Hadoop 2 - Pseudo Node Installation

Getting Started

This tutorial has been created for following environment:

- ✓ Ubuntu Linux 64-bit
- ✓ JDK 1.8.0_05
- ✓ Hadoop 2.7.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:

1.Installing Java v1.8
2.Configuring SSH access.

sudo apt-get install vim

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-62 bit, extract it into “/usr/local”

```
boss@solaiv[]# cd /opt

boss@solaiv[]#  sudo tar xvpzf /home/itadmin/Downloads/jdk-8u5-linux-x64.tar.gz

boss@solaiv[]# cd /opt/jdk1.8.0_05
```

Step 2:

- ✓ Open the “/etc/profile” file and Add the following line as per the version
- ✓ set a environment for Java
- ✓ Use the root user to save the /etc/proflie or use gedit instead of vi .
- ✓ The 'profile' file contains commands that ought to be run for login shells

```
boss@solaiv[]# sudo vi  /etc/profile
```

```
#--insert JAVA_HOME
JAVA_HOME=/opt/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
```

save the file using by pressing “Esc” key followed by :wq!

Step 3: Source the /etc/profile

```
boss@solaiv[]# source /etc/profile
```

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”

```
boss@solaiv[]#  update-alternatives --install "/usr/bin/java" java "/opt/jdk1.8.0_05/bin/java" 1

boss@solaiv[]#  update-alternatives --config java
--type selection number:

boss@solaiv[]#  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 yo

sudo apt-get install openssh-server

```
--You will be asked to enter password,
      root@solaiv[]# ssh localhost

      root@solaiv[]# ssh-keygen
      root@solaiv[]# ssh-copy-id -i  localhost

--After above 2 steps, You will be connected without password,
      root@solaiv[]# ssh localhost
      root@solaiv[]# exit
```

3) Hadoop installation

✓ Now Download Hadoop from the official Apache, preferably a stable release version of Hadoop 2.7.x and extract the contents of the Hadoop package to a location of your choice.

✓ We chose location as “/opt/”

Step 1: Download the tar.gz file of latest version Hadoop (hadoop-2.7.x) from the official site .
Step 2: Extract(untar) the downloaded file from this commands to /opt/bigdata

```
    root@solaiv[]# cd /opt
    root@solaiv[/opt]# sudo tar xvpzf /home/itadmin/Downloads/hadoop-2.7.0.tar.gz
    root@solaiv[/opt]# cd  hadoop-2.7.0/
```

Like java, update Hadop environment variable in /etc/profile

```
         boss@solaiv[]# sudo vi  /etc/profile
```

```
        #--insert HADOOP_PREFIX
        HADOOP_PREFIX=/opt/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
```

save the file using by pressing “Esc” key followed by :wq!

Step 3: Source the /etc/profile

```
        boss@solaiv[]# source /etc/profile
```

Verify Hadoop installation

```
boss@solaiv[]# cd $HADOOP_PREFIX

boss@solaiv[]# 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

```
boss@solaiv[]#  cd $HADOOP_PREFIX/etc/hadoop

boss@solaiv[]# vi hadoop-env.sh
```

Paste following line at beginning of the file

```
export JAVA_HOME=/usr/local/jdk1.8.0_05

export HADOOP_PREFIX=/opt/hadoop-2.7.0
```

Modify the **core-site.xml**

```
boss@solaiv[]#  cd $HADOOP_PREFIX/etc/hadoop

boss@solaiv[]# vi core-site.xml
```

Paste following between <configuration> tags

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

Modify the **hdfs-site.xml**

```
boss@solaiv[]# vi hdfs-site.xml
```

Paste following between <configuration> tags

```
<configuration>

   <property>
          <name>dfs.replication</name>
          <value>1</value>
   </property>
```

```
</configuration>
```

YARN configuration - Single Node

modify the **mapred-site.xml**

```
boss@solaiv[]# cp mapred-site.xml.template mapred-site.xml

boss@solaiv[]# vi mapred-site.xml
```

Paste following between <configuration> tags

```

<configuration>

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

</configuration>
```

Modiy yarn-site.xml

```
boss@solaiv[]# vi 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)

```
root@solaiv[]# cd $HADOOP_PREFIX
root@solaiv[]# bin/hadoop namenode -format
```

Start NameNode daemon and DataNode daemon: (port 50070)

```
root@solaiv[]# sbin/start-dfs.sh
```

To know the running daemons jut type jps or /usr/local/jdk1.8.0_05/bin/jps

Start ResourceManager daemon and NodeManager daemon: (port 8088)

```
root@solaiv[]# sbin/start-yarn.sh
```

To stop the running process

```
root@solaiv[]# sbin/stop-dfs.sh
```

To know the running daemons jut type jps or /usr/local/jdk1.8.0_05/bin/jps

Start ResourceManager daemon and NodeManager daemon: (port 8088)

```
root@solaiv[]# sbin/stop-yarn.sh
```

Make the HDFS directories required to execute MapReduce jobs:

```
  $ bin/hdfs dfs -mkdir /user
  $ bin/hdfs dfs -mkdir /user/mit
```

- Copy the input files into the distributed filesystem:

```
  $ bin/hdfs dfs -put <input-path>/* /input
```

- Run some of the examples provided:

```
  $ bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.5.1.jar grep
/input /output '(CSE)'
```

- Examine the output files:

Copy the output files from the distributed filesystem to the local filesystem and examine them:

```
  $ bin/hdfs dfs -get output output
  $ cat output/*
```

or

View the output files on the distributed filesystem:

```
  $ bin/hdfs dfs -cat /output/*
```