Some information on the setup

Start Hadoop

```
hduser@bd-1:/home/ubuntu$ $HADOOP_HOME/sbin/start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as hduser in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [bd-1]
Starting datanodes
Starting secondary namenodes [bd-1]
Starting resourcemanager
Starting nodemanagers
```

Check Java processes

- NameNode:

```
hduser@bd-1:/home/ubuntu$ $JAVA_HOME/bin/jps
13429 NameNode
14166 ResourceManager
14376 NodeManager
13931 SecondaryNameNode
14700 Jps
```

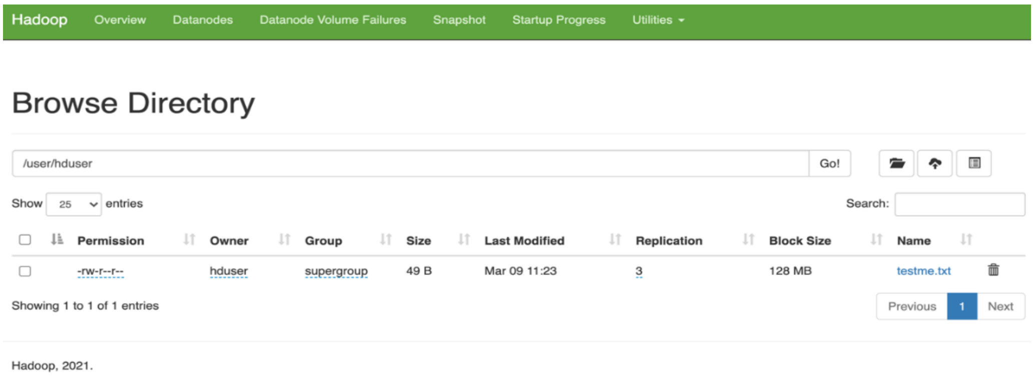
- DataNodes

```
hduser@bd-3:/home/ubuntu$ $JAVA_HOME/bin/jps
5321 Jps
5004 DataNode
5197 NodeManager

hduser@bd-4:/home/ubuntu$ $JAVA_HOME/bin/jps
5508 DataNode
5814 Jps
5687 NodeManager

hduser@bd-5:/home/ubuntu$ $JAVA_HOME/bin/jps
5185 NodeManager
5354 Jps
4991 DataNode
```

Load test file into HDFS

```
hdfs dfs -put ~/testme.txt /user/hduser
```

```
ubuntu@bd-1:~$ su - hduser
hduser@bd-1:~$ hdfs dfs -ls /user/hduser
Found 2 items
-rw-r--r--   3 hduser supergroup         49 2021-03-09 11:23 /user/hduser/testme.txt
drwxr-xr-x   - hduser supergroup          0 2021-03-11 07:25 /user/hduser/tmp
```



Browse Directory

/user/hduser

□	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
□	-rw-r--r--	hduser	supergroup	49 B	Mar 09 11:23	3	128 MB	testme.txt

Showing 1 to 1 of 1 entries

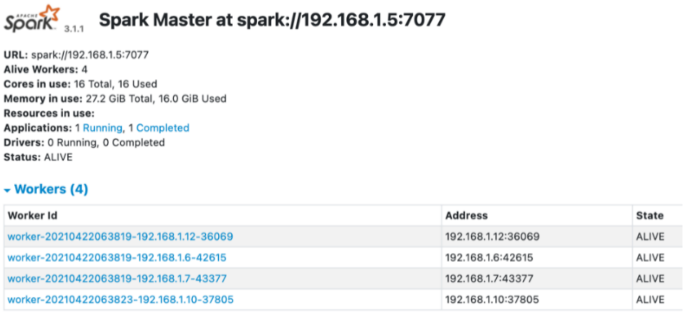
Hadoop, 2021.

Spark start Master:

```
hduser@bd-1:~$ $HADOOP_HOME/sbin/start-dfs.sh
Starting namenodes on [bd-1]
Starting datanodes
Starting secondary namenodes [bd-1]
hduser@bd-1:~$ $HADOOP_HOME/sbin/start-yarn.sh
Starting resourcemanager
Starting nodemanagers
hduser@bd-1:~$ $SPARK_HOME/sbin/start-master.sh
starting org.apache.spark.deploy.master.Master, logging to /opt/spark/logs/spark-
hduser-org.apache.spark.deploy.master.Master-1-bd-1.out
hduser@bd-1:~$ $JAVA_HOME/bin/jps
11745 Master
11170 ResourceManager
11380 NodeManager
10933 SecondaryNameNode
10631 DataNode
11801 Jps
10426 NameNode
```

Spark start Workers:

```
hduser@bd-1:~$ $SPARK_HOME/sbin/start-workers.sh

starting org.apache.spark.deploy.master.Master, logging to /opt/spark/logs/spark-
hduser-org.apache.spark.deploy.master.Master-1-bd-1.out
192.168.1.7: starting org.apache.spark.deploy.worker.Worker, logging to
/opt/spark/logs/spark-hduser-org.apache.spark.deploy.worker.Worker-1-bd-5.out
192.168.1.6: starting org.apache.spark.deploy.worker.Worker, logging to
/opt/spark/logs/spark-hduser-org.apache.spark.deploy.worker.Worker-1-bd-3.out
192.168.1.12: starting org.apache.spark.deploy.worker.Worker, logging to
/opt/spark/logs/spark-hduser-org.apache.spark.deploy.worker.Worker-1-bd-4.out
hduser@bd-1:~$
```



Spark Master at spark://192.168.1.5:7077

URL: spark://192.168.1.5:7077
Alive Workers: 4
Cores in use: 16 Total, 16 Used
Memory in use: 27.2 GiB Total, 16.0 GiB Used
Resources in use:
Applications: 1 Running, 1 Completed
Drivers: 0 Running, 0 Completed
Status: ALIVE

▾ Workers (4)

Worker Id	Address	State
worker-20210422063819-192.168.1.12-36069	192.168.1.12:36069	ALIVE
worker-20210422063819-192.168.1.6-42615	192.168.1.6:42615	ALIVE
worker-20210422063819-192.168.1.7-43377	192.168.1.7:43377	ALIVE
worker-20210422063823-192.168.1.10-37805	192.168.1.10:37805	ALIVE

Start Spark Shell

```
hduser@bd-1:/$ $SPARK_HOME/bin/spark-shell
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use
setLogLevel(newLevel).
Spark context Web UI available at http://bd-1:4040
Spark context available as 'sc' (master = local[*], app id = local-1617097320542).
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 3.1.1
      /_/

Using Scala version 2.12.10 (Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0_281)
Type in expressions to have them evaluated.
```

Start Pyspark

```
hduser@bd-1:/$ pyspark
Python 3.6.9 (default, Jan 26 2021, 15:33:00)
[GCC 8.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use
setLogLevel(newLevel).
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 3.1.1
      /_/

Using Python version 3.6.9 (default, Jan 26 2021 15:33:00)
Spark context Web UI available at http://bd-1:4040
Spark context available as 'sc' (master = local[*], app id = local-1617097231272).
SparkSession available as 'spark'.
```

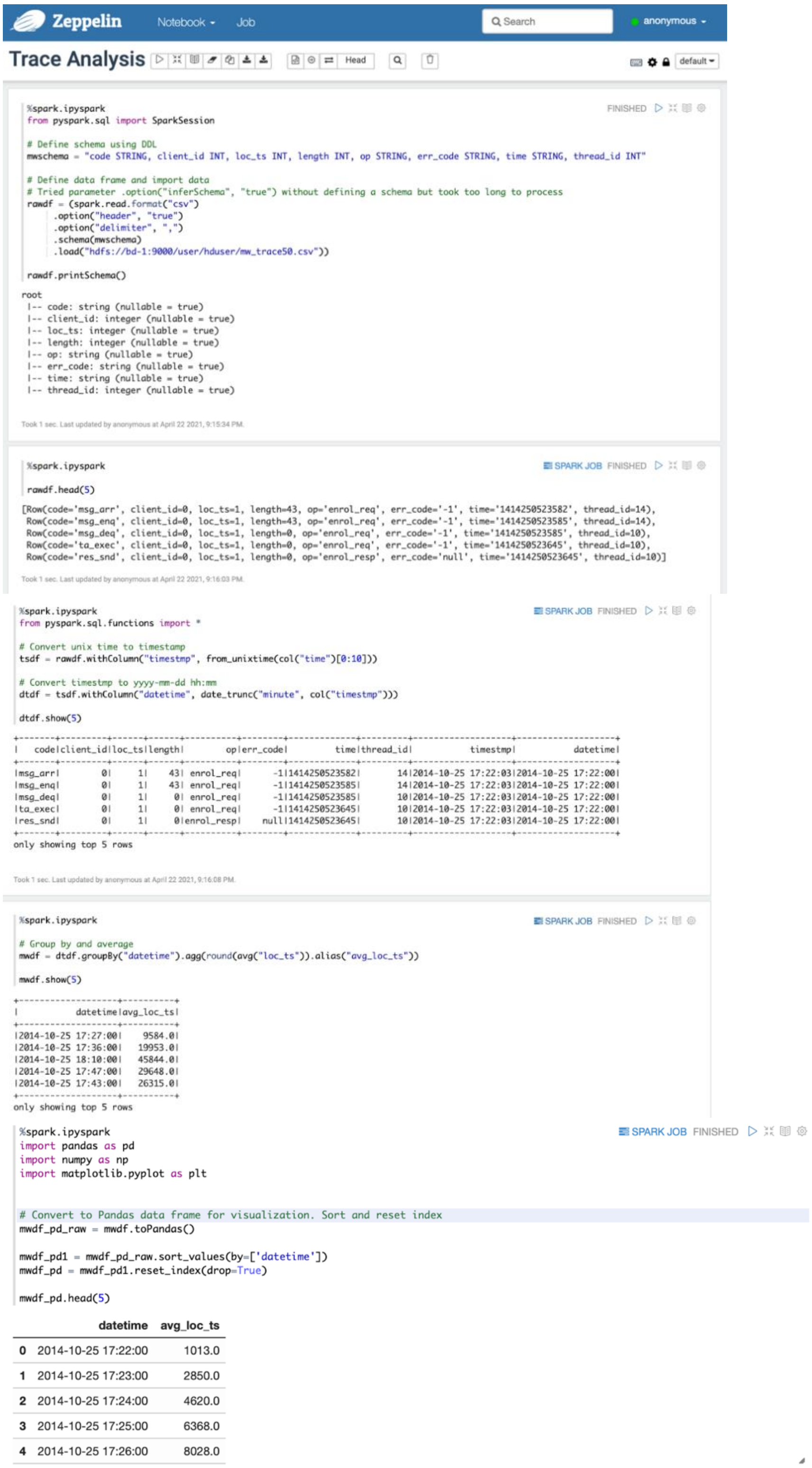
Load test file:

```
$SPARK_HOME/bin/spark-shell

scala> val lines = sc.textFile("hdfs:///user/hduser/testme.txt")
lines: org.apache.spark.rdd.RDD[String] = hdfs:///user/hduser/testme.txt MapPartitionsRDD[3] at textFile at <console>:24

scala> lines.count()
res1: Long = 10

scala> lines.first()
res2: String = heute

scala> print (lines.collect())
```

Zeppelin Start:

```
hduser@bd-2:~$ /opt/zeppelin/bin/zeppelin-daemon.sh start
Please specify HADOOP_CONF_DIR if USE_HADOOP is true
Zeppelin start                                             [  OK  ]

<property>
  <name>zeppelin.server.addr</name>
  <value>192.168.1.10</value>
  <description>Server binding address</description>
</property>
```

Simple analysis of a trace file



```
%spark.ipyspark
from pyspark.sql import SparkSession

# Define schema using DDL
mwschema = "code STRING, client_id INT, loc_ts INT, length INT, op STRING, err_code STRING, time STRING, thread_id INT"

# Define data frame and import data
# Tried parameter .option("inferSchema", "true") without defining a schema but took too long to process
rawdf = (spark.read.format("csv")
      .option("header", "true")
      .option("delimiter", ",")
      .schema(mwschema)
      .load("hdfs://bd-1:9000/user/hduser/mw_trace50.csv"))

rawdf.printSchema()
```

```
root
 |-- code: string (nullable = true)
 |-- client_id: integer (nullable = true)
 |-- loc_ts: integer (nullable = true)
 |-- length: integer (nullable = true)
 |-- op: string (nullable = true)
 |-- err_code: string (nullable = true)
 |-- time: string (nullable = true)
 |-- thread_id: integer (nullable = true)
```

Took 1 sec. Last updated by anonymous at April 22 2021, 9:15:34 PM.

```
%spark.ipyspark

rawdf.head(5)
```

```
[Row(code='msg_arr', client_id=0, loc_ts=1, length=43, op='enrol_req', err_code='-1', time='1414250523582', thread_id=14),
 Row(code='msg_enq', client_id=0, loc_ts=1, length=43, op='enrol_req', err_code='-1', time='1414250523585', thread_id=14),
 Row(code='msg_deq', client_id=0, loc_ts=1, length=0, op='enrol_req', err_code='-1', time='1414250523585', thread_id=10),
 Row(code='ta_exec', client_id=0, loc_ts=1, length=0, op='enrol_req', err_code='-1', time='1414250523645', thread_id=10),
 Row(code='res_snd', client_id=0, loc_ts=1, length=0, op='enrol_resp', err_code='null', time='1414250523645', thread_id=10)]
```

Took 1 sec. Last updated by anonymous at April 22 2021, 9:16:03 PM.

```
%spark.ipyspark
from pyspark.sql.functions import *

# Convert unix time to timestamp
tsdf = rawdf.withColumn("timestmp", from_unixtime(col("time")[0:10]))

# Convert timestmp to yyyy-mm-dd hh:mm
dtdf = tsdf.withColumn("datetime", date_trunc("minute", col("timestmp")))

dtdf.show(5)
```

```
+-------+---------+------+------+----------+--------+-------------+---------+-------------------+-------------------+
|   code|client_id|loc_ts|length|        op|err_code|         time|thread_id|           timestmp|           datetime|
+-------+---------+------+------+----------+--------+-------------+---------+-------------------+-------------------+
|msg_arr|        0|     1|    43| enrol_req|      -1|1414250523582|       14|2014-10-25 17:22:03|2014-10-25 17:22:00|
|msg_enq|        0|     1|    43| enrol_req|      -1|1414250523585|       14|2014-10-25 17:22:03|2014-10-25 17:22:00|
|msg_deq|        0|     1|     0| enrol_req|      -1|1414250523585|       10|2014-10-25 17:22:03|2014-10-25 17:22:00|
|ta_exec|        0|     1|     0| enrol_req|      -1|1414250523645|       10|2014-10-25 17:22:03|2014-10-25 17:22:00|
|res_snd|        0|     1|     0|enrol_resp|    null|1414250523645|       10|2014-10-25 17:22:03|2014-10-25 17:22:00|
+-------+---------+------+------+----------+--------+-------------+---------+-------------------+-------------------+
only showing top 5 rows
```

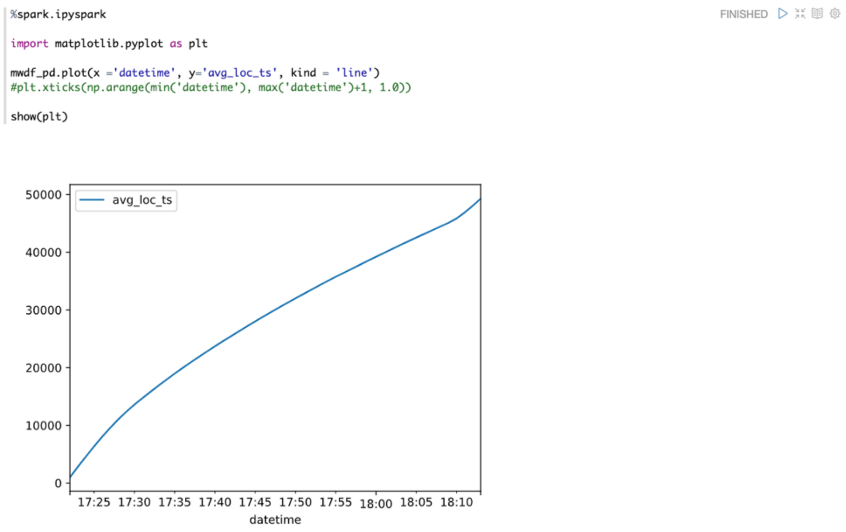
Took 1 sec. Last updated by anonymous at April 22 2021, 9:16:08 PM.

```
%spark.ipyspark

# Group by and average
mwdf = dtdf.groupBy("datetime").agg(round(avg("loc_ts")).alias("avg_loc_ts"))

mwdf.show(5)
```

```
+-------------------+----------+
|           datetime|avg_loc_ts|
+-------------------+----------+
|2014-10-25 17:27:00|    9584.0|
|2014-10-25 17:36:00|   19953.0|
|2014-10-25 18:10:00|   45844.0|
|2014-10-25 17:47:00|   29648.0|
|2014-10-25 17:43:00|   26315.0|
+-------------------+----------+
only showing top 5 rows
```

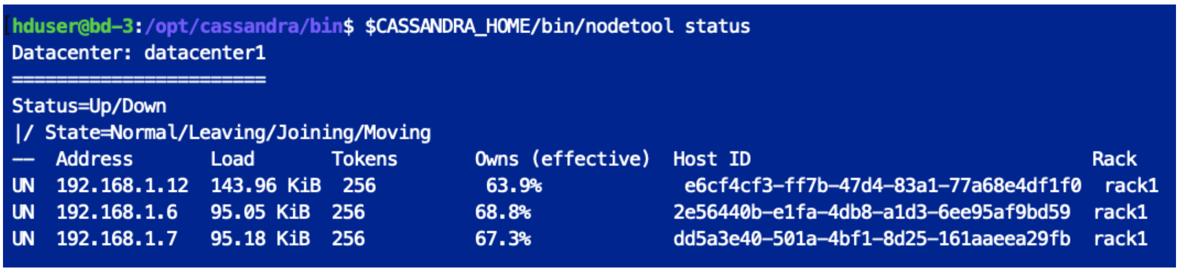
```
%spark.ipyspark
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt


# Convert to Pandas data frame for visualization. Sort and reset index
mwdf_pd_raw = mwdf.toPandas()

mwdf_pd1 = mwdf_pd_raw.sort_values(by=['datetime'])
mwdf_pd = mwdf_pd1.reset_index(drop=True)

mwdf_pd.head(5)
```

	datetime	avg_loc_ts
0	2014-10-25 17:22:00	1013.0
1	2014-10-25 17:23:00	2850.0
2	2014-10-25 17:24:00	4620.0
3	2014-10-25 17:25:00	6368.0
4	2014-10-25 17:26:00	8028.0



```
%spark.ipyspark

import matplotlib.pyplot as plt

mwdf_pd.plot(x ='datetime', y='avg_loc_ts', kind = 'line')
#plt.xticks(np.arange(min('datetime'), max('datetime')+1, 1.0))

show(plt)
```

Cassandra Start
$CASSANDRA_HOME/bin/cassandra



```
hduser@bd-3:/opt/cassandra/bin$ $CASSANDRA_HOME/bin/nodetool status
Datacenter: datacenter1
=======================
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
--  Address       Load        Tokens       Owns (effective)  Host ID                               Rack
UN  192.168.1.12  143.96 KiB  256          63.9%             e6cf4cf3-ff7b-47d4-83a1-77a68e4df1f0  rack1
UN  192.168.1.6   95.05 KiB   256          68.8%             2e56440b-e1fa-4db8-a1d3-6ee95af9bd59  rack1
UN  192.168.1.7   95.18 KiB   256          67.3%             dd5a3e40-501a-4bf1-8d25-161aaeea29fb  rack1
```

Cql-Shell Test
$CASSANDRA_HOME/bin/cqlsh bd-3 9042

```
hduser@bd-3:/opt/cassandra/conf$ $CASSANDRA_HOME/bin/cqlsh bd-3 9042
Connected to BD_Cluster at bd-3:9042.
[cqlsh 5.0.1 | Cassandra 3.11.10 | CQL spec 3.4.4 | Native protocol v4]
Use HELP for help.
cqlsh>
```