

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.

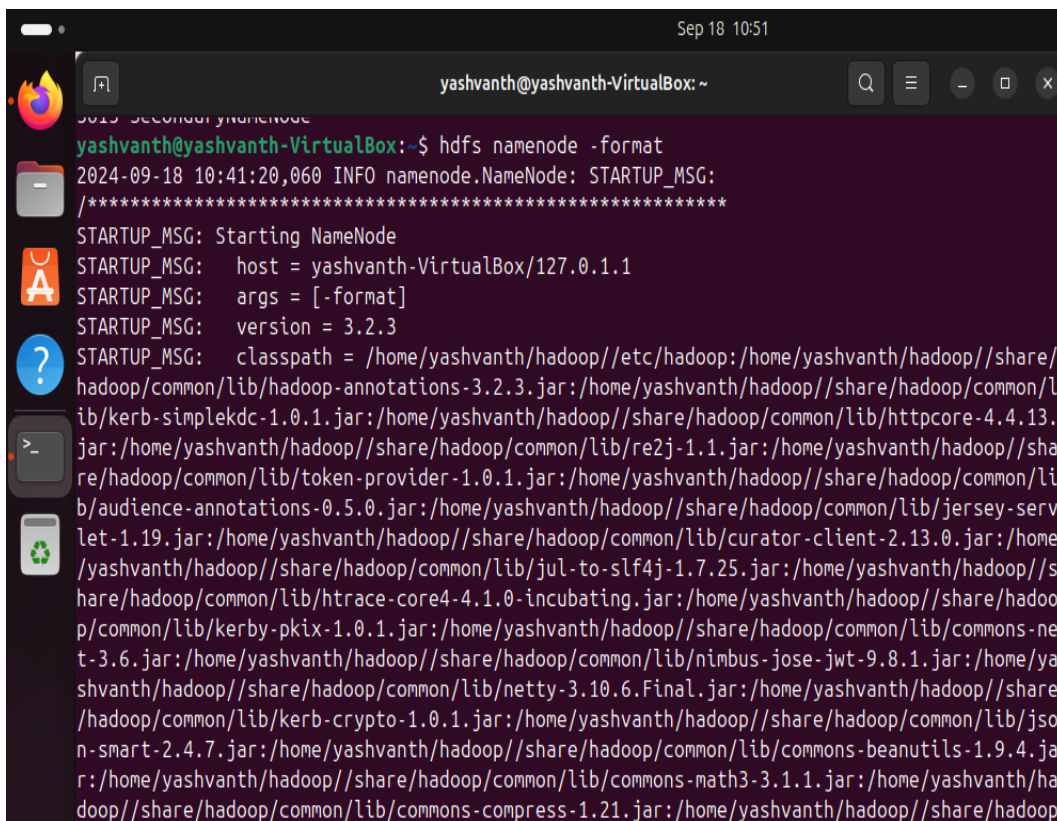
Procedure:**Step 1 : Install Java Development Kit**

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:


```

yashvanth@yashvanth-VirtualBox: ~$ hdfs namenode -format
2024-09-18 10:41:20,060 INFO namenode.NameNode: STARTUP_MSG:
/*****
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   host = yashvanth-VirtualBox/127.0.1.1
STARTUP_MSG:   args = [-format]
STARTUP_MSG:   version = 3.2.3
STARTUP_MSG:   classpath = /home/yashvanth/hadoop/etc/hadoop:/home/yashvanth/hadoop/share/
hadoop/common/lib/hadoop-annotations-3.2.3.jar:/home/yashvanth/hadoop/share/hadoop/common/lib/kerb-simplekdc-1.0.1.jar:/home/yashvanth/hadoop/share/hadoop/common/lib/httpcore-4.4.13.jar:/home/yashvanth/hadoop/share/hadoop/common/lib/re2j-1.1.jar:/home/yashvanth/hadoop/share/hadoop/common/lib/token-provider-1.0.1.jar:/home/yashvanth/hadoop/share/hadoop/common/lib/audience-annotations-0.5.0.jar:/home/yashvanth/hadoop/share/hadoop/common/lib/jersey-servlet-1.19.jar:/home/yashvanth/hadoop/share/hadoop/common/lib/curator-client-2.13.0.jar:/home/yashvanth/hadoop/share/hadoop/common/lib/jul-to-slf4j-1.7.25.jar:/home/yashvanth/hadoop/share/hadoop/common/lib/htrace-core4-4.1.0-incubating.jar:/home/yashvanth/hadoop/share/hadoop/common/lib/kerby-pkix-1.0.1.jar:/home/yashvanth/hadoop/share/hadoop/common/lib/commons-net-3.6.jar:/home/yashvanth/hadoop/share/hadoop/common/lib/nimbus-jose-jwt-9.8.1.jar:/home/yashvanth/hadoop/share/hadoop/common/lib/netty-3.10.6.Final.jar:/home/yashvanth/hadoop/share/hadoop/common/lib/kerb-crypto-1.0.1.jar:/home/yashvanth/hadoop/share/hadoop/common/lib/jsn-smart-2.4.7.jar:/home/yashvanth/hadoop/share/hadoop/common/lib/commons-beanutils-1.9.4.jar:/home/yashvanth/hadoop/share/hadoop/common/lib/commons-math3-3.1.1.jar:/home/yashvanth/hadoop/share/hadoop/common/lib/commons-compress-1.21.jar:/home/yashvanth/hadoop/share/hadoop

```

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
```

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
```

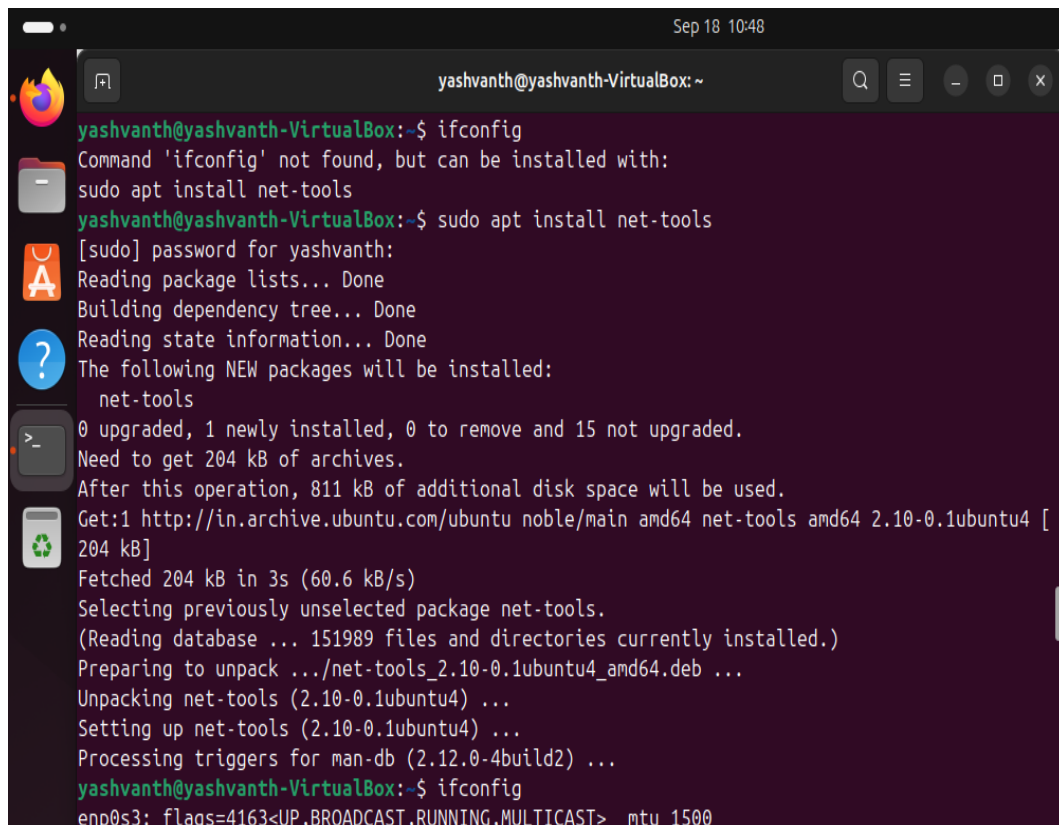
```
$ chmod 640 ~/.ssh/authorized_keys
```

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:



```

yashvanth@yashvanth-VirtualBox: ~
yashvanth@yashvanth-VirtualBox:~$ ifconfig
Command 'ifconfig' not found, but can be installed with:
sudo apt install net-tools
yashvanth@yashvanth-VirtualBox:~$ sudo apt install net-tools
[sudo] password for yashvanth:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  net-tools
0 upgraded, 1 newly installed, 0 to remove and 15 not upgraded.
Need to get 204 kB of archives.
After this operation, 811 kB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu noble/main amd64 net-tools amd64 2.10-0.1ubuntu4 [
204 kB]
Fetched 204 kB in 3s (60.6 kB/s)
Selecting previously unselected package net-tools.
(Reading database ... 151989 files and directories currently installed.)
Preparing to unpack .../net-tools_2.10-0.1ubuntu4_amd64.deb ...
Unpacking net-tools (2.10-0.1ubuntu4) ...
Setting up net-tools (2.10-0.1ubuntu4) ...
Processing triggers for man-db (2.12.0-4build2) ...
yashvanth@yashvanth-VirtualBox:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500

```

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:

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

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/yarnsite.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

```

yashvanth@yashvanth-VirtualBox:~$ start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as yashvanth 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 [yashvanth-VirtualBox]
Starting resourcemanager
Starting nodemanagers
yashvanth@yashvanth-VirtualBox:~$ jps
8195 Jps
7540 SecondaryNameNode
7175 NameNode
7720 ResourceManager
8040 NodeManager
7340 DataNode

```

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

\$ jps

```

yashvanth@yashvanth-VirtualBox:~$ jps
8195 Jps
7540 SecondaryNameNode
7175 NameNode
7720 ResourceManager
8040 NodeManager
7340 DataNode

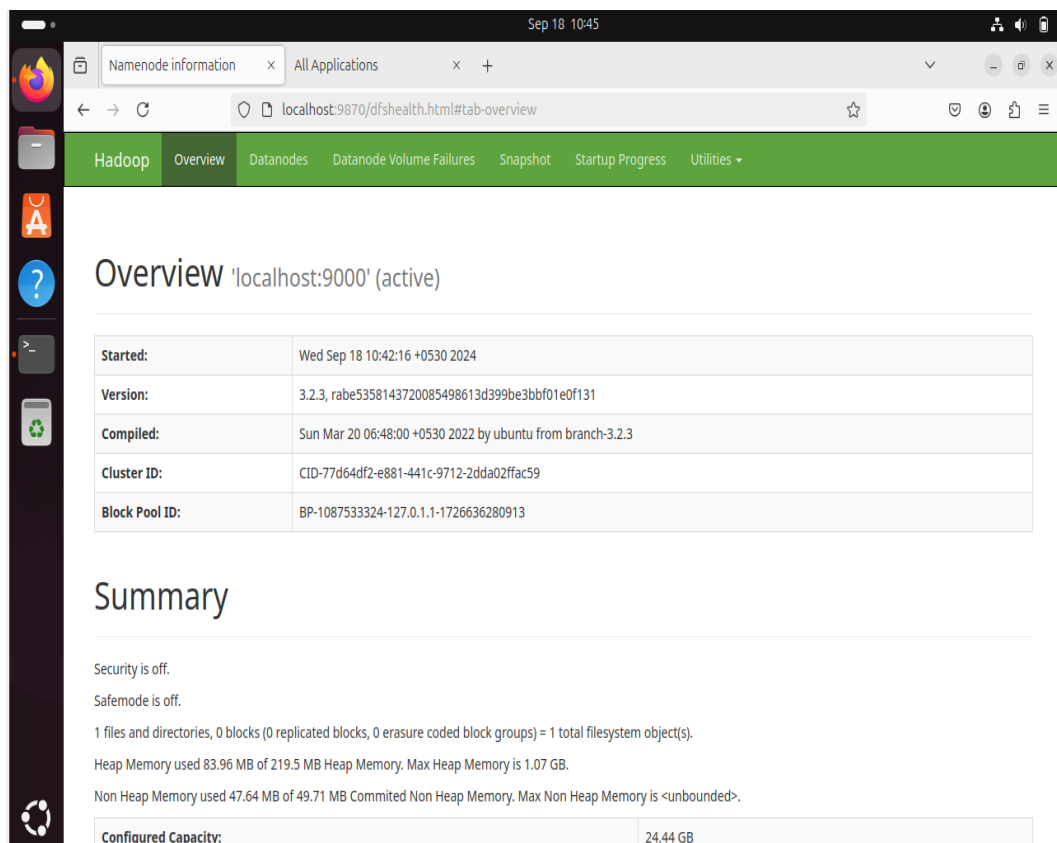
```

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>



Sep 18 10:45

Namenode information x All Applications x +

localhost:9870/dfshealth.html#tab-overview

Hadoop Overview Datanodes Datanode Volume Failures Snapshot Startup Progress Utilities

Overview 'localhost:9000' (active)

Started:	Wed Sep 18 10:42:16 +0530 2024
Version:	3.2.3, rabe5358143720085498613d399be3bbf01e0f131
Compiled:	Sun Mar 20 06:48:00 +0530 2022 by ubuntu from branch-3.2.3
Cluster ID:	CID-77d64df2-e881-441c-9712-2dda02ffac59
Block Pool ID:	BP-1087533324-127.0.1.1-1726636280913

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 83.96 MB of 219.5 MB Heap Memory. Max Heap Memory is 1.07 GB.

Non Heap Memory used 47.64 MB of 49.71 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-serverip:8088>. You should see the following screen: <http://192.168.16:8088>

The screenshot shows the Hadoop Namenode web interface. The top navigation bar includes 'Namenode information' and 'All Applications'. The main content area is titled 'All Applications'. On the left, there is a sidebar with a 'Cluster' section containing links for 'About', 'Nodes', 'Node Labels', 'Applications', and a 'Scheduler' section. The main content area displays several metrics tables:

- Cluster Metrics:** A table with columns for Apps Submitted, Apps Pending, Apps Running, Apps Completed, Containers Running, Used Resources, and Total Resources. All values are 0.
- Cluster Nodes Metrics:** A table with columns for Active Nodes, Decommissioning Nodes, Decommissioned Nodes, and Lost Nodes. Active Nodes is 1, and the others are 0.
- Scheduler Metrics:** A table with columns for Scheduler Type, Scheduling Resource Type, Minimum Allocation, and Maximum Allocation. The scheduler is Capacity Scheduler, and the resource type is [memory-mb (unit=M), vcores].
- Applications Table:** A table with columns for ID, User, Name, Application Type, Queue, Application Priority, StartTime, LaunchTime, FinishTime, State, FinalStatus, Running Containers, Allocated CPU VCoers, and Allocated Memory MB. The table is empty, showing 'No data available in table'.

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
```

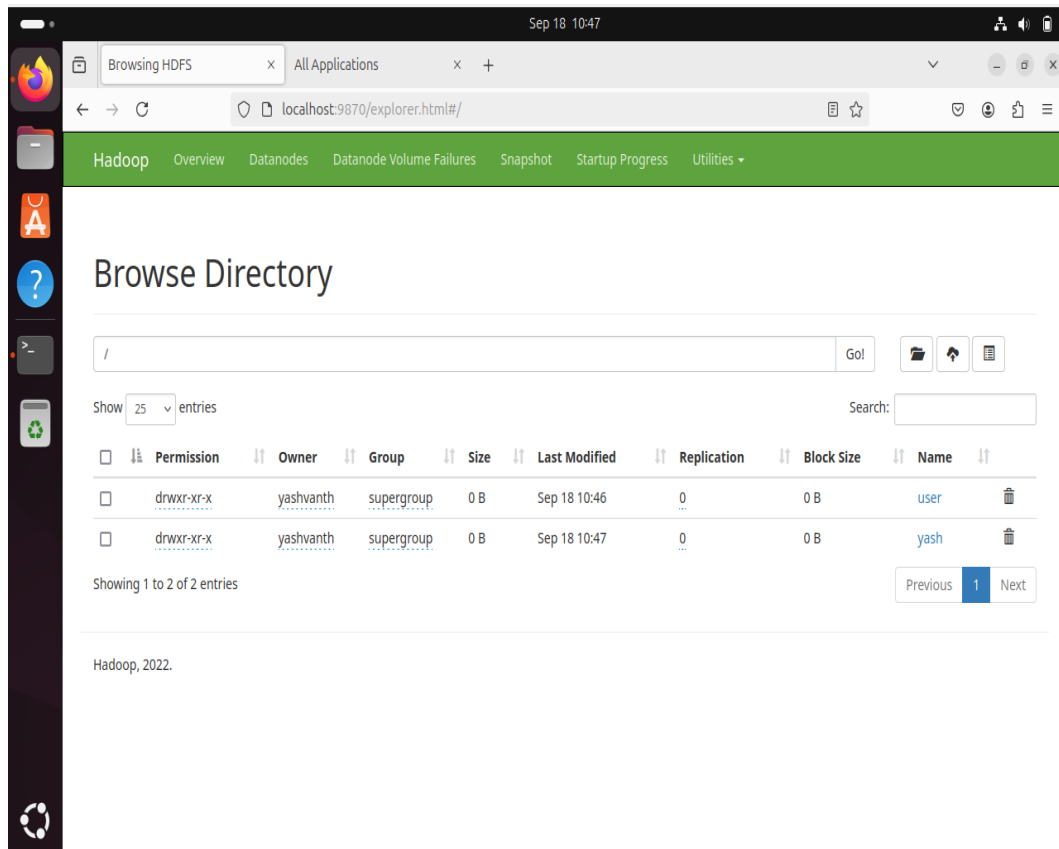
Next, run the following command to list the above directory:

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.

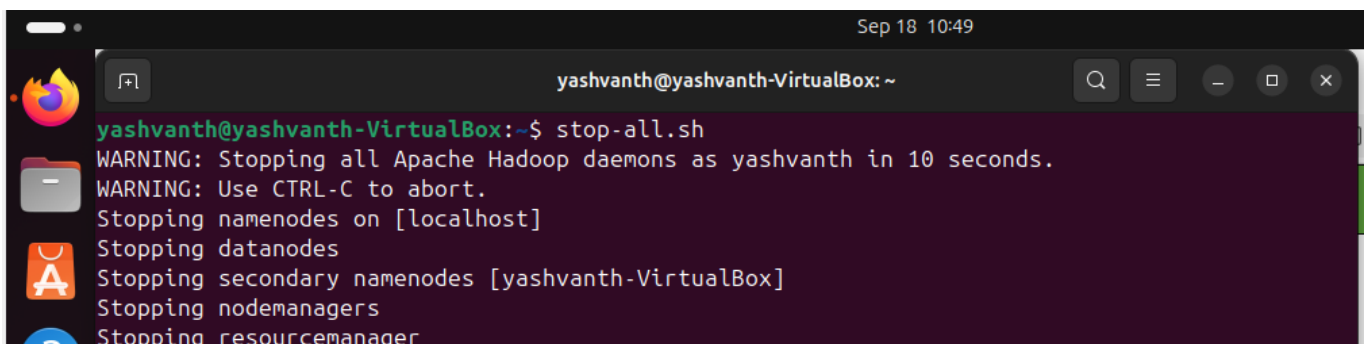
Go to the web interface, click on the Utilities => Browse the file system. You should see your directories which you have created earlier in the following screen:



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 Ubutu linux system have been successfully completed.