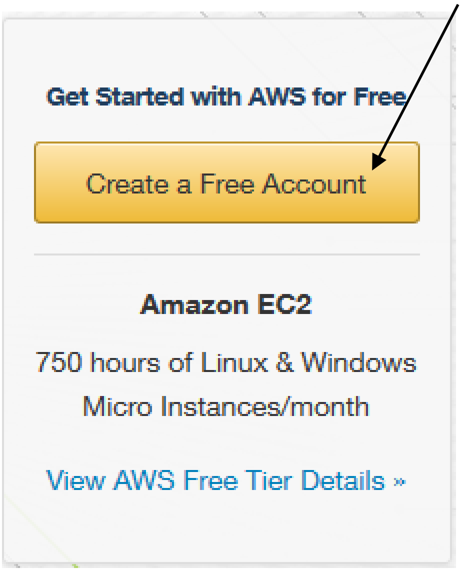
Steps to set up AWS and install Hadoop environment.

A. Accessing Free Hadoop workshop at www.colaberry.com
   a. Launch Google Chrome browser
   b. Create a log in at www.colaberry.com using http://bit.ly/1NrRfHR
   c. Activate your account using the activation email sent
   d. Login to www.colaberry.com and fill the information
   e. Click on ‘Enroll’ button beside ‘FREE Apache Hadoop Workshop’
   f. Log out and log in again to access the class material
   g. Click on the lock button to unlock the content

B. Signing up for Amazon Web Services(AWS) free tier:
   1. Visit https://aws.amazon.com
   2. Click on create a free account button on the right


Get Started with AWS for Free

Amazon EC2
750 hours of Linux & Windows Micro Instances/month

   3. In the login window, enter your email address, Choose I am a new user and click sign in.
   4. If you already have an Amazon user id (like if you have shopped on Amazon or using Amazon, kindle), it will complain that email id already exists. In that case, login with your existing amazon id and it will ask you to sign up for aws. Otherwise go ahead with next step.
   5. Enter login credentials (name, email id confirmation, password) and click on create account.
   6. You will get message that you have not yet signed up for AWS. Click on Sign Up for AWS to continue
   7. Enter your name and address information, along with captcha and Enter.

8. Next it will ask for payment information. The free tier is mentioned here. But still they need the **credit card for any extra charges**. They normally charge and cancel one $ to verify you credit card before approving your account.
9. It will go through the confirmation process as below, including calling your number and asking for a pin (automated). Complete this process.

10. After you complete the process it will take some time (normally few hours) to activate your account. Once activated, you can login and use the machines.

C. Creating an AWS free tier instance:

1. Login at https://aws.amazon.com
2. Click on services, EC2 to go to EC2 management console
3. Click on Instances (on the left list) and choose Launch Instance
4. Step 1: Choose AMI – Amazon Linux 64-bit, make sure it is indicated as 'Free tier eligible'
5. Step2: Choose t2.micro instance that is indicated as 'free tier eligible'
6. Step3: Configure instance details – click on Next: Add storage
7. Step4: Choose General Purpose (SSD) and modify Size to 30 GB. This is the limit for free tier.
8. Choose review and launch to launch the instance
9. IMPORTANT: In the next step, make sure you create a new key pair and down load key pair and save on your computer. Remember the name and location where you stored. This is required to access your aws system using telnet.
10. Use the instructions under: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/putty.html to connect to your instance using putty. Note that puttygen is now under winscp.
11. Connect to your instance using putty

D. Installing hadoop on aws

1. #Connect to your instance with ec2-user using putty sudo you update to update all the packages using the following command:

```
sudo yum update
```

2. #Check that java (1.6 or higher) is installed with

   ```
   java –version
   ```

3. #Check if jps command is present, if not install jps with following commands:
   #Note: Make sure that the java version is same as the version provided by earlier command.

   ```
   sudo yum provides  /usr/lib/jvm/java-1.7.0-openjdk-1.7.0.75.x86_64/bin/jps
   sudo yum install  /usr/lib/jvm/java-1.7.0-openjdk-1.7.0.75.x86_64/bin/jps
   ```

4. #Download the one click install of cloudera hadoop distribution

   ```
   wget http://archive.cloudera.com/cdh4/one-click-install/redhat/6/x86_64/cloudera-cdh-4-0.x86_64.rpm
   ```

5. #Install cloudera hadoop package using Yum

   ```
   sudo yum --nogpgcheck localinstall cloudera-cdh-4-0.x86_64.rpm
   ```

   #select y and press enter when asked for confirmation

6. #install job tracker.
   #This install will take longer as all common hadoop libraries will be downloaded

   ```
   sudo yum install hadoop-0.20-mapreduce-jobtracker
   ```

7. #install namenode

   ```
   sudo yum install hadoop-hdfs-namenode
   ```

8. #Install tasktracker and datanode
   #In fully distributed mode, we will do this on all worker nodes, not on master nodes

   ```
   sudo yum install hadoop-0.20-mapreduce-tasktracker hadoop-hdfs-datanode
   ```

9. #install client.
   #in fully distributed mode, you will do this only where you will run client commands

   ```
   sudo yum install hadoop-client
   ```

10. #check installation using the below command:

    ```
    find / -name hadoop 2>/dev/null
    ```

#Result should be the following folders:

```
/etc/default/hadoop

/etc/hadoop

/usr/bin/hadoop

/usr/lib/hadoop

/usr/lib/hadoop/bin/hadoop

/usr/lib/hadoop/etc/hadoop

/usr/lib/hadoop-0.20-mapreduce/bin/hadoop

/usr/lib/hadoop-0.20-mapreduce/include/hadoop

/usr/lib/hadoop-yarn/etc/hadoop
```

11. #check daemons

    ```
    ls -l /etc/init.d/hadoop*
    ```

12. #check configuration

    ```
    cd /etc/hadoop/conf
    ```

    You will need vi to modify the files. I have given some vi commands here for quick reference. Practice these on a temporary file like vi /tmp/myfile

Description	Command
To navigate left, right, up, down	Use arrow keys
To delete the current line	dd (Use 2dd to delete 2 lines, etc)
To go to end of file	:$ and enter
To go to end of line	$
To insert a line after current line	O
To insert at the current point	I
To get out of the insert mode	Escape
To undo the previous change	U
To save the file and exit	Escape :wq enter
To exit without saving the file	Escape :q! enter

13. #modify the core-site.xml (you will have to use sudo vi for editing a file) and add lines below:

    sudo vi core-site.xml

```
    <configuration>

<property>

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

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

</property>

</configuration>
```

14. # edit hdfs-site.xml to add modifications:

    sudo vi hdfs-site.xml

    Go to the line containing <configuration> and delete the all the lines until closing tag </configuration> using ‘dd’ to delete each line.
    Type o and paste the following lines, then press escape :wq enter to save the file.

```
<configuration>

<property>
```

```
  <name>dfs.name.dir</name>
  <value>/var/lib/hadoop-hdfs/name</value>
</property>
<property>
  <name>dfs.data.dir</name>
  <value>/var/lib/hadoop-hdfs/data</value>
</property>
<property>
  <name>dfs.replication</name>
  <value>1</value>
</property>
</configuration>
```

15. #create the name and data directories

    ```
    sudo -u hdfs mkdir -p /var/lib/hadoop-hdfs/name /var/lib/hadoop-hdfs/data
    sudo -u hdfs mkdir -p /var/lib/hadoop-mapreduce
    sudo chmod 1777 /var/lib/hadoop-mapreduce
    ```

16. #set java heapsize to 128MB in /etc/default/hadoop. Default is 1000MB
    # You need to do sudo vi to edit the file

    ```
    sudo vi /etc/default/hadoop
    ```

    Go to the end of the file with :$ enter and then type o to add a new line. Paste the below statement and type escape :wq enter to save the file.

    ```
    export HADOOP_HEAPSIZE=128
    ```

17. #format the namenode, make sure there are no errors

    ```
    sudo -u hdfs hadoop namenode -format
    ```

#start name node daemon

```
sudo service hadoop-hdfs-namenode start
sudo service hadoop-hdfs-datanode start
```

18. #/tmp directory to be created in hdfs before jobtracker starts

```
sudo -u hdfs hdfs dfs -mkdir /tmp

sudo -u hdfs hdfs dfs -chmod 1777 /tmp

hdfs dfs -ls /

sudo -u hdfs hdfs dfs -mkdir /user

sudo -u hdfs hdfs dfs -mkdir /user/ec2-user

sudo -u hdfs hdfs dfs -chown ec2-user /user/ec2-user
```

19. #add jobtracker rpc and java opts

#Add to mapred-site.xml:

```
cd /etc/hadoop/conf

sudo vi mapred-site.xml
```

Go to the line containing <configuration> and delete the lines till </configuration>.
Type o and paste the following lines, then press escape :wq enter to save the file.

```
<configuration>

<property>

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

   <value>localhost:8021</value>

</property>

<property>
```

```
    <name>mapred.local.dir</name>
    <value >/var/lib/hadoop-mapreduce </value>
</property>
<property>
    <name>mapred.java.child.opts</name>
    <value>-Xmx128m</value>
</property>
</configuration>
```

20.#Create some temp directories for mapreduce job in hdfs

```
sudo -u hdfs hdfs dfs -mkdir /tmp/hadoop-mapred
sudo -u hdfs hdfs dfs -chmod 1777 /tmp/hadoop-mapred
sudo -u hdfs hdfs dfs -mkdir  /tmp/hadoop-mapred/mapred
sudo -u hdfs hdfs dfs -chmod 1777 /tmp/hadoop-mapred/mapred
sudo -u hdfs hdfs dfs -mkdir /tmp/hadoop-mapred/system
sudo -u hdfs hdfs dfs -chmod 1777 /tmp/hadoop-mapred/system
sudo -u hdfs hdfs dfs -mkdir /tmp/hadoop-mapred/staging
sudo -u hdfs hdfs dfs -chmod 1777 /tmp/hadoop-mapred/staging
```

21.#start jobtracker and tasktracker daemons

```
sudo service hadoop-0.20-mapreduce-jobtracker start
sudo service hadoop-0.20-mapreduce-tasktracker start
```

22.#run an example job

```
cd
```

```
cat /var/log/hadoop-hdfs/hadoop-hdfs-namenode-*.log  > logfile

hdfs dfs -put logfile

hadoop jar /usr/lib/hadoop-0.20-mapreduce/hadoop-examples.jar

hadoop jar /usr/lib/hadoop-0.20-mapreduce/hadoop-examples.jar wordcount logfile outdir

hdfs dfs -cat outdir/part*
```

23. #run a streaming job

```
cd

cp /var/log/hadoop-hdfs/hadoop-hdfs-namenode-*.log > logfile

hdfs dfs -put logfile <destination>

hadoop jar /usr/lib/hadoop-mapreduce/hadoop-streaming.jar  -input logfile  -output countout -mapper 'sed -n "/INFO/p"' -reducer 'wc -l' -numReduceTasks 1

hdfs dfs -cat countout/part*
```