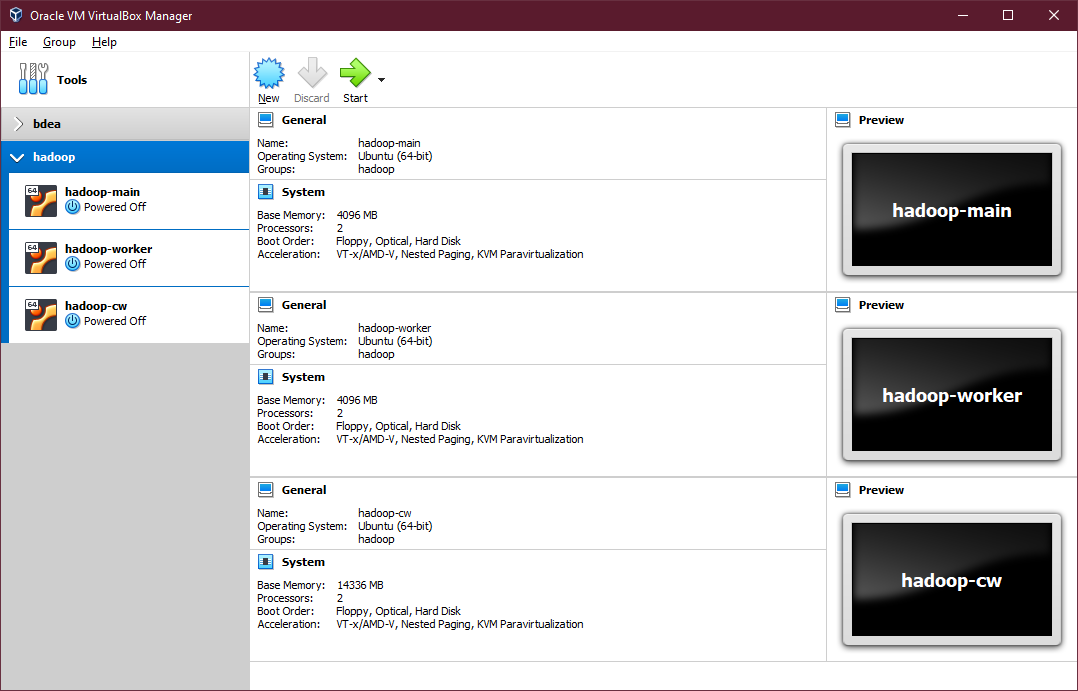
MS4S21 CW-1

Mark Baber – 17076749

# 1 – Experiment 1

This section will detail what was carried out for experiment 1, which was to create a Hadoop cluster using Ubuntu Virtual Machines (VMs) on our local machines. This will be done with [Oracles VirtualBox](#_x.1_-_VirtualBox) software. The main task was to replace the existing worker node with a new worker node with the following specifications:



(Figure 1 – Screenshot of the 3 node hadoop cluster)

Seeing as I ran into some issues with my existing cluster from the tutorials, I took a day to re-create the whole cluster again following the notes from the tutorials which I manipulated for myself. The next few steps will be the sequence of events which happened from start to finish, with a working 3 node cluster.

## – Hadoop-Main

After installing Ubuntu Desktop 20.04 Long Term Support ([Instruction Here](https://ubuntu.com/tutorials/install-ubuntu-desktop#1-overview)), I installed the updates and openssh-server so I could ssh into the main node from my computer using [Windows Terminal](#_x.2_-_Windows). This would allow me to copy and paste some of the commands I initially edited during the tutorials, to speed up my process of setting up the VMs.

### 1.1.1 – Set Hostname

I checked the hostname on the main node to make sure it was correctrly assigned as: **hadoop-main**. This was done by editing the file with ***sudo nano /etc/hostname*** and deleting the hostname and changing it to **hadoop-main**.

### 1.1.2 – Set Hosts

Next was to assign the hosts correctly for the main node and the two worker nodes – I already knew I was going to set a static IP as it is good practice, so these were hard coded in a range which I knew was free on my local network. Setting the hosts file was done by typing into a terminal, **sudo nano /etc/hosts** and adding the following:

* 127.0.0.1 localhost
* #127.0.1.1 hadoop-xyz (this would be VirtualMachine hostname)
* 192.168.1.173 hadoop-main
* 192.168.1.174 hadoop-worker
* 192.168.1.175 hadoop-cw

### 1.1.3 – Download and Install Java

To install java on Linux, this can be done with the follow code: **sudo apt install openjdk-8-jdk** which is an open-source version of java freely distributed. Now that java is installed, the path needs to be assigned to the **~/.bashrc** file. This can be done with the following code: **sudo nano ~/.bashrc** and add the following code at the very bottom of the file:

* # JAVA
* export JAVA\_HOME=/usr/lib/jvm/java-8-openjdk-amd64/jre

and apply these changes with this: **source ~/.bashrc** – this is all that is currently needed for setting up java on Linux.

### 1.1.4 – Download and Install Hadoop

This next section will cover how to download and install Hadoop which is required for our clusters, this code can be run from everywhere but I usually change directory to downloads when downloading files. This can be done with:

* **cd Downloads/**
* **wget** [**https://archive.apache.org/dist/hadoop/common/hadoop-3.2.2/hadoop-3.2.2.tar.gz**](https://archive.apache.org/dist/hadoop/common/hadoop-3.2.2/hadoop-3.2.2.tar.gz)

Next will be to extract the archive which can be done with **tar -xvf hadoop-3.2.2.tar.gz** and should be moved to /usr/local/hadoop, this can be done with the following code: **sudo mv hadoop-3.2.2 /usr/local/hadoop** – now that we have moved the hadoop files to our local user files, we would need to add the path to **~/.bashrc** again. This can be done with: **sudo nano ~/.bashrc** and adding the 3 lines to the bottom of the file (under the Java entry)

* # HADOOP
* export PATH=$PATH:/usr/local/hadoop/bin:/usr/local/hadoop/sbin
* export CONF=/usr/local/hadoop/etc/hadoop

and applying the changes again with **source ~/.bashrc** – this will conclude this section.

### 1.1.5 – Cloning the VM

After getting this far and setting up a few basic things, the next step is to clone the VM. This can be done by right clicking the VM and clicking clone with these settings:

Graphical user interface, text, email

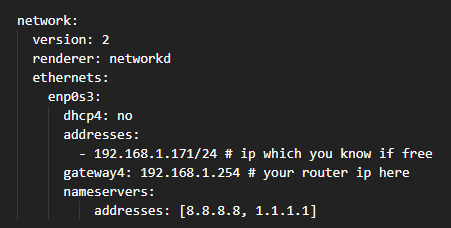
Description automatically generated

(Figure 2 – Screenshot of Cloning a VM)

And using the **Full Clone** option for the next setting – this will create (as the name suggests) a full clone of the VM but will generate a new mac address for the network adapter.

### 1.1.6 – Set Static IP

Now that the VM has been cloned, this is where I set a static IP for my VMs which I know are free on my local network. This can be done with the following code: **sudo nano /etc/netplan/01-network-manager-all.yaml** and pasting the following code:



(Figure 3 – Screenshot of how to set static IP)

### 1.1.7 – Generate SSH Keys

The next step will be to generate SSH keys from the main node to the two worker nodes, this is an easy task to do, but will need to be done after completing a few dependencies:

* [Changing the hostname of the 2 worker nodes](#_1.1.1_–_Set)
* [Setting static IPs for the 2 worker nodes](#_1.1.6_–_Set)

To generate your ssh keys from the main node, this can be done with: **ssh-keygen -t rsa** and accepting the default (which is not best practice, but this is only local). These keys would then need to be copied to the two worker nodes. This can be done with:

* **ssh-copy-id hadoop-cw@hadoop-main** # copy the key to main
* **ssh-copy-id hadoop-cw@hadoop-worker** # copy the key to worker 1
* **ssh-copy-id hadoop-cw@worker-cw** # copy the key to worker 2

These can be tested with all the machine running and typing from the main node: **ssh hadoop-cw@hadoop-worker** – you should be asked about accepting the connection and asked for the password. If successful you should see the terminal for **hadoop-cw@hadoop-worker**.

### 1.1.8 – Modifying Hadoop Files

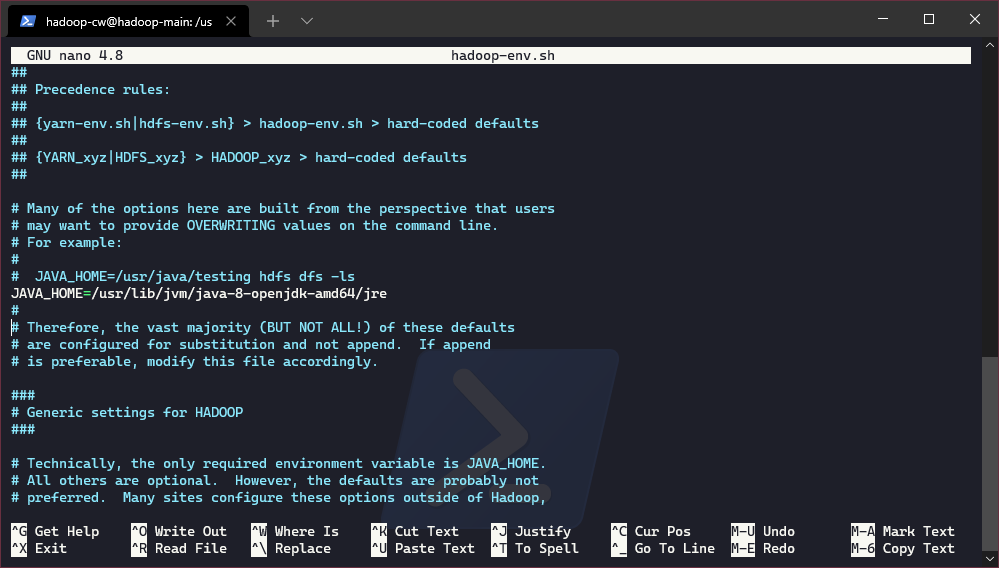
This step requires a few hadoop files to be modified, seeing as this will require going back and forth the path, we will create a command which will make changing directories much easier. This can be done by adding this code to the **~/.bashrc** file again, with: **sudo nano ~/.bashrc** and at the bottom of the file adding: **export CONF=usr/local/hadoop/etc/hadoop**, saving the changes and typing **source ~/.bashrc**. This will allow us to easily navigator to the hadoop folder with **cd $CONF/**.

#### 1.1.8.1 – hadoop-env.sh

Now this has been added, we need to change 4 files for the main node, these files are:

* hadoop-env.sh
* core-site.xml
* hdfs-site.xml
* workers

The first file will be changed with **sudo nano $CONF/hadoop-env.sh**, for this file we need to add the Java path which we created earlier. Within this file look for the java which is commented out and add our customised one like so:



(Figure 4 – hadoop-env.sh modified)

#### 1.1.8.2 – core-site.xml

Next we will need to change core-site.xml, this can be done with **sudo nano $CONF/core-site.xml**, within this file we want to add:

**<configuration>**

**<property>**

**<name>fs.defaultFS</name>**

**<value>hdfs://hadoop-main:9000</value>**

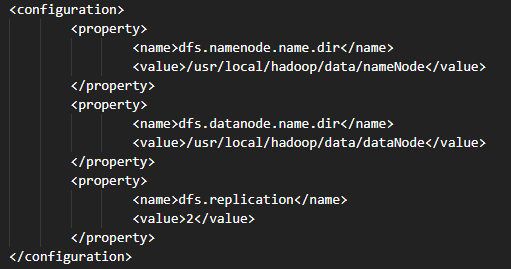
**</property>**

**</configuration>**

This is all for this file.

#### 1.1.8.3 – hdfs-site.xml

Next we will change hdfs-site.xml, this can be done with **sudo nano $CONF/hdfs-site.xml**, within this file we want to add:



(Figure 5 – hdfs-site.xml)

This is all we need for this file.

#### 1.1.8.4 – workers

The workers file is just there to list our worker nodes, for this we can access it with: **sudo nano workers**, within this file we want to add the two worker node hostnames, mine is as follows:

* hadoop-worker
* hadoop-cw

### 1.1.9 – Copy files to worker nodes

Now all these files have been modified, this is where we want to securely copy these files to our workers. This can be done with 2 lines of code:

* scp $CONF/\* hadoop-cw@bdea-worker:/usr/local/hadoop/etc/hadoop/
* scp $CONF/\* hadoop-cw@worker-cw:/usr/local/hadoop/etc/hadoop/

You should be presented with a success notification, and this concludes this section.

### 1.1.10 – Adding HADOOP to bashrc

To make sure we can call hadoop from everywhere on the machine, we want to add a few paths to bashrc. This can be done with **sudo nano ~/.bashrc** and within this file, again at the bottom add:

* export HADOOP\_HOME=/usr/local/hadoop
* export HADOOP\_COMMON\_HOME=$HADOOP\_HOME
* export HADOOP\_CONF\_DIR=$HADOOP\_HOME/etc/hadoop
* export HADOOP\_HDFS\_HOME=$HADOOP\_HOME
* export HADOOP\_MAPPED\_HOME=$HADOOP\_HOME

saving the changes and applying the changes with **source ~/.bashrc**. Now before moving on to running a map reduce task, there are a few more changes to add to the worker nodes.

## – Hadoop-Worker

To begin with our first worker, lets assigned a new hostname as it will be the same as the main node when we cloned the VM. Seeing as these instructions have already been described, I’ll just link to that section from [here](#_1.1.1_–_Set).

The next step is to make sure we have a static IP which we can use when connecting via ssh – again these steps have been mentioned so will be linked [here](#_1.1.6_–_Set).

### – Modify files

This section will show the files which need to be modified to get the workers up and running, the only file which needs to be modified is yarn-site.xml. Just make sure this file has this in the file:

* **<configuration>**
  + **<property>**
    - **<name>yarn.resourcemanager.hostname</name>**
    - **<value>hadoop-main</value>**
  + **</property>**
* **</configuration>**

Now this file has been modified, we want to copy this file to the other worker node, both are all set up. This can be done with:

* **scp yarn-site.xml hadoop-cw@worker-cw:/usr/local/hadoop/etc/hadoop/**

After this, we are finished for the worker

Software

## x.1 - VirtualBox – Version: 6.1.18 r142142 (Qt5.6.2)

## x.2 - Windows Terminal - Version: 1.7.1033.0