Hadoop Installation Guide and HDFS Hands On - H1

Every step is to be executed on the home directory. Use `cd` to move to home directory.

The commands in the guide use `$USER` as the notation for your username.

Change any `/home/$USER/` to `/home/<your SRN>/`

Execute the following commands to move to the home directory and updating the package list and the system. This guide assumes that you are working with Ubuntu or a Debian based distribution.

```
cd
sudo apt update -y
sudo apt upgrade -y
```

Step 1 - Installing Java

Since Hadoop 3.3.3 may not support newer versions of Java, we install Java 8 using the following command.

```
sudo apt install openjdk-8-jdk -y
```

Check if Java is successfully installed and the version with the following commands.

```
java -version
javac -version
```

Step 2 - Setup passwordless SSH for Hadoop

We install the following packages to allow us to setup an ssh server on the system as well as a client to remote into it with the following commands.

```
sudo apt install openssh-server openssh-client -y
```

Enable passwordless SSH

Generate an SSH key pair and define the location is is to be stored in id_rsa. Then use the cat command to store the public key as authorized_keys in the ssh directory. Follow these exact commands with change in permissions.

```
ssh-keygen -t rsa -P '' -f ~/.ssh/id_rsa
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
chmod 0600 ~/.ssh/authorized_keys
```

Verify passwordless SSH is setup and working with

```
ssh localhost
```

If the above command does not ask you for a password, you have successfully setup passwordless SSH.

Type `exit` or press `Ctrl+d` to quit the SSH session.

Step 3 - Downloading Hadoop

Use the link given below to download and extract hadoop using the following commands.

```
cd
wget https://dlcdn.apache.org/hadoop/common/hadoop-3.3.3/hadoop-
3.3.3.tar.gz
tar xzf /home/$USER/hadoop-3.3.3.tar.gz
```

Step 4 - Single Node Deployment

The current setup is called pseudo-distributed mode, allows each Hadoop daemon to run as a single Java process. A Hadoop environment is configured by editing the following list of configuration files:

- .bashrc
- hadoop-env.sh
- core-site.xml
- hdfs-site.xml
- mapred-site-xml
- yarn-site.xml

Before editing the above mentioned files, we need to make a few directories for our namenodes and datanodes along with the required permissions.

```
cd
mkdir dfsdata
mkdir tmpdata
mkdir dfsdata/datanode
mkdir dfsdata/namenode
```

Change permissions for the directories using the following commands. Remember to replace `$USER` with your username.

```
sudo chown -R $USER:$USER /home/$USER/dfsdata/
sudo chown -R $USER:$USER /home/$USER/dfsdata/datanode/
sudo chown -R $USER:$USER /home/$USER/dfsdata/namenode/
```

Editing and Setting up the ~/.bashrc config file

Open .bashrc with any text editor of your choice. This guide recommends using `nano`.

```
sudo nano ~/.bashrc
```

Scroll to the bottom of the file. Copy and paste the below mentioned statements to the end of the file.

```
#Hadoop Path Configs
export HADOOP_HOME=/home/$USER/hadoop-3.3.3
export HADOOP_INSTALL=$HADOOP_HOME
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$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"
```

Press `Ctrl+o` to save and `Ctrl+x` to exit nano . Apply changes to `bash` with the following command.

```
source ~/.bashrc
```

You can verify if the changes have been made by using the `echo` command and checking if the corresponding path gets printed in the terminal.

```
echo $HADOOP_HOME
echo $PATH
```

Setup hadoop-env.sh

Open the file with

```
sudo nano $HADOOP_HOME/etc/hadoop/hadoop-env.sh
```

Scroll down until you find the commented line `# export JAVA_HOME=`. Uncomment the line and replace the path with your Java path. The final line should look like this

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
```

Save and exit the file as shown previously.

Setup core-site.xml

Open the file with

```
sudo nano $HADOOP_HOME/etc/hadoop/core-site.xml
```

Replace the existing configuration tags with the following

```
<configuration>
<property>
  <name>hadoop.tmp.dir</name>
  <value>/home/$USER/tmpdata</value>
</property>
<property>
  <name>fs.default.name</name>
  <value>hdfs://127.0.0.1:9000</value>
</property>
</configuration>
```

Save and exit the file.

Setup hdfs-site.xml

Open the file using

```
sudo nano $HADOOP_HOME/etc/hadoop/hdfs-site.xml
```

Replace the existing configuration tags with the following

```
<configuration>
<property>
  <name>dfs.name.dir</name>
  <value>/home/$USER/dfsdata/namenode</value>
</property>
<property>
  <name>dfs.data.dir</name>
  <value>/home/$USER/dfsdata/datanode</value>
</property>
<property>
  <name>dfs.replication</name>
  <value>1</value>
</property>
</configuration>
```

Save and exit the file after making all the changes.

Setup mapred-site.xml

Open the file with

```
sudo nano $HADOOP_HOME/etc/hadoop/mapred-site.xml
```

Replace the existing configuration tags with the following

```
<configuration>
<property>
  <name>mapreduce.framework.name</name>
  <value>yarn</value>
</property>
</configuration>
```

Save and exit the file.

Setup yarn-site.xml

Open the file with

```
sudo nano $HADOOP_HOME/etc/hadoop/yarn-site.xml
```

Replace the existing configuration tags with the following

```
<configuration>
<property>
  <name>yarn.nodemanager.aux-services</name>
  <value>mapreduce_shuffle</value>
</property>
<property>
  <name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
  <value>org.apache.hadoop.mapred.ShuffleHandler</value>
</property>
<property>
  <name>yarn.resourcemanager.hostname</name>
  <value>127.0.0.1</value>
</property>
<property>
  <name>yarn.acl.enable</name>
  <value>0</value>
</property>
<property>
  <name>yarn.nodemanager.env-whitelist</name>

<value>JAVA_HOME,HADOOP_COMMON_HOME,HADOOP_HDFS_HOME,HADOOP_CONF_DIR,CLA
SSPATH_PERPEND_DISTCACHE,HADOOP_YARN_HOME,HADOOP_MAPRED_HOME</value>
</property>
</configuration>
```

Save and exit the file.

Step 5 - Format HDFS NameNode

Before starting Hadoop for the first time, the namenode must be formatted. Use the following command.

```
hdfs namenode -format
```

A `SHUTDOWN` message will signify the end of the formatting process.

If you have reached this stage, it signifies that you have successfully installed hadoop.

Step 6 - Starting Hadoop

Navigate to the `hadoop` folder and execute the following commands. `start-all.sh` is a shell script that is used to start all the processes that hadoop requires.

```
cd
cd hadoop-3.3.3/sbin/
./start-all.sh
```

Type `jps` to find all the Java Processes started by the shell script. You should see a total of 6 processes, including the `jps` process. Note that the order of the items and the process IDs will be different.

```
2994 DataNode
3219 SecondaryNameNode
3927 Jps
3431 ResourceManager
2856 NameNode
3566 NodeManager
```

Step 7 - Accessing Hadoop from the Browser

You can access Hadoop on `localhost` on the following ports

- NameNode - http://localhost:9870
- DataNode - http://localhost:9864
- YARN Manager - http://localhost:8088

To stop all processes when you are done with your work, execute the following command.

```
cd
cd hadoop-3.3.3/sbin/
./stop-all.sh
```