**INSTALLATION DOCUMENTATION (UBUNTU 14.04)**

**Pip Installation:**

$ sudo apt-get install python-pip python-dev --yes

**TensorFlow Installation (CPU only):**

$ sudo pip install --upgrade https://storage.googleapis.com/tensorflow/linux/cpu/tensorflow-0.8.0-cp27-none-linux\_x86\_64.whl

Note: to upgrade installation, it is good practice to use “pip uninstall” first to get a clean installation of the updated TensorFlow.

To test installation:

cc@ml-1:~$ python

Python 2.7.6 (default, Jun 22 2015, 17:58:13)

[GCC 4.8.2] on linux2

Type "help", "copyright", "credits" or "license" for more information.

>>> import tensorflow as tf

>>> hello = tf.constant('Hello sir.')

>>> sess = tf.Session()

>>> print(sess.run(hello))

Hello sir.

>>>

**SciPy Stack Installation:**

$ sudo apt-get install python-numpy python-scipy python-matplotlib ipython ipython-notebook python-pandas python-sympy python-nose --yes

$ sudo pip install jupyter

**Docker Installation:**

Check for updates, make sure that “apt” works with https and that CA certificates are installed:

$ sudo apt-get update

$ sudo apt-get install apt-transport-https ca-certificates

Add the new GPG key:

$ sudo apt-key adv --keyserver hkp://p80.pool.sks-keyservers.net:80 --recv-keys 58118E89F3A912897C070ADBF76221572C52609D

Open /etc/apt/sources.list directory.

Create docker.list file:

$ sudo vi docker.list

Add an entry for Ubuntu 14.04 operating system:

deb https://apt.dockerproject.org/repo ubuntu-trusty main

Save and close the docker.list file. Update and then purge old repo if it exists:

$ sudo apt-get update

$ sudo apt-get purge lxc-docker

Verify that “apt” is pulling from the right repository:

$ apt-cache policy docker-engine

Install apparmor:

$ sudo apt-get install apparmor

Install Docker:

$ sudo apt-get install docker-engine

Start the docker daemon:

$ sudo service docker start

Verify docker is installed correctly:

$ sudo docker run hello-world

**Hadoop Single Node Installation:**

1. Install Java 8 onto system, check java version:

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

$ sudo apt-get update

$ sudo apt-get install oracle-java8-installer

$ sudo apt-get install oracle-java8-set-default

$ java –version

1. Install ssh onto system, check proper installation:

$ sudo apt-get install ssh

$ sudo apt-get install openssh-server

$ which ssh (should output /user/bin/ssh)

$ which sshd (should output /user/sbin/sshd)

1. Add a Hadoop group to system, add user to group (enter new UNIX password), give sudo access:

$ sudo addgroup hadoop

$ sudo adduser --ingroup hadoop **hduser**

**$ sudo usermod -a -G sudo hduser**

**OR Add following to /etc/sudoers/:**

**$ hduser ALL=(ALL:ALL) ALL**

1. Download and unpack Apache Hadoop 2.7, change ownership of download location:

$ wget **http://apache.rediris.es/hadoop/common/hadoop-2.7.0/hadoop-2.7.0.tar.gz**

**$ sudo tar -xzvf hadoop-2.7.0.tar.gz -C /usr/local/lib/**

**$ sudo chown -R hduser:hadoop /usr/local/lib/hadoop-2.7.0**

1. Create HDFS directories:

$ sudo mkdir -p /var/lib/hadoop/hdfs/namenode

$ sudo mkdir -p /var/lib/hadoop/hdfs/datanode

$ sudo chown -R **hduser** /var/lib/hadoop

1. Generate/setup SSH certificates for hduser:

$ sudo su hduser

$ ssh-keygen –t rsa –P “”

1. Add the key that was just created to the list of authorized keys for ease of access:

$ cat $HOME/.ssh/id\_rsa.pub >> $HOME/.ssh/authorized\_keys

1. Check if ssh setup works:

$ ssh localhost

1. Now have to setup configuration files:

First find out where Java has been installed to add that to ~/.bashrc file

$ readlink -f /usr/bin/java

1. Next, use part of this output to add the following to the bottom of ~/.bashrc file:

# JAVA / HADOOP INFORMATION

export JAVA\_HOME=/usr/lib/jvm/java-8-oracle

export HADOOP\_INSTALL=/usr/local/lib/hadoop-2.7.0

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

export PATH=$PATH:$HADOOP\_INSTALL/sbin

export HADOOP\_MAPRED\_HOME=$HADOOP\_INSTALL

export HADOOP\_COMMON\_HOME=$HADOOP\_INSTALL

export HADOOP\_HDFS\_HOME=$HADOOP\_INSTALL

export YARN\_HOME=$HADOOP\_INSTALL

export HADOOP\_COMMON\_LIB\_NATIVE\_DIR=$HADOOP\_INSTALL/lib/native

export HADOOP\_OPTS="-Djava.library.path=$HADOOP\_INSTALL/lib/native"

1. Then source the file to save changes:

$ source ~/.bashrc

1. Add the found java installation location to Hadoop-env.sh file (ensures that java starts up with Hadoop)

$ sudo vi /usr/local/lib/hadoop-2.7.0/etc/hadoop/hadoop-env.sh (change JAVA\_HOME to same directory as was done for ~/.bashrc file)

**NOTE FOR FOLLOWING STEPS: SHOULD BE HADOOP USER!!**

1. Navigate to /usr/local/lib/hadoop-2.7.0/etc/hadoop, then open core-site.xml

$ vi core-site.xml

Edit this file to have the following:

<configuration>

<property>

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

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

</property>

</configuration>

1. Next edit yarn-site.xml similarly to have the following:

$ vi yarn-site.xml

<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>

</configuration>

1. Create mapred-site.xml from template provided:

$ cp /usr/local/lib/hadoop-2.7.0/etc/hadoop/mapred-site.xml.template /usr/local/lib/hadoop-2.7.0/etc/hadoop/mapred-site.xml

Modify mapred-site.xml to have the following:

<configuration>

<property>

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

<value>yarn</value>

</property>

</configuration>

1. Modify hdfs-site.xml to have the following:

<configuration>

<property>

<name>dfs.replication</name>

<value>1</value>

</property>

<property>

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

<value>file:/var/lib/hadoop/hdfs/namenode</value>

</property>

<property>

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

<value>file:/var/lib/hadoop/hdfs/datanode</value>

</property>

</configuration>

1. Format HDFS file system:

$ hdfs namenode -format

1. Once this is done, start Hadoop:

$ start-dfs.sh

$ start-yarn.sh

1. Check if everything is running:

$ jps

Should get something similar to:

76048 SecondaryNameNode

76946 Jps

75542 NameNode

76699 NodeManager

75755 DataNode

76332 ResourceManager

**Spark Single Node Installation:**

1. Download and unpack pre-built Apache Spark 1.6.1:

$ wget http://d3kbcqa49mib13.cloudfront.net/spark-1.6.1-bin-hadoop2.6.tgz

$ sudo tar -xvzf spark-1.6.1-bin-hadoop2.6.tgz

1. Make sure that Java is installed (see Hadoop Single Node Installation Step 1 above).
2. To run Spark in pyspark, enter into Spark folder and type:

$ ./bin/pyspark

**Hadoop Multiple Node Installation:**

In this section, I will add extra information that is needed to install multiple node hadoop cluster compared to single node installation. If steps are the same, I will reference which steps.

[**http://pingax.com/install-apache-hadoop-ubuntu-cluster-setup/**](http://pingax.com/install-apache-hadoop-ubuntu-cluster-setup/)

**^^ this is the method used. I will do it again and write all of the necessary steps**