

UECS2363 SOFTWARE CONSTRUCTION AND CONFIGURATION

Practical 7 : Virtual Box and Vagrant

Vagrant is a open-source command line utility for managing lifecycle of virtual machines, e.g. for VirtualBox, Hyper-V, Docker, VMware, and AWS. Usage of Vagrant is crucial for continuous development of a software project and make sure collaborators get to have the same environment for executing and testing the software developed.

For instance, developer A developed an application in C++ programming language and the application runs on a Linux workstation and interacts with a database. Developer A tested the application with dummy data that was inserted into the database that the application is supposed to interact with. Then developer A wanted to pass the application to developer B, C and D who are the collaborators of the application development project. However, each of them is with workstations of different operating systems. In order to ensure all the others will have the access to same environment to test the same C++ application and database, the application need to be put into a VirtualBox (virtual environment) and pass to the collaborators. Executing the application on a virtual machine, will enable all the collaborators execute and test the application, as if they are having a Linux OS on their workstation.

In order to achieve what developer A tried to do, one has to have:

- 1) VirtualBox (<https://www.virtualbox.org/wiki/Downloads>)
- 2) Vagrant (<https://www.vagrantup.com/docs/installation>)
- 3) SSH client (<https://cmder.net/>)

Preliminaries.

Firstly, let's download and install VirtualBox by Oracle from the link below:

<https://www.virtualbox.org/wiki/Downloads>

Click "Windows hosts" on the webpage to download and install as shown in Figure 1.



Figure 1: Select "Windows hosts", download installation and follow default installation instructions.

Successful installation will allow the workstation to work with VirtualBox as shown in Figure 2.

UECS2363 SOFTWARE CONSTRUCTION AND CONFIGURATION



Figure 2: VirtualBox by Oracle.

Then, download and install Vagrant from the following link:

<https://www.vagrantup.com/downloads>

Click “Download” button on the webpage and install Vagrant with default installation as shown in Figure 3.

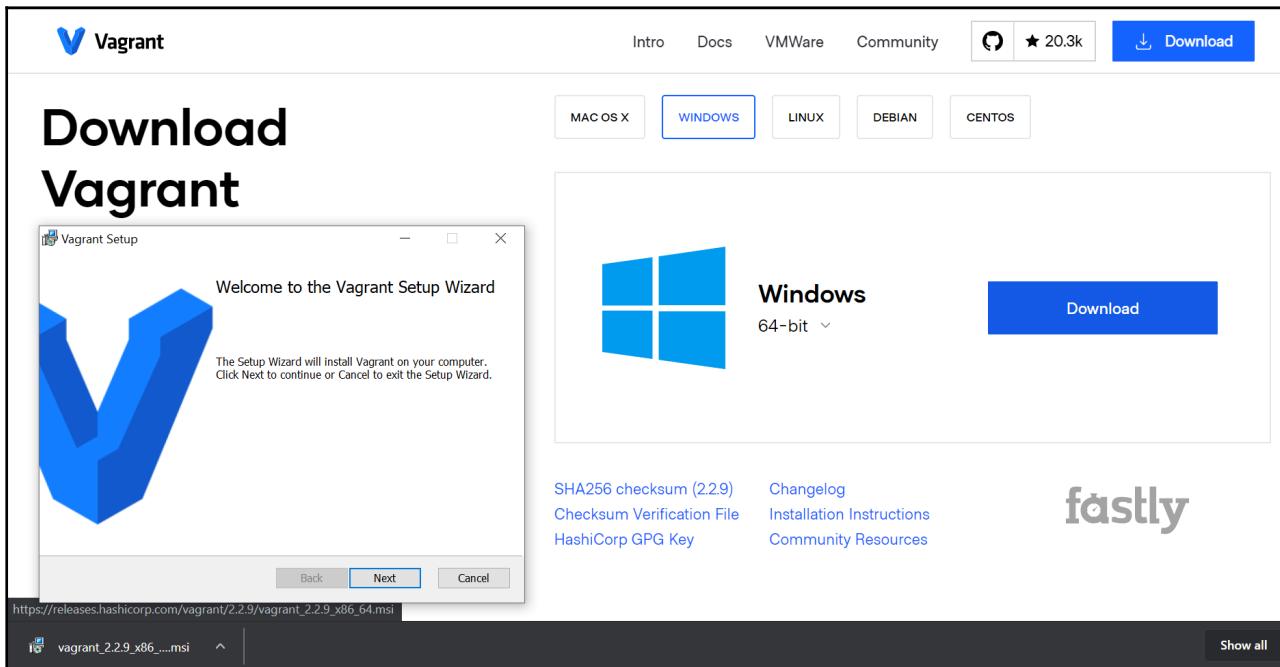


Figure 3: Vagrant download and installation.

While download and installation is happening, download the SSH client; Cmder, from the following link:

<https://cmder.net/>

On the webpage, scroll to “Download” section and click “Download Full” as shown in Figure 4.

UECS2363 SOFTWARE CONSTRUCTION AND CONFIGURATION

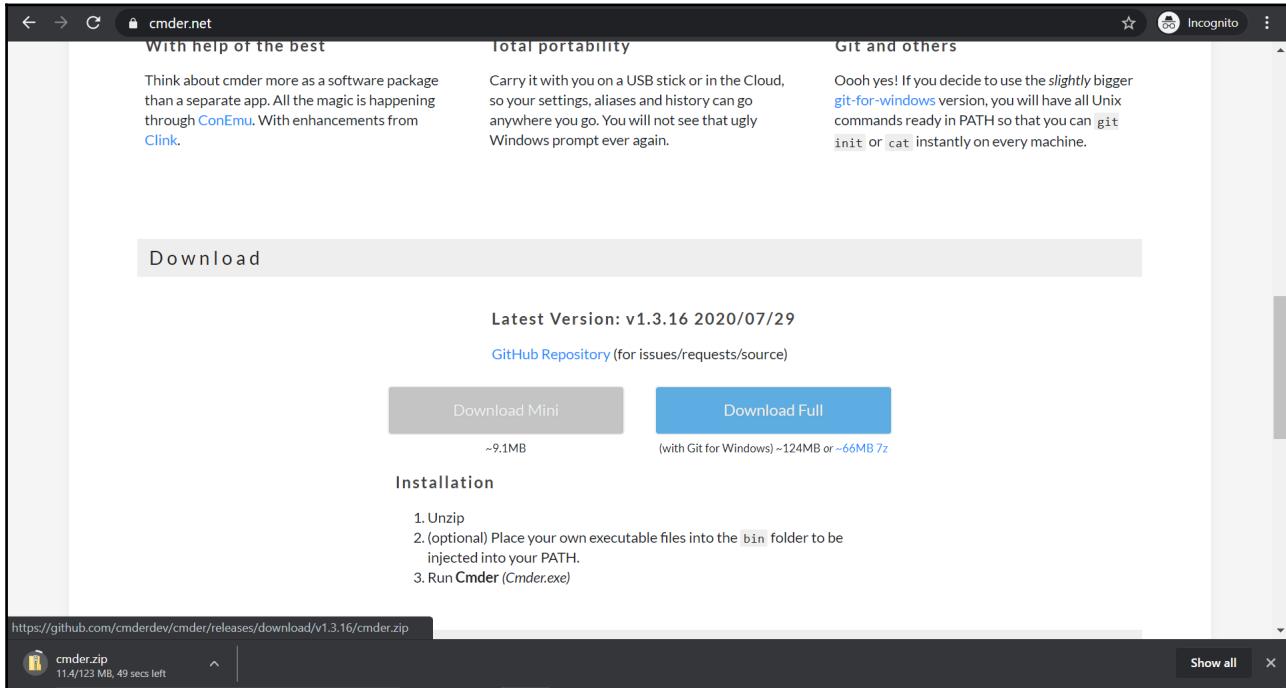


Figure 4: Download Cmder SSH client.

Cmder download is a zip file. Thus, the tool will be available once extracted from the zip. For easy access, extract in a filepath that is conveniently reachable or easily remembered.

Installation of Vagrant will only be completed after restarting the workstation. Make sure to save changes of everything done before restarting the workstation.

Getting Started.

Launch cmder.exe (extracted from Cmder.zip) and check the version of Vagrant as shown in Figure 5. ***Note: if there is a warning pop-up, just click “Run Anyway” and “Ignore and Continue”.*

A screenshot of a terminal window titled 'Cmder'. The window shows command-line output. It starts with a message: 'Running Git for Windows one time Post Install....' followed by a series of file paths being copied. Then it creates a user startup file: 'Creating user startup file: "C:\Program Files\cmder\config\user_profile.cmd"' and shows '1 file(s) copied.'. Below this, the user types 'C:\Program Files\cmder' and then runs the command 'λ vagrant -v'. The output shows 'Vagrant 2.2.9'. Finally, the user types 'C:\Program Files\cmder' again and ends with a single character 'λ'. The window has a standard Windows taskbar at the bottom.

Figure 5: Verification of successful installation of Vagrant in workstation.

Using command of `vagrant -v`, detail of Vagrant's version that had been installed in the workstation will be in the output.

UECS2363 SOFTWARE CONSTRUCTION AND CONFIGURATION

In order to use Vagrant, a Vagrant Box need to be created. Instructions to create a virtual box is in the following link:

<https://app.vagrantup.com/boxes/search>

From the list of Vagrant Box options, select Ubuntu/trusty64 Vagrant Box and the website will direct to the instructions of configuring the selected Vagrant Box as shown in Figure 6.

The screenshot shows a web browser window for the Vagrant Cloud website (app.vagrantup.com/boxes/trusty64). The page title is "ubuntu / trusty64 Vagrant box". It includes a "How to use this box with Vagrant" section with a "Vagrantfile" button and a "New" button, followed by a code snippet: "vagrant init ubuntu/trusty64" and "vagrant up". Below this is a "v20190514.0.0 currently released version" section, which notes that the version was created 9 months ago and has no description. The page header features the HashiCorp logo and navigation links for Search, Pricing, Vagrant, Help, Create an Account, and Sign In.

Figure 6: Ubuntu/trusty64 Vagrant Box Configuration Instructions.

“Vagrantfile” menu contains scripts for configuration of a Vagrant Box, while “New” menu contains scripts for initialization of a Vagrant Box.

Choose a folder where you would like the Virtual Box to be initialized and cd to the filepath in Cmder. Then, execute `vagrant box add ubuntu/trusty64` command in order to add a virtual box into the specified folder, as shown in Figure 7.

***Note: adding of virtual box will take some time for the first time.*

The screenshot shows a Cmder terminal window with the following command history:
C:\Program Files\cmder
λ cd C:\Users\looyi\VirtualBox

C:\Users\looyi\VirtualBox
λ vagrant box add ubuntu/trusty64
==> box: Loading metadata for box 'ubuntu/trusty64'
 box: URL: https://vagrantcloud.com/ubuntu/trusty64
==> box: Adding box 'ubuntu/trusty64' (v20190514.0.0) for provider: virtualbox
 box: Downloading: https://vagrantcloud.com/ubuntu/boxes/trusty64/versions/201905
14.0.0/providers/virtualbox.box
Download redirected to host: cloud-images.ubuntu.com
 box:
==> box: Successfully added box 'ubuntu/trusty64' (v20190514.0.0) for 'virtualbox'!

C:\Users\looyi\VirtualBox
λ |

Figure 7: Add ubuntu/trusty64 Vagrant Box into specified folder.

Then, execute the initialization command provided in the “New” menu on the website as shown in Figure 8.

UECS2363 SOFTWARE CONSTRUCTION AND CONFIGURATION

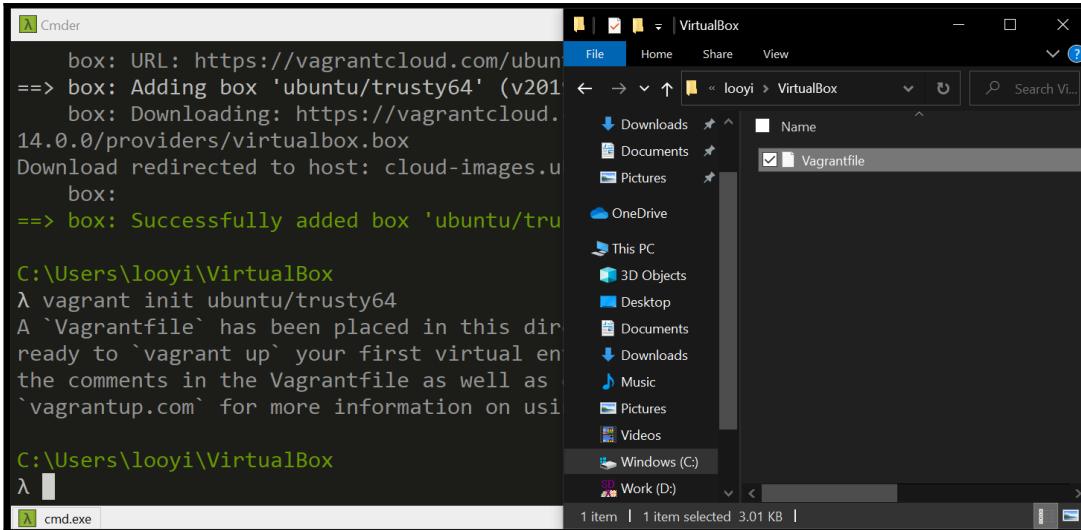


Figure 8: Generation of VagrantFile through Vagrant Box initialization.

Access the VagrantFile using Visual Studio Code and make sure to make changes to the scripts as illustrated in Figure 9. ***Note: Download extensions for Vagrant files in order to work properly with Vagrant files.*

```
1 # -*- mode: ruby -*-
2 # vi: set ft=ruby :
3
4 Vagrant.configure("2") do |config|
5   # Box Settings
6   config.vm.box = "ubuntu/trusty64"
7
8   # Provider Settings
9   config.vm.provider "virtualbox" do |vb|
10
11 end
12
```

Figure 9: Preliminary configurations to enable VM start by `vagrant up` command.

Next, execute `vagrant up` command as enlisted on the website “New” menu, in order to start the virtual box. A display of connections established means that virtual box successfully up running. This also imply that we have successfully entered the virtual box / environment that had been created before this. One may double confirm successful connection of virtual box through launching VirtualBox by Oracle. VirtualBox shows a connection to a virtual environment as illustrated in Figure 10.

UECS2363 SOFTWARE CONSTRUCTION AND CONFIGURATION

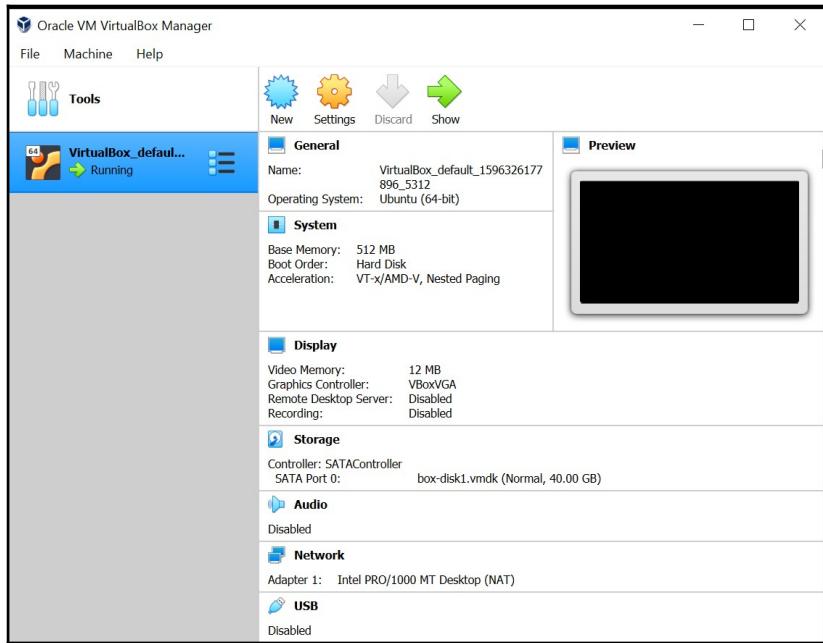


Figure 10: Connection to virtual machine shown in VirtualBox.

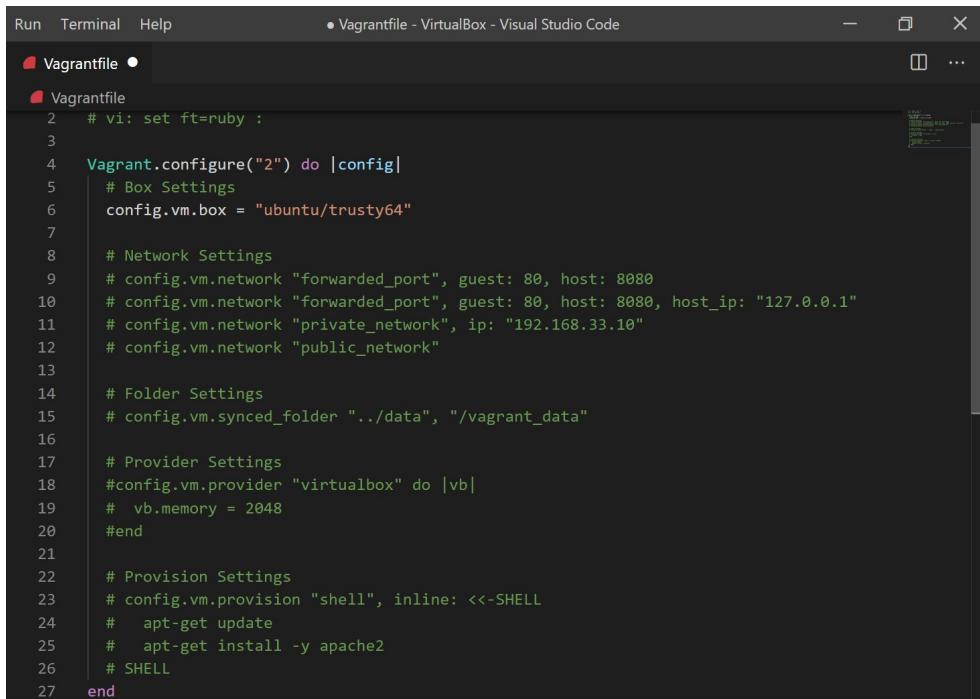
In this environment, one may import the application/software to be executed and tested. If we would like to transfer the virtual environment to other developers, just pass the VagrantFile to the other developers.

Five (5) Core Fundamentals to explore:

- 1) config.vm.box – Virtual box options
- 2) config.vm.provider – Virtual box performance settings
- 3) config.vm.network – How the host is able to see the virtual box/application
- 4) config.vm.synced_folder – Access of project development files
- 5) config.vm.provision – what we want to setup (the platforms to execute and test the software/application)

Access the VagrantFile using Visual Studio Code and find the 5 core areas as illustrated in Figure 11. ***Note: Download extensions for Vagrant files in order to work properly with Vagrant files.*

UECS2363 SOFTWARE CONSTRUCTION AND CONFIGURATION



A screenshot of the Visual Studio Code interface. The title bar says "Run Terminal Help" and "• Vagrantfile - VirtualBox - Visual Studio Code". The main editor area shows a Vagrantfile with code starting from line 2:

```
2 # vi: set ft=ruby :  
3  
4 Vagrant.configure("2") do |config|  
5   # Box Settings  
6   config.vm.box = "ubuntu/trusty64"  
7  
8   # Network Settings  
9   # config.vm.network "forwarded_port", guest: 80, host: 8080  
10  # config.vm.network "forwarded_port", guest: 80, host: 8080, host_ip: "127.0.0.1"  
11  # config.vm.network "private_network", ip: "192.168.33.10"  
12  # config.vm.network "public_network"  
13  
14   # Folder Settings  
15   # config.vm.synced_folder "../data", "/vagrant_data"  
16  
17   # Provider Settings  
18   #config.vm.provider "virtualbox" do |vb|  
19   #   vb.memory = 2048  
20   #end  
21  
22   # Provision Settings  
23   # config.vm.provision "shell", inline: <<-SHELL  
24   #   apt-get update  
25   #   apt-get install -y apache2  
26   # SHELL  
27  
28 end
```

Figure 11: 5 Core areas to explore in Vagrant.

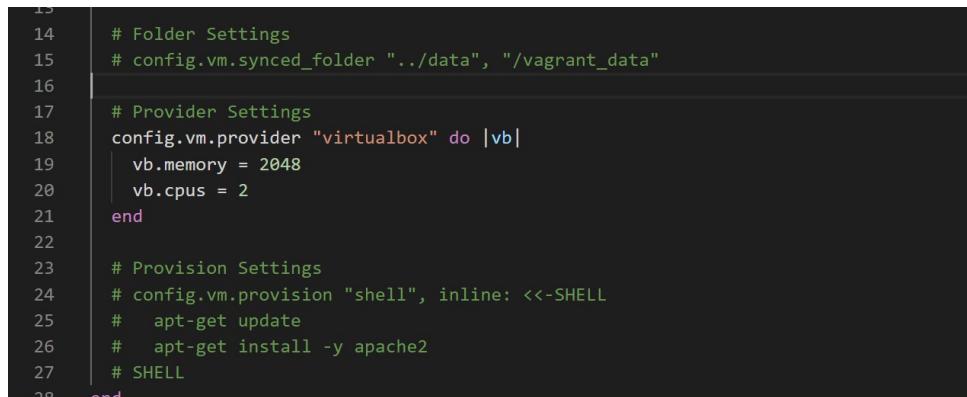
Some basics of Vagrant commands before proceeding:

- 1) `vagrant destroy` – Delete virtual box
- 2) `vagrant suspend` – Shut down / suspend connection to virtual box
- 3) `vagrant resume` – Restore suspended virtual box
- 4) `vagrant reload` – Reload the virtual box

Configuring Virtual Box Provider Settings

It is crucial that the providing environment is powerful enough to execute and test an application that is fed in. Looking at the provider using Oracle Virtual Box manager give an overall detail of memory and CPU that are allocated for the virtual box.

In order to increase the provision, changes need to be made in the #Provider Settings with `vb.memory` and `vb.cpus` settings as illustrated in Figure 12.



A screenshot of the Visual Studio Code interface showing a modified Vagrantfile. The code includes the following changes in lines 18-21:

```
15  
16  
17  
18   # Provider Settings  
19   config.vm.provider "virtualbox" do |vb|  
20     vb.memory = 2048  
21     vb.cpus = 2  
22  
23  
24  
25  
26  
27  
28 end
```

Figure 12: Change Memory and CPU allocation for Vagrant Box.

UECS2363 SOFTWARE CONSTRUCTION AND CONFIGURATION

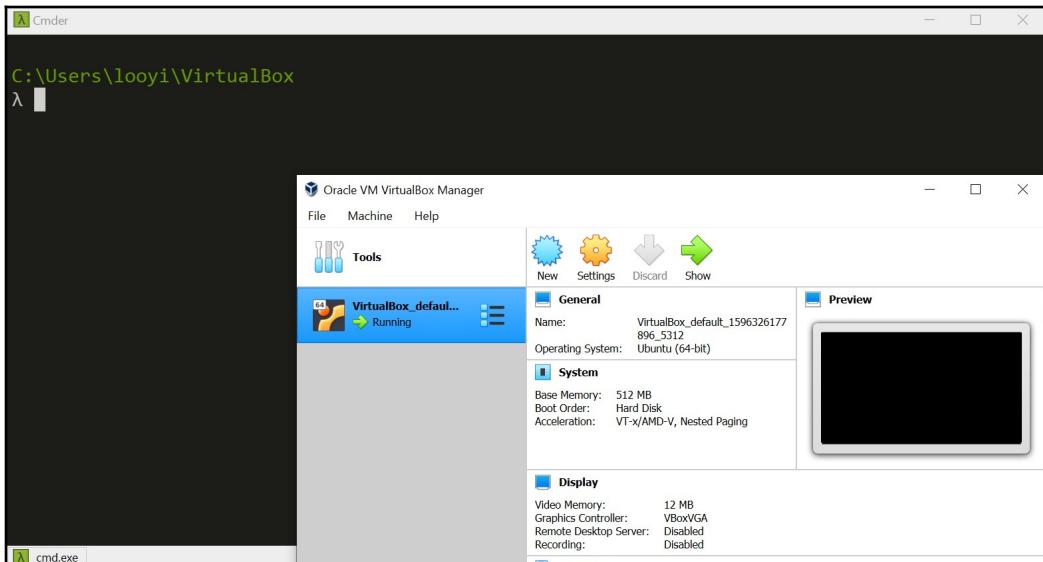


Figure 13: Before change of provision settings.

Execute `vagrant reload` in order to run the virtual box with new provision settings that had been saved in Visual Code Studio as shown on .

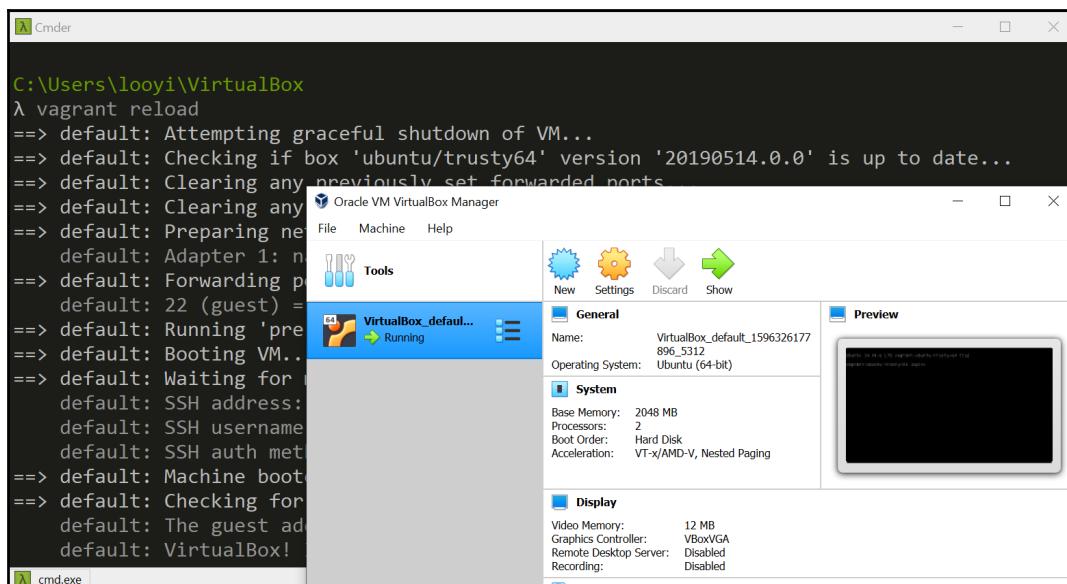


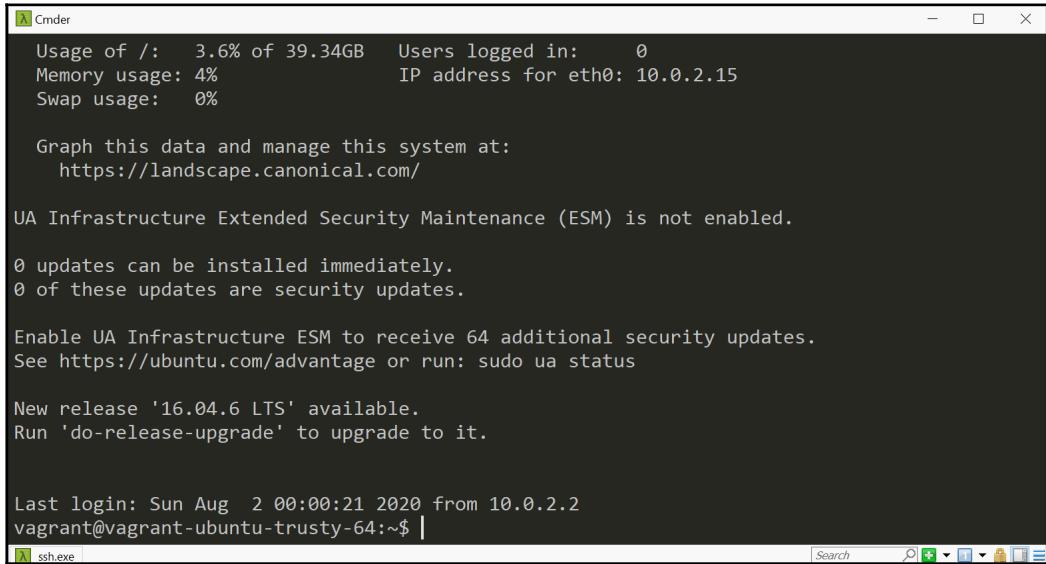
Figure 14: After reloading vagrant box with changes of provision settings applied.

The performance of the vagrant box may be changed using the aforementioned method.

UECS2363 SOFTWARE CONSTRUCTION AND CONFIGURATION

Configuring Virtual Box Network Settings.

Now, let's go into the virtual box itself by executing `vagrant ssh` as illustrated in Figure 15.



```
Usage of /: 3.6% of 39.34GB Users logged in: 0
Memory usage: 4% IP address for eth0: 10.0.2.15
Swap usage: 0%

Graph this data and manage this system at:
https://landscape.canonical.com/

UA Infrastructure Extended Security Maintenance (ESM) is not enabled.

0 updates can be installed immediately.
0 of these updates are security updates.

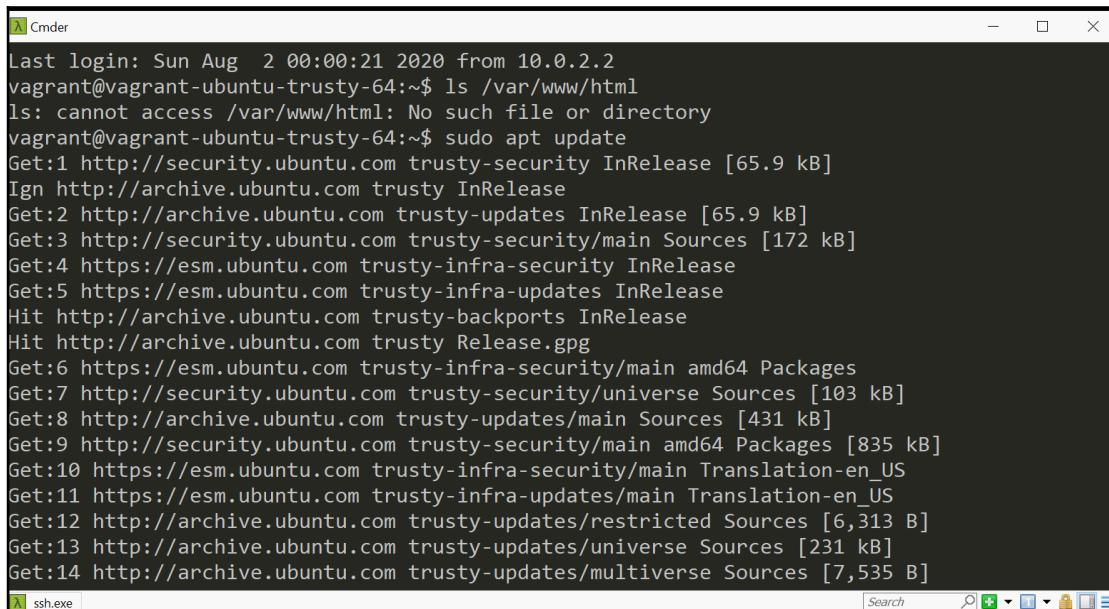
Enable UA Infrastructure ESM to receive 64 additional security updates.
See https://ubuntu.com/advantage or run: sudo ua status

New release '16.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Sun Aug 2 00:00:21 2020 from 10.0.2.2
vagrant@vagrant-ubuntu-trusty-64:~$ |
```

Figure 15: Access into virtual box.

`vagrant@vagrant-ubuntu-trusty-64:~$` means that we're inside the virtual box. However, this is just an empty ubuntu environment. Thus, let's install 2 software into the virtual box by executing the `sudo apt` command. Before using `sudo apt` to install software, do a quick update on `sudo apt` by executing `sudo apt update`.

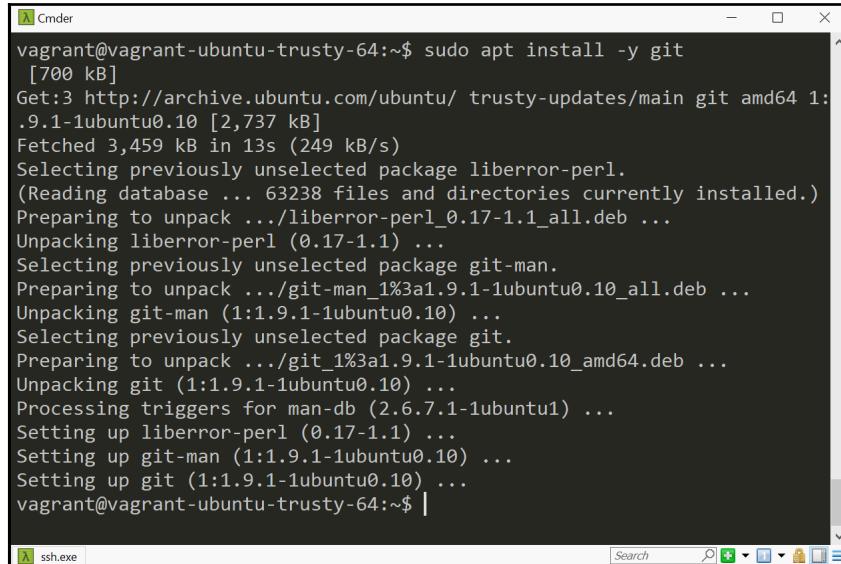


```
Last login: Sun Aug 2 00:00:21 2020 from 10.0.2.2
vagrant@vagrant-ubuntu-trusty-64:~$ ls /var/www/html
ls: cannot access /var/www/html: No such file or directory
vagrant@vagrant-ubuntu-trusty-64:~$ sudo apt update
Get:1 http://security.ubuntu.com trusty-security InRelease [65.9 kB]
Ign http://archive.ubuntu.com trusty InRelease
Get:2 http://archive.ubuntu.com trusty-updates InRelease [65.9 kB]
Get:3 http://security.ubuntu.com trusty-security/main Sources [172 kB]
Get:4 https://esm.ubuntu.com trusty-infra-security InRelease
Get:5 https://esm.ubuntu.com trusty-infra-updates InRelease
Hit http://archive.ubuntu.com trusty-backports InRelease
Hit http://archive.ubuntu.com trusty Release.gpg
Get:6 https://esm.ubuntu.com trusty-infra-security/main amd64 Packages
Get:7 http://security.ubuntu.com trusty-security/universe Sources [103 kB]
Get:8 http://archive.ubuntu.com trusty-updates/main Sources [431 kB]
Get:9 http://security.ubuntu.com trusty-security/main amd64 Packages [835 kB]
Get:10 https://esm.ubuntu.com trusty-infra-security/main Translation-en_US
Get:11 https://esm.ubuntu.com trusty-infra-updates/main Translation-en_US
Get:12 http://archive.ubuntu.com trusty-updates/restricted Sources [6,313 B]
Get:13 http://archive.ubuntu.com trusty-updates/universe Sources [231 kB]
Get:14 http://archive.ubuntu.com trusty-updates/multiverse Sources [7,535 B]
```

Figure 16: `sudo apt update`.

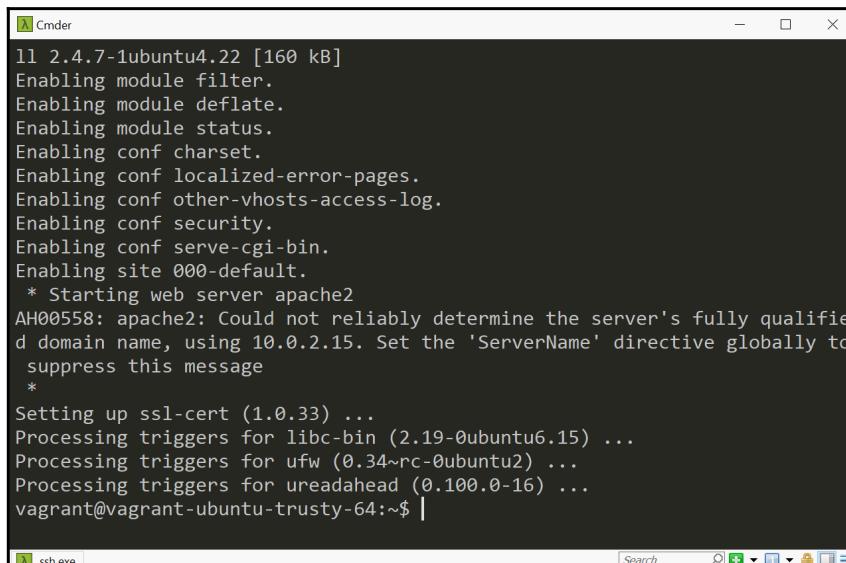
Then, install Git into the virtual box by executing `sudo apt install -y git`. Install Apache server into the virtual box by executing `sudo apt install -y apache2` as shown in Figure 17 and 18.

UECS2363 SOFTWARE CONSTRUCTION AND CONFIGURATION



```
vagrant@vagrant-ubuntu-trusty-64:~$ sudo apt install -y git
[700 kB]
Get:3 http://archive.ubuntu.com/ubuntu/ trusty-updates/main git amd64 1:9.1-1ubuntu0.10 [2,737 kB]
Fetched 3,459 kB in 13s (249 kB/s)
Selecting previously unselected package liberror-perl.
(Reading database ... 63238 files and directories currently installed.)
Preparing to unpack .../liberror-perl_0.17-1.1_all.deb ...
Unpacking liberror-perl (0.17-1.1) ...
Selecting previously unselected package git-man.
Preparing to unpack .../git-man_1%3a1.9.1-1ubuntu0.10_all.deb ...
Unpacking git-man (1:1.9.1-1ubuntu0.10) ...
Selecting previously unselected package git.
Preparing to unpack .../git_1%3a1.9.1-1ubuntu0.10_amd64.deb ...
Unpacking git (1:1.9.1-1ubuntu0.10) ...
Processing triggers for man-db (2.6.7.1-1ubuntu1) ...
Setting up liberror-perl (0.17-1.1) ...
Setting up git-man (1:1.9.1-1ubuntu0.10) ...
Setting up git (1:1.9.1-1ubuntu0.10) ...
vagrant@vagrant-ubuntu-trusty-64:~$ |
```

Figure 17: Git installation by executing `sudo apt install -y git`.



```
11 2.4.7-1ubuntu4.22 [160 kB]
Enabling module filter.
Enabling module deflate.
Enabling module status.
Enabling conf charset.
Enabling conf localized-error-pages.
Enabling conf other-vhosts-access-log.
Enabling conf security.
Enabling conf serve-cgi-bin.
Enabling site 000-default.
 * Starting web server apache2
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 10.0.2.15. Set the 'ServerName' directive globally to suppress this message
 *
Setting up ssl-cert (1.0.33) ...
Processing triggers for libc-bin (2.19-0ubuntu6.15) ...
Processing triggers for ufw (0.34~rc-0ubuntu2) ...
Processing triggers for ureadahead (0.100.0-16) ...
vagrant@vagrant-ubuntu-trusty-64:~$ |
```

Figure 18: Apache server installation by executing `sudo apt install -y apache2`.

Now the virtual box has Git and an Apache server in it. Accessing filepath `/var/www/html` using `ls` command as shown in Figure 19, gives an output of `index.html` file.

UECS2363 SOFTWARE CONSTRUCTION AND CONFIGURATION

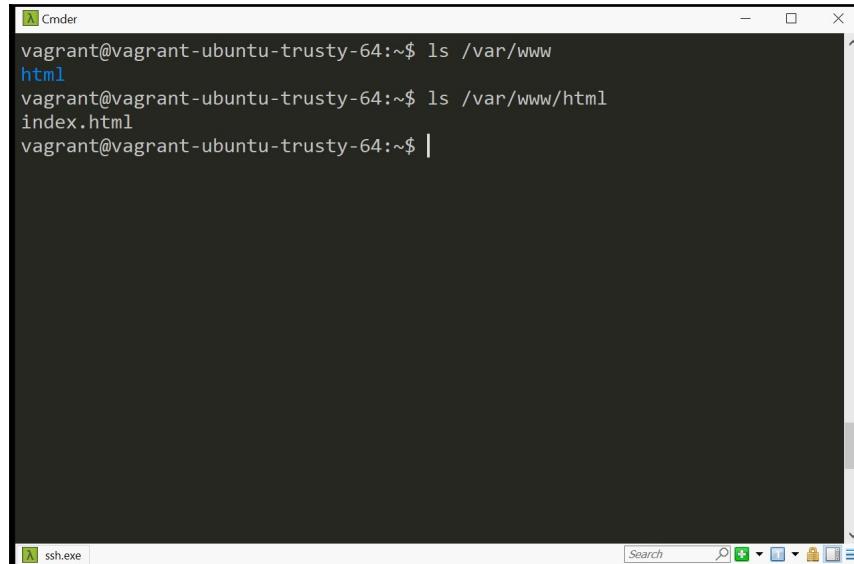


Figure 19: An Apache server setup in vagrant box.

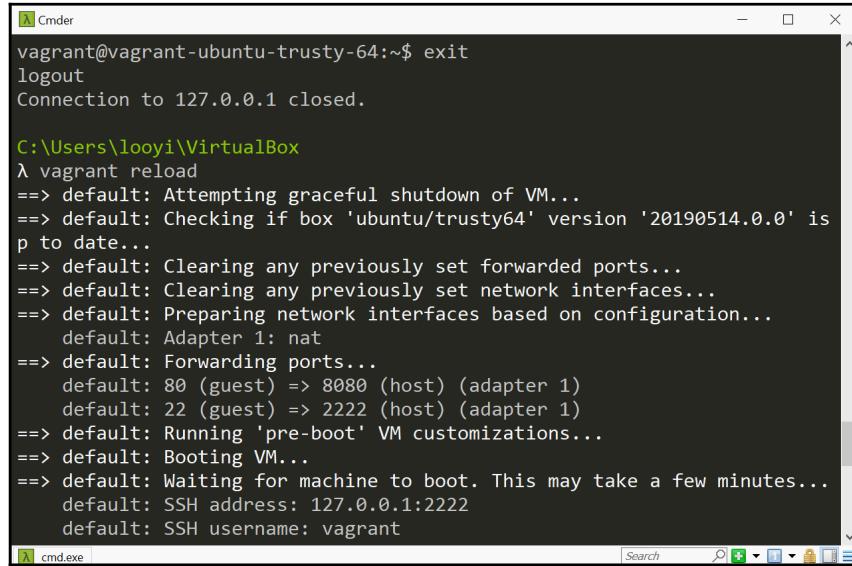
We cannot go to the browser and access index.html. Thus, we need to look back into the VagrantFile #Network Settings. Uncomment the first configuration which says the guest (virtual box) is 80 and host (local workstation) is 8080 as illustrated in Figure 20.

```
4 Vagrant.configure("2") do |config|
5   # Box Settings
6   config.vm.box = "ubuntu/trusty64"
7
8   # Network Settings
9   config.vm.network "forwarded_port", guest: 80, host: 8080
10  # config.vm.network "forwarded_port", guest: 80, host: 8080, host_ip: "127.0.0.1"
11  # config.vm.network "private_network", ip: "192.168.33.10"
12  # config.vm.network "public_network"
13
14  # Folder Settings
15  # config.vm.synced_folder "../data", "/vagrant_data"
16
17  # Provider Settings
18  config.vm.provider "virtualbox" do |vb|
19    vb.memory = 2048
20    vb.cpus = 2
21  end
22
```

Figure 20: Configure virtual box to be on localhost.

Save the changes in the VagrantFile in VS Code, then exit the `vagrant ssh` by executing `exit` command. Then execute `vagrant reload` again as shown in Figure 21.

UECS2363 SOFTWARE CONSTRUCTION AND CONFIGURATION



```
vagrant@vagrant-ubuntu-trusty-64:~$ exit
logout
Connection to 127.0.0.1 closed.

C:\Users\looyi\VirtualBox
λ vagrant reload
==> default: Attempting graceful shutdown of VM...
==> default: Checking if box 'ubuntu/trusty64' version '20190514.0.0' is
p to date...
==> default: Clearing any previously set forwarded ports...
==> default: Clearing any previously set network interfaces...
==> default: Preparing network interfaces based on configuration...
    default: Adapter 1: nat
==> default: Forwarding ports...
    default: 80 (guest) => 8080 (host) (adapter 1)
    default: 22 (guest) => 2222 (host) (adapter 1)
==> default: Running 'pre-boot' VM customizations...
==> default: Booting VM...
==> default: Waiting for machine to boot. This may take a few minutes...
    default: SSH address: 127.0.0.1:2222
    default: SSH username: vagrant
```

Figure 21: Reload virtual box with configured network.

Now, access the browser using localhost with port 8080 and see that the Apache server index.html is able to load as illustrated in Figure 22.

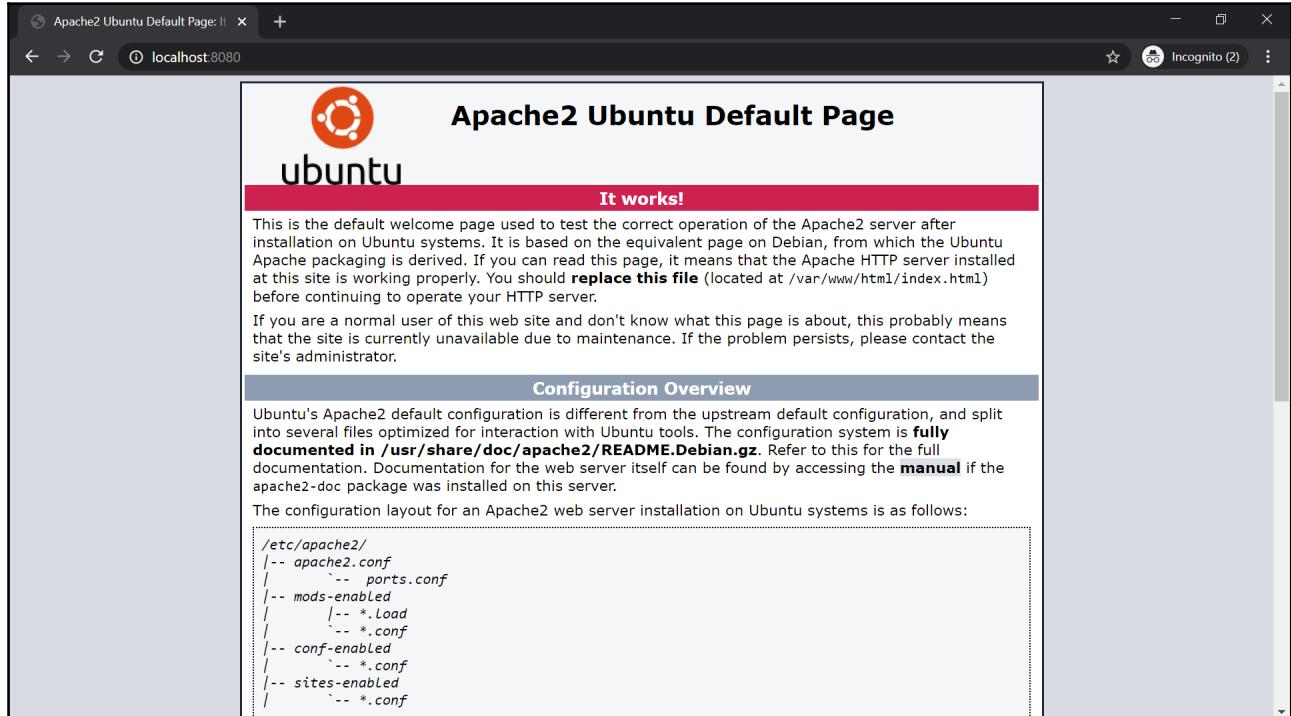


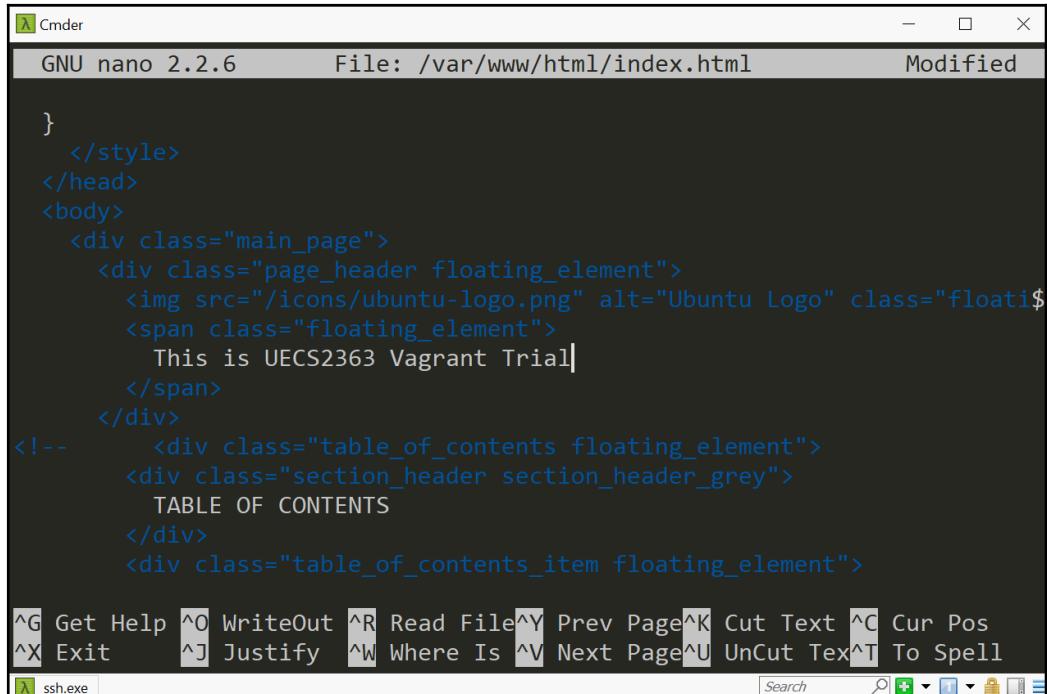
Figure 22: Running virtual box on localhost.

Q: There are a few network settings available. What if “private network” is enabled? How should one access the index.html then?

UECS2363 SOFTWARE CONSTRUCTION AND CONFIGURATION

Configuring Virtual Box Folder Settings.

If we would like to access the index.html file and change the scriptings directly from the Cmder, executing `sudo nano /var/www/html/index.html` command will bring straight into editing of the file in nano editor as illustrated in Figure 23.



The screenshot shows a terminal window titled 'Cmder' with the command 'GNU nano 2.2.6' at the top. The file path 'File: /var/www/html/index.html' is also displayed. The nano editor is displaying the content of the index.html file, which contains HTML code. A cursor is visible in the middle of the page content. At the bottom of the terminal, there is a series of keyboard shortcuts for nano editor commands, followed by a search bar and some icons.

```
}
</style>
</head>
<body>
    <div class="main_page">
        <div class="page_header floating_element">
            
            <span class="floating_element">
                This is UECS2363 Vagrant Trial|
            </span>
        </div>
    <!-->
        <div class="table_of_contents floating_element">
            <div class="section_header section_header_grey">
                TABLE OF CONTENTS
            </div>
            <div class="table_of_contents_item floating_element">
```

^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U Uncut Tex ^T To Spell

Figure 23: Editing of index.html in nano editor.

Exit editing by ‘Ctrl + X’ then ‘Y’ to save changes and ‘Enter’ to finalize the change on index.html. Reloading the browser will reflect the changes made as illustrated in Figure 24.

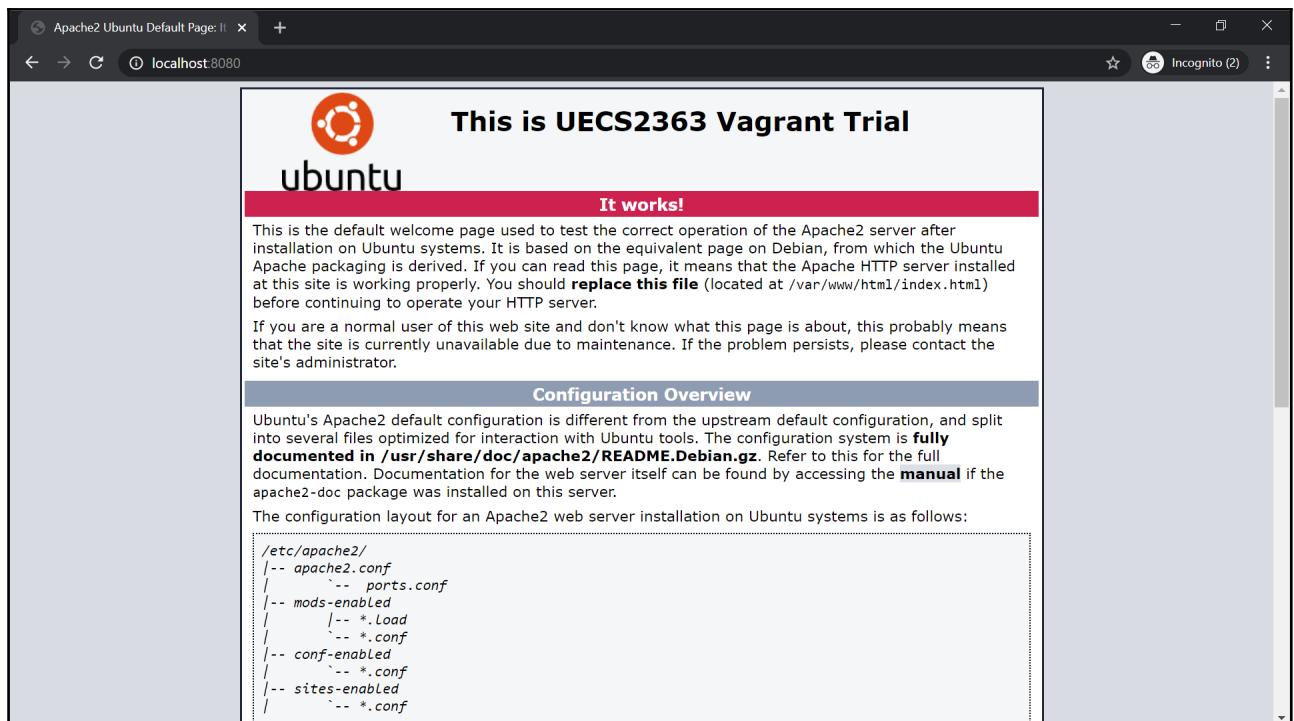
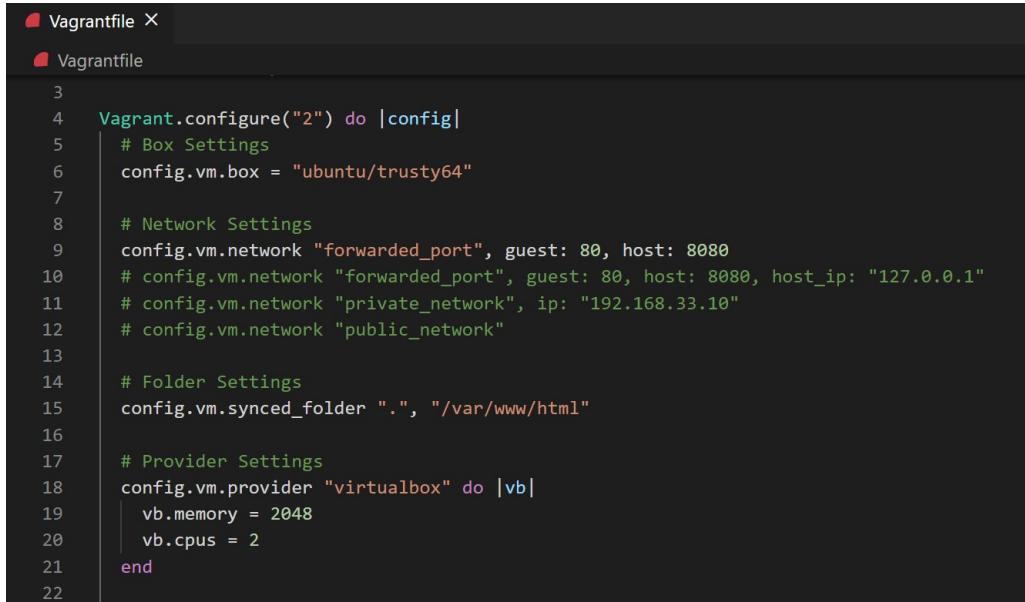


Figure 24: Edited index.html.

UECS2363 SOFTWARE CONSTRUCTION AND CONFIGURATION

This way of working is way too tedious; as we do not want to edit every file using nano or vim editor. Thus the synced_folder is useful for this. Looking back into the VagrantFile, focus on the #Folder Settings. Uncomment the folder setting with the first parameter (host / local workstation folder) set as '.' current folder and virtual box folder as '/var/www/html' as shown in Figure 25.



```
 3
 4 Vagrant.configure("2") do |config|
 5   # Box Settings
 6   config.vm.box = "ubuntu/trusty64"
 7
 8   # Network Settings
 9   config.vm.network "forwarded_port", guest: 80, host: 8080
10   # config.vm.network "forwarded_port", guest: 80, host: 8080, host_ip: "127.0.0.1"
11   # config.vm.network "private_network", ip: "192.168.33.10"
12   # config.vm.network "public_network"
13
14   # Folder Settings
15   config.vm.synced_folder ".", "/var/www/html"
16
17   # Provider Settings
18   config.vm.provider "virtualbox" do |vb|
19     vb.memory = 2048
20     vb.cpus = 2
21   end
22
```

Figure 25: Setting synced_folder for virtual box.

Save the changes in the VagrantFile in VS Code, then exit the vagrant ssh by executing exit command. Then execute vagrant reload again. If successful, the browser should load a folder which contains the VagrantFile as shown in Figure 26.

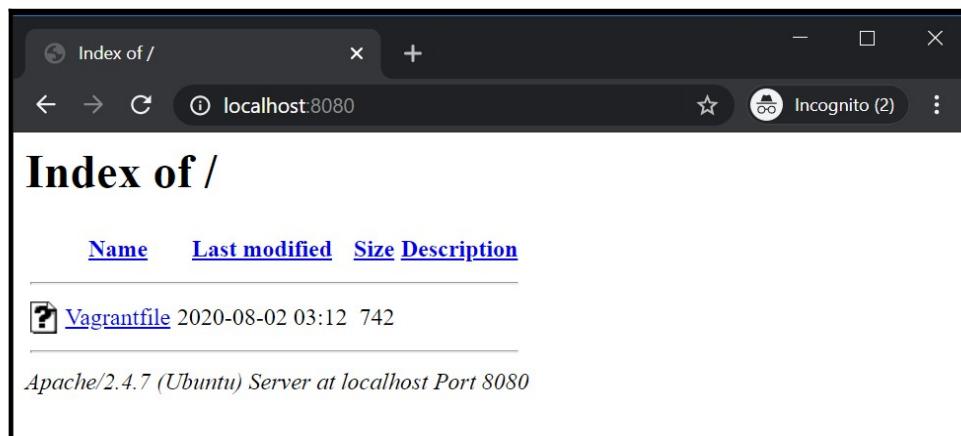


Figure 26: Loading synced_folder.

Now, you may add an index.html file in VS Code, besides VagrantFile and see to it that as you save the index.html file, the content is loaded at the refresh of browser.

UECS2363 SOFTWARE CONSTRUCTION AND CONFIGURATION

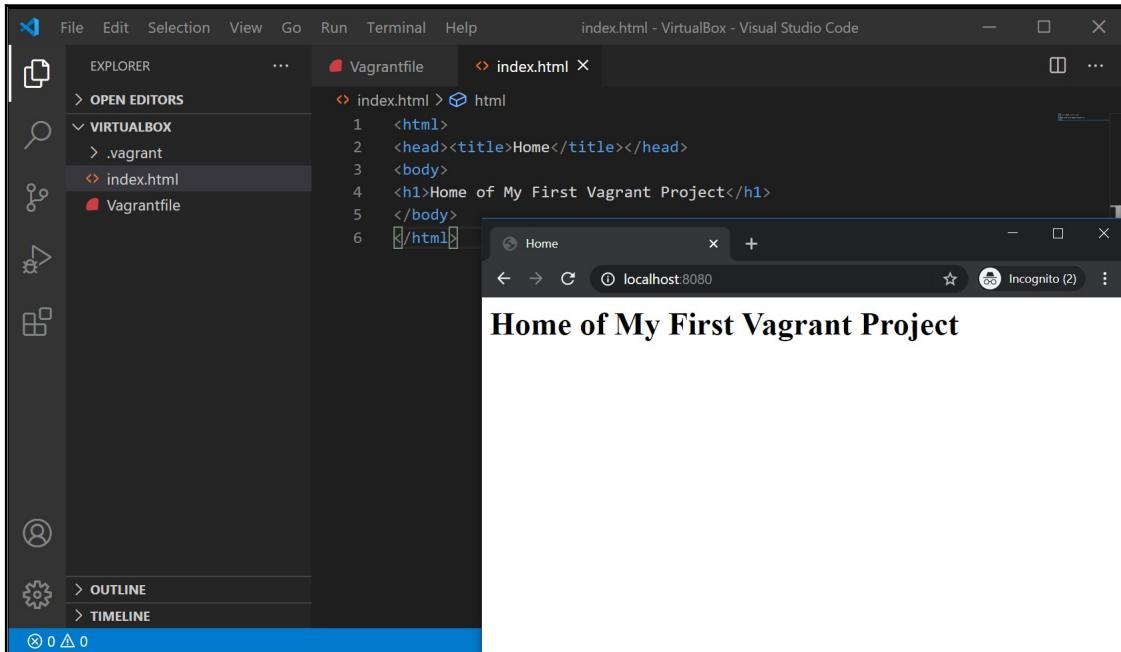


Figure 27: Editing software project files through synced_folder.

Configuring Virtual Box Provision Settings.

In order to look into this fundamental, we need to first delete the virtual box we created earlier. Execute `vagrant destroy` command for this. This step will not delete the VagrantFile, it just delete the box. Now, uncomment the configurations under #Provision Settings of the VagrantFile in VS Code and save it as shown in Figure 28.

The screenshot shows the Visual Studio Code interface with a dark theme. The Explorer sidebar lists files: .vagrant, index.html, and Vagrantfile. The Vagrantfile is open in the editor, showing configuration code. The code includes sections for network settings, provider settings (VirtualBox), and provision settings (shell). The provision settings section contains commands to update apt-get and install apache2. The status bar at the bottom right indicates the file is saved with 0 changes.

```
# config.vm.network "private_network", ip: "192.168.33.10"
# config.vm.network "public_network"

# Folder Settings
config.vm.synced_folder ".", "/var/www/html"

# Provider Settings
config.vm.provider "virtualbox" do |vb|
  vb.memory = 2048
  vb.cpus = 2
end

# Provision Settings
config.vm.provision "shell", inline: <<-SHELL
  apt-get update
  apt-get install -y apache2
SHELL
end
```

Figure 28: Configure provision for virtual box.

This will imply that when `vagrant up` command is executed, the scripts under the `config.vm.provision` will be executed as well. After saving the changes of provision configuration, back to the Cmder and execute `vagrant up` command to run the virtual box with the new settings.

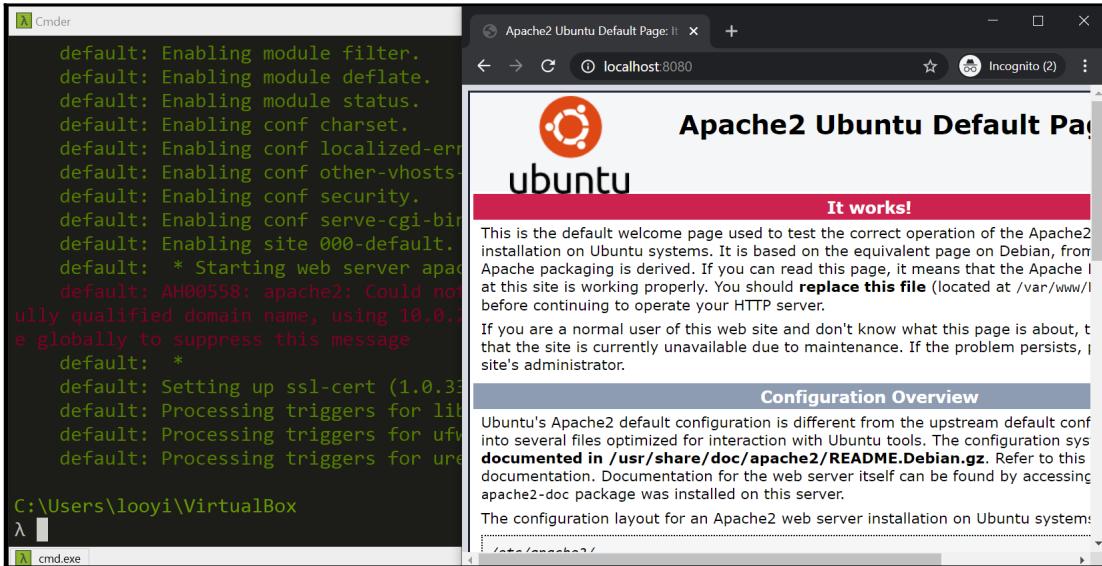


Figure 29: Virtual box and localhost after the changes.

Q: Why now the index.html became Apache2 Ubuntu default index.html?

Summary

1. Vagrant is a open-source command line utility for managing lifecycle of virtual machines.
2. VirtualBox or virtual box created from Vagrant, are virtual environments created for execution and testing of software/application development.
3. VagrantFile is the file that will be passed to collaborators so that the same environment of original developer can be replicated in workstations of collaborators.
4. 5 Core Fundamentals to explore:
 - a. config.vm.box – Virtual box options
 - b. config.vm.provider – Virtual box performance settings
 - c. config.vm.network – How the host is able to see the virtual box/application
 - d. config.vm.synced_folder – Access of project development files
 - e. config.vm.provision – what we want to setup (the platforms to execute and test the software/application)