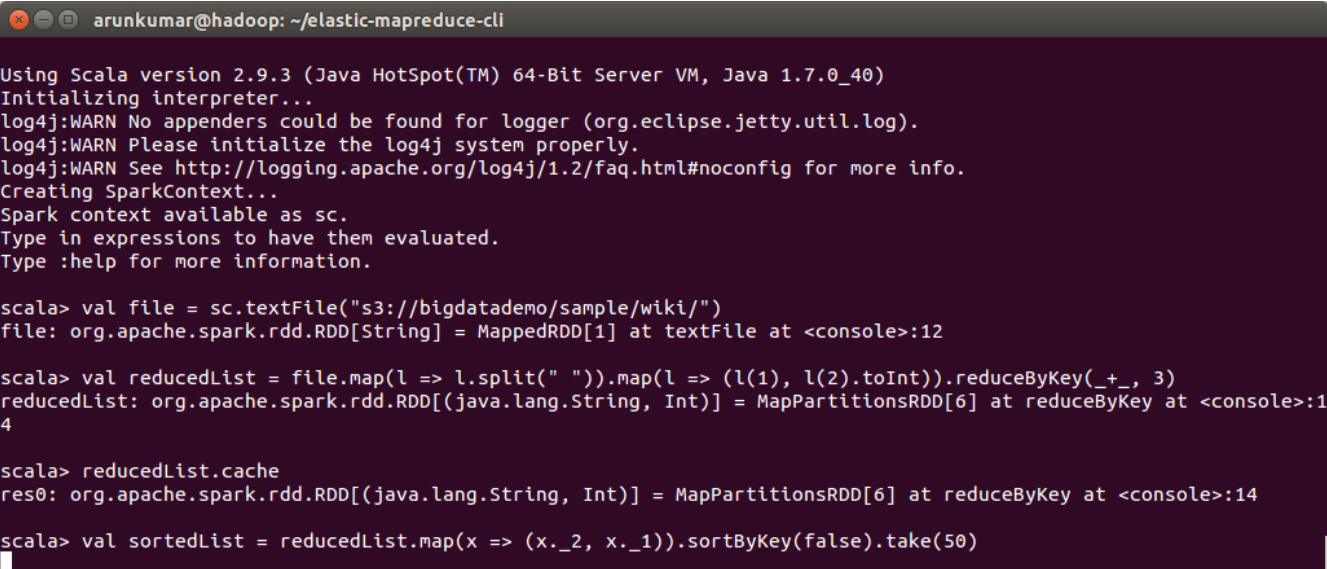
10. Run the below commands one by one.

```
val file = sc.textFile("s3://bigdatademo/sample/wiki/")
```

```
val reducedList = file.map(l => l.split(" ")).map(l => (l(1), l(2).toInt)).reduceByKey(_+_ , 3)
```

```
reducedList.cache
```

```
val sortedList = reducedList.map(x => (x._2, x._1)).sortByKey(false).take(50)
```



```
arunkumar@hadoop: ~/elastic-mapreduce-cli
Using Scala version 2.9.3 (Java HotSpot(TM) 64-Bit Server VM, Java 1.7.0_40)
Initializing interpreter...
log4j:WARN No appenders could be found for logger (org.eclipse.jetty.util.log).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
Creating SparkContext...
Spark context available as sc.
Type in expressions to have them evaluated.
Type :help for more information.

scala> val file = sc.textFile("s3://bigdatademo/sample/wiki/")
file: org.apache.spark.rdd.RDD[String] = MappedRDD[1] at textFile at <console>:12

scala> val reducedList = file.map(l => l.split(" ")).map(l => (l(1), l(2).toInt)).reduceByKey(_+_ , 3)
reducedList: org.apache.spark.rdd.RDD[(java.lang.String, Int)] = MapPartitionsRDD[6] at reduceByKey at <console>:14

scala> reducedList.cache
res0: org.apache.spark.rdd.RDD[(java.lang.String, Int)] = MapPartitionsRDD[6] at reduceByKey at <console>:14

scala> val sortedList = reducedList.map(x => (x._2, x._1)).sortByKey(false).take(50)
```

The first line tells Spark which file to process. In the second line we split each line of the dataset into multiple fields, taking the first and the second fields (page title and pageview count) and perform a groupBy based on the key (pagetitle). The third line caches the data in memory in case we need to re-run this job. This eliminates the need to read our dataset from Amazon S3 again. The last line sorts the list and provides the result.

As the cluster contains only 1 machine, it may take a while to process the input. The output should look like :

```
sortedList: Array[(Int, java.lang.String)] = Array((328476,Special:Search), (217924,Main_Page),
(73900,Special:Random), (65047,404_error/),
(55814,%E3%83%A1%E3%82%A4%E3%83%B3%E3%83%9A%E3%83%BC%E3%82%B8),
(21521,Special:Export/Where_Is_My_Mind), (19722,Wikipedia:Portada),
(18312,%E7%89%B9%E5%88%A5:%E6%A4%9C%E7%B4%A2), (17080,Pagina_principale)
```

Note: For faster processing, please create cluster with minimum 3 machines.

11.Clean up.

a. Disconnect from the master node by terminating your SSH session.

b. Then execute the below command with your job id.

```
elastic-mapreduce --terminate -j j-367J67T8QGKAD  
                (Job id of your job)
```

or

clicking 'terminate' in the cluster.

Very important : Please terminate the cluster. Otherwise AWS charges can go high as 100\$ in a week time!