# Difficulties and Issues while installing Apache™ Hadoop® on Windows and Linux Distributions

Hadoop’s key design goal is to provide storage and computation on lots of homogenous “commodity” machines; usually a machine running Linux. With that goal in mind, the Hadoop team has logically focused on Linux platforms in their development and documentation. Their Quickstart includes the caveat that “Win32 is supported as a development platform. Distributed operation has not been well tested on Win32, so this is not a production platform.” If you want to use Windows to run Hadoop in pseudo-distributed or distributed mode, you’re pretty much left on your own.

## Windows

We invested some time installing Hadoop on Windows 8. So many difficulties have shown as the Hadoop installation guide contains information that only work on Linux/Unix environment. Some tools had to be installed in Windows platform in order to execute commands that install Hadoop. Hadoop assumes that the platform is Linux. So one of the tools we used was **Cygwin. Cygwin** provides a set of tools that allow the existence of a Linux-style command line as well as a Linux-centric software like OpenSSH that would help provide a passwordless secure shell to communicate between computers in the Hadoop cluster.

The problem with Hadoop being installed in a Windows is that the Hadoop distribution code needs to be compiled on a windows machine and all the problems arise from the compilation process. The errors that show up are java-related exceptions and problems.

Another problem has risen when we obtained an unofficial compiled version of Hadoop for Windows. The Hadoop tools (Mapred, Yarn) did not work as there were problems with the compatibility.

At last, we stopped preparing Hadoop for Windows and moved on to installing it in Ubuntu LTS14.04 which is a Linux based operating system.

## Ubuntu Linux LTS 14.04

Moving to Ubuntu Linux to install Hadoop was very easy despite the fact that it required so many commands to be executed. The only difficulty the was faced in Ubuntu was the familiarity with the **Terminal.** Installing Hadoop on Ubuntu was done using the following steps:

### Hadoop Pre-requisites

a) sudo apt-get update

b) sudo apt-get install default-jdk

c) java -version

d) sudo addgroup hadoop

e) sudo adduser --ingroup hadoop hduser

f) sudo adduser hduser sudo

g) sudo apt-get install openssh-server

h) su hduser

i) ssh-keygen -t rsa -P ""

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

### Installing Hadoop

1) wget http://mirrors.sonic.net/apache/hadoop/common/hadoop-2.7.1/hadoop-2.7.1.tar.gz

2) tar xvzf hadoop-2.7.1.tar.gz

3) sudo mv hadoop-2.7.1 /usr/local/hadoop

4) sudo chown -R hduser:hadoop /usr/local/hadoop

5) sudo mkdir -p /usr/local/hadoop\_project

6) sudo mkdir -p /usr/local/hadoop\_project/hfs/namenode

7) sudo mkdir -p /usr/local/hadoop\_project/hdfs/datanode

8) sudo chown -R hduser /usr/local/hadoop\_project

### Configuring hadoop files

#### 1) BASHRC

sudo nano ~/.bashrc

#append the code below to the end of the file and save it

export JAVA\_HOME=/usr/lib/jvm/java-7-openjdk-amd64

export HADOOP\_HOME=/usr/local/hadoop

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

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

export HADOOP\_MAPRED\_HOME=$HADOOP\_HOME

export HADOOP\_COMMON\_HOME=$HADOOP\_HOME

export HADOOP\_HDFS\_HOME=$HADOOP\_HOME

export YARN\_HOME=$HADOOP\_HOME

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

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

#type the command below afterward

source ~/.bashrc

#then go to the hadoop-env.sh file

sudo nano /usr/local/hadoop/etc/hadoop/hadoop-env.sh

#look for the export JAVA\_HOME and replace the part after the = with the path to java.

export JAVA\_HOME=/usr/lib/jvm/java-7-openjdk-amd64

#### 2) Configuration file :

$ sudo nano core-site.xml

<configuration>

<property>

<name>fs.defaultFS</name>

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

</property>

</configuration>

$ sudo nano 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>

$ sudo cp mapred.site.xml.template mapred-site.xml

$ sudo nano mapred-site.xml

<configuration>

<property>

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

<value>yarn</value>

</property>

</configuration>

$ sudo nano hdfs-site.xml

<configuration>

<property>

<name>dfs.replication</name>

<value>1</value>

</property>

<property>

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

<value>file:/usr/local/hadoop/hadoop\_data/hdfs/namenode</value>

</property>

<property>

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

<value>file:/usr/local/hadoop/hadoop\_store/hdfs/datanode</value>

</property>

</configuration>

$ cd

$ mkdir -p /usr/local/hadoop/hadoop\_data/hdfs/namenode

$ mkdir -p /usr/local/hadoop/hadoop\_data/hdfs/datanode

$ sudo chown chaal:chaal -R /usr/local/hadoop

$ hdfs namenode -format

$ start-all.sh

$ jps