Automated Setup & Installation Guide

for

Spark & Kafka

Single Node Cluster Environment

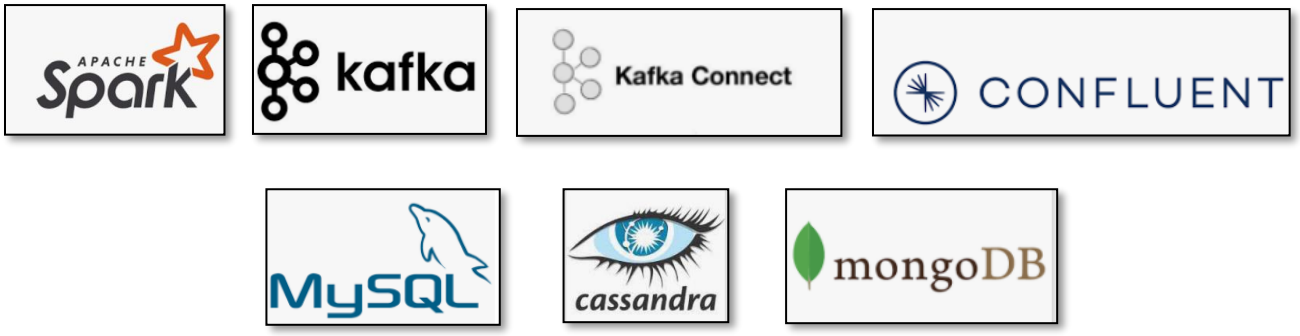
(Pseudo Distributed mode)

using light-weight script

with

MySQL/Cassandra/MongoDB/Confluent

Version :- 2021V1



Developed & Tested

by

RAJU CHAL

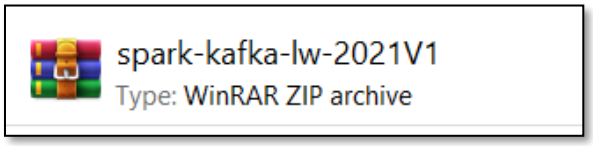
LKM , Accenture- ATCI

Contents :-

Context

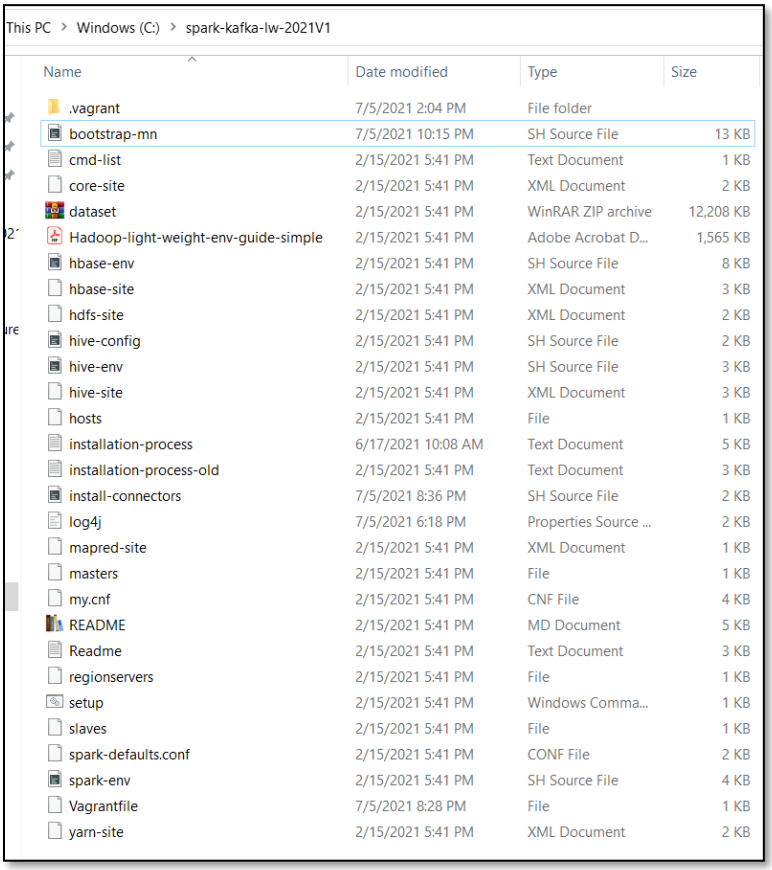
We will be using automated script for installation & configurations of "`Spark/Kafka Single Node Cluster`" on Laptop /Desktop using light-weight script shared with you .

Script:-



File Name :- `spark-kafka-lw-2021V1.zip`

Contents of script :-



Software with version to be installed

Software	Version
Operating System	Ubuntu Linux
Spark	3.0.2
Sbt	1.2.0
Cassandra	3.11.10
MongoDB	4.2.13
Kafka	2.8.0
Scala	2.12.2
JDK	8u131
MySQL	5.7
MySQL JDBC	5.1.47 & 8.0.25
Python	3.6
Confluent Community Edition	6.2.0
Confluent Kafka Datagen Connector	Latest
Confluent Kafka JDBC Source/Sink Connector	10.2.0
Confluent Kafka HDFS3 Sink Connector	1.1.1
Confluent Kafka MySQL Debezium Source Connector	1.5.0
Confluent Kafka Cassandra Sink Connector	2.0.0
Confluent Kafka MongoDB Source/Sink Connector	1.5.1

Download & Install the pre-requisite software

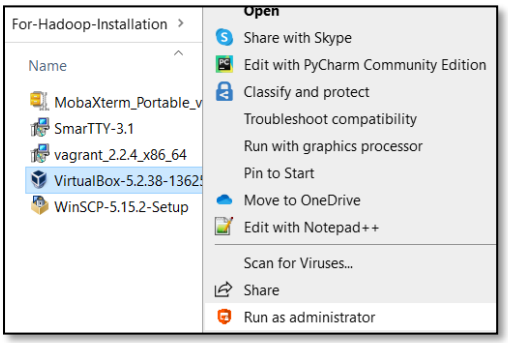
Pre-requisite:-

- **During entire installation procedure your Laptop/Desktop should be connected with Internet.**
- **Minimum RAM required:- 8 GB**

1) Download and Install Oracle Virtual Box

https://download.virtualbox.org/virtualbox/5.2.38/VirtualBox-5.2.38-136252-Win.exe

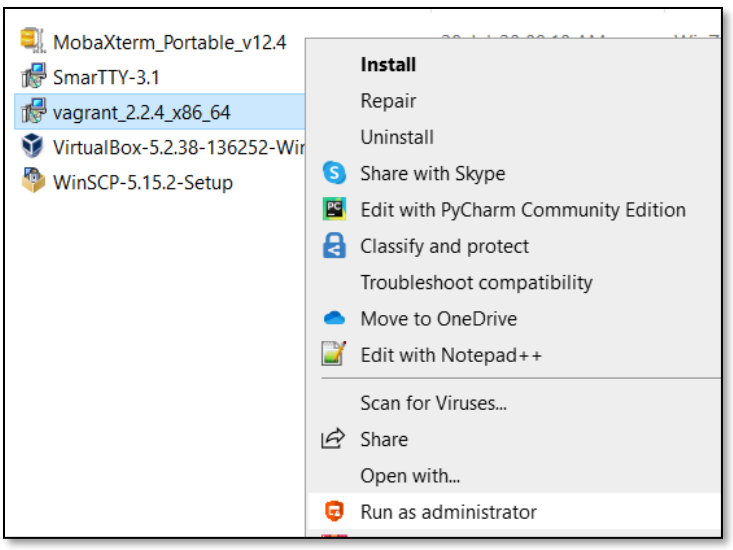
Right click on downloaded software → click on "Run as administrator"



2) Download and Install Vagrant version 2.2.4

https://releases.hashicorp.com/vagrant/2.2.4/vagrant_2.2.4_x86_64.msi

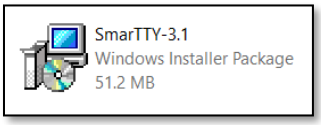
Right click on downloaded software → click on "Run as administrator"



After installation "RESTART" the system

3) Download SmarTTY

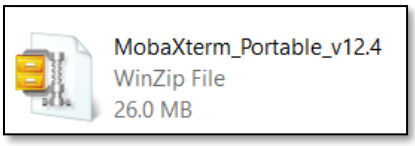
http://sysprogs.com/getfile/409/SmarTTY-3.1.msi



OR

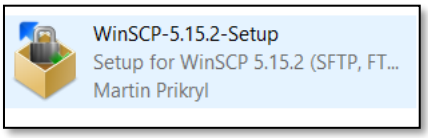
Download MobaXTerm

https://download.mobatek.net/2012020021813110/MobaXterm_Portable_v20.1.zip



4) Download WinSCP

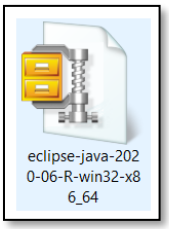
https://winscp.net/eng/download.php



5) Eclipse Download (OPTIONAL)

https://ftp.yz.yamagata-u.ac.jp/pub/eclipse//technology/epp/downloads/release/2020-06/R/eclipse-java-2020-06-R-win32-x86_64.zip

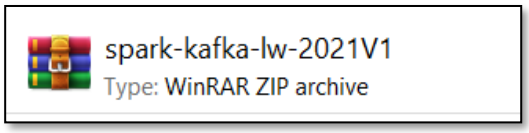
unzip and run it



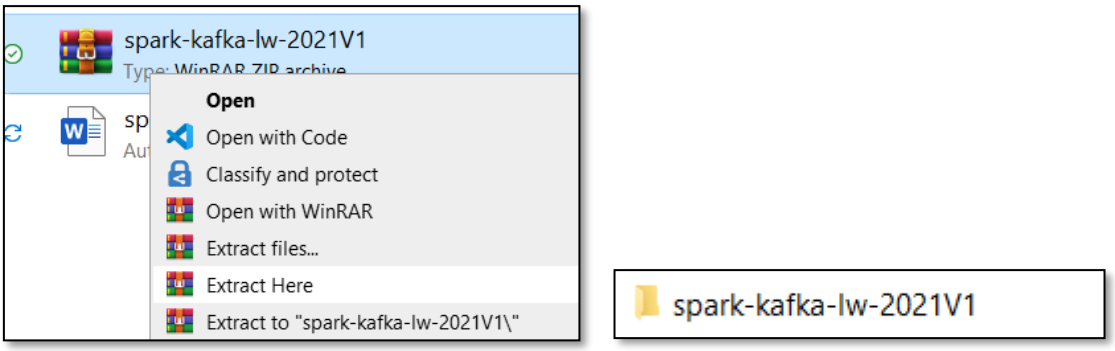
Note :- You can download latest version of **WinSCP, SmarTTY, MobaXTerm & Eclipse (Optional)**

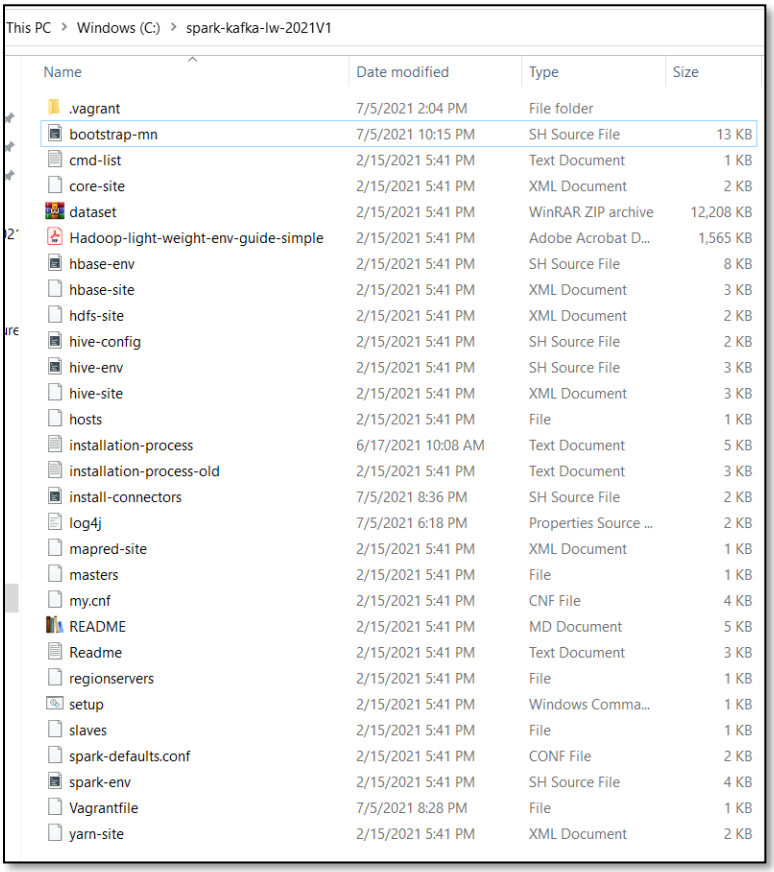
Installation Process

1. Download the shared zip file - **`spark-kafka-lw-2021V1.zip`**

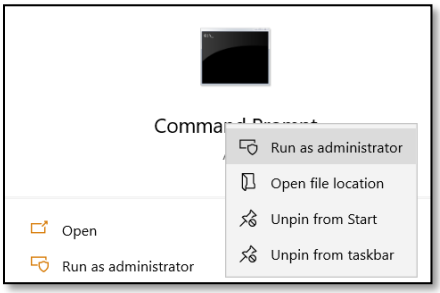


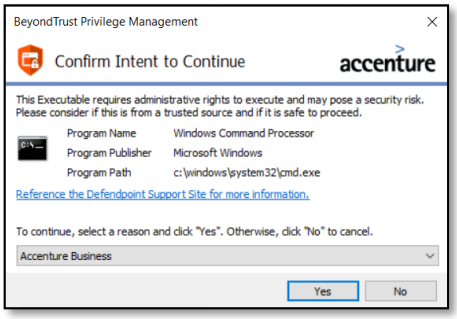
2. Unzip it → Right click on the ZIP file →Click on "Extract Here" → copy the extracted root folder to C-Drive



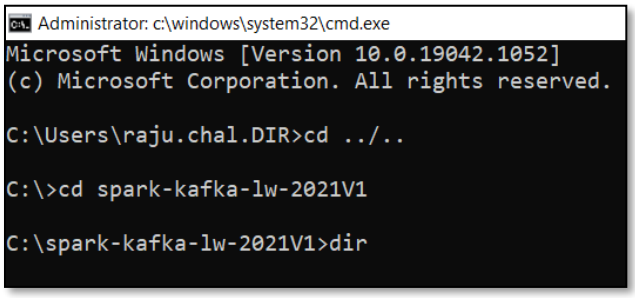


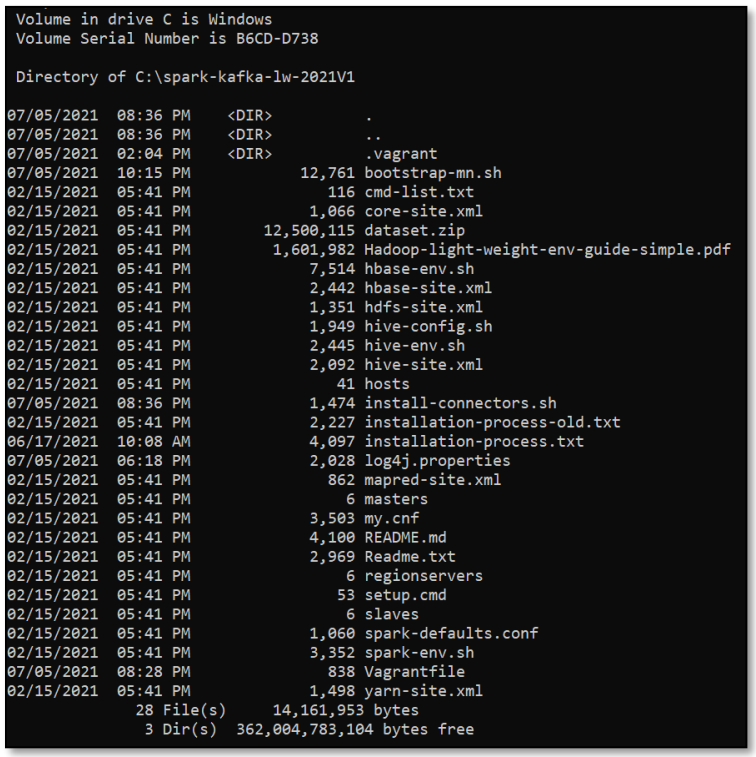
3. Open **command prompt** of Windows in **Administrator** mode





4. Change the directory to the extracted folder **spark-kafka-lw-2021V1 → run "setup.cmd" command**





C:\spark-kafka-lw-2021V1>setup.cmd

```
C:\spark-kafka-lw-2021V1>setup.cmd

C:\spark-kafka-lw-2021V1>vagrant box add ubuntu/trusty64 --insecure
==> box: Loading metadata for box 'ubuntu/trusty64'
    box: URL: https://vagrantcloud.com/ubuntu/trusty64
==> box: Adding box 'ubuntu/trusty64' (v20190514.0.0) for provider: virtualbox
The box you're attempting to add already exists. Remove it before
adding it again or add it with the `--force` flag.

Name: ubuntu/trusty64
Provider: virtualbox
Version: 20190514.0.0

C:\spark-kafka-lw-2021V1>vagrant up
Bringing machine 'KafkaMaster' up with 'virtualbox' provider...
==> KafkaMaster: Importing base box 'ubuntu/trusty64'...
==> KafkaMaster: Matching MAC address for NAT networking...
_
```

```
KafkaMaster: starting org.apache.spark.deploy.master.Master, logging to /home/vagrant/bigdata/spark/logs/spark-vagrant-org.apache.spark.deploy.master.Master-1-master.out
KafkaMaster: master: starting org.apache.spark.deploy.worker.Worker, logging to /home/vagrant/bigdata/spark/logs/spark-vagrant-org.apache.spark.deploy.worker.Worker-1-master.out
KafkaMaster: 20130 Jps
KafkaMaster: 20087 Worker
KafkaMaster: 19960 Master
KafkaMaster: ------------RUNNING SPARK EXAMPLE ------------------------------
KafkaMaster: Pi is roughly 3.1384791384791386
KafkaMaster:  Stoping Spark Services
KafkaMaster: master: stopping org.apache.spark.deploy.worker.Worker
KafkaMaster: stopping org.apache.spark.deploy.master.Master
KafkaMaster:  Starting Confluent Services
KafkaMaster: The local commands are intended for a single-node development environment only,
```

Wait till you get back the Command Prompt [**`c:\spark-kafka-lw-2021V1>`**]

Depending on the bandwidth total installation may take 45 mins to 1 hr time

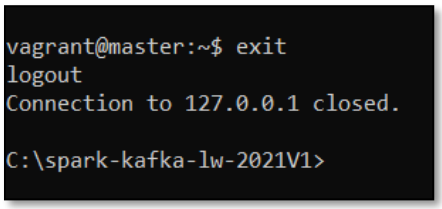
```
    KafkaMaster: |
    KafkaMaster: /
    KafkaMaster: Kafka is [DOWN]
    KafkaMaster: Stopping ZooKeeper
    KafkaMaster: |
    KafkaMaster: ZooKeeper is [DOWN]
    KafkaMaster:  Your Kafka environment is ready

C:\spark-kafka-lw-2021V1>
```

5. After getting back the Command Prompt type "**vagrant ssh**" to login to Linux Box

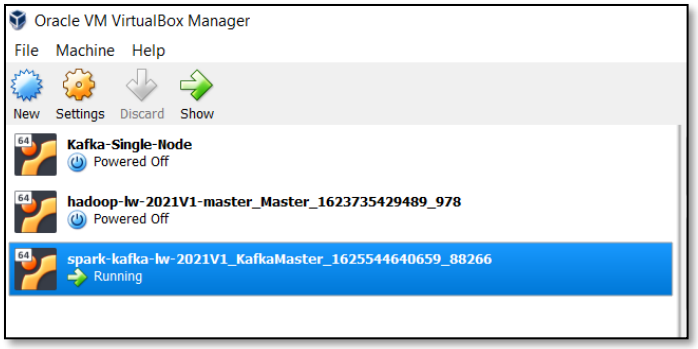
   `C:\spark-kafka-lw-2021V1>vagrant ssh`

```
C:\spark-kafka-lw-2021V1>vagrant ssh
Welcome to Ubuntu 14.04.6 LTS (GNU/Linux 3.13.0-170-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

  System information as of Tue Jul  6 04:42:29 UTC 2021

  System load:  0.03              Processes:           89
  Usage of /:   9.6% of 39.34GB   Users logged in:     1
  Memory usage: 6%                IP address for eth0: 10.0.2.15
  Swap usage:   0%                IP address for eth1: 192.168.56.70

  Graph this data and manage this system at:
    https://landscape.canonical.com/

New release '16.04.7 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


Last login: Tue Jul  6 04:42:29 2021 from 192.168.56.1
vagrant@master:~$
```

To exit from the Linux Prompt , type "exit"

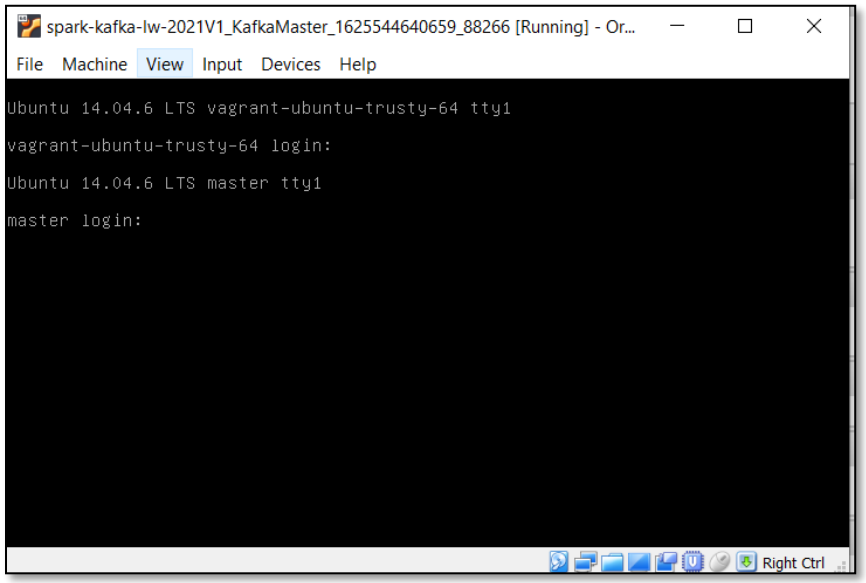


6. Open the **Oracle VirtualBox** that you have already installed, you will observe one Linux machine is running as shown below



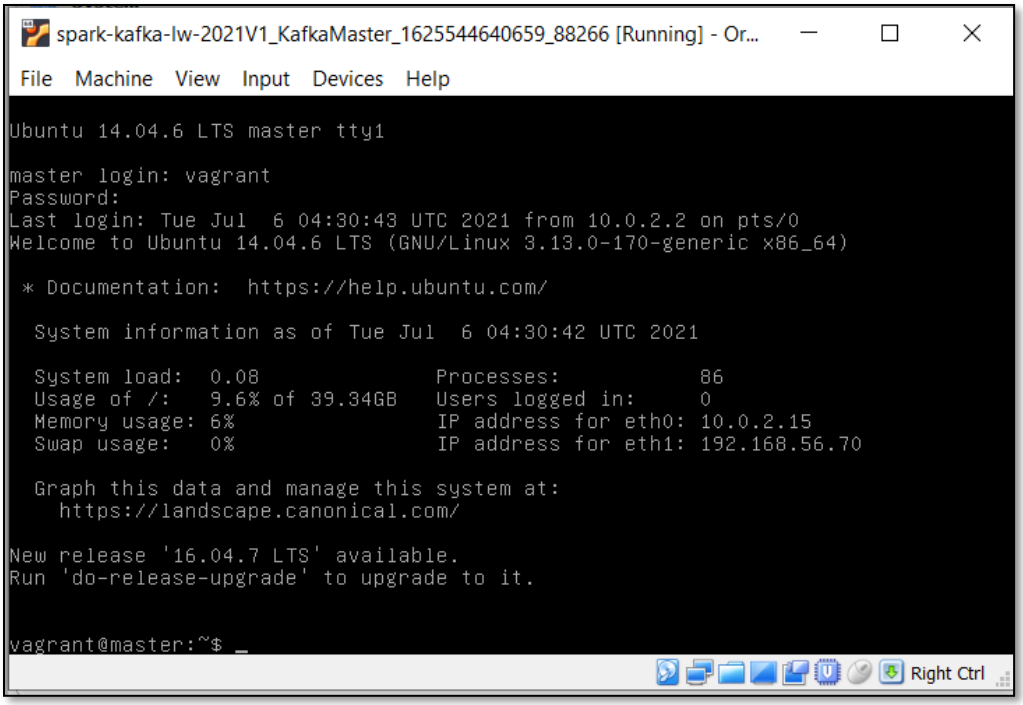
Note :- If it is not able to start, then - > You need to **enable Virtualization** on your laptop/desktop to create a virtualized environment on your desktop. The steps for the same depend on your laptop/desktop model. You should take help from Tech Support

7. Select the Linux box and click on the **Show** button in the toolbar, you will be getting the following screen



Login User Name – vagrant

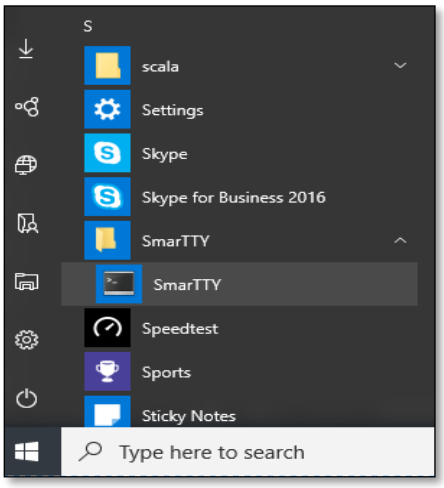
Password - vagrant


spark-kafka-lw-2021V1_KafkaMaster_1625544640659_88266 [Running] - Or...
File Machine View Input Devices Help
Ubuntu 14.04.6 LTS master tty1
master login: vagrant
Password:
Last login: Tue Jul 6 04:30:43 UTC 2021 from 10.0.2.2 on pts/0
Welcome to Ubuntu 14.04.6 LTS (GNU/Linux 3.13.0-170-generic x86_64)
* Documentation: https://help.ubuntu.com/
System information as of Tue Jul 6 04:30:42 UTC 2021
System load: 0.08 Processes: 86
Usage of /: 9.6% of 39.34GB Users logged in: 0
Memory usage: 6% IP address for eth0: 10.0.2.15
Swap usage: 0% IP address for eth1: 192.168.56.70
Graph this data and manage this system at:
https://landscape.canonical.com/
New release '16.04.7 LTS' available.
Run 'do-release-upgrade' to upgrade to it.
vagrant@master:~$ _
Right Ctrl

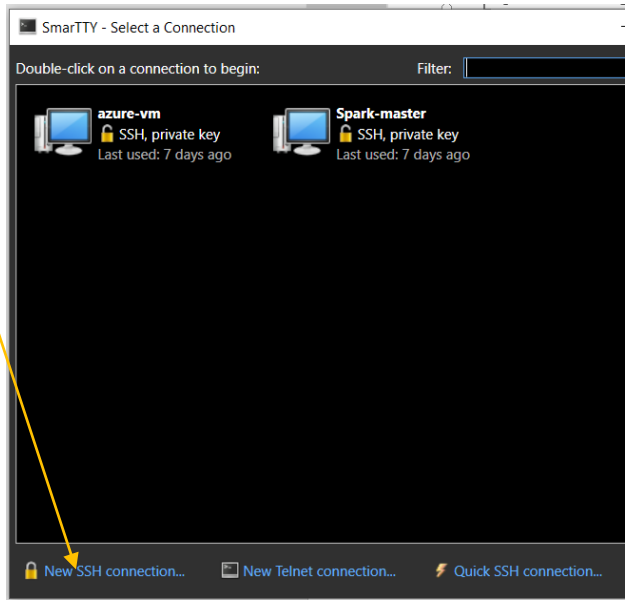
Connecting SmarTTY with the Linux Node

1. **Install SmarTTY .**
   a. **SmarTTY is a free multi-tabbed SSH client that supports copying files and directories with SCP on-the-fly and editing files in-place.**

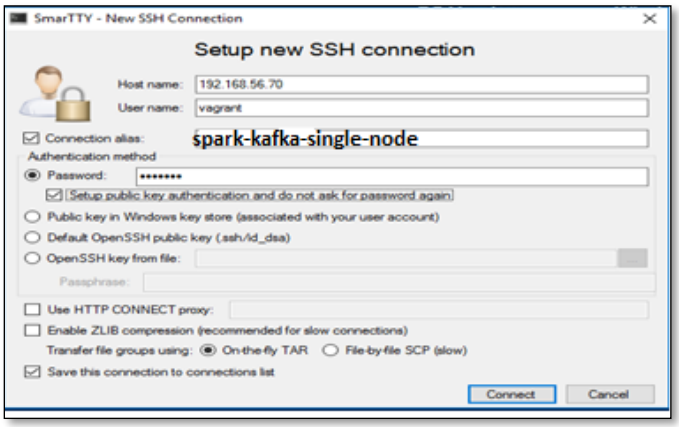
2. **To Connect SMartTTY with the Node , click on SmartTTy menu ,**



3. **Click on "New SSH Connection "**



4. **Fill the dialog box with the following information as shown below**

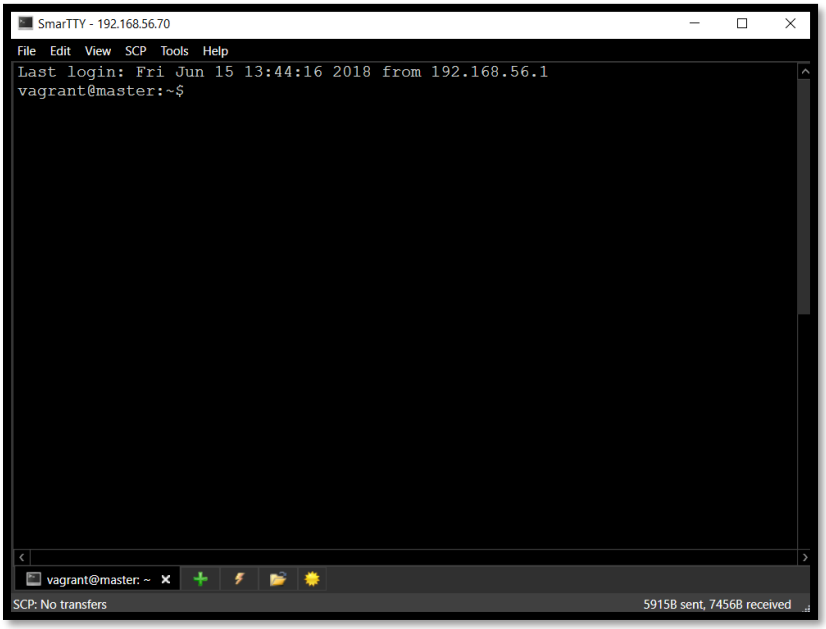


Host Name :- 192.168.56.70

User Name :- vagrant

Password :- vagrant

Click on "Connect"



You can open Multiple TAB connected with the Linux Node by clicking on + sign.

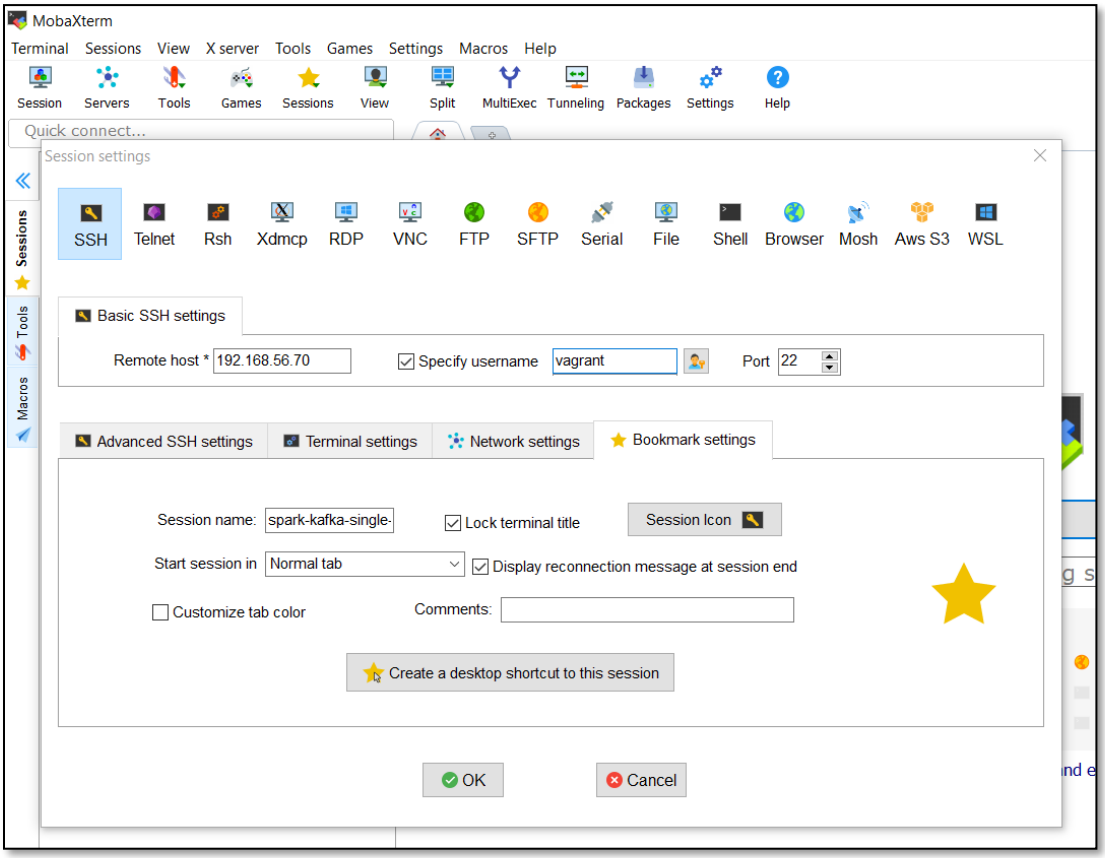
Now your Spark / Kafka environment is ready .

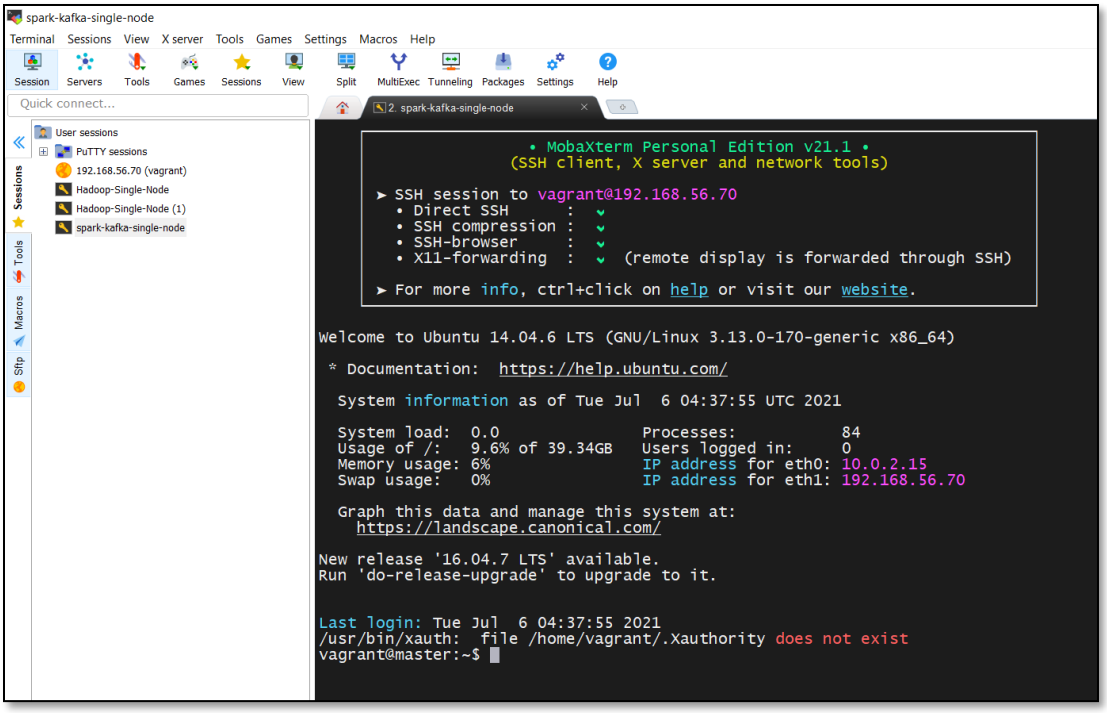
Connecting MobaXTerm with the Linux Node

1. Open **MobaXTerm**

Tool Bar → Click on "SSH" button → Click on "SSH" button

2. Fill the dialog box with the following information as shown below





Now your Spark / Kafka environment is ready .

Check Spark Services

Start the Spark Services

```
vagrant@master:~$ start-master.sh

vagrant@master:~$ start-slaves.sh
```

Check the services :-

```
$ jps
```

```
vagrant@master:~$ start-master.sh
starting org.apache.spark.deploy.master.Master, logging to /home/vagrant/bigdata/spark/logs/spark-vagrant-org.apache.spark.deploy.master.Master-1-master.out
vagrant@master:~$ start-slaves.sh
master: starting org.apache.spark.deploy.worker.Worker, logging to /home/vagrant/bigdata/spark/logs/spark-vagrant-org.apache.spark.deploy.worker.Worker-1-master.out
```

```
vagrant@master:~$ spark-shell --master spark://master:7077
```

```
Spark context Web UI available at http://master:4040
Spark context available as 'sc' (master = spark://master:7077, app id = app-20210706065010-0001).
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 3.0.2
      /_/

Using Scala version 2.12.10 (Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0_131)
Type in expressions to have them evaluated.
Type :help for more information.

scala>
```

```
scala> :q
vagrant@master:~$
```

Check PySpark Service

```
vagrant@master:~$ pyspark --master spark://master:7077
```

```
Python 3.4.3 (default, Nov 12 2018, 22:25:49)
[GCC 4.8.4] 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).
/home/vagrant/bigdata/spark/python/pyspark/context.py:227: DeprecationWarning: Support for Python 2 and Python 3 prior to version 3.6 is deprecated as of Spark 3.0. See also the plan for dropping Python 2 support at https://spark.apache.org/news/plan-for-dropping-python-2-support.html.
  DeprecationWarning)
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 3.0.2
      /_/

Using Python version 3.4.3 (default, Nov 12 2018 22:25:49)
SparkSession available as 'spark'.
>>>
```

```
>>> quit()
vagrant@master:~$
```

Note (If Require Python 3.6) –

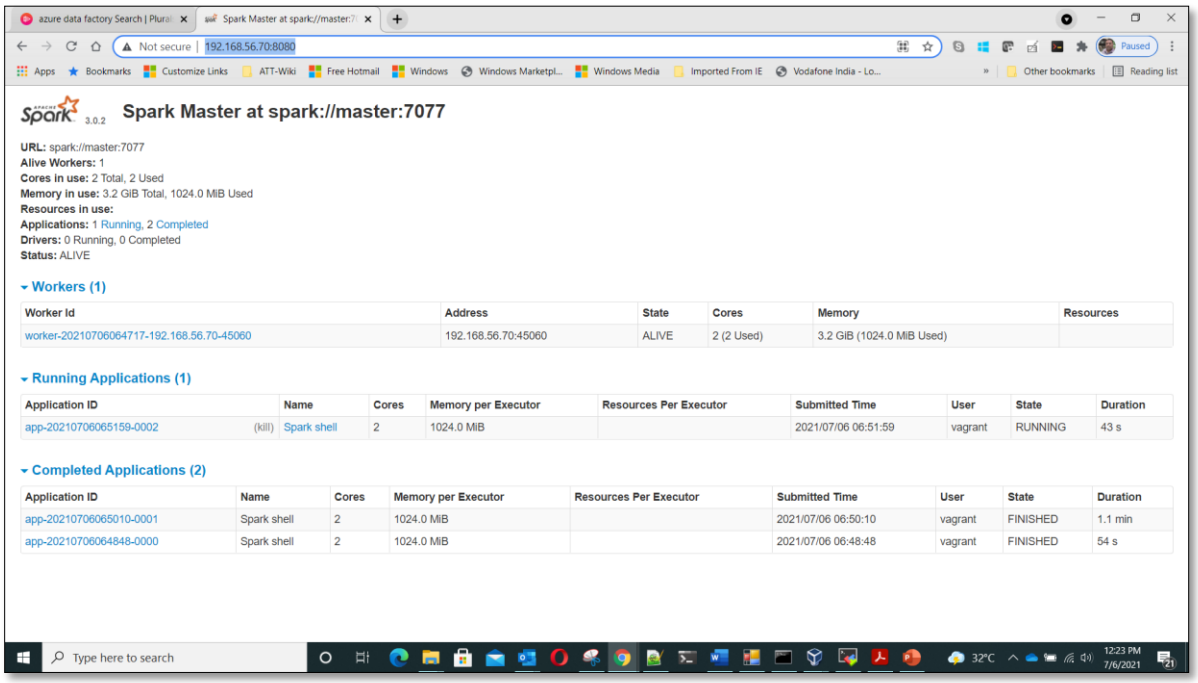
- Install Python3.6
- Add the following environment variable

vagrant@master:~$ vi .bashrc

export PYSPARK_PYTHON=python3.6

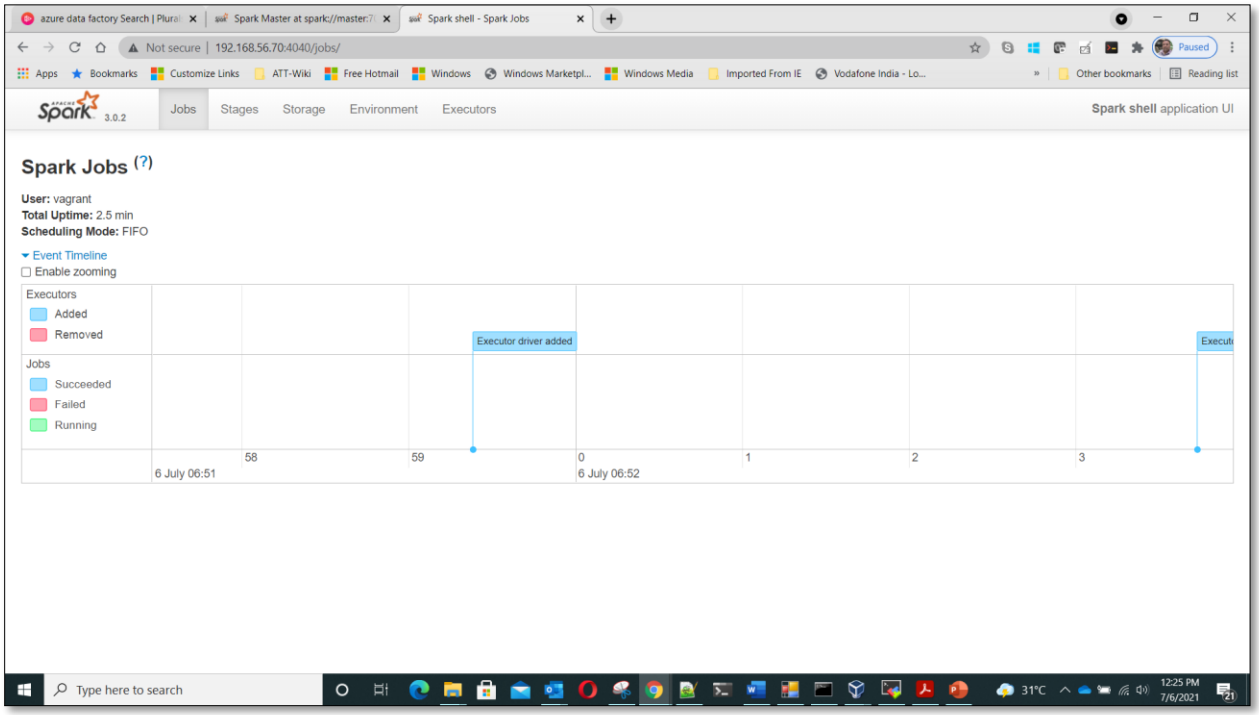
Spark Master Web Interface

http://192.168.56.70:8080/



Spark Job Web Interface

http://192.168.56.70:4040/



Shutdown the Node

If you want to shutdown your node completely ,

please type the following command in the **$** prompt (Either in Putty or in the Linux node directly).

```
$ sudo init 0
```

Your node will be shutdown.

Next time when you want to start it ,

- you have to open it from the **Oracle Virtual Box.**
- Select the node from the Oracle Virtual Box, click on the "**Start**" button .
- After the node has been started in the Virtual Box, connect it from windows using **Putty** .

Start the services again

For Spark (Mandatory)

```
$ start-master.sh
$ start-slaves.sh
```

Check Kafka Services

Start Zookeeper Service

```
vagrant@master:~$ cd bigdata/kafka/
```

vagrant@master:~/bigdata/kafka$ ./bin/zookeeper-server-start.sh config/zookeeper.properties

```
[2021-07-06 08:53:48,390] INFO binding to port 0.0.0.0/0.0.0.0:2181 (org.apache.zookeeper.server.NIOServerCnxnFactory)
[2021-07-06 08:53:48,454] INFO zookeeper.snapshotSizeFactor = 0.33 (org.apache.zookeeper.server.ZKDatabase)
[2021-07-06 08:53:48,482] INFO Snapshotting: 0x0 to /tmp/zookeeper/version-2/snapshot.0 (org.apache.zookeeper.server.persistence.FileTxnSnapLog)
[2021-07-06 08:53:48,510] INFO Snapshotting: 0x0 to /tmp/zookeeper/version-2/snapshot.0 (org.apache.zookeeper.server.persistence.FileTxnSnapLog)
[2021-07-06 08:53:48,570] INFO PrepRequestProcessor (sid:0) started, reconfigEnabled=false (org.apache.zookeeper.server.PrepRequestProcessor)
[2021-07-06 08:53:48,581] INFO Using checkIntervalMs=60000 maxPerMinute=10000 (org.apache.zookeeper.server.ContainerManager)
```

Start Kafka Broker Service

Open another TAB

vagrant@master:~$ jps

2292 Jps
1766 QuorumPeerMain

vagrant@master:~$ cd bigdata/kafka/
vagrant@master:~/bigdata/kafka$ ./bin/kafka-server-start.sh config/server.properties

```
[2021-07-06 08:59:19,710] INFO [SocketServer listenerType=ZK_BROKER, nodeId=0] Starting socket server acceptors and processors (kafka.network.SocketServer)
[2021-07-06 08:59:19,749] INFO [SocketServer listenerType=ZK_BROKER, nodeId=0] Started data-plane acceptor and processor(s) for endpoint : ListenerName(PLAINTEXT) (kafka.network.SocketServer)
[2021-07-06 08:59:19,749] INFO [SocketServer listenerType=ZK_BROKER, nodeId=0] Started socket server acceptors and processors (kafka.network.SocketServer)
[2021-07-06 08:59:19,759] INFO Kafka version: 2.8.0 (org.apache.kafka.common.utils.AppInfoParser)
[2021-07-06 08:59:19,759] INFO Kafka commitId: ebb1d6e21cc92130 (org.apache.kafka.common.utils.AppInfoParser)
[2021-07-06 08:59:19,759] INFO Kafka startTimeMs: 1625561959749 (org.apache.kafka.common.utils.AppInfoParser)
[2021-07-06 08:59:19,761] INFO [KafkaServer id=0] started (kafka.server.KafkaServer)
[2021-07-06 08:59:19,871] INFO [broker-0-to-controller-send-thread]: Recorded new controller, from now on will use broker master:9092 (id: 0 rack: null) (kafka.server.BrokerToControllerRequestThread)
```

Open another TAB

vagrant@master:~$ jps

2896 Jps
2485 Kafka
1766 QuorumPeerMain

Stop All Services

Press CTRL+C in the respective TAB

1. **First Stop Kafka Broker**
2. **Second Stop Zookeeper Service**

OR

Use the Third TAB

```
$kill -9 <Process ID of Kafka Broker>   i.e.  $ kill -9 2485
$ kill -9 <Process ID of Zookeeper Service >   i.e.  $ kill -9 1766
```

To Start All Services in the background

```
vagrant@master:~$ cd bigdata/kafka/
vagrant@master:~/bigdata/kafka$ ./bin/zookeeper-server-start.sh config/zookeeper.properties &
vagrant@master:~/bigdata/kafka$ ./bin/kafka-server-start.sh config/server.properties  &
```

Check Confluent Services

Start All Services

```
vagrant@master:~$ confluent local services start
```

```
The local commands are intended for a single-node development environment only,
NOT for production usage. https://docs.confluent.io/current/cli/index.html

Using CONFLUENT_CURRENT: /tmp/confluent.465510
Starting ZooKeeper
ZooKeeper is [UP]
Starting Kafka
Kafka is [UP]
Starting Schema Registry
Schema Registry is [UP]
Starting Kafka REST
Kafka REST is [UP]
Starting Connect
Connect is [UP]
Starting ksqlDB Server
ksqlDB Server is [UP]
vagrant@master:~$
```

```
vagrant@master:~$ jps
```

```
3187 KafkaRestMain
2933 QuorumPeerMain
3238 ConnectDistributed
3446 Jps
3094 SchemaRegistryMain
2999 Kafka
3372 KsqlServerMain
```

Stop All Services

```
vagrant@master:~$ confluent local services stop
```

```
The local commands are intended for a single-node development environment only,
NOT for production usage. https://docs.confluent.io/current/cli/index.html

Using CONFLUENT_CURRENT: /tmp/confluent.465510
Stopping ksqlDB Server
ksqlDB Server is [DOWN]
Stopping Connect
Connect is [DOWN]
Stopping Kafka REST
Kafka REST is [DOWN]
Stopping Schema Registry
Schema Registry is [DOWN]
Stopping Kafka
Kafka is [DOWN]
Stopping ZooKeeper
ZooKeeper is [DOWN]
```

```
vagrant@master:~$ jps
```

3550 Jps

For Confluent Enterprise Version

https://www.confluent.io/installation

https://packages.confluent.io/archive/6.2/confluent-6.2.0.tar.gz

Check MySQL Services

```
vagrant@master:~$ mysql -u root -p
Enter password: root
```

mysql> show databases;

```
+--------------------+
| Database           |
+--------------------+
| information_schema |
| metastore_db       |
| mysql              |
| performance_schema |
+--------------------+
4 rows in set (0.24 sec)
```

Check Cassandra Services

Start Cassandra in the foreground

```
$ cassandra -f
```

from the command line.

Press "Control-C" to stop Cassandra.

Start Cassandra in the background

```
$ cassandra
```

To Stop Cassandra running in Background

```
kill -9 pid
```

Know Cassandra PID

```
vagrant@master:~$ ps aux | grep cassandra
```

Verify that Cassandra is running

```
$ nodetool status
```

```
Datacenter: datacenter1
=======================
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
--  Address    Load       Tokens       Owns (effective)  Host ID                               Rack
UN  127.0.0.1  70.72 KiB  256          100.0%            1f70cda0-b52a-4e45-a0d3-122b020e7dac  rack1
```

Configuration files are located in the conf sub-directory.

Due to this, it is necessary to either start Cassandra with root privileges or change **conf/cassandra.yaml**

CQLSH

cqlsh is a command line shell for interacting with Cassandra through CQL. It is shipped with every Cassandra package, and can be found in the **bin**/ directory alongside the **cassandra** executable. It connects to the single node specified on the command line.

For example:

```
$ bin/cqlsh localhost
```

```
Connected to Test Cluster at localhost:9042.
[cqlsh 5.0.1 | Cassandra 3.8 | CQL spec 3.4.2 | Native protocol v4]
Use HELP for help.
```

```
cqlsh> SELECT cluster_name, listen_address FROM system.local;

 cluster_name | listen_address
--------------+----------------
 Test Cluster |      127.0.0.1
(1 rows)
cqlsh>
```

Check MongoDB Services

Start MongoDB server

```
$ mongod
```

```
2018-06-15T15:28:41.663+0530 I COMMAND  [initandlisten] setting featureCompatibilityVersion to 3.6
2018-06-15T15:28:41.685+0530 I STORAGE  [initandlisten] createCollection: local.startup_log with generated UUID: ee022a43-f237-4c10-bb71-d0094eb5
c8ea
2018-06-15T15:28:41.699+0530 I FTDC     [initandlisten] Initializing full-time diagnostic data capture with directory '/data/db/diagnostic.data'
2018-06-15T15:28:41.700+0530 I NETWORK  [initandlisten] waiting for connections on port 27017
```

Start Mongo Shell in another TAB

Open Another TAB to start Mongo Shell

```
$ mongo
```

```
MongoDB shell version v4.2.13
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("c8fb8628-bbd1-44f8-a3ee-6ccdf3682c1d") }
MongoDB server version: 4.2.13
Welcome to the MongoDB shell.
For interactive help, type "help".
For more comprehensive documentation, see
        https://docs.mongodb.com/
Questions? Try the MongoDB Developer Community Forums
        https://community.mongodb.com
Server has startup warnings:
```

```
>
```

```
> show databases;
```

admin 0.000GB

config 0.000GB

local 0.000GB

```
> use admin;
```

switched to db admin

```
> show collections;
```

system.version

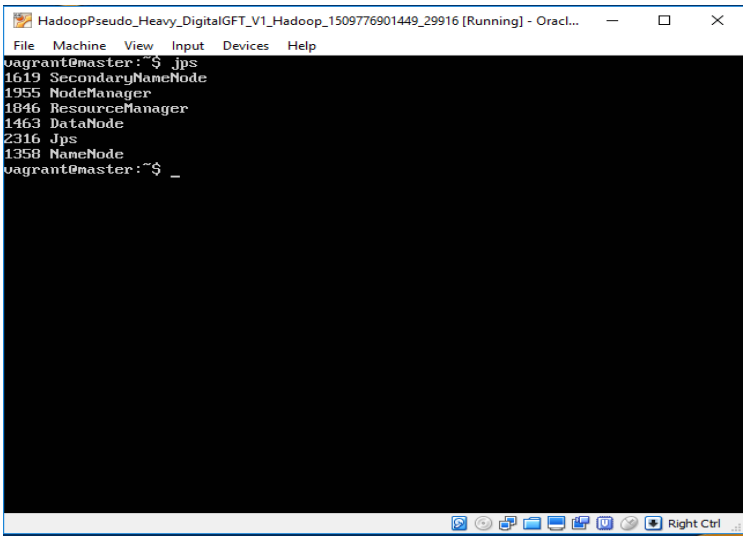
```
> quit()
```

```
vagrant@master:~$
```

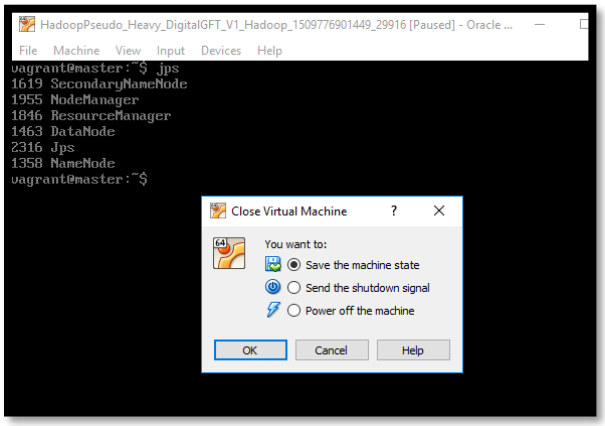
Press CTRL+C to stop the MongoDB server in the First TAB

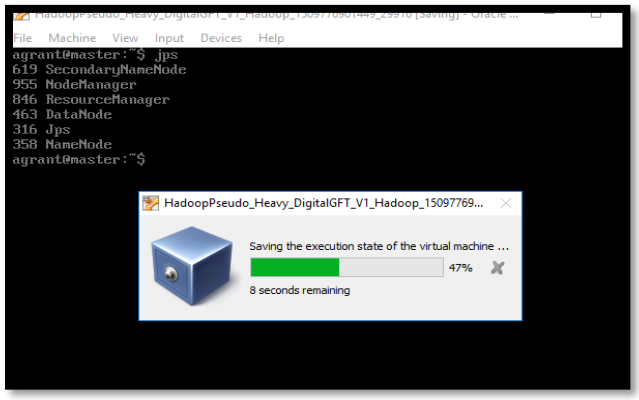
Suspend the Linux Node from Virtual Box

1. **Click on the "close" button of the Linux Window opened in Virtual Box**

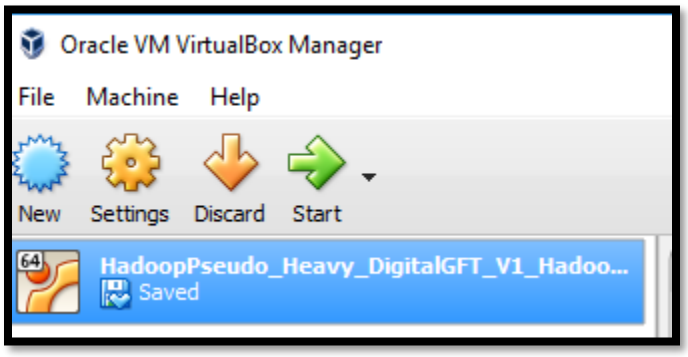


2. **It will open another dialog box asking about the operations of your choice , click on the choice "Save the machine state " →Click on "OK"**



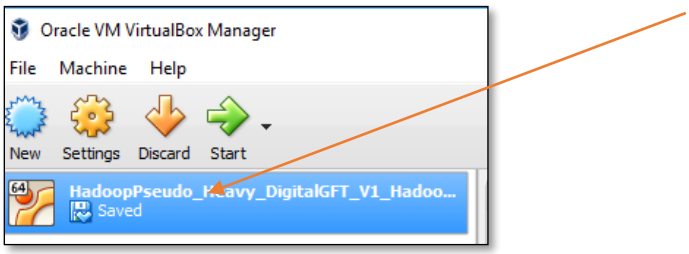


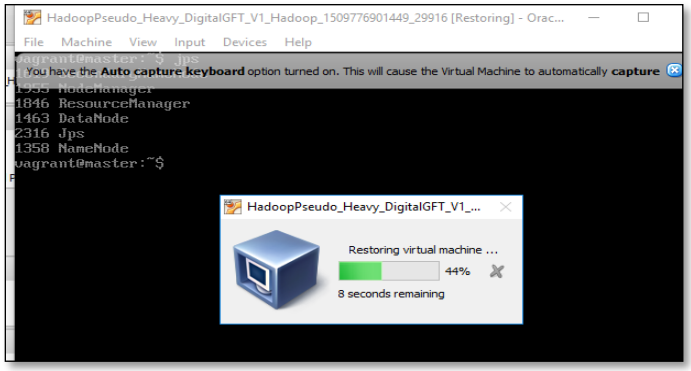
3. **In the Virtual Box window the Linux node will be shown in "SAVED" mode .**



To start the Linux node from "saved" state

Select the Linux Node in the Virtual Box window (shown in "saved" mode) → click on "Start" button





Check the "Services" using "jps" command; if the services are not running , start the services.

```
$ jps
```

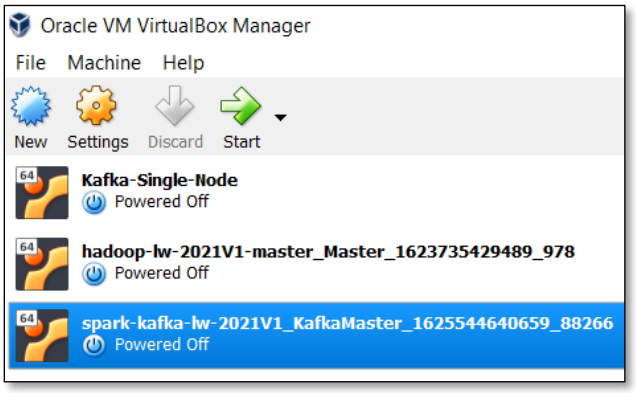
Shutdown the Node

To shutdown the Node completely

Type the following command in the **$** prompt (Either in Putty or in the Linux node directly).

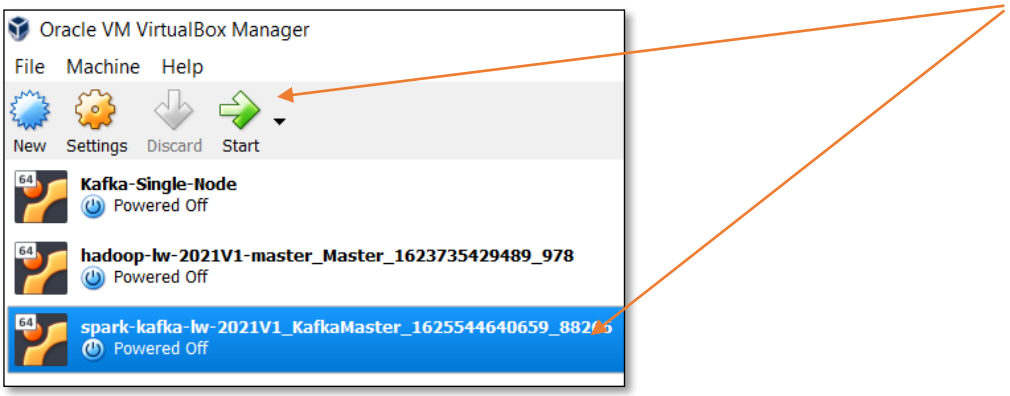
$ sudo init 0

The node will be shutdown and it will shown as **"Powered Off" state** in the Virtual Box Window.



To Start the Node from "Powered Off" state

- Open the **Oracle Virtual Box.**
- Select the node from the Oracle Virtual Box, click on the "**Start**" button .
- After the node has been started in the Virtual Box, connect it from windows using **Putty** or **SmarTTY**.



Start the spark services using the following commands: -

```
$ start-master.sh
$ start-slaves.sh
```

Check the services :-

```
$ jps
```

Check Spark Master web interface

`http://192.168.56.70:8080`