Name: Deep Pawar(A20545137)
Professor: Joseph Rosen
Institute: Illinois Institute of Technology

CSP 554: Big Data Technologies

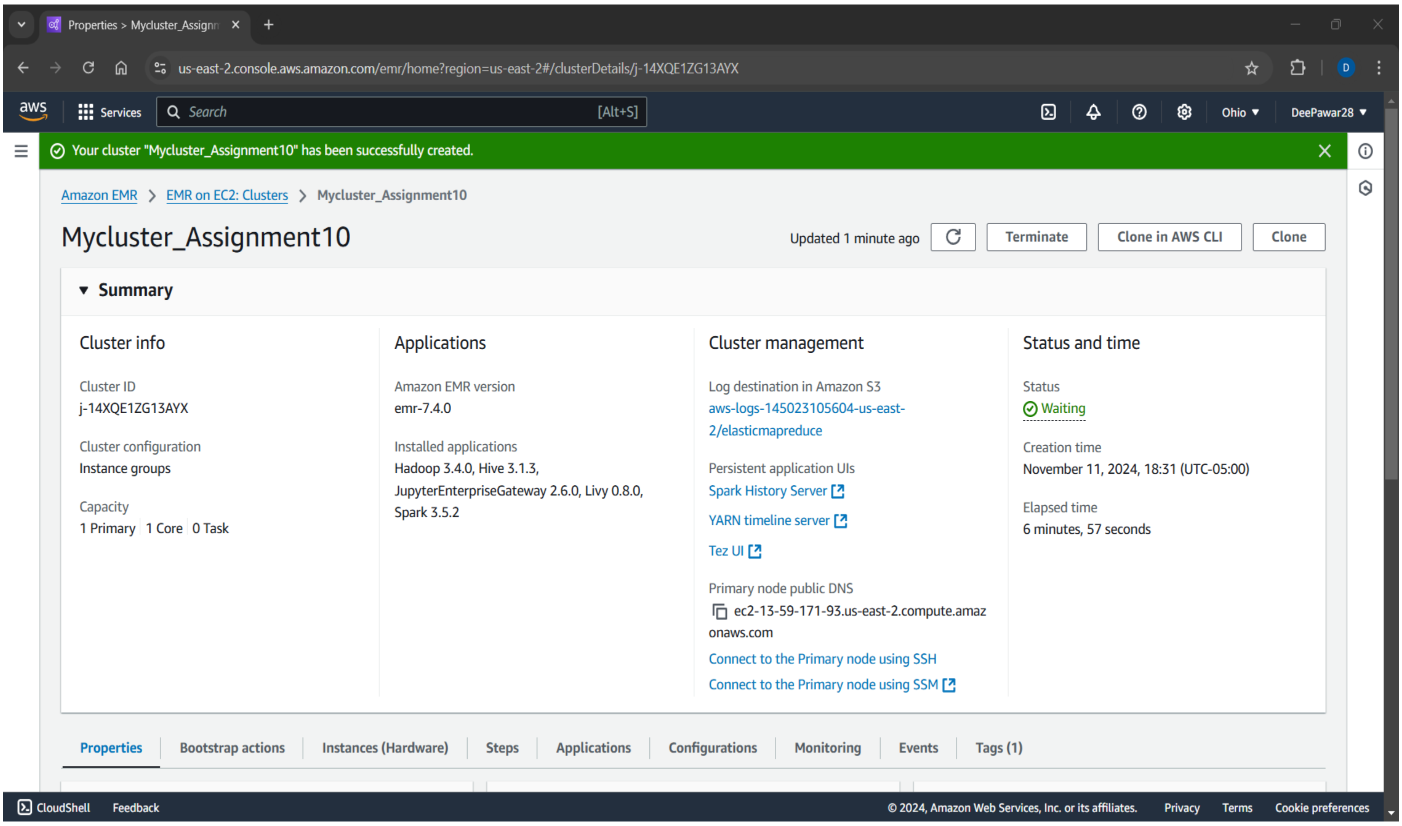
Fall 2024 - Assignment 10

- **Questions and Answers:**

Exercise 1) 5 points extra credit

Follow the document "Spark Streaming Demo Instructions" included with this assignment and execute the demo code. Provide enough screen shots to indicate you have completed the document through step 12. Then remember to terminate your VM.

1) Start up an EMR cluster as previously, but instead of choosing the "Core Hadoop" configuration choose the "Spark" configuration (see below), otherwise proceed as before.



Your cluster "Mycluster_Assignment10" has been successfully created.

Amazon EMR > EMR on EC2: Clusters > Mycluster_Assignment10

Mycluster_Assignment10

Updated 1 minute ago

Summary

Cluster info

Cluster ID
j-14XQE1ZG13AYX

Cluster configuration
Instance groups

Capacity
1 Primary | 1 Core | 0 Task

Applications

Amazon EMR version
emr-7.4.0

Installed applications
Hadoop 3.4.0, Hive 3.1.3, JupyterEnterpriseGateway 2.6.0, Livy 0.8.0, Spark 3.5.2

Cluster management

Log destination in Amazon S3
aws-logs-145023105604-us-east-2/elasticmapreduce

Persistent application UIs
Spark History Server
YARN timeline server
Tez UI

Primary node public DNS
ec2-13-59-171-93.us-east-2.compute.amazonaws.com

Connect to the Primary node using SSH
Connect to the Primary node using SSM

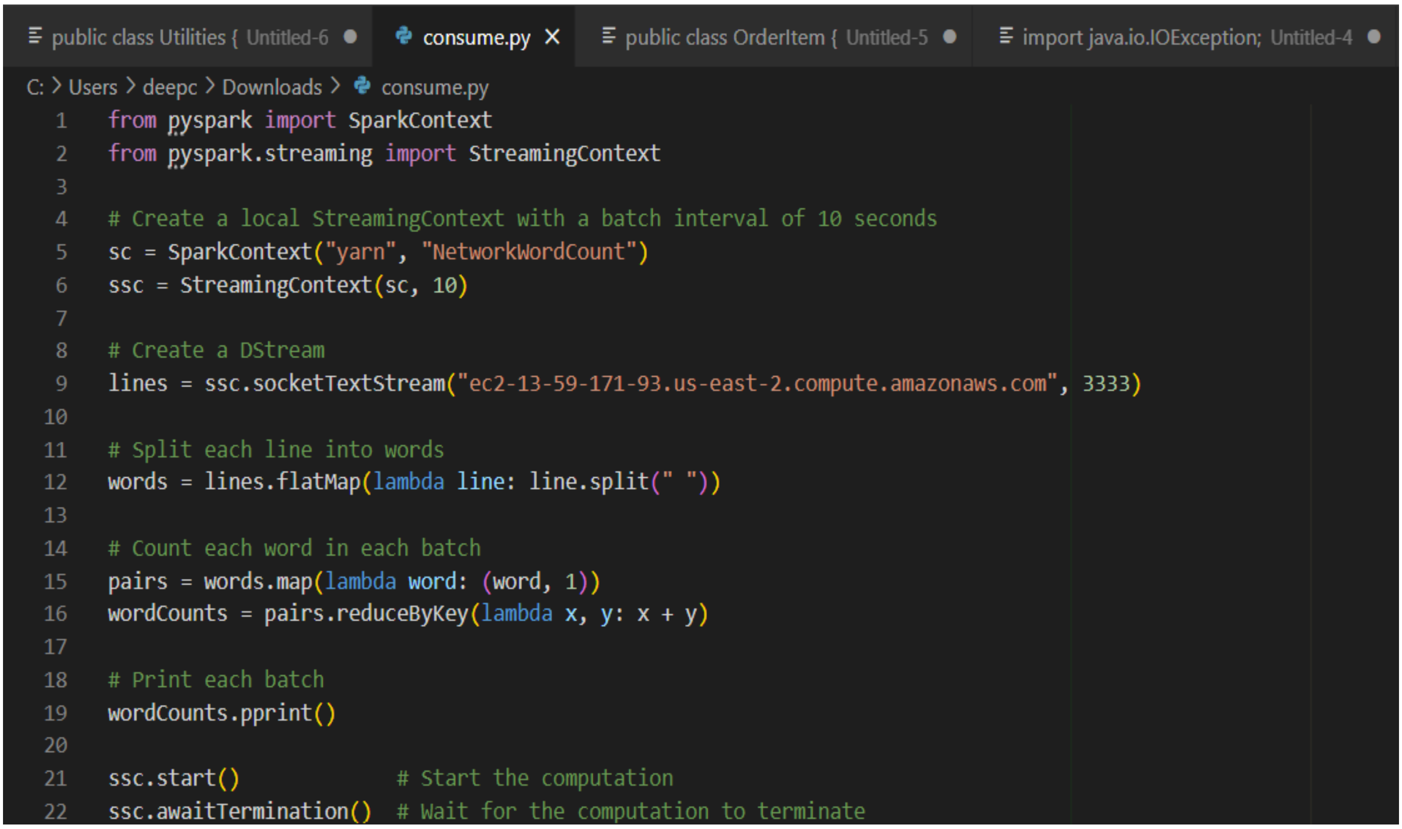
Status and time

Status
Waiting

Creation time
November 11, 2024, 18:31 (UTC-05:00)

Elapsed time
6 minutes, 57 seconds

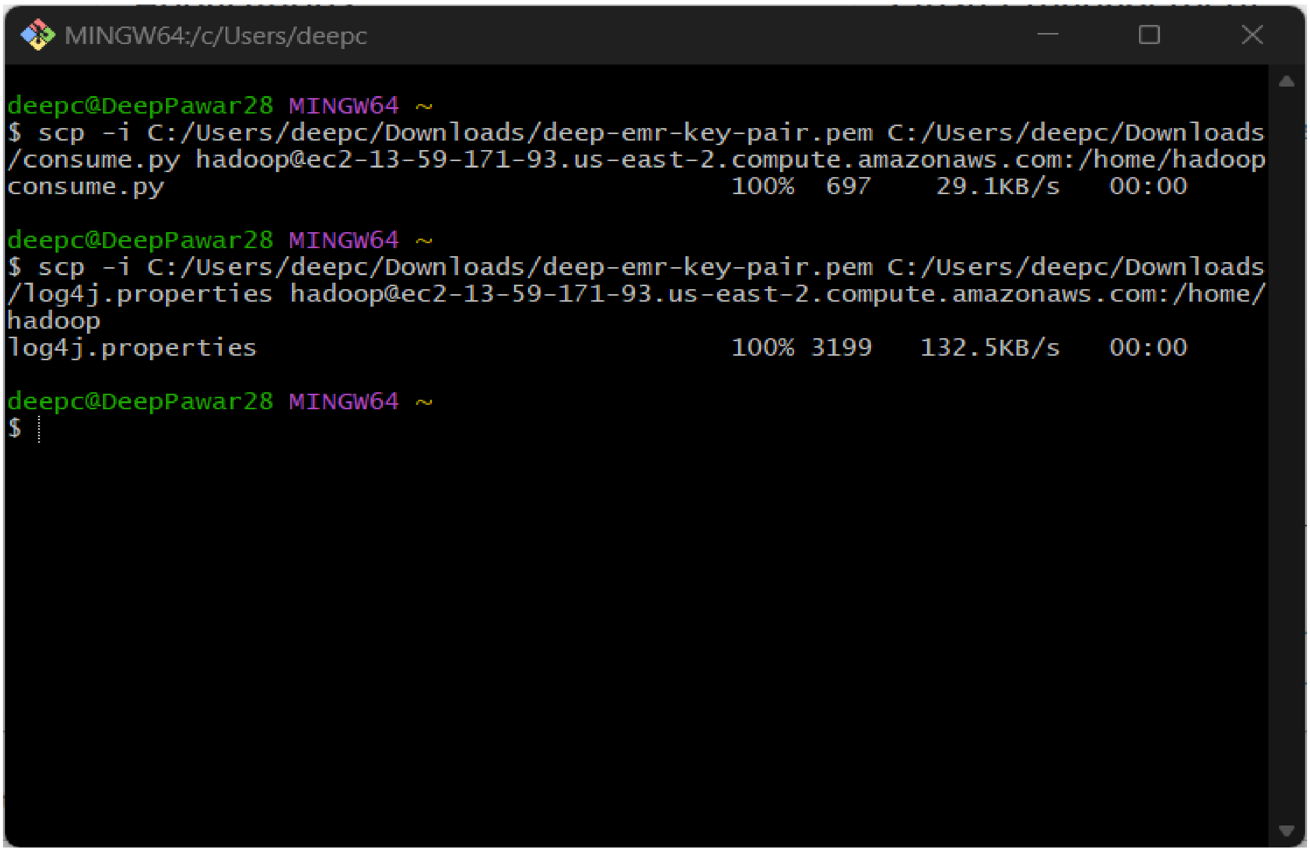
4) There is one item you must change in consume.py. In the following line, you must replace <Master public DNS> with your own public DNS name (found as described above)


public class Utilities { Untitled-6 ●
consume.py ×
public class OrderItem { Untitled-5 ●
import java.io.IOException; Untitled-4 ●
C: > Users > deepc > Downloads > consume.py

```python
from pyspark import SparkContext
from pyspark.streaming import StreamingContext

# Create a local StreamingContext with a batch interval of 10 seconds
sc = SparkContext("yarn", "NetworkWordCount")
ssc = StreamingContext(sc, 10)

# Create a DStream
lines = ssc.socketTextStream("ec2-13-59-171-93.us-east-2.compute.amazonaws.com", 3333)

# Split each line into words
words = lines.flatMap(lambda line: line.split(" "))

# Count each word in each batch
pairs = words.map(lambda word: (word, 1))
wordCounts = pairs.reduceByKey(lambda x, y: x + y)

# Print each batch
wordCounts.pprint()

ssc.start()            # Start the computation
ssc.awaitTermination() # Wait for the computation to terminate
```

5) scp this modified consume.py file to your EMR cluster primary (master) node. You may need to answer a security question with “Y/y” or “Yes”.

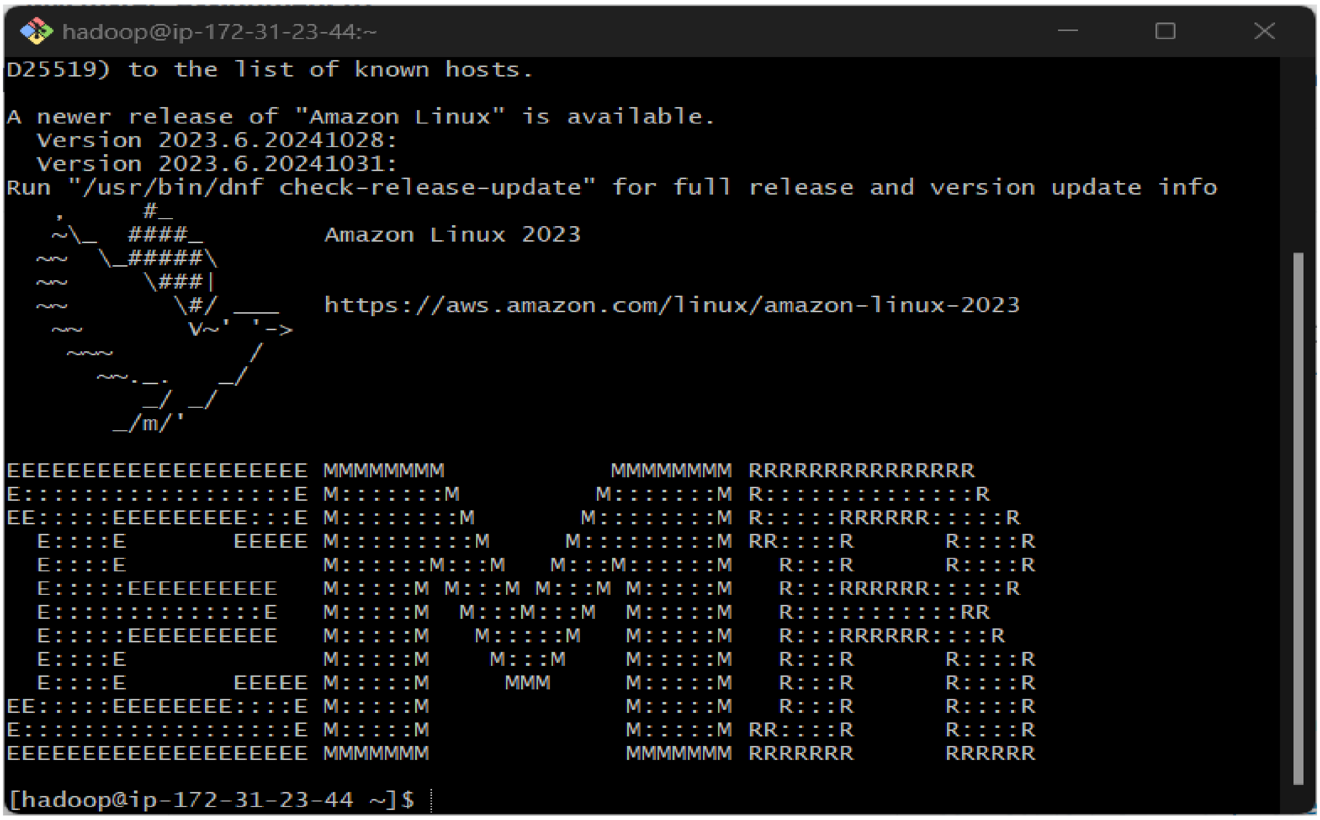
6) Then scp the file log4j.properties to your EMR cluster primary (master) node.



```
deepc@DeepPawar28 MINGW64 ~
$ scp -i C:/Users/deepc/Downloads/deep-emr-key-pair.pem C:/Users/deepc/Downloads
/consume.py hadoop@ec2-13-59-171-93.us-east-2.compute.amazonaws.com:/home/hadoop
consume.py                                    100%   697     29.1KB/s    00:00

deepc@DeepPawar28 MINGW64 ~
$ scp -i C:/Users/deepc/Downloads/deep-emr-key-pair.pem C:/Users/deepc/Downloads
/log4j.properties hadoop@ec2-13-59-171-93.us-east-2.compute.amazonaws.com:/home/
hadoop
log4j.properties                              100% 3199    132.5KB/s    00:00

deepc@DeepPawar28 MINGW64 ~
$
```

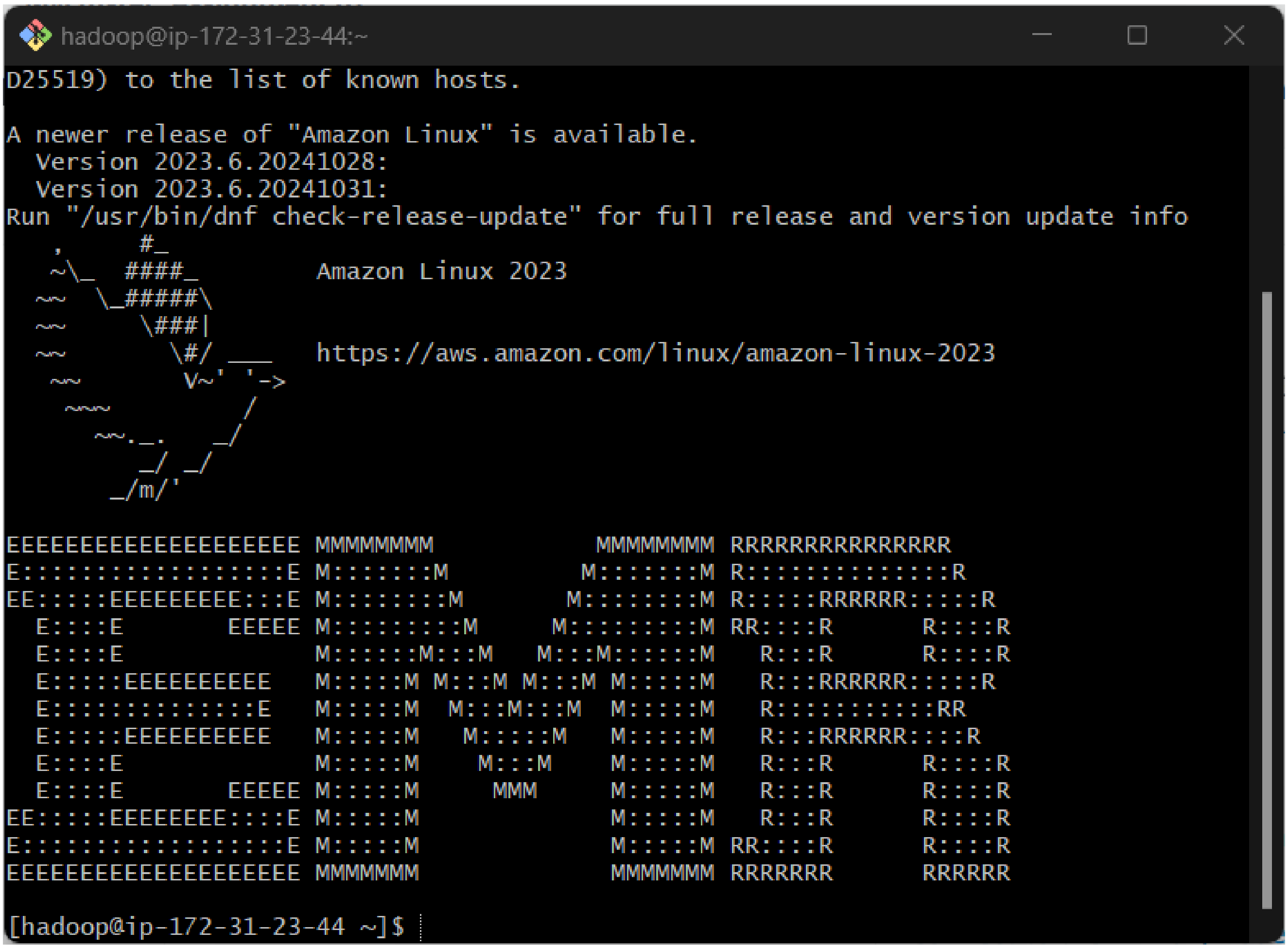
7) Open two terminal sessions to the EMR primary node. We will call one the EC2-1 window and the other the EC2-2 window.

- **EC2-1 window**



```
D25519) to the list of known hosts.

A newer release of "Amazon Linux" is available.
  Version 2023.6.20241028:
  Version 2023.6.20241031:
Run "/usr/bin/dnf check-release-update" for full release and version update info
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'

EEEEEEEEEEEEEEEEEEEE MMMMMMMM           MMMMMMMM RRRRRRRRRRRRRRR
E::::::::::::::::::E M:::::::M         M:::::::M R::::::::::::::R
EE:::::EEEEEEEEE:::E M::::::::M       M::::::::M R:::::RRRRRR:::::R
  E::::E       EEEEE M:::::::::M     M:::::::::M RR::::R      R::::R
  E::::E             M::::::M:::M   M:::M::::::M   R:::R      R::::R
  E:::::EEEEEEEEEE   M:::::M M:::M M:::M M:::::M   R:::RRRRRR:::::R
  E::::::::::::::E   M:::::M  M:::M:::M  M:::::M   R:::::::::::RR
  E:::::EEEEEEEEEE   M:::::M   M:::::M   M:::::M   R:::RRRRRR::::R
  E::::E             M:::::M    M:::M    M:::::M   R:::R      R::::R
  E::::E       EEEEE M:::::M     MMM     M:::::M   R:::R      R::::R
EE:::::EEEEEEEE::::E M:::::M             M:::::M   R:::R      R::::R
E::::::::::::::::::E M:::::M             M:::::M RR::::R      R::::R
EEEEEEEEEEEEEEEEEEEE MMMMMMM             MMMMMMM RRRRRRR      RRRRRR

[hadoop@ip-172-31-23-44 ~]$
```

- **EC2-2 window**



```
D25519) to the list of known hosts.

A newer release of "Amazon Linux" is available.
  Version 2023.6.20241028:
  Version 2023.6.20241031:
Run "/usr/bin/dnf check-release-update" for full release and version update info
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'

EEEEEEEEEEEEEEEEEEEE MMMMMMMM           MMMMMMMM RRRRRRRRRRRRRRR
E::::::::::::::::::E M:::::::M         M:::::::M R::::::::::::::R
EE:::::EEEEEEEEE:::E M::::::::M       M::::::::M R:::::RRRRRR:::::R
  E::::E       EEEEE M:::::::::M     M:::::::::M RR::::R      R::::R
  E::::E             M::::::M:::M   M:::M::::::M   R:::R      R::::R
  E:::::EEEEEEEEEE   M:::::M M:::M M:::M M:::::M   R:::RRRRRR:::::R
  E::::::::::::::E   M:::::M  M:::M:::M  M:::::M   R:::::::::::RR
  E:::::EEEEEEEEEE   M:::::M   M:::::M   M:::::M   R:::RRRRRR::::R
  E::::E             M:::::M    M:::M    M:::::M   R:::R      R::::R
  E::::E       EEEEE M:::::M     MMM     M:::::M   R:::R      R::::R
EE:::::EEEEEEEE::::E M:::::M             M:::::M   R:::R      R::::R
E::::::::::::::::::E M:::::M             M:::::M RR::::R      R::::R
EEEEEEEEEEEEEEEEEEEE MMMMMMM             MMMMMMM RRRRRRR      RRRRRR

[hadoop@ip-172-31-23-44 ~]$
```

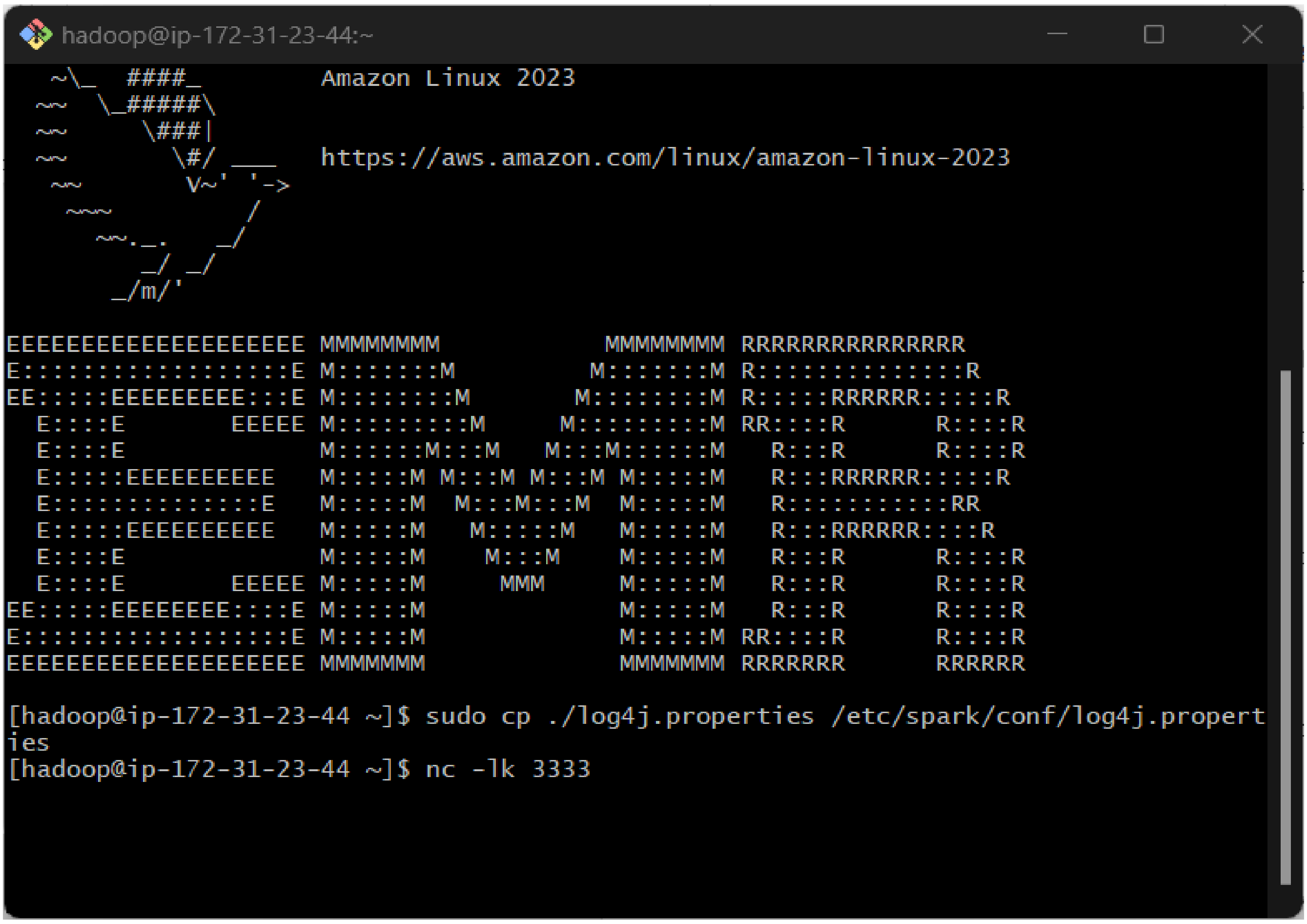
8) In the EC2-1 window enter the following:

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

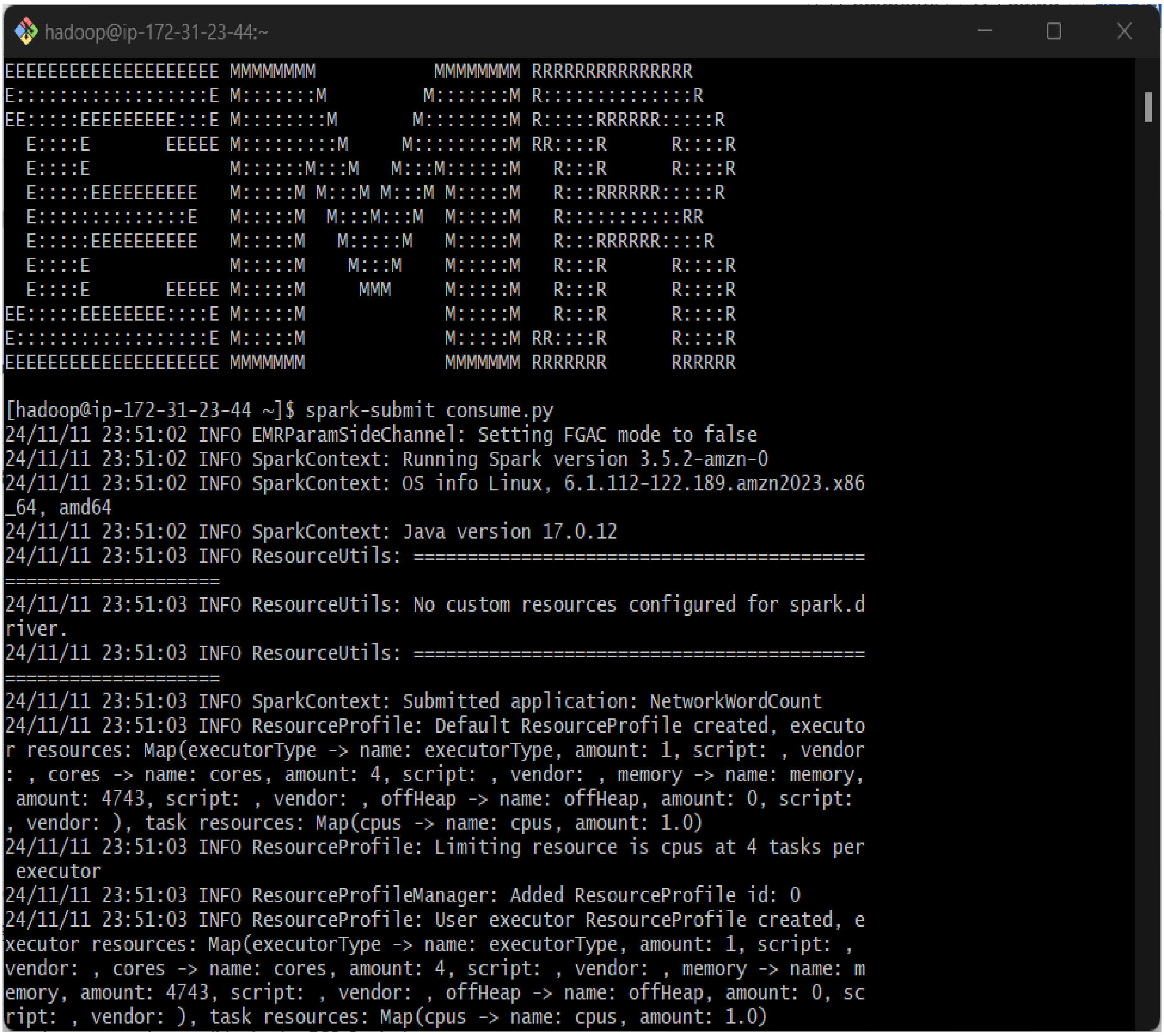


```
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'

EEEEEEEEEEEEEEEEEEEE MMMMMMMM           MMMMMMMM RRRRRRRRRRRRRRR
E::::::::::::::::::E M:::::::M         M:::::::M R::::::::::::::R
EE:::::EEEEEEEEE:::E M::::::::M       M::::::::M R:::::RRRRRR:::::R
  E::::E       EEEEE M:::::::::M     M:::::::::M RR::::R      R::::R
  E::::E             M::::::M:::M   M:::M::::::M   R:::R      R::::R
  E:::::EEEEEEEEEE   M:::::M M:::M M:::M M:::::M   R:::RRRRRR:::::R
  E::::::::::::::E   M:::::M  M:::M:::M  M:::::M   R:::::::::::RR
  E:::::EEEEEEEEEE   M:::::M   M:::::M   M:::::M   R:::RRRRRR::::R
  E::::E             M:::::M    M:::M    M:::::M   R:::R      R::::R
  E::::E       EEEEE M:::::M     MMM     M:::::M   R:::R      R::::R
EE:::::EEEEEEEE::::E M:::::M             M:::::M   R:::R      R::::R
E::::::::::::::::::E M:::::M             M:::::M RR::::R      R::::R
EEEEEEEEEEEEEEEEEEEE MMMMMMM             MMMMMMM RRRRRRR      RRRRRR

[hadoop@ip-172-31-23-44 ~]$ sudo cp ./log4j.properties /etc/spark/conf/log4j.propert
ies
[hadoop@ip-172-31-23-44 ~]$ nc -lk 3333
```

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

spark-submit consume.py

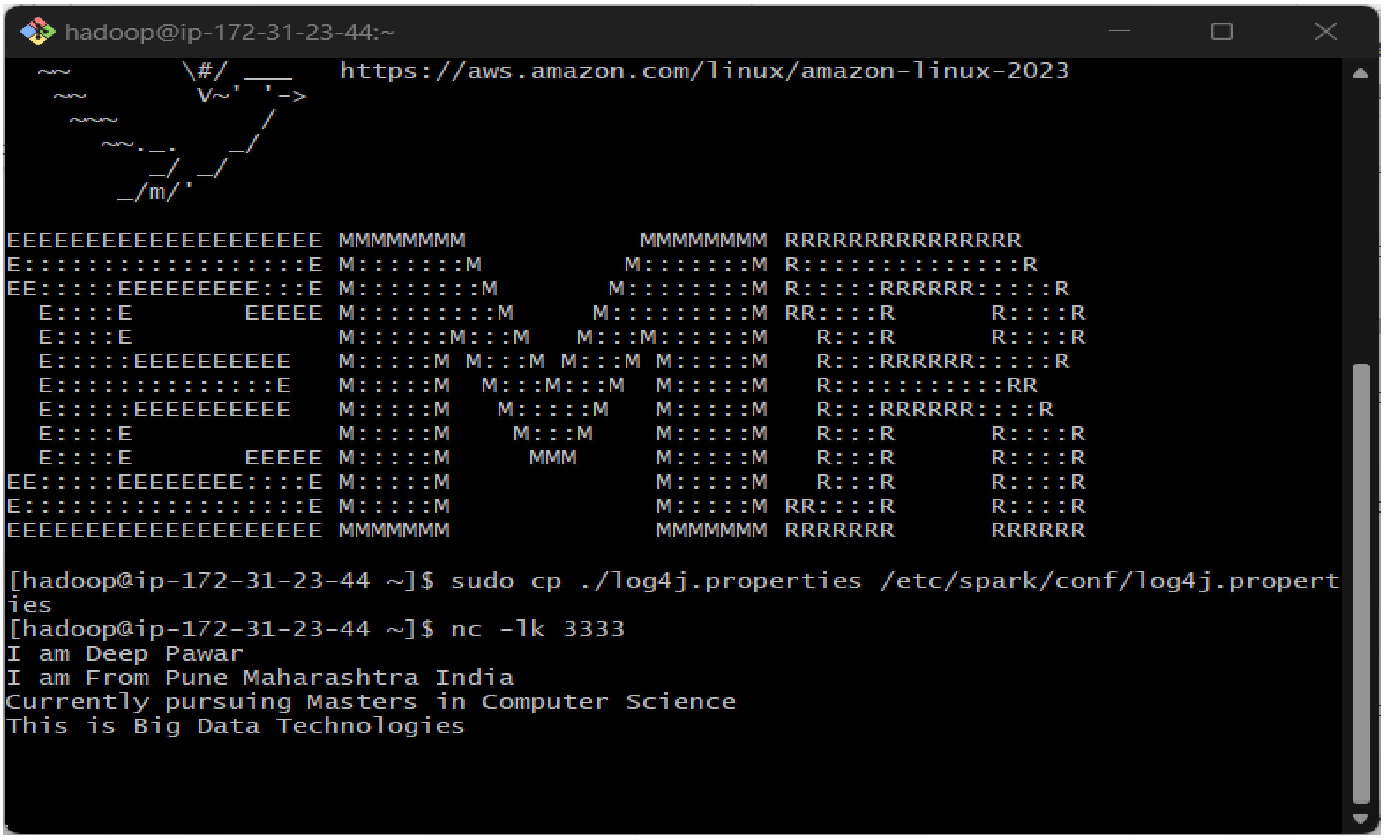
This takes a while to start up. So, wait for some messages issued to the console before continuing. Note, when you do this you might see a message beginning with “WARN StreamingContext:...” which you can ignore.



```
EEEEEEEEEEEEEEEEEEEE MMMMMMMM           MMMMMMMM RRRRRRRRRRRRRRR
E::::::::::::::::::E M:::::::M         M:::::::M R::::::::::::::R
EE:::::EEEEEEEEE:::E M::::::::M       M::::::::M R:::::RRRRRR:::::R
  E::::E       EEEEE M:::::::::M     M:::::::::M RR::::R      R::::R
  E::::E             M::::::M:::M   M:::M::::::M   R:::R      R::::R
  E:::::EEEEEEEEEE   M:::::M M:::M M:::M M:::::M   R:::RRRRRR:::::R
  E::::::::::::::E   M:::::M  M:::M:::M  M:::::M   R:::::::::::RR
  E:::::EEEEEEEEEE   M:::::M   M:::::M   M:::::M   R:::RRRRRR::::R
  E::::E             M:::::M    M:::M    M:::::M   R:::R      R::::R
  E::::E       EEEEE M:::::M     MMM     M:::::M   R:::R      R::::R
EE:::::EEEEEEEE::::E M:::::M             M:::::M   R:::R      R::::R
E::::::::::::::::::E M:::::M             M:::::M RR::::R      R::::R
EEEEEEEEEEEEEEEEEEEE MMMMMMM             MMMMMMM RRRRRRR      RRRRRR

[hadoop@ip-172-31-23-44 ~]$ spark-submit consume.py
24/11/11 23:51:02 INFO EMRParamSideChannel: Setting FGAC mode to false
24/11/11 23:51:02 INFO SparkContext: Running Spark version 3.5.2-amzn-0
24/11/11 23:51:02 INFO SparkContext: OS info Linux, 6.1.112-122.189.amzn2023.x86
_64, amd64
24/11/11 23:51:02 INFO SparkContext: Java version 17.0.12
24/11/11 23:51:03 INFO ResourceUtils: ==========================================
====================
24/11/11 23:51:03 INFO ResourceUtils: No custom resources configured for spark.d
river.
24/11/11 23:51:03 INFO ResourceUtils: ==========================================
====================
24/11/11 23:51:03 INFO SparkContext: Submitted application: NetworkWordCount
24/11/11 23:51:03 INFO ResourceProfile: Default ResourceProfile created, executo
r resources: Map(executorType -> name: executorType, amount: 1, script: , vendor
: , cores -> name: cores, amount: 4, script: , vendor: , memory -> name: memory,
 amount: 4743, script: , vendor: , offHeap -> name: offHeap, amount: 0, script:
, vendor: ), task resources: Map(cpus -> name: cpus, amount: 1.0)
24/11/11 23:51:03 INFO ResourceProfile: Limiting resource is cpus at 4 tasks per
 executor
24/11/11 23:51:03 INFO ResourceProfileManager: Added ResourceProfile id: 0
24/11/11 23:51:03 INFO ResourceProfile: User executor ResourceProfile created, e
xecutor resources: Map(executorType -> name: executorType, amount: 1, script: ,
vendor: , cores -> name: cores, amount: 4, script: , vendor: , memory -> name: m
emory, amount: 4743, script: , vendor: , offHeap -> name: offHeap, amount: 0, sc
ript: , vendor: ), task resources: Map(cpus -> name: cpus, amount: 1.0)
```

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

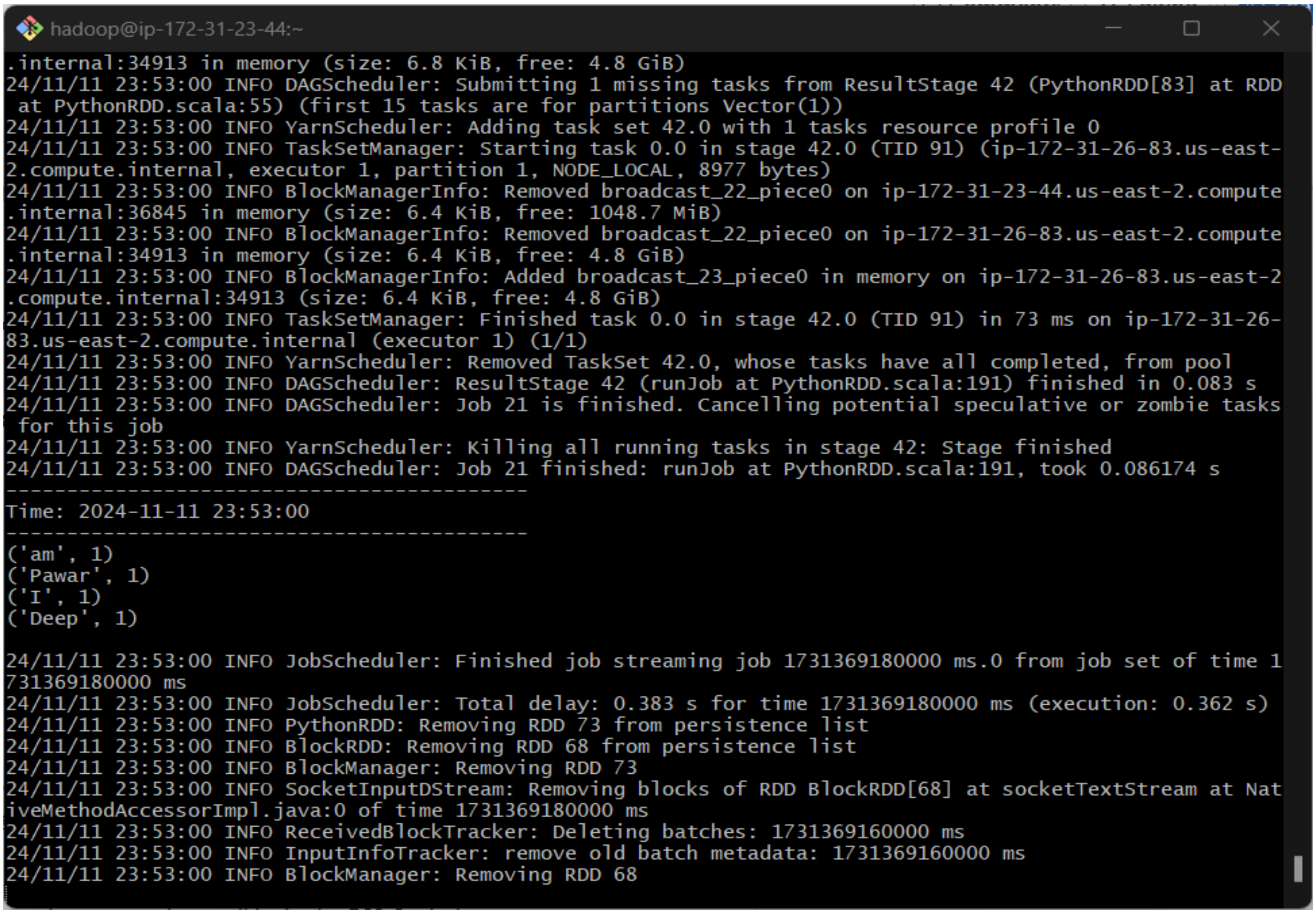
- **EC2-1 window**



hadoop@ip-172-31-23-44:~

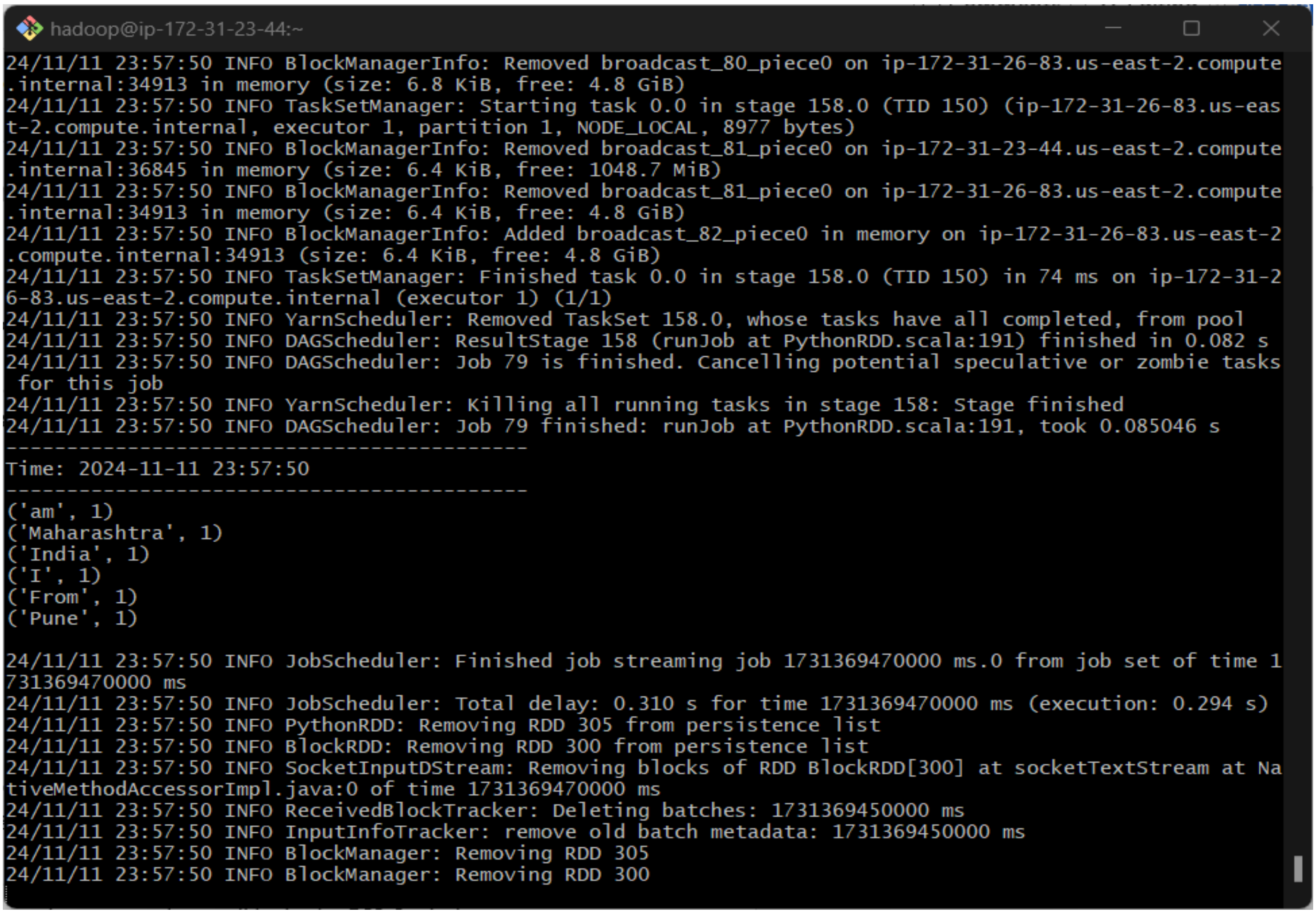
```
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'

EEEEEEEEEEEEEEEEEEEE MMMMMMMM           MMMMMMMM RRRRRRRRRRRRRRR
E::::::::::::::::::E M:::::::M         M:::::::M R::::::::::::::R
EE:::::EEEEEEEEE:::E M::::::::M       M::::::::M R:::::RRRRRR:::::R
  E::::E       EEEEE M:::::::::M     M:::::::::M RR::::R      R::::R
  E::::E             M::::::M:::M   M:::M::::::M   R:::R      R::::R
  E:::::EEEEEEEEEE   M:::::M M:::M M:::M M:::::M   R:::RRRRRR:::::R
  E::::::::::::::E   M:::::M  M:::M:::M  M:::::M   R:::::::::::RR
  E:::::EEEEEEEEEE   M:::::M   M:::::M   M:::::M   R:::RRRRRR::::R
  E::::E             M:::::M    M:::M    M:::::M   R:::R      R::::R
  E::::E       EEEEE M:::::M     MMM     M:::::M   R:::R      R::::R
EE:::::EEEEEEEE::::E M:::::M             M:::::M   R:::R      R::::R
E::::::::::::::::::E M:::::M             M:::::M RR::::R      R::::R
EEEEEEEEEEEEEEEEEEEE MMMMMMM             MMMMMMM RRRRRRR      RRRRRR

[hadoop@ip-172-31-23-44 ~]$ sudo cp ./log4j.properties /etc/spark/conf/log4j.propert
ies
[hadoop@ip-172-31-23-44 ~]$ nc -lk 3333
I am Deep Pawar
I am From Pune Maharashtra India
Currently pursuing Masters in Computer Science
This is Big Data Technologies
```

- **EC2-2 window**

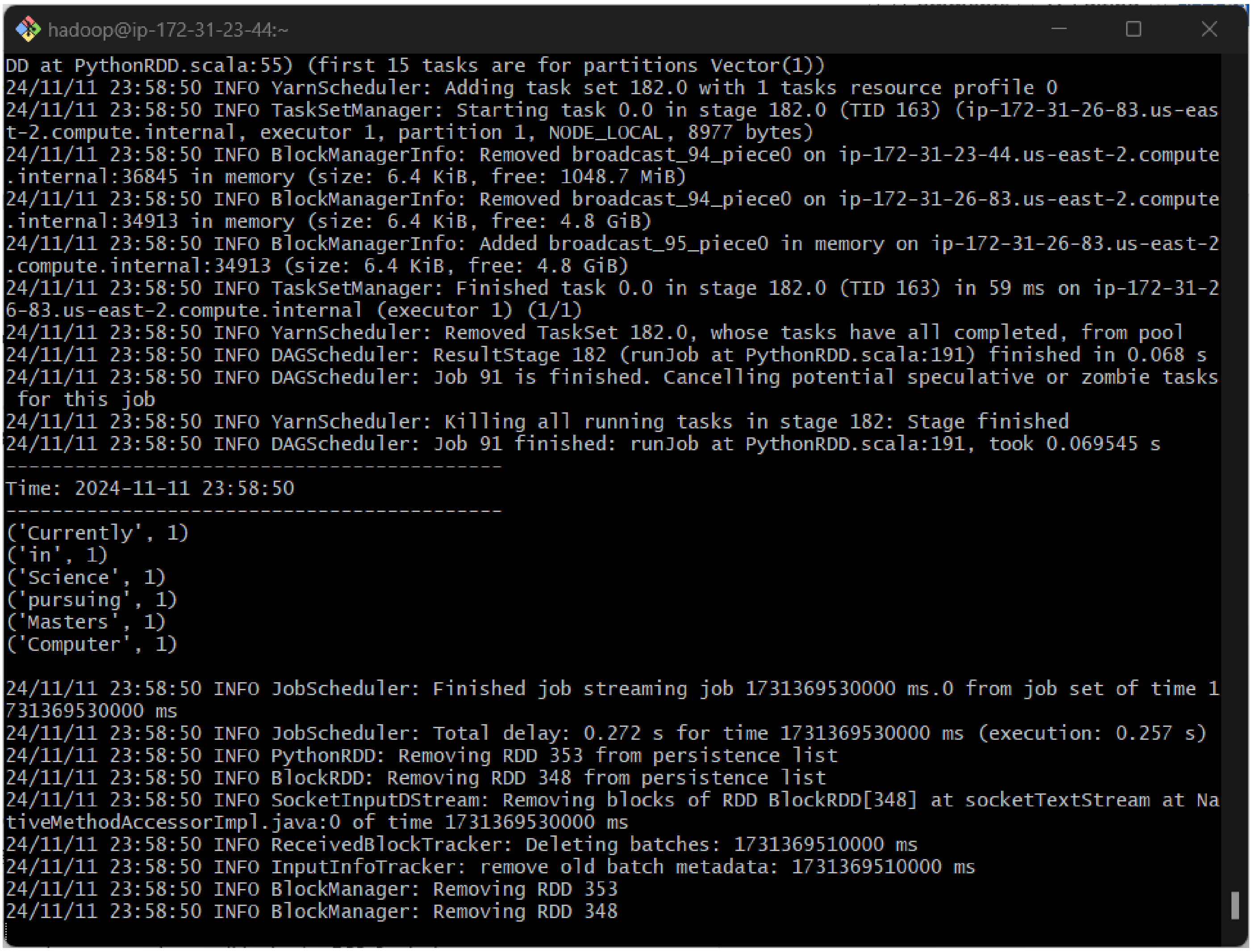


hadoop@ip-172-31-23-44:~

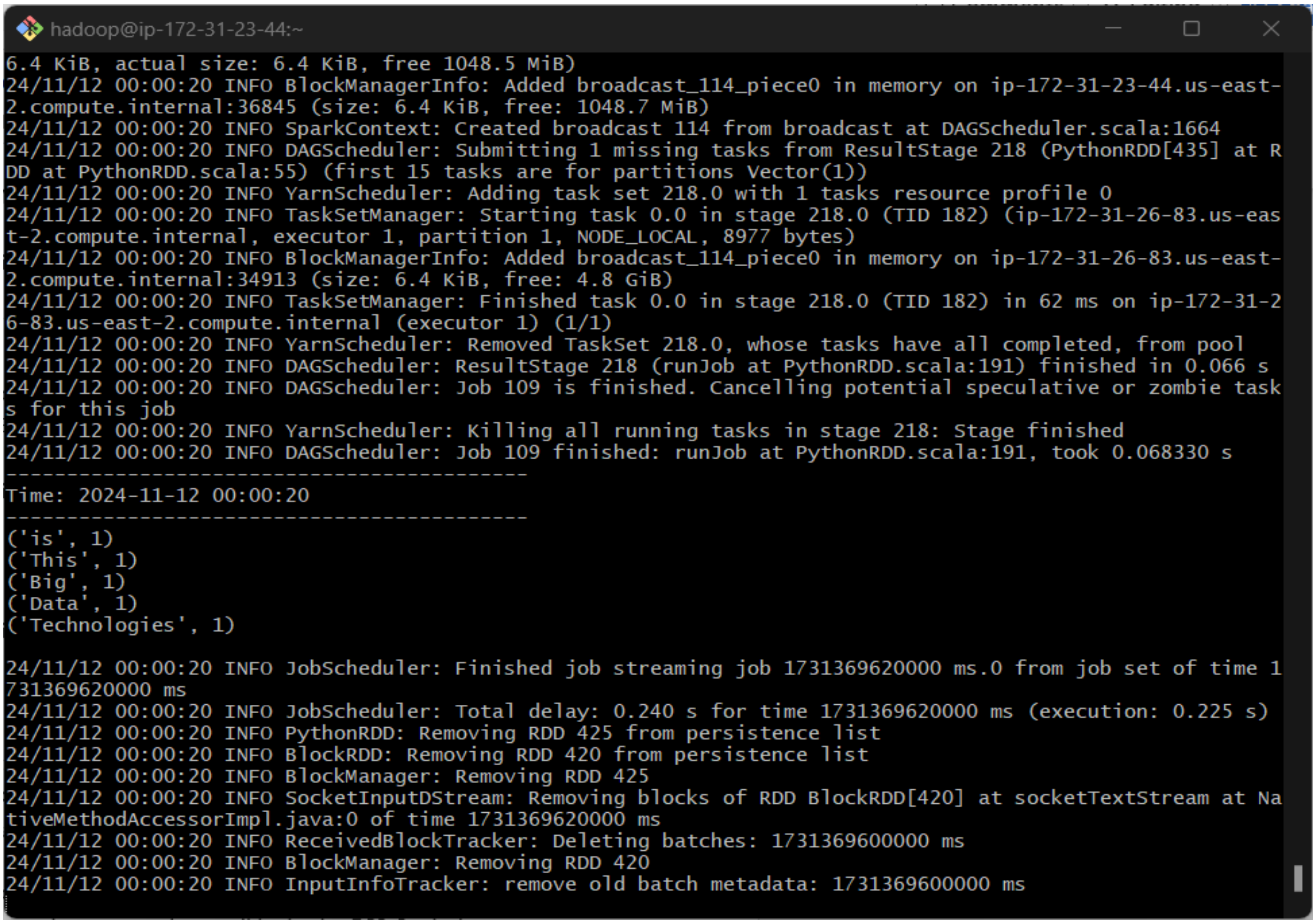
```
.internal:34913 in memory (size: 6.8 KiB, free: 4.8 GiB)
24/11/11 23:53:00 INFO DAGScheduler: Submitting 1 missing tasks from ResultStage 42 (PythonRDD[83] at RDD
 at PythonRDD.scala:55) (first 15 tasks are for partitions Vector(1))
24/11/11 23:53:00 INFO YarnScheduler: Adding task set 42.0 with 1 tasks resource profile 0
24/11/11 23:53:00 INFO TaskSetManager: Starting task 0.0 in stage 42.0 (TID 91) (ip-172-31-26-83.us-east-
2.compute.internal, executor 1, partition 1, NODE_LOCAL, 8977 bytes)
24/11/11 23:53:00 INFO BlockManagerInfo: Removed broadcast_22_piece0 on ip-172-31-23-44.us-east-2.compute
.internal:36845 in memory (size: 6.4 KiB, free: 1048.7 MiB)
24/11/11 23:53:00 INFO BlockManagerInfo: Removed broadcast_22_piece0 on ip-172-31-26-83.us-east-2.compute
.internal:34913 in memory (size: 6.4 KiB, free: 4.8 GiB)
24/11/11 23:53:00 INFO BlockManagerInfo: Added broadcast_23_piece0 in memory on ip-172-31-26-83.us-east-2
.compute.internal:34913 (size: 6.4 KiB, free: 4.8 GiB)
24/11/11 23:53:00 INFO TaskSetManager: Finished task 0.0 in stage 42.0 (TID 91) in 73 ms on ip-172-31-26-
83.us-east-2.compute.internal (executor 1) (1/1)
24/11/11 23:53:00 INFO YarnScheduler: Removed TaskSet 42.0, whose tasks have all completed, from pool
24/11/11 23:53:00 INFO DAGScheduler: ResultStage 42 (runJob at PythonRDD.scala:191) finished in 0.083 s
24/11/11 23:53:00 INFO DAGScheduler: Job 21 is finished. Cancelling potential speculative or zombie tasks
 for this job
24/11/11 23:53:00 INFO YarnScheduler: Killing all running tasks in stage 42: Stage finished
24/11/11 23:53:00 INFO DAGScheduler: Job 21 finished: runJob at PythonRDD.scala:191, took 0.086174 s
-------------------------------------------
Time: 2024-11-11 23:53:00
-------------------------------------------
('am', 1)
('Pawar', 1)
('I', 1)
('Deep', 1)

24/11/11 23:53:00 INFO JobScheduler: Finished job streaming job 1731369180000 ms.0 from job set of time 1
731369180000 ms
24/11/11 23:53:00 INFO JobScheduler: Total delay: 0.383 s for time 1731369180000 ms (execution: 0.362 s)
24/11/11 23:53:00 INFO PythonRDD: Removing RDD 73 from persistence list
24/11/11 23:53:00 INFO BlockRDD: Removing RDD 68 from persistence list
24/11/11 23:53:00 INFO BlockManager: Removing RDD 73
24/11/11 23:53:00 INFO SocketInputDStream: Removing blocks of RDD BlockRDD[68] at socketTextStream at Nat
iveMethodAccessorImpl.java:0 of time 1731369180000 ms
24/11/11 23:53:00 INFO ReceivedBlockTracker: Deleting batches: 1731369160000 ms
24/11/11 23:53:00 INFO InputInfoTracker: remove old batch metadata: 1731369160000 ms
24/11/11 23:53:00 INFO BlockManager: Removing RDD 68
```



```
24/11/11 23:57:50 INFO BlockManagerInfo: Removed broadcast_80_piece0 on ip-172-31-26-83.us-east-2.compute
.internal:34913 in memory (size: 6.8 KiB, free: 4.8 GiB)
24/11/11 23:57:50 INFO TaskSetManager: Starting task 0.0 in stage 158.0 (TID 150) (ip-172-31-26-83.us-eas
t-2.compute.internal, executor 1, partition 1, NODE_LOCAL, 8977 bytes)
24/11/11 23:57:50 INFO BlockManagerInfo: Removed broadcast_81_piece0 on ip-172-31-23-44.us-east-2.compute
.internal:36845 in memory (size: 6.4 KiB, free: 1048.7 MiB)
24/11/11 23:57:50 INFO BlockManagerInfo: Removed broadcast_81_piece0 on ip-172-31-26-83.us-east-2.compute
.internal:34913 in memory (size: 6.4 KiB, free: 4.8 GiB)
24/11/11 23:57:50 INFO BlockManagerInfo: Added broadcast_82_piece0 in memory on ip-172-31-26-83.us-east-2
.compute.internal:34913 (size: 6.4 KiB, free: 4.8 GiB)
24/11/11 23:57:50 INFO TaskSetManager: Finished task 0.0 in stage 158.0 (TID 150) in 74 ms on ip-172-31-2
6-83.us-east-2.compute.internal (executor 1) (1/1)
24/11/11 23:57:50 INFO YarnScheduler: Removed TaskSet 158.0, whose tasks have all completed, from pool
24/11/11 23:57:50 INFO DAGScheduler: ResultStage 158 (runJob at PythonRDD.scala:191) finished in 0.082 s
24/11/11 23:57:50 INFO DAGScheduler: Job 79 is finished. Cancelling potential speculative or zombie tasks
 for this job
24/11/11 23:57:50 INFO YarnScheduler: Killing all running tasks in stage 158: Stage finished
24/11/11 23:57:50 INFO DAGScheduler: Job 79 finished: runJob at PythonRDD.scala:191, took 0.085046 s
-------------------------------------------
Time: 2024-11-11 23:57:50
-------------------------------------------
('am', 1)
('Maharashtra', 1)
('India', 1)
('I', 1)
('From', 1)
('Pune', 1)

24/11/11 23:57:50 INFO JobScheduler: Finished job streaming job 1731369470000 ms.0 from job set of time 1
731369470000 ms
24/11/11 23:57:50 INFO JobScheduler: Total delay: 0.310 s for time 1731369470000 ms (execution: 0.294 s)
24/11/11 23:57:50 INFO PythonRDD: Removing RDD 305 from persistence list
24/11/11 23:57:50 INFO BlockRDD: Removing RDD 300 from persistence list
24/11/11 23:57:50 INFO SocketInputDStream: Removing blocks of RDD BlockRDD[300] at socketTextStream at Na
tiveMethodAccessorImpl.java:0 of time 1731369470000 ms
24/11/11 23:57:50 INFO ReceivedBlockTracker: Deleting batches: 1731369450000 ms
24/11/11 23:57:50 INFO InputInfoTracker: remove old batch metadata: 1731369450000 ms
24/11/11 23:57:50 INFO BlockManager: Removing RDD 305
24/11/11 23:57:50 INFO BlockManager: Removing RDD 300
```

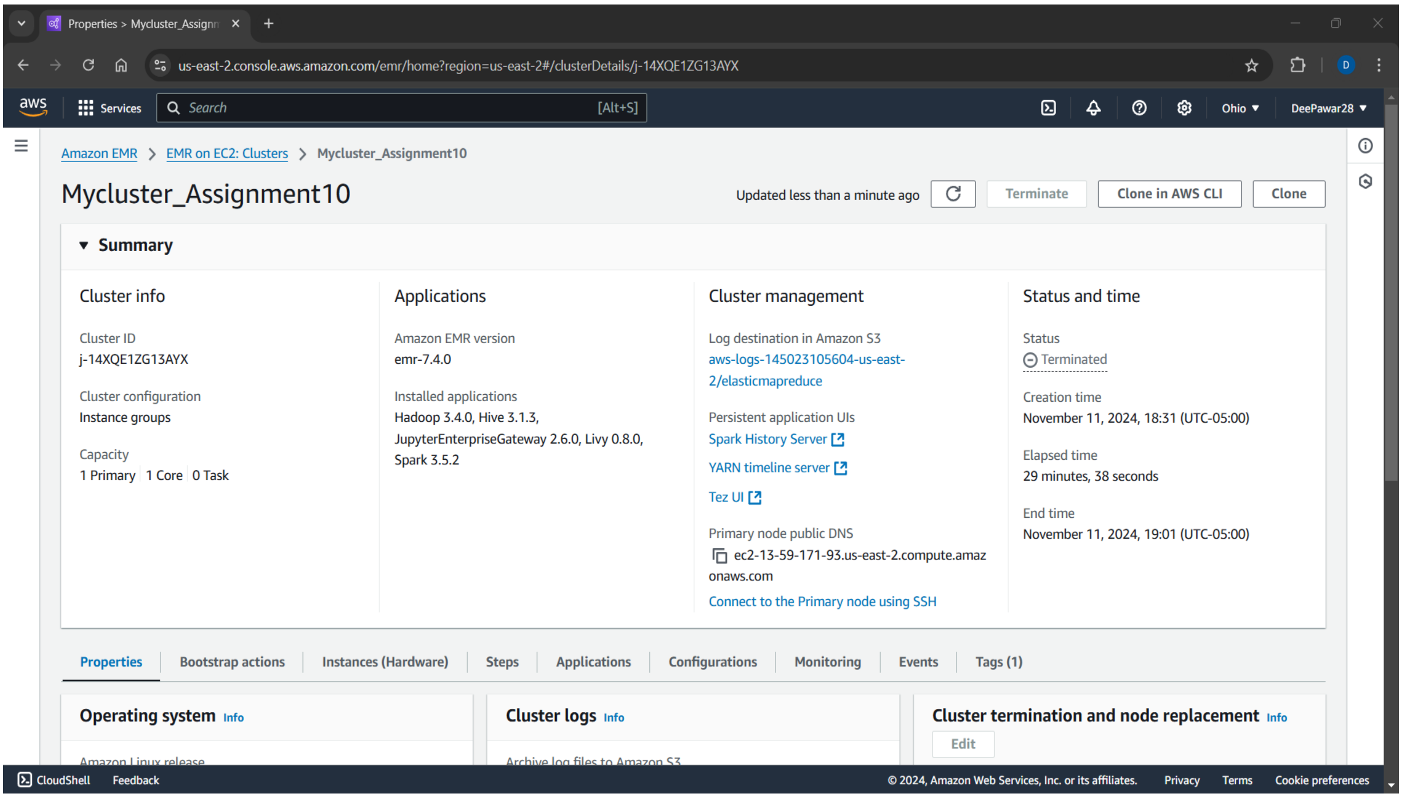


```
DD at PythonRDD.scala:55) (first 15 tasks are for partitions Vector(1))
24/11/11 23:58:50 INFO YarnScheduler: Adding task set 182.0 with 1 tasks resource profile 0
24/11/11 23:58:50 INFO TaskSetManager: Starting task 0.0 in stage 182.0 (TID 163) (ip-172-31-26-83.us-eas
t-2.compute.internal, executor 1, partition 1, NODE_LOCAL, 8977 bytes)
24/11/11 23:58:50 INFO BlockManagerInfo: Removed broadcast_94_piece0 on ip-172-31-23-44.us-east-2.compute
.internal:36845 in memory (size: 6.4 KiB, free: 1048.7 MiB)
24/11/11 23:58:50 INFO BlockManagerInfo: Removed broadcast_94_piece0 on ip-172-31-26-83.us-east-2.compute
.internal:34913 in memory (size: 6.4 KiB, free: 4.8 GiB)
24/11/11 23:58:50 INFO BlockManagerInfo: Added broadcast_95_piece0 in memory on ip-172-31-26-83.us-east-2
.compute.internal:34913 (size: 6.4 KiB, free: 4.8 GiB)
24/11/11 23:58:50 INFO TaskSetManager: Finished task 0.0 in stage 182.0 (TID 163) in 59 ms on ip-172-31-2
6-83.us-east-2.compute.internal (executor 1) (1/1)
24/11/11 23:58:50 INFO YarnScheduler: Removed TaskSet 182.0, whose tasks have all completed, from pool
24/11/11 23:58:50 INFO DAGScheduler: ResultStage 182 (runJob at PythonRDD.scala:191) finished in 0.068 s
24/11/11 23:58:50 INFO DAGScheduler: Job 91 is finished. Cancelling potential speculative or zombie tasks
 for this job
24/11/11 23:58:50 INFO YarnScheduler: Killing all running tasks in stage 182: Stage finished
24/11/11 23:58:50 INFO DAGScheduler: Job 91 finished: runJob at PythonRDD.scala:191, took 0.069545 s
-------------------------------------------
Time: 2024-11-11 23:58:50
-------------------------------------------
('Currently', 1)
('in', 1)
('Science', 1)
('pursuing', 1)
('Masters', 1)
('Computer', 1)

24/11/11 23:58:50 INFO JobScheduler: Finished job streaming job 1731369530000 ms.0 from job set of time 1
731369530000 ms
24/11/11 23:58:50 INFO JobScheduler: Total delay: 0.272 s for time 1731369530000 ms (execution: 0.257 s)
24/11/11 23:58:50 INFO PythonRDD: Removing RDD 353 from persistence list
24/11/11 23:58:50 INFO BlockRDD: Removing RDD 348 from persistence list
24/11/11 23:58:50 INFO SocketInputDStream: Removing blocks of RDD BlockRDD[348] at socketTextStream at Na
tiveMethodAccessorImpl.java:0 of time 1731369530000 ms
24/11/11 23:58:50 INFO ReceivedBlockTracker: Deleting batches: 1731369510000 ms
24/11/11 23:58:50 INFO InputInfoTracker: remove old batch metadata: 1731369510000 ms
24/11/11 23:58:50 INFO BlockManager: Removing RDD 353
24/11/11 23:58:50 INFO BlockManager: Removing RDD 348
```



```
6.4 KiB, actual size: 6.4 KiB, free 1048.5 MiB)
24/11/12 00:00:20 INFO BlockManagerInfo: Added broadcast_114_piece0 in memory on ip-172-31-23-44.us-east-
2.compute.internal:36845 (size: 6.4 KiB, free: 1048.7 MiB)
24/11/12 00:00:20 INFO SparkContext: Created broadcast 114 from broadcast at DAGScheduler.scala:1664
24/11/12 00:00:20 INFO DAGScheduler: Submitting 1 missing tasks from ResultStage 218 (PythonRDD[435] at R
DD at PythonRDD.scala:55) (first 15 tasks are for partitions Vector(1))
24/11/12 00:00:20 INFO YarnScheduler: Adding task set 218.0 with 1 tasks resource profile 0
24/11/12 00:00:20 INFO TaskSetManager: Starting task 0.0 in stage 218.0 (TID 182) (ip-172-31-26-83.us-eas
t-2.compute.internal, executor 1, partition 1, NODE_LOCAL, 8977 bytes)
24/11/12 00:00:20 INFO BlockManagerInfo: Added broadcast_114_piece0 in memory on ip-172-31-26-83.us-east-
2.compute.internal:34913 (size: 6.4 KiB, free: 4.8 GiB)
24/11/12 00:00:20 INFO TaskSetManager: Finished task 0.0 in stage 218.0 (TID 182) in 62 ms on ip-172-31-2
6-83.us-east-2.compute.internal (executor 1) (1/1)
24/11/12 00:00:20 INFO YarnScheduler: Removed TaskSet 218.0, whose tasks have all completed, from pool
24/11/12 00:00:20 INFO DAGScheduler: ResultStage 218 (runJob at PythonRDD.scala:191) finished in 0.066 s
24/11/12 00:00:20 INFO DAGScheduler: Job 109 is finished. Cancelling potential speculative or zombie task
s for this job
24/11/12 00:00:20 INFO YarnScheduler: Killing all running tasks in stage 218: Stage finished
24/11/12 00:00:20 INFO DAGScheduler: Job 109 finished: runJob at PythonRDD.scala:191, took 0.068330 s
-------------------------------------------
Time: 2024-11-12 00:00:20
-------------------------------------------
('is', 1)
('This', 1)
('Big', 1)
('Data', 1)
('Technologies', 1)

24/11/12 00:00:20 INFO JobScheduler: Finished job streaming job 1731369620000 ms.0 from job set of time 1
731369620000 ms
24/11/12 00:00:20 INFO JobScheduler: Total delay: 0.240 s for time 1731369620000 ms (execution: 0.225 s)
24/11/12 00:00:20 INFO PythonRDD: Removing RDD 425 from persistence list
24/11/12 00:00:20 INFO BlockRDD: Removing RDD 420 from persistence list
24/11/12 00:00:20 INFO BlockManager: Removing RDD 425
24/11/12 00:00:20 INFO SocketInputDStream: Removing blocks of RDD BlockRDD[420] at socketTextStream at Na
tiveMethodAccessorImpl.java:0 of time 1731369620000 ms
24/11/12 00:00:20 INFO ReceivedBlockTracker: Deleting batches: 1731369600000 ms
24/11/12 00:00:20 INFO BlockManager: Removing RDD 420
24/11/12 00:00:20 INFO InputInfoTracker: remove old batch metadata: 1731369600000 ms
```

12) Remember to terminate your EMR instance after you are done!


Properties > Mycluster_Assignm
us-east-2.console.aws.amazon.com/emr/home?region=us-east-2#/clusterDetails/j-14XQE1ZG13AYX
aws
Services
Search
[Alt+S]
Ohio ▾
DeePawar28 ▾
Amazon EMR > EMR on EC2: Clusters > Mycluster_Assignment10
Updated less than a minute ago
Terminate
Clone in AWS CLI
Clone
Properties
Bootstrap actions
Instances (Hardware)
Steps
Applications
Configurations
Monitoring
Events
Tags (1)
Edit
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Mycluster_Assignment10

▾ Summary

Cluster info

Cluster ID
j-14XQE1ZG13AYX

Cluster configuration
Instance groups

Capacity
1 Primary | 1 Core | 0 Task

Applications

Amazon EMR version
emr-7.4.0

Installed applications
Hadoop 3.4.0, Hive 3.1.3, JupyterEnterpriseGateway 2.6.0, Livy 0.8.0, Spark 3.5.2

Cluster management

Log destination in Amazon S3
aws-logs-145023105604-us-east-2/elasticmapreduce

Persistent application UIs
Spark History Server
YARN timeline server
Tez UI

Primary node public DNS
ec2-13-59-171-93.us-east-2.compute.amazonaws.com

Connect to the Primary node using SSH

Status and time

Status
Terminated

Creation time
November 11, 2024, 18:31 (UTC-05:00)

Elapsed time
29 minutes, 38 seconds

End time
November 11, 2024, 19:01 (UTC-05:00)

Operating system Info

Amazon Linux release

Cluster logs Info

Archive log files to Amazon S3

Cluster termination and node replacement Info