HOMEWORK 1

Name: Rahul Purushottam Gaonkar (rpg283)

Running Hadoop Job using Docker:

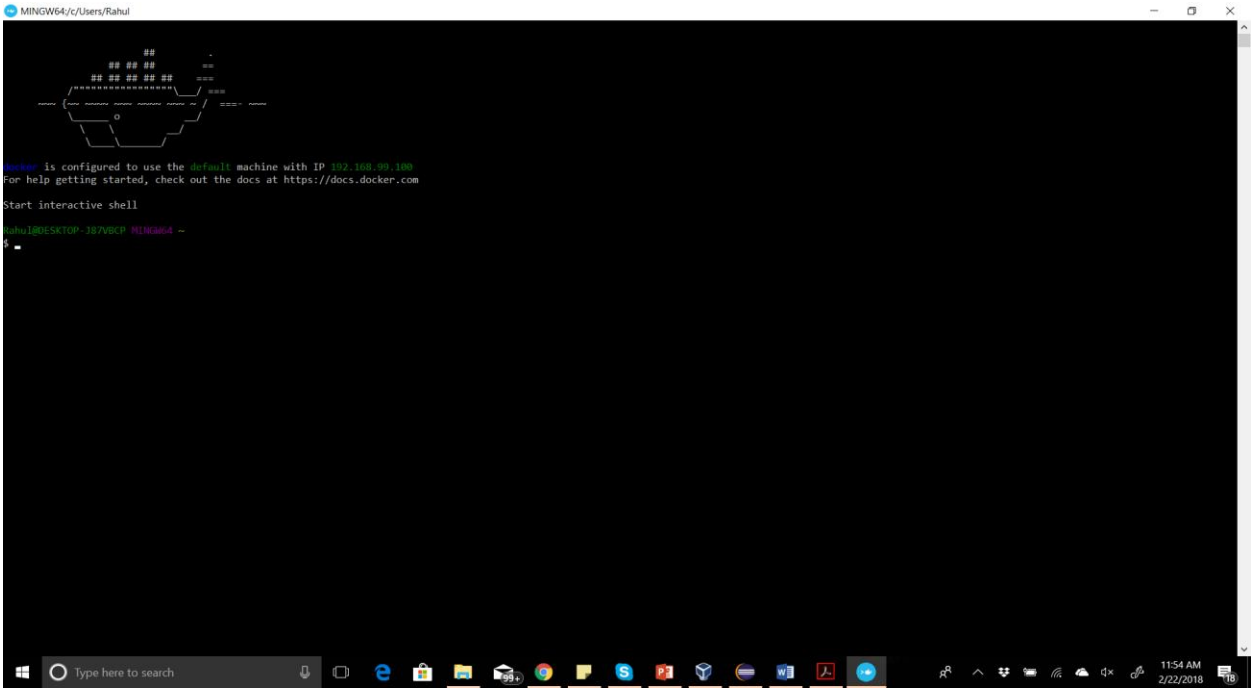


Figure1: Docker Running

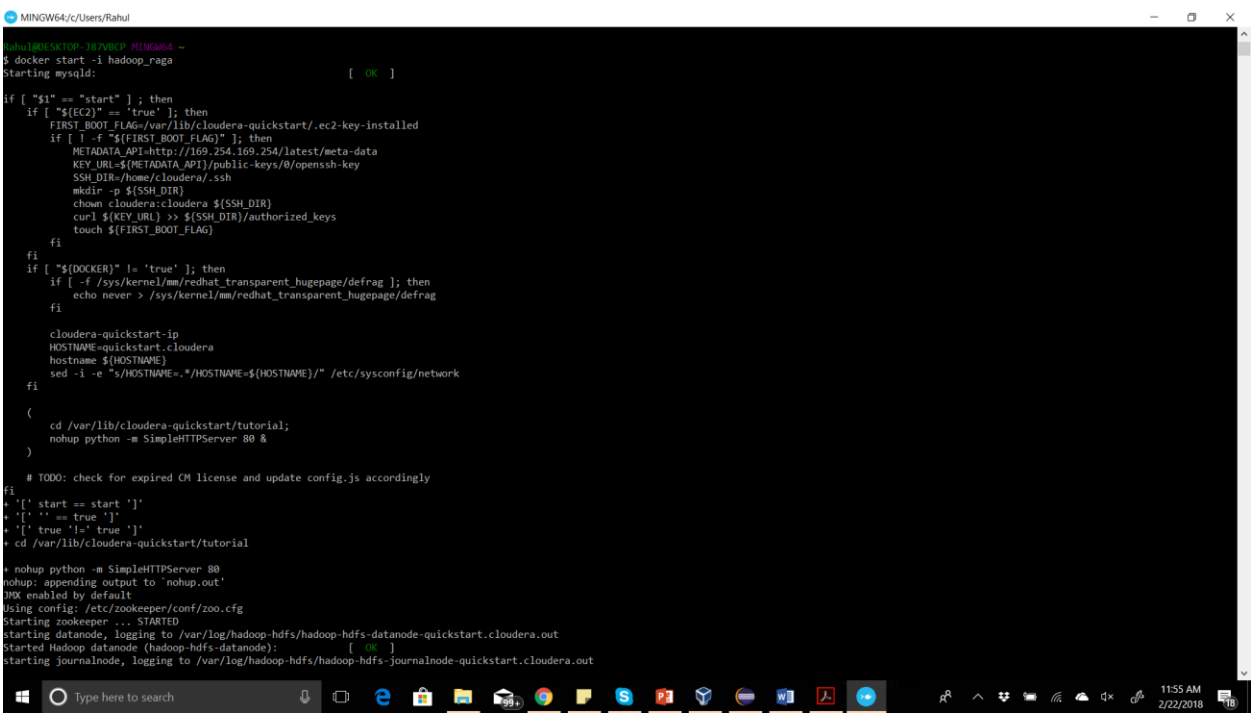


Figure 2: Starting Hadoop Container

```
@quickstart/
  setting CATALINA_BASE=/var/lib/oozie/tomcat-deployment
  setting OOZIE_HTTPS_PORT=11443
  setting OOZIE_HTTPS_KEYSTORE_PASS=password
  setting CATALINA_OPTS="$CATALINA_OPTS -Doozie.https.port=${OOZIE_HTTPS_PORT}"
  setting CATALINA_OPTS="$CATALINA_OPTS -Doozie.https.keystore.pass=${OOZIE_HTTPS_KEYSTORE_PASS}"
  setting CATALINA_OPTS="$CATALINA_OPTS -Xmx1024m"
  setting OOZIE_CONFIG=/etc/oozie/conf
  setting OOZIE_LOG=/var/log/oozie
Setting OOZIE_CONFIG_FILE:   oozie-site.xml
Using   OOZIE_DATA:          /var/lib/oozie
Using   OOZIE_LOG:           /var/log/oozie
Setting OOZIE_LOG4J_FILE:    oozie-log4j.properties
Setting OOZIE_LOG4J_RELOAD:  10
Setting OOZIE_HTTP_HOSTNAME: quickstart.cloudera
Setting OOZIE_HTTP_PORT:     11000
Setting OOZIE_ADMIN_PORT:     11001
Using   OOZIE_HTTPS_PORT:     11443
Setting OOZIE_BASE_URL:      http://quickstart.cloudera:11000/oozie
Using   CATALINA_BASE:       /var/lib/oozie/tomcat-deployment
Setting OOZIE_HTTPS_KEYSTORE_FILE:     /var/lib/oozie/.keystore
Using   OOZIE_HTTPS_KEYSTORE_PASS:     password
Setting OOZIE_INSTANCE_ID:       quickstart.cloudera
Setting CATALINA_OUT:        /var/log/oozie/catalina.out
Using   CATALINA_PID:        /var/run/oozie/oozie.pid

Using   CATALINA_OPTS:        -Doozie.https.port=11443 -Doozie.https.keystore.pass=password -Xmx1024m -Doozie.https.port=11443 -Doozie.https.keystore.pass=password -Xmx1024m -Dderby.stream.error.file=/var/log/oo
zie/derby.log
Adding to CATALINA_OPTS:     -Doozie.home.dir=/usr/lib/oozie -Doozie.config.dir=/etc/oozie/conf -Doozie.log.dir=/var/log/oozie -Doozie.data.dir=/var/lib/oozie -Doozie.instance.id=quickstart.cloudera -Doozie.conf
ig.file=oozie-site.xml -Doozie.log4j.file=oozie-log4j.properties -Doozie.log4j.reload=10 -Doozie.http.hostname=quickstart.cloudera -Doozie.admin.port=11001 -Doozie.http.port=11000 -Doozie.https.port=11443 -Doozi
e.base.url=http://quickstart.cloudera:11000/oozie -Doozie.https.keystore.file=/var/lib/oozie/.keystore -Doozie.https.keystore.pass=password -Djava.library.path=:/usr/lib/hadoop/lib/native:/usr/lib/hadoop/lib/nat
ive

Using CATALINA_BASE:   /var/lib/oozie/tomcat-deployment
Using CATALINA_HOME:   /usr/lib/bigtop-tomcat
Using CATALINA_TMPDIR: /var/lib/oozie
Using JRE_HOME:        /usr/java/jdk1.7.0_67-cloudera
Using CLASSPATH:       /usr/lib/bigtop-tomcat/bin/bootstrap.jar
Using CATALINA_PID:    /var/run/oozie/oozie.pid
Starting Solr server daemon:                               [  OK  ]
Using CATALINA_BASE:   /var/lib/solr/tomcat-deployment
Using CATALINA_HOME:   /usr/lib/solr/../bigtop-tomcat
Using CATALINA_TMPDIR: /var/lib/solr/
Using JRE_HOME:        /usr/java/jdk1.7.0_67-cloudera
Using CLASSPATH:       /usr/lib/solr/../bigtop-tomcat/bin/bootstrap.jar
Using CATALINA_PID:    /var/run/solr/solr.pid
Existing PID file found during start.
Removing/clearing stale PID file.
Started Impala Catalog Server (catalogd) :                 [  OK  ]
Started Impala Server (impalad):                           [  OK  ]
[root@quickstart /]#
```

Figure 3: Hadoop Container Started

```
@quickstart/
[root@quickstart /]# hadoop fs -ls
[root@quickstart /]# hadoop fs -mkdir rahul
[root@quickstart /]# hadoop fs -ls
Found 1 items
drwxr-xr-x   - root supergroup          0 2018-02-22 16:59 rahul
[root@quickstart /]#
```

Figure 4: Created a directory of name Rahul in Hadoop filesystem

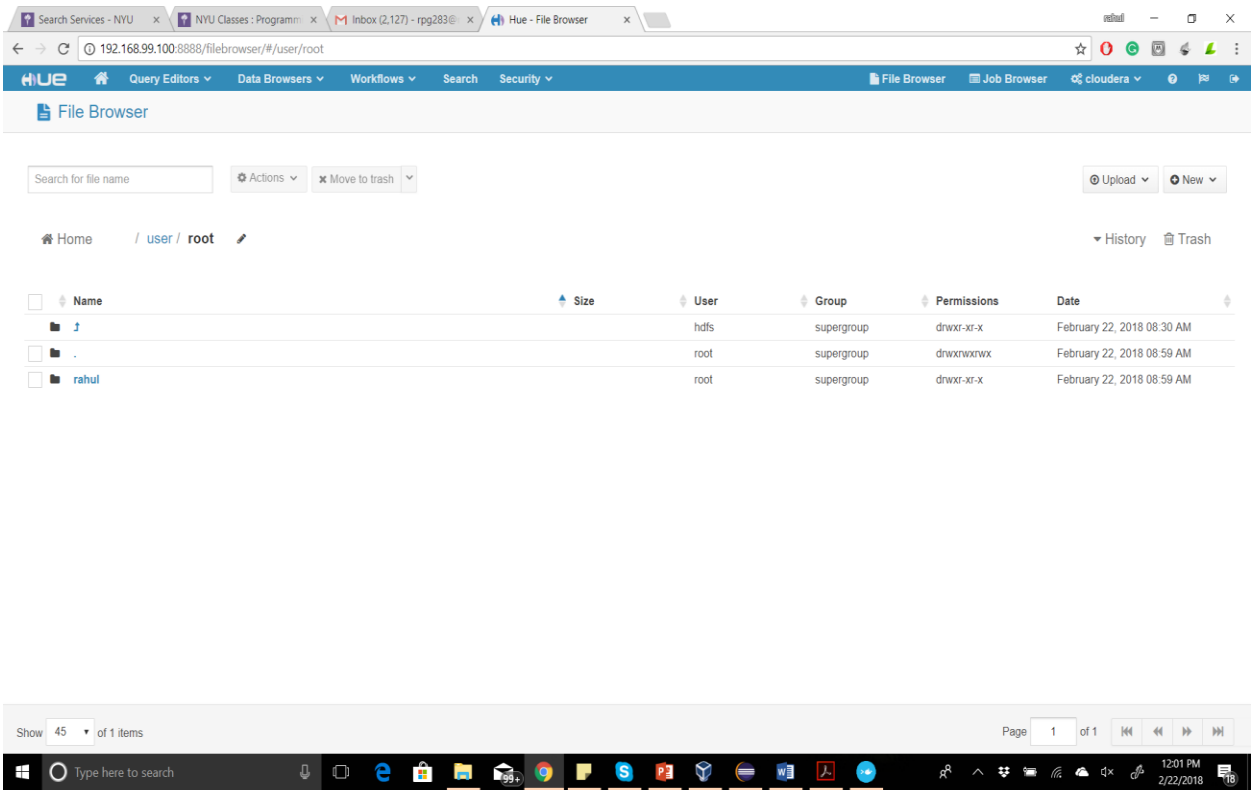


Figure 5: Directory name Rahul reflected in Hue

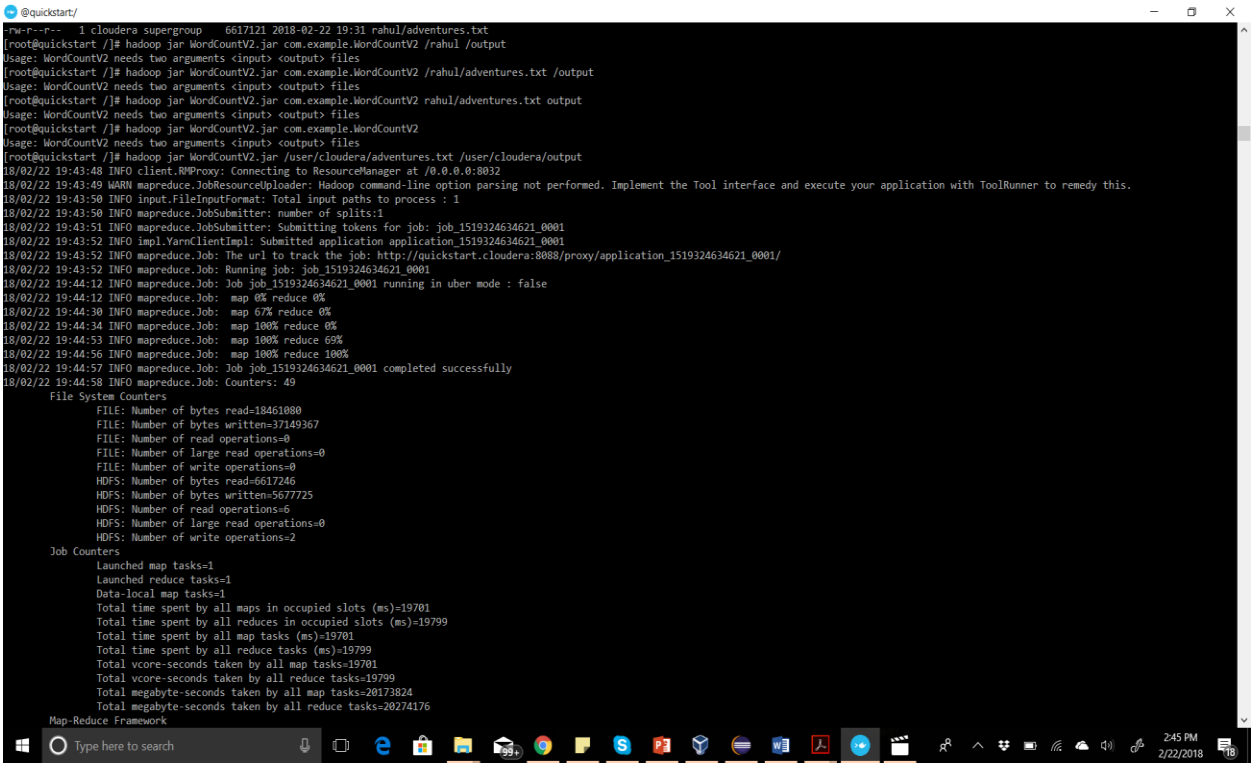
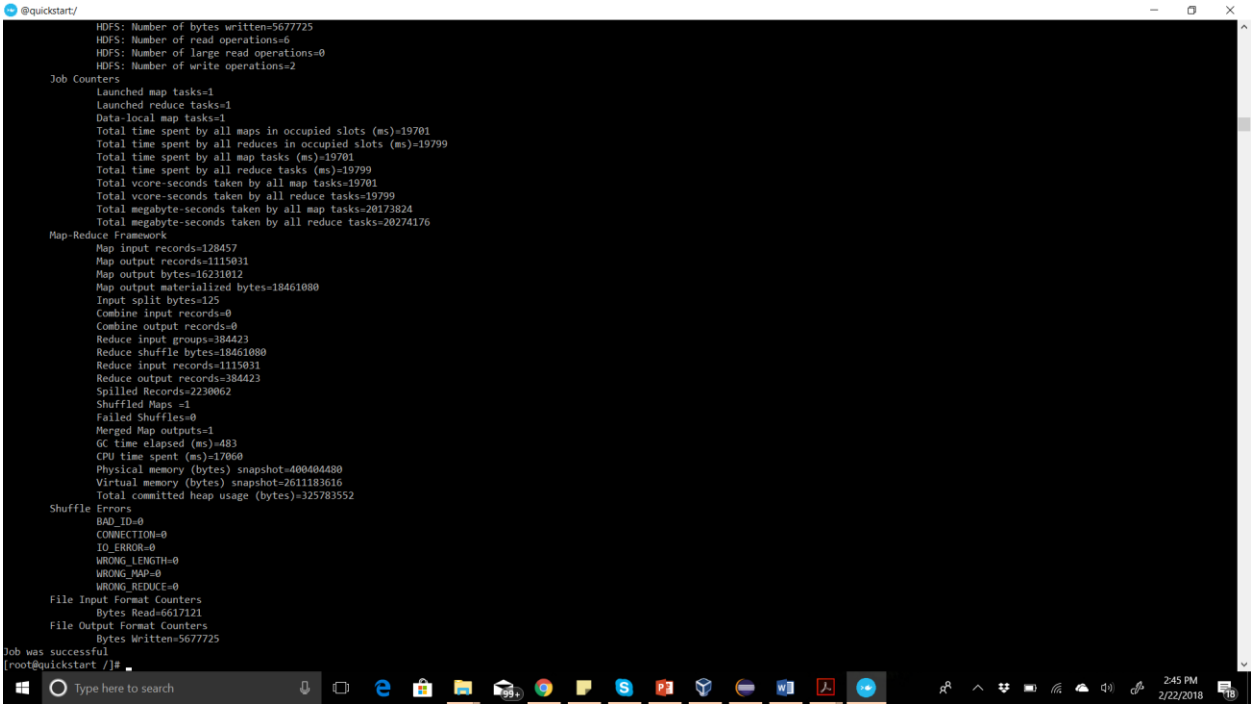


Figure 6: Running the MapReduce program (Job) of WordCountV2 to get the Phrase Count (i.e. Word Pair Count)

Command Used: hadoop jar WordCountV2.jar /user/cloudera/adventures.txt /user/cloudera/output

Placed the input file in user/cloudera and set the output path as /user/cloudera/output.



```
@quickstart/
                HDFS: Number of bytes written=5677725
                HDFS: Number of read operations=6
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
                Launched map tasks=1
                Launched reduce tasks=1
                Data-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=19701
                Total time spent by all reduces in occupied slots (ms)=19799
                Total time spent by all map tasks (ms)=19701
                Total time spent by all reduce tasks (ms)=19799
                Total vcore-seconds taken by all map tasks=19701
                Total vcore-seconds taken by all reduce tasks=19799
                Total megabyte-seconds taken by all map tasks=20173824
                Total megabyte-seconds taken by all reduce tasks=20274176
        Map-Reduce Framework
                Map input records=128457
                Map output records=1115031
                Map output bytes=16231012
                Map output materialized bytes=18461080
                Input split bytes=125
                Combine input records=0
                Combine output records=0
                Reduce input groups=384423
                Reduce shuffle bytes=18461080
                Reduce input records=1115031
                Reduce output records=384423
                Spilled Records=2230062
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=483
                CPU time spent (ms)=17060
                Physical memory (bytes) snapshot=400404480
                Virtual memory (bytes) snapshot=2611183616
                Total committed heap usage (bytes)=325783552
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=6617121
        File Output Format Counters
                Bytes Written=5677725
Job was successful
[root@quickstart /]#
```

Figure 7: Job was Successful

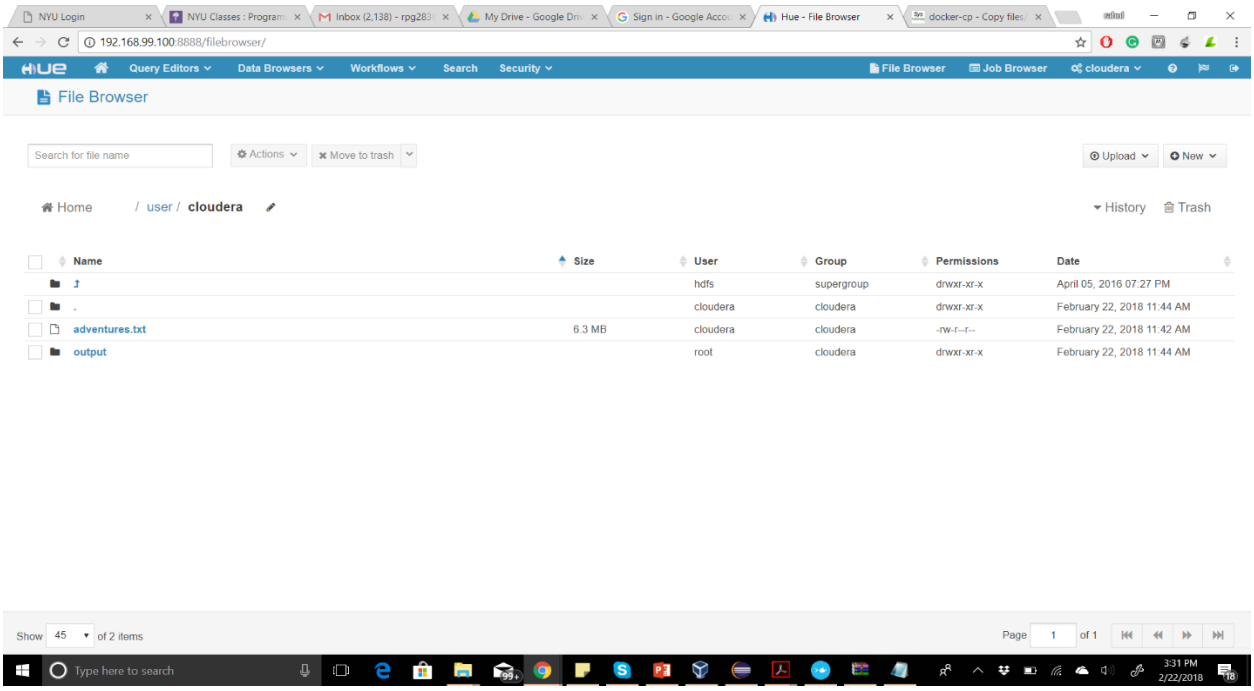


Figure 8: Output folder generated in HDFS (shown in hue)

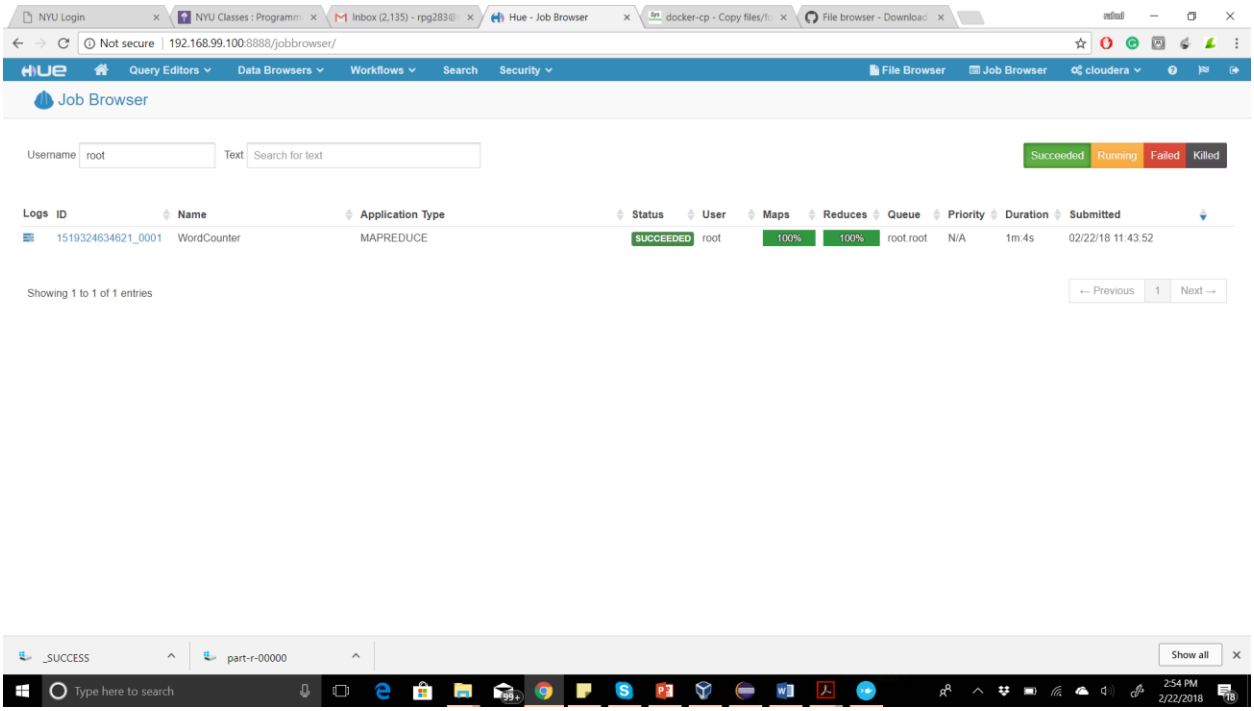


Figure 9: Job Successful Status in hue

Steps to run the Hadoop Job in Docker:

1. Place the WordCountV2.jar file in the hadoop container.
2. Place the adventures.txt file in the HDFS. I had placed it in user/cloudera/adventures.txt
3. Then run the job by executing the command **hadoop jar WordCountV2.jar /user/cloudera/adventures.txt /user/cloudera/output**
4. The output path argument should be a unique path, or it will throw an exception. I have mentioned **/user/cloudera/output** where I will get the output file generated.

Running Hadoop Job using AWS EMR:

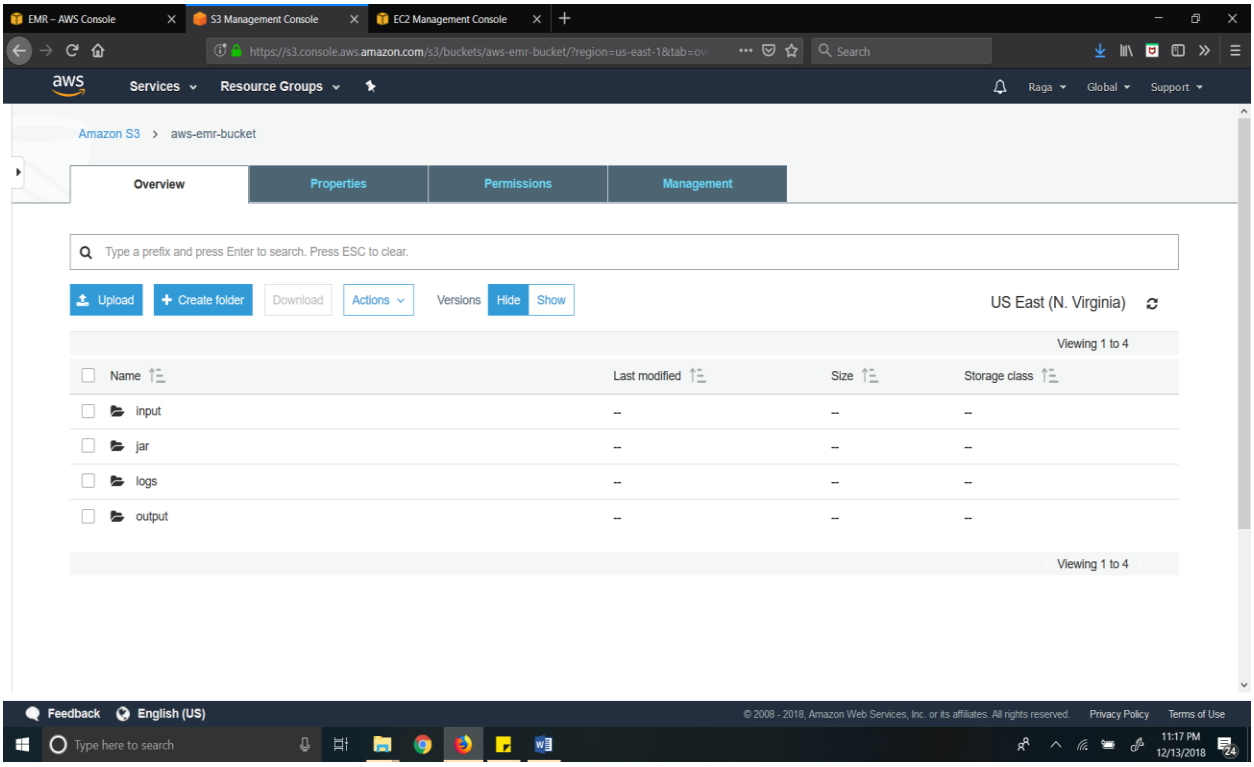
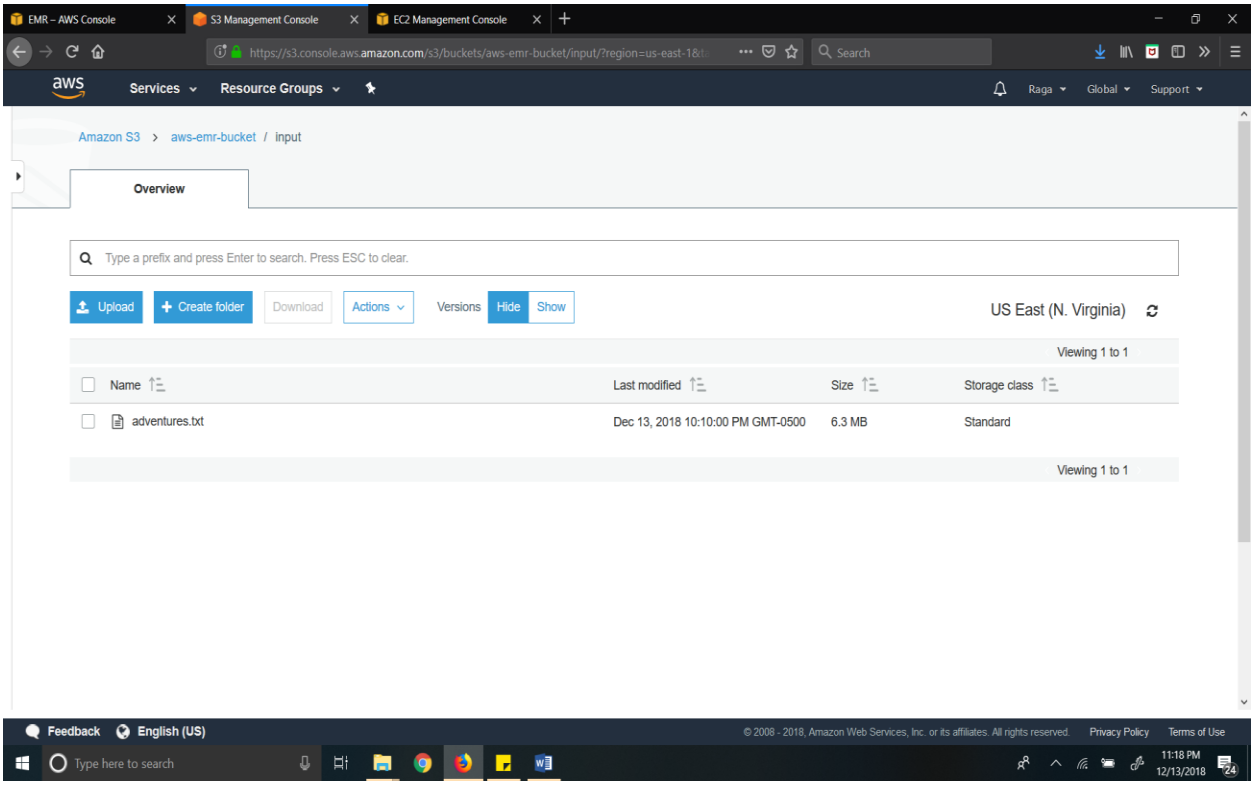


Figure 1: Input Folder (Input File), jar folder (WordCountV2 jar), output folder and log folder (AWS EMR Logs) generated in S3 bucket



EMR – AWS Console | S3 Management Console | EC2 Management Console

https://s3.console.aws.amazon.com/s3/buckets/aws-emr-bucket/jar/?region=us-east-1&tab

Services | Resource Groups | Raga | Global | Support

Amazon S3 > aws-emr-bucket / jar

Overview

Type a prefix and press Enter to search. Press ESC to clear.

Upload | Create folder | Download | Actions | Versions Hide Show | US East (N. Virginia)

Viewing 1 to 1

Name	Last modified	Size	Storage class
WordCountV2.jar	Dec 13, 2018 10:10:34 PM GMT-0500	6.3 KB	Standard

Viewing 1 to 1

Feedback | English (US) | © 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved. | Privacy Policy | Terms of Use

Type here to search | 11:18 PM 12/13/2018

EMR – AWS Console | S3 Management Console | EC2 Management Console

https://s3.console.aws.amazon.com/s3/buckets/aws-emr-bucket/logs/j-1ZQVU86EO7S56/?re

Services | Resource Groups | Raga | Global | Support

Amazon S3 > aws-emr-bucket / logs / j-1ZQVU86EO7S56

Overview

Type a prefix and press Enter to search. Press ESC to clear.

Upload | Create folder | Download | Actions | Versions Hide Show | US East (N. Virginia)

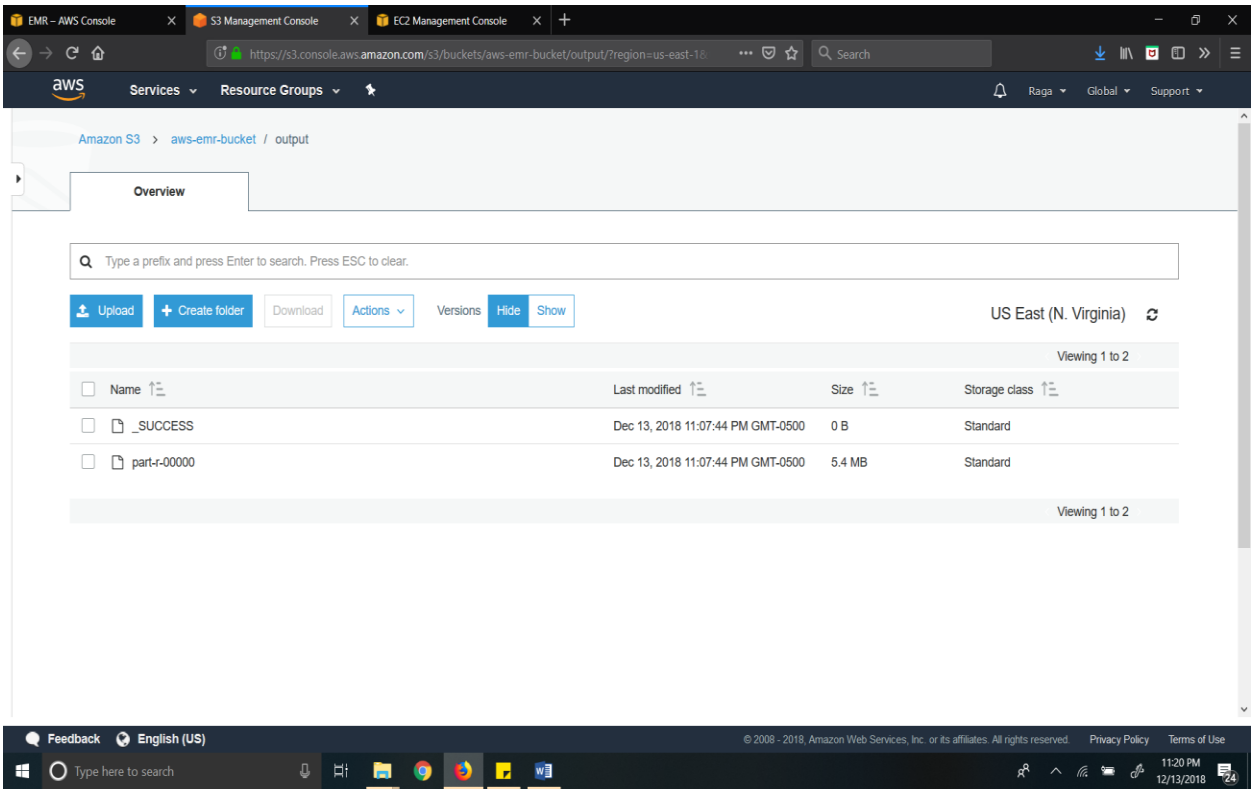
Viewing 1 to 4

Name	Last modified	Size	Storage class
containers	--	--	--
hadoop-mapreduce	--	--	--
node	--	--	--
steps	--	--	--

Viewing 1 to 4

Feedback | English (US) | © 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved. | Privacy Policy | Terms of Use

Type here to search | 11:20 PM 12/13/2018



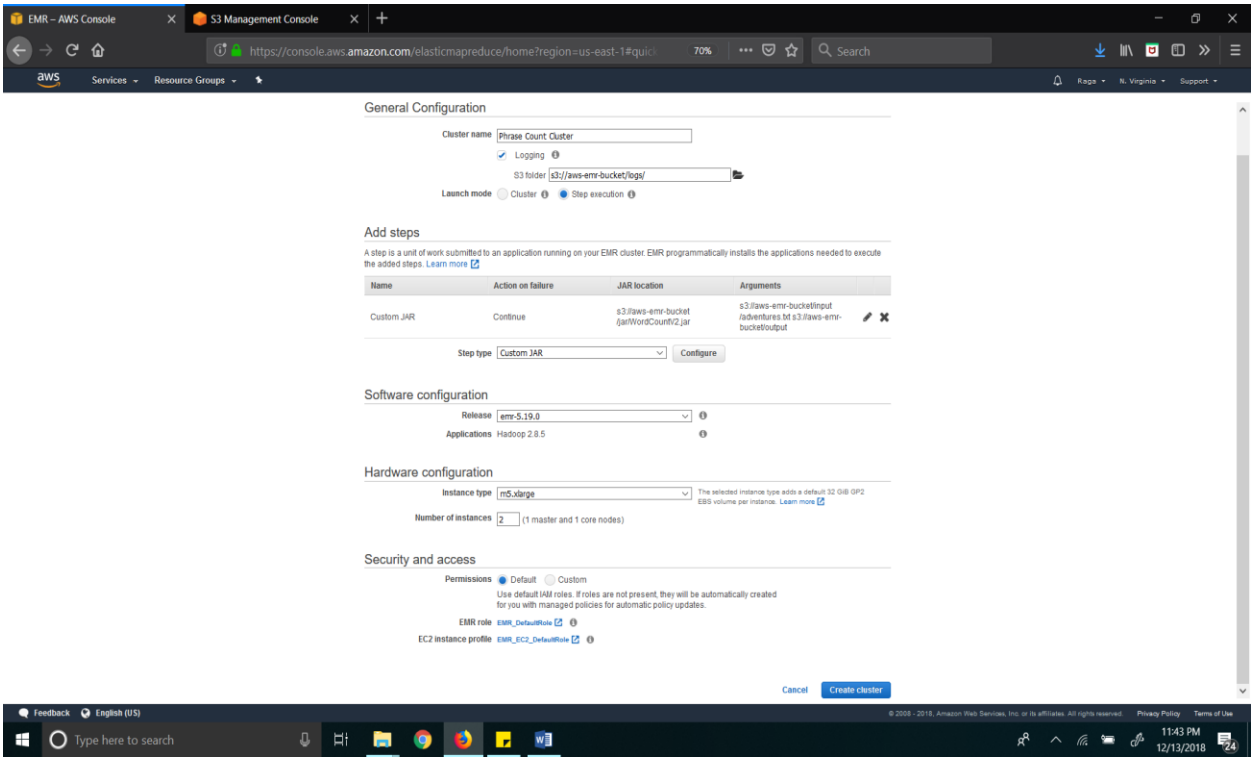


Figure 2: Setting General Configuration for creating the AWS EMR Cluster

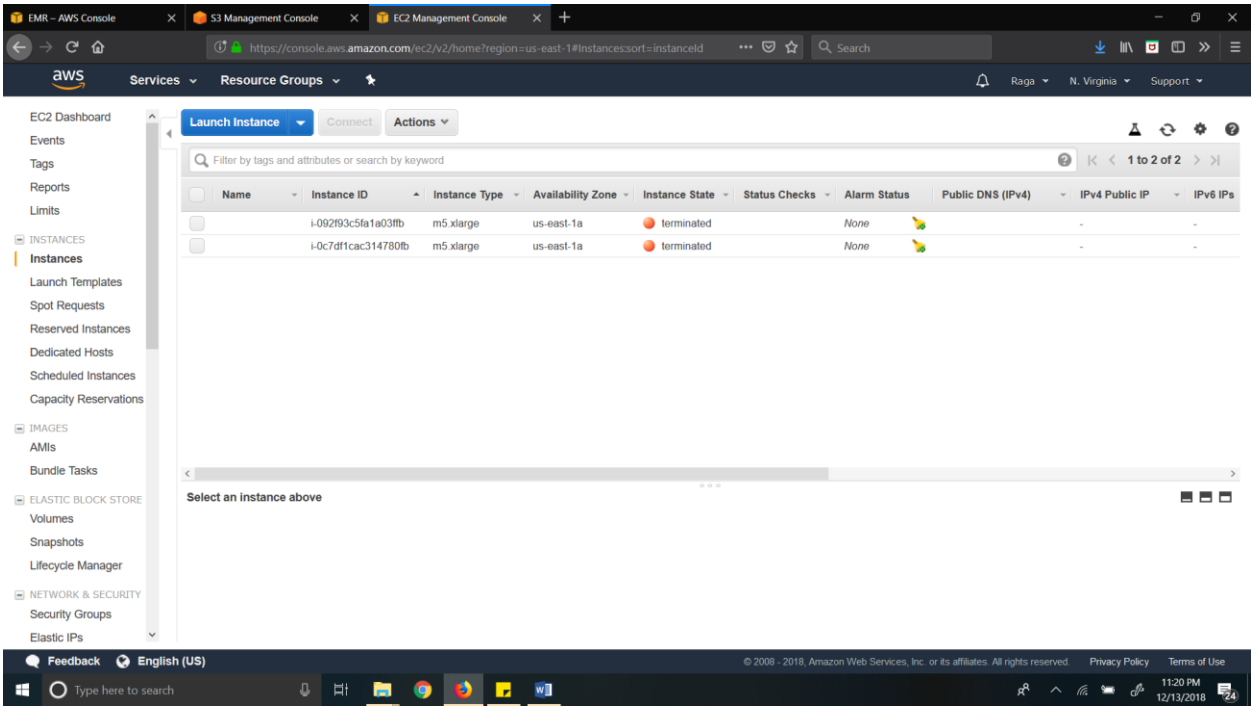


Figure 3: EC2 instances created automatically after creating the AWS EMR cluster

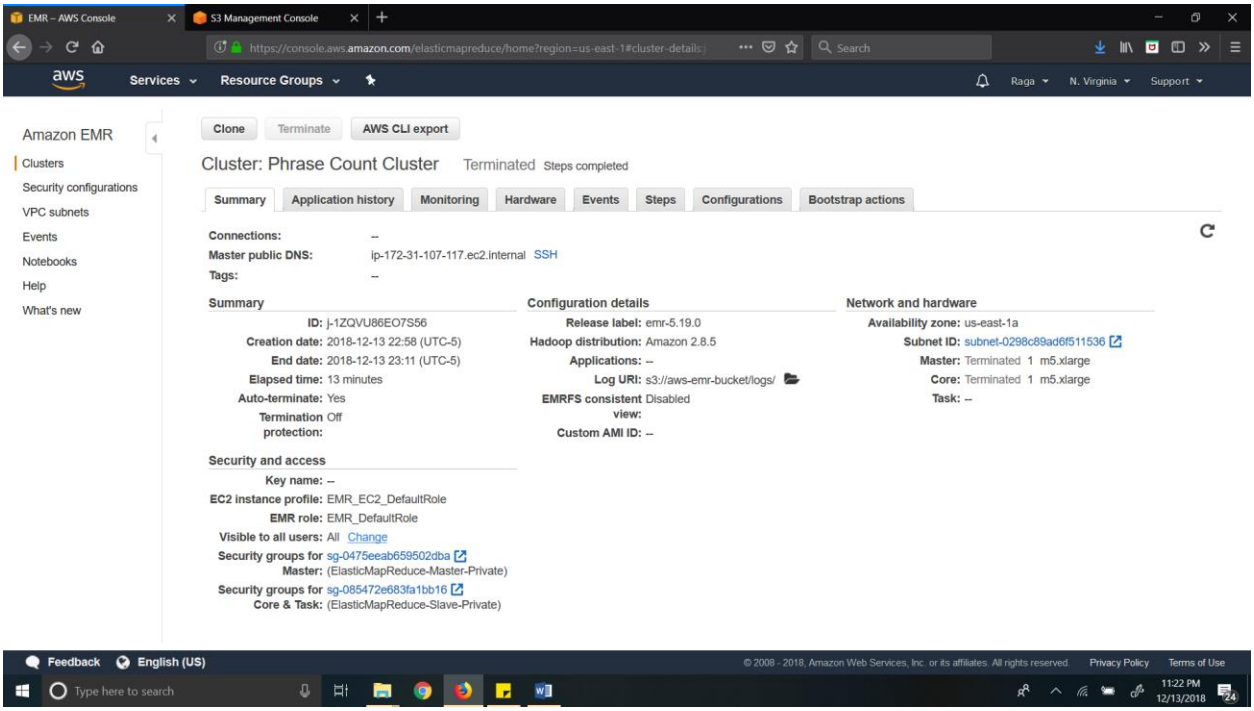


Figure 4: Viewing Cluster configuration after creating the AWS EMR Cluster

EMR – AWS Console | S3 Management Console

https://console.aws.amazon.com/elasticmapreduce/home?region=us-east-1#cluster-details:

Services | Resource Groups | Raga | N. Virginia | Support

Amazon EMR

- Clusters
- Security configurations
- VPC subnets
- Events
- Notebooks
- Help
- What's new

Clone | Terminate | AWS CLI export

Cluster: Phrase Count Cluster

Terminated Steps completed

Summary | Application history | Monitoring | Hardware | Events | Steps | Configurations | Bootstrap actions

Amazon EMR collects information from YARN applications on your cluster and keeps historical information for up to seven days after applications have completed. Detailed application history is only available for Spark. Learn more

YARN applications (1)

Filter: All applications | Filter applications ... | 1 applications (all loaded)

Application ID	Type	Action	Status	Start time (UTC-5)	Duration	Finish time (UTC-5)	User
application_1544760317813_0001	MapReduce	WordCounter	Succeeded	2018-12-13 23:07 (UTC-5)	31 s	2018-12-13 23:07 (UTC-5)	hadoop

Diagnostics: Succeeded

Feedback | English (US)

 Privacy Policy | Terms of Use

Type here to search | 11:23 PM 12/13/2018

EMR – AWS Console | S3 Management Console

https://console.aws.amazon.com/elasticmapreduce/home?region=us-east-1#cluster-details:

Services | Resource Groups | Raga | N. Virginia | Support

Amazon EMR

- Clusters
- Security configurations
- VPC subnets
- Events
- Notebooks
- Help
- What's new

Clone | Terminate | AWS CLI export

Cluster: Phrase Count Cluster

Terminated Steps completed

Summary | Application history | Monitoring | Hardware | Events | Steps | Configurations | Bootstrap actions

Add task instance group

Instance groups

Filter: Filter instance groups ... | 2 instance groups (all loaded)

ID	Status	Node type & name	Instance type	Instance count	Purchasing opti
ig-2U1W5QEAZA4B4	Terminated (1 Requested)	MASTER Master Instance Group	m5.xlarge 4 vCore, 16 GiB memory, EBS only storage EBS Storage: 32 GiB	0 Instances	On-demand
ig-27JNRKU4OXU70	Terminated (1 Requested)	CORE Core Instance Group	m5.xlarge 4 vCore, 16 GiB memory, EBS only storage EBS Storage: 32 GiB	0 Instances	On-demand

Feedback | English (US)

 Privacy Policy | Terms of Use

Type here to search | 11:23 PM 12/13/2018

EMR – AWS Console | S3 Management Console

https://console.aws.amazon.com/elasticmapreduce/home?region=us-east-1#cluster-details

Services | Resource Groups | Raga | N. Virginia | Support

Amazon EMR

- Clusters
- Security configurations
- VPC subnets
- Events
- Notebooks
- Help
- What's new

Clone | Terminate | AWS CLI export

Cluster: Phrase Count Cluster

Terminated Steps completed

Summary | Application history | Monitoring | Hardware | Events | Steps | Configurations | Bootstrap actions

Time	Event description	Source ID	Source type	Event type	Severity	Full date & time
Dec 13 11:12 PM	Amazon EMR Cluster j-1ZQVU86EO7S56 (Phrase Count Cluster) has terminated at 2018-12-14 04:11 UTC with a reason of ALL_STEPS_COMPLETED.	j-1ZQVU86EO7S56	Cluster	Cluster State Change	INFO	December 13, 2018 at 11:12:13 PM (UTC-5)
Dec 13 11:08 PM	Amazon EMR cluster j-1ZQVU86EO7S56 (Phrase Count Cluster) finished running all pending steps at 2018-12-14 04:07 UTC.	j-1ZQVU86EO7S56	Cluster	Cluster State Change	INFO	December 13, 2018 at 11:08:19 PM (UTC-5)
Dec 13 11:08 PM	Step s-ZOB9482I7FOO (Custom JAR) in Amazon EMR cluster j-1ZQVU86EO7S56 (Phrase Count Cluster) completed execution at 2018-12-14 04:07 UTC. The step started running at 2018-12-14 04:07 UTC and took 0 minutes to complete.	s-ZOB9482I7FOO	Step	Step State Change	INFO	December 13, 2018 at 11:08:17 PM (UTC-5)
Dec 13 11:07 PM	Step s-ZOB9482I7FOO (Custom JAR) in Amazon EMR cluster j-1ZQVU86EO7S56 (Phrase Count Cluster) started running at 2018-12-14 04:07 UTC.	s-ZOB9482I7FOO	Step	Step State Change	INFO	December 13, 2018 at 11:07:26 PM (UTC-5)
Dec 13 11:07 PM	Step s-2H33HGZECJIRI (Setup hadoop debugging) in Amazon EMR cluster j-1ZQVU86EO7S56 (Phrase Count Cluster) completed execution at 2018-12-14 04:07 UTC. The step started running at 2018-12-14 04:07 UTC and took 0 minutes to complete.	s-2H33HGZECJIRI	Step	Step State Change	INFO	December 13, 2018 at 11:07:25 PM (UTC-5)
Dec 13 11:07 PM	Amazon EMR cluster j-1ZQVU86EO7S56 (Phrase Count Cluster) began running steps at 2018-12-14 04:06 UTC.	j-1ZQVU86EO7S56	Cluster	Cluster State Change	INFO	December 13, 2018 at 11:07:05 PM (UTC-5)

Feedback | English (US)

Privacy Policy | Terms of Use

Type here to search

11:23 PM 12/13/2018

EMR – AWS Console | S3 Management Console

https://console.aws.amazon.com/elasticmapreduce/home?region=us-east-1#cluster-details

Services | Resource Groups | Raga | N. Virginia | Support

Amazon EMR

- Clusters
- Security configurations
- VPC subnets
- Events
- Notebooks
- Help
- What's new

Clone | Terminate | AWS CLI export

Cluster: Phrase Count Cluster

Terminated Steps completed

Summary | Application history | Monitoring | Hardware | Events | Steps | Configurations | Bootstrap actions

Add step | Clone step | Cancel step

Steps

View all interactive jobs | View all jobs

Filter: All steps | Filter steps ... | 2 steps (all loaded)

ID	Name	Status	Start time (UTC-5)	Elapsed time	Log files	Actions
s-ZOB9482I7FOO	Custom JAR	Completed	2018-12-13 23:07 (UTC-5)	38 seconds	View logs	View jobs

JAR location : s3://aws-emr-bucket/jar/WordCountV2.jar
Main class : None
Arguments : s3://aws-emr-bucket/input/adventures.txt s3://aws-emr-bucket/output
Action on failure: Continue

ID	Name	Status	Start time (UTC-5)	Elapsed time	Log files	Actions
s-2H33HGZECJIRI	Setup hadoop debugging	Completed	2018-12-13 23:07 (UTC-5)	2 seconds	View logs	View jobs

JAR location : command-runner.jar
Main class : None
Arguments : state-pusher-script
Action on failure: Terminate cluster

Feedback | English (US)

Privacy Policy | Terms of Use

Type here to search

11:25 PM 12/13/2018

Steps to run the Hadoop Job in AWS EMR Cluster:

1. Place the **adventures.txt** file and **WordCountV2.jar** in the **input** and **jar** folder of S3 bucket respectively.
2. Create the **logs** and **output** folder for AWS EMR logs and Hadoop job output respectively.
3. Set the following general configuration settings while creating the AWS EMR cluster:
   a. Set the AWS EMR cluster log path to the log folder of S3 bucket.
   b. Select the launch mode as **Step Execution**.
   c. Select the Step Type as **Custom JAR.**
   d. Provide the **WordCountV2.jar S3 bucket location** in the JAR location and the input and output location in the Arguments.
   e. Select the EC2 instance type and number of instances for the job.
   f. Once we click on create cluster, AWS creates the EC2 instances automatically and runs the Hadoop job and generates the output in the output folder and finally terminates the EC2 instances once the job is completed.