**Spark Streaming:**

#### Intro:

Spark Streaming is an extension of the core Spark API that enables scalable, high-throughput, fault-tolerant stream processing of live data streams.

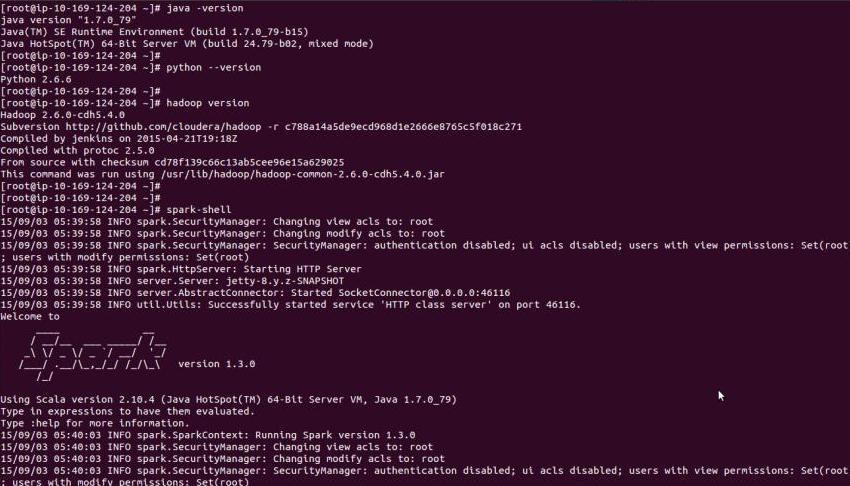
1. Pre-Requirements to start Spark Streaming
2. How to generate data using Python.
3. How to Access, Initialize, Dependencies Spark Streaming on Spark shell & Scala.
4. Files live stream example using Spark Streaming from Scala shell and using sbt.

Let’s go!

#### Step-1. Pre-Requirements to start Spark Streaming:

Below listed are required to start Spark Streaming.

1. Java 1.6+
2. Hadoop 1.X or 2.X - optional (for file storage or we can use local file system also)
3. Spark 1.X
4. Data source (we will be using filestream)
5. Python



This AMI “**UCB W205 Base - ami-98848cf0**” will be containing all required software’s for this lab.

#### Step-2. How to generate data using Python:

Here is the small snippet of code to generate data using python.

1. Create python file name “generate\_data.py” to generate data, you can find code in attached file “generate\_data.py” or you can create by using below python code.

#!/usr/bin/python

import threading

import time

import string

import random

import os

import uuid

class Generate\_events(threading.Thread):

def \_\_init\_\_(self, events\_count, file\_name):

threading.Thread.\_\_init\_\_(self)

if os.path.exists(file\_name):

self.file\_name = open(file\_name, 'a')

else:

self.file\_name = open(file\_name, 'w')

self.events\_count = events\_count

def \_\_gen\_sal(self):

while True:

sal = range(1000000)

yield random.choice(sal)

def \_\_gen\_emp\_number(self):

while True:

sal = range(1000000)

yield random.choice(sal)

def \_\_gen\_bonus(self):

while True:

bonus = range(500000)

yield random.choice(bonus)

def \_\_gen\_name(self):

alphabets = list(string.ascii\_lowercase)

while True:

yield ''.join(random.choice(alphabets) for \_ in range(6))

def \_get\_name(self):

return self.\_\_gen\_name().next()

def \_get\_sal(self):

return self.\_\_gen\_sal().next()

def \_get\_bonus(self):

return self.\_\_gen\_bonus().next()

def \_get\_emp\_number(self):

return self.\_\_gen\_emp\_number().next()

def run(self):

try:

for event in range(self.events\_count):

self.file\_name.write("\n" + str(self.\_get\_emp\_number()) + ',' + self.\_get\_name() + ',' + str(self.\_get\_sal()) + ',' + str(self.\_get\_bonus()))

self.file\_name.close()

except Exception, e:

print e

while(True):

#id = os.urandom(32)

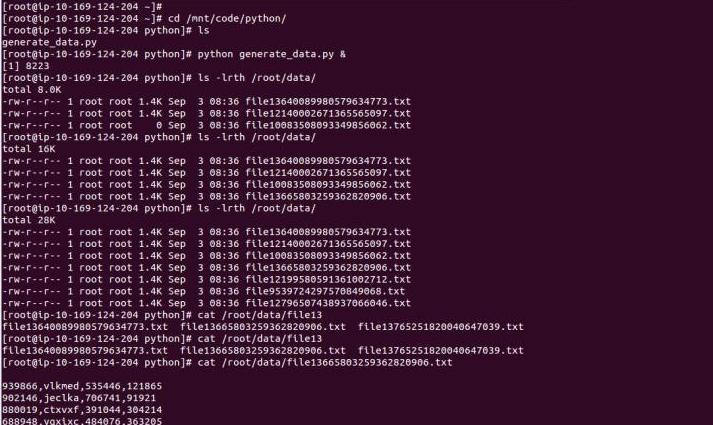
id = uuid.uuid4().int & (1<<64)-1

file = 'file'+str(id)+'.txt'

Generate\_events(50, '/root/data/'+file).start()

time.sleep(5)

1. create a folder call “data” in /root/ directory
2. Input Data set contains these columns “emp\_number,name,salary,bonus,filename”
3. you can run python file using this command “python generate\_data.py”
4. you can find result data in /root/data/ “ls -lrth /root/data/”



#### Step-3. How to Access, Initialize, Dependencies Spark Streaming on Spark shell:

**Access :**

You can access spark shell using below command on terminal

spark-shell for spark shell

spark-sql for spark sql



#### Dependencies for Spark Streaming :

Below listed imports are dependencies for Spark Streaming, please copy past into spark-shell terminal

import org.apache.hadoop.io.\_

import org.apache.hadoop.mapred.OutputFormat

import org.apache.spark.\_

import org.apache.spark.streaming.\_

import org.apache.spark.streaming.StreamingContext.\_

import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat



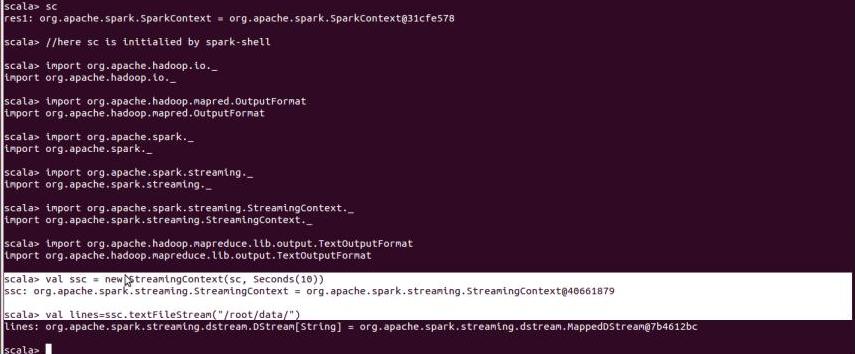
#### Initialize spark:

please copy paste below code into spark-shell

val ssc = new StreamingContext(sc, Seconds(10))

val lines=ssc.textFileStream(“/root/data/”)

here sc is sparkContext which has been initialized by spark-shell



#### Step-4.Files live stream example using Spark Streaming from Scala shell :

**on Spark shell:**

* please start data generate using this command  
   python generate\_data.py & (step-2)
* make a folder “mkdir /root/outputdata”
* Past below code on spark-shell.
* The below code compute the salary and bonus and create a employee monthly income

import org.apache.hadoop.io.\_

import org.apache.hadoop.mapred.OutputFormat

import org.apache.spark.\_

import org.apache.spark.streaming.\_

import org.apache.spark.streaming.StreamingContext.\_

import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat

val ssc = new StreamingContext(sc, Seconds(10))

val lines=ssc.textFileStream("file:///root/data/")

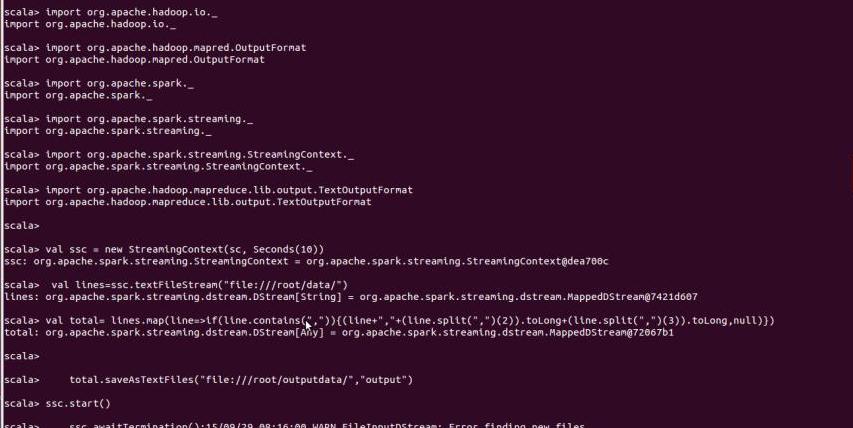
val total= lines.map(line=>if(line.contains(",")){(line+","+(line.split(",")(2)).toLong+(line.split(",")(3)).toLong,null)})

total.saveAsTextFiles("file:///root/outputdata/","output")

ssc.start()

ssc.awaitTermination();

* once “sss.start ” started on spark shell you can see result on “/root/outputdata”.



**NOTES:**

**Where you can find source code?:**

1. You can find data generator code in file name “generate\_data.py”
2. You can find Spark Streaming scala code here sparkstreaming.txt file

**Where you can find data?:**

1. You can find Input data in “/root/data/” all files contains data generated by “generate\_data.py” python code
2. You can find output results in “/root/outputdata/” all files contains data generated by Spark Streaming application