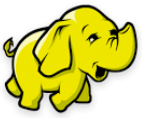
Appendix - I



Hadoop Setup Guide

Prerequisites:

GNU/Linux OS, JVM (JDK), Apache Hadoop

Required Software

1. JavaTM 1.5 + versions, preferably from Sun, must be installed

2. ssh must be installed and sshd must be running to use the Hadoop scripts that manage remote Hadoop daemons. (By default part of Linux system)

3. Hadoop: http://hadoop.apache.org/releases.html (Latest version 2.7.2 (binary) – Size 202 MB)

Steps for Installation [For version: Jdk- 1.7, Hadoop-2.7.2]

1. Install Linux/Ubuntu (If using Windows, use Vmware player and Ubuntu image)

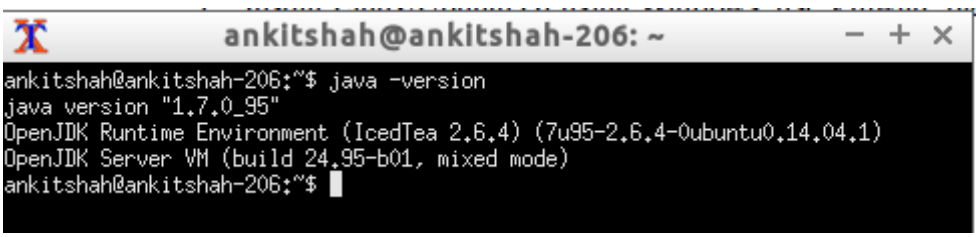
2. First, update the package index & for that command is

```
sudo apt-get update
sudo apt-get install ssh
sudo gedit /etc/ssh/sshd_config
```

3. Install Java

Java is the main prerequisite for Hadoop. First of all, you should verify the existence of java in your system using the command "java -version". The syntax of java version command is given below.

```
$ java -version
```



If JAVA not installed you can install by following commands:

```
sudo apt-get install default-jdk

or

sudo apt-get install openjdk-7-jre
```

4. Download the Hadoop package

Download the binaries to your home directory. Use the default user 'user' for the installation.

```
$wgethttp://www.apache.org/dyn/closer.cgi/hadoop/common/hadoop-2.7.2/hadoop-2.7.2.tar.gz
```

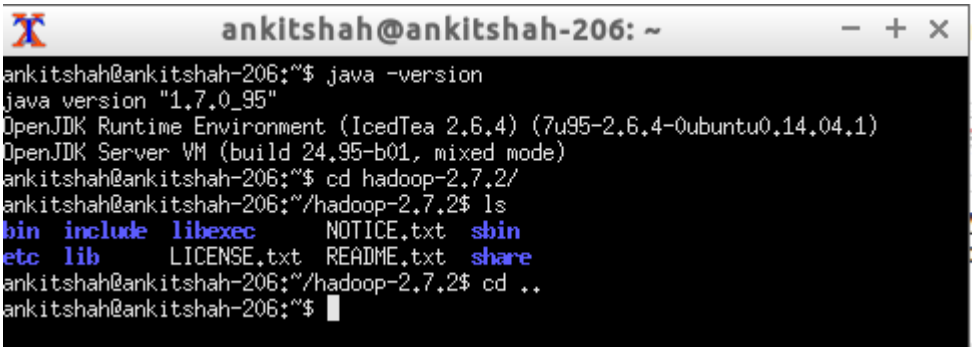
But the file size is 202 MB so it's preferable to download it using some downloader and copy the **hadoop-2.7.2-src.tar.gz** file in your home folder.

5. Extract the **hadoop-2.7.2-src.tar.gz** directory manually or using command:

```
tar -xvf hadoop-2.7.2.tar.gz
```

This will create folder (directory) named: hadoop-2.7.2

6. Cross check hadoop directory



7. Create Hadoop User for common access

It is important to create same username on **all machines** to avoid multiple password entries.

```
$sudo adduser hduser

$sudo adduser hduser sudo
```

8. Move Hadoop to usr/local directory

```
$sudo mv hadoop-2.7.2 /usr/local/
```

9. Move Hadoop to usr/local directory

```
sudo gedit /etc/hosts
```

10. Update the '.bashrc' file to add important Apache Hadoop environment variables for user.

a) Change directory to home.

$ cd

b) Edit the file

$ sudo gedit .bashrc (this command will open one file)

----------Set Hadoop environment Variables - @ the end of File----------

export HADOOP_HOME=/usr/local/hadoop-2.7.2

export HADOOP_CONF_DIR=/usr/local/hadoop-2.7.2/etc/hadoop

export HADOOP_MAPRED_HOME=$HADOOP_HOME

export HADOOP_COMMON_HOME=$HADOOP_HOME

export HADOOP_HDFS_HOME=$HADOOP_HOME

export YARN_HOME=$HADOOP_HOME

export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-i386

export PATH=$PATH:/$HADOOP_HOME/bin

export PATH=$PATH:/$HADOOP_HOME/sbin

export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native

export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib"

P.S.: Hadoop & Jdk according to your installed version

c) Source the .bashrc file to set the hadoop environment variables without having to invoke anew shell:

$. ~/.bashrc

11.Setup the Hadoop Cluster

11.1 Configure JAVA_HOME

Configure JAVA_HOME in 'hadoop-env.sh'. This file specifies environment variables that affect the JDK used by Apache Hadoop 2.0 daemons started by the Hadoop start-up scripts:

$cd $HADOOP_CONF_DIR

```
$sudo gedit hadoop-env.sh
```

Update the JAVA_HOME to:

```
export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-i386
```

11.2 Configure the Default File system

The 'core-site.xml' file contains the configuration settings for Apache Hadoop Core such as I/O settings that are common to HDFS, YARN and MapReduce. Configure default files system

(Parameter: fs.default.name) used by clients in core-site.xml

```
$cd $HADOOP_CONF_DIR

$sudo gedit core-site.xml
```

Add the following line in between the configuration tag:

<property>

<name>fs.defaultFS</name>

<value>hdfs://localhost:9000</value>

</property>

11.3 Configure MapReduce framework

This file contains the configuration settings for MapReduce.

Configure mapred-site.xml and specify framework details.

```
$sudo gedit mapred-site.xml
```

Add the following line in between the configuration tag:

<configuration>

<property>

<name>mapred.job.tracker</name>

<value>localhost:54311</value>

</property>

</configuration>

11.4 Create NameNode and DataNode directory

Create DataNode and NameNode directories to store HDFS data.

```
$sudo mkdir -p /usr/local/hadoop_tmp/hdfs/namenode

$sudo mkdir -p /usr/local/hadoop_tmp/hdfs/datanode

$sudo chown -Rhduser /usr/local/hadoop_tmp
```

11.5 Configure the HDFS

This file contains the configuration settings for HDFS daemons; the Name Node and the data nodes. Configure hdfs-site.xml and specify default block replication, and NameNode and DataNode directories for HDFS. The actual number of replications can be specified when the file is created. The default is used if replication is not specified in create time.

```
$sudo gedit hdfs-site.xml
```

Add the following line in between the configuration tag:

<property>

<name>dfs.replication</name>

<value>1</value>

</property>

<property>

<name>dfs.namenode.name.dir</name>

<value>file:/usr/local/hadoop_tmp/hdfs/namenode

</value>

</property>

<property>

<name>dfs.datanode.data.dir</name>

<value>file:/usr/local/hadoop_tmp/hdfs/datanode

</value>

</property>

11.6 Start the DFS services

The first step in starting up your Hadoop installation is formatting the Hadoop file-system, which is implemented on top of the local file-systems of your cluster. This is required on the first time Hadoop installation. Do not format a running Hadoop file-system, this will cause all your data to be erased.

To format the file-system, run the command:

```
$hdfs namenode –format

(To execute command: Go to Hadoop folder > Go to bin folder)

or

$hadoop namenode –format (deprecated from latest Hadoop version)
```

$start-all.sh

http://localhost:50070/

$stop-all.sh

Multinode- Clone

```
$sudo gedit /etc/hosts
```

Add hostnames with IP address

```
$cd $HADOOP_CONF_DIR
```

```
$sudo gedit hdfs-site.xml
```

<property>

<name>dfs.replication</name>

<value>2</value>

</property>

<property>

<name>dfs.namenode.name.dir</name>

<value>file:/usr/local/hadoop_tmp/hdfs/namenode

</value>

</property>

```
$cd $HADOOP_CONF_DIR
$sudo gedit core-site.xml
```

<property>

<name>fs.defaultFS</name>

<value>hdfs://hadoopmaster:9000</value>

</property>

```
$sudo gedit mapred-site.xml
```

<configuration>

<property>

<name>mapreduce.job.tracker</name>

<value>hadoopmaster:54311</value>

</property>

<property>

<name>mapreduce.framework.name</name>

<value>yarn</value>

</property>

</configuration>

$sudo gedit yarn-site.xml

<property>

<name>yarn.resourcemanager.hostname</name>

<value>hadoopmaster</value>

<description>The hostname of the RM.</description>

</property>

<property>

<name>yarn.nodemanager.aux-services</name>

<value>mapreduce_shuffle</value>

<description>shuffle service that needs to be set for Map Reduce to run </description>

</property>

<property>

<name>yarn.resourcemanager.scheduler.address</name>

<value>hadoopmaster:8030</value>

</property>

<property>

<name>yarn.resourcemanager.address</name>

<value>hadoopmaster:8032</value>

</property>

<property>

<name>yarn.resourcemanager.webapp.address</name>

<value>hadoopmaster:8088</value>

</property>

<property>

<name>yarn.resourcemanager.resource-tracker.address</name>

<value>hadoopmaster:8031</value>

</property>

```
$sudo gedit /usr/local/hadoop-2.7.2/etc/hadoop/slaves
```

Now create clone of Master

Master

```
$sudo gedit /usr/local/hadoop-2.7.2/etc/hadoop/masters
$sudo rm -r /usr/local/hadoop_tmp
$sudo mkdir /usr/local/hadoop_tmp
$sudo mkdir -p /usr/local/hadoop_tmp/hdfs/namenode
$sudo chown -Rhduser /usr/local/hadoop_tmp
If required check for the chmod also
$sudo chmod 755 -R/usr/local/hadoop-2.7.2
$sudo chown -R hduser /usr/local/hadoop-2.7.2
$hdfs namenode -format
```

All Slaves

```
$cd $HADOOP_CONF_DIR
$sudo gedit hdfs-site.xml
```

<property>

<name>dfs.replication</name>

<value>2</value>

</property>

<property>

<name>dfs.datanode.data.dir</name>

<value>file:/usr/local/hadoop_tmp/hdfs/datanode

</value>

</property>

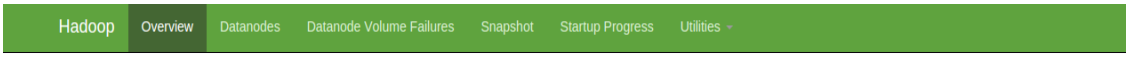
```
$sudo rm -rf  /usr/local/hadoop_tmp
$sudo mkdir -p  /usr/local/hadoop_tmp
$sudo mkdir -p  /usr/local/hadoop_tmp/hdfs/datanode
$sudo chown -Rhduser /usr/local/hadoop_tmp (change ownership)
$sudo chmod 755 -R /usr/local/hadoop-2.7.2 (check user access)
$sudo chown -R hduser /usr/local/hadoop-2.7.2
$sudo reboot
```

Master

```
$sudo /etc/init.d/networking restart
$ssh-keygen -t rsa -P ‘“’
$cat $HOME/.ssh/id_rsa.pub>>$HOME/.ssh/authorized_keys
$ssh-copy-id -i ~/.ssh/id_rsa.pub hduser@slave1 (do it for all slaves)
$ssh hadoopmaster
$ssh slave1 (should be able to login w/o password)
```

```
$start-all.sh
```

http://hadoopmaster:50070/



Overview 'hadoopmaster:9000' (active)

Started:	Wed Mar 09 09:35:43 IST 2016
Version:	2.7.2, rb165c4fe8a74265c792ce23f546c64604acf0e41
Compiled:	2016-01-26T00:08Z by jenkins from (detached from b165c4f)
Cluster ID:	CID-86d0dbe1-8d84-404f-9762-47a22a6b6a04
Block Pool ID:	BP-129920715-192.168.3.206-1457433015398

Summary

Security is off.

Safemode is off.

1 files and directories, 0 blocks = 1 total filesystem object(s).

Heap Memory used 97.01 MB of 171.5 MB Heap Memory. Max Heap Memory is 889 MB.

Non Heap Memory used 21.31 MB of 27.5 MB Commited Non Heap Memory. Max Non Heap Memory is 176 MB.

Summary

Security is off.

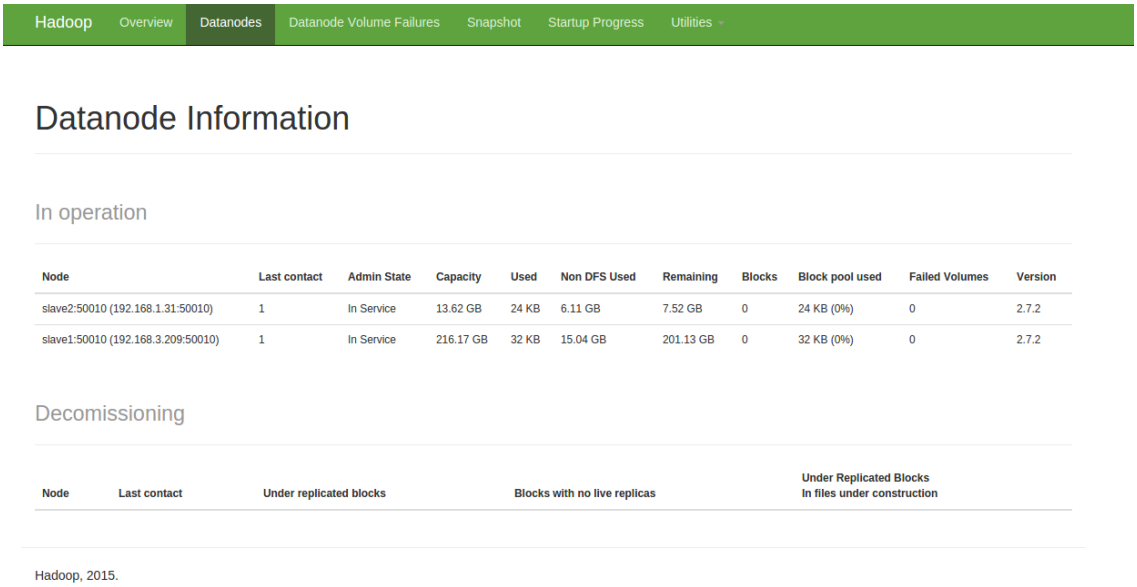
Safemode is off.

1 files and directories, 0 blocks = 1 total filesystem object(s).

Heap Memory used 97.01 MB of 171.5 MB Heap Memory. Max Heap Memory is 889 MB.

Non Heap Memory used 21.31 MB of 27.5 MB Commited Non Heap Memory. Max Non Heap Memory is 176 MB.

Configured Capacity:	229.8 GB
DFS Used:	56 KB (0%)
Non DFS Used:	21.14 GB
DFS Remaining:	208.65 GB (90.8%)
Block Pool Used:	56 KB (0%)
DataNodes usages% (Min/Median/Max/stdDev):	0.00% / 0.00% / 0.00% / 0.00%
Live Nodes	2 (Decommissioned: 0)
Dead Nodes	0 (Decommissioned: 0)
Decommissioning Nodes	0
Total Datanode Volume Failures	0 (0 B)
Number of Under-Replicated Blocks	0
Number of Blocks Pending Deletion	0



To copy Directory/File on Network Client

sudo scp -rq **<source><dest>**

sudo scp –rq **/usr/local/hadoop-2.7.2hduser@192.168.3.13:/usr/local/**

sudo scp –rq **hduser@192.168.3.13:/usr/local/hadoop-2.7.2/usr/local/**

Exclusive commands to start/stop Hadoop Daemons from Master

$hadoop-daemon.sh start namenode

$hadoop-daemon.sh start datanode

$yarn-daemon.sh start resourcemanager

$yarn-daemon.sh start nodemanager

$mr-jobhistory-daemon.sh start historyserver

$start-dfs.sh

$start-yarn.sh

Appendix - II
Grid'5000 Hadoop Setup

SSH (Secure SHell) is a network protocol and application generally used to access a shell account on a remote machine. It is the main tool used to access the Grid'5000 testbed.

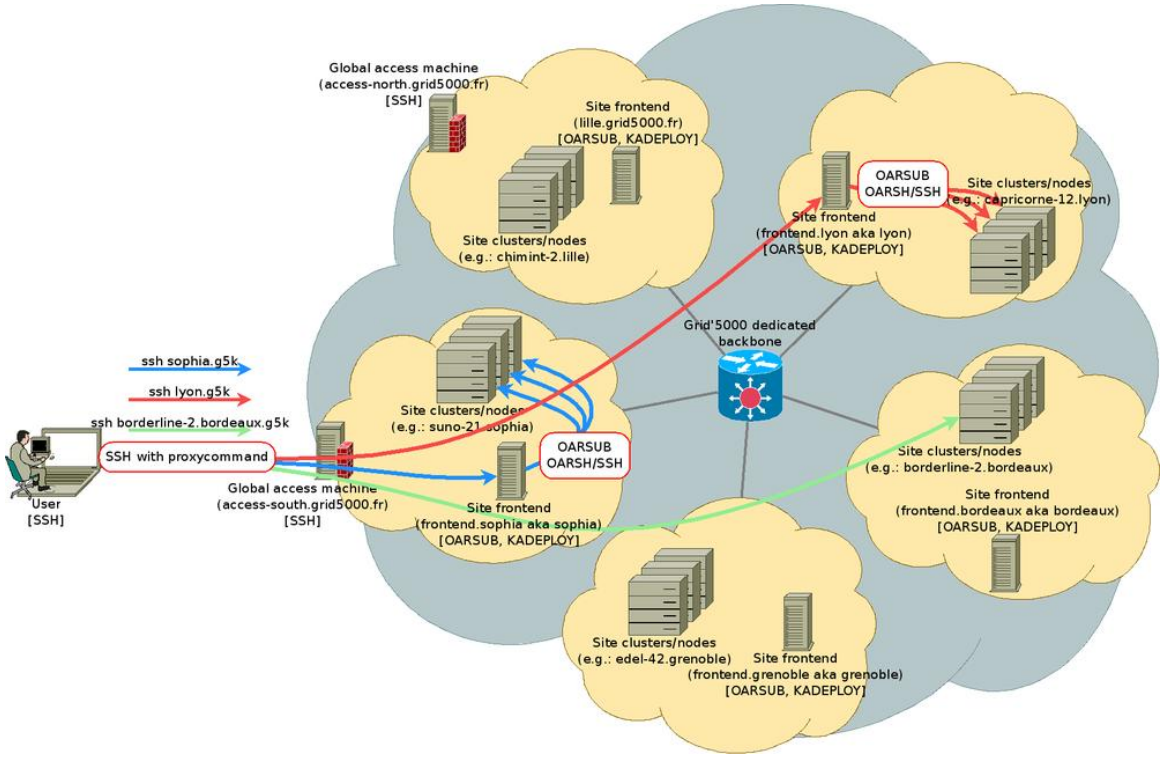
When connecting to a remote machine, the standard way to authenticate is via the remote account's login and password.

To seup SSH access and all please refer link: https://www.grid5000.fr/w/SSH

To setup Hadoop using default parameters refer link: https://www.grid5000.fr/w/Hadoop_On_Execo

After setting SSH access we're ready to go

Step: 1 ssh anshah@access.grid5000.fr



Step: 2 oarsub -I -t allow_classic_ssh -l nodes=10,walltime=1

Step: 3 hadoop_g5k-master/scripts/./hg5k --create $OAR_FILE_NODES --version 2

Step: 4 hadoop_g5k-master/scripts/./hg5k --bootstrap hadoop-2.7.2.tar.gz

Step: 5 hadoop_g5k-master/scripts/./hg5k --initialize --start

Ganglia

https://intranet.grid5000.fr/ganglia/

Grid5000 API

https://api.grid5000.fr/

Default Job Run on Grid'5000

hadoop_g5k-master/scripts/./hg5k --jarjob hadoop-2.7.2
/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.2.jar wordcount

Appendix - III
Hadoop Node Label Setting

1. yarn-site.xml

```
<property>
<name>yarn.node-labels.enabled</name>
<value>true</value>
</property>
<property>
<name>yarn.node-labels.fs-store.root-dir</name>
<value>file:///home/anshah/yarn/node-labels/</value>
</property>
```

2. Add Node Labels

// To add node labels

```
yarn rmadmin -addToClusterNodeLabels “X,Y”
```

//To check node labels have been added

```
yarn cluster –list-node-labels
```

//To remove node labels have been added

```
yarn rmadmin -removeFromClusterNodeLabels "X,Y"
```

3. Assign Node Labels to Cluster Nodes

// Assign Node Label to Node

```
yarn rmadmin -replaceLabelsOnNode "<node1>:<port>=<label1>”
```

// Remove assignment

```
yarn rmadmin -replaceLabelsOnNode "node-1”
```

4. Associating Node Labels with Queues

Refresh Queues

```
yarn rmadmin -refreshQueues
```

5. Confirm Node Label Assignments

//List all running nodes

```
yarn node -list
```

//List all node labels in the cluster:

```
yarn cluster –list-node-labels
```

//List the status of a node

```
yarn node -status <Node_ID>  //use full id generated using yarn node –
list
```

Useful Links

https://developer.ibm.com/hadoop/2017/03/10/yarn-node-labels/
https://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/NodeLabel.html
https://docs.hortonworks.com/HDPDocuments/HDP2/HDP-2.6.4/bk_yarn-resource-management/content/configuring_node_labels.html

Appendix - IV
Saksham Usage

Setup & Deployment

We have integrated "Saksham" policy into the Hadoop-2.7.2 distribution. Saksham policy has no additional dependencies. For a guide how to deploy Hadoop please refer Appendix I, Hadoop documentation guide for installation.

The Saksham JAR files can be found in the Hadoop directory share/hadoop/tools/lib. You can execute them as you normally would, with the Hadoop jar command. The following sections describe the commands in detail.

Saksham Block Rearrangement

The Saksham policy needs a configuration file which defines various parameters given in table. You will find an empty XML configuration file called config.xmlin the etc/hadoop configuration directory. Below table lists all available parametersfor the configuration file. For a complete example of the configuration file, please have a look at sample configuration given.

Configuration Parameter	Description
saksham.hardwaretypes	Comma-separated different types of hardware configurations
saksham.<type>.hosts	Comma-separated list of host names or IP addresses of target DataNodes
saksham.<type>.priority	Assign priority=2 for the hardwaretype where we want to rearrange blocks and priority=1 will not store anything
saksham.hdfsblocks	HDFS location of blocks for rearrangement

Table: config.xml

As the Saksham policy is integrated into the hdfs command, you can also just executethe following command.

```
$ hdfs saksham
```

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
<property>
<name>saksham.hardwaretypes</name>
<value>cluster1,cluster2</value>
</property>
<property>
<name>saksham.cluster1.hosts</name>
<value>
        parapide-1.rennes.grid5000.fr,
        parapide-2.rennes.grid5000.fr,
        parapide-3.rennes.grid5000.fr,
        parapide-4.rennes.grid5000.fr
</value>
</property>
<property>
<name>saksham.cluster1.priority</name>
<value>1</value>
</property>
<property>
<name>saksham.cluster2.hosts</name>
<value>
        parasilo-1.rennes.grid5000.fr,
        parasilo-2.rennes.grid5000.fr,
        parasilo-3.rennes.grid5000.fr,
        parasilo-4.rennes.grid5000.fr,
        parasilo-5.rennes.grid5000.fr,
        parasilo-6.rennes.grid5000.fr
</value>
</property>
<property>
<name>saksham.cluster2.priority</name>
<value>2</value>
</property>
<property>
<name>saksham.hdfsblocks</name>
<value>
        hdfs block location
</value>
</property>
</configuration>
```