Deploy Hadoop on Cluster

Install Hadoop with YARN in Distributed Mode

This document explains how to setup Hadoop on a multi-node cluster. One Node will act as Master rest all the nodes will act as slaves.

DATAFLAIR WEB SERVICES PVT LTD

http://data-flair.com
+91-8451097879
+91-7718877477

DataFlair Web Services Pvt Ltd
http://www.data-flair.com
info@data-flair.com

Contents

Objective:

This document describes how to install and configure a multi-node Hadoop cluster with YARN. Once the installation is done you can perform Hadoop Distributed File System (HDFS) and Hadoop Map-Reduce operations.

Recommended Platform:

- OS: Linux is supported as a development and production platform. You can use Ubuntu 14.04 or later (you can also use other Linux flavors like: CentOS, Redhat, etc.)
- Hadoop: Cloudera Distribution for Apache hadoop CDH5.x (you can use Apache hadoop 2.x)

Install Hadoop on Master:

1. Prerequisites:

1.1. Add Entries in hosts file:

Edit hosts file ($sudo nano /etc/hosts) and add entries of master and slaves:

MASTER-IP master
SLAVE01-IP slave01
SLAVE02-IP slave02

(NOTE: In place of MASTER-IP, SLAVE01-IP, SLAVE02-IP put the value of corresponding IP)

1.2. Install Java 7 (Recommended Oracle Java):

1.2.1 Install Python Software Properties:

$sudo apt-get install python-software-properties

1.2.2 Add Repository:

$sudo add-apt-repository ppa:webupd8team/java

1.2.3 Update the source list:

$sudo apt-get update

1.2.4 Install Java:

$sudo apt-get install oracle-java7-installer

1.3 Configure SSH:

1.3.1 Install Open SSH Server-Client:

$sudo apt-get install openssh-server openssh-client

1.3.2 Generate Key Pairs:

$ssh-keygen -t rsa -P ""

1.3.3 Configure password-less SSH:

Copy the content of .ssh/id_rsa.pub (of master) to .ssh/authorized_keys (of all the slaves as well as master)

1.3.4 Check by SSH to all the Slaves:

$ssh slave01
$ssh slave02

2. Install Hadoop:

2.1 Download Hadoop:

http://archive.cloudera.com/cdh5/cdh/5/hadoop-2.5.0-cdh5.3.2.tar.gz

2.2 Untar Tar ball:

$tar xzf hadoop-2.5.0-cdh5.3.2.tar.gz

(Note: All the required jars, scripts, configuration files, etc. are available in HADOOP_HOME directory (hadoop-2.5.0-cdh5.3.2))

2.3 Setup Configuration:

2.3.1 Edit .bashrc:

Edit .bashrc file located in user's home directory and add following environment variables:

```
export HADOOP_PREFIX="/home/ubuntu/hadoop-2.5.0-cdh5.3.2"
export PATH=$PATH:$HADOOP_PREFIX/bin
export PATH=$PATH:$HADOOP_PREFIX/sbin
export HADOOP_MAPRED_HOME=${HADOOP_PREFIX}
export HADOOP_COMMON_HOME=${HADOOP_PREFIX}
export HADOOP_HDFS_HOME=${HADOOP_PREFIX}
export YARN_HOME=${HADOOP_PREFIX}
```

(Note: After above step restart the Terminal/Putty, so that all the environment variables will come into effect)

2.3.1.1 Check environment variables

Check whether the environment variables added in .bashrc file are available:

$bash
$hdfs *(It should not give error: command not found)*

2.3.2 Edit hadoop-env.sh:

Edit configuration file hadoop-env.sh (located in HADOOP_HOME/etc/hadoop) and set JAVA_HOME:

export JAVA_HOME=<path-to-the-root-of-your-Java-installation> (eg: /usr/lib/jvm/java-7-oracle/)

2.3.3 Edit core-site.xml:

Edit configuration file core-site.xml (located in HADOOP_HOME/etc/hadoop) and add following entries:

```
<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://master:9000</value>
  </property>
  <property>
    <name>hadoop.tmp.dir</name>
    <value>/home/ubuntu/hdata</value>
  </property>
</configuration>
```

Note: /home/ubuntu/hdata is a sample location; please specify a location where you have Read Write privileges

2.3.4 Edit hdfs-site.xml:

Edit configuration file hdfs-site.xml (located in HADOOP_HOME/etc/hadoop) and add following entries:

```
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>2</value>
  </property>
</configuration>
```

2.3.5 Edit mapred-site.xml:

Edit configuration file mapred-site.xml (located in HADOOP_HOME/etc/hadoop) and add following entries:

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

```
</configuration>
```

2.3.5 Edit yarn-site.xml:

Edit configuration file mapred-site.xml (located in HADOOP_HOME/etc/hadoop) and add following entries:

```
<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.resource-tracker.address</name>
    <value>master:8025</value>
  </property>
  <property>
    <name>yarn.resourcemanager.scheduler.address</name>
    <value>master:8030</value>
  </property>
  <property>
    <name>yarn.resourcemanager.address</name>
    <value>master:8040</value>
  </property>
</configuration>
```

2.3.5 Edit salves:

Edit configuration file slaves (located in HADOOP_HOME/etc/hadoop) and add following entries:

```
slave01
slave02
```

"Hadoop is setup on Master, now setup Hadoop on all the Slaves"

3. Install Hadoop On Slaves:

3.1 Setup Pre-requisites on all the slaves:

Run following steps on all the slaves:

- "1.1. Add Entries in hosts file"
- "1.2. Install Java 7 (Recommended Oracle Java)"

3.2 Copy configured setups from master to all the slaves

3.2.1 Create tar-ball of configured setup:

$ tar czf hadoop.tar.gz hadoop-2.5.0-cdh5.3.2 *(NOTE: Run this command on Master)*

3.2.2 Copy the configured tar-ball on all the slaves

$ scp hadoop.tar.gz slave01:~ (NOTE: Run this command on Master)
$ scp hadoop.tar.gz slave02:~ *(NOTE: Run this command on Master)*

3.3 Un-tar configured hadoop setup on all the slaves

$tar xzf hadoop.tar.gz (NOTE: Run this command on all the slaves)

"Hadoop is setup on all the Slaves. Now Start the Cluster"

4. Start the Cluster:

4.1 Format the name node:

$bin/hdfs namenode -format *(Note: Run this command on Master)*

(NOTE: This activity should be done once when you install hadoop, else it will delete all the data from HDFS)

4.2 Start HDFS Services:

$sbin/start-dfs.sh *(Note: Run this command on Master)*

4.3 Start YARN Services:

$sbin/start-yarn.sh *(Note: Run this command on Master)*

4.4 Check whether services have been started

4.4.1 Check daemons on Master

$jps
NameNode
ResourceManager

4.4.2 Check daemons on Slaves

$jps
DataNode
NodeManager