# Apache Spark Installation, Configuration and Administration on top of HDFS

## Preparation for Apache Spark Installation:

### Resources Details:

Out of the following hosts, we will use only selected two for our cluster configuration.

| SN | Hostname | FQDN | IP |
|---|---|---|---|
| 1 | bdrenfdludcf01 | bdrenfdludcf01.dle.asiaconnect.bdren.net.bd | 103.28.121.5 |
| 2 | bdrenfdludcf02 | bdrenfdludcf02.dle.asiaconnect.bdren.net.bd | 103.28.121.7 |
| 3 | bdrenfdludcf03 | bdrenfdludcf03.dle.asiaconnect.bdren.net.bd | 103.28.121.30 |
| 4 | bdrenfdludcf04 | bdrenfdludcf04.dle.asiaconnect.bdren.net.bd | 103.28.121.67 |
| 5 | bdrenfdludcf05 | bdrenfdludcf05.dle.asiaconnect.bdren.net.bd | 103.28.121.34 |
| 6 | bdrenfdludcf06 | bdrenfdludcf06.dle.asiaconnect.bdren.net.bd | 103.28.121.66 |

### Configure Hosts

Login into both hosts as hadoop user and add all hosts in /etc/hosts file. Other hosts are optional.

```
# vim /etc/hosts
103.28.121.5  bdrenfdludcf01  bdrenfdludcf01.dle.asiaconnect.bdren.net.bd
103.28.121.7  bdrenfdludcf02  bdrenfdludcf02.dle.asiaconnect.bdren.net.bd
103.28.121.30 bdrenfdludcf03 bdrenfdludcf03.dle.asiaconnect.bdren.net.bd
103.28.121.67 bdrenfdludcf04  bdrenfdludcf04.dle.asiaconnect.bdren.net.bd
103.28.121.34 bdrenfdludcf05  bdrenfdludcf05.dle.asiaconnect.bdren.net.bd
103.28.121.66 bdrenfdludcf06  bdrenfdludcf06.dle.asiaconnect.bdren.net.bd
```

Prefer if we login into all the machines using Putty or Any SSH Client.

### IP address check in your own hosts

```
#ip addr show

 1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state
UNKNOWN qlen 1
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
      valid_lft forever preferred_lft forever
   inet6 ::1/128 scope host
      valid_lft forever preferred_lft forever

2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
pfifo_fast state UP qlen 1000
   link/ether 08:00:27:b0:fe:53 brd ff:ff:ff:ff:ff:ff
   inet 192.168.0.104/24 brd 192.168.0.255 scope global dynamic enp0s3
      valid_lft 6925sec preferred_lft 6925sec
   inet6 fe80::7aed:2a0d:40d:bc24/64 scope link
      valid_lft forever preferred_lft forever
```

## CPU Check

```
# more /proc/cpuinfo | grep 'core id' | wc -l
```

The count will show you no of CPU.

**Important Note:** Please make sure you have at least two CPU cores assigned for master and worker node.

## Memory Check:

```
# more /proc/meminfo | grep -i 'Mem'
```

Please see the MemTotal

## Set All the Host Names (If require)

To set all the host names on a system, enter the following command as hadoop:

```
# hostnamectl set-hostname bdrenfdludcf01
```

*# hostnamectl set-hostname* bdrenfdludcf02

**Important Note:** We are now skipping all other steps followed just before main Hadoop Installation as it is already done e.g. Password less login, creating hadoop user etc.

## Installing Python 3.4

It is easy to install python using yum utility which we have installed as part of our Hadoop Installation. You would need to ensure internet connectivity from the operating host. Other you may need to download particular rpm package for Cent OS and then install using 'rpm -ivh' command.

*# sudo yum install python34-setuptools*

It would install various tools related to python in your operating system.

Now to be able to download and install python packages with dependencies resolution, we would need to install python installer named pip.

*# sudo yum install python3-pip*

After successful installation of pip, you would be able to download and install python packages.

e.g. pip3 install <python_package>

Installing python pandas used in data analysis.

*# sudo pip3 install pandas*

Installing python package to access HDFS.

*# sudo pip3 install hdfs*

Now we can check python version

*#python3.4 -V*

OR

*#python3.4 --version*

# Set Environment For Python 3 is required for hadoop user in .bashrc by adding green marked lines.

*#vim ~/.bashrc*

*# Set Environment For Python 3 in required user .bashrc*

*export PATH="/usr/bin:$PATH"*

*#Python for Spark*

```
export PYTHONPATH="/usr/lib/python3.4/site-packages:$PYTHONPATH"
export PYSPARK_PYTHON=/usr/bin/python3.4

# User specific aliases and functions
alias python=/usr/bin/python3.4
```

Reload the configuration for the hadoop user

```
#source ~/.bashrc
```

## Apache Spark Installation, Configuration & Administration:

### Downloading the Spark

Download Apache Spark and Place in following directory for all of our machines planned to work either as Master or Worker.

```
# cd /downloads
[hadoop@bdrenfdludcf01downloads]# ls -l
total 537576
-rw-r--r--. 1 hadoop hadoop 210606807 Apr  8 13:05 hadoop-2.7.1.tar.gz
drwxr-xr-x. 8   10  143       233 Apr 11  2015 jdk1.7.0_79
-rw-r--r--. 1 hadoop hadoop 153512879 Apr  8 13:05 jdk-7u79-linux-x64.tar.gz
-rw-r--r--  1 hadoop hadoop 186354175 Apr 29 21:44 spark-2.0.0-bin-hadoop2.7.tar
# sudo tar xvf spark-2.0.0-bin-hadoop2.7.tar
[hadoop@bdrenfdludcf01downloads]# sudo mv spark-2.0.0-bin-hadoop2.7 spark
[hadoop@bdrenfdludcf01downloads]# sudo mv spark /usr/local/
```

### Configuration in spark-env.sh

Creating development directory and provide access to spark process on that. In parallel, we are creating a JARS directory for keeping all external libraries or packages.

```
[hadoop@bdrenfdludcf02 ~]$ mkdir -p /home/hadoop/development
[hadoop@bdrenfdludcf02 ~]$ chmod g+s /home/hadoop/
[hadoop@bdrenfdludcf02 ~]$ mkdir -p /home/hadoop/.ivy2/jars
```

Setting Up Spark Environment

```
[hadoop@bdrenfdludcf01downloads]# sudo cp /usr/local/spark/conf/spark-env.sh.template /usr/local/spark/conf/spark-env.sh
```

Create /usr/local/spark/conf/spark-env.sh and add below lines to the file

```
#sudo vim /usr/local/spark/conf/spark-env.sh
```

```
export JAVA_HOME=/usr/local/jdk1.7.0_79
SPARK_MASTER_WEBUI_PORT=9999
SPARK_JAVA_OPTS=-Dspark.driver.port=53411
HADOOP_HOME=/usr/local/hadoop
HADOOP_CONF_DIR=$HADOOP_HOME/conf
SPARK_MASTER_IP=bdrenfdludcf01
#Python for Spark
export PYTHONPATH="/usr/lib/python3.4/site-packages/:$PYTHONPATH"
export PYSPARK_PYTHON=/usr/bin/python3.4
export SPARK_CLASSPATH=/home/hadoop/.ivy2/jars:/home/hadoop/development:$SPARK_CLASSPATH
```

## Configuration in spark-defaults.conf

Create /usr/local/spark/conf/spark-defaults.conf for all hosts and add below lines to the file.

```
[hadoop@bdrenfdludcf01 downloads]$ cp spark-defaults.conf.template spark-defaults.conf
[hadoop@bdrenfdludcf01 downlaods]$ sudo vim /usr/local/spark/conf/spark-defaults.conf
```

```
spark.master          spark:// bdrenfdludcf01:7077
spark.serializer       org.apache.spark.serializer.KryoSerializer
#spark.driver.memory 256m
#spark.executor.memory 256m
#spark.driver.cores 1
#spark.executor.cores 1
spark.executorEnv.PYTHONHASHSEED 321
```

Important Note: Please look into following link for more details: https://spark.apache.org/docs/latest/configuration.html

## Define Worker Nodes

Append hostnames of all the slave/worker nodes in /usr/local/spark/conf/slaves file.

```
[hadoop@bdrenfdludcf01downloads]#sudo  cp /usr/local/spark/conf/slaves.template /usr/local/spark/conf/slaves

[hadoop@bdrenfdludcf01downloads]# sudo vim /usr/local/spark/conf/slaves
```

```
bdrenfdludcf01
bdrenfdludcf02
#Please don't add your master if you don't have multiple CPU in master.
```

## Creating Spark Events

Create /tmp/spark-events for all the master and worker nodes.

```
# sudo mkdir –p /tmp/spark-events
# sudo chmod 777 /tmp/spark-events
```

## Starting or Stopping Spark

Start/Stop Spark using below commands using hadoop user

```
# sudo chown hadoop:wheel -R /usr/local/spark
```

To Start Spark Cluster:

```
# /usr/local/spark/sbin/start-all.sh
```

To Stop Spark Cluster:

```
# /usr/local/spark/sbin/stop-all.sh
```

Let’s start journey with Apache Spark with Python:

```
$ /usr/local/spark/bin/pyspark
```

You can access SPARK UI in Browser by below URL

Spark Master URL: links http://bdrenfdludcf01:9999/

If you would need to start/stop your HDFS then please run following command:

```
$HADOOP_HOME/sbin/start-dfs.sh
$HADOOP_HOME/sbin/stop-dfs.sh
```

Check JPS command to see all HDFS and Spark processes are running.

```
[hadoop@bdrenfdludcf01 ~]$ jps
2048 DataNode
1942 NameNode
2525 Worker
2457 Master
2248 SecondaryNameNode
3304 Jps
[hadoop@bdrenfdludcf01 ~]$
```

## Monitor Spark Applications Running

Per spark-submit instance it will create a URL for review, and you can find incremental port number to see subsequent concurrent spark applications.

http://bdrenfdludcf01:4040/jobs/

There are several ways to monitor Spark applications: web UIs, metrics, and external instrumentation.

**Important Note:** Please check the port number carefully before putting into browser given by the spark. It can be even 4041, 4042 etc.

Web Interfaces

Every SparkContext launches a web UI, by default on port 4040, that displays useful information about the application. This includes:

- A list of scheduler stages and tasks
- A summary of RDD sizes and memory usage
- Environmental information.
- Information about the running executors

You can access this interface by simply opening http://<driver-node>:4040 in a web browser. If multiple SparkContexts are running on the same host, they will bind to successive ports beginning with 4040 (4041, 4042, etc).

Note that this information is only available for the duration of the application by default. To view the web UI after the fact, set spark.eventLog.enabled to true before starting the application. This configures Spark to log Spark events that encode the information displayed in the UI to persisted storage.

Let’s do some exercise on Spark today:

```
[hadoop@bdrenfdludcf01 ~]$ pyspark

Python 3.4.10 (default, Oct  4 2019, 19:14:13)

[GCC 4.8.5 20150623 (Red Hat 4.8.5-39)] on linux

Type "help", "copyright", "credits" or "license" for more information.

Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties

Setting default log level to "WARN".

To adjust logging level use sc.setLogLevel(newLevel).

20/04/18 10:18:41 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

20/04/18 10:18:41 WARN SparkConf:
```

```
SPARK_JAVA_OPTS was detected (set to '-Dspark.driver.port=53411').
This is deprecated in Spark 1.0+.

Please instead use:
 - ./spark-submit with conf/spark-defaults.conf to set defaults for an application
 - ./spark-submit with --driver-java-options to set -X options for a driver
 - spark.executor.extraJavaOptions to set -X options for executors
 - SPARK_DAEMON_JAVA_OPTS to set java options for standalone daemons (master or worker)

20/04/18 10:18:41 WARN SparkConf: Setting 'spark.executor.extraJavaOptions' to '-Dspark.driver.port=53411' as a work-around.
20/04/18 10:18:41 WARN SparkConf: Setting 'spark.driver.extraJavaOptions' to '-Dspark.driver.port=53411' as a work-around.
20/04/18 10:18:41 WARN SparkConf:
SPARK_CLASSPATH was detected (set to '/home/hadoop/.ivy2/jars:/home/hadoop/development:').
This is deprecated in Spark 1.0+.

Please instead use:
 - ./spark-submit with --driver-class-path to augment the driver classpath
 - spark.executor.extraClassPath to augment the executor classpath

20/04/18 10:18:41 WARN SparkConf: Setting 'spark.executor.extraClassPath' to '/home/hadoop/.ivy2/jars:/home/hadoop/development:' as a work-around.
```

```
20/04/18 10:18:41 WARN SparkConf: Setting 'spark.driver.extraClassPath'
to '/home/hadoop/.ivy2/jars:/home/hadoop/development:' as a work-
around.

Welcome to

      ____              __

     / __/__  ___ _____/ /__

    _\ \/ _ \/ _ `/ __/  '_/

   /__ / .__/\_,_/_/ /_/\_\   version 2.0.0

      /_/


Using Python version 3.4.10 (default, Oct  4 2019 19:14:13)

SparkSession available as 'spark'.

>>> from pyspark.sql import SQLContext

>>> sqlContext = SQLContext(sc)

>>>
df=sqlContext.read.format("csv").option("header",'true').load("hdfs://bdr
enfdludcf01:9000/mydir/trips.csv")

>>>
df_trips=sqlContext.read.format("csv").option("header",'true').load("hdfs:
//bdrenfdludcf01:9000/mydir/trips.csv")

>>>
df_users=sqlContext.read.format("csv").option("header",'true').load("hdfs
://bdrenfdludcf01:9000/mydir/users.csv")

>>> df_trips.show()

+------+---------+--------+--------+--------+--------+-----------+

|tripid| tripdest|trippick|triptime|tripcost|driverid|passengerid|

+------+---------+--------+--------+--------+--------+-----------+

|  2000|   adabor|  banani|      60|     300|    1025|       1030|

|  2001|dhanmondi|  adabor|      25|     100|    1026|       1029|
|  2002|  gulshan|  banani|      20|      90|    1028|       1024|
|  2003|old dhaka| gulshan|      70|     400|    1026|       1027|
+------+---------+--------+--------+--------+--------+-----------+
```

```
>>> df_users.show()
+------+--------+-----------+---------+
|userid|username|     mobile|     role|
+------+--------+-----------+---------+
|  1024|  sultan|01815818277|passenger|
|  1025|  shimul|01915818277|   driver|
|  1026|   ratan|01715818277|   driver|
|  1027|    babu|01515818277|passenger|
|  1028|   titon|01615818277|   driver|
|  1029|  zobair|01815818299|passenger|
|  1030| amitava|01815818233|passenger|
+------+--------+-----------+---------+
```

```
>>> join = df_users.join(df_trips, df_users.userid == df_trips.passengerid,
"leftouter")
>>> join.show()
+------+--------+-----------+---------+------+---------+--------+--------+--------+--------
+-----------+
|userid|username|     mobile|     role|tripid|
tripdest|trippick|triptime|tripcost|driverid|passengerid|
+------+--------+-----------+---------+------+---------+--------+--------+--------+--------
+-----------+
|  1024|  sultan|01815818277|passenger|  2002|  gulshan|  banani|
20|      90|    1028|       1024|
|  1025|  shimul|01915818277|   driver|  null|     null|    null|    null|
null|    null|       null|
|  1026|   ratan|01715818277|   driver|  null|     null|    null|    null|
null|    null|       null|
|  1027|    babu|01515818277|passenger|  2003|old dhaka| gulshan|
70|     400|    1026|       1027|
|  1028|   titon|01615818277|   driver|  null|     null|    null|    null|
null|    null|       null|
|  1029|  zobair|01815818299|passenger|  2001|dhanmondi|  adabor|
25|     100|    1026|       1029|
|  1030| amitava|01815818233|passenger|  2000|   adabor|  banani|
60|     300|    1025|       1030|
+------+--------+-----------+---------+------+---------+--------+--------+--------+--------
+-----------+
```

```
>>> join.show()
+------+--------+-----------+---------+------+---------+--------+--------+--------+--------
+-----------+
|userid|username|     mobile|     role|tripid|
tripdest|trippick|triptime|tripcost|driverid|passengerid|
+------+--------+-----------+---------+------+---------+--------+--------+--------+--------
+-----------+
|  1024|  sultan|01815818277|passenger|  2002|  gulshan|  banani|
20|      90|    1028|       1024|
|  1025|  shimul|01915818277|   driver|  null|     null|    null|    null|
null|    null|       null|
|  1026|   ratan|01715818277|   driver|  null|     null|    null|    null|
null|    null|       null|
|  1027|    babu|01515818277|passenger|  2003|old dhaka| gulshan|
70|     400|    1026|       1027|
|  1028|   titon|01615818277|   driver|  null|     null|    null|    null|
null|    null|       null|
|  1029|  zobair|01815818299|passenger|  2001|dhanmondi|  adabor|
25|     100|    1026|        1029|

|  1030| amitava|01815818233|passenger|  2000|   adabor|  banani|
60|     300|    1025|        1030|

+------+--------+-----------+---------+------+---------+--------+--------+--------+--------
+-----------+
```

```
>>> join.filter(join['passengerid'] > 1024).show()

+------+--------+-----------+---------+------+---------+--------+--------+--------+--------
+-----------+

|userid|username|     mobile|     role|tripid|
tripdest|trippick|triptime|tripcost|driverid|passengerid|

+------+--------+-----------+---------+------+---------+--------+--------+--------+--------
+-----------+

|  1027|    babu|01515818277|passenger|  2003|old dhaka| gulshan|
70|     400|    1026|        1027|

|  1029|  zobair|01815818299|passenger|  2001|dhanmondi|  adabor|
25|     100|    1026|        1029|

|  1030| amitava|01815818233|passenger|  2000|   adabor|  banani|
60|     300|    1025|        1030|

+------+--------+-----------+---------+------+---------+--------+--------+--------+--------
+-----------+
```

```
>>> join.groupBy("passengerid").count().show()

+-----------+-----+

|passengerid|count|

+-----------+-----+

|       null|    3|

|       1030|    1|

|       1027|    1|
|      1024|    1|
|      1029|    1|
+-----------+-----+
```

```
>>> join.write.csv("join.csv")
>>> join.show()
```

```
+------+--------+-----------+---------+------+---------+--------+--------+--------+--------
+-----------+
|userid|username|     mobile|     role|tripid|
tripdest|trippick|triptime|tripcost|driverid|passengerid|
+------+--------+-----------+---------+------+---------+--------+--------+--------+--------
+-----------+
|  1024|  sultan|01815818277|passenger|  2002|  gulshan|  banani|
20|      90|    1028|       1024|
|  1025|  shimul|01915818277|   driver|  null|     null|    null|    null|
null|    null|       null|
|  1026|   ratan|01715818277|   driver|  null|     null|    null|    null|
null|    null|       null|
|  1027|    babu|01515818277|passenger|  2003|old dhaka| gulshan|
70|     400|    1026|       1027|
|  1028|   titon|01615818277|   driver|  null|     null|    null|    null|
null|    null|       null|
|  1029|  zobair|01815818299|passenger|  2001|dhanmondi|  adabor|
25|     100|    1026|       1029|
|  1030| amitava|01815818233|passenger|  2000|   adabor|  banani|
60|     300|    1025|       1030|
+------+--------+-----------+---------+------+---------+--------+--------+--------+--------
+-----------+
```

```
>>> join.select("userid","mobile").show()
```

```
+------+-----------+
|userid|     mobile|
+------+-----------+
|  1024|01815818277|
|  1025|01915818277|
|  1026|01715818277|
|  1027|01515818277|
|  1028|01615818277|
|  1029|01815818299|
|  1030|01815818233|
+------+-----------+
>>> df.createOrReplaceTempView("trips")
>>> sqlDF = sqlContext.sql("SELECT tripid, tripdest FROM trips")
>>> sqlDF.show()
+------+---------+
|tripid| tripdest|
+------+---------+
|  2000|   adabor|
|  2001|dhanmondi|
|  2002|  gulshan|
|  2003|old dhaka|
+------+---------+
>>>
# Let us do some RDD operation
>>> numbers = sc.parallelize([14,21,88,99,455])
>>> log_values = numbers.map(lambda n : math.log10(n))
>>> log_values.collect()
```

```
[1.146128035678238, 1.3222192947339193, 1.9444826721501687,
1.99563519459755, 2.6580113966571126]

>>> numbers = sc.parallelize([1,2,3,4,5,6,7,8,9,10,11,12,13,14,15],5)

>>> numbers.collect()

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]

>>> numbers = sc.parallelize([1,2,3,4,5,6,7,8,9,10,11,12,13,14,15],7)

>>> numbers.collect()

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]

>>> counts =
sc.textFile("hdfs://bdrenfdludcf01:9000/mydir/names").flatMap(lambda
line: line.split(" ")).map(lambda word: (word, 1)).reduceByKey(lambda a,
b: a + b)

>>> counts.saveAsTextFile("/home/hadoop/development/output5")
```

Let us see the output5 folder

```
>>> counts.collect()

[('Orange', 2), ('Pam', 1), ('I', 1), ('Mango', 3), ('fruit', 2), ('love', 1),
('Guava', 1), ('Apple', 3), ('Jack', 1)]

>>> df=counts.toDF()

>>> df.show()

+------+---+
|    _1| _2|
+------+---+
|Orange|  2|
|   Pam|  1|
|     I|  1|
| Mango|  3|
| fruit|  2|
|  love|  1|
| Guava|  1|
| Apple|  3|
|  Jack|  1|
+------+---+
>>> df.coalesce(1).write.csv("/home/hadoop/development/newoutput.csv")
>>>exit()
```

**Important Note:** More dataframe programming examples are found at
https://spark.apache.org/docs/2.3.0/sql-programming-guide.html
https://spark.apache.org/docs/2.1.0/api/python/pyspark.html