How to Install Hadoop on Ubuntu

Elaborated by Yudith Cardinale

Step 1: Install Java

The default Ubuntu repositories contain both Java 8 and Java 11. Use the following command to install it.

```
sudo apt update && sudo apt install openjdk-11-jdk  (or openjdk-11-jdk)
```

Once you have successfully installed it, check the current Java version:

```
java -version
```

```
yudithcardinale$ java -version

openjdk version "1.8.0_292"
OpenJDK Runtime Environment (AdoptOpenJDK)(build 1.8.0_292-b10)
OpenJDK 64-Bit Server VM (AdoptOpenJDK)(build 25.292-b10, mixed mode)
```

You can find the location of the JAVA_HOME directory by running the following command. That will be required for the configuration.

```
dirname $(dirname $(readlink -f $(which java)))

yudithcardinale$ dirname $(dirname $(readlink -f $(which java)))

/usr/lib/jvm/java-8-openjdk-amd64
```

Step 2: Configure ssh password-less

```
ssh-keygen -t rsa -P ""
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
```

Change permissions

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

After this step open terminal and enter “ssh localhost”, you should log in without a password and that indicates your settings is successful

```
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. Answer yes

Step 3: Install Hadoop

Use the following command to download Hadoop 3.3.4:

```
wget https://dlcdn.apache.org/hadoop/common/hadoop-3.3.4/hadoop-3.3.4.tar.gz
```

Once you've downloaded the file, you can unzip it to a folder on your hard drive.

```
tar xzf hadoop-3.3.4.tar.gz
```

Rename the extracted folder to remove version information. This is an optional step, but if you don't want to rename, then adjust the remaining configuration paths.

```
mv hadoop-3.3.4 hadoop
```

Next, you will need to configure Hadoop and Java Environment Variables on your system. Open the ~/.bashrc file in your favorite text editor:

```
vim ~/.bashrc
```

Append the below lines to the file. You can find the JAVA_HOME location by running dirname $ (dirname $(readlink -f $(which java))) command on the terminal.

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
export HADOOP_HOME=/home/ycardinale/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 the file and close it.

Load the above configuration in the current environment.

```
source ~/.bashrc
```

You also need to configure **JAVA_HOME** in **hadoop-env.sh** file. Edit the Hadoop environment variable file in the text editor:

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

Search for the “export JAVA_HOME” and configure it with the value found in step 1. Add the following line (according to the Java path):

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

Until this step, you have the standalone configuration of hadoop.

Step 6: Pseudo-distributed hadoop configuration

For the pseudo-distributed operating mode, we need to configure the following files available under the etc directory.

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:

```
mkdir -p ~/hadoopdata/hdfs/{namenode,datanode}
```

Next, edit the **core-site.xml** file and update with your system hostname:

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

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

Hdfs-site.xml

Open $HADOOP_HOME/etc/hadoop/Hdfs-site.xml file in terminal and add below properties

```
<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>
```

Yarn-site.xml

Open $HADOOP_HOME/etc/hadoop/yarn-site.xml file and add below properties

```
<configuration>
<!-- Site specific YARN configuration properties →
   <property>
     <name>yarn.nodemanager.aux-services</name>
     <value>mapreduce_shuffle</value>
   </property>
   <property>
     <name>yarn.nodemanager.env-whitelist</name>
<value>JAVA_HOME,HADOOP_COMMON_HOME,HADOOP_HDFS_HOME,HADOOP_CONF_DIR,CLASSPATH_PRE
PEND_DISTCACHE,HADOOP_YARN_HOME,HADOOP_MAPRED_HOME</value>
   </property>
</configuration>
```

Mapred-site.xml

Open $HADOOP_HOME/etc/hadoop/mapred-site.xml file in termial and add below properties.

```
<configuration>
<property>
  <name>mapreduce.framework.name</name>
  <value>yarn</value>
</property>
<property>
  <name>mapreduce.application.classpath</name>
  <value>$HADOOP_MAPRED_HOME/share/hadoop/mapreduce/*:$HADOOP_MAPRED_HOME/share/
hadoop/mapreduce/lib/*</value>
  </property>
</configuration>
```

hdfs format

```
$HADOOP_HOME/bin/hdfs namenode -format
```

Note: Open terminal and Initialize Hadoop cluster by formatting HDFS directory

Step 7: Final Step

Run start-all.sh in the sbin folder

```
$HADOOP_HOME/sbin/start-all.sh

Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [K9-MAC-061.local]
Starting resourcemanager
Starting nodemanagers
```

Use JPS command to check if all name node, Data node, resource manager is started successfully

```
4929 DataNode
5294 NodeManager
5200 ResourceManager
5354 Jps
5046 SecondaryNameNode
4831 NameNode
```

How to Access Hadoop web interfaces (Hadoop Health)

```
NameNode                  : http://localhost:9870
NodeManager               : http://localhost:8042
Resource Manager (Yarn)   : http://localhost:8088/cluster
```