

**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 hive 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:

```
thirueswaran-v@thirueswaran-v-VirtualBox:~$ java -version
openjdk version "1.8.0_422"
OpenJDK Runtime Environment (build 1.8.0_422-8u422-b05-1~24.04-b05)
OpenJDK 64-Bit Server VM (build 25.422-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:

```
thirueswaran-v@thirueswaran-v-VirtualBox:~$ sudo adduser hadoop
info: Adding user 'hadoop' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group 'hadoop' (1001) ...
info: Adding new user 'hadoop' (1001) with group 'hadoop (1001)' ...
info: Creating home directory '/home/hadoop' ...
info: Copying files 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
info: Adding new user 'hadoop' to supplemental / extra groups 'users' ...
info: Adding user 'hadoop' to group 'users' ...
thirueswaran-v@thirueswaran-v-VirtualBox:~$ su -hadoop
```

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

```
thirueswaran-v@thirueswaran-v-VirtualBox:~$ ssh-keygen -t rsa -P '' -f ~/.ssh/id_rsa
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
Generating public/private rsa key pair.
/home/thirueswaran-v/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Your identification has been saved in /home/thirueswaran-v/.ssh/id_rsa
Your public key has been saved in /home/thirueswaran-v/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:kJw9MvnQ/8ait6j1FMuQZd0E3zb8XoifB4rK8nz25ew thirueswaran-v@thirueswaran-v-VirtualBox
The key's randomart image is:
+---[RSA 3072]-----+
|
| ..
| . * .o
| X + +..o
| * * .. =
| S o o +
| o =. o o|
| . =.+o.+|
| .+ ==o.+o o|
| .+B=oo..E. |
+---[SHA256]-----+
thirueswaran-v@thirueswaran-v-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@thirueswaran-v-VirtualBox:~$ ssh localhost
hostkeys_find_by_key_hostfile: hostkeys_foreach failed for /home/hadoop/.ssh/known_hosts: Not a directory
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ED25519 key fingerprint is SHA256:TUQ9UdYybEUd2I1jxhR0oseyHcY2elVAIk9gAnuv/fc.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Failed to add the host to the list of known hosts (/home/hadoop/.ssh/known_hosts).
hadoop@localhost's password:
client_input_hostkeys: hostkeys_foreach failed for /home/hadoop/.ssh/known_hosts: Not a directory
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-41-generic x86_64)

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

Expanded Security Maintenance for Applications is not enabled.

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

8 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

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@thirueswaran-v-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
```

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:

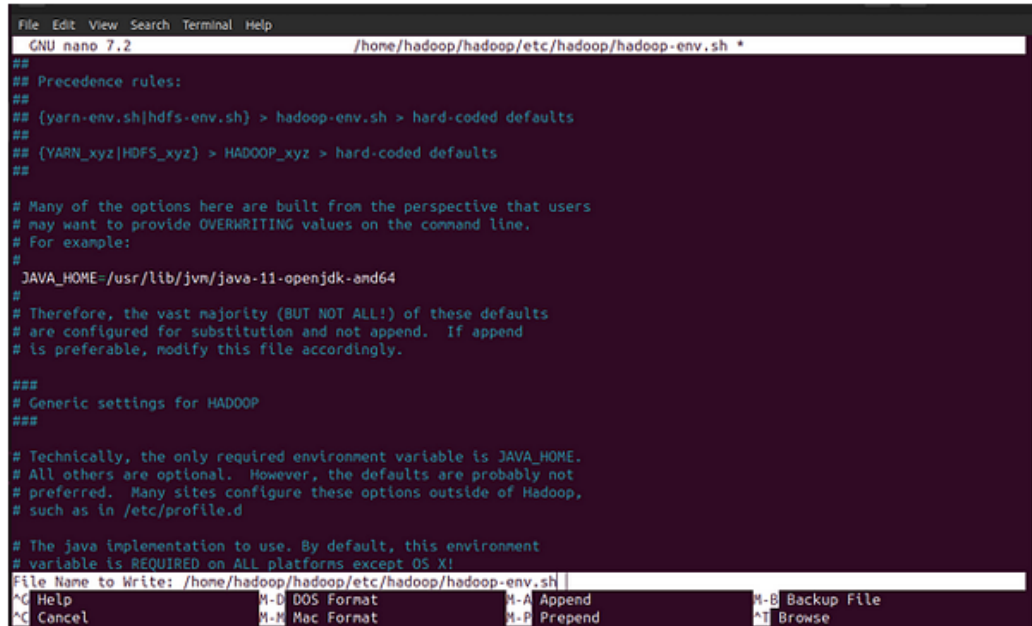
```
$ source ~/.bashrc
```

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    ^M-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 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

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

\$ jps

```
thirueswaran-v@thirueswaran-v-VirtualBox:~$ jps
14000 NodeManager
13329 NameNode
14439 Jps
13656 SecondaryNameNode
13882 ResourceManager
13451 DataNode
thirueswaran-v@thirueswaran-v-VirtualBox:~$
```

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

ipconfig 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

```
thirueswaran-v@thirueswaran-v-VirtualBox:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fe80::a00:27ff:fe61:5412 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:61:54:12 txqueuelen 1000 (Ethernet)
    RX packets 1191629 bytes 1714077401 (1.7 GB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 105500 bytes 9372134 (9.3 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 63307 bytes 6003384 (6.0 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 63307 bytes 6003384 (6.0 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

thirueswaran-v@thirueswaran-v-VirtualBox:~$
```

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>

Overview 'localhost:9000' (✓active)

Started:	Sat Sep 07 13:27:00 +0530 2024
Version:	3.4.0, rbd8b77f398f626bb7791783192ee7a5dfaec760
Compiled:	Mon Mar 04 12:05:00 +0530 2024 by root from (HEAD detached at release-3.4.0-RC3)
Cluster ID:	CID-7e1ffe8c-aca5-423a-ab20-8ecbe5c24588
Block Pool ID:	BP-597443322-127.0.1.1-1725695749992

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 134.74 MB of 239.5 MB Heap Memory. Max Heap Memory is 1.1 GB.
 Non Heap Memory used 49.87 MB of 51.33 MB Committed Non Heap Memory. Max Non Heap Memory is «unbounded».

Configured Capacity: 0 B

To access Resource Manager, open your web browser and visit the URL [http://your-server ip:8088](http://your-server-ip:8088). You should see the following screen:

<http://192.168.16:8088>

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running
0	0	0	0	0

Cluster Nodes Metrics

Active Nodes	Decommissioning Nodes	Decommissioned Nodes
1	0	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation
Capacity Scheduler	[memory-mb (unit=M), vcores]	<memory:1024, vCores:1>	<memory:8192, vCores:4>

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:

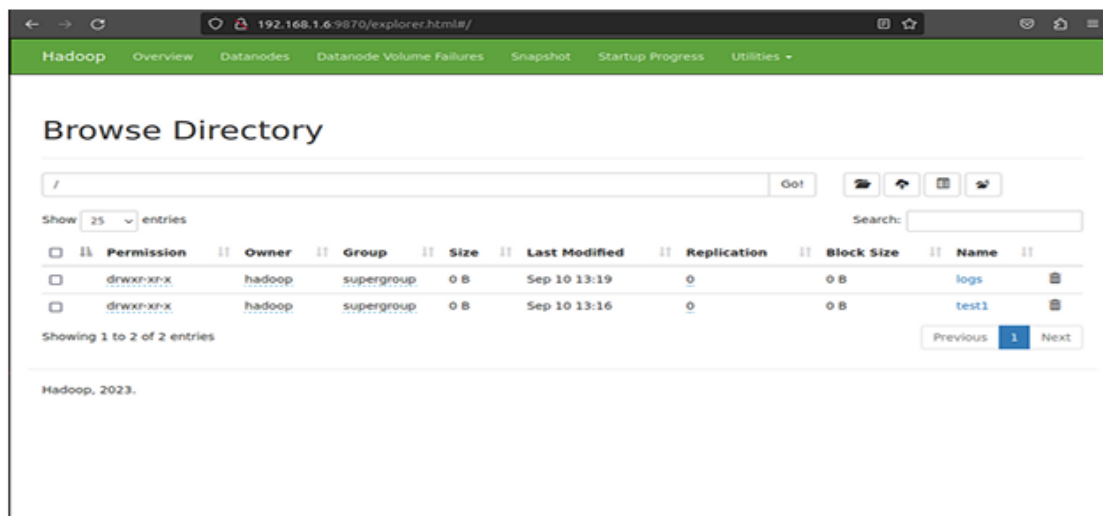
```
$ hdfs dfs -ls /
```

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:



Step 15 – Stop Hadoop Cluster

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

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

Result:

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