

**ROLLNO: 210701268**

**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:**

**1) 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
```

**2) Step 2 : Verify the Java version**

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

```
$ java -version Output:
```

**3) 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**

**4) 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
```

**5) Step 5 : Switch user**

Switch to the newly created hadoop user:

```
$ su - hadoop
```

**6) 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

**7) 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
```

### 8) 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:

### 9) Step 9 : Switch user

Again switch to hadoop. So, First, change the user to hadoop with the following command: **\$ su-hadoop**

### 10) Step 10 : Install hadoop

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

```
$ wgethttps://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:

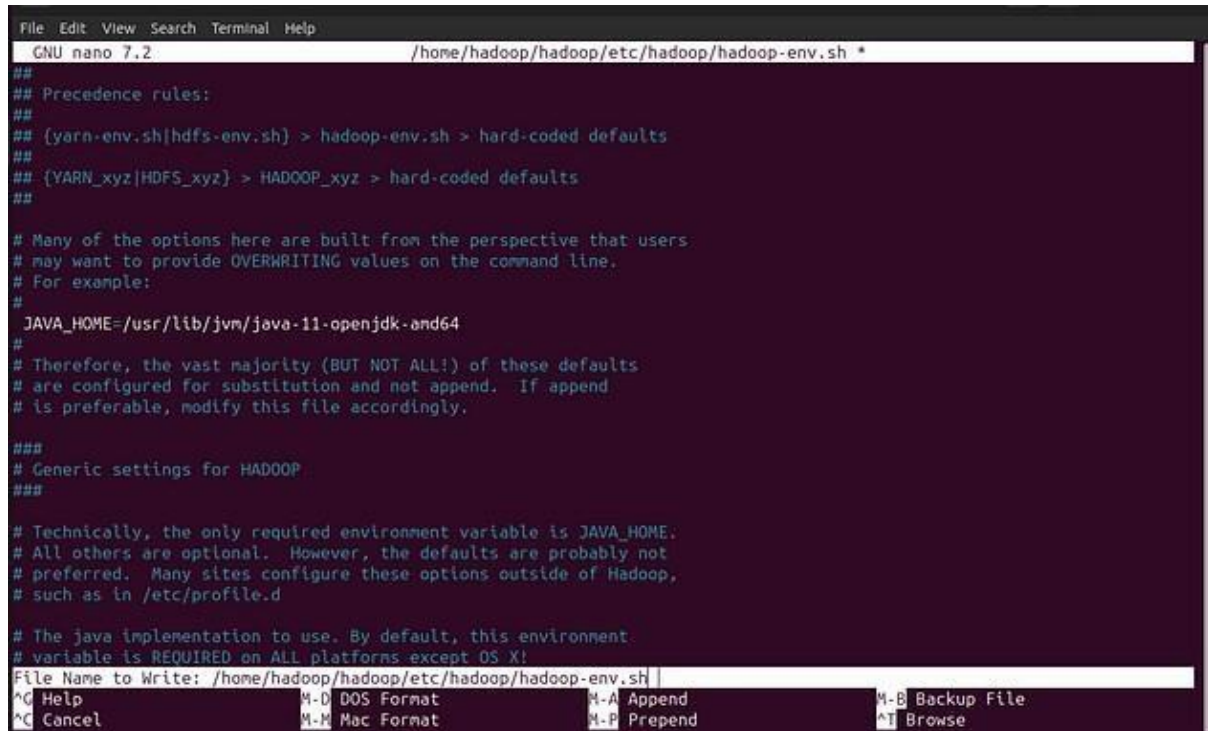
```
s$ source ~/.bashrc
```

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 ^M-T Browse
```

Save and close the file when you are finished.

### 11) 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}
```

12) Step 12 :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 .

### 13) Step 13 – 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
```

```

sudhashreem@sudhashreem-VirtualBox:~$ start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as sudhashreem 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 2323. Stop it first and ensure /tmp/hadoop-sudhashreem-namenode.pid file is empty before retry.
Starting datanodes
localhost: datanode is running as process 2447. Stop it first and ensure /tmp/hadoop-sudhashreem-datanode.pid file is empty before retry.
Starting secondary namenodes [sudhashreem-VirtualBox]
sudhashreem-VirtualBox: secondarynamenode is running as process 2633. Stop it first and ensure /tmp/hadoop-sudhashreem-secondarynamenode.pid file is empty before retry.
Starting resource manager
resource manager is running as process 2894. Stop it first and ensure /tmp/hadoop-sudhashreem-resource manager.pid file is empty before retry.
Starting nodemanagers
localhost: nodemanager is running as process 3013. Stop it first and ensure /tmp/hadoop-sudhashreem-nodemanager.pid file is empty before retry.
sudhashreem@sudhashreem-VirtualBox:~$ jps

```

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

\$ jps

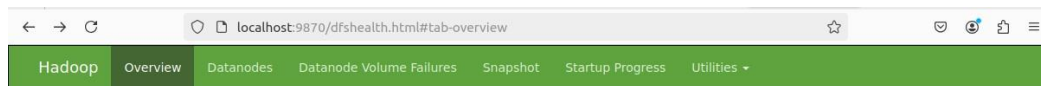
```

sudhashreem@sudhashreem-VirtualBox:~$ jps
4336 NodeManager
3777 DataNode
3654 NameNode
3977 SecondaryNameNode
4219 ResourceManager
8094 Jps
sudhashreem@sudhashreem-VirtualBox:~$

```

#### 14) Step 14 – Access Hadoop Namenode and Resource Manager

- First we need to know our ipaddress, In Ubuntu we need to install net-tools to run ipconfig.  
If you installing net-tools for the first time switch to default user:
- Then run ifconfig command to know our ip address: **ifconfig**
- To access the Namenode, open your web browser and visit the URL <http://yourserverip:9870>.
- You should see the following in your browser by typing : <https://localhost:9870/>



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

Started:	Sat Sep 14 19:40:37 +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-76e164a6-773c-4c98-ad99-ef182be086d3
Block Pool ID:	BP-1511180079-127.0.1.1-1726199105209

## Summary

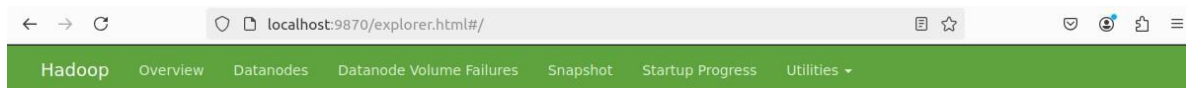
Security is off.

Safemode is off.

120 files and directories, 78 blocks (78 replicated blocks, 0 erasure coded block groups) = 198 total filesystem object(s).

Heap Memory used 130.24 MB of 335.5 MB Heap Memory. Max Heap Memory is 871.5 MB.

Non Heap Memory used 71.99 MB of 73.53 MB Committed Non Heap Memory. Max Non Heap Memory is <unbounded>.



## Browse Directory

/

Go!

Show

25

entries

Search:

<input type="checkbox"/>	<div>⌵</div> Permission	<div>⌵</div> Owner	<div>⌵</div> Group	<div>⌵</div> Size	<div>⌵</div> Last Modified	<div>⌵</div> Replication	<div>⌵</div> Block Size	<div>⌵</div> Name	<div>⌵</div>
<input type="checkbox"/>	drwxr-xr-x	sudhashreem	supergroup	0 B	Sep 13 13:39	0	0 B	exp2	<div>🗑</div>
<input type="checkbox"/>	drwxr-xr-x	sudhashreem	supergroup	0 B	Sep 14 20:22	0	0 B	exp3	<div>🗑</div>
<input type="checkbox"/>	drwxr-xr-x	sudhashreem	supergroup	0 B	Sep 13 14:07	0	0 B	exp4	<div>🗑</div>
<input type="checkbox"/>	drwxr-xr-x	sudhashreem	supergroup	0 B	Sep 14 11:19	0	0 B	home	<div>🗑</div>
<input type="checkbox"/>	drwxr-xr-x	sudhashreem	supergroup	0 B	Sep 14 14:22	0	0 B	tmp	<div>🗑</div>

Showing 1 to 5 of 5 entries

Previous

1

Next

Hadoop, 2024.

### 15) Step 15– 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.

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:



←

→

↺

localhost:9870/explorer.html#/

📄

☆

🔖

👤

🔗

☰

Hadoop

Overview

Datanodes

Datanode Volume Failures

Snapshot

Startup Progress

Utilities ▾

Browse Directory

/

Go!

📁

🔼

📄

🔍

Show

25 ▾

entries

Search:

Showing 1 to 5 of 5 entries

Previous

1

Next

Hadoop, 2024.

## 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 Ubutu linux system have been successfully completed.