Created by: Abid Aziz
Department of Electrical and Computer Engineering, RUET
Email: AzizAbid1@proton.me
Phone: +8801642818018

Red Lines are Commands to write.

STEP 1:

Disable SELinux (Remain at root user)

Before starting, it is a good idea to disable the SELinux in your system.

To disable SELinux, open the /etc/selinux/config file with the following command:

```
nano /etc/selinux/config
```

Change the following line:

```
SELINUX=disabled
```

Save the file when you are finished. Next, restart your system to apply the SELinux changes.

STEP 2:

Disable IPv6 (Remain at root user)

```
nano /etc/sysctl.conf
```
(When you're using nano editor, just copy-paste these lines and Press Ctrl+X, then Y, hit enter)

Append IPv6 disable settings: At the bottom of the sysctl.conf file, add the following lines:

```
# Disable IPv6

net.ipv6.conf.all.disable_ipv6 = 1

net.ipv6.conf.default.disable_ipv6 = 1

net.ipv6.conf.lo.disable_ipv6 = 1
```

Save the file and exit the editor

Apply the sysctl settings:

After editing the sysctl.conf file, apply the changes without rebooting:

```
sudo sysctl -p
```

This command will load the settings from the sysctl.conf file, including your new changes to disable IPv6.

sudo reboot(Optional)

STEP 3:

Create a Hadoop User (Remain at root user)

It is a good idea to create a separate user to run Hadoop for security reasons.

Run the following command to create a new user with name hadoopuser:

useradd hadoopuser

Next, set the password for this user with the following command:

passwd hadoop

(set there a password and remember it for future use)

STEP 4:

(Remain at root user)

Add the user to the wheel group. This is the recommended way to grant sudo access because it avoids directly editing the sudoers file:

usermod -aG wheel hadoopuser (suppose you create hadoopuser, or anything else)

This command adds the hadoop user to the wheel group. Members of this group are typically granted sudo privileges by default in CentOS and other RHEL-based systems.

STEP 5:

Install Java. (Remain at root user)

Paste the following command

wget --no-cookies --no-check-certificate --header "Cookie: oraclelicense=accept-securebackup-cookie" https://javadl.oracle.com/webapps/download/GetFile/1.8.0_271-b09/61ae65e088624f5aaa0b1d2d801acb16/linux-i586/jdk-8u271-linux-x64.tar.gz

Then type:

tar -xvzf jdk1.8.0_271

mv jdk1.8.0_271 /usr/local

sudo vi /etc/profile.d/java.sh

Then Add the following lines to set JAVA_HOME and PATH:

export JAVA_HOME=/usr/local/jdk1.8.0_271

export PATH=$PATH:$JAVA_HOME/bin

Save and close the file.

Then type: source /etc/profile (to see the application of Java -version.)

Type: java -version (it will show you the java version you installed)

STEP 6:

Install and Configure SSH

Since SSH is typically installed by default, you might only need to ensure it's enabled and started:

sudo dnf install openssh-server # Only if not already installed. No need to write this command if error comes

sudo systemctl enable sshd

sudo systemctl start sshd

Note the service name is sshd (SSH daemon) rather than ssh.

Switch to the Hadoop User and Configure SSH Keys. Switching to the Hadoop user and generating SSH keys is similar:

su - hadoopuser

Then generate SSH keys (make sure to hit ENTER when asked for a passphrase to leave it empty for Hadoop setup):

ssh-keygen -t rsa -P ""

Creates the key pair with no passphrase

Now, append the public key to the authorized keys and set appropriate permissions:

cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys

chmod 600 ~/.ssh/authorized_keys

Then, copy the SSH key to ensure password-less SSH login:

ssh-copy-id localhost

Finally, test SSH login to localhost:

ssh localhost

This should log in without asking for a password, indicating your SSH setup is correct.

STEP 7:

Install Hadoop (remain at root user)

wget http://apachemirror.wuchna.com/hadoop/common/hadoop-3.3.6/hadoop-3.3.6.tar.gz

(Hit ENTR- take time to install)

tar xzf hadoop-3.3.6.tar.gz

rm-rf hadoop-3.3.6.tar.gz

mv hadoop-3.3.6 /usr/local

ln-sf /usr/local/hadoop-3.3.6/ /usr/local/hadoop

chown-R hadoopuser:hadoopgroup /usr/local/hadoop-3.3.6/

Now switch to Hadoop user

su– hadoopuser

sudo nano ~/.bashrc

Append the below

Java configuration

export JAVA_HOME=/usr/local/jdk1.8.0_271

export PATH=$JAVA_HOME/bin:$PATH

Hadoop configuration

export HADOOP_HOME=/usr/local/hadoop

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_CONF_DIR=$HADOOP_HOME/etc/hadoop

export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native

export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib/native"

Adding Hadoop binaries to PATH

export PATH=$PATH:$HADOOP_HOME/bin:$HADOOP_HOME/sbin

Then:

source ~/.bashrc (to apply changes)

Apply Changes:

After making these changes, save the file and apply the changes by sourcing your .bashrc:

source ~/.bashrc

Verify the Environment Variables. Check if the variables are set correctly:

echo $JAVA_HOME

echo $HADOOP_HOME

echo $PATH

This should display the paths you set and show that the Hadoop and Java binaries are correctly included in your system's PATH.

Now Double check Java Version, path and also Hadoop path

STEP 8:

Switch to **hadoopuser**. Then type:

cd $HADOOP_HOME/etc/hadoop

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

Append the below

export JAVA_HOME="/usr"

Create Directory where the namenode will exist. For this, type these commands:

sudo mkdir -p /usr/local/hadoop-3.3.6/hadoopdata/hdfs/namenode

sudo chown -R hadoopuser:hadoopgroup /usr/local/hadoop-3.3.6/hadoopdata

sudo chmod -R 700 /usr/local/hadoop-3.3.6/hadoopdata

STEP 9:

Configure Hadoop (Remain at hadoopuser)

nano $HADOOP_HOME/etc/hadoop/core-site.xml

Inside the <configuration> </configuration>, Append these lines and then save and exit

```
<property>
 <name>fs.defaultFS</name>
 <value>hdfs://localhost:9000</value>
</property>
```

Then:

nano $HADOOP_HOME/etc/hadoop/hdfs-site.xml

Inside the <configuration> </configuration>, Append these lines and then save and exit

```
<property>
 <name>dfs.replication</name>
 <value>1</value>
</property>
<property>
 <name>dfs.namenode.name.dir</name>
 <value>file:/usr/local/hadoop/data/namenode</value>
</property>
<property>
 <name>dfs.datanode.data.dir</name>
 <value>file:/usr/local/hadoop/data/datanode</value>
</property>
```

Then:

nano mapred-site.xml.template

Inside the <configuration> </configuration>, Append these lines and then save and exit

```
<property>
  <name>mapreduce.framework.name</name>
  <value>yarn</value>
</property>
```

Type: mv mapred-site.xml.template mapred-site.xml

Then:

nano yarn-site.xml

Inside the <configuration> </configuration>, Append these lines and then save and exit

```
<property>
  <name>yarn.nodemanager.aux-services</name>
  <value>mapreduce_shuffle</value>
</property>
```

STEP 10:

Format namenode. Switch to Hadoop user

su hadoopuser

hadoop namenode-format

#It will take some time to fully format the namenode

Now it's time to Start Services. Remain in Hadoop user

start-all.sh

Then verify with this command:

jps Hit ENTER

All the 6 services to appear

(i.e. Jps/SecondaryNamenode/NodeManager/Datanode/ResouceManager/NameNode)

STEP 11:

Now it's time to verify hadoop works perfectly.

Access the Hadoop Web Interface:

NameNode: http://localhost:9870/

ResourceManager: http://localhost:8088/

Finally some tips:

Configuration Files: If you're looking for configuration files to edit (like core-site.xml, hdfs-site.xml, etc.), these should be within the etc/hadoop directory.

You can check this directory more closely:

ls -lrt /usr/local/hadoop/etc/hadoop

Executable Scripts: If you're looking for the scripts to start Hadoop services, look in the sbin directory:

ls -l /usr/local/hadoop/sbin

Hadoop Commands: If you're looking for Hadoop command-line tools, they are in the bin directory:

ls -l /usr/local

Problems might face:

1. Java and Hadoop Path does not exist.

Solution: Properly change the /.bashrc file with the commands listed above. See if there is any mistake there or over-written something.

2. Do not try to install Java from dnf install java-1.8.0-openjdk-devel command, because the company stopped installing option for java 8,11 and other older version directly.

Rather you should try from wget command listed above.