MS4S21 - Big Data Engineering and Applications

Moizzah Asif, J418, moizzah.asif@southwales.ac.uk

University of South Wales

Contents

Apache Hadoop multi-node cluster on Ubuntu VMs

This chapter will walk you through step-wise general guidelines of creating a Apache hadoop cluster 3.2.1 at present using Ubuntu latest stable release via VMs.

1.1 VM creation

Please note these steps can be followed to create any OS's VM in general, however the Linux Ubuntu 19.10 is used as example here.

1. Open virtual box and click on the new icon.
2. You are about to create a Ubuntu VM, set the values of each field as shown in Fig 1.1, and then press continue. Please note that chose the machine folder based on where you want to save it on your computer. If you are using university computer, then please choose your network drive.

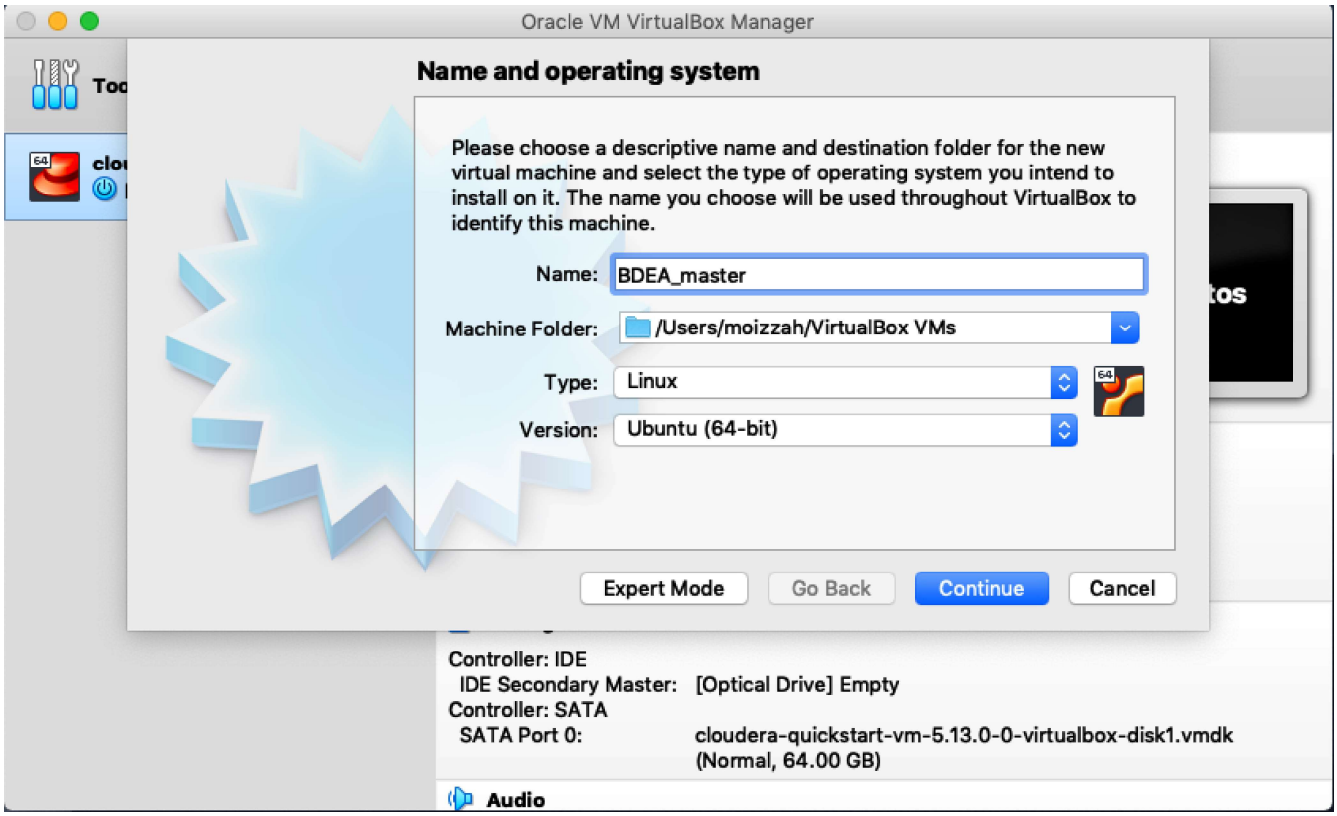

Figure 1.1: Ubuntu VM creation on virtual box - 1

3. Set RAM/memory to 12 GB/ 12288 MBs, and press continue.

4. Select the radio button which enables you to create a virtual hard disk, and then press create.

5. It will be followed by a pop up window where you can specify the hard disk file type as shown in Fig 1.2, please select VHD (virtual Hard Disk) to stay consistent with lectures, and then press continue.

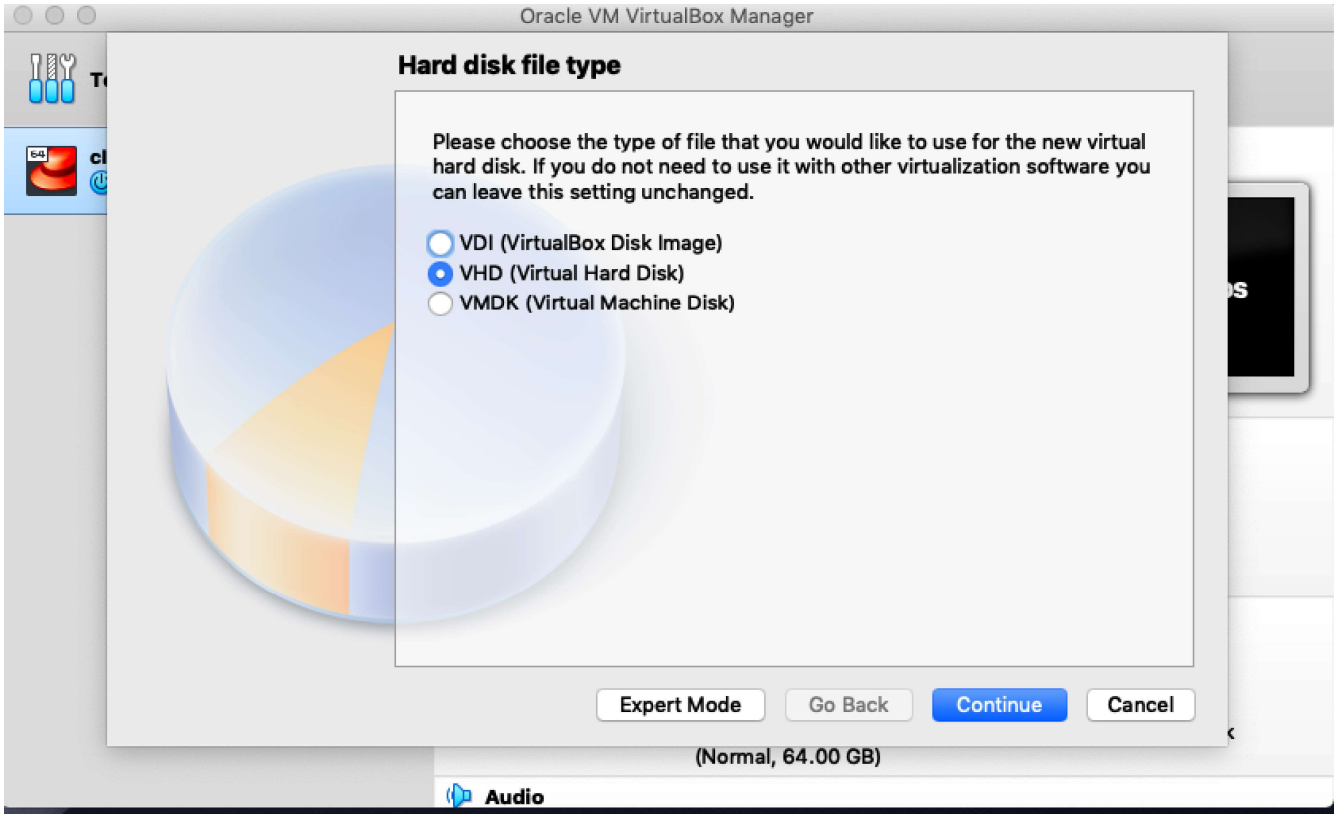

Figure 1.2: Ubuntu VM creation on virtual box - Hard disk file type

6. Select Fixed size radio button for physical storage on hard disk, press continue and then either keep or change the default value of 10 GB. This decision should be made based on your local machine's available hard drive storage, as well as the number of VMs you will have to run at once. In this case consider at least two VMS at a time to simulate the cluster. Press continue and wait for the VM to be created. Please note that you have just configured the physical and memory storage requirement until now. This is similar to brining home a computer which doesn't have any operating system installed on it.

7. You should be able to see the VM on Virtual Box left pane now. As shown in Fig 1.3 the name of this empty VM should be what you named it in the first step of creation.

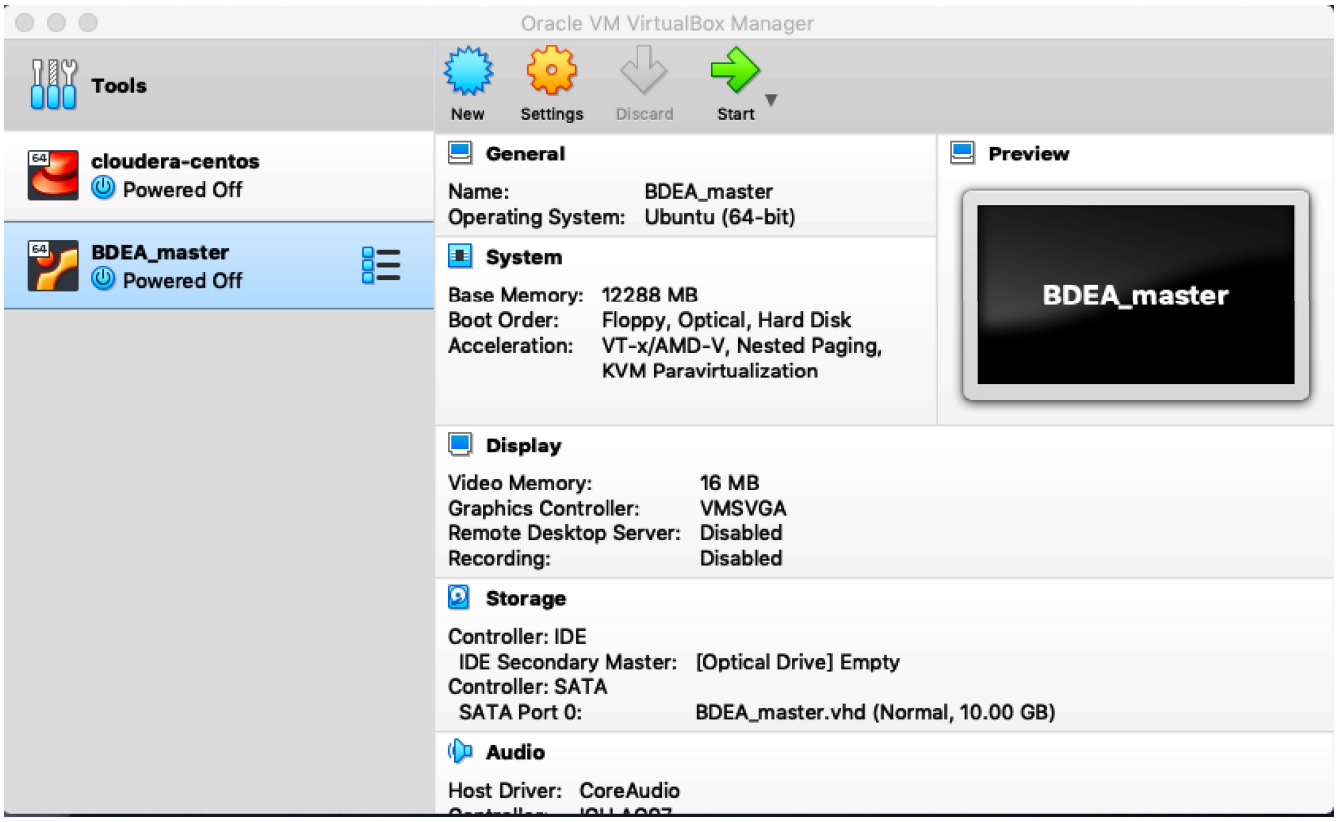

Figure 1.3: Ubuntu VM creation on virtual box - Empty VM

1.2 Ubuntu 20.04 installation

You will have to use the .iso image provided in BB Week 1 folder or the one you have downloaded on your computer or saved on your uni network drive.

1. Start the VM that you have just created in the previous section; click on the folder icon with a green arrow; Click on the add icon in the new pop up window;select the .iso image from the location where you have saved/downloaded it; The file should appear on the pop up windows as shown in Fig 1.4; choose this file and process with installation in the next window that pops up on your screen. The essential options in the process are listed below, please make sure that you have selected them.

   (a) install ubuntu

   (b) English UK

   (c) normal installation

   (d) erase disk an install ubuntu(it's empty already)

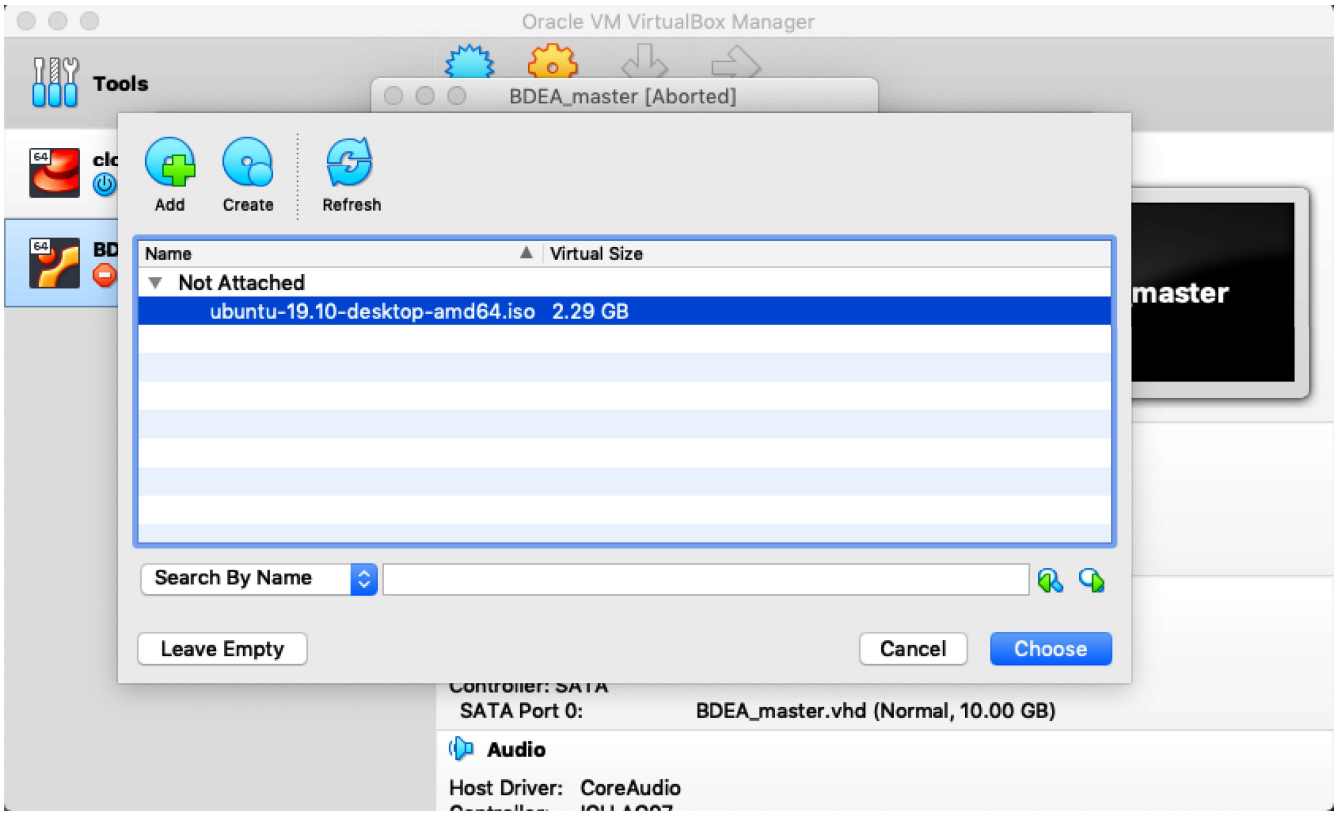

Figure 1.4: Ubuntu VM creation on virtual box - adding iso image 1

2. The screen display shown in Fig 1.5 helps you create a user profile on ubuntu. Please name the user profile intuitively. A user profile BDEA (Big Data Engineering and Applications) will be created for this tutorial's purpose.

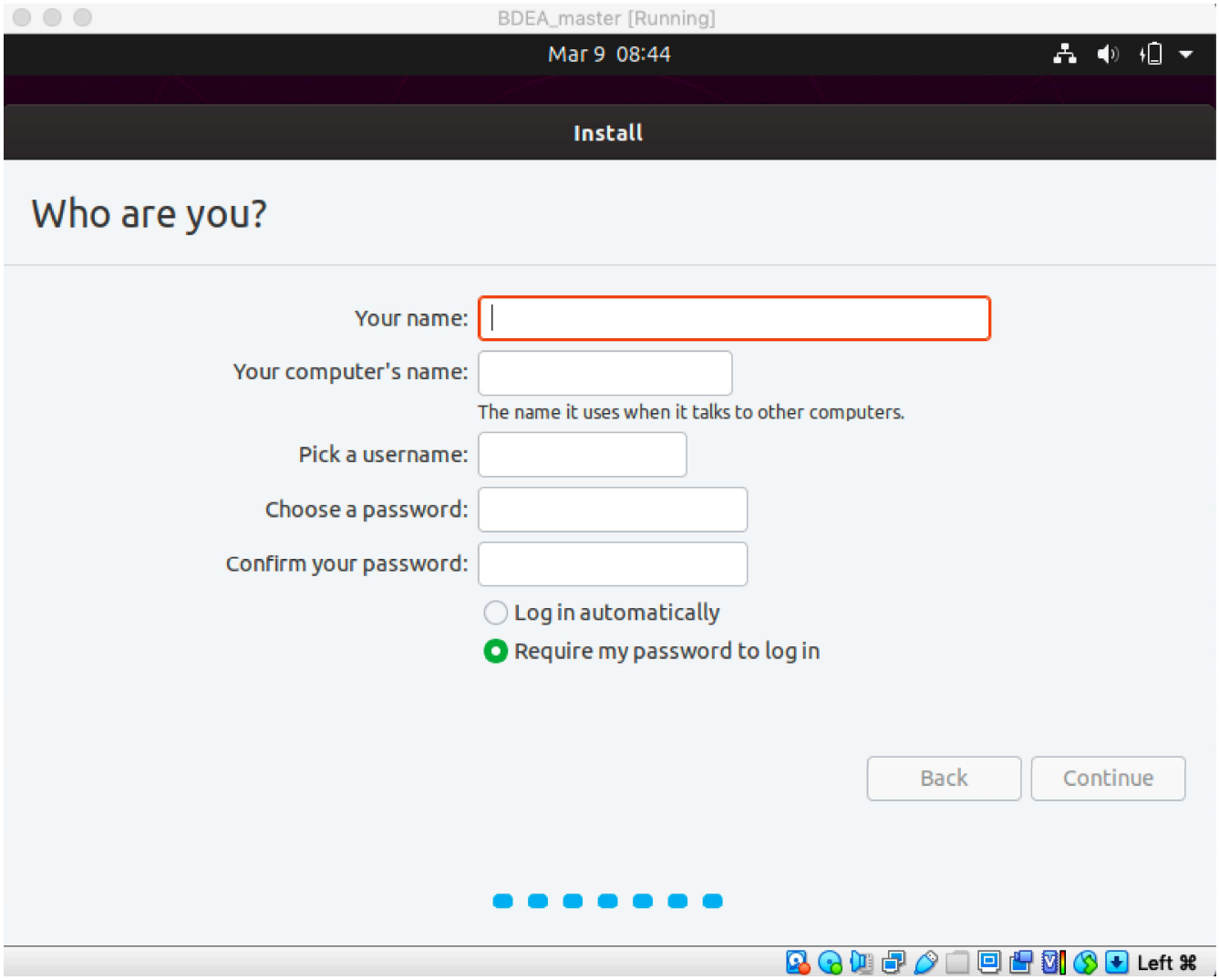

Figure 1.5: Ubuntu user creation

1.3 Setting up Ubuntu for Hadoop-3.2.2 hdfs cluster

Once you arrive at your Ubuntu's desktop, open Terminal so that you may set up the machine for hadoop-3.2.2 installation. Follow the guidelines listed below after opening terminal.

For more information on linux terminal please use the forum in blackboard module.

1. upgrade and update apt with sudo priviliges (APT - advanced packaging tool) - *sudo apt update/upgrade*
2. Install openssh server - *sudo apt install openssh-server*; verify the status now: *sudo systemctl status ssh*. You should be able to get an output similar to Fig 1.6
3. check hostname - *sudo hostname*; rename to hadoop-master - *sudo hostname hadoop-master*; verify successful renaming as shown in Fig 1.6.

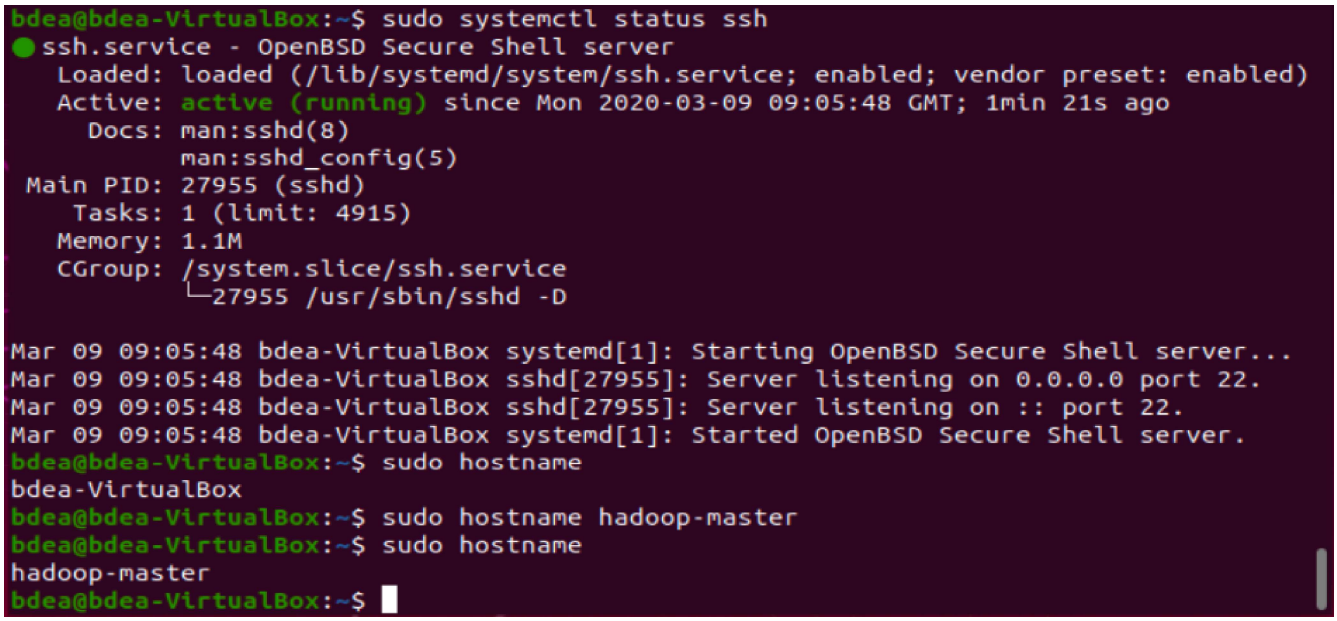

Figure 1.6: openssh-server and hostname verification

4. install gedit text editor for smooth editing - *sudo apt install gedit*. If you are working on remote hosts, this editor may not be available and you may have to work with vim. (A healthy discussion on vim had been conducted during lectures. for more info, please create a thread in the forum, or use one if it is already there.)

5. update the hosts file by adding hadoop-master as a host.

   (a) Install ifconfig to find the ip address of your machine via terminal - *sudo apt install net-tools*. Fig 1.7 shows the output and script of installing net-tools and listing the internet configurations of your machine.

   (b) Find the ip address of your vm - *ifconfig*. Fig 1.7 shows the ip address highlighted manually.

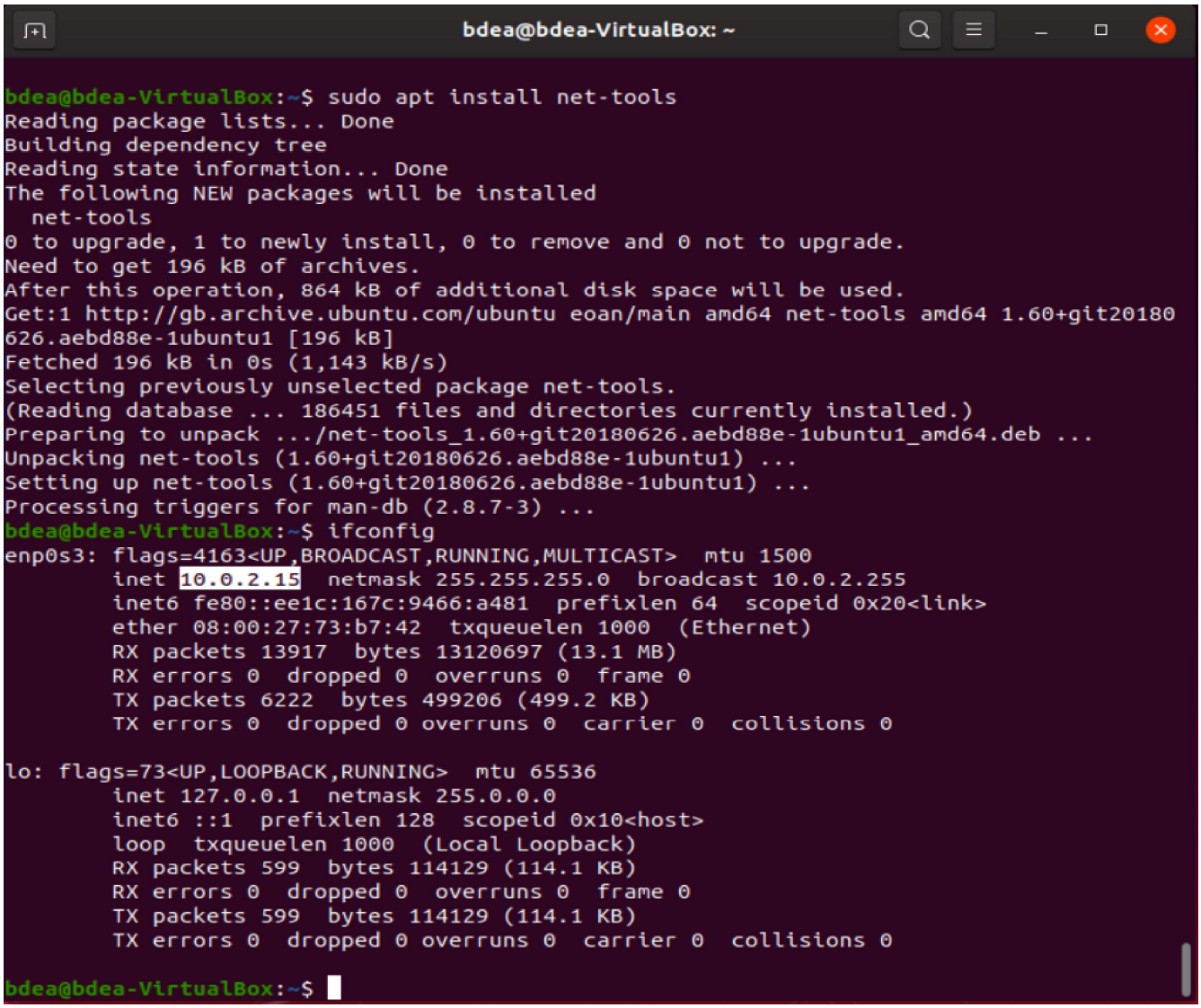

Figure 1.7: net-tools installation and ifconfig output

(c) open the hosts file for editing, to add hadoop-master as a host and map it to its ip address - *sudo gedit hosts*. The hosts file configuration shown in Fig 1.8 shows the correct format of adding a new hosts; in this example the new host is hadoop-master. Please note the highlighted line refers to a host which does not exist any more, please remove this line to avoid any carried forward errors. You may as well comment this line to keep as a reminder for future references.

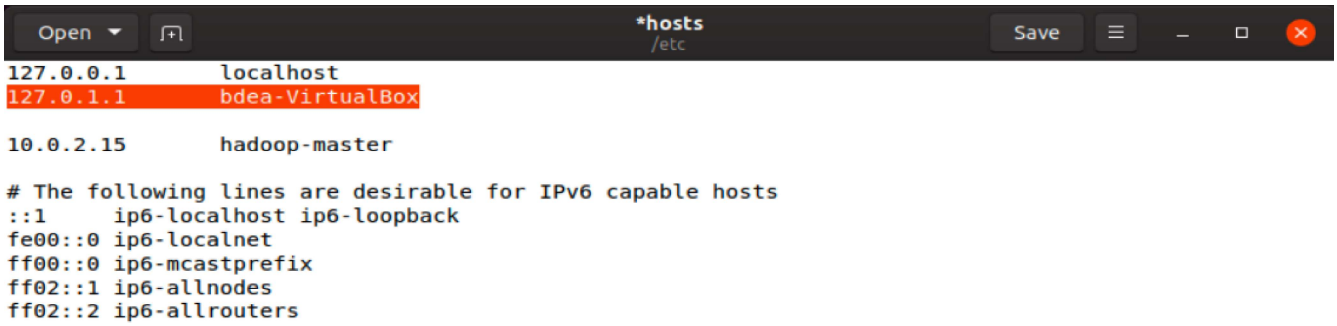

Figure 1.8: interim configuration of hosts file

6. Download, install and configure the latest stable Hadoop-3.2.2 compatible version of Java. Please follow the instructions in next section.

1.4 Java for Hadoop-3.2.2

Hadoop-3.2.2 is not compatible with Java 11 for development purposes. A hdfs cluster could be successfully simulated with Java 11, however, some of the big data services on hadoop cluster can not work with Java 11, such as: mapreduce framework on hadoop.

Therefore, we will work with Java 8, as it the latest stable and compatible version of Java with hadoop-3.2.2. You can follow the next set of instructions for Java 11 as well.

1. Install Java 8 JDK - *sudo apt install openjdk-8-jdk*
   The Java Development Kit (JDK) is a software development environment used for developing Java applications and applets. It includes the Java Runtime Environment (JRE), an interpreter/loader (Java), a compiler (javac), an archiver (jar), a documentation generator (Javadoc) and other tools needed in Java development

2. Configure bash file to set user variable JAVA_HOME. Setting this variable will allow the OS to locate Java directory whenever Java is called by any process/application. Either enter the absolute to path to bash file (it is located i your home directory) or open it from the home directory as follows - *sudo gedit .bashrc*

   The value of JAVA_HOME should be set as - */usr/lib/jvm/java-8-openjdk-amd64/jre*. Please refer to Fig for exact syntax of variable initialisation in bash file.

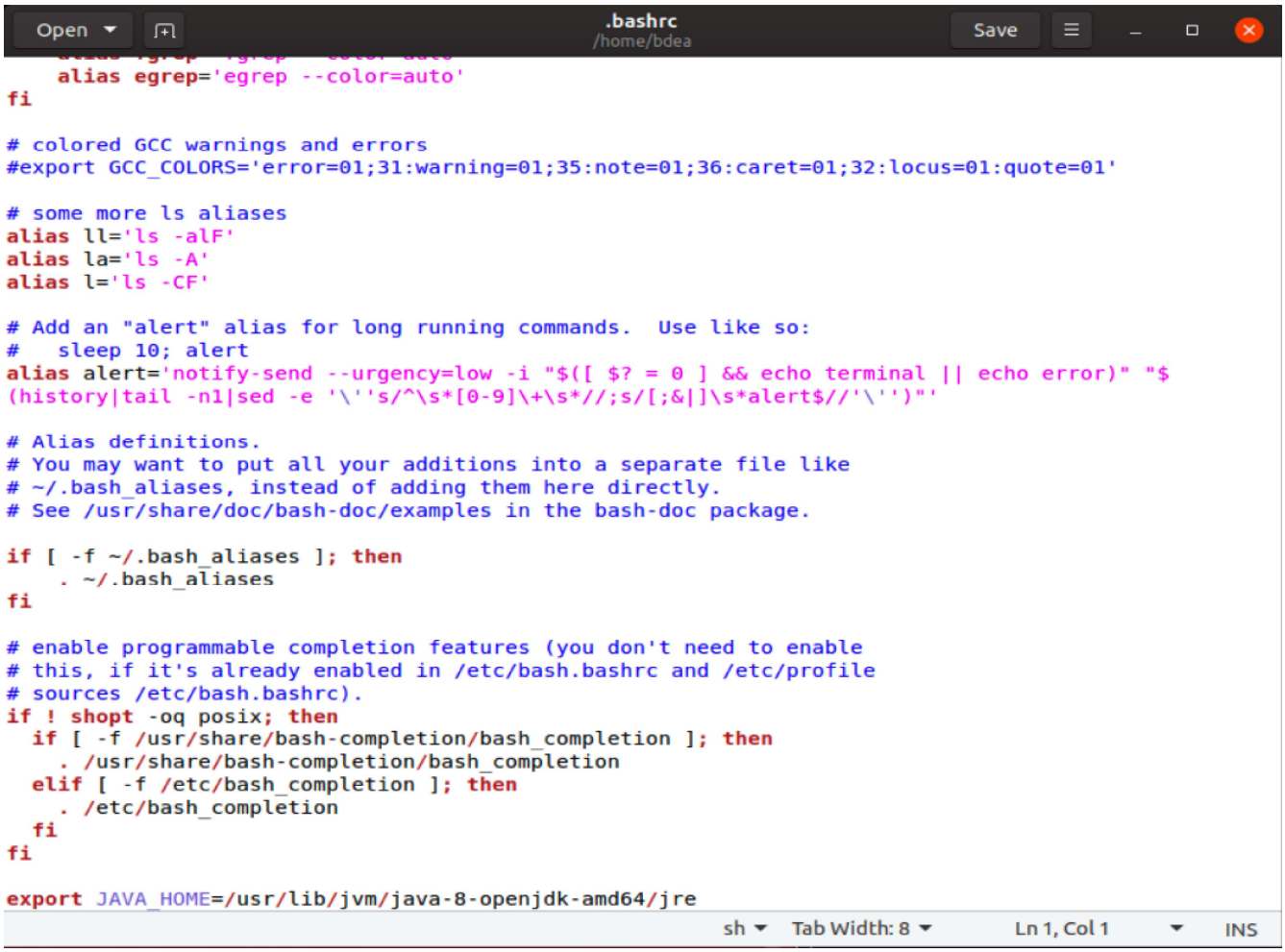

Figure 1.9: Initialising JAVA_HOME in bash file

1.5 Download and configure hadoop

Unlike Java, hadoop is not installed. Hadoop's directories are downloaded and configuration files are modified to ensure smooth execution and run of cluster.

1. Download hadoop-3.2.2 from apache hadoop website - *wgethttps://archive.apache.org/dist/hadoop/common/hadoop-3.2.2/hadoop-3.2.2.tar.gz*
2. unzip and delete the tar file was follows:
   - *tar -xvf hadoop-3.2.2.tar.gz*
   - *rm hadoop-3.2.2.tar.gz*
3. move hadoop to the usr local directory while renaming it to hadoop - *sudo mv hadoop-3.2.2 /usr/local/hadoop*
4. Add hadoop bin and sbin directories to the PATH variable using bash file. Please follow the syntax in Fig 1.10 to concatentae the paths to PATH variable.

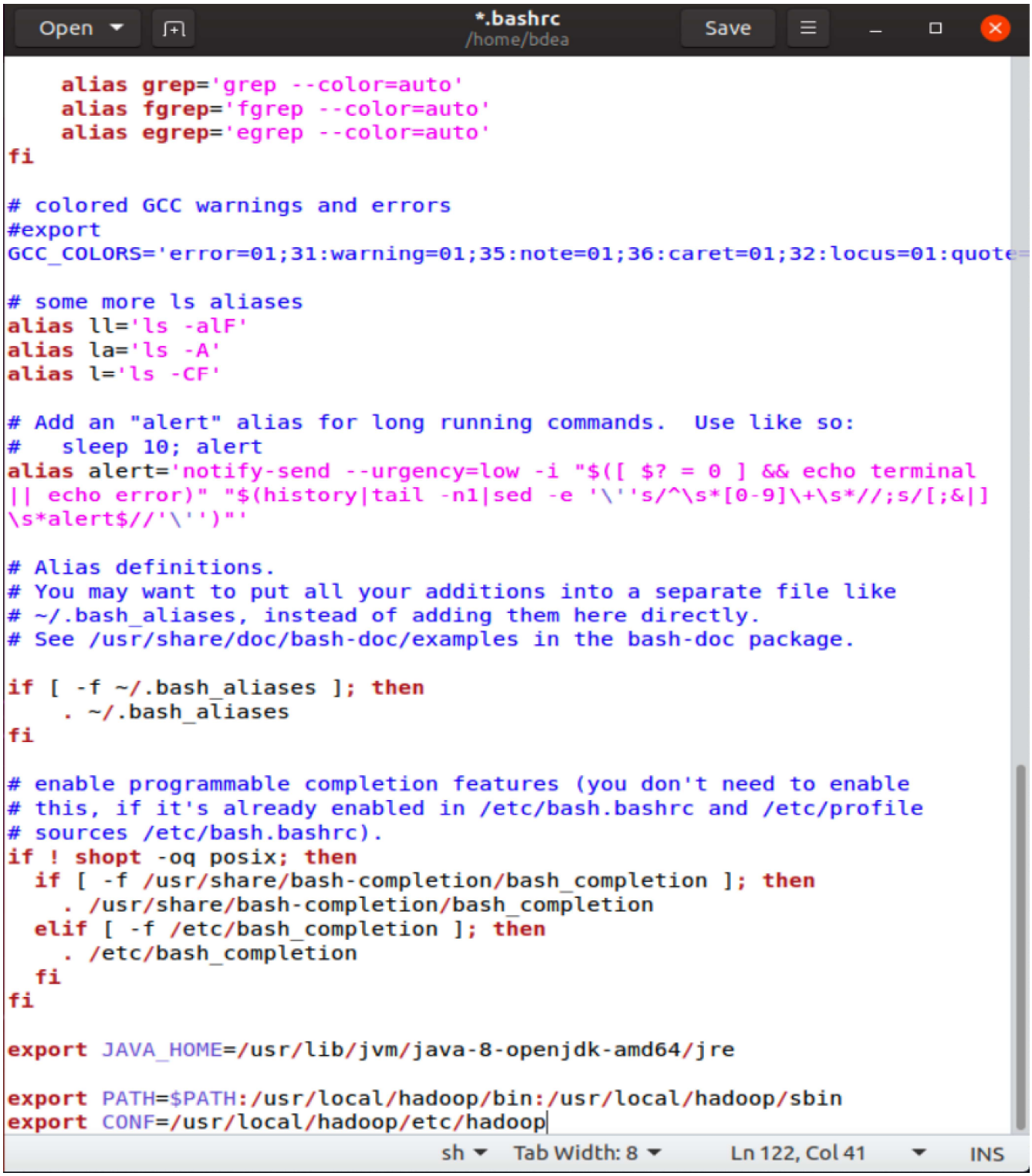

Figure 1.10: Bash file with Hadoop bin and sbin added to path

5. create another environment variable using the bash file, so that accessing the hadoop config directory becomes easier. Note the creation of CONF variable in Fig 1.10

Don't forget to source the bash file whenever you update it.

Before we start editing the hadoop configuration files, it's is essential that we clone the master VM to create a worker VM. Cloning will help save time and effort at this stage.