

Exp.1 Downloading and installing Hadoop, Understanding different Hadoop modes, Startup scripts, Configuration files.

AIM:

To Download and install Hadoop, Understanding different Hadoop modes, Startup scripts, Configuration files.

Procedure:

Step 1 : Install Java Development Kit

The default Ubuntu repositories contain Java 8 and Java 11 both. But, Install Java 8 because it only works on this version. Use the following command to install it.

```
$sudo apt update&&sudo apt install openjdk-8-jdk
```

Step 2 : Verify the Java version

Once installed, verify the installed version of Java with the following command:

```
$ java -version
```

Output:

```
sanjay@sanjay-VirtualBox:~$ java -version
openjdk version "1.8.0_382"
OpenJDK Runtime Environment (build 1.8.0_382-8u382-ga-1~23.04.1-b05)
OpenJDK 64-Bit Server VM (build 25.382-b05, mixed mode)
```

Step 3: Install SSH

SSH (Secure Shell) installation is vital for Hadoop as it enables secure communication between nodes in the Hadoop cluster. This ensures data integrity, confidentiality, and allows for efficient distributed processing of data across the cluster.

```
$sudo apt install ssh
```

Step 4 : Create the hadoop user :

All the Hadoop components will run as the user that you create for Apache Hadoop, and the user will also be used for logging in to Hadoop's web interface.

Run the command to create user and set password:

```
$ sudo adduser hadoop
```

Output:

```

sanjay@sanjay-VirtualBox:~$ sudo adduser hadoop
Adding user 'hadoop' ...
Adding new group 'hadoop' (1001) ...
Adding new user 'hadoop' (1001) with group 'hadoop (1001)' ...
adduser: The home directory '/home/hadoop' already exists. Not copying from '/etc/skel'.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: password updated successfully
Changing the user information for hadoop
Enter the new value, or press ENTER for the default
    Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n] y
Adding new user 'hadoop' to supplemental / extra groups 'users' ...
Adding user 'hadoop' to group 'users' ...
sanjay@sanjay-VirtualBox:~$ su - hadoop
Password:
hadoop@sanjay-VirtualBox:~$

```

Step 5 : Switch user

Switch to the newly created hadoop user:

\$ su - hadoop

Step 6 : Configure SSH

Now configure password-less SSH access for the newly created hadoop user, so didn't enter the key to save file and passphrase. Generate an SSH keypair (generate Public and Private Key Pairs)first

\$ssh-keygen -t rsa

```

hadoop@sanjay-VirtualBox:~$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/hadoop/.ssh/id_rsa):
Created directory '/home/hadoop/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/hadoop/.ssh/id_rsa
Your public key has been saved in /home/hadoop/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:yu8Hsie3mbQ7UifnFH6iam4kFLRRbEb9zVYGutbaYyg hadoop@sanjay-VirtualBox
The key's randomart image is:
+---[RSA 3072]-----+
|      .o+.  ..+o |
|      .o+ .  o  + |
|      .+   . o.= |
|      .   E..=o. |
|      . S   ... . |
|      .o.= o     |
|      o*.B       |
|      +o*++ o .  |
|      X0*..+o    |
+---[SHA256]-----+
hadoop@sanjay-VirtualBox:~$

```

Step 7 : Set permissions :

Next, append the generated public keys from id_rsa.pub to authorized_keys and set proper permission:

\$ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys

\$ chmod 640 ~/.ssh/authorized_keys

Step 8 : SSH to the localhost

Next, verify the password less SSH authentication with the following command:

\$ ssh localhost

You will be asked to authenticate hosts by adding RSA keys to known hosts. Type yes and hit Enter to authenticate the localhost:

```
hadoop@sanjay-VirtualBox: ~/hadoop$ ssh localhost
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ED25519 key fingerprint is SHA256:2ZsL3K5BKG6h8lsZpTufDvB69zWFKS7lFjvsnhW53I.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'localhost' (ED25519) to the list of known hosts.
Welcome to Ubuntu 23.04 (GNU/Linux 6.2.0-32-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

122 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

hadoop@sanjay-VirtualBox: $
```

Step 9 : Switch user

Again switch to hadoop. So, First, change the user to hadoop with the following command:

\$ su-hadoop

Step 10 : Install hadoop

Next, download the latest version of Hadoop using the wget command:

\$ wget <https://downloads.apache.org/hadoop/common/hadoop-3.3.6/hadoop-3.3.6.tar.gz>

Once downloaded, extract the downloaded file:

\$ tar -xvzf hadoop-3.3.6.tar.gz

Next, rename the extracted directory to hadoop:

\$ mv hadoop-3.3.6 hadoop

```
hadoop@sanjay-VirtualBox:~$ mv hadoop-3.3.6 hadoop
hadoop@sanjay-VirtualBox:~$ ls
hadoop  hadoop-3.3.6.tar.gz
hadoop@sanjay-VirtualBox:~$
```

Next, you will need to configure Hadoop and Java Environment Variables on your system. Open the ~/.bashrc file in your favorite text editor. Use nano editor , to pasting the code we use ctrl+shift+v for saving the file ctrl+x and ctrl+y ,then hit enter:

Next, you will need to configure Hadoop and Java Environment Variables on your system.

Open the ~/.bashrc file in your favorite text editor:

\$ nano ~/.bashrc

Append the below lines to file.

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
export HADOOP_HOME=/home/hadoop/hadoop
export HADOOP_INSTALL=$HADOOP_HOME
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export HADOOP_YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib/native"
```

Save and close the file. Then, activate the environment variables with the following command:

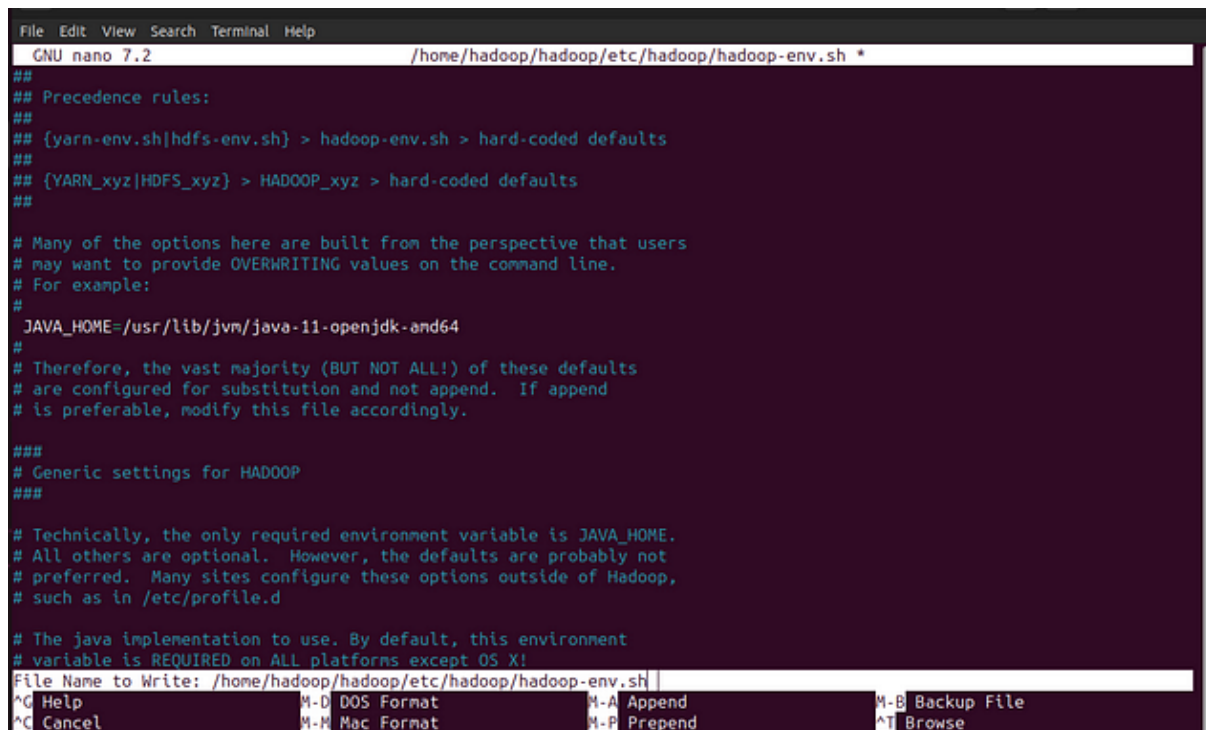
```
s$ source ~/.bashrc
```

Next, open the Hadoop environment variable file:

```
$ nano $HADOOP_HOME/etc/hadoop/hadoop-env.sh
```

Search for the “export JAVA_HOME” and configure it.

```
JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
```



```
File Edit View Search Terminal Help
GNU nano 7.2 /home/hadoop/hadoop/etc/hadoop/hadoop-env.sh *
##
## Precedence rules:
##
## {yarn-env.sh|hdfs-env.sh} > hadoop-env.sh > hard-coded defaults
## {YARN_xyz|HDFS_xyz} > HADOOP_xyz > hard-coded defaults
##
# Many of the options here are built from the perspective that users
# may want to provide OVERWRITING values on the command line.
# For example:
#
# JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64
#
# Therefore, the vast majority (BUT NOT ALL!) of these defaults
# are configured for substitution and not append. If append
# is preferable, modify this file accordingly.
###
# Generic settings for HADOOP
###
# Technically, the only required environment variable is JAVA_HOME.
# All others are optional. However, the defaults are probably not
# preferred. Many sites configure these options outside of Hadoop,
# such as in /etc/profile.d
#
# The java implementation to use. By default, this environment
# variable is REQUIRED on ALL platforms except OS X!
File Name to Write: /home/hadoop/hadoop/etc/hadoop/hadoop-env.sh
^C Help      M-D DOS Format  M-A Append     M-B Backup File
^C Cancel    M-M Mac Format  M-P Prepend    ^T Browse
```

Save and close the file when you are finished.

Step 11 : Configuring Hadoop :

First, you will need to create the namenode and datanode directories inside the Hadoop user home directory. Run the following command to create both directories:

```
$ cd hadoop/
```

```
$ mkdir -p ~/hadoopdata/hdfs/{namenode,datanode}
```

- Next, edit the core-site.xml file and

```
hadoop@sanjay-VirtualBox:~$ nano $HADOOP_HOME/etc/hadoop/hadoop-env.sh
hadoop@sanjay-VirtualBox:~$ cd hadoop/
hadoop@sanjay-VirtualBox:~/hadoop$ mkdir -p ~/hadoopdata/hdfs/{namenode,datanode}
hadoop@sanjay-VirtualBox:~/hadoop$
```

update with your system hostname:

```
$ nano $HADOOP_HOME/etc/hadoop/core-site.xml
```

Change the following name as per your system hostname:

```
<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://localhost:9000</value>
  </property>
</configuration>
```

Save and close the file.

Then, edit the hdfs-site.xml file:

```
$ nano $HADOOP_HOME/etc/hadoop/hdfs-site.xml
```

- Change the NameNode and DataNode directory paths as shown below:

```
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>1</value>
  </property>

  <property>
    <name>dfs.namenode.name.dir</name>
    <value>file:///home/hadoop/hadoopdata/hdfs/namenode</value>
  </property>

  <property>
    <name>dfs.datanode.data.dir</name>
    <value>file:///home/hadoop/hadoopdata/hdfs/datanode</value>
  </property>
</configuration>
```

- Then, edit the mapred-site.xml file:

```
$ nano $HADOOP_HOME/etc/hadoop/mapred-site.xml
```

- Make the following changes:

```
<configuration>
  <property>
    <name>yarn.app.mapreduce.am.env</name>
    <value>HADOOP_MAPRED_HOME=$HADOOP_HOME/home/hadoop/hadoop/bin/hadoop</value>
  </property>
  <property>
    <name>mapreduce.map.env</name>
    <value>HADOOP_MAPRED_HOME=$HADOOP_HOME/home/hadoop/hadoop/bin/hadoop</value>
  </property>
  <property>
    <name>mapreduce.reduce.env</name>
    <value>HADOOP_MAPRED_HOME=$HADOOP_HOME/home/hadoop/hadoop/bin/hadoop</value>
  </property>
</configuration>
```

- Then, edit the yarn-site.xml file:
\$nano \$HADOOP_HOME/etc/hadoop/yarn-site.xml
- Make the following changes:

```
<configuration>
  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>
</configuration>
```

Save the file and close it .

Step 12 – Start Hadoop Cluster

Before starting the Hadoop cluster. You will need to format the Namenode as a hadoop user.

Run the following command to format the Hadoop Namenode:

```
$hdfs namenode -format
```

Once the namenode directory is successfully formatted with hdfs file system, you will see the message “Storage directory /home/hadoop/hadoopdata/hdfs/namenode has been successfully formatted “

Then start the Hadoop cluster with the following command.

```
$ start-all.sh
```



```

hadoop@sanjay-VirtualBox:~$ start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as hadoop in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [sanjay-VirtualBox]
Starting resourcemanager
Starting nodemanagers
hadoop@sanjay-VirtualBox:~$

```

You can now check the status of all Hadoop services using the jps command:

\$ jps

```

hadoop@sanjay-VirtualBox:~/hadoop$ jps
7235 NodeManager
6677 DataNode
7593 Jps
6554 NameNode
7116 ResourceManager
6893 SecondaryNameNode
hadoop@sanjay-VirtualBox:~/hadoop$

```

Step 13 – Access Hadoop Namenode and Resource Manager

- First we need to know our ipaddress, In Ubuntu we need to install net-tools to run ifconfig command,
If you installing net-tools for the first time switch to default user:

\$sudo apt install net-tools

- Then run ifconfig command to know our ip address:

ifconfig

```

hadoop@sanjay-VirtualBox:~/hadoop$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.6 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 2401:4900:1c28:46c4:f76c:b206:abe3:2d45 prefixlen 64 scopeid 0x0<global>
    inet6 2401:4900:1c28:46c4:ed13:53f4:5c05:50c6 prefixlen 64 scopeid 0x0<global>
    inet6 fe80::112b:300a:9242:51f3 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:83:31:35 txqueuelen 1000 (Ethernet)
    RX packets 645228 bytes 934388358 (934.3 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 93618 bytes 8998032 (8.9 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 3331 bytes 491873 (491.8 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 3331 bytes 491873 (491.8 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

hadoop@sanjay-VirtualBox:~/hadoop$

```

Here my ip address is 192.168.1.6.

- To access the Namenode, open your web browser and visit the URL <http://your-server-ip:9870>.
- You should see the following screen:
<http://192.168.1.6:9870>

← → ↻

192.168.1.6:9870/dfshealth.html#tab-overview

☆

🔍

☰

Hadoop

Overview

Datanodes

Datanode Volume Failures

Snapshot

Startup Progress

Utilities ▾

Overview

'localhost:9000' (✓active)

| | |
|----------------|--|
| Started: | Sun Sep 10 13:08:22 +0530 2023 |
| Version: | 3.3.6, r1be78238728da9266a4f88195058f08fd012bf9c |
| Compiled: | Sun Jun 18 13:52:00 +0530 2023 by ubuntu from (HEAD detached at release-3.3.6-RC1) |
| Cluster ID: | CID-dc5a1253-b0cd-4686-a807-fd0dd4c5a9a |
| Block Pool ID: | BP-1272319295-127.0.1.1-1694331447796 |

Summary

Security is off.

Safemode is off.

1 files and directories, 0 blocks (0 replicated blocks, 0 erasure coded block groups) = 1 total filesystem object(s).

Heap Memory used 69.85 MB of 107 MB Heap Memory. Max Heap Memory is 748 MB.

Non Heap Memory used 51.89 MB of 55.44 MB Committed Non Heap Memory. Max Non Heap Memory is <unbounded>.

Configured Capacity:

24.44 GB

To access Resource Manage, open your web browser and visit the URL `http://your-server-ip:8088`. You should see the following screen:

<http://192.168.16:8088>


← → ↻

192.168.1.6:8088/cluster

☆

🔍

☰



Cluster

About

Nodes

Node Labels

Applications

NEW

NEW SAVING

SUBMITTED

ACCEPTED

RUNNING

FINISHED

FAILED

KILLED

Scheduler

Tools

Cluster Metrics

| Apps Submitted | Apps Pending | Apps Running | Apps Completed | Containers Running |
|----------------|--------------|--------------|----------------|--------------------|
| 0 | 0 | 0 | 0 | 0 |

Cluster Nodes Metrics

| Active Nodes | Decommissioning Nodes | Decom |
|--------------|-----------------------|-------|
| 1 | 0 | 0 |

Scheduler Metrics

| Scheduler Type | Scheduling Resource Type | Min |
|--------------------|-------------------------------|----------------------|
| Capacity Scheduler | [memory-mb (unit=Mi), vcores] | <memory:1024, vCore: |

Show 20 ▾ entries

| ID ▾ | User ▾ | Name ▾ | Application Type ▾ | Application Tags ▾ | Queue ▾ | Application Priority ▾ | StartTime ▾ | LaunchTime ▾ | Finis |
|-----------------------------|--------|--------|--------------------|--------------------|---------|------------------------|-------------|--------------|-------|
| Showing 0 to 0 of 0 entries | | | | | | | | | |

Step 14 – Verify the Hadoop Cluster

At this point, the Hadoop cluster is installed and configured. Next, we will create some directories in the HDFS filesystem to test the Hadoop.

Let's create some directories in the HDFS filesystem using the following command:

```
$ hdfsdfs -mkdir /test1
$ hdfsdfs -mkdir /logs
```


Next, run the following command to list the above directory:

```
$ hdfs dfs -ls /
```

You should get the following output:

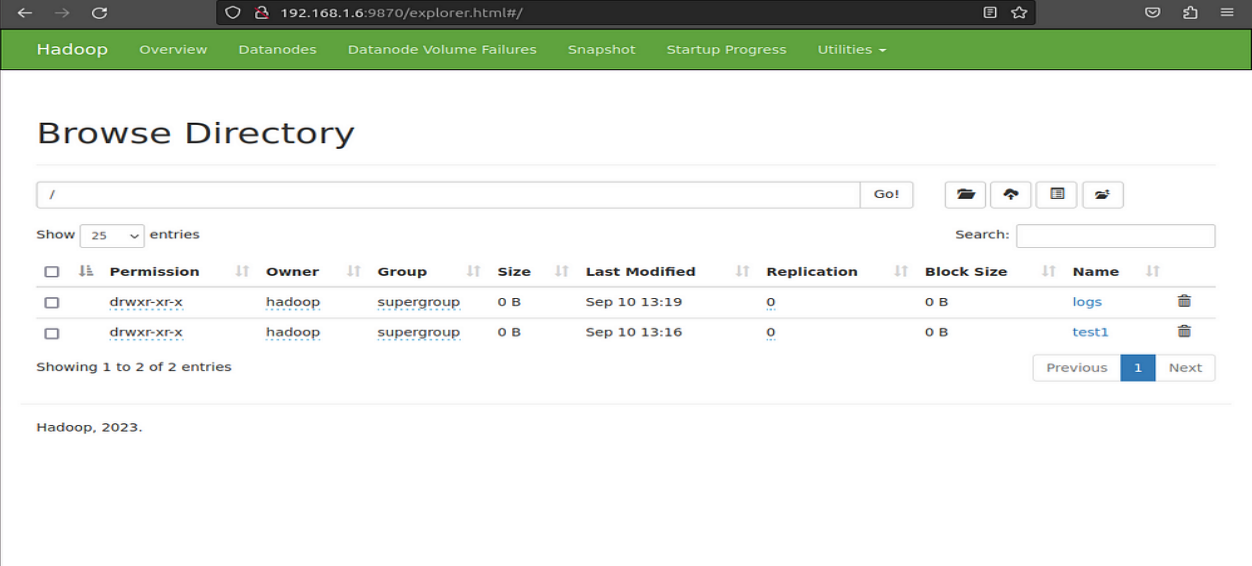
```
hadoop@sanjay-VirtualBox:~/hadoop$ hdfs dfs -ls /
Found 2 items
drwxr-xr-x - hadoop supergroup 0 2023-09-10 13:16 /logs
drwxr-xr-x - hadoop supergroup 0 2023-09-10 13:16 /test1
hadoop@sanjay-VirtualBox:~/hadoop$
```

Also, put some files to hadoop file system. For the example, putting log files from host machine to hadoop file system.

```
$ hdfs dfs -put /var/log/* /logs/
```

You can also verify the above files and directory in the Hadoop Namenode web interface.

Go to the web interface, click on the Utilities => Browse the file system. You should see your directories which you have created earlier in the following screen:



The screenshot shows the Hadoop web interface with the 'Utilities' menu open and 'Browse the file system' selected. The 'Browse Directory' page displays a table of files in the root directory. The table has columns for Permission, Owner, Group, Size, Last Modified, Replication, Block Size, and Name. Two entries are shown: 'logs' and 'test1', both with permissions 'drwxr-xr-x', owner 'hadoop', group 'supergroup', size '0 B', and last modified 'Sep 10 13:19' and 'Sep 10 13:16' respectively. The page also shows a search bar and pagination controls.

| Permission | Owner | Group | Size | Last Modified | Replication | Block Size | Name |
|------------|--------|------------|------|---------------|-------------|------------|-------|
| drwxr-xr-x | hadoop | supergroup | 0 B | Sep 10 13:19 | 0 | 0 B | logs |
| drwxr-xr-x | hadoop | supergroup | 0 B | Sep 10 13:16 | 0 | 0 B | test1 |

Step 15 – Stop Hadoop Cluster

To stop the Hadoop all services, run the following command:

```
$ stop-all.sh
```

```
hadoop@sanjay-VirtualBox:~/hadoop$ stop-all.sh
WARNING: Stopping all Apache Hadoop daemons as hadoop in 10 seconds.
WARNING: Use CTRL-C to abort.
Stopping namenodes on [localhost]
Stopping datanodes
Stopping secondary namenodes [sanjay-VirtualBox]
Stopping nodemanagers
Stopping resourcemanager
hadoop@sanjay-VirtualBox:~/hadoop$
```

Result:

The step-by-step installation and configuration of Hadoop on Ubuntu linux system have been successfully completed.