Hadoop install and configure

<http://askubuntu.com/questions/144433/how-to-install-hadoop>

Note: I use account li as hadoop user, so just ignore the creation of hadoop user part.

The guide is mostly straight forward and here it is:

1. Install Java

sudo add-apt-repository ppa:webupd8team/java sudo apt-get update && sudo apt-get upgrade sudo apt-get install oracle-java7-installer

1. Create Hadoop user

sudo addgroup hadoop sudo adduser --ingroup hadoop hduser

Where hduser is the Hadoop user you want to have.

1. Configuring SSH

su - hduser ssh-keygen -t rsa -P "" cat .ssh/id\_rsa.pub >> .ssh/authorized\_keys

To be sure that SSH installation went well, you can open a new terminal and try to create ssh session using hduser by the following command:

ssh localhost

reinstall ssh if localhost does not connect

sudo apt-get install openssh-server

1. Edit Sudoers

sudo visudo

1. Add at the end the line to add hduser into sudoers

hduser (ALL)=(ALL:ALL) ALL

To save press CTRL+X, type Y and press ENTER

1. Disable IPv6

sudo gedit /etc/sysctl.conf

Copy the following lines at the end of the file:

#disable ipv6 net.ipv6.conf.all.disable\_ipv6 = 1 net.ipv6.conf.default.disable\_ipv6 = 1 net.ipv6.conf.lo.disable\_ipv6 = 1

If you face a problem telling you, you don't have permissions, just run the previous command with the root account (In case sudo is not enough. For me it was)

1. Now reboot.

You can also do sudo sysctl -p but I rather reboot.

After rebooting, check to make sure IPv6 is off:

cat /proc/sys/net/ipv6/conf/all/disable\_ipv6

it should say **1**. If it says **0**, you missed something.

1. Installing Hadoop

Download a Hadoop release from the following website:

<http://mirror.nexcess.net/apache/hadoop/common/>

Here, we download [hadoop-1.2.1.tar.gz](http://mirror.nexcess.net/apache/hadoop/common/hadoop-1.2.1/hadoop-1.2.1.tar.gz).

Untar the file rename it to hadoop and put it to: /home/li/hadoop.

1. Update $HOME/.bashrc

You will need to update the .bachrc for hduser (and for every user you need to administer Hadoop). To open .bachrc file, you will need to open it as root:

sudo gedit /home/hduser/.bashrc

Then you will add the following configurations at the end of .bachrc file

# Set Hadoop-related environment variables export HADOOP\_PREFIX=/home/li/hadoop

# Set JAVA\_HOME (we will also configure JAVA\_HOME directly for Hadoop later on)`

export JAVA\_HOME=$(dirname $(dirname $(readlink –f /usr/bin/java)))

Now for some helpful alias:

# Some convenient aliases and functions for running Hadoop-related commands unalias fs &> /dev/null alias fs="hadoop fs" unalias hls &> /dev/null alias hls="fs -ls" # Add Hadoop bin/ directory to PATH export PATH=$PATH:$HADOOP\_HOME/bin

1. Configuring Hadoop

The following are configuration files we can use to do the proper configuration. Some of the files you will be using with Hadoop are (More information in [this site](http://wiki.apache.org/hadoop/GettingStartedWithHadoop)):

start-dfs.sh - Starts the Hadoop DFS daemons, the namenode and datanodes. Use this before start-mapred.sh

stop-dfs.sh - Stops the Hadoop DFS daemons.

start-mapred.sh - Starts the Hadoop Map/Reduce daemons, the jobtracker and tasktrackers.

stop-mapred.sh - Stops the Hadoop Map/Reduce daemons.

start-all.sh - Starts all Hadoop daemons, the namenode, datanodes, the jobtracker and tasktrackers. Deprecated; use start-dfs.sh then start-mapred.sh

stop-all.sh - Stops all Hadoop daemons. Deprecated; use stop-mapred.sh then stop-dfs.sh

But before we start using them, we need to modify several files in the /conf folder.

**hadoop-env.sh**

Look for the file hadoop-env.sh, we need to only update the JAVA\_HOME variable in this file:

sudo gedit /home/hduser/hadoop/conf/hadoop-env.sh

or

sudo nano /home/hduser/hduser/hadoop/conf/hadoop-env.sh

Then change the following line:

# export JAVA\_HOME=/usr/lib/j2sdk1.5-sun

To

export JAVA\_HOME=$(dirname $(dirname $(readlink –f /usr/bin/java)))

Note: if you get Error: JAVA\_HOME is not set Error while starting the services, you forgot to uncomment the previous line (just remove #).

**core-site.xml**

Now we need to create a temp directory for Hadoop framework. If you need this environment for testing or a quick prototype (e.g. develop simple hadoop programs for your personal test ...), I suggest to create this folder under /home/hduser/ directory, otherwise, you should create this folder in a shared place under shared folder (like /usr/local ...) but you may face some security issues. But to overcome the exceptions that may caused by security (like java.io.IOException), I have created the tmp folder under hduser space.

To create this folder, type the following command:

mkdir /home/li/hdfstmp

Now, we can open hadoop/conf/core-site.xml to edit the hadoop.tmp.dir entry. We can open the core-site.xml using text editor:

sudo gedit /home/hduser/hadoop/conf/core-site.xml

or

nano /home/hduser/hduser/hadoop/conf/core-site.xml

Then add the following configurations between .. xml elements:

<property>

<name>hadoop.tmp.dir</name>

<value>/home/li/hdfstmp</value>

<description>A base for other temporary directories.</description> </property>

<property>

<name>fs.default.name</name>

<value>hdfs://localhost:54310</value>

<description>The name of the default file system.

A URI whose scheme and authority determine the FileSystem implementation. The uri's scheme determines the config property (fs.SCHEME.impl) naming the FileSystem implementation class. The uri's authority is used to determine the host, port, etc. for a filesystem.</description>

</property>

Now edit mapred-site.xml

<property> <name>mapred.job.tracker</name> <value>localhost:54311</value> <description>The host and port that the MapReduce job tracker runs at. If "local", then jobs are run in-process as a single map and reduce task. </description> </property>

Now edit hdfs-site.xml

<property>

<name>dfs.replication</name>

<value>1</value>

<description>Default block replication.

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. </description>

</property>

1. Formatting NameNode

Now you can start working on the Node. First format:

~/hduser/hadoop/bin/hadoop namenode -format

or

hadoop namenode -format

You should format the NameNode in your HDFS. You should not do this step when the system is running. It is usually done once at first time of your installation.

1. Starting Hadoop Cluster

Navigate to hadoop/bin directory and run the ./start-all.sh script.

cd ~/hduser/hadoop/bin/

start-all.sh

Since we have already add hadoop/bin to $PATH, we can start it directly:

start-mapred.sh

This will start a Namenode, Datanode, Jobtracker and a Tasktracker on your machine.

1. Checking if Hadoop is running

There is a nice tool called jps. You can use it to ensure that all the services are up. In your hadoop bin folder type:

jps

It should show you all Hadoop related processes.

export JAVA\_HOME=$(dirname $(dirname $(readlink -f /usr/bin/java)))

export PATH=$PATH:$JAVA\_HOME/bin

export M2\_HOME=/usr/share/maven2

export HADOOP\_PREFIX=/home/li/hadoop

export HADOOP\_CONF\_DIR=$HADOOP\_PREFIX/conf

export PATH=$PATH:$HADOOP\_PREFIX/bin

