BÁO CÁO LAB5 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

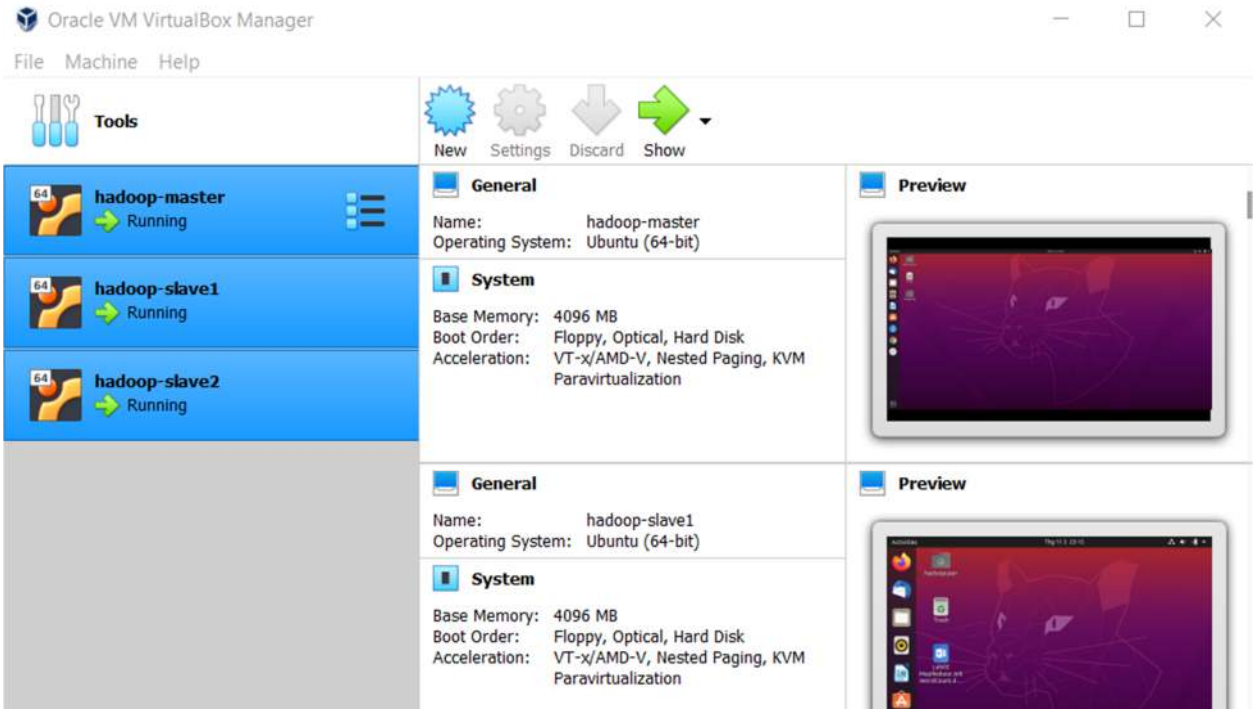
Nguyễn Khương Duy 20183513

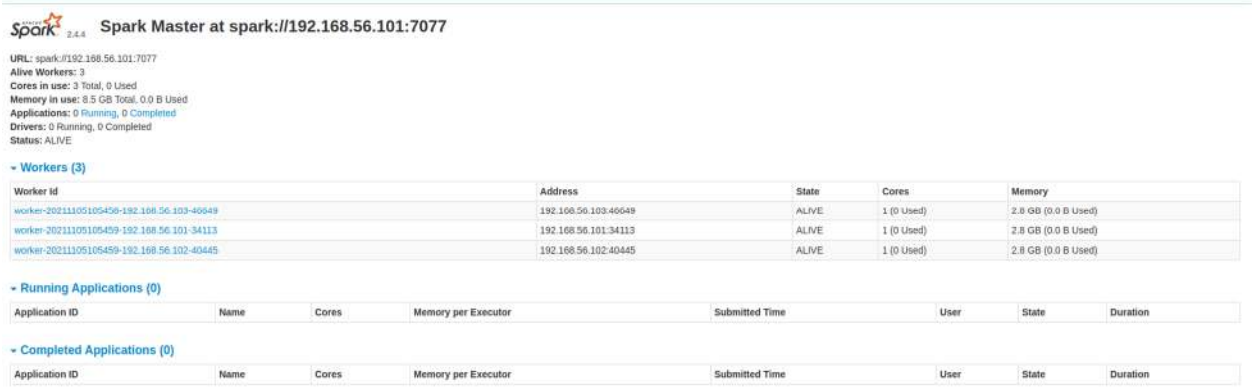
BÀI LÀM:

I. Cài đặt Spark Cluster

1st Step: Cài đặt Spark Cluster trên 3 máy

Step 1 — 3 máy Master, Slave1, Slave2





Ta thấy Alive Workers: 3

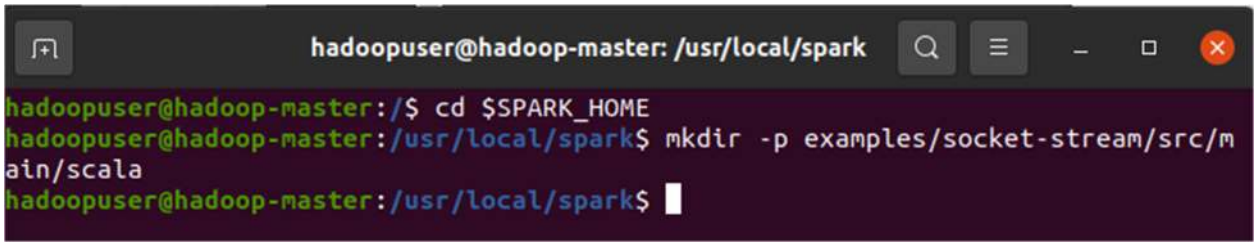
II, Create a WordCount Application

- Move to the SPARK_HOME folder:

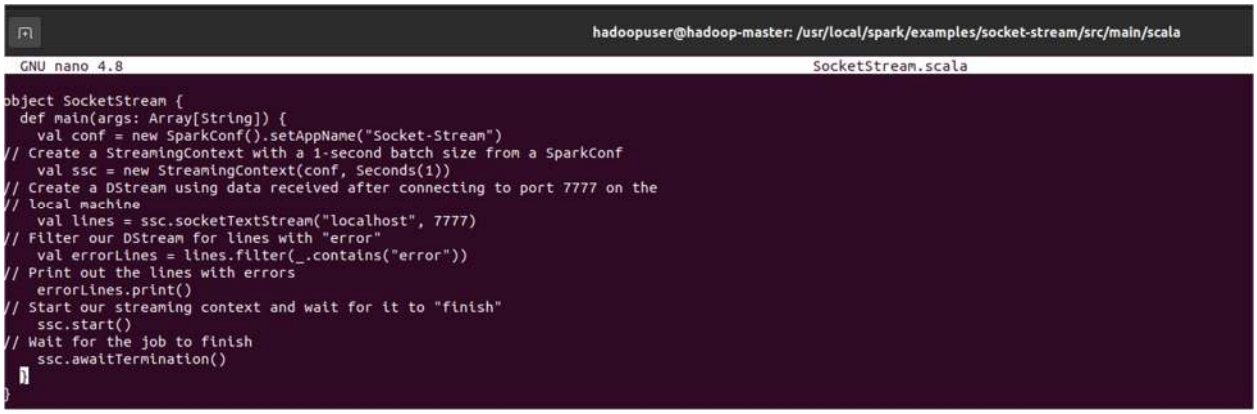
$ cd $SPARK_HOME

- Create a directory to save the source code:

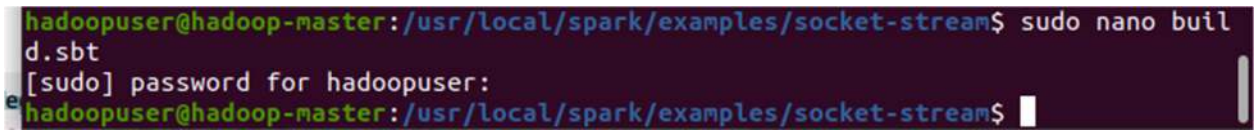
$ mkdir -p examples/socket-stream/src/main/scala

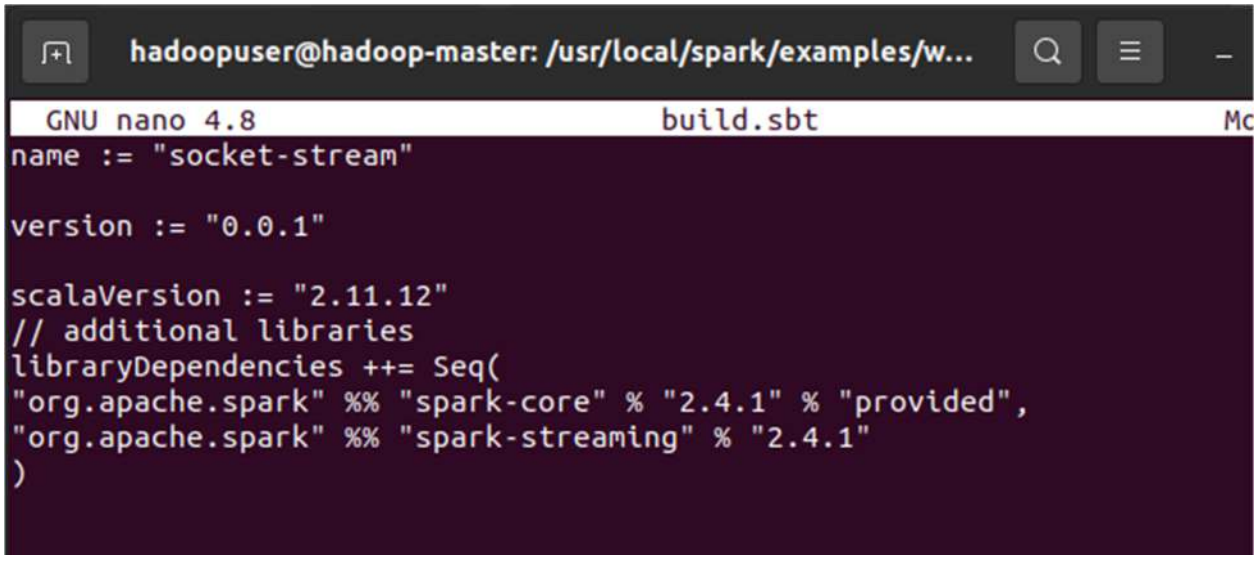


- Create a file name SocketStream.scala in $SPARK_HOME/examples/socket-stream/src/main/scala folder:

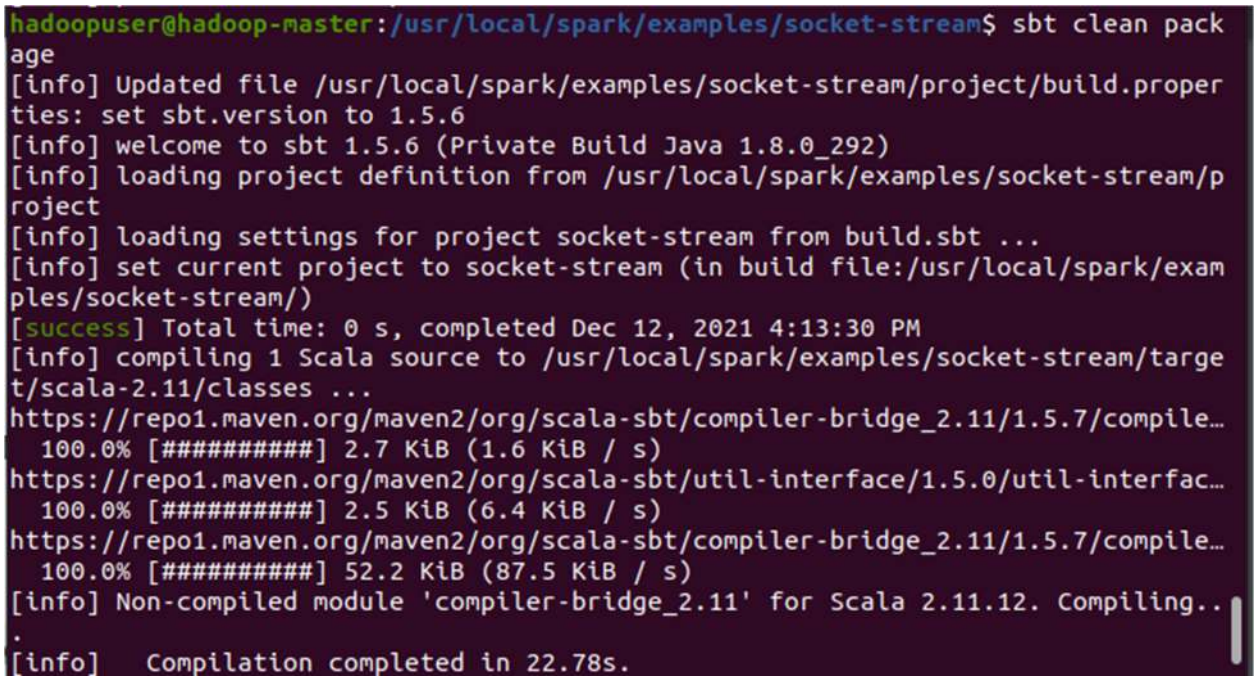


- Create file build.sbt in $SPARK_HOME/examples/socket-stream:



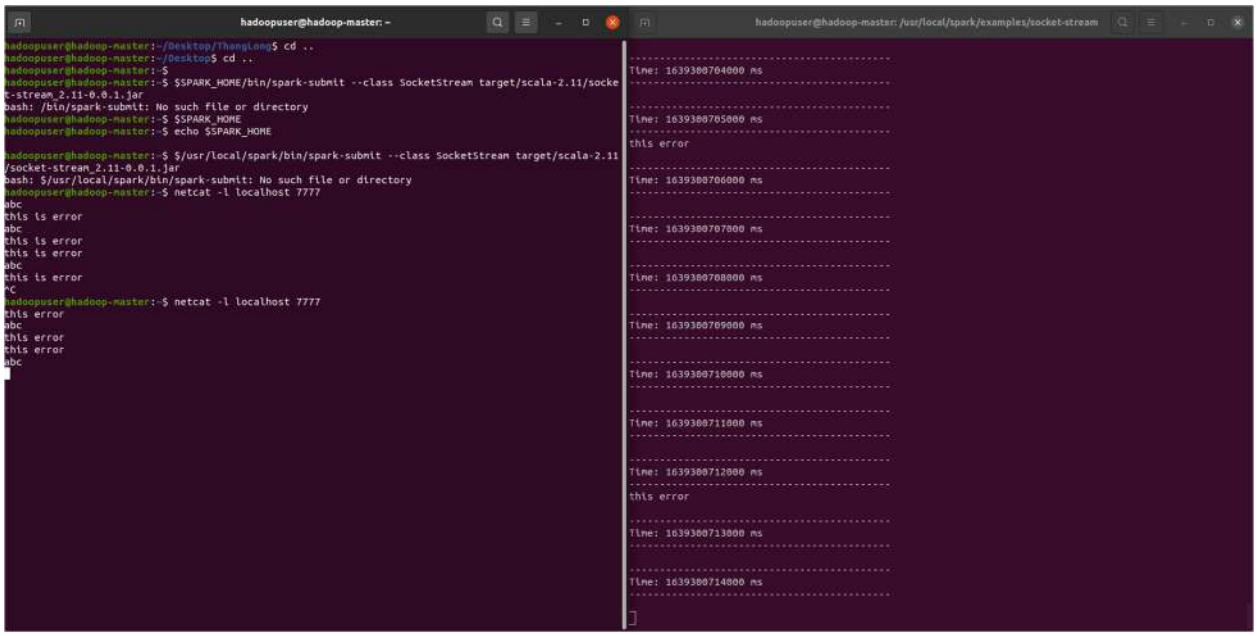


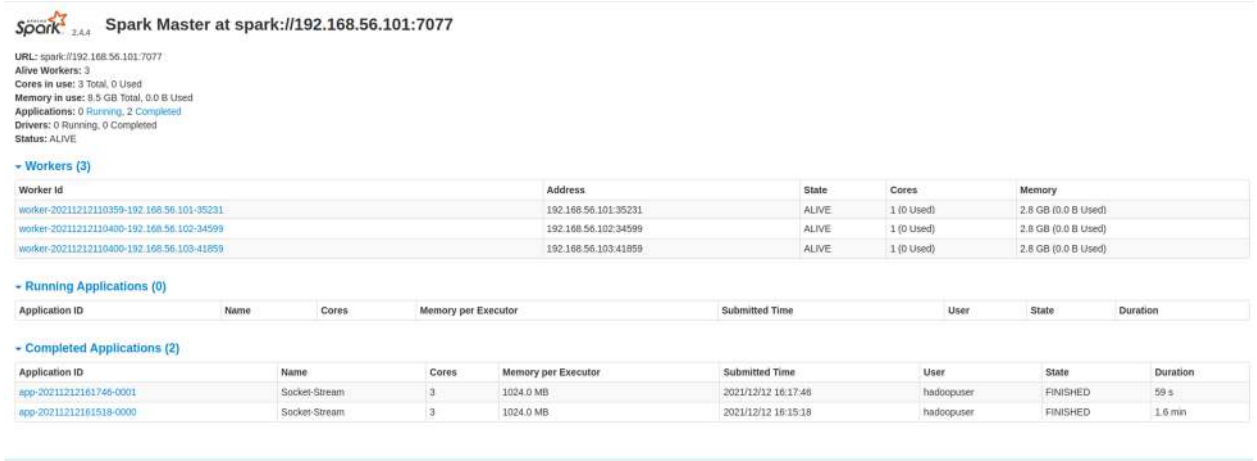
- Build application : $ sbt clean package

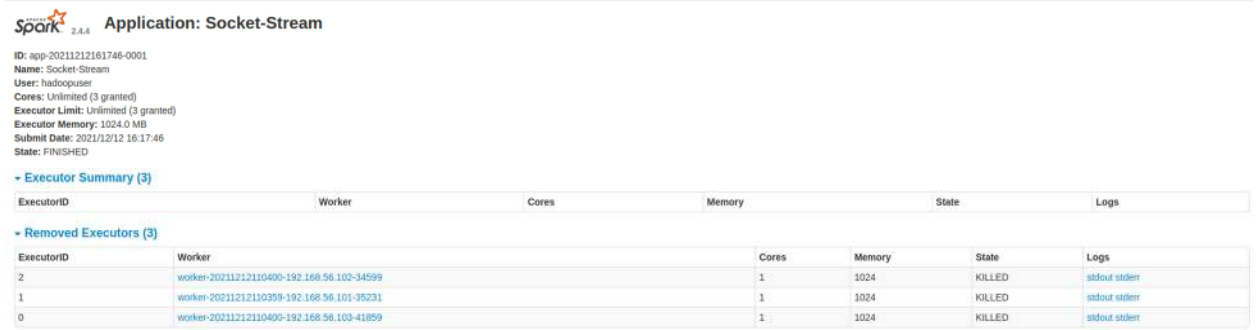


- Submit and run in Spark :

$ $SPARK_HOME/bin/spark-submit --class SocketStream --master: spark://192.168.56.101:7077 target/scala-2.11/socket-stream_2.11-0.0.1.jar







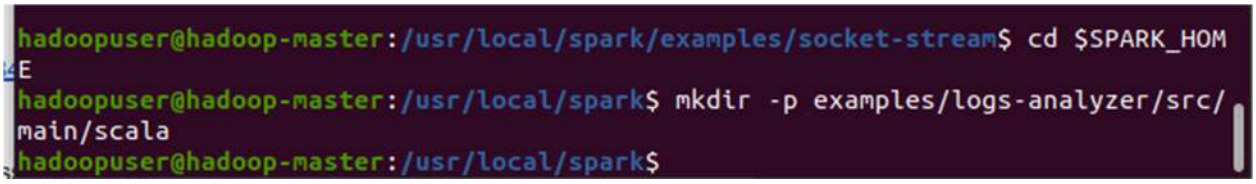
III, Create a Log Analyzer

- Move to the SPARK_HOME folder:

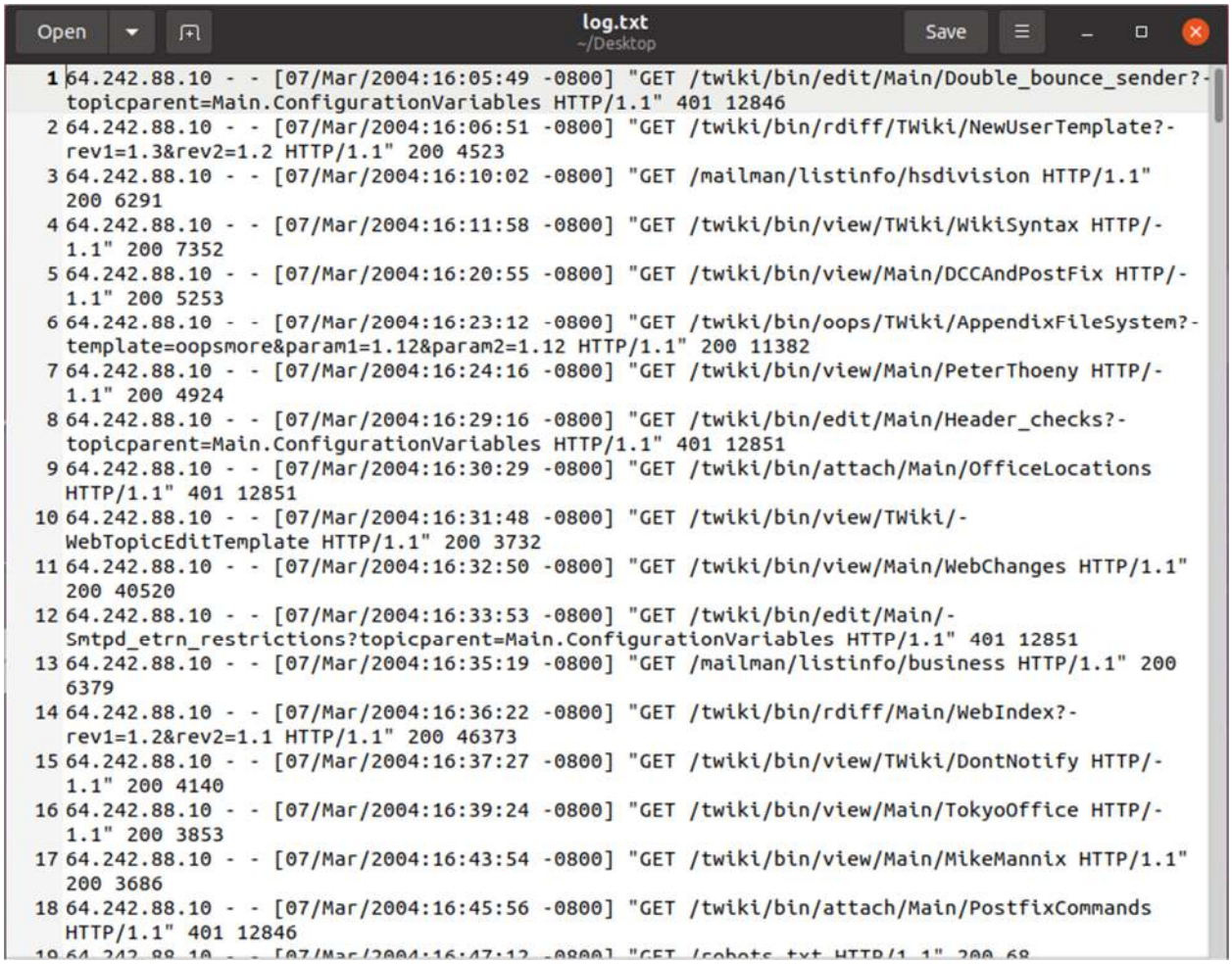
$ cd $SPARK_HOME

- Create a directory to save the source code:

$ mkdir -p examples/logs-analyzer/src/main/scala



- File log.txt:



- Create a file name ApacheAccessLog.scala in $SPARK_HOME/examples/logs-analyzer/src/main/scala folder:

src > main > scala > ApacheAccessLog.scala

```scala
case class ApacheAccessLog(
    ipAddress: String,
    clientIdentd: String,
    userId: String,
    dateTime: String,
    method: String,
    endpoint: String,
    protocal: String,
    responseCode: Int,
    contentSize: Long){
  }

object ApacheAccessLog {
     val PATTERN = """^(\S+) (\S+) (\S+) \[([\w:/]+\s[+\-]\d{4})\] "(\S+) (\S+) (\S+)" (\d{3}) (\d+)""".r
     def parseLogLine(log: String): ApacheAccessLog = {
    log match {
       case PATTERN(ipAddress,
     clientIdentd,
     userId,
     dateTime,
     method,
     endpoint,
     protocol,
     responseCode,
     contentSize) => new ApacheAccessLog(ipAddress,
            clientIdentd,
            userId,
            dateTime,
            method,
            endpoint,
            protocol,
            responseCode.toInt,
            contentSize.toLong)
      case _ => throw new RuntimeException(s"""Cannot parse log line $log""")
    }
     }
  }
```

- Create a file name LogAnalyzerStreaming.scala in $SPARK_HOME/examples/logs-analyzer/src/main/scala folder:

src > main > scala > LogAnalyzerStreaming.scala

```scala
import org.apache.spark.SparkConf
import org.apache.spark.rdd.RDD
import org.apache.spark.streaming.dstream.DStream
import org.apache.spark.streaming.{Seconds, StreamingContext}
import org.apache.spark._
import org.apache.spark.streaming._
import org.apache.spark.streaming.StreamingContext._ // not necessary since Spark 1.3

```

src > main > scala > LogAnalyzerStreaming.scala

```scala
import org.apache.spark.streaming._
import org.apache.spark.streaming.StreamingContext._ // not necessary since Spark 1.3

object LogAnalyzerStreaming{
    def main(args: Array[String]){
        val WINDOW_LENGTH = Seconds(30)
    val SLIDE_INTERVAL = Seconds(10)

    val sparkConf = new SparkConf().setAppName("Log Analyzer Streaming in Scala")
    val streamingContext = new StreamingContext(sparkConf, SLIDE_INTERVAL)

    val logLinesDStream: DStream[String] = streamingContext.socketTextStream("localhost", 9999)

    val accessLogsDStream: DStream[ApacheAccessLog] = logLinesDStream.map(ApacheAccessLog.parseLogLine).cache()

    val windowDStream: DStream[ApacheAccessLog] = accessLogsDStream.window(WINDOW_LENGTH, SLIDE_INTERVAL)


    windowDStream.foreachRDD(accessLogs => {
        if(accessLogs.count() == 0){
        println("No access logs received in this time interval")
        } else {
        val contentSizes: RDD[Long] = accessLogs.map(_.contentSize).cache()
        println("Content Size Avg: %s, Min: %s, Max: %s".format(contentSizes.reduce(_+_) / contentSizes.count, contentSizes.min, contentSizes.max))

        val responseCodeToCount: Array[(Int, Long)] = accessLogs.map(_.responseCode -> 1L).reduceByKey(_+_).take(100)
        println(s"""Response code counts: ${responseCodeToCount.mkString("[", ",", "]")}""")

        val ipAddresses: Array[String] = accessLogs.map(_.ipAddress -> 1L).reduceByKey(_+_).filter(_._2 > 10).map(_._1).take(100)
        println(s"""IPAddresses > 10 times: ${ipAddresses.mkString("[", ",", "]")}""")

        val topEndpoints: Array[(String, Long)] = accessLogs.map(_.endpoint -> 1L).reduceByKey(_+_).top(10)(Ordering.by[(String, Long), Long](_._2))
        println(s"""Top Endpoints: ${topEndpoints.mkString("[", ",", "]")}""")
        }
    })

    streamingContext.start()
        streamingContext.awaitTermination()
    }
}
```


- Create file build.sbt in $SPARK_HOME/examples/logs-analyzer

build.sbt

```
name := "log-analyzer"

version := "0.0.1"

scalaVersion := "2.11.12"
// additional libraries
libraryDependencies ++= Seq(
"org.apache.spark" %% "spark-core" % "2.4.1" % "provided",
"org.apache.spark" %% "spark-streaming" % "2.4.1"
)
```

- Create the shell script named "stream.sh" that emulates network stream by periodically

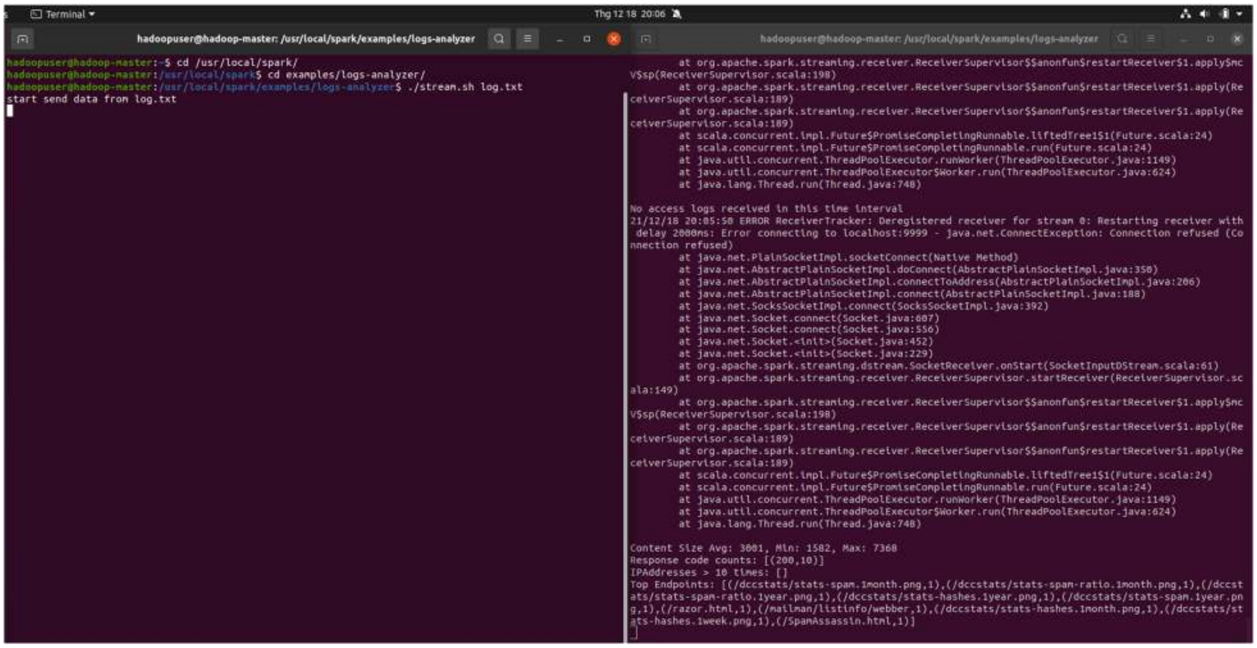
sending portions of the sample log file to a network socket:

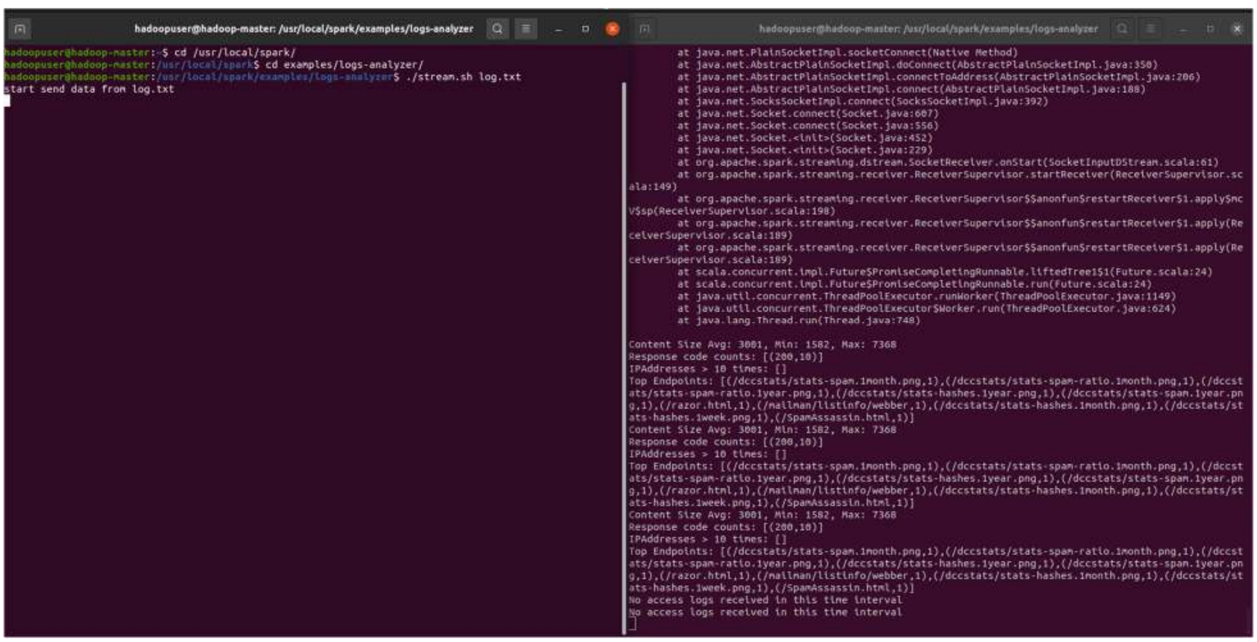
```
$ stream.sh
#!/bin/sh

set -o nounset
set -o errexit

test $# -eq 1 || ( echo "Incorrect number of arguments" ; exit 1 )

file="$1"
echo "start send data from $file"

network_port=9999
lines_in_batch=100
interval_sec=10

n_lines=$(cat $file | wc -l)

cursor=10

while test $cursor -le $n_lines
do
    tail -n $cursor $file | head -n $lines_in_batch | nc -l $network_port
    cursor=$(($cursor + $lines_in_batch))
    sleep $interval_sec
done
```

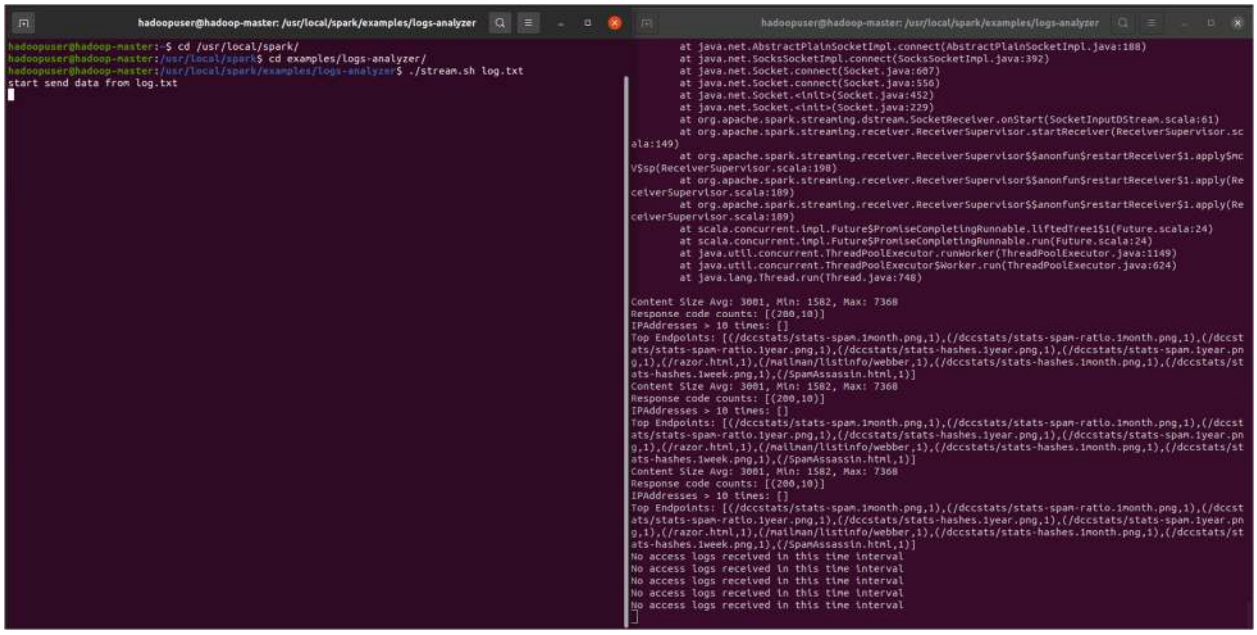
- Build application : $ sbt clean package

```
hadoopuser@hadoop-master:/usr/local/spark/examples/logs-analyzer$ sbt clean pack
age
[info] welcome to sbt 1.5.6 (Private Build Java 1.8.0_292)
[info] loading project definition from /usr/local/spark/examples/logs-analyzer/p
roject
[info] loading settings for project logs-analyzer from build.sbt ...
[info] set current project to log-analyzer (in build file:/usr/local/spark/examp
les/logs-analyzer/)
[success] Total time: 0 s, completed Dec 18, 2021 7:07:09 PM
[info] compiling 2 Scala sources to /usr/local/spark/examples/logs-analyzer/targ
et/scala-2.11/classes ...
[success] Total time: 22 s, completed Dec 18, 2021 7:07:31 PM
hadoopuser@hadoop-master:/usr/local/spark/examples/logs-analyzer$
```

- Submit and run in Spark : $ $SPARK_HOME/bin/spark-submit --class "LogAnalyzerStreaming" --master: spark://192.168.56.101:7077 target/scala-2.11/log-analyzer_2.11-0.0.1.jar


Terminal
Thg 12 18 20:06
hadoopuser@hadoop-master: /usr/local/spark/examples/logs-analyzer
hadoopuser@hadoop-master:~$ cd /usr/local/spark/
hadoopuser@hadoop-master:/usr/local/spark$ cd examples/logs-analyzer/
hadoopuser@hadoop-master:/usr/local/spark/examples/logs-analyzer$ ./stream.sh log.txt
start send data from log.txt
hadoopuser@hadoop-master: /usr/local/spark/examples/logs-analyzer
at org.apache.spark.streaming.receiver.ReceiverSupervisor$$anonfun$restartReceiver$1.applymcVsp(ReceiverSupervisor.scala:198)
at org.apache.spark.streaming.receiver.ReceiverSupervisor$$anonfun$restartReceiver$1.apply(ReceiverSupervisor.scala:189)
at org.apache.spark.streaming.receiver.ReceiverSupervisor$$anonfun$restartReceiver$1.apply(ReceiverSupervisor.scala:189)
at scala.concurrent.impl.Future$PromiseCompletingRunnable.liftedTree1$1(Future.scala:24)
at scala.concurrent.impl.Future$PromiseCompletingRunnable.run(Future.scala:24)
at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
at java.lang.Thread.run(Thread.java:748)
No access logs received in this time interval
21/12/18 20:05:58 ERROR ReceiverTracker: Deregistered receiver for stream 0: Restarting receiver with delay 2000ms: Error connecting to localhost:9999 - java.net.ConnectException: Connection refused (Connection refused)
at java.net.PlainSocketImpl.socketConnect(Native Method)
at java.net.AbstractPlainSocketImpl.doConnect(AbstractPlainSocketImpl.java:350)
at java.net.AbstractPlainSocketImpl.connectToAddress(AbstractPlainSocketImpl.java:206)
at java.net.AbstractPlainSocketImpl.connect(AbstractPlainSocketImpl.java:188)
at java.net.SocksSocketImpl.connect(SocksSocketImpl.java:392)
at java.net.Socket.connect(Socket.java:607)
at java.net.Socket.connect(Socket.java:556)
at java.net.Socket.<init>(Socket.java:452)
at java.net.Socket.<init>(Socket.java:229)
at org.apache.spark.streaming.dstream.SocketReceiver.onStart(SocketInputDStream.scala:61)
at org.apache.spark.streaming.receiver.ReceiverSupervisor.startReceiver(ReceiverSupervisor.scala:149)
at org.apache.spark.streaming.receiver.ReceiverSupervisor$$anonfun$restartReceiver$1.applymcVsp(ReceiverSupervisor.scala:198)
at org.apache.spark.streaming.receiver.ReceiverSupervisor$$anonfun$restartReceiver$1.apply(ReceiverSupervisor.scala:189)
at org.apache.spark.streaming.receiver.ReceiverSupervisor$$anonfun$restartReceiver$1.apply(ReceiverSupervisor.scala:189)
at scala.concurrent.impl.Future$PromiseCompletingRunnable.liftedTree1$1(Future.scala:24)
at scala.concurrent.impl.Future$PromiseCompletingRunnable.run(Future.scala:24)
at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
at java.lang.Thread.run(Thread.java:748)
Content Size Avg: 3001, Min: 1582, Max: 7368
Response code counts: [(200,10)]
IPAddresses > 10 times: []
Top Endpoints: [(/dccstats/stats-spam.1month.png,1),(/dccstats/stats-spam-ratio.1month.png,1),(/dccstats/stats-spam-ratio.1year.png,1),(/dccstats/stats-hashes.1year.png,1),(/dccstats/stats-spam.1year.png,1),(/razor.html,1),(/mailman/listinfo/webber,1),(/dccstats/stats-hashes.1month.png,1),(/dccstats/stats-hashes.1week.png,1),(/SpamAssassin.html,1)]


hadoopuser@hadoop-master: /usr/local/spark/examples/logs-analyzer
hadoopuser@hadoop-master:~$ cd /usr/local/spark/
hadoopuser@hadoop-master:/usr/local/spark$ cd examples/logs-analyzer/
hadoopuser@hadoop-master:/usr/local/spark/examples/logs-analyzer$ ./stream.sh log.txt
start send data from log.txt
hadoopuser@hadoop-master: /usr/local/spark/examples/logs-analyzer
at java.net.PlainSocketImpl.socketConnect(Native Method)
at java.net.AbstractPlainSocketImpl.doConnect(AbstractPlainSocketImpl.java:350)
at java.net.AbstractPlainSocketImpl.connectToAddress(AbstractPlainSocketImpl.java:206)
at java.net.AbstractPlainSocketImpl.connect(AbstractPlainSocketImpl.java:188)
at java.net.SocksSocketImpl.connect(SocksSocketImpl.java:392)
at java.net.Socket.connect(Socket.java:607)
at java.net.Socket.connect(Socket.java:556)
at java.net.Socket.<init>(Socket.java:452)
at java.net.Socket.<init>(Socket.java:229)
at org.apache.spark.streaming.dstream.SocketReceiver.onStart(SocketInputDStream.scala:61)
at org.apache.spark.streaming.receiver.ReceiverSupervisor.startReceiver(ReceiverSupervisor.scala:149)
at org.apache.spark.streaming.receiver.ReceiverSupervisor$$anonfun$restartReceiver$1.applymcVsp(ReceiverSupervisor.scala:198)
at org.apache.spark.streaming.receiver.ReceiverSupervisor$$anonfun$restartReceiver$1.apply(ReceiverSupervisor.scala:189)
at org.apache.spark.streaming.receiver.ReceiverSupervisor$$anonfun$restartReceiver$1.apply(ReceiverSupervisor.scala:189)
at scala.concurrent.impl.Future$PromiseCompletingRunnable.liftedTree1$1(Future.scala:24)
at scala.concurrent.impl.Future$PromiseCompletingRunnable.run(Future.scala:24)
at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
at java.lang.Thread.run(Thread.java:748)
Content Size Avg: 3001, Min: 1582, Max: 7368
Response code counts: [(200,10)]
IPAddresses > 10 times: []
Top Endpoints: [(/dccstats/stats-spam.1month.png,1),(/dccstats/stats-spam-ratio.1month.png,1),(/dccstats/stats-spam-ratio.1year.png,1),(/dccstats/stats-hashes.1year.png,1),(/dccstats/stats-spam.1year.png,1),(/razor.html,1),(/mailman/listinfo/webber,1),(/dccstats/stats-hashes.1month.png,1),(/dccstats/stats-hashes.1week.png,1),(/SpamAssassin.html,1)]
Content Size Avg: 3001, Min: 1582, Max: 7368
Response code counts: [(200,10)]
IPAddresses > 10 times: []
Top Endpoints: [(/dccstats/stats-spam.1month.png,1),(/dccstats/stats-spam-ratio.1month.png,1),(/dccstats/stats-spam-ratio.1year.png,1),(/dccstats/stats-hashes.1year.png,1),(/dccstats/stats-spam.1year.png,1),(/razor.html,1),(/mailman/listinfo/webber,1),(/dccstats/stats-hashes.1month.png,1),(/dccstats/stats-hashes.1week.png,1),(/SpamAssassin.html,1)]
Content Size Avg: 3001, Min: 1582, Max: 7368
Response code counts: [(200,10)]
IPAddresses > 10 times: []
Top Endpoints: [(/dccstats/stats-spam.1month.png,1),(/dccstats/stats-spam-ratio.1month.png,1),(/dccstats/stats-spam-ratio.1year.png,1),(/dccstats/stats-hashes.1year.png,1),(/dccstats/stats-spam.1year.png,1),(/razor.html,1),(/mailman/listinfo/webber,1),(/dccstats/stats-hashes.1month.png,1),(/dccstats/stats-hashes.1week.png,1),(/SpamAssassin.html,1)]
No access logs received in this time interval
No access logs received in this time interval



Spark Master at spark://192.168.56.101:7077

URL: spark://192.168.56.101:7077
Alive Workers: 3
Cores in use: 3 Total, 3 Used
Memory in use: 8.5 GB Total, 3.0 GB Used
Applications: 1 Running, 6 Completed
Drivers: 0 Running, 0 Completed
Status: ALIVE

Workers (3)

Worker Id	Address	State	Cores	Memory
worker-20211212110359-192.168.56.101-35231	192.168.56.101:35231	ALIVE	1 (1 Used)	2.8 GB (1024.0 MB Used)
worker-20211212110400-192.168.56.102-34599	192.168.56.102:34599	ALIVE	1 (1 Used)	2.8 GB (1024.0 MB Used)
worker-20211212110400-192.168.56.103-41859	192.168.56.103:41859	ALIVE	1 (1 Used)	2.8 GB (1024.0 MB Used)

Running Applications (1)

Application ID	Name	Cores	Memory per Executor	Submitted Time	User	State	Duration
app-20211212175057-0006 (kill)	Log Analyzer Streaming in Scala	3	1024.0 MB	2021/12/12 17:50:57	hadoopuser	RUNNING	4.7 min

Completed Applications (6)

Application ID	Name	Cores	Memory per Executor	Submitted Time	User	State	Duration
app-20211212175022-0005	Log Analyzer Streaming in Scala	3	1024.0 MB	2021/12/12 17:50:22	hadoopuser	FINISHED	19 s
app-20211212174923-0004	Log Analyzer Streaming in Scala	3	1024.0 MB	2021/12/12 17:49:23	hadoopuser	FINISHED	51 s
app-20211212173540-0003	Log Analyzer Streaming in Scala	3	1024.0 MB	2021/12/12 17:35:40	hadoopuser	FINISHED	11 min
app-20211212172036-0002	Log Analyzer Streaming in Scala	3	1024.0 MB	2021/12/12 17:20:36	hadoopuser	FINISHED	8.3 min
app-20211212161748-0001	Socket-Stream	3	1024.0 MB	2021/12/12 16:17:48	hadoopuser	FINISHED	59 s
app-20211212161518-0000	Socket-Stream	3	1024.0 MB	2021/12/12 16:15:18	hadoopuser	FINISHED	1.6 min

Application: Log Analyzer Streaming in Scala

ID: app-20211212175022-0005
Name: Log Analyzer Streaming in Scala
User: hadoopuser
Cores: Unlimited (3 granted)
Executor Limit: Unlimited (3 granted)
Executor Memory: 1024.0 MB
Submit Date: 2021/12/12 17:50:22
State: FINISHED

Executor Summary (3)

ExecutorID	Worker	Cores	Memory	State	Logs

Removed Executors (3)

ExecutorID	Worker	Cores	Memory	State	Logs
2	worker-20211212110400-192.168.56.102-34599	1	1024	KILLED	stdout stderr
1	worker-20211212110359-192.168.56.101-35231	1	1024	KILLED	stdout stderr
0	worker-20211212110400-192.168.56.103-41859	1	1024	KILLED	stdout stderr

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-20211218185741-192.168.56.101-33787	192.168.56.101:33787	ALIVE	1 (0 Used)	2.8 GB (0.0 B Used)
worker-20211218185742-192.168.56.102-36657	192.168.56.102:36657	ALIVE	1 (0 Used)	2.8 GB (0.0 B Used)
worker-20211218185742-192.168.56.103-41619	192.168.56.103:41619	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-20211218191810-0000	Log Analyzer Streaming in Scala	3	1024.0 MB	2021/12/18 19:18:10	hadoopuser	FINISHED	49 min

Application: Log Analyzer Streaming in Scala

ID: app-20211218191810-0000
Name: Log Analyzer Streaming in Scala
User: hadoopuser
Cores: Unlimited (3 granted)
Executor Limit: Unlimited (3 granted)
Executor Memory: 1024.0 MB
Submit Date: 2021/12/18 19:18:10
State: FINISHED

Executor Summary (4)

ExecutorID	Worker	Cores	Memory	State	Logs
2	worker-20211218185741-192.168.56.101-33787	1	1024	EXITED	stdout stderr

Removed Executors (3)

ExecutorID	Worker	Cores	Memory	State	Logs
1	worker-20211218185742-192.168.56.103-41619	1	1024	KILLED	stdout stderr
3	worker-20211218185741-192.168.56.101-33787	1	1024	KILLED	stdout stderr
0	worker-20211218185742-192.168.56.102-36657	1	1024	KILLED	stdout stderr