

EXPERIMENT 1

Name: Raghav kamboj

SapID: 500102126

Roll No: R2142220137

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:

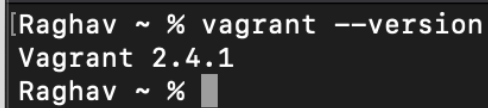
1. Setting Up the Environment:

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
```



```
[Raghav ~ % vagrant --version  
Vagrant 2.4.1  
Raghav ~ % ]
```

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

```
[Raghav ~ % mkdir vaglab.  
[Raghav ~ % cd vaglab.  
[Raghav vaglab. % vagrant init  
==> vagrant: A new version of Vagrant is available: 2.4.2 (installed version: 2.  
4.1)!  
==> vagrant: To upgrade visit: https://www.vagrantup.com/downloads.html  
  
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.  
[Raghav vaglab. % open Vagrantfile  
[Raghav vaglab. % █
```

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
```

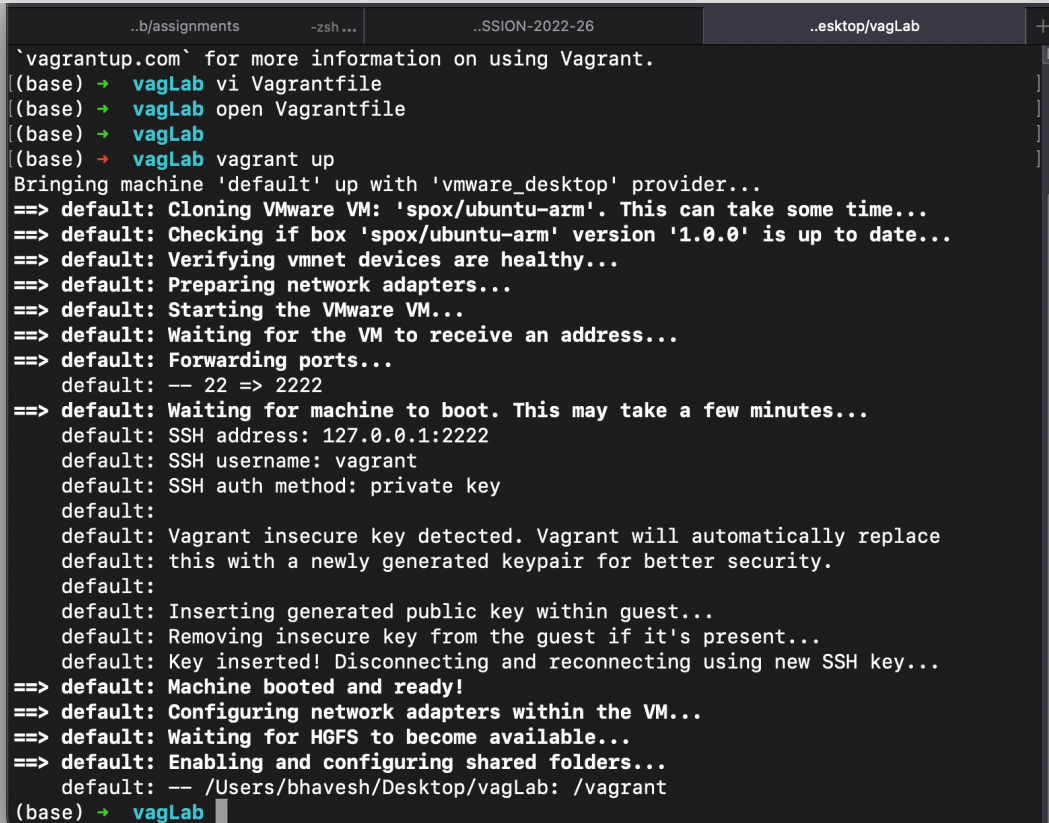
```
Vagrant.configure("2") do |config|
  config.vm.box = "spox/ubuntu-arm"
  config.vm.box_version = "1.0.0"
  config.vm.network "private_network", ip: "192.168.56.13"
  config.vm.provider "vmware_desktop" do |vmware|
    vmware.gui = true
    vmware.allowlist_verified = true
  end
end
```

4. Launching and Managing the VM:

Start the VM:

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

vagrant up

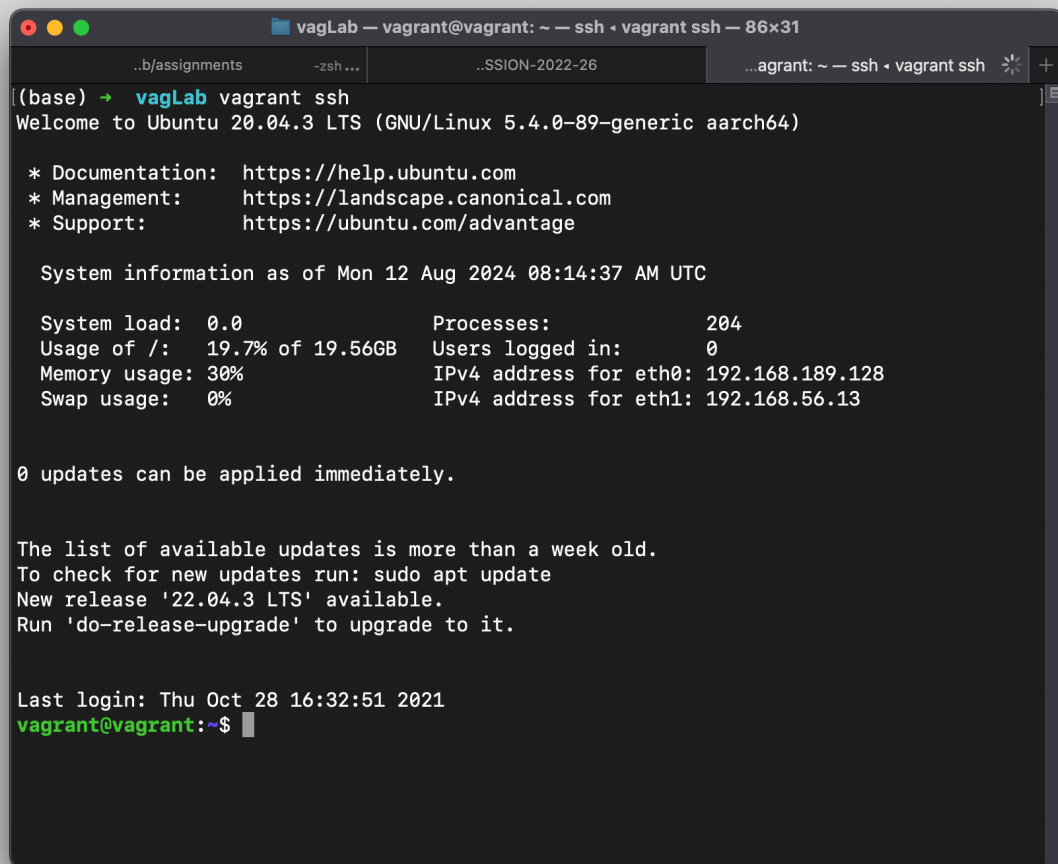
A terminal window with a dark background and light-colored text. The window has tabs at the top: '..b/assignments', '~zsh...', '..SSION-2022-26', and '..esktop/vagLab'. The terminal shows the following commands and output:

```
`vagrantup.com` for more information on using Vagrant.
(base) → vagLab vi Vagrantfile
(base) → vagLab open Vagrantfile
(base) → vagLab
(base) → vagLab vagrant up
Bringing machine 'default' up with 'vmware_desktop' provider...
==> default: Cloning VMware VM: 'spox/ubuntu-arm'. This can take some time...
==> default: Checking if box 'spox/ubuntu-arm' version '1.0.0' is up to date...
==> default: Verifying vmnet devices are healthy...
==> default: Preparing network adapters...
==> default: Starting the VMware VM...
==> default: Waiting for the VM to receive an address...
==> default: Forwarding ports...
default: -- 22 => 2222
==> 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!
==> default: Configuring network adapters within the VM...
==> default: Waiting for HGFS to become available...
==> default: Enabling and configuring shared folders...
default: -- /Users/bhavesesh/Desktop/vagLab: /vagrant
(base) → vagLab
```

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



```
((base) → vagLab vagrant ssh
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.4.0-89-generic aarch64)

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

System information as of Mon 12 Aug 2024 08:14:37 AM UTC

System load:  0.0           Processes:      204
Usage of /:   19.7% of 19.56GB Users logged in: 0
Memory usage: 30%          IPv4 address for eth0: 192.168.189.128
Swap usage:   0%           IPv4 address for eth1: 192.168.56.13

0 updates can be applied immediately.

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
New release '22.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Thu Oct 28 16:32:51 2021
vagrant@vagrant:~$
```

- 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

```
[vagrant@vagrant:~]$ exit  
logout  
(base) → vagLab vagrant halt  
==> default: Attempting graceful shutdown of VM...  
==> default: Stopping the VMware VM...  
(base) → vagLab
```

Destroy the VM (optional):

To remove the VM completely, use the following command:

vagrant destroy

```
[vagrant@vagrant:~]$ exit  
logout  
(base) → vagLab vagrant destroy  
default: Are you sure you want to destroy the 'default' VM? [y/N] y  
==> default: Stopping the VMware VM...  
==> default: Deleting the VM...  
(base) → vagLab
```

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.