BÁO CÁO LAB4 LƯU TRỮ XỬ LÝ DỮ LIỆU LỚN

Nhóm: Squad Game

Thành viên: Lại Ngọc Thăng Long 20183581

Nguyễn Đình Dũng 20183506

Nguyễn Thành Long 20183586

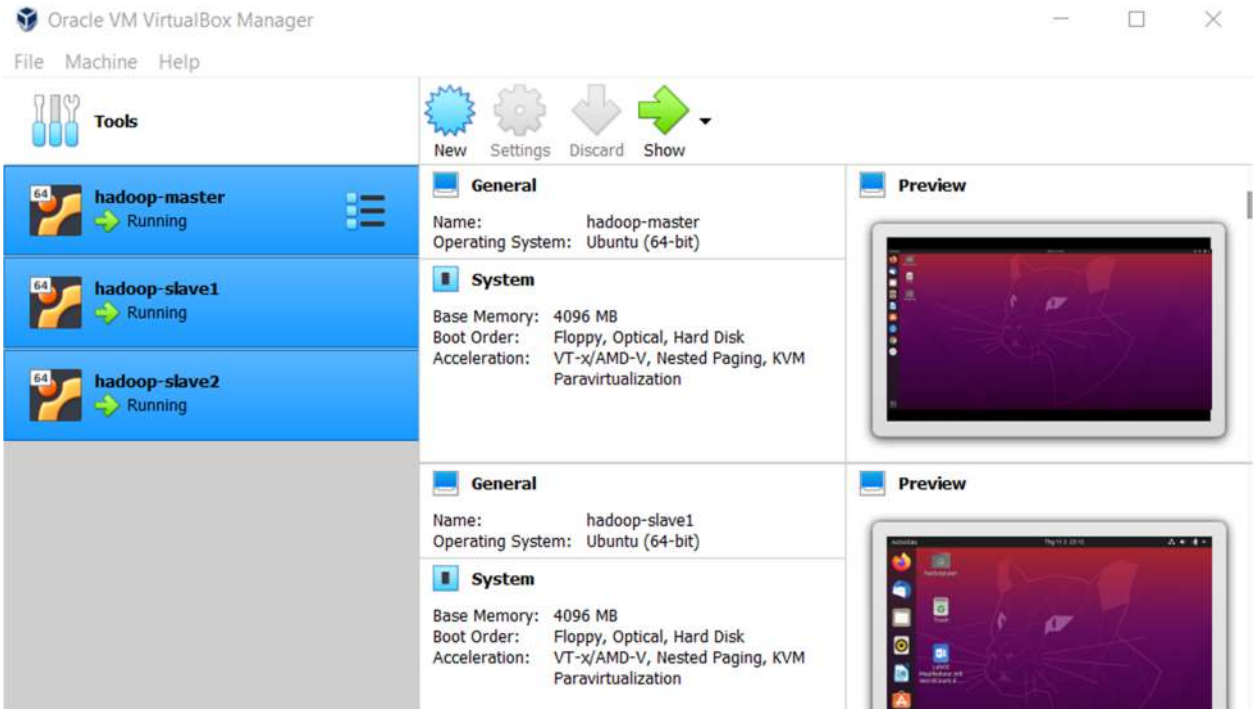
Nguyễn Khương Duy 20183513

BÀI LÀM:

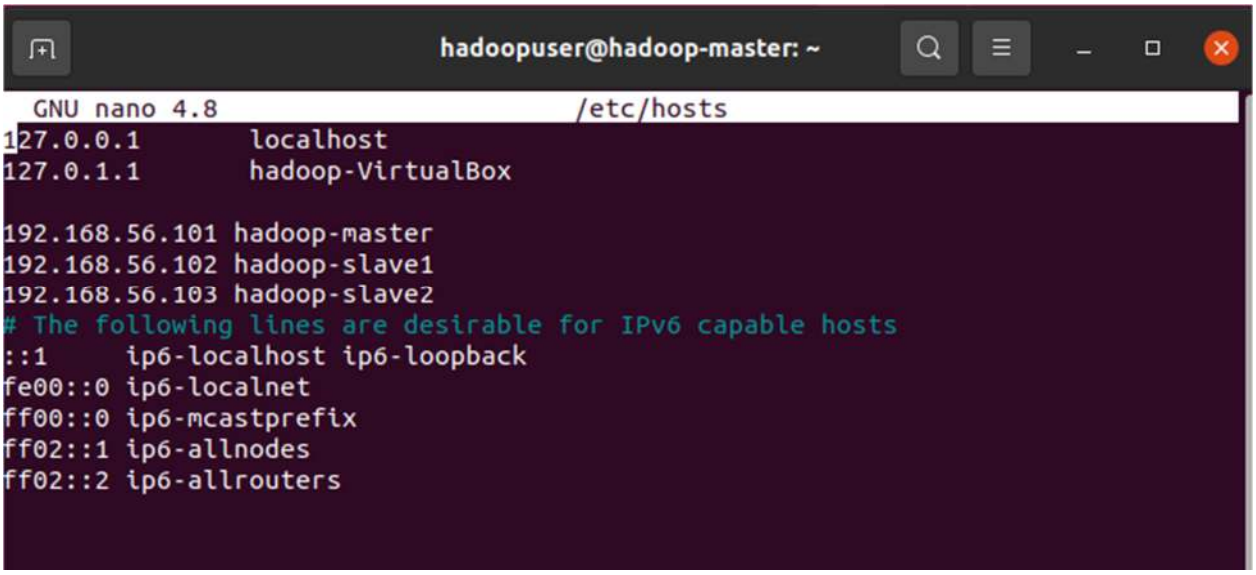
I. Cài đặt Spark Cluster

1st Step: Cài đặt Spark Cluster trên 3 máy

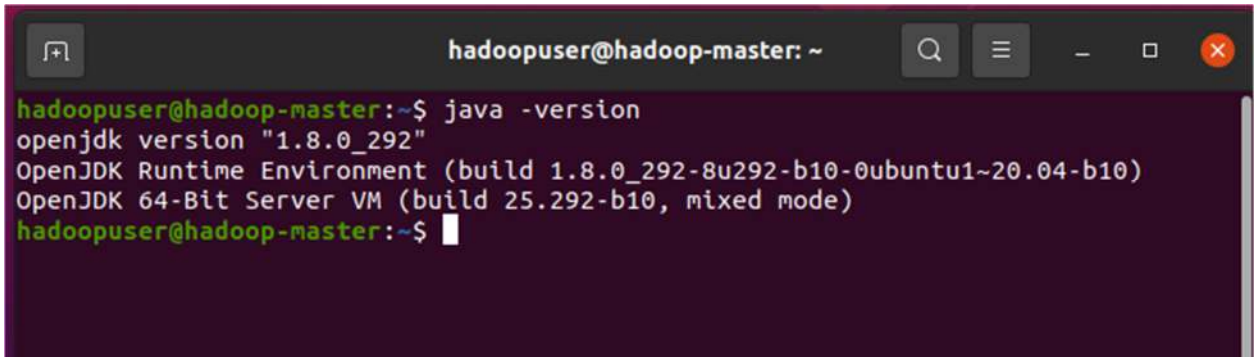
Step 1 — 3 máy Master, Slave1, Slave2



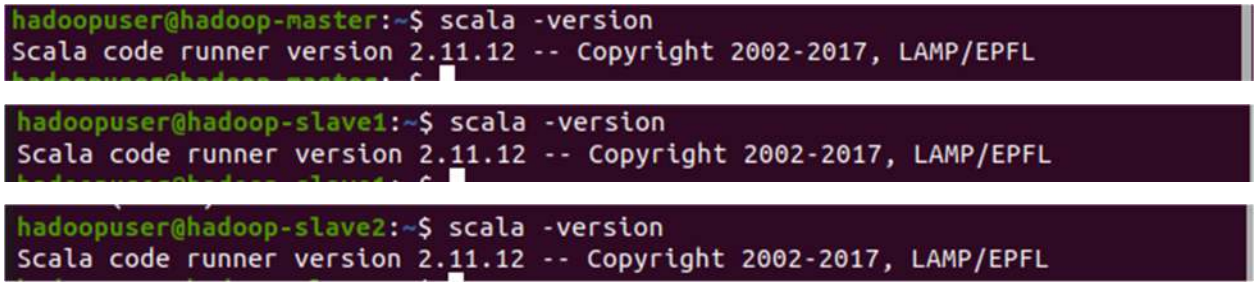
Step 2 — Cập nhật file hosts



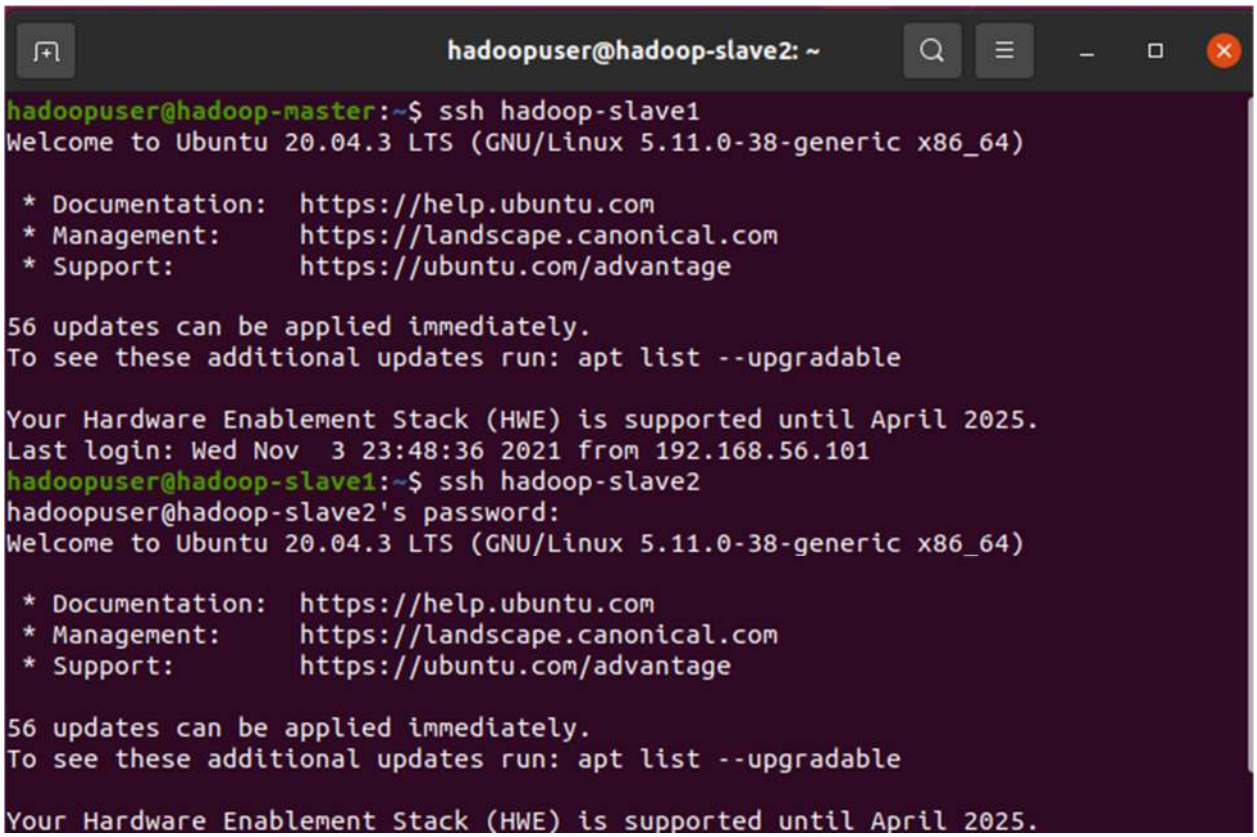
Step 3 — Cài đặt java



Step 4 — Cài đặt Scala (trên cả 3 máy)



Step 5 — Tạo key và gửi cho các máy master, slave1, slave2(để start, truy cập, tương tác được máy worker từ máy master). Check SSH



Step 6 — Download và cài Spark (trên cả 3 máy)

1, Download bằng lệnh sau:

wget http://archive.apache.org/dist/spark/spark-2.4.4/spark-2.4.4-bin-hadoop2.7.tgz

2, Giải nén bằng lệnh:

$ tar xvf spark-2.4.4-bin-hadoop2.7.tgz

3, Move Spark vào thư mục tương ứng:

4, Setup env cho Apache Spark: sửa file bashrc bằng lệnh:

```
$ sudo nano .bashrc
```

5, Thêm dòng này vào cuối tệp

```
export PATH=$PATH:/usr/local/spark/bin
```

6, Save lại, dùng lệnh sau để cập nhật biến môi trường cho terminal:

```
$ source ~/.bashrc
```

Step 7 — Configuration Apache Spark (Chỉ làm trên máy Master)

Chỉnh sửa spark-env.sh

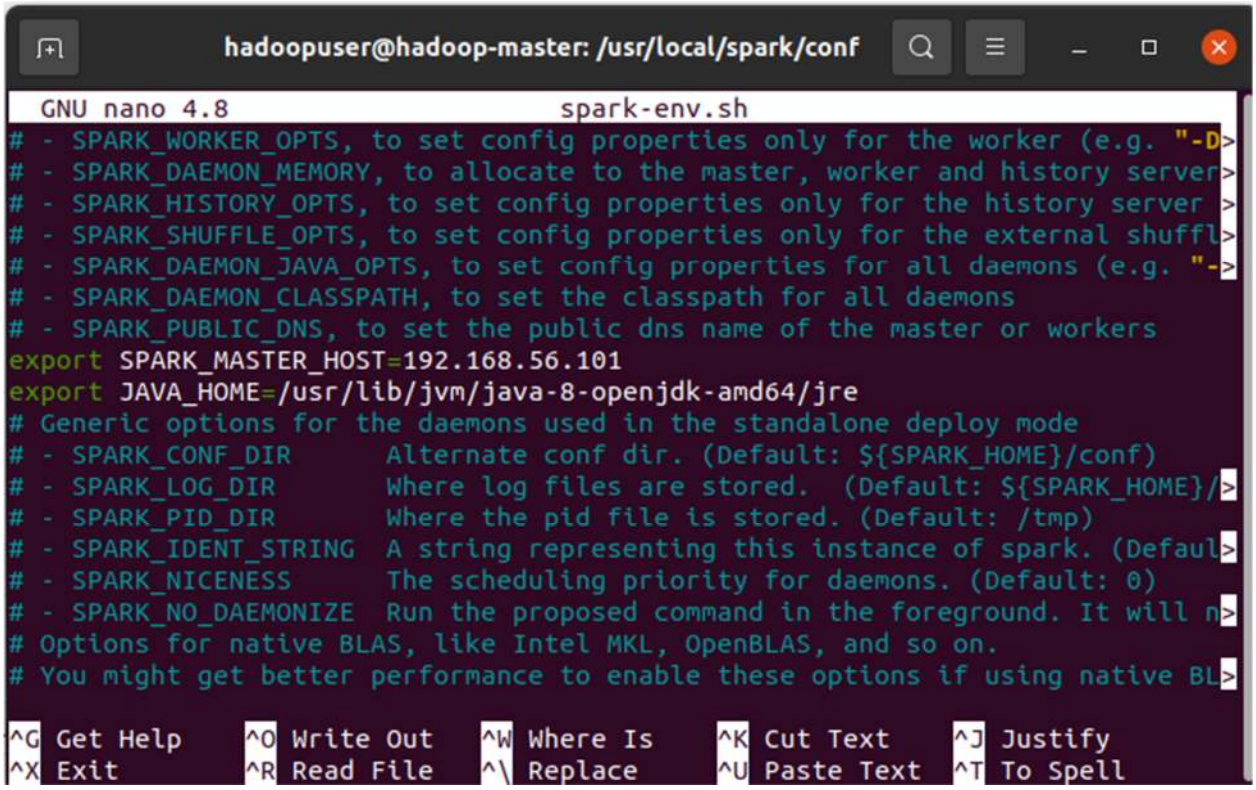
Di chuyển đến thư mục spark conf và tạo một bản sao của spark-env.sh và đổi tên nó:

```
$ cd /usr/local/spark/conf
$ cp spark-env.sh.template spark-env.sh
```

Tiếp theo edit config file spark-env.sh:

```
$ sudo nano spark-env.sh
```

Thêm dòng sau vào:

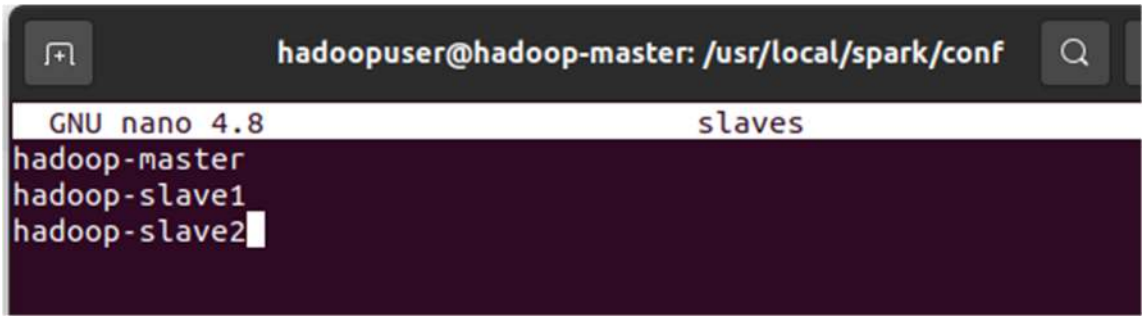


Add Workers

Sửa file slaves:

```
$ sudo nano slaves
```

Thêm các dòng sau:



Step 8 — Start Spark Cluster (Chỉ làm trên máy Master)

Sử dụng lệnh sau để start:

```
$ cd /usr/local/spark
$ ./sbin/start-all.sh
```

Để stop cluster dùng lệnh:

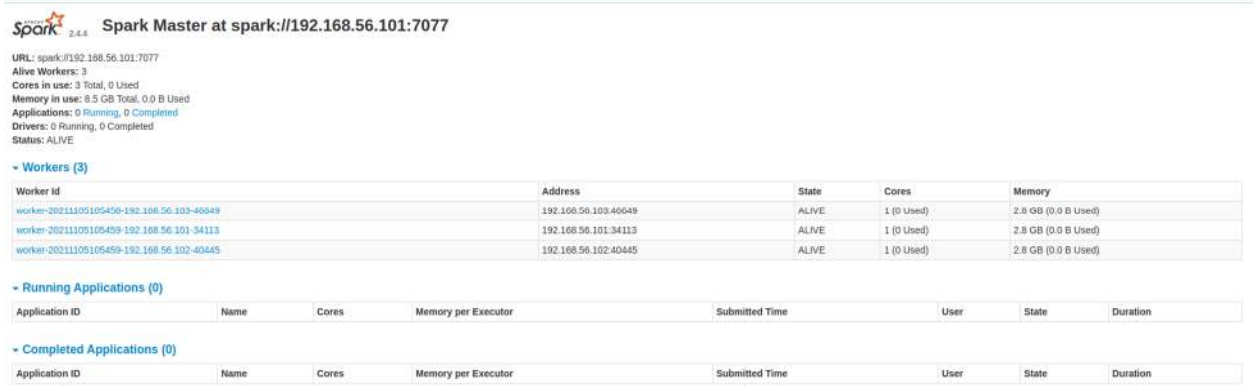
```
$ ./sbin/stop-all.sh
```

Check service dùng lệnh:

```
jps
```

Xem trên UI vào địa chỉ:

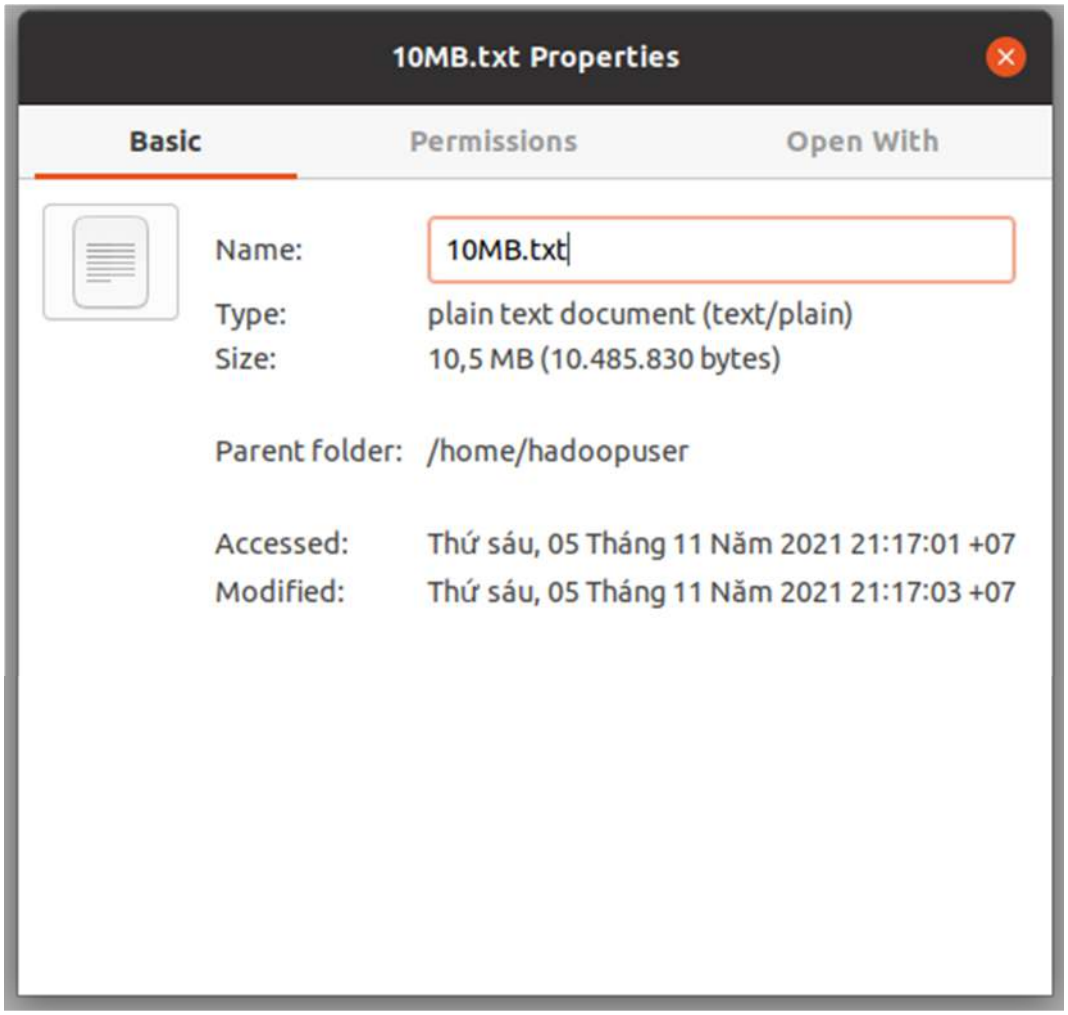
```
192.168.56.101:8080
```



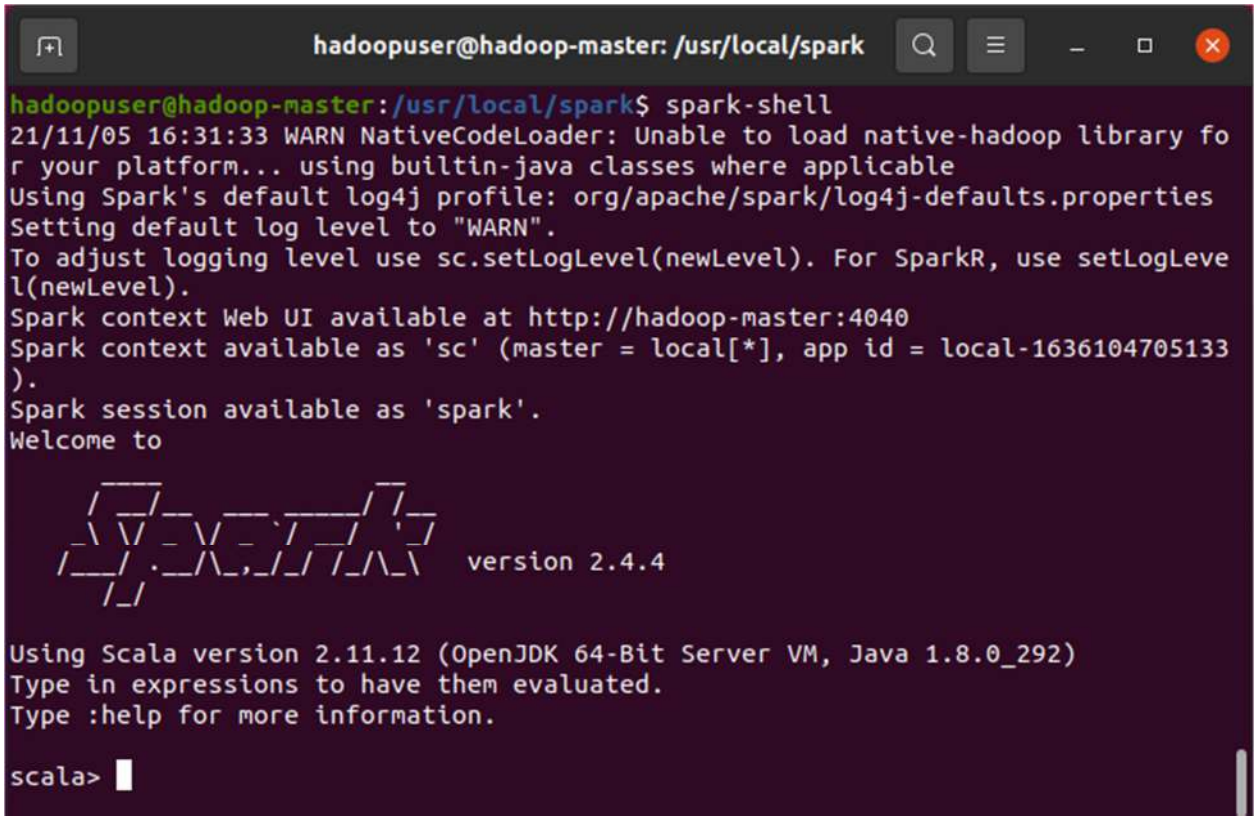
Ta thấy Alive Workers: 3 (Lý do Alive Workers = 3 là do dùng start-all.sh có nghĩa là tính luôn cả Worker trên máy 192.168.56.101)

II, Chạy chương trình WordCount

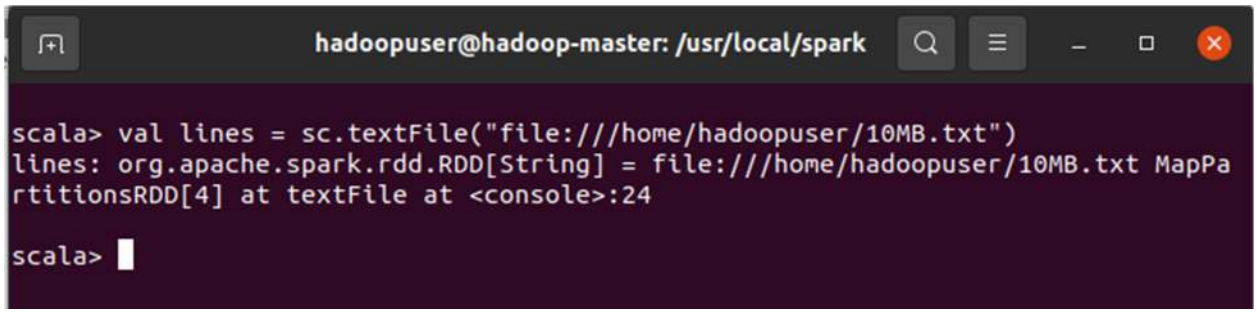
Step 1: Chuẩn bị data



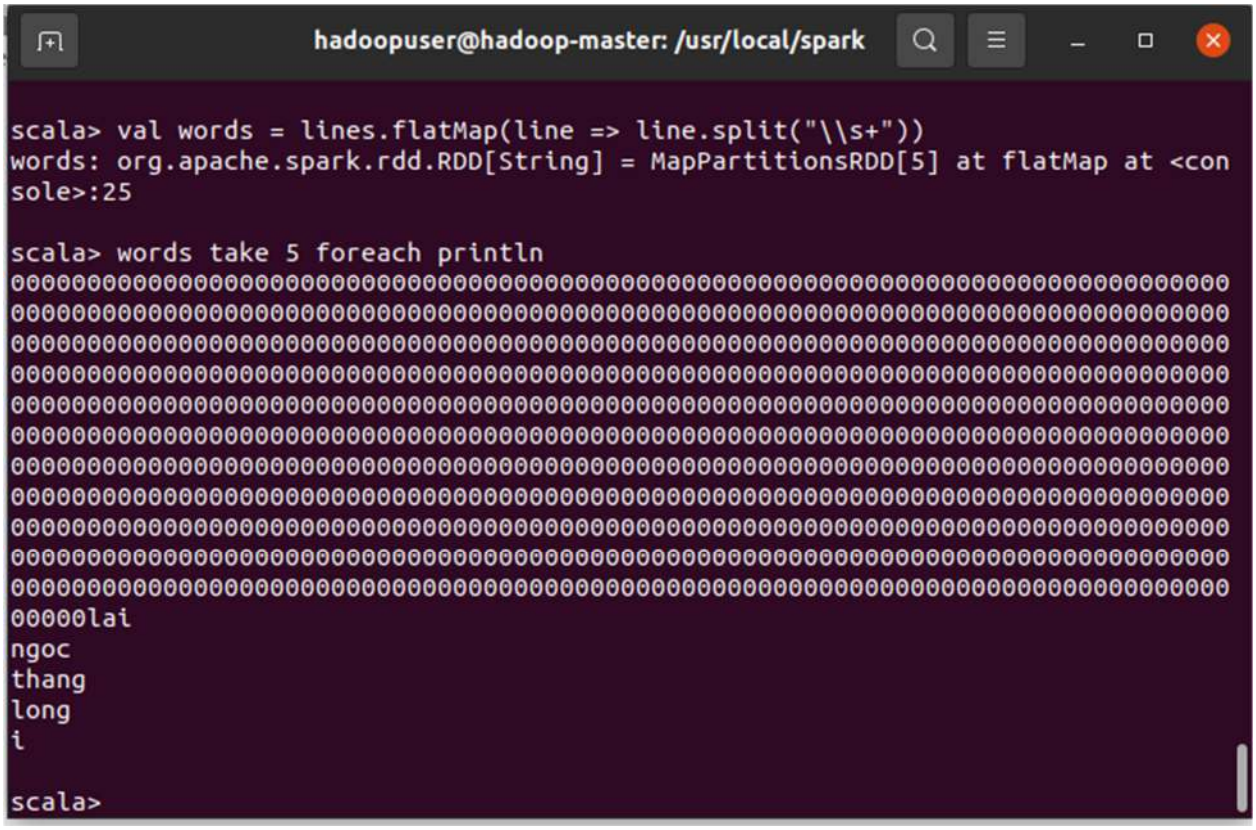
Step 2: Mở Spark-shell

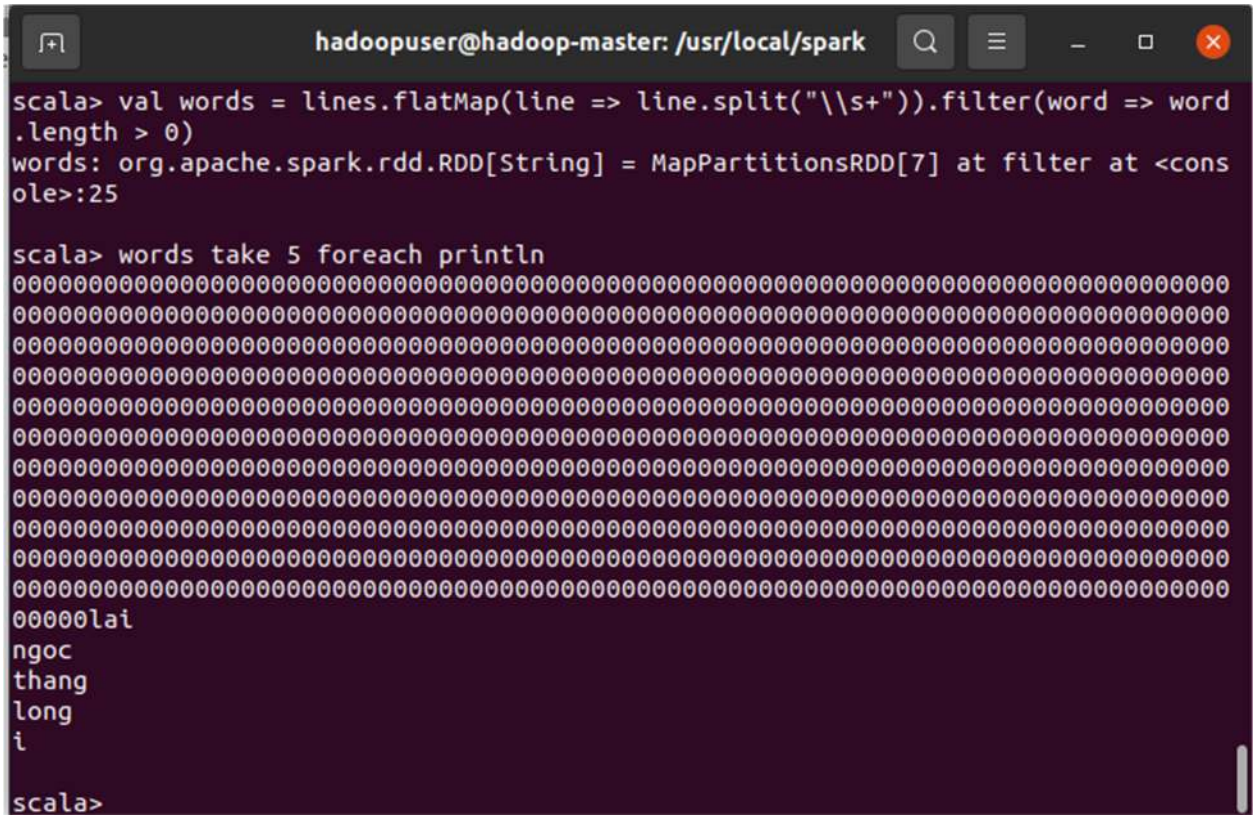


Step 3: Tạo 1 RDD



Step 4: Transformations


hadoopuser@hadoop-master: /usr/local/spark
scala> val words = lines.flatMap(line => line.split("\\s+"))
words: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[5] at flatMap at <con
sole>:25
scala> words take 5 foreach println
00000lai
ngoc
thang
long
i
scala>


hadoopuser@hadoop-master: /usr/local/spark
scala> val words = lines.flatMap(line => line.split("\\s+")).filter(word => word
.length > 0)
words: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[7] at filter at <cons
ole>:25
scala> words take 5 foreach println
00000lai
ngoc
thang
long
i
scala>

```
scala> val pairs = words.map(word => (word, 1))
pairs: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[9] at map at <console>:25

scala> pairs take 5 foreach println
(0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000lai,1)
(ngoc,1)
(thang,1)
(long,1)
(i,1)

scala> 
```

Step 5: Reduction

```
scala> val wordCounts = pairs.reduceByKey((a, b) => a + b)
wordCounts: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[10] at reduc
eByKey at <console>:25

scala> val summary = wordCounts.takeOrdered(10)(Ordering[Int].reverse.on(_._2)
)
[Stage 4:>
                                             summary: Array[(String, Int)] = Ar
ray((ha,2), (long,2), (ngoc,2), (tinh,2), (thang,2), (am,1), (0000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000...
scala> summary.foreach(println)
```

```
scala> summary.foreach(println)
(ha,2)
(long,2)
(ngoc,2)
(tinh,2)
(thang,2)
(am,1)
(000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000
```

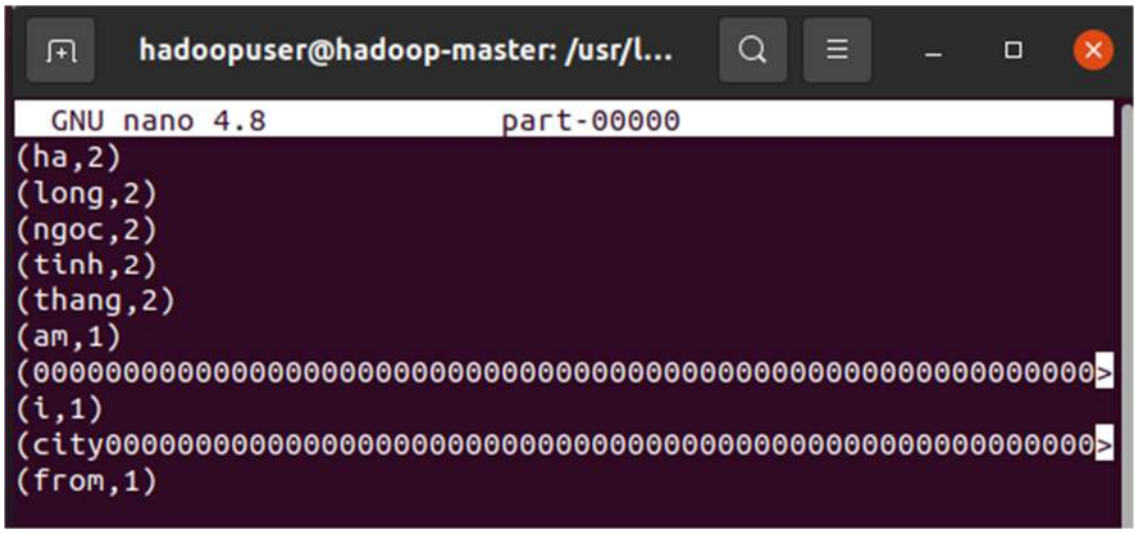
Step 6: Writing to file

```
scala> val summaryRDD = sc.makeRDD(summary)
summaryRDD: org.apache.spark.rdd.RDD[(String, Int)] = Paralle
lCollectionRDD[12] at makeRDD at <console>:26

scala> summaryRDD.saveAs
saveAsHadoopDataset         saveAsObjectFile
saveAsHadoopFile            saveAsSequenceFile
saveAsNewAPIHadoopDataset   saveAsTextFile
saveAsNewAPIHadoopFile

scala> summaryRDD.saveAsTextFile("test")

scala>
```

Sau khi lưu ta được:

```
hadoopuser@hadoop-master:/usr/local/spark$ cd test
hadoopuser@hadoop-master:/usr/local/spark/test$ ls
part-00000  _SUCCESS
hadoopuser@hadoop-master:/usr/local/spark/test$
```

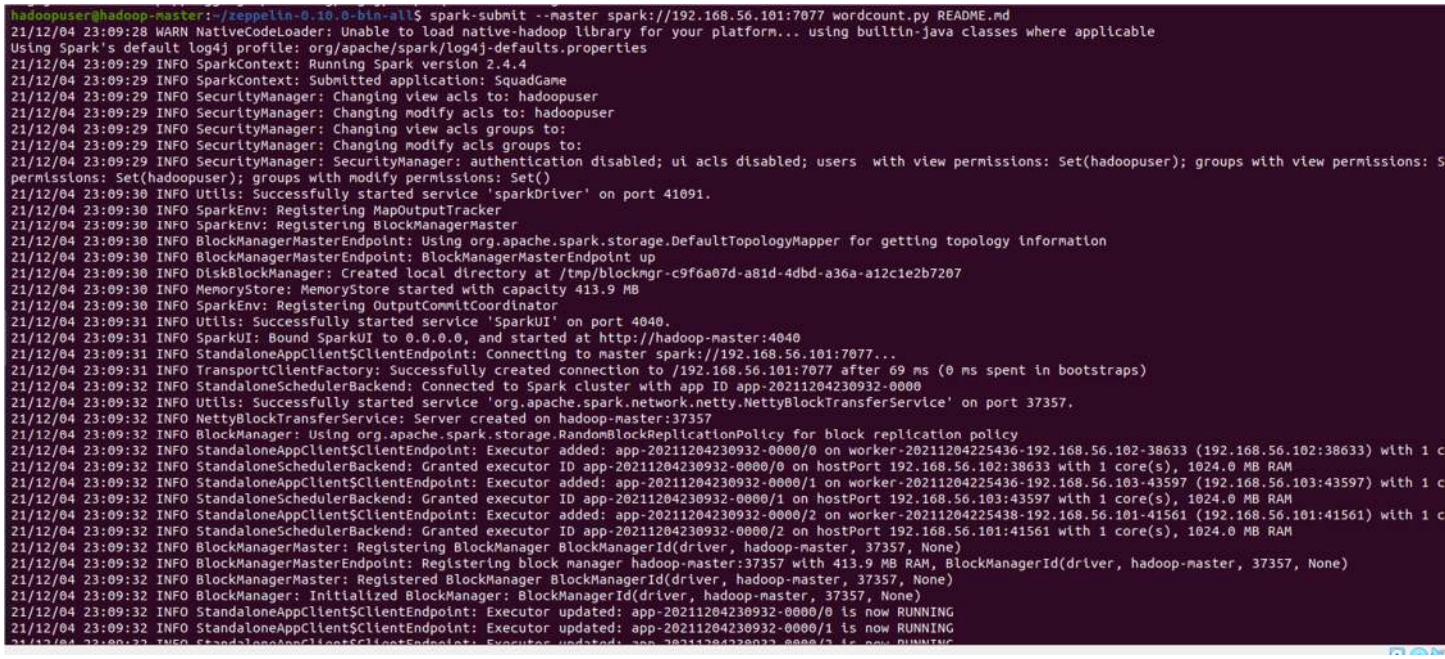
Xem nội dung file part-00000



III, Chạy chương trình WordCount trên Spark cluster

Dùng câu lệnh:

$spark-submit --master spark://192.168.56.101:7077 wordcount.py README.md



```
        at org.apache.spark.rdd.HadoopRDD.compute(HadoopRDD.scala:95)
        at org.apache.spark.rdd.RDD.computeOrReadCheckpoint(RDD.scala:324)
        at org.apache.spark.rdd.RDD.iterator(RDD.scala:288)
        at org.apache.spark.rdd.MapPartitionsRDD.compute(MapPartitionsRDD.scala:52)
        at org.apache.spark.rdd.RDD.computeOrReadCheckpoint(RDD.scala:324)
        at org.apache.spark.rdd.RDD.iterator(RDD.scala:288)
        at org.apache.spark.api.python.PythonRDD.compute(PythonRDD.scala:65)
        at org.apache.spark.rdd.RDD.computeOrReadCheckpoint(RDD.scala:324)
        at org.apache.spark.rdd.RDD.iterator(RDD.scala:288)
        at org.apache.spark.api.python.PairwiseRDD.compute(PythonRDD.scala:103)
        at org.apache.spark.rdd.RDD.computeOrReadCheckpoint(RDD.scala:324)
        at org.apache.spark.rdd.RDD.iterator(RDD.scala:288)
        at org.apache.spark.scheduler.ShuffleMapTask.runTask(ShuffleMapTask.scala:99)
        at org.apache.spark.scheduler.ShuffleMapTask.runTask(ShuffleMapTask.scala:55)
        at org.apache.spark.scheduler.Task.run(Task.scala:123)
        at org.apache.spark.executor.Executor$TaskRunner$$anonfun$10.apply(Executor.scala:408)
        at org.apache.spark.util.Utils$.tryWithSafeFinally(Utils.scala:1360)
        at org.apache.spark.executor.Executor$TaskRunner.run(Executor.scala:414)
        at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
        ... 1 more

21/12/04 23:09:41 INFO SparkContext: Invoking stop() from shutdown hook
21/12/04 23:09:41 INFO SparkUI: Stopped Spark web UI at http://hadoop-master:4040
21/12/04 23:09:41 INFO StandaloneSchedulerBackend: Shutting down all executors
21/12/04 23:09:41 INFO CoarseGrainedSchedulerBackend$DriverEndpoint: Asking each executor to shut down
21/12/04 23:09:42 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
21/12/04 23:09:42 INFO MemoryStore: MemoryStore cleared
21/12/04 23:09:42 INFO BlockManager: BlockManager stopped
21/12/04 23:09:42 INFO BlockManagerMaster: BlockManagerMaster stopped
21/12/04 23:09:42 INFO OutputCommitCoordinator$OutputCommitCoordinatorEndpoint: OutputCommitCoordinator stopped!
21/12/04 23:09:42 INFO SparkContext: Successfully stopped SparkContext
21/12/04 23:09:42 INFO ShutdownHookManager: Shutdown hook called
21/12/04 23:09:42 INFO ShutdownHookManager: Deleting directory /tmp/spark-1a0273a4-4fbf-4f6d-b10b-d324fec5c4a2/pyspark-c8efe61b-71e5-4272-9ef6-319e30e214f1
21/12/04 23:09:42 INFO ShutdownHookManager: Deleting directory /tmp/spark-1a0273a4-4fbf-4f6d-b10b-d324fec5c4a2
21/12/04 23:09:42 INFO ShutdownHookManager: Deleting directory /tmp/spark-8f78369c-a64d-4a7a-8b38-10c0689dfa3e
hadoopuser@hadoop-master:~/zeppelin-0.10.0-bin-all$
```

Spark Master at spark://192.168.56.101:7077

URL: spark://192.168.56.101:7077
Alive Workers: 3
Cores in use: 3 Total, 0 Used
Memory in use: 8.5 GB Total, 0.0 B Used
Applications: 0 Running, 1 Completed
Drivers: 0 Running, 0 Completed
Status: ALIVE

Workers (3)

Worker Id	Address	State	Cores	Memory
worker-20211204225436-192.168.56.102-38633	192.168.56.102:38633	ALIVE	1 (0 Used)	2.8 GB (0.0 B Used)
worker-20211204225436-192.168.56.103-43597	192.168.56.103:43597	ALIVE	1 (0 Used)	2.8 GB (0.0 B Used)
worker-20211204225438-192.168.56.101-41561	192.168.56.101:41561	ALIVE	1 (0 Used)	2.8 GB (0.0 B Used)

Running Applications (0)

Application ID	Name	Cores	Memory per Executor	Submitted Time	User	State	Duration

Completed Applications (1)

Application ID	Name	Cores	Memory per Executor	Submitted Time	User	State	Duration
app-20211204230932-0000	SquadGame	3	1024.0 MB	2021/12/04 23:09:32	hadoopuser	FINISHED	10 s

Application: SquadGame

ID: app-20211204230932-0000
Name: SquadGame
User: hadoopuser
Cores: Unlimited (3 granted)
Executor Limit: Unlimited (3 granted)
Executor Memory: 1024.0 MB
Submit Date: 2021/12/04 23:09:32
State: FINISHED

Executor Summary (3)

ExecutorID	Worker	Cores	Memory	State	Logs

Removed Executors (3)

ExecutorID	Worker	Cores	Memory	State	Logs
2	worker-20211204225438-192.168.56.101-41561	1	1024	KILLED	stdout stderr
1	worker-20211204225436-192.168.56.103-43597	1	1024	KILLED	stdout stderr
0	worker-20211204225436-192.168.56.102-38633	1	1024	KILLED	stdout stderr