

## REGISTER NO.: 210701263

Exp. No: 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 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:**

```
subbu@subbu:~$ java -version
openjdk version "1.8.0_422"
OpenJDK Runtime Environment (build 1.8.0_422-8u422-b05-1~22.04-b05)
OpenJDK 64-Bit Server VM (build 25.422-b05, mixed mode)
subbu@subbu:~$
```

##### 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
```

##### 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
```

## REGISTER NO.: 210701263

### 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:

```
subbu@subbu:~$ ssh localhost
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 6.8.0-40-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.

40 updates can be applied immediately.
27 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

New release '24.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Fri Sep 13 07:38:09 2024 from 127.0.0.1
```

### 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:

## REGISTER NO.: 210701263

\$ **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**

```
subbu@subbu:~$ ls
apache-hive-3.1.2-bin      exp4      Pictures
apache-hive-3.1.2-bin.tar.gz exp41     pig
Desktop                  exp6      Public
Documents                hadoop     snap
Downloads                hadoop-3.3.6.tar.gz Templates
exp2                     hive       Videos
exp3                     Music      word_count.txt.save
subbu@subbu:~$
```

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:

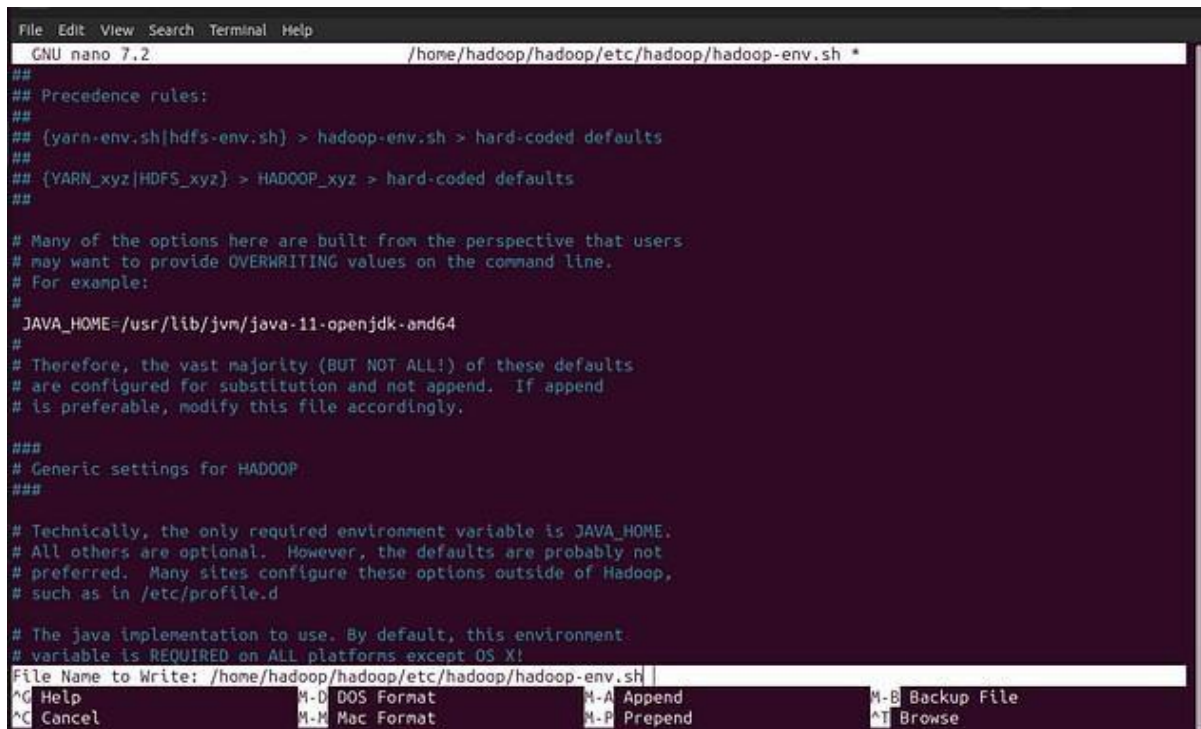
**source ~/.bashrc**

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 OVERRIDING 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
^O 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 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:

## REGISTER NO.: 210701263

```
<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
```

```
subbu@subbu:~$ start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as subbu in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
localhost: namenode is running as process 3019. Stop it first and ensure /tmp
/hadoop-subbu-namenode.pid file is empty before retry.
Starting datanodes
localhost: datanode is running as process 3146. Stop it first and ensure /tmp
/hadoop-subbu-datanode.pid file is empty before retry.
Starting secondary namenodes [subbu]
subbu: secondarynamenode is running as process 3374. Stop it first and ensure
/tmp/hadoop-subbu-secondarynamenode.pid file is empty before retry.
Starting resourcemanager
resourcemanager is running as process 3578. Stop it first and ensure /tmp/had
oop-subbu-resourcemanager.pid file is empty before retry.
Starting nodemanagers
localhost: nodemanager is running as process 3700. Stop it first and ensure /
tmp/hadoop-subbu-nodemanager.pid file is empty before retry.
```

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

```
$ jps
```

```
subbu@subbu:~$ jps
5970 RunJar
3700 NodeManager
6185 RunJar
11066 Jps
3146 DataNode
3578 ResourceManager
3019 NameNode
3374 SecondaryNameNode
subbu@subbu:~$
```

### 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**

Here my ip address is 192.168.1.6.

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

<http://192.168.1.6:9870>



## REGISTER NO.: 210701263

**Overview** 'localhost:9000' (✓active)

<b>Started:</b>	Fri Sep 20 21:34:43 +0530 2024
<b>Version:</b>	3.3.6, r1be78238728da9266a4f88195058f08fd012bf9c
<b>Compiled:</b>	Sun Jun 18 13:52:00 +0530 2023 by ubuntu from (HEAD detached at release-3.3.6-RC1)
<b>Cluster ID:</b>	CID-4ee06551-10c1-403f-b60e-9e0cb7fb787d
<b>Block Pool ID:</b>	BP-576718551-127.0.1.1-1726193722123

### Summary

Security is off.  
Safemode is off.

106 files and directories, 53 blocks (53 replicated blocks, 0 erasure coded block groups) = 159 total filesystem object(s).

Heap Memory used 144.43 MB of 264.5 MB Heap Memory. Max Heap Memory is 871.5 MB.

Non Heap Memory used 69.87 MB of 71.98 MB Committed Non Heap Memory. Max Non Heap Memory is <unbounded>.

<b>Configured Capacity:</b>	28.87 GB
-----------------------------	----------

To access Resource Manage, open your web browser and visit the URL <http://your-serverip: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	Min Resource
Capacity Scheduler	[memory-mb (unit=Mi), vcores]	<memory:1024, vCores:1>

Show 20 entries

ID	User	Name	Application Type	Application Tags	Queue	Application Priority	StartTime	LaunchTime	FinishTime
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:

## REGISTER NO.: 210701263

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

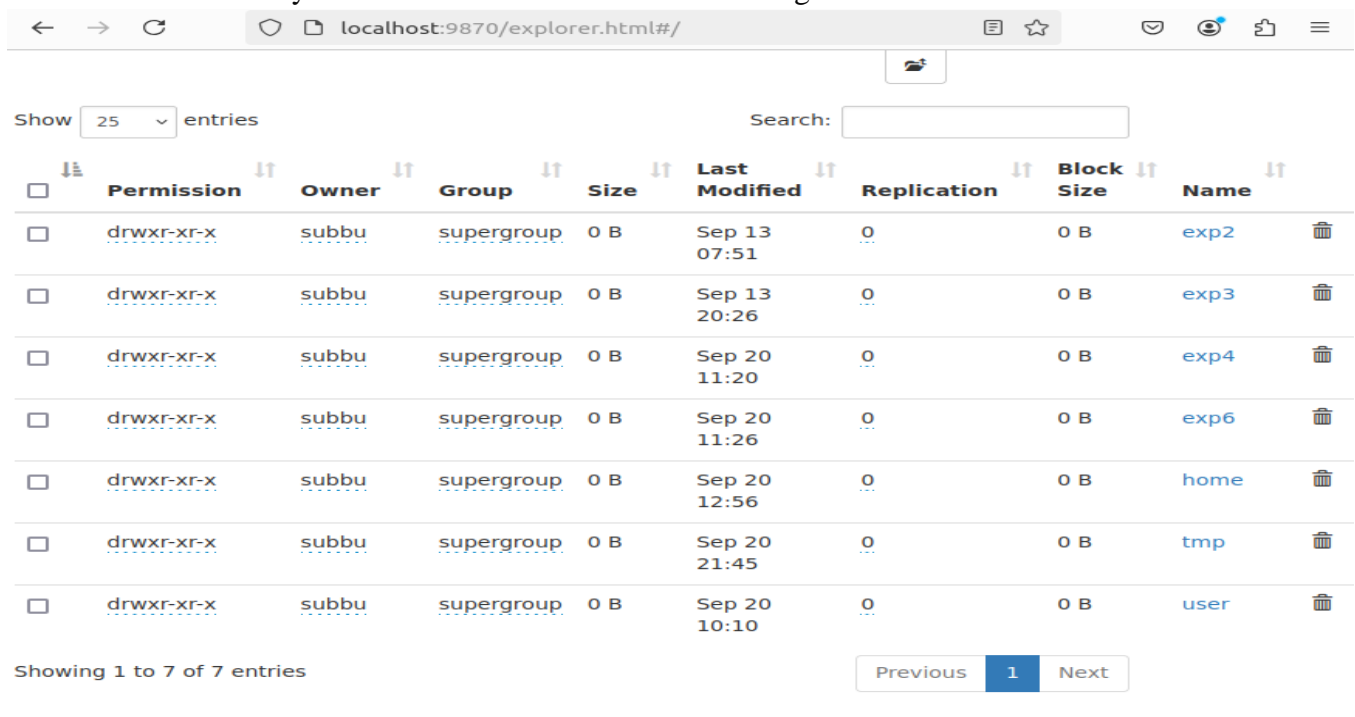
Next, run the following command to list the above directory:

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:



Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxr-xr-x	subbu	supergroup	0 B	Sep 13 07:51	0	0 B	exp2
drwxr-xr-x	subbu	supergroup	0 B	Sep 13 20:26	0	0 B	exp3
drwxr-xr-x	subbu	supergroup	0 B	Sep 20 11:20	0	0 B	exp4
drwxr-xr-x	subbu	supergroup	0 B	Sep 20 11:26	0	0 B	exp6
drwxr-xr-x	subbu	supergroup	0 B	Sep 20 12:56	0	0 B	home
drwxr-xr-x	subbu	supergroup	0 B	Sep 20 21:45	0	0 B	tmp
drwxr-xr-x	subbu	supergroup	0 B	Sep 20 10:10	0	0 B	user

### 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.