

Exp No: 1

INSTALLATION OF HADOOP

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
```

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:

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.

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:

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:
- Then, edit the mapred-site.xml file:
\$nano \$HADOOP_HOME/etc/hadoop/mapred-site.xml

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

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
```

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

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>

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

<http://192.168.16:8088>

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:

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:

SCREENSHOTS

```
Apache Hadoop Distribution
2024-08-13 22:24:48,714 INFO handler.ContextHandler: Stopped o.e.j.s.ServletContextHandler@47605f2f[logs,/logs,file:///D:/Softwares/Hadoop/hadoop-3.4.0.tar/hadoop-3.4.0/logs/,STOPPED]
2024-08-13 22:24:48,727 INFO impl.MetricsSystemImpl: Stopping NameNode metrics system...
2024-08-13 22:24:48,727 INFO impl.MetricsSystemImpl: NameNode metrics system stopped.
2024-08-13 22:24:48,727 INFO impl.MetricsSystemImpl: NameNode metrics system shutdown complete.
2024-08-13 22:24:48,727 ERROR namenode.NameNode: Failed to start namenode.
org.apache.hadoop.hdfs.server.common.InconsistentFSStateException: Directory D:\hadoop-3.3.1\data\namenode is in an inconsistent state: storage directory does not exist or is not accessible.
    at org.apache.hadoop.hdfs.server.namenode.FSImage.recoverStorageDirs(FSImage.java:392)
    at org.apache.hadoop.hdfs.server.namenode.FSImage.recoverTransitionRead(FSImage.java:243)
    at org.apache.hadoop.hdfs.server.namenode.FSNamesystem.loadFSImage(FSNamesystem.java:1275)
    at org.apache.hadoop.hdfs.server.namenode.FSNamesystem.loadFromDisk(FSNamesystem.java:834)
    at org.apache.hadoop.hdfs.server.namenode.NameNode.loadNamesystem(NameNode.java:817)
    at org.apache.hadoop.hdfs.server.namenode.NameNode.initialize(NameNode.java:984)
    at org.apache.hadoop.hdfs.server.namenode.NameNode.<init>(NameNode.java:1158)
    at org.apache.hadoop.hdfs.server.namenode.NameNode.<init>(NameNode.java:1133)
    at org.apache.hadoop.hdfs.server.namenode.NameNode.createNameNode(NameNode.java:1906)
    at org.apache.hadoop.hdfs.server.namenode.NameNode.main(NameNode.java:1971)
2024-08-13 22:24:48,730 INFO util.ExitUtil: Exiting with status 1: org.apache.hadoop.hdfs.server.common.InconsistentFSStateException: Directory D:\hadoop-3.3.1\data\namenode is in an inconsistent state: storage directory does not exist or is not accessible.
2024-08-13 22:24:48,730 INFO namenode.NameNode: SHUTDOWN_MSG:
/*****
SHUTDOWN_MSG: Shutting down NameNode at DESKTOP-SBBT9TG/192.168.56.1
*****/

D:\Softwares\Hadoop\hadoop-3.4.0.tar\hadoop-3.4.0\sbin>start-yarn.cmd
starting yarn daemons

D:\Softwares\Hadoop\hadoop-3.4.0.tar\hadoop-3.4.0\sbin>
```

```

Apache Hadoop Distribution
retry policy is RetryUpToMaximumCountWithFixedSleep(maxRetries=10, sleepTime=1000 MILLISECONDS)
2024-08-13 22:25:54,318 INFO ipc.Client: Retrying connect to server: localhost/127.0.0.1:9000. Already tried 8 time(s);
retry policy is RetryUpToMaximumCountWithFixedSleep(maxRetries=10, sleepTime=1000 MILLISECONDS)
2024-08-13 22:25:57,344 INFO ipc.Client: Retrying connect to server: localhost/127.0.0.1:9000. Already tried 9 time(s);
retry policy is RetryUpToMaximumCountWithFixedSleep(maxRetries=10, sleepTime=1000 MILLISECONDS)
2024-08-13 22:25:59,367 WARN datanode.DataNode: Problem connecting to server: localhost/127.0.0.1:9000
2024-08-13 22:26:07,485 INFO ipc.Client: Retrying connect to server: localhost/127.0.0.1:9000. Already tried 0 time(s);
retry policy is RetryUpToMaximumCountWithFixedSleep(maxRetries=10, sleepTime=1000 MILLISECONDS)
2024-08-13 22:26:19,400 INFO ipc.Client: Retrying connect to server: localhost/127.0.0.1:9000. Already tried 1 time(s);
retry policy is RetryUpToMaximumCountWithFixedSleep(maxRetries=10, sleepTime=1000 MILLISECONDS)
2024-08-13 22:26:13,467 INFO ipc.Client: Retrying connect to server: localhost/127.0.0.1:9000. Already tried 2 time(s);
retry policy is RetryUpToMaximumCountWithFixedSleep(maxRetries=10, sleepTime=1000 MILLISECONDS)
2024-08-13 22:26:16,523 INFO ipc.Client: Retrying connect to server: localhost/127.0.0.1:9000. Already tried 3 time(s);
retry policy is RetryUpToMaximumCountWithFixedSleep(maxRetries=10, sleepTime=1000 MILLISECONDS)
2024-08-13 22:26:19,556 INFO ipc.Client: Retrying connect to server: localhost/127.0.0.1:9000. Already tried 4 time(s);
retry policy is RetryUpToMaximumCountWithFixedSleep(maxRetries=10, sleepTime=1000 MILLISECONDS)
2024-08-13 22:26:22,683 INFO ipc.Client: Retrying connect to server: localhost/127.0.0.1:9000. Already tried 5 time(s);
retry policy is RetryUpToMaximumCountWithFixedSleep(maxRetries=10, sleepTime=1000 MILLISECONDS)
2024-08-13 22:26:25,639 INFO ipc.Client: Retrying connect to server: localhost/127.0.0.1:9000. Already tried 6 time(s);
retry policy is RetryUpToMaximumCountWithFixedSleep(maxRetries=10, sleepTime=1000 MILLISECONDS)
2024-08-13 22:26:28,675 INFO ipc.Client: Retrying connect to server: localhost/127.0.0.1:9000. Already tried 7 time(s);
retry policy is RetryUpToMaximumCountWithFixedSleep(maxRetries=10, sleepTime=1000 MILLISECONDS)
2024-08-13 22:26:31,711 INFO ipc.Client: Retrying connect to server: localhost/127.0.0.1:9000. Already tried 8 time(s);
retry policy is RetryUpToMaximumCountWithFixedSleep(maxRetries=10, sleepTime=1000 MILLISECONDS)
2024-08-13 22:26:34,739 INFO ipc.Client: Retrying connect to server: localhost/127.0.0.1:9000. Already tried 9 time(s);
retry policy is RetryUpToMaximumCountWithFixedSleep(maxRetries=10, sleepTime=1000 MILLISECONDS)
2024-08-13 22:26:36,757 WARN datanode.DataNode: Problem connecting to server: localhost/127.0.0.1:9000
2024-08-13 22:26:44,888 INFO ipc.Client: Retrying connect to server: localhost/127.0.0.1:9000. Already tried 0 time(s);
retry policy is RetryUpToMaximumCountWithFixedSleep(maxRetries=10, sleepTime=1000 MILLISECONDS)

```

```


Apache Hadoop Distribution
2024-08-13 22:25:34,931 INFO util.JvmPauseMonitor: Starting JVM pause monitor
2024-08-13 22:25:34,974 INFO ipc.CallQueueManager: Using callQueue: class java.util.concurrent.LinkedBlockingQueue, queueCapacity: 50000, scheduler: class org.apache.hadoop.ipc.DefaultRpcScheduler, ipcBackoff: false, ipcFailOver: false.
2024-08-13 22:25:34,988 INFO ipc.Server: Listener at 0.0.0.0:8030
2024-08-13 22:25:34,989 INFO pb.RpcServerFactoryPBImpl: Adding protocol org.apache.hadoop.yarn.api.ApplicationMasterProtocol to the server
2024-08-13 22:25:34,989 INFO ipc.Server: IPC Server Listener on 8030: starting
2024-08-13 22:25:35,134 INFO ipc.CallQueueManager: Using callQueue: class java.util.concurrent.LinkedBlockingQueue, queueCapacity: 50000, scheduler: class org.apache.hadoop.ipc.DefaultRpcScheduler, ipcBackoff: false, ipcFailOver: false.
2024-08-13 22:25:35,134 INFO ipc.Server: Listener at 0.0.0.0:8032
2024-08-13 22:25:35,134 INFO ipc.Server: Starting Socket Reader #1 for port 8032
2024-08-13 22:25:35,147 INFO pb.RpcServerFactoryPBImpl: Adding protocol org.apache.hadoop.yarn.api.ApplicationClientProtocol to the server
2024-08-13 22:25:35,147 INFO ipc.Server: IPC Server Responder: starting
2024-08-13 22:25:35,197 INFO ipc.Server: IPC Server Listener on 8032: starting
2024-08-13 22:25:36,299 INFO webproxy.ProxyCA: Created Certificate for OU=YARN-3fd21288-c809-4482-b2b1-c918b2ae3433
2024-08-13 22:25:36,380 INFO recovery.RMStateStore: Storing CA Certificate and Private Key
2024-08-13 22:25:36,380 INFO ResourceManager: ResourceManager: Transitioned to active state
2024-08-13 22:25:51,212 INFO ResourceManager.ResourceTrackerService: NodeManager from node DESKTOP-SBBT9TG:52991 (httpPort: 8042) registered with capability: <memory:8192, vCores:8>, assigned nodeId DESKTOP-SBBT9TG:52991
2024-08-13 22:25:51,222 INFO rmnode.RMNodeLocal: DESKTOP-SBBT9TG:52991 Node Transitioned from NM to RUNNING
2024-08-13 22:25:51,244 INFO capacity.AbstractLeafQueue: LeafQueue: root.default update max app related, maxApplications=10000, maxApplicationsPerUser=10000, Abs Cap:1.0, Cap: 1.0, MaxCap: 1.0
2024-08-13 22:25:51,244 INFO capacity.CapacityScheduler: Added node DESKTOP-SBBT9TG:52991 clusterResource: <memory:8192, vCores:8>
2024-08-13 22:25:51,244 INFO capacity.AbstractLeafQueue: LeafQueue: root.default update max app related, maxApplications=10000, maxApplicationsPerUser=10000, Abs Cap:1.0, Cap: 1.0, MaxCap: 1.0

```

```

Apache Hadoop Distribution
INFO: Registering org.apache.hadoop.yarn.server.nodemanager.webapp.JAXContextResolver as a provider class
Aug 13, 2024 10:25:44 PM com.sun.jersey.server.impl.application.WebApplicationImpl _initiate
INFO: Initiating Jersey application, version 'Jersey: 1.19.4 05/24/2017 03:20 PM'
Aug 13, 2024 10:25:44 PM com.sun.jersey.guice.spi.container.GuiceComponentProviderFactory getComponentProvider
INFO: Binding org.apache.hadoop.yarn.server.nodemanager.webapp.JAXContextResolver to GuiceManagedComponentProvider with the scope "Singleton"
Aug 13, 2024 10:25:45 PM com.sun.jersey.guice.spi.container.GuiceComponentProviderFactory getComponentProvider
INFO: Binding org.apache.hadoop.yarn.webapp.GenericExceptionHandler to GuiceManagedComponentProvider with the scope "Singleton"
Aug 13, 2024 10:25:45 PM com.sun.jersey.guice.spi.container.GuiceComponentProviderFactory getComponentProvider
INFO: Binding org.apache.hadoop.yarn.server.nodemanager.webapp.NMWebServices to GuiceManagedComponentProvider with the scope "Singleton"
2024-08-13 22:25:45,534 INFO handler.ContextHandler: Started o.e.j.w.WebAppContext@ba1bb1d[/node, / file:///C:/Users/Hares/AppData/Local/Temp/jetty-0.0.0-8042-hadoop-yarn-common-3.4.0-jar--any-3589818351802933540/webapp, AVAILABLE]{jar:file:/D:/Software/hadoop/hadoop-3.4.0.tar/hadoop-3.4.0/share/hadoop/yarn/hadoop-yarn-common-3.4.0.jar/webapps/node}
2024-08-13 22:25:45,558 INFO server.AbstractConnector: Started ServerConnector@4c4db8504[HTTP/1.1, (http/1.1)]{0.0.0.0:8042}
2024-08-13 22:25:45,558 INFO server.Server: Started @28007ms
2024-08-13 22:25:45,569 INFO webapp.WebApps: Web app node started at 8042
2024-08-13 22:25:45,569 INFO nodemanager.NodeStatusUpdaterImpl: Node ID assigned is : DESKTOP-SBBT9TG:52991.
2024-08-13 22:25:45,565 INFO util.JvmPauseMonitor: Starting JVM pause monitor
2024-08-13 22:25:45,723 INFO client.DefaultHMRMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8031.
2024-08-13 22:25:45,806 INFO nodemanager.NodeStatusUpdaterImpl: Running Applications Size : 0.
2024-08-13 22:25:51,244 INFO security.NMContainerTokenSecretManager: Rolling master-key for container-tokens, got key with id -165124843
2024-08-13 22:25:51,244 INFO security.NMTokenSecretManagerInNM: Rolling master-key for container-tokens, got key with id 88792469
2024-08-13 22:25:51,244 INFO nodemanager.NodeStatusUpdaterImpl: Registered with ResourceManager as DESKTOP-SBBT9TG:52991 with total resource of <memory:8192, vCores:8>

```



Cluster

About Nodes

Node Labels

Applications

NEW SUBMITTED

ACCEPTED

FINISHED

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	Decommissioned Nodes
1	0	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Min Capacity Scheduler	Max Capacity Scheduler
	[memory-mb (unit=M), vcores]		<memory:1024, vCores:1>

Showing 0 to 0 of 0 entries

ID	User	Name	Application Type	Application Tags	Queue	Application Priority	StartTime	LaunchTime	FinishTime
Showing 0 to 0 of 0 entries									

