HADOOP CLUSTER SETUP GUIDE:

Passwordless SSH Sessions:

Before we start our installation, we have to ensure that passwordless SSH Login is possible to any of the Linux machines of CS120. In order to do so, log in to your account and run the following commands (separately):
[For Further understanding, look up: https://hadoopabcd.wordpress.com/2015/05/16/linux-password-less-ssh-logins/]

```
ssh-keygen -t rsa -P "" -f ~/.ssh/id_rsa
cat $HOME/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys
```

Setting up Config Files:

Step 1. Create a new folder for your configuration files:

```
mkdir ~/hadoopConf
```

Step 2. Use below commands (separately) to download and unpack the files into ~/hadoopConf

```
wget http://www.cs.colostate.edu/~cs555/conf.tar
tar xvf conf.tar -C ~/hadoopConf
```

For the files core-site.xml, mapred-site.xml and hdfs-site.xml, **anything with "HOST" or "PORT" in the config files should be replaced with actual hostnames and port numbers assigned to you (refer below table).**

Student (CID)	Port range
amartin	30500 - 30549
ashhcree	30550 - 30599
cacaleb	30600 - 30649
cwestbro	30650 - 30699
ericlipe	30700 - 30749
jborg	30750 - 30799
kbass40	30800 - 30849
mfernst	30850 - 30899
menukaw	30900 - 30949
sbydergo	30950 - 30999
wenqin	31000 - 31049

Make sure that you select 5-10 nodes (CS120 machines only) for your workers file that is detailed below in Step 4.
You need to select a set of available ports from the non-privileged port range. Do not select any ports between 56,000 and 57,000. These are dedicated for the shared HDFS cluster. To reduce possible port conflicts, you will be each assigned a port range

Step 3. Set the environment variables in your .bashrc file

```
export HADOOP_HOME="/usr/local/hadoop/latest"
export HADOOP_COMMON_HOME="${HADOOP_HOME}"
export HADOOP_HDFS_HOME="${HADOOP_HOME}"
export HADOOP_MAPRED_HOME="${HADOOP_HOME}"
export YARN_HOME="${HADOOP_HOME}"
export HADOOP_CONF_DIR="<path to your config directory mentioned in Step 2>"
export YARN_CONF_DIR="${HADOOP_CONF_DIR}"
export HADOOP_LOG_DIR="/tmp/${USER}/hadoop-logs"
export YARN_LOG_DIR="/tmp/${USER}/yarn-logs"
export JAVA_HOME="/usr/lib/jvm/java-1.8.0-openjdk"
export HADOOP_OPTS="-Dhadoop.tmp.dir=/tmp/${USER}"
export HADOOP_CLASSPATH="${JAVA_HOME}/lib/tools.jar"
export YARN_NODEMANAGER_OPTS="-Dhadoop.tmp.dir=/tmp/${USER}"
```

For HADOOP_CONF_DIR environment variable, set the path of your hadoopConf directory (created in Step 1).

Reflect changes made in .bashrc file by executing “source ~/.bashrc” in command line interface.

Step 4. Create masters and workers file
Masters file should contain the name of the machine you choose as your secondary namenode. Workers contains the list of all your worker nodes. This assignment requires at least 5 such nodes.

Running HDFS

a. **First Time Configuration**

Step 1. Log into your namenode
You will have specified your namenode in core-site.xml.

```
<property>
```

```
 <name>fs.default.name</name>
 <value>hdfs://host:port</value>
</property>
```

Step 2. Format your namenode.

```
$HADOOP_HOME/bin/hdfs namenode -format
```

b. **Starting HDFS**

Run the start script:

```
$HADOOP_HOME/sbin/start-dfs.sh
```

Step 3. Check the web portal to make sure that HDFS is running, open on your browser the url **http://<namenode>:<port given>** for the property **dfs.namenode.http-address** in hdfs-site.xml

c. **Stopping HDFS**

Step 1. Log into your namenode

Step 2. Run the stop script:

```
$HADOOP_HOME/sbin/stop-dfs.sh
```

Running Yarn

a**. Starting Yarn** (Make sure HDFS running)

Step 1. Log into your resource manager
(Resource Manager is the node you set as HOST
in the yarn-site.xml)

Step 2. Run the start script

```
$HADOOP_HOME/sbin/start-yarn.sh
```

Step 3. Check the web portal to make sure yarn is running
http://<resourcemanager>:<port given for the property yarn.resourcemanager.webapp.address>

b. **Stopping Yarn**

Step 1. Log into your resource manager

Step 2. Run the stop script

```
$HADOOP_HOME/sbin/stop-yarn.sh
```

Running a Job in Yarn

a. **Setup**

Step 1. Open mapred-site.xml

Step 2. Change the value of property `mapreduce.framework.name` to `yarn`

To be safe, restart HDFS and yarn

Hadoop Standalone Operations

The following are a few standalone operations that would help you manage files in HDFS and perform certain necessary Hadoop operations (NOTE: First setup your cluster):

- The command to see the contents of your HDFS is similar to the ls command you use in UNIX

  $HADOOP_HOME/bin/hadoop fs -ls /

  For instance, if you have a directory called tmp in your HDFS, in order to look into tmp, just type

  $HADOOP_HOME/bin/hadoop fs -ls /tmp

- In order to create a folder (let's call it dataLoc) in HDFS where you want to put data, use the following command:

  $HADOOP_HOME/bin/hadoop fs -mkdir /dataLoc

  Now to upload the data into dataLoc, run the following command:

  $HADOOP_HOME/bin/hadoop fs -put <SOURCE> /dataLoc