PA0: Setting up the Environment (0 Points)

In this assignment you will set up the virtual machine (VM) that you will use to run all your programming assignments in this course. This will make it easy to install all dependencies for the programming assignments, saving you the tedium of installing individual packages and ensuring your development environment is correct.

1 Download Assignment

You should have by now downloaded the assignment from Brightspace. The assignment should be inside the directory “`Assignment/PA0/`”.

2 Install Vagrant

Vagrant is a tool for configuring a VM using instructions given in a single “Vagrantfile”.

MacOS and Windows: You will need to install Vagrant using the correct download link for your computer here: https://www.vagrantup.com/downloads.html

Windows only: You will be asked to restart your computer at the end of the installation. Click Yes to do so right away, or restart manually later, but don’t forget to do so or Vagrant will not work!

Linux: First, make sure your package installer is up to date by running the command “`sudo apt-get update`”. To install Vagrant, you must have the “Universe” repository on your computer; run “`sudo apt-add-repository`” universe to add it. Finally, run “`sudo apt-get install vagrant`” to install vagrant.

3 Install VirtualBox

VirtualBox is a VM provider (hypervisor).

MacOS and Windows: You will need to install VirtualBox using the correct download link for your computer here: https://www.virtualbox.org/wiki/Downloads. The links are under the heading “VirtualBox X.X.XX platform packages”.

Windows only: Use all the default installation settings, but you can uncheck the ”Start Oracle VirtualBox X.X.XX after installation” checkbox.

Linux: Run the command “`sudo apt-get install virtualbox`”.

4 Install Git (and SSH-capable Terminal on Windows)

Git is a distributed version control system.

MacOS and Windows: You will need to install Git using the correct download link for your computer here: https://git-scm.com/downloads

MacOS only: Once you have opened the .dmg installation file, you will see a Finder window including a .pkg file, which is the installer. Opening this normally may give you a prompt saying it can’t be opened because it is from an unidentified developer. To override this protection, instead right-click on that .pkg file and select “Open”. This will show a prompt asking you if you are sure you want to open it. Select “Yes”. This will take you to the (straightforward) installation.

Windows only: You will be given many options to choose from during the installation; using all the defaults will be sufficient for this course (you can uncheck "View release notes" at the end). The installation includes a SSH-capable terminal usually located at "`C:\Program Files\Git\bin\bash.exe`". You should use this as your terminal in this class, unless you prefer another SSH-capable terminal (the command prompt will not work). Feel free to create a shortcut to it; copying and pasting the executable somewhere else will not work.

Linux: "`sudo apt-get install git`"

5 Provision Virtual Machine using Vagrant

Open your terminal (use the one mentioned in step 3 if using Windows) and "`cd`" to the directory where you downloaded the assignment directory. Then "`cd Assignment`" to enter the assignments directory. Type "`ls`" to list all the files in directory "`Assignment/`". You should see a file named "Vagrantfile". Run the command "`vagrant up`" to start the VM. You will likely have to wait several minutes. You may see warnings/errors in red, but you should ignore them.

The following commands will allow you to stop the VM at any point (such as when you are done working on an assignment for the day):

"`vagrant suspend`" will save the state of the VM and stop it.

"`vagrant destroy`" will remove all traces of the VM from your system. Use this command only once you have submitted the assignment.

To re-start the VM, simply run the command "`vagrant up`".

Additionally, the command "`vagrant status`" will allow you to check the status of your machine in case you are unsure (e.g. running, powered off, saved...). You must be in some subdirectory of the directory containing the Vagrantfile to use any of the commands above, otherwise Vagrant will not know which VM you are referring to.

Note. The VirtualBox application that was installed in Step 2 provides a visual interface as an alternative to these commands, where you can see the status of your VM and power it on/off or save its state. However, it is **not** recommended to use it since it is not integrated with Vagrant, and typing commands should be no slower. It is also not an alternative to the initial "`vagrant up`" which creates the VM.

6 Test SSH

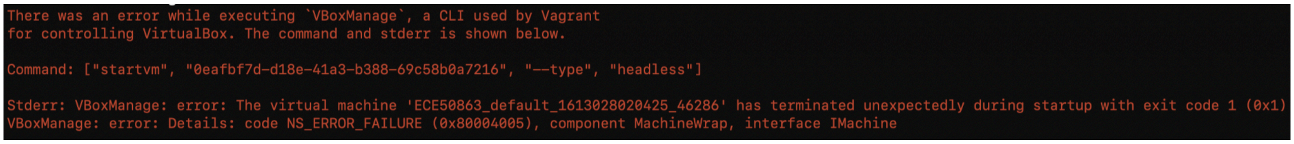
Run the command "`vagrant ssh`" from your terminal. This is the command you will use every time you want to access the VM. If this command works, then your terminal prompt will change to "`vagrant@ubuntu:/vagrant$`". The "`vagrant/`" directory on the VM is shared with "`Assignment/`" directory on your host machine. To verify, type "`ls`" in your VM terminal to see the list of files in the "`vagrant/`" directory. It should be the same as in the "`Assignment/`" directory on your host machine. Vagrant is especially useful because of this shared directory structure. You don't need to copy files to and from the VM. Any file or directory inside the "`Assignment/`" directory is automatically shared between your computer and the VM. This means you can use your IDE or editor of choice on your host machine to write your code (but will still have to build and run the code within the VM).

The command "`logout`" will stop the SSH connection at any point. To re-connect, re-run the command "`vagrant ssh`".

7 Troubleshooting

In this section, we describe how to troubleshoot some of the common issues while running Vagrant.

7.1 Example Issue 1



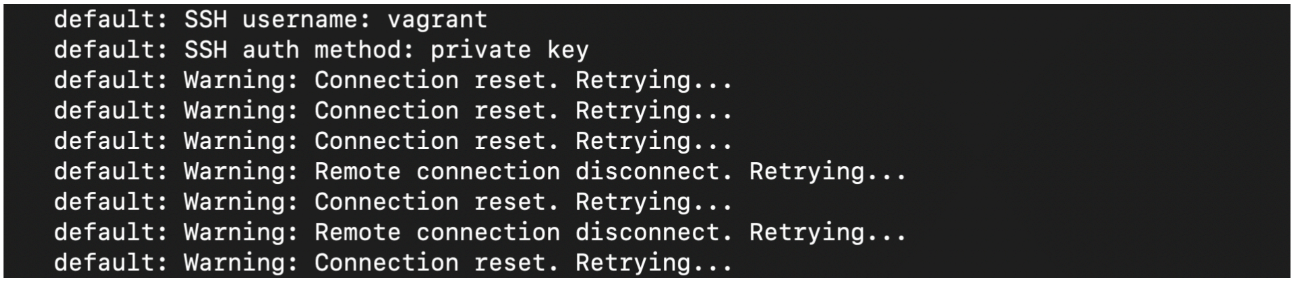
```
There was an error while executing `VBoxManage`, a CLI used by Vagrant
for controlling VirtualBox. The command and stderr is shown below.

Command: ["startvm", "0eafbf7d-d18e-41a3-b388-69c58b0a7216", "--type", "headless"]

Stderr: VBoxManage: error: The virtual machine 'ECE50863_default_1613028020425_46286' has terminated unexpectedly during startup with exit code 1 (0x1)
VBoxManage: error: Details: code NS_ERROR_FAILURE (0x80004005), component MachineWrap, interface IMachine
```

This issue generally shows up on MacOs, and has to do with the permission issues. Go to the following link for troubleshooting: https://apple.stackexchange.com/questions/323960/virtualbox-ns-error-failure-0x80004005-mac

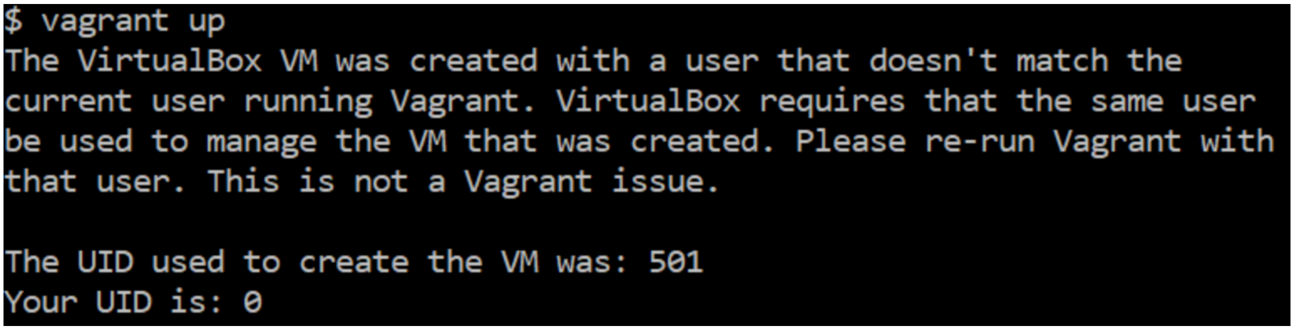
7.2 Example Issue 2



```
    default: SSH username: vagrant
    default: SSH auth method: private key
    default: Warning: Connection reset. Retrying...
    default: Warning: Connection reset. Retrying...
    default: Warning: Connection reset. Retrying...
    default: Warning: Remote connection disconnect. Retrying...
    default: Warning: Connection reset. Retrying...
    default: Warning: Remote connection disconnect. Retrying...
    default: Warning: Connection reset. Retrying...
```

This issue generally shows up when using the command "`vagrant halt`". To troubleshoot, simply destroy the VM by running "`vagrant destroy`", followed by "`vagrant up`". Do not use "`vagrant halt`" but instead use "`vagrant suspend`" to suspend the VM.

7.3 Example Issue 3



```
$ vagrant up
The VirtualBox VM was created with a user that doesn't match the
current user running Vagrant. VirtualBox requires that the same user
be used to manage the VM that was created. Please re-run Vagrant with
that user. This is not a Vagrant issue.

The UID used to create the VM was: 501
Your UID is: 0
```

Try deleting the "`.vagrant/`" directory inside the "`Assignment/`" directory, and re-run "`vagrant up`". Or open the file "`.vagrant/machines/default/virtualbox/creator_uid`" and change the file content from 501 to 0, and re-run "`vagrant up`".

8 Extra Note for Windows Users

Line endings are symbolized differently in DOS (Windows) and Unix (Linux/MacOS). In the former, they are represented by a carriage return and line feed (CRLF, or `\r\n`), and in the latter, just a line feed (LF, or `\n`). This can lead to parsing problems within the VM, which runs Ubuntu (Unix). Fortunately, this only seems to affect the shell scripts (*.sh files). If you want to write/edit shell scripts to help yourself with testing, or if you encounter this problem with some other type of file, use the preinstalled program "dos2unix". Run "`dos2unix [file]`" to convert it to Unix format (before editing/running in VM), and run "`unix2dos [file]`" to convert it to DOS format (before editing on Windows). A good hint that you need to do this when running from the VM is some error message involving $\hat{}M$ (carriage return). A good hint you need to do this when editing on Windows is the lack of new lines. Remember, doing this should only be necessary if you want to write/edit edit shell scripts.

9 Running the Assignment

On your host machine, modify the "`hello.py`" file inside "`Assignment/PA0/`" directory by replacing "Vishal" with your own name. To run the code, go to the VM terminal and "`cd PA0`" and run the command "`python hello.py`". If the command succeeds, you should see the output on the terminal. If so, you are done with this assignment!

10 Submission

No submission is required for this assignment.