SCC.411 Week 14 Lab – Getting Started

In this lab you will complete the following objectives:

- Configure no password SSH and port forwarding into a Virtual Machine.
- Install and configure Hadoop environmental variables.
- Start and run a Hadoop Cluster in pseudo-distributed mode.
- Basic manipulation of the HDFS and some MapReduce examples.

The majority of Big Data systems are underpinned by a distributed system composed by multiple machines forming a cluster. Getting Big Data frameworks (processing, storage, etc) to correctly function requires lower-level configuration that we will be learning over the next few weeks.

Framework selection: Whilst SCC.411 lab exercises focus on Hadoop MapReduce and Hive, if you would like to investigate/deploy different Big Data frameworks that leverage multiple VMs (Spark, Cassandra, BigTable, Kubernetes etc.) you are welcome to do so, just keep in mind that we will be focusing support on the exercises provided here. If you are unsure kindly ask in in advanced.

Failure is fun!

Be prepared to repeatedly restart the installation process when you are first starting out.

Successful configuration, operation, and maintenance of any distributed system is a complicated and potentially delicate process. Hadoop is no exception. It is expected that you will encounter numerous challenges in getting a cluster to successfully operate and is part of the learning process.

While this setup guide will attempt to make this process as painless as possible, there will be subtle nuances that you will need to identify and correct related to networking, resource usage, and software configuration. We advise that if you encounter problems spend some time trying to identify and solve the problem yourself before immediately asking us or TAs for assistance.

1 VM Operation

All of your work will be performed using Virtual Machines (VMs) operated by the school, and are thus accessible if you are within the University VPN. You have been each assigned two VMs that we will give you the addresses for. I would recommend that you write these addresses down – you will be using them frequently.

For the first few weeks we will focus on using a single VM to become familiar with the environment, after which we will be connecting multiple VMs together in later exercises.

Using Unix Environments

As Big Data system installation and configuration makes heavy use of terminal-based systems, this module will require familiarity with navigating, copying, and editing files in Unix environments.

If you have had little to no exposure to working in Unix and using vim, chmod, ssh, and scp there are a large number of materials and tutorials that will be helpful:

https://learncodethehardway.org/unix/bash_cheat_sheet.pdf

If you are absolutely lost, we can guide you through the very basics.

2 General Advice

Backup & Automate: Back things up on a regular basis - you will likely be reformatting/restarting your system repeatedly. For example, when you create and populate your system with larger datasets, I would recommend that you extract key results out of the VM into your physical machine on a semiregular basis. Once you are confident with setting up and running the Hadoop platform I would also strongly recommend that you create a bash script to automate the entire process.

Careful Resource Management: When experimenting you will likely encounter difficulties with insufficient memory or disk storage to run your system. This can range from the application simply not running to the entire VM crashing. Kindly exercise caution and keep track of storage requirements (e.g. do not assign 20GB heap size to Java processes, or queries that create a massive Cartesian product).

Reflect on Exercises: When completing these worksheets, if you encounter a command you are unfamiliar with, I would strongly recommend you stop and do some research on what it means before using it. If you just follow the guide blindly without understanding, it will make it difficult for you diagnose problems, as well as tackle more open questions that you will encounter in later weeks.

Self-installation: You are welcome to install Hadoop on your own machines for experimentation. Be warned that the setup provided in this worksheet will likely deviate based on your OS. Be mindful that there are various versions of Hadoop available that vary quite considerably.

IMPORTANT! The power of sudo

To make the cluster function correctly you will require sudo root access (administrator rights) to configure parts of the VM. This means you will have full control over everything within the VM. Be extremely careful with this power. For example, it is possible for you to easily delete core components of the OS by accident (or even the entire OS itself!).

On each terminal line, you will see one of the following characters

#	<-	Indicates you are currently logged in as root (admin access)
$	<-	Indicates you are currently logged in as a regular user

I strongly recommend that you use only root when it is strictly necessary. Typically you will be prompted when executing particular commands.

3 Login to your VM

As mentioned in Section 1, you have each been assigned your own VM residing within the school cluster accessible via the VPN. You will need to connect to the VM via SSH. SSH (Secure Shell) is a network protocol for secure networking over an unsecured network. This will allow you connect to securely access remote machines.

There are multiple ways for you to access these VMs. The first is for you to connect to the teaching VMs provided by the school, and *then* connecting to the SCC.411 VMs. We would recommend this approach as it makes it somewhat easier to connect data to your University H: drive.

You can access the VM by using VMWare Horizon, accessible via:

https://lancs-lab-uag.lancaster.ac.uk/portal/

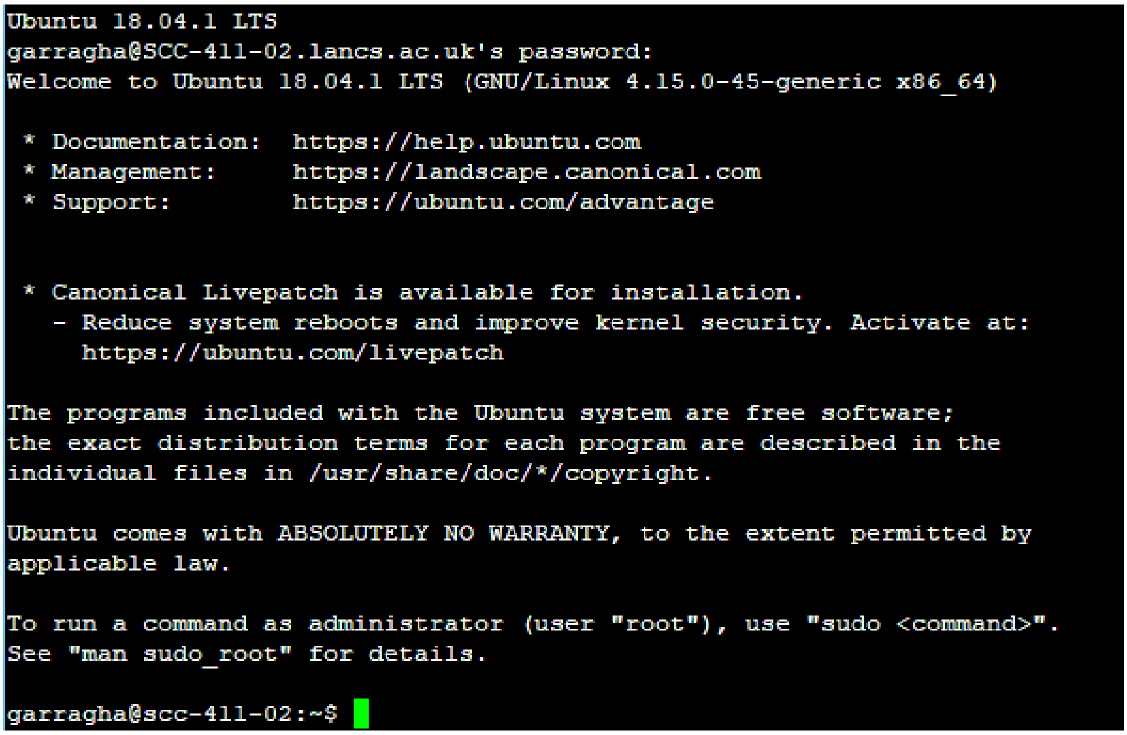
You will then want to select *SCC Ubuntu*, after loading you can use your student logon credentials to gain access to the Linux environment.

Accessing your SCC.411 VMs can be performed via the following terminal command:

```
ssh <your student user name>@<your assigned VM hostname>
```

As a reminder, you will be provided with your VM login details before the start of your first lab.

You will be prompted for your username and password, successful login will look something like this:



```
Ubuntu 18.04.1 LTS
garragha@SCC-411-02.lancs.ac.uk's password:
Welcome to Ubuntu 18.04.1 LTS (GNU/Linux 4.15.0-45-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage


 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

garragha@scc-411-02:~$
```

Alternatively, providing you are on the University VPN, you should be able to access the SCC.411 VMs directly from your own personal computer or an application such as Putty (if using Windows).

4 Add Hadoop user

We are going to add a new user named *hadoop* to your system, you will need root access to create a new user. Hence, use the following command:

```
sudo adduser hadoop
```

You will be prompted to enter a new UNIX password, you can either use your current student login or another password of your choice. You will be prompted to provide other information such as Full Name, Room Number, Work Phone, etc. Feel free to fill these in or keep these empty.

Log into your newly created Hadoop user via:

```
su hadoop
```

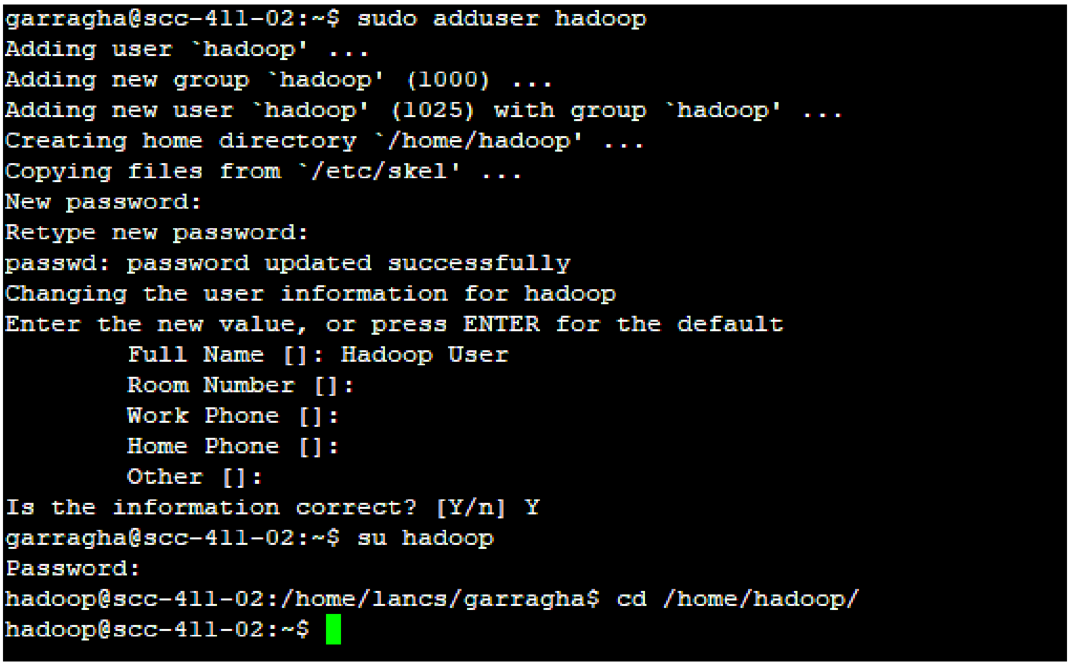
Important: Don't lose your hadoop password! Unless you want to delete all your data contents and start again next time you login. It is your choice whether you wish to use your student password (not advised) or create a new one.

Alternatively, you should be able to alter your hadoop password as root user.

You can delete the user at any time using the command

```
userdel hadoop
```

Remember to also remove the newly created hadoop directory if you wish.



```
garragha@scc-411-02:~$ sudo adduser hadoop
Adding user `hadoop' ...
Adding new group `hadoop' (1000) ...
Adding new user `hadoop' (1025) with group `hadoop' ...
Creating home directory `/home/hadoop' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for hadoop
Enter the new value, or press ENTER for the default
        Full Name []: Hadoop User
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] Y
garragha@scc-411-02:~$ su hadoop
Password:
hadoop@scc-411-02:/home/lancs/garragha$ cd /home/hadoop/
hadoop@scc-411-02:~$
```

The Hadoop User

Why are we logging into a VM with your student login and then immediately making a new user Hadoop? The reason is two-fold. First, it is good practise to assign users of user groups for specific tasks. Second and more importantly, Hadoop has traditionally been configured to operate assuming the username is Hadoop (unless explicitly configured otherwise) for operation and setting permissions.

Hence, making a hadoop user simply minimizes the amount of configuration and the risk of issues

5 No Password SSH

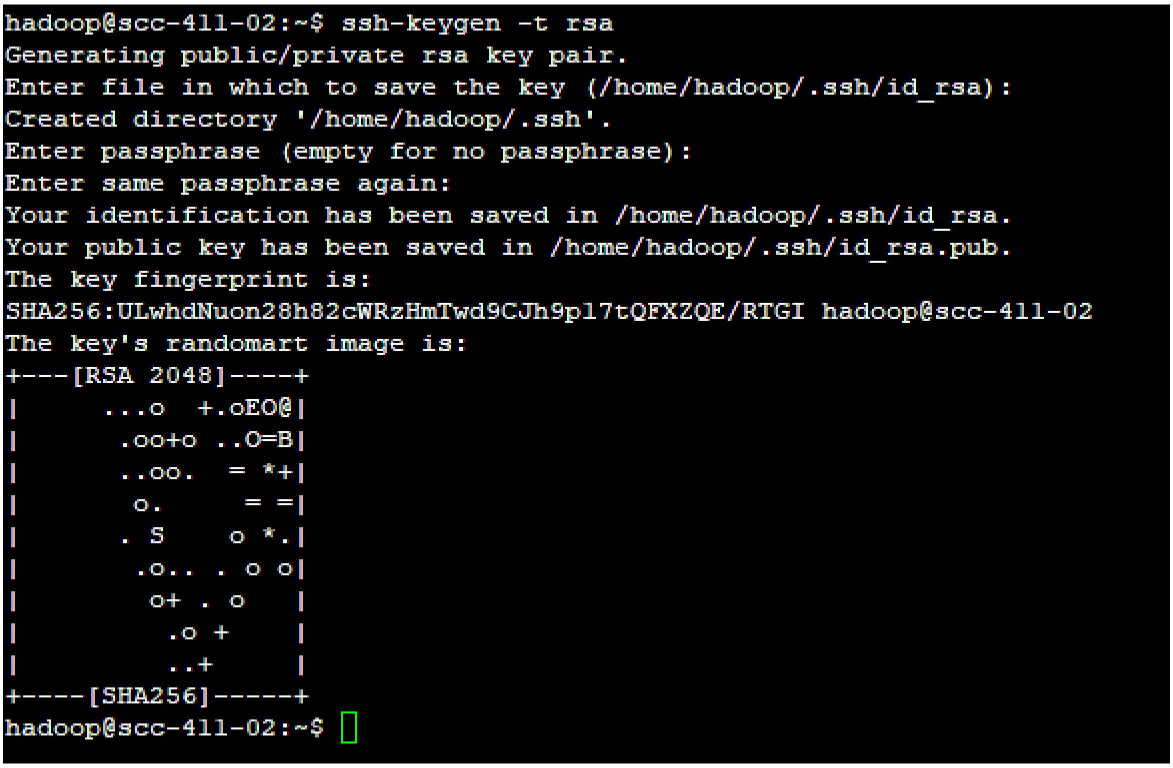
An interesting feature of SSH is that you can use it to setup no password login between remote machines. This has the advantage of making logging into remote systems less tedious, and in the context of Hadoop is required to successfully interconnect multiple machines together.

Hadoop is designed to operate within single and multiple machine environments. Thus, for Hadoop to function correctly you will need to allow no password SSH. As we are starting by setting up Hadoop on a single machine, we will have to enable our VM to no password SSH into itself.

To perform this action, you will need to generate Public and Private Key Pairs. This can be achieved via the following command:

```
ssh-keygen -t rsa
```

You will be prompted to specific which directory to save the key as well as a new password; keep both of these blank for now. This should create a new directory */home/hadoop/.ssh*, and a generate file named *id_rsa.pub*.



```
hadoop@scc-411-02:~$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/hadoop/.ssh/id_rsa):
Created directory '/home/hadoop/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/hadoop/.ssh/id_rsa.
Your public key has been saved in /home/hadoop/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:ULwhdNuon28h82cWRzHmTwd9CJh9pl7tQFXZQE/RTGI hadoop@scc-411-02
The key's randomart image is:
+---[RSA 2048]----+
|     ...o  +.oEO@|
|      .oo+o ..O=B|
|      ..oo.  = *+|
|       o.      = =|
|      . S    o *.|
|       .o.. . o o|
|        o+ . o   |
|         .o +    |
|         ..+     |
+----[SHA256]-----+
hadoop@scc-411-02:~$
```

You will need to copy the contents of the public key id_rsa.pub into a file *authorized_keys*

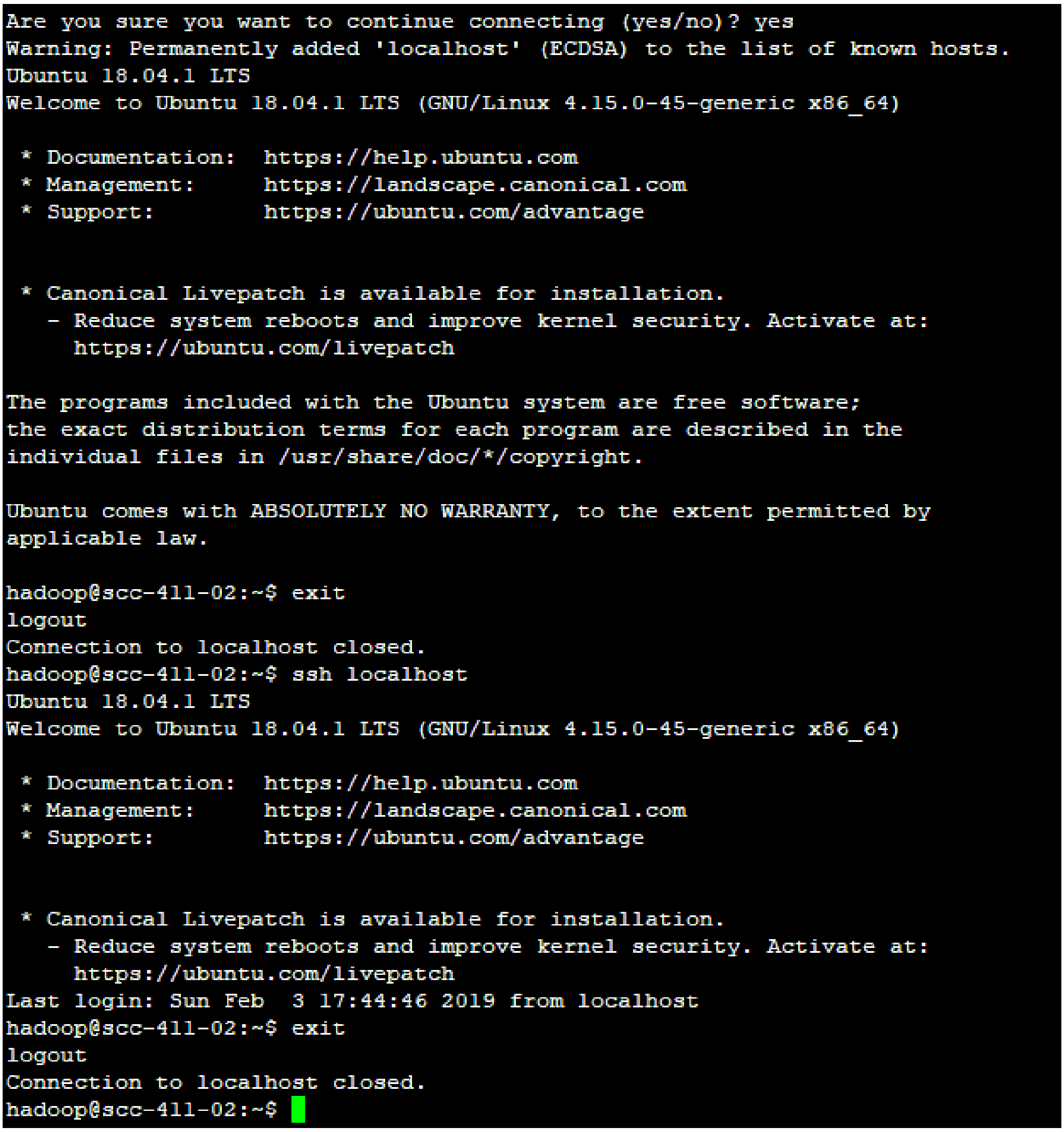
```
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
```

If you inspect the contents of the authorized_keys file, you should see that it now contains text. You can verify whether no password SSH is successful by logging into your VM again via localhost:

```
ssh localhost
```

You will be again prompted to provide your password for the user Hadoop. After logging into the terminal type, *exit* to return. Type *ssh localhost* again and you should now not be prompted for a password.

If you are prompted for your password again, you will need to delete your authorized_keys file, as well as your id_rsa and id_rsa.pub files and try the above process again.



```
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.
Ubuntu 18.04.1 LTS
Welcome to Ubuntu 18.04.1 LTS (GNU/Linux 4.15.0-45-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage


 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

hadoop@scc-411-02:~$ exit
logout
Connection to localhost closed.
hadoop@scc-411-02:~$ ssh localhost
Ubuntu 18.04.1 LTS
Welcome to Ubuntu 18.04.1 LTS (GNU/Linux 4.15.0-45-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage


 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch
Last login: Sun Feb  3 17:44:46 2019 from localhost
hadoop@scc-411-02:~$ exit
logout
Connection to localhost closed.
hadoop@scc-411-02:~$ 
```

6 Install Hadoop

We now need to download Hadoop into your VM to install it. You can download the files from the Apache archive through wget:

```
wget http://archive.apache.org/dist/hadoop/common/hadoop-2.8.5/hadoop-2.8.5.tar.gz
```

After successful downloading, place the file within your hadoop home directory */home/hadoop/* if you haven't already.

You will need then to extract the .tar file using the following command:

```
tar -xzvf hadoop-2.8.5.tar.gz
```

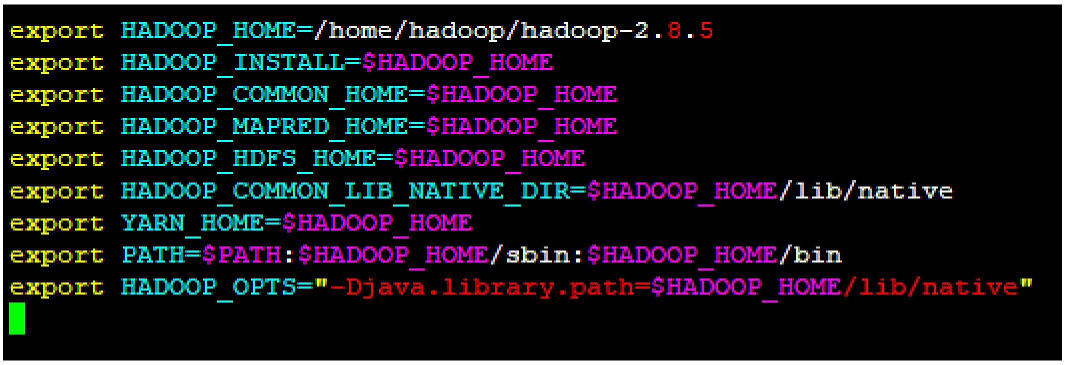
You should now have created a new directory named *hadoop-2.8.5*.

7 Environment Configuration

For Hadoop to operate correctly (and to make it easier to execute basic Hadoop commands) you will need to set variables across multiple files. You are able to edit these files via vim or nano.

First, we are going to edit your *.bashrc* for user hadoop. The file *bashrc* can be found at */home/hadoop/.bashrc*

Add and save the following information to the bottom of the *.bashrc* file:



```
export HADOOP_HOME=/home/hadoop/hadoop-2.8.5
export HADOOP_INSTALL=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export YARN_HOME=$HADOOP_HOME
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib/native"
```

You will need to source this file for your current login session using:

```
source ~/.bashrc
```

Note: What do you think the last line HADOOP_OPTS actually does? What would happen if we removed 'native' from the directory? Moreover, if you are unsure what bashrc actually does, we would suggest to look it up online – it does a lot of fancy stuff we take for granted!

7.1 hadoop-env.sh

Next we will be editing and configuring variables within Hadoop itself. This will control everything from data replication to heap size assigned to processes. To start, we are going to provide the basic configuration to get Hadoop and the Hadoop Distributed File System (HDFS) successfully running.

Hadoop variables
There are a vast amount of variables that you could potentially play with. These can be found:

https://mapr.com/docs/61/ReferenceGuide/ConfigurationFiles.html

Navigate to */etc/hadoop* directory within hadoop-2.8.5 (*/home/hadoop/hadoop-2.8.5/etc/hadoop/*) that should contain several files. Open hadoop-env.sh and edit the file to include the following:

```
export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64/

export HADOOP_CONF_DIR=${HADOOP_CONF_DIR:-"/home/hadoop/hadoop-2.8.5/etc/hadoop"}
```

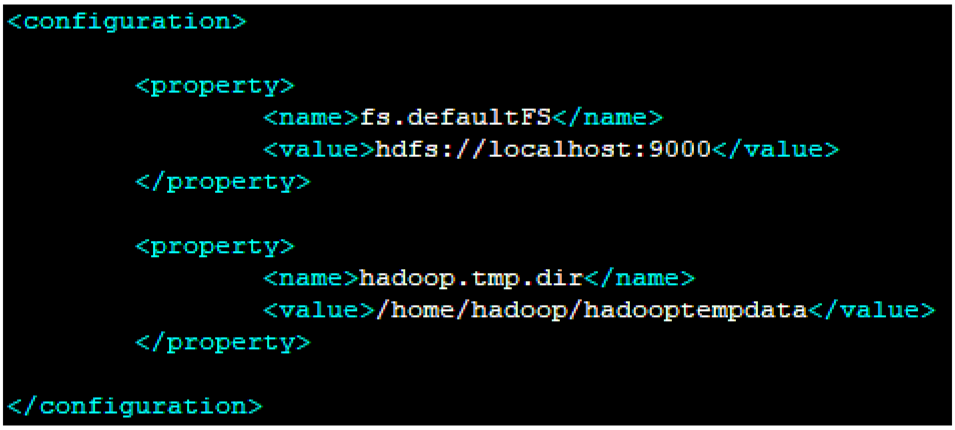
This points Hadoop to where java and Hadoop configuration directory is located on your VM.

You can check any time what your current java execution is by using the command

```
java -version
```

7.2 core-site.xml

The *core-site.xml* is found in the same directory as *hadoop-env.sh (/home/hadoop/hadoop-2.8.5/etc/hadoop*). Add the following text:



```
<configuration>

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

        <property>
                <name>hadoop.tmp.dir</name>
                <value>/home/hadoop/hadooptempdata</value>
        </property>

</configuration>
```

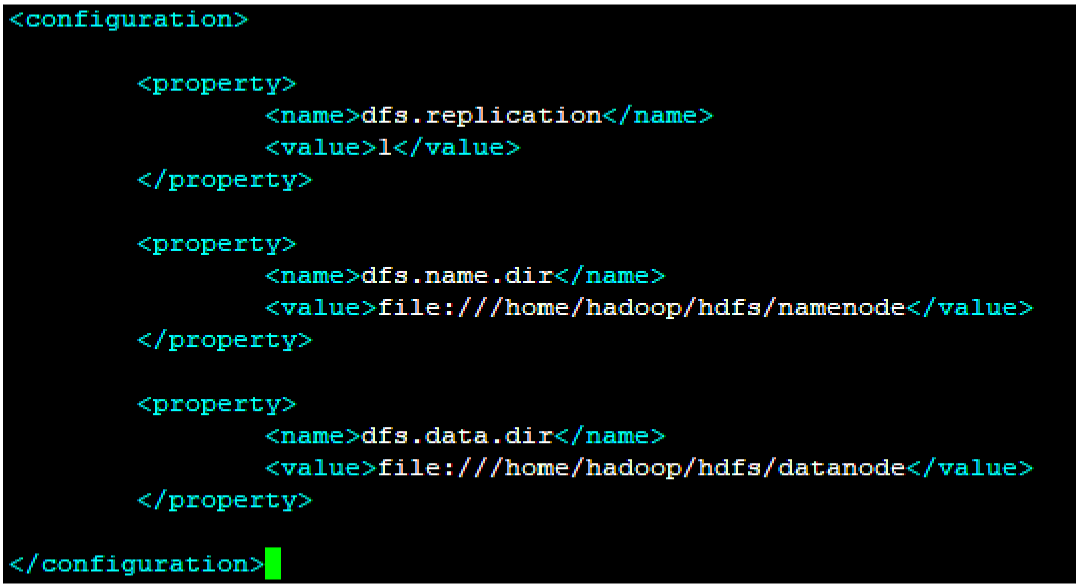
As the above variable requires a directory to store temporary data, you will also need to actually make the directory before starting Hadoop.

```
mkdir hadooptempdata
```

Make sure you create this in the correct directory (*mkdir* will create a directory in whatever directory you are currently inside, so before doing so go back to */home/hadoop*)

7.3 hdfs-site.xml

The Hadoop Distributed File System (HDFS) allows for distributed storage across multiple machines. Edit the *hdfs-site.xml* found in the same directory again *(/home/hadoop/hadoop-2.8.5/etc/hadoop*) and add the following:



```
<configuration>

        <property>
                <name>dfs.replication</name>
                <value>1</value>
        </property>

        <property>
                <name>dfs.name.dir</name>
                <value>file:///home/hadoop/hdfs/namenode</value>
        </property>

        <property>
                <name>dfs.data.dir</name>
                <value>file:///home/hadoop/hdfs/datanode</value>
        </property>

</configuration>
```

You will once again need to create two new directories *namenode* and *datanode* (make sure you are within the home/hadoop directory when doing so).

```
mkdir -p hdfs/namenode

mkdir -p hdfs/datanode
```

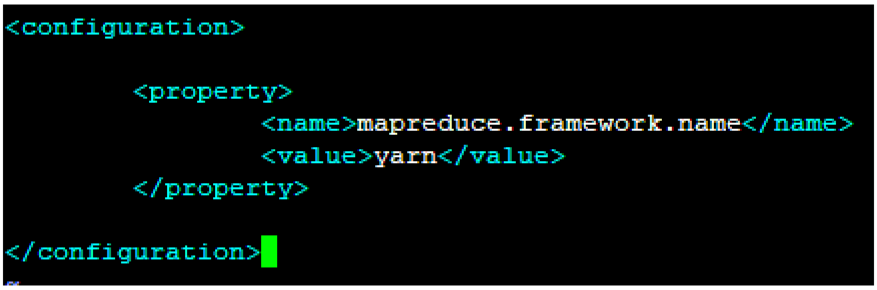
7.4 mapred-site.xml

again (/home/hadoop/hadoop-2.8.5/etc/hadoop)

Create a new mapred-site.xml file form the existing template via the following command:

```
cp mapred-site.xml.template mapred-site.xml
```

Then add the following:

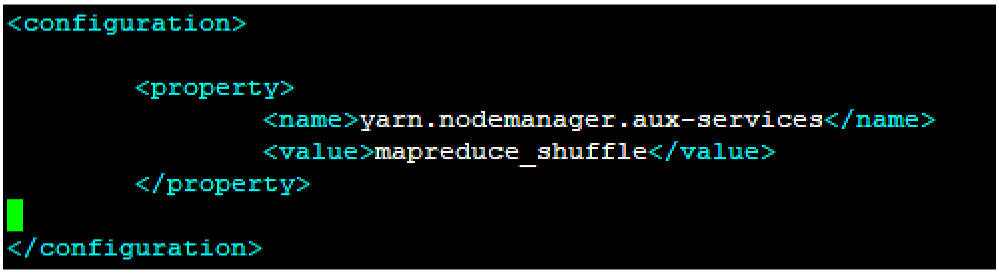


```
<configuration>

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

</configuration>
```

This will make MapReduce use the YARN framework (more detail about YARN will be discussed in later lectures).

7.5 yarn-site.xml

Add the following to the yarn-site.xml file



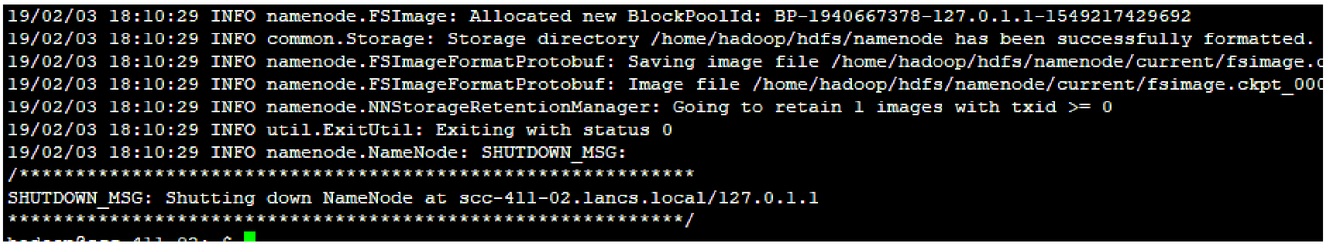
```
<configuration>

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

</configuration>
```

That's nearly all the configuration we need to do! A couple more things before we can actually begin starting Hadoop…

8 Hadoop Start

Before loading Hadoop we need to format the namenode, this can be performed by the command:

```
hdfs namenode -format
```



```
19/02/03 18:10:29 INFO namenode.FSImage: Allocated new BlockPoolId: BP-1940667378-127.0.1.1-1549217429692
19/02/03 18:10:29 INFO common.Storage: Storage directory /home/hadoop/hdfs/namenode has been successfully formatted.
19/02/03 18:10:29 INFO namenode.FSImageFormatProtobuf: Saving image file /home/hadoop/hdfs/namenode/current/fsimage.
19/02/03 18:10:29 INFO namenode.FSImageFormatProtobuf: Image file /home/hadoop/hdfs/namenode/current/fsimage.ckpt_000
19/02/03 18:10:29 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >= 0
19/02/03 18:10:29 INFO util.ExitUtil: Exiting with status 0
19/02/03 18:10:29 INFO namenode.NameNode: SHUTDOWN_MSG:
/************************************************************
SHUTDOWN_MSG: Shutting down NameNode at scc-411-02.lancs.local/127.0.1.1
************************************************************/
```

Important: The command *namenode -format* **will delete ALL data currently within your hadoop file system**. While this is the easiest way to solve problems related to cluster operation and data synchronization, this comes at the price of losing all your data. Echoing previous advice, for the purposes of learning how Hadoop works this is fine, but you may wish to exercise caution once you begin playing with larger amounts of data.

Start the HDFS by the command:

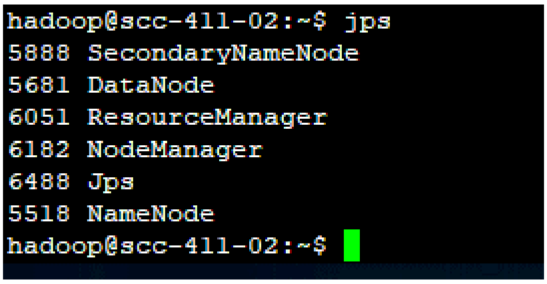
```
start-dfs.sh
```

After this has completed, start YARN with the command:

```
start-yarn.sh
```

I would advise for you to inspect each of these scripts to understand exactly what they are doing.

Use the *jps* command to view currently executing java process. If you have configured Hadoop correctly, you should see six processes running.



```
hadoop@scc-411-02:~$ jps
5888 SecondaryNameNode
5681 DataNode
6051 ResourceManager
6182 NodeManager
6488 Jps
5518 NameNode
hadoop@scc-411-02:~$
```

If you are missing a process (for example the DataNode) it means there is a misconfiguration somewhere within your setup. You can begin to diagnose this problem by inspecting Hadoop log files within the *hadoop-2.8.5/logs/* directory.

You can stop Hadoop at any time with the following commands:

```
stop-yarn.sh
stop-dfs.sh
```

Finally, you can run the command *hadoop version* and *hdfs version* to check whether they have extracted correctly.

9 HDFS

The HDFS is built to handle distributed storage across multiple machines, but also operates within a single machine.

You can create directories within the HDFS through the command line by doing the following:

```
hdfs dfs -mkdir /firstdir
hdfs dfs -mkdir /seconddir
hdfs dfs -ls /
```

10 Add data to HDFS

Let's pull some data from an open data repository and add it to the HDFS. The linked data below contains chronic disease indicators from the US.

```
https://catalog.data.gov/dataset/u-s-chronic-disease-indicators-cdi
```

Download the .csv version of this file. You can either download this to your physical machine then *scp* (secure copy command) onto the VM or pull directly into your VM via *wget*. Make sure you change the file name to something more understandable such as *chronic_disease.csv* Feel free to browse and inspect the data.

Insert the data into the HDFS via the put command

```
hdfs dfs -put /home/hadoop/chronic_disease.csv  /diseasedata
```

You can check whether it has been successfully added by issuing the following command

```
hdfs dfs -ls /diseasedata
```

Alternatively, you can extract data from the HDFS via the get command.

There are a variety of other commands to manipulate data inside the HDFS that shares many commonalities with basic Unix commands (*cp*, *du*, *etc*). A list of these commands can be found via the command:

```
hdfs dfs -help
```

I would advise you experiment with these commands yourself. For example, try to find out how much storage space is currently used per directory, and copying/moving data between directories.

To check that MapReduce is working, there are a number of examples programs that can be run.

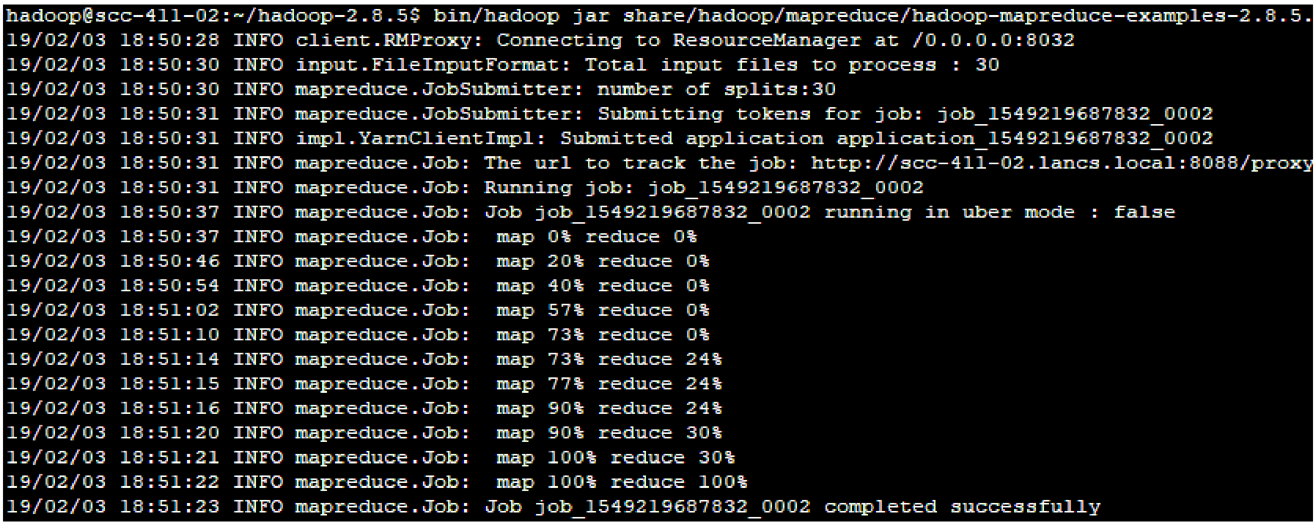
Make a new directory within the HDFS called */user/hadoop*. Next, copy the files within *etc/hadoop* into a directory called input.

If this has been performed correctly you should be able to view all the configuration files under */user/hadoop/input* within the HDFS.

Next, from within the *hadoop-2.8.5* directory, run the following command:

```
bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-
2.8.5.jar grep input output 'dfs[a-z.]+'
```

If successful, your terminal should resemble the following:



```
hadoop@scc-411-02:~/hadoop-2.8.5$ bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.8.5.
19/02/03 18:50:28 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
19/02/03 18:50:30 INFO input.FileInputFormat: Total input files to process : 30
19/02/03 18:50:30 INFO mapreduce.JobSubmitter: number of splits:30
19/02/03 18:50:31 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1549219687832_0002
19/02/03 18:50:31 INFO impl.YarnClientImpl: Submitted application application_1549219687832_0002
19/02/03 18:50:31 INFO mapreduce.Job: The url to track the job: http://scc-411-02.lancs.local:8088/proxy
19/02/03 18:50:31 INFO mapreduce.Job: Running job: job_1549219687832_0002
19/02/03 18:50:37 INFO mapreduce.Job: Job job_1549219687832_0002 running in uber mode : false
19/02/03 18:50:37 INFO mapreduce.Job:  map 0% reduce 0%
19/02/03 18:50:46 INFO mapreduce.Job:  map 20% reduce 0%
19/02/03 18:50:54 INFO mapreduce.Job:  map 40% reduce 0%
19/02/03 18:51:02 INFO mapreduce.Job:  map 57% reduce 0%
19/02/03 18:51:10 INFO mapreduce.Job:  map 73% reduce 0%
19/02/03 18:51:14 INFO mapreduce.Job:  map 73% reduce 24%
19/02/03 18:51:15 INFO mapreduce.Job:  map 77% reduce 24%
19/02/03 18:51:16 INFO mapreduce.Job:  map 90% reduce 24%
19/02/03 18:51:20 INFO mapreduce.Job:  map 90% reduce 30%
19/02/03 18:51:21 INFO mapreduce.Job:  map 100% reduce 30%
19/02/03 18:51:22 INFO mapreduce.Job:  map 100% reduce 100%
19/02/03 18:51:23 INFO mapreduce.Job: Job job_1549219687832_0002 completed successfully
```

If you wish to view this data for yourself, you will want to extract the data from the output directory via the get command.

11 Namenode and YARN manager

It is possible to view the operational status of a Hadoop cluster (including currently executing jobs) via the web browser. Because you are connecting to a VM that only contains terminal, you will need to allow the browser from your lab machine to tunnel into the VM. This can be achieved by port forwarding.

> **Note:** It's possible that this section may not work for your environment. In testing we found that this worked for some and not for others (typically resultant of changes to Chrome and/or the University port access). Thus, do not worry if you are able to complete section 11 – you may see this as an extra "nice to have", rather than something that is required.

If you are not familiar with the concepts of port forwarding and SSH tunnelling I would recommend you look up these concepts, as well as subsequent advantages and disadvantages of doing so.

Whilst Hadoop is running within the VM, open a new terminal and type the following command:

```
ssh -D 9537 -C hadoop@<your_VM_hostname_here>
```

Close Chrome if it is currently running, and then open another new terminal and type the command:

```
google-chrome –proxy-server="socks://127.0.0.1:9537"
```

Now within the newly opened browser enter the URLs 127.0.0.1:8088 and 127.0.0.1:50070. You should be able to see both the cluster manager and namenode managers. If you are receiving server not found messages, it means you are not connected to the server, and you may need to repeat this process again (and/or restart Hadoop).

Try running another MapReduce job, and whilst it is running refresh the cluster manager page. You should see the status of the job, as well as inspect log data for completed jobs.

12 Further exercises

Congratulations, you have successfully setup Hadoop, manipulated the HDFS, and ran your first MapReduce job!

There are a variety of other types of examples MapReduce jobs that you can experiment with (WordSort, WordCount, etc). Open up the .jar files on your physical machines to see how the operate. For example, try to use the disease data you uploaded previously in conjunction with some of the available code. You should also investigate some of the **-env.sh* files that provide more detail how Hadoop performed setup across multiple machines, and assigns appropriate process memory.

13 Troubleshooting FAQ

I don't have the correct permissions to perform the necessary commands

Try using sudo to issue the command. Remember that hadoop itself is not part of sudo, thus if you want to become root you will either need to exit to your student ID in the VM, or start another terminal to access the same VM. If you are still denied access even as root user let us know.

My namenode/datanode is missing when I run jps

This can occur due a number of reasons including port binding, misconfiguration. insufficient memory, and incorrect permissions. First make sure that you have made no typos in environment configuration detailed within Section 7. Depending on how heavily you have altered your VM, restarting it might also be useful. This can be performed by the command

```
/sbin/shutdown -r now
```

You will need to wait for approximately 30 seconds or so to try reconnect again.

HDFS appears to be broken

The easiest way is to wipe a clean slate for the HDFS, this can be done via the command HDFS

```
namenode –format
```

The managers are not displaying in the Internet browser

This is likely due to not all Hadoop processes running (check with *jps*) or incorrect port forwarding. Try performing the steps in Section 12 again, as well as try restarting Hadoop.

I have gone back constantly to make changes, and now I'm lost/it's a mess

Don't worry, it is rare to get it working entirely first time, and part of the learning process! The log files within Hadoop are a good starting place to detect potential issues, however they might not adequately describe the precise cause. Given we currently have no critical data, sometimes it might be easier to just start again afresh. If you decide to do this make sure that you delete the hadoop directory, variables you've added within *.bashrc,* and the namenode and datanode directories.