Hadoop 2 - Pseudonode Installation

This document has been created for following environment:

- Ubuntu Linux 64-bit
- JDK jdk1.8.0_05
- Hadoop 2.x stable release

Note: In this document we have used only compatible versions of Hadoop ecosystem tools or software downloaded from the official Apache hadoop website. Preferably use a stable release of the particular tool.

Prerequisites:

- Installing Java v1.8
- Configuring SSH access.

1) Installing Java

Hadoop is a framework written in Java for running applications on large clusters of commodity hardware. Hadoop needs Java 6 or above to work.

Step 1: Download Jdk tar.gz file for linux-64 bit, extract it (you can also do this by right click → extract here)

```
atagtr2019@msmsivam[]# cd /home/student/Downloads

atagtr2019@msmsivam[]# tar xvpzf /home/student/Downloads/jdk-8u5-linux-x64.tar.gz

atagtr2019@msmsivam[]# cd /home/student/Downloads/jdk1.8.0_05
```

Step 2:

- ✓ Open the "~/.bashrc" file and Add the following line as per the version
- ✓ set a environment for Java
- ✓ The 'profile' file contains commands that ought to be run for login shells

```
atagtr2019@msmsivam[]# gedit ~/.bashrc

#--insert JAVA_HOME
```

JAVA_HOME=/home/student/Downloads/jdk1.8.0_05

#--in PATH variable just append at the end of the line

PATH=$PATH:$JAVA_HOME/bin

#--Append JAVA_HOME at end of the export statement

export PATH JAVA_HOME

Step 3: Source the ~/.bashrc

Step 3: Update the java alternatives

- ✓ By default OS will have a open jdk. Check by “java -version”. You will be prompt “openJDK”
- ✓ If you also have openjdk installed then you'll need to update the java alternatives:
- ✓ If your system has more than one version of Java, configure which one your system causes by entering the following command in a terminal window
- ✓ By default OS will have a open jdk. Check by “java -version”. You will be prompt “Java HotSpot(TM) 64-Bit Server”

atagtr2019@msmsivam[]# update-alternatives --install "/usr/bin/java" java "/home/student/Downloads/jdk1.8.0_05/bin/java" 1

atagtr2019@msmsivam[]# update-alternatives --config java --type selection number:

atagtr2019@msmsivam[]# java –version

2) Configure ssh

- ✓ Hadoop requires SSH access to manage its nodes, i.e. remote machines plus your local machine if you want to use Hadoop on it (which is what we want to do in this short tutorial). For our single-node setup of Hadoop, we therefore need to configure SSH access to localhost.
- ✓ The need to create a Password-less SSH Key generation based authentication is so that the master node can then login to slave nodes (and the secondary node) to start/stop them easily without any delays for authentication
- ✓ If you skip this step, then have to provide password

Generate an SSH key for the user. Then Enable password-less SSH access to you.

If SSH is not installed, Please put the command to install:

#sudo apt-get install openssh-server

--You will be asked to enter password,

atagtr2019@msmsivam[]# ssh localhost

--enter password

atagtr2019@msmsivam[]# exit

atagtr2019@msmsivam []# ssh-keygen

atagtr2019@msmsivam []# ssh-copy-id -i localhost

--After above 2 steps, You will be connected without password,

atagtr2019@msmsivam []# ssh localhost

atagtr2019@msmsivam []# exit

3) Hadoop installation

- ✓ Now Download Hadoop from the official Apache, preferably a stable release version of Hadoop 2.5.x and extract the contents of the Hadoop package to a location of your choice.

Step 1: Download the tar.gz file of latest version Hadoop (hadoop-2.x) from the official site .

Step 2: Extract(untar) the downloaded file

atagtr2019@msmsivam[]# cd /home/student/Downloads

atagtr2019@msmsivam[/]# tar xvpzf /home /student /Downloads

/hadoop-2.7.0.tar.gz

atagtr2019t@msmsivam[/]# cd hadoop-2.7.0

Like java, update Hadoop environment variable in ~/.bashrc

atagtr2019@msmsivam[]# gedit ~/.bashrc

#--insert HADOOP_PREFIX

HADOOP_PREFIX=/home/student/Downloads/hadoop-2.7.0

#--in PATH variable just append at the end of the line

PATH=$PATH:$HADOOP_PREFIX/bin

#--Append HADOOP_PREFIX at end of the export statement

export PATH JAVA_HOME HADOOP_PREFIX

Step 3: Source the ~/.bashrc

atagtr2019@msmsivam[]# source ~/.bashrc

Verify Hadoop installation

atagtr2019@msmsivam[]# cd $HADOOP_PREFIX

atagtr2019@msmsivam[]# bin/hadoop version

3.1) Modify the Hadoop Configuration Files

- ✓ In this section, we will configure the directory where Hadoop will store its configuration files, the network ports it listens to, etc. Our setup will use Hadoop Distributed File System,(HDFS), even though we are using only a single local machine.
- ✓ Add the following properties in the various hadoop configuration files which is available under $HADOOP_PREFIX/etc/hadoop/
- ✓ core-site.xml, hdfs-site.xml, mapred-site.xml & yarn-site.xml

Update Java, hadoop path to the Hadoop environment file

atagtr2019@msmsivam[]# cd $HADOOP_PREFIX/etc/hadoop

atagtr2019@msmsivam[]# gedit hadoop-env.sh

Paste following line at beginning of the file

export JAVA_HOME=/home/student/Downloads/jdk1.8.0_05

export HADOOP_PREFIX=/home/student/Downloads/hadoop-2.7.0

Modify the core-site.xml

atagtr2019@msmsivam[]# cd $HADOOP_PREFIX/etc/hadoop

atagtr2019@msmsivam[]# gedit core-site.xml

Paste following between <configuration> tags

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

Re-initialize the base directories

```
atagtr2019@msmsivam# mkdir -p /home/student/hdata/
atagtr2019@msmsivam# mkdir -p /home/student/hdata/data
atagtr2019@msmsivam# mkdir -p /home/student/hdata/name
```

Modify the hdfs-site.xml

```
atagtr2019@msmsivam[]# gedit hdfs-site.xml
```

Paste following between <configuration> tags

```
<configuration>
    <property>
        <name>dfs.replication</name>
        <value>1</value>
    </property>
    <property>
        <name>dfs.namenode.name.dir</name>
        <value>/home/student/hdata/name</value>
    </property>
    <property>
        <name>dfs .datanode.data.dir</name>
        <value>/home/student/hdata/data</value>
```

```
        </property>
    </configuration>
```

YARN configuration - Single Node

Modify the mapred-site.xml

```
atagtr2019@msmsivam[]#cp mapred-site.xml.template mapred-site.xml
atagtr2019@msmsivam[]# gedit mapred-site.xml
```

Paste following between <configuration> tags

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

Modiy yarn-site.xml

```
atagtr2019@msmsivam[]# gedit yarn-site.xml
```

Paste following between <configuration> tags

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

Formatting the HDFS file-system via the NameNode

- ✓ The first step to starting up your Hadoop installation is formatting the Hadoop files system which is implemented on top of the local file system of our "cluster" which includes only our local machine. We need to do this the first time you set up a Hadoop cluster.
- ✓ Do not format a running Hadoop file system as you will lose all the data currently in the cluster (in HDFS)

```
atagtr2019@msmsivam[]# cd $HADOOP_PREFIX
atagtr2019@msmsivam[]# bin/hadoop namenode –format
```

Start NameNode daemon and DataNode daemon: (port 50070)

```
atagtr2019@msmsivam[]# sbin/start-dfs.sh
```

To know the running daemons jut type jps or $JAVA_HOME/bin/jps

Start ResourceManager daemon and NodeManager daemon: (port 8088)

```
atagtr2019@msmsivam[]# sbin/start-yarn.sh
```

To Stop all the services

```
atagtr2019@msmsivam[]# sbin/stop-all.sh
```