Introduction to Hadoop Distributed System

It is an open-source software framework that supports data-intensive distributed applications, licensed under Apache v2 license. It is a flexible and highly-available architecture for large-scale computation and data processing on a network of commodity hardware.

Install JDK on Ubuntu

Before installing the Java Runtime Environment (JRE) and Java Development Kit (JDK) update and upgrade packages in the Ubuntu system.

```
sudo apt-get update
```

```
sudo apt-get upgrade
```

The easiest option for installing Java is to use the version packaged with Ubuntu. By default, Ubuntu includes OpenJDK, which is an open-source variant of the JRE and JDK. Check if Java is already installed:

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

If Java is not currently installed, then run the following command:

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

Verify the installation with:

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

Install OpenSSH on Ubuntu

The ssh command provides a secure encrypted connection between two hosts over an insecure network. This connection can also be used for terminal access, file transfers, and for executing commands on the remote machine. Install the OpenSSH server and client using the following command:

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

Add users for Hadoop Environment

Create the new user and group using the command :

```
sudo adduser hadoop
```

Providesudo permission to newly created user :

```
sudo adduser hduser sudo
```

Switch to the newly created user and enter the corresponding password:

```
su - hadoop
```

To check and verify present user login:

```
whoami
```

Configure passwordless SSH for Hadoop User

The user will able to SSH to the localhost without being prompted for a password. Generate Public and Private Key Pairs with the following command:

```
ssh-keygen -t rsa -P " -f ~/.ssh/id_rsa
```

Copy the public keys from id_rsa.pub to authorized_keys.

```
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
```

Set the permissions for your user with the chmod command:

```
chmod 0600 ~/.ssh/authorized_keys
```

The new user is now able to SSH without needing to enter a password every time. Verify everything is set up correctly by using the hadoop user to SSH to localhost:

```
ssh localhost

exit
```

After an initial prompt, the Hadoop user is now able to establish an SSH connection to the localhost seamlessly.

Download and Install Hadoop on Ubuntu

Visit the official Apache Hadoop project page, and select the version of Hadoop you want to implement. Select 3.3.6 version

https://hadoop.apache.org/releases.html

Version	Release date	Source download	Binary download	Release notes
3.3.6	2023 Jun 23	source (checksum signature)	binary (checksum signature) binary-aarch64 (checksum signature)	Announcement
3.2.4	2022 Jul 22	source (checksum signature)	binary (checksum signature)	Announcement
2.10.2	2022 May 31	source (checksum signature)	binary (checksum signature)	Announcement

Select the latest 3.3.6 and click on binary to download. The steps outlined in this tutorial use the Binary download for Hadoop Version 3.3.6. It up to the user which version it wants to use installation steps will remain the same.

After Downloaded copy and paste to the Hadoop folder which is located in home

```
tar xzf hadoop-3.3.6.tar.gz

mv hadoop-3.3.6 hadoop

ls
```

Setting up the environment variables (bashrc)

Now Hadoop download process is completed and the next is to setup Configuration for Hadoop. Open bashrc file using any editor like vim, gedit, nano, etc., I prefer nano it is like any text editor of windows:

```
nano ~/.bashrc
```

Edit the bashrc file for the Hadoop user via setting up the following Hadoop environment variables at bottom of bashrc file:

```
#Hadoop Related Options
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 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"
Once add the variables, save and exit the .bashrc file.
```

To apply changes to the current environment run the following:

```
source ~/.bashrc
```

Configuration Changes in hadoop-env.sh file

For making changes in hadoop-env.sh it is required to get the path of java. To locate the correct Java path, run the following command in your terminal window:

```
which javac
readlink -f /usr/bin/javac
```

The section of the path just before the /bin/java directory needs to be assigned to the $JAVA_HOME variable in hadoop-env.sh file.

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

It is a need to define which Java implementation is to be utilized.

Configuration Changes in core-site.xml file

Open core-site.xml to make change in-between configuration.

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

Add the below configuration to override the default values for the temporary directory and add your HDFS URL to replace the default local file system setting:

```
<configuration>
<property>
  <name>fs.default.name</name>
  <value>hdfs://localhost:9000</value>
</property>
</configuration>
```

Configuration Changes in hdfs-site.xml file

Open hdfs-site.xml to make change in-between configuration.

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

Add the below configuration to the file and, if needed, adjust the NameNode and DataNode directories to your custom locations:

```
<configuration>
<property>
 <name>dfs.replication</name>
 <value>1</value>
</property>
<property>
  <name>dfs.name.dir</name>
  <value>file:///home/hadoop/hadoopdata/hdfs/namenode</value>
</property>
<property>
  <name>dfs.data.dir</name>
  <value>file:///home/hadoop/hadoopdata/hdfs/datanode</value>
</property>
</configuration>
```

Configuration Changes in mapred-site-xml file

Use the following command to access the mapred-site.xml file and define MapReduce values:

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

Add the below configuration to change the default MapReduce framework name value to yarn:

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

Configuration Changes in yarn-site.xml file

Open the yarn-site.xml file in a nano editor:

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

It is used to define settings relevant to YARN. It contains configurations for the Node Manager, Resource Manager, Containers, and Application Master. Add the following configuration to the file.

```
<configuration>
 <property>
  <name>yarn.nodemanager.aux-services</name>
  <value>mapreduce_shuffle</value>
 </property>
</configuration>
```

Format NameNode

It is important to format the NameNode before starting Hadoop services for the first time. Now format the namenode using the following command, make sure that Storage directory is :

```
cd ~/hadoop/sbin
hdfs namenode -format
```

Start Hadoop Cluster

Navigate to the hadoop/sbin directory and execute the following commands to start the NameNode and DataNode:

```
./start-dfs.sh
```

It will take a few second to start and generate output

Once the namenode, datanodes, and secondary namenode are up and running, start the YARN resource and nodemanagers by typing:

```
./start-yarn.sh
```

As with the previous command, the output informs you that the processes are starting.

To verify all the Hadoop services/daemons are started successfully you can use the jps command.

```
Jps
```

Access Hadoop UI from Browser

You can access both the Web UI for NameNode and YARN Resource Manager via any of the browsers like Google Chrome/Mozilla Firefox.

Hadoop NameNode started on default port 9870.

http://localhost:9870/

Now access port 8042 for getting the information about the cluster and all applications.

http://localhost:8042/

Access port 9864 to get details about your Hadoop node.

http://localhost:9864/