EXPERIMENT - 1

Name: Raman Boora

SapID: 500109408

Roll No: R2142221160

Lab Exercise: Introduction to Vagrant and Vagrantfile

This exercise will guide them through setting up a virtual environment using Vagrant, configuring the environment via a Vagrantfile, and managing the virtual machines (VMs) with basic Vagrant commands.

Objective:

- Learn how to set up and configure virtual environments using Vagrant.
- Understand the structure and components of a Vagrantfile.
- Gain hands-on experience in managing virtual machines using Vagrant commands.

Prerequisites:

- Basic knowledge of virtualization concepts.
- Familiarity with command-line interfaces.
- Installation of Vagrant and VirtualBox (or any other supported provider) on your local machine.

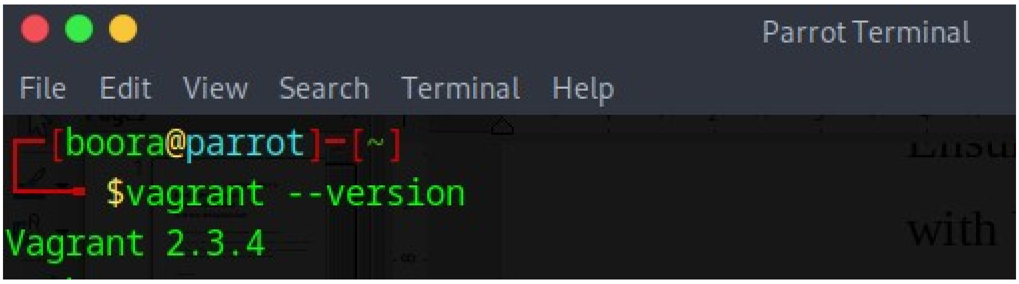
Step-by-Step Exercise:

1. Setting Up the Environment:

Install Vagrant:

- Download and install Vagrant from the official website.
- Ensure you have VirtualBox installed as it is a commonly used provider with Vagrant.
- Verify Installation:
- Open a terminal or command prompt.
- Run the following commands to verify the installation:

vagrant --version



```
┌─[boora@parrot]─[~]
└──╼ $vagrant --version
Vagrant 2.3.4
```

2. Creating a New Vagrant Project:

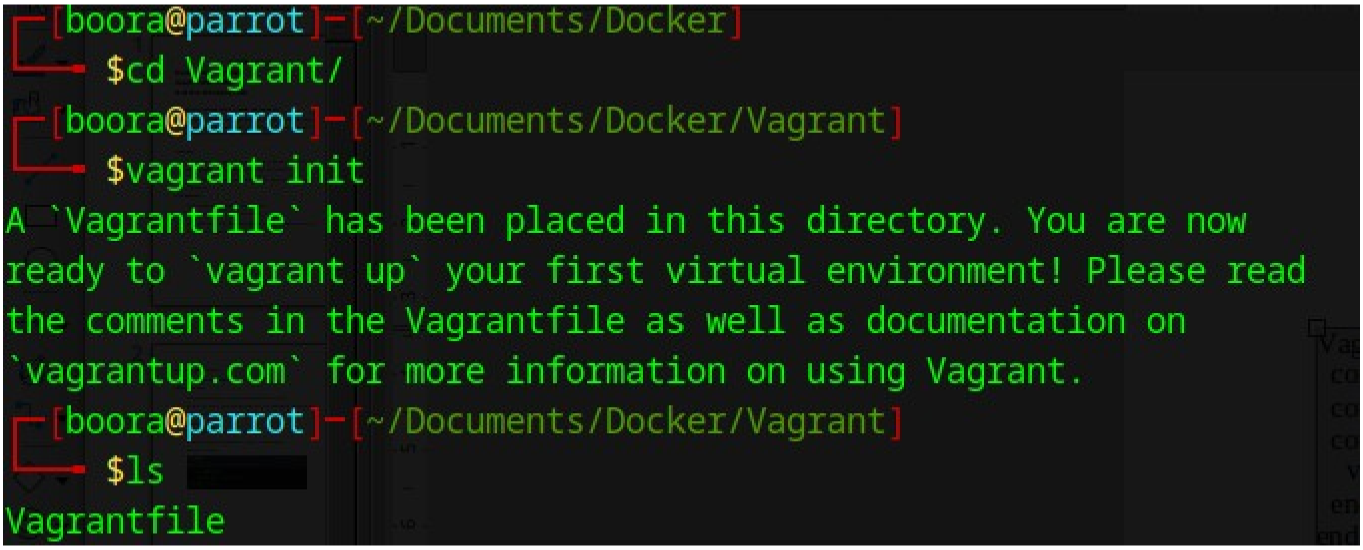
- Create a Project Directory:
- In your terminal, create a new directory for your Vagrant project and navigate into it:

```
mkdir vagrant_lab

cd vagrant_lab
```

Initialize Vagrant:

- Run the following command to initialize a new Vagrantfile in your project directory:

  vagrant init



```
┌─[boora@parrot]─[~/Documents/Docker]
└──╼ $cd Vagrant/
┌─[boora@parrot]─[~/Documents/Docker/Vagrant]
└──╼ $vagrant init
A `Vagrantfile` has been placed in this directory. You are now
ready to `vagrant up` your first virtual environment! Please read
the comments in the Vagrantfile as well as documentation on
`vagrantup.com` for more information on using Vagrant.
┌─[boora@parrot]─[~/Documents/Docker/Vagrant]
└──╼ $ls [redacted]
Vagrantfile
```

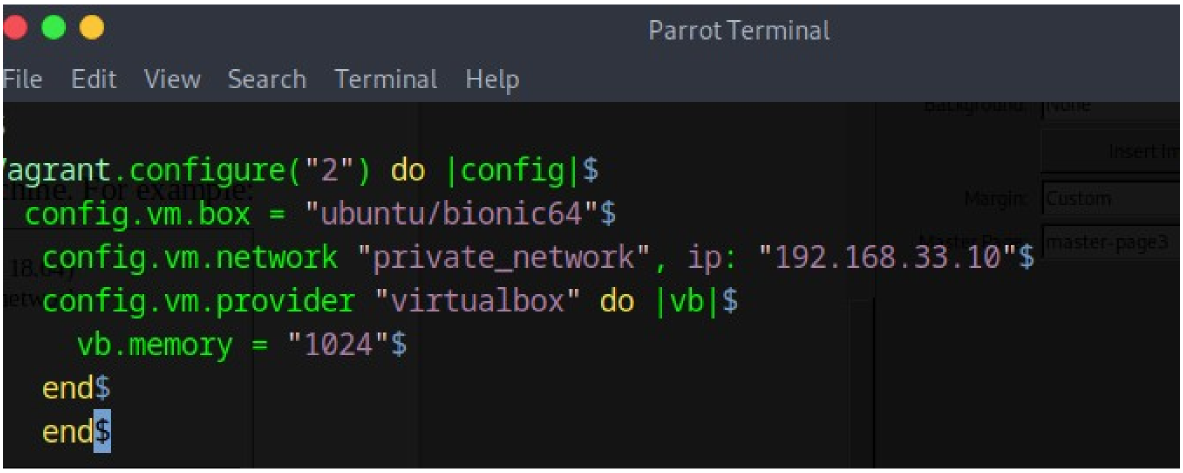
This command will generate a Vagrantfile in the current directory.

3. Understanding the Vagrantfile:

- Open the Vagrantfile:
- Open the Vagrantfile in a text editor of your choice.
- The Vagrantfile is a Ruby-based configuration file used to define the virtual environment.
- Basic Vagrantfile Configuration:

- Modify the Vagrantfile to configure a basic virtual machine. For example:

```
Vagrant.configure("2") do |config|
  config.vm.box = "ubuntu/bionic64"  # Specifies the base box to use (Ubuntu 18.04)
  config.vm.network "private_network", type: "dhcp"  # Configures a private network
  config.vm.provider "virtualbox" do |vb|
    vb.memory = "1024"  # Allocates 1GB of RAM to the VM
  end
end
```



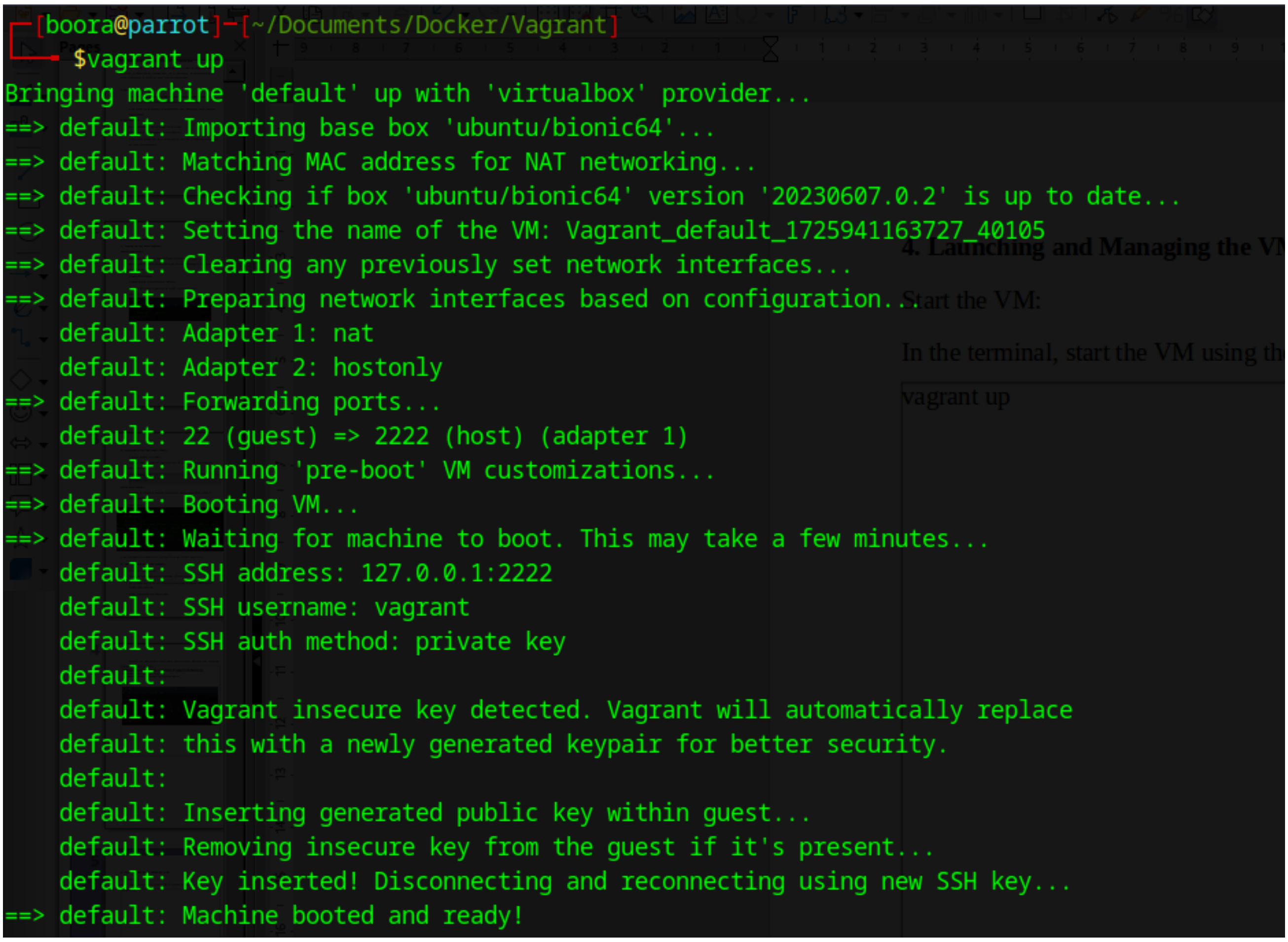
```
Vagrant.configure("2") do |config|$
 config.vm.box = "ubuntu/bionic64"$
  config.vm.network "private_network", ip: "192.168.33.10"$
  config.vm.provider "virtualbox" do |vb|$
    vb.memory = "1024"$
  end$
  end$
```

4. Launching and Managing the VM:

Start the VM:

In the terminal, start the VM using the following command:

vagrant up

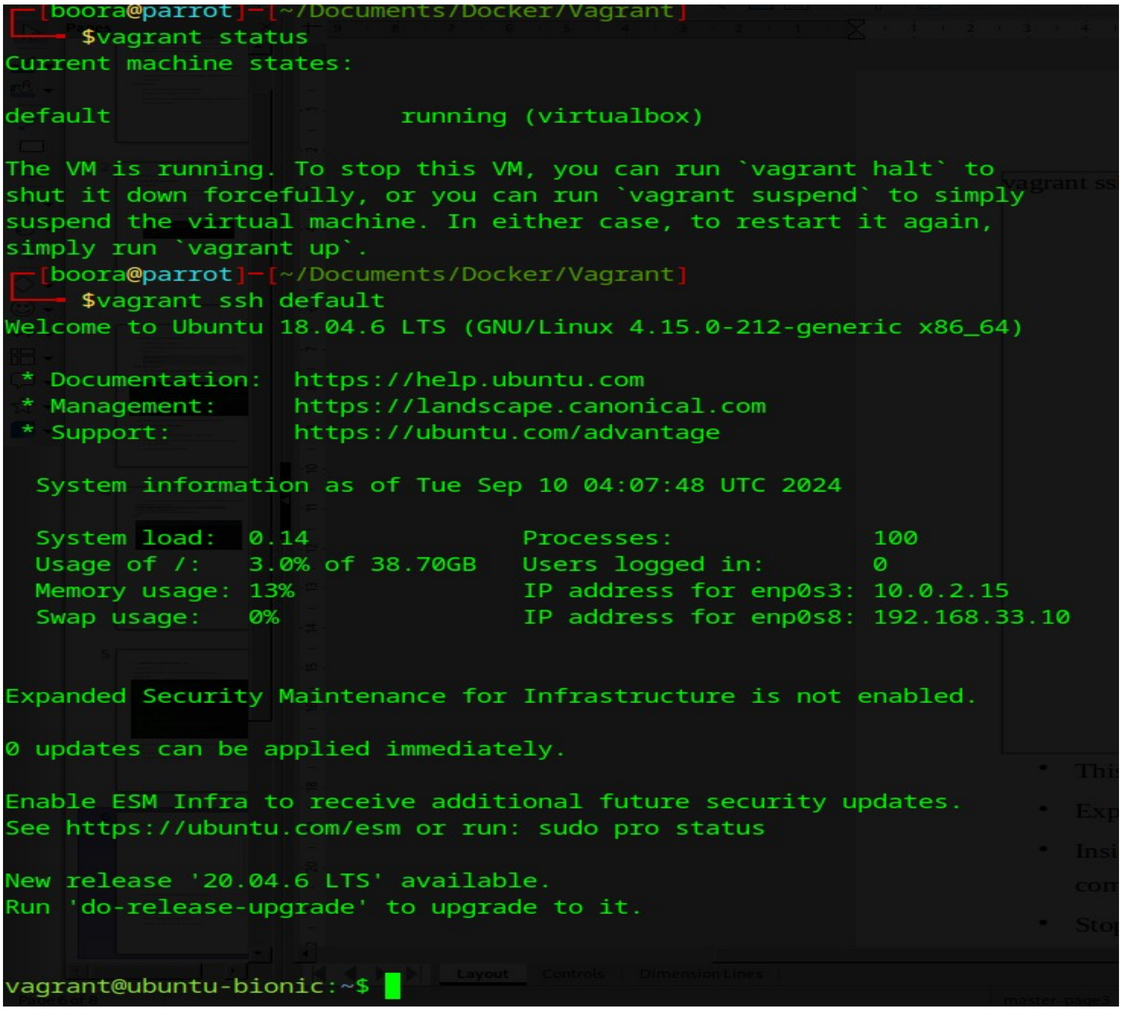


```
┌─[boora@parrot]─[~/Documents/Docker/Vagrant]
└──╼ $vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Importing base box 'ubuntu/bionic64'...
==> default: Matching MAC address for NAT networking...
==> default: Checking if box 'ubuntu/bionic64' version '20230607.0.2' is up to date...
==> default: Setting the name of the VM: Vagrant_default_1725941163727_40105
==> default: Clearing any previously set network interfaces...
==> default: Preparing network interfaces based on configuration...
    default: Adapter 1: nat
    default: Adapter 2: hostonly
==> default: Forwarding ports...
    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
    default: SSH auth method: private key
    default:
    default: Vagrant insecure key detected. Vagrant will automatically replace
    default: this with a newly generated keypair for better security.
    default:
    default: Inserting generated public key within guest...
    default: Removing insecure key from the guest if it's present...
    default: Key inserted! Disconnecting and reconnecting using new SSH key...
==> default: Machine booted and ready!
```

Vagrant will download the specified box (if not already downloaded) and launch the VM.

- SSH into the VM:
- Connect to the running VM using SSH:

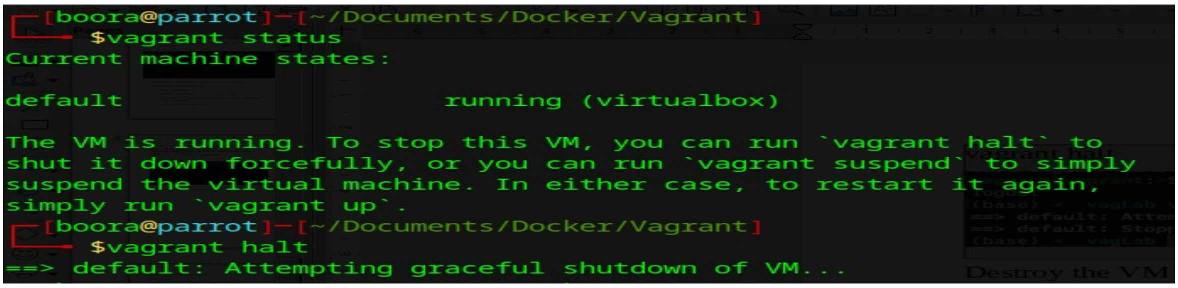
vagrant ssh



```
┌─[boora@parrot]─[~/Documents/Docker/Vagrant]
└──╼ $vagrant status
Current machine states:

default                   running (virtualbox)

The VM is running. To stop this VM, you can run `vagrant halt` to
shut it down forcefully, or you can run `vagrant suspend` to simply
suspend the virtual machine. In either case, to restart it again,
simply run `vagrant up`.
┌─[boora@parrot]─[~/Documents/Docker/Vagrant]
└──╼ $vagrant ssh default
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 4.15.0-212-generic x86_64)

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

  System information as of Tue Sep 10 04:07:48 UTC 2024

  System load:  0.14              Processes:             100
  Usage of /:   3.0% of 38.70GB   Users logged in:       0
  Memory usage: 13%               IP address for enp0s3: 10.0.2.15
  Swap usage:   0%                IP address for enp0s8: 192.168.33.10


Expanded Security Maintenance for Infrastructure is not enabled.

0 updates can be applied immediately.

Enable ESM Infra to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

New release '20.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


vagrant@ubuntu-bionic:~$
```

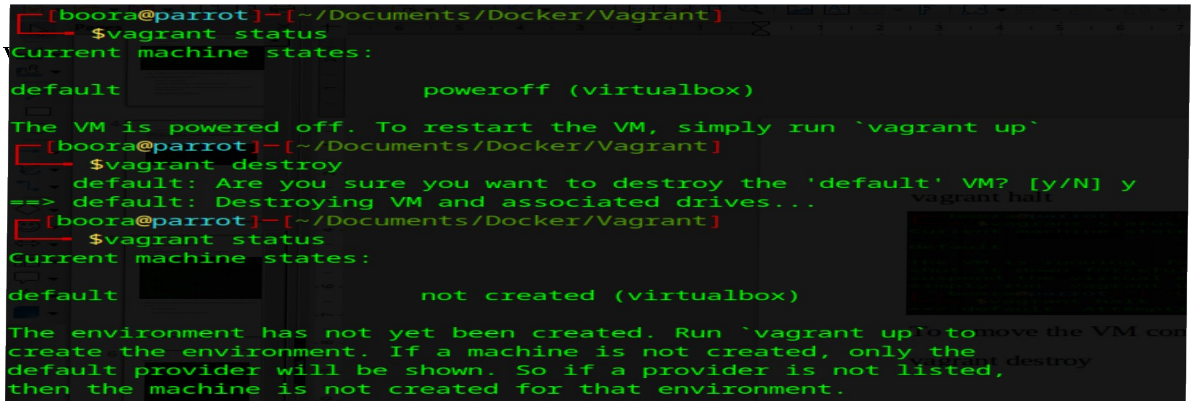
- This command will log you into the VM’s shell.
- Exploring the VM:
- Inside the VM, explore the filesystem, install packages, and run commands to understand the environment.
- Stop the VM:
- Exit the SSH session by typing exit.
- Stop the VM with the following command:

vagrant halt



```
┌─[boora@parrot]─[~/Documents/Docker/Vagrant]
└──╼ $vagrant status
Current machine states:

default                   running (virtualbox)

The VM is running. To stop this VM, you can run `vagrant halt` to
shut it down forcefully, or you can run `vagrant suspend` to simply
suspend the virtual machine. In either case, to restart it again,
simply run `vagrant up`.
┌─[boora@parrot]─[~/Documents/Docker/Vagrant]
└──╼ $vagrant halt
==> default: Attempting graceful shutdown of VM...
```

To remove the VM completely, use the following command:



```
┌─[boora@parrot]─[~/Documents/Docker/Vagrant]
└──╼ $vagrant status
Current machine states:

default                   poweroff (virtualbox)

The VM is powered off. To restart the VM, simply run `vagrant up`
┌─[boora@parrot]─[~/Documents/Docker/Vagrant]
└──╼ $vagrant destroy
    default: Are you sure you want to destroy the 'default' VM? [y/N] y
==> default: Destroying VM and associated drives...
┌─[boora@parrot]─[~/Documents/Docker/Vagrant]
└──╼ $vagrant status
Current machine states:

default                   not created (virtualbox)

The environment has not yet been created. Run `vagrant up` to
create the environment. If a machine is not created, only the
default provider will be shown. So if a provider is not listed,
then the machine is not created for that environment.
```

This will remove all traces of the VM, including any data stored on it. Explore the benefits of using Vagrant for development and testing environments.

Submission:

- Submit a brief report including the Vagrantfile you configured, screenshots of the running VM, and the output of any commands run within the VM.
- Reflect on the learning experience and any challenges faced during the exercise.

This lab exercise provides a hands-on introduction to Vagrant, focusing on creating and managing virtual environments through a Vagrantfile. It offers both foundational learning and opportunities to explore more advanced features.

My Views :

Vagrant is tool which automates the building of Virtual Machines.