EXPERIMENT 1

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.

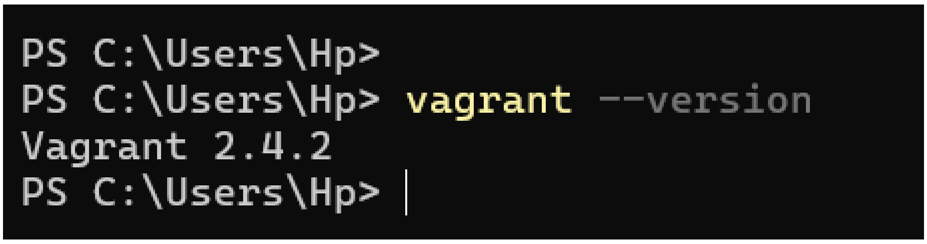
Step-by-Step Exercise:

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
```

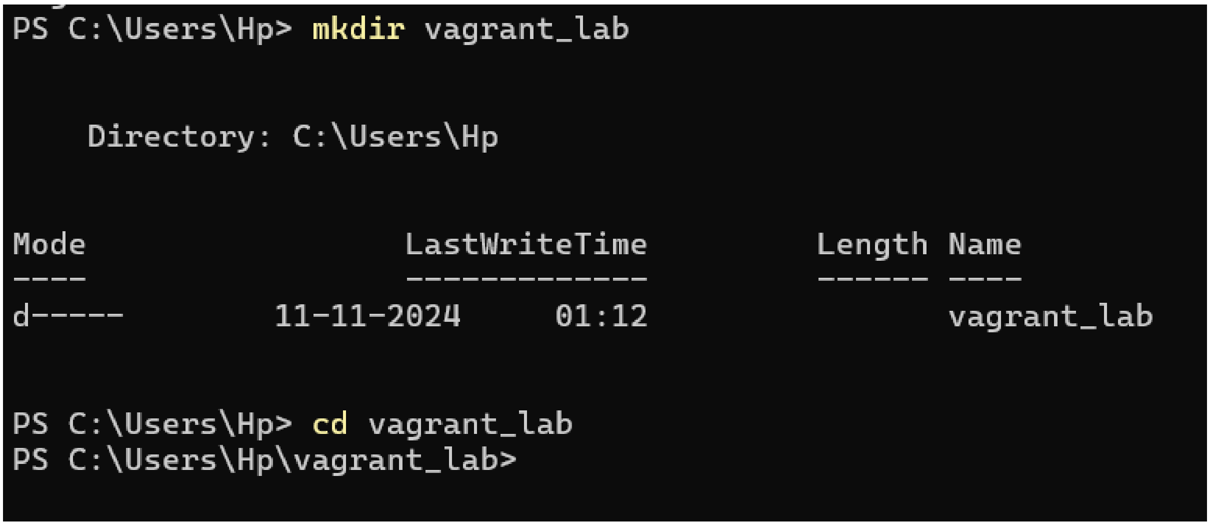


```
PS C:\Users\Hp>
PS C:\Users\Hp> vagrant --version
Vagrant 2.4.2
PS C:\Users\Hp>
```

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
```



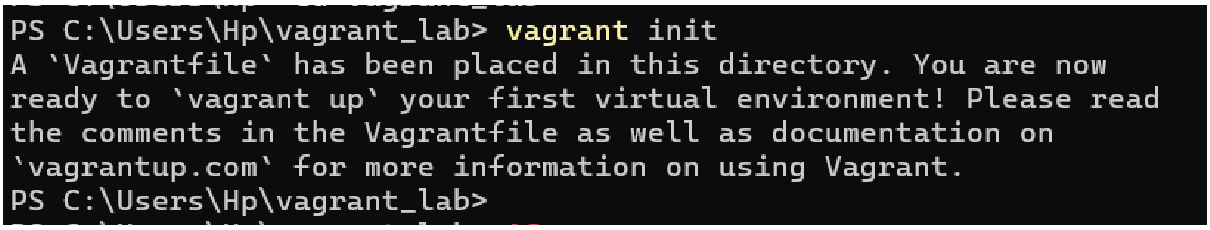
```
PS C:\Users\Hp> mkdir vagrant_lab


    Directory: C:\Users\Hp


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        11-11-2024     01:12                vagrant_lab


PS C:\Users\Hp> cd vagrant_lab
PS C:\Users\Hp\vagrant_lab>
```

Initialize Vagrant:

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

```
vagrant init
```

This command will generate a Vagrantfile in the current directory.



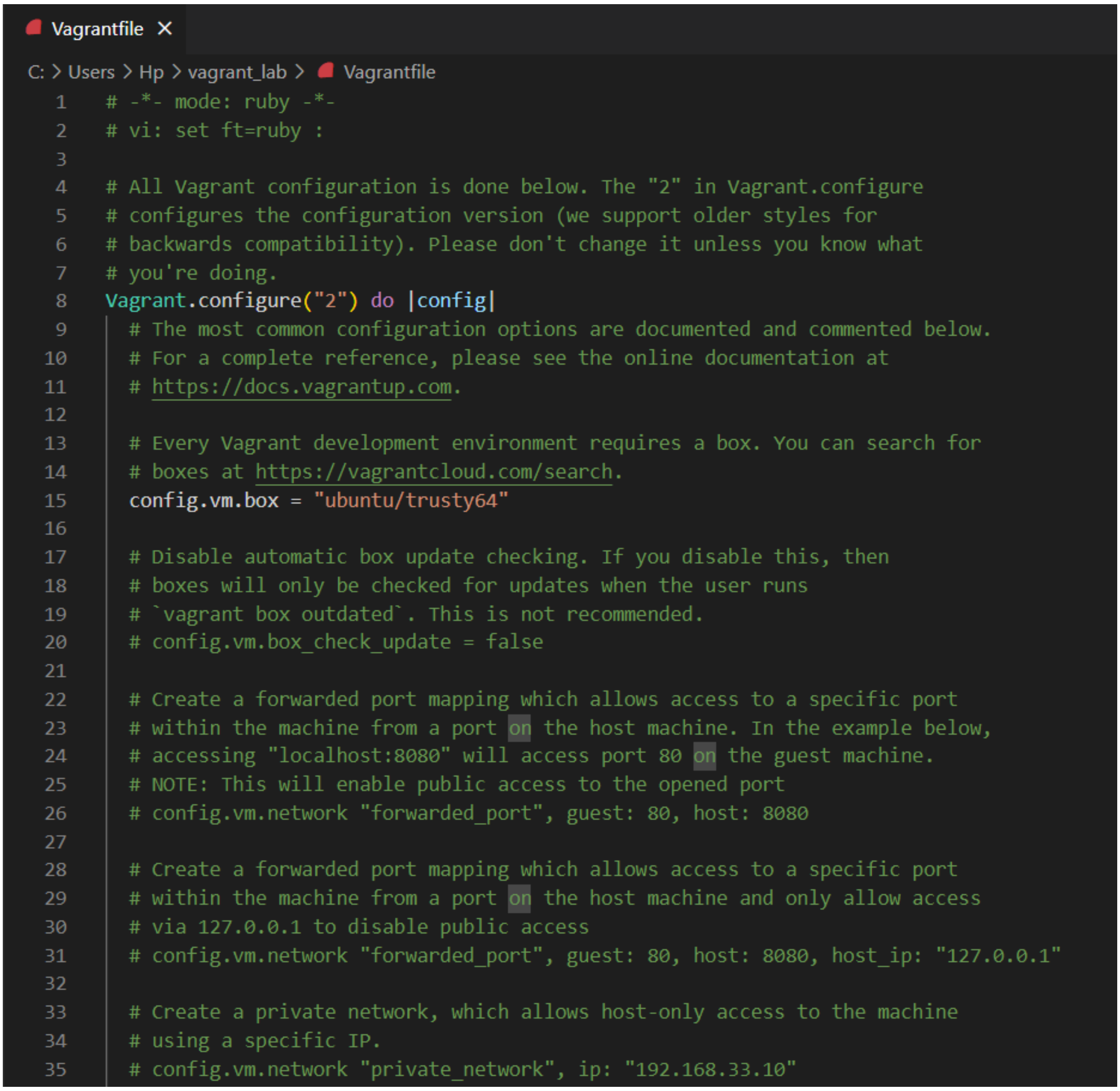
```
PS C:\Users\Hp\vagrant_lab> vagrant init
A `Vagrantfile` has been placed in this directory. You are now
ready to `vagrant up` your first virtual environment! Please read
the comments in the Vagrantfile as well as documentation on
`vagrantup.com` for more information on using Vagrant.
PS C:\Users\Hp\vagrant_lab>
```

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
```



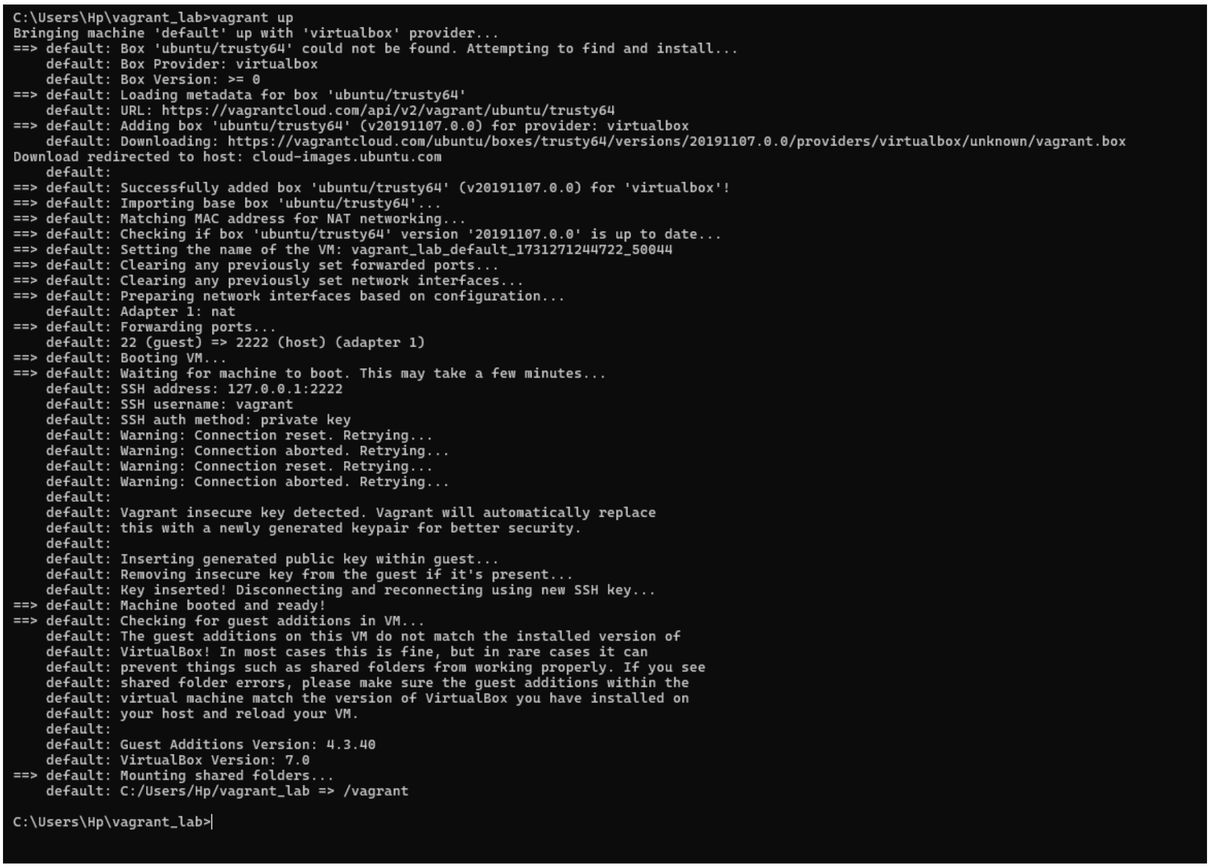
```
# -*- mode: ruby -*-
# vi: set ft=ruby :

# All Vagrant configuration is done below. The "2" in Vagrant.configure
# configures the configuration version (we support older styles for
# backwards compatibility). Please don't change it unless you know what
# you're doing.
Vagrant.configure("2") do |config|
  # The most common configuration options are documented and commented below.
  # For a complete reference, please see the online documentation at
  # https://docs.vagrantup.com.

  # Every Vagrant development environment requires a box. You can search for
  # boxes at https://vagrantcloud.com/search.
  config.vm.box = "ubuntu/trusty64"

  # Disable automatic box update checking. If you disable this, then
  # boxes will only be checked for updates when the user runs
  # `vagrant box outdated`. This is not recommended.
  # config.vm.box_check_update = false

  # Create a forwarded port mapping which allows access to a specific port
  # within the machine from a port on the host machine. In the example below,
  # accessing "localhost:8080" will access port 80 on the guest machine.
  # NOTE: This will enable public access to the opened port
  # config.vm.network "forwarded_port", guest: 80, host: 8080

  # Create a forwarded port mapping which allows access to a specific port
  # within the machine from a port on the host machine and only allow access
  # via 127.0.0.1 to disable public access
  # config.vm.network "forwarded_port", guest: 80, host: 8080, host_ip: "127.0.0.1"

  # Create a private network, which allows host-only access to the machine
  # using a specific IP.
  # config.vm.network "private_network", ip: "192.168.33.10"
```

4. Launching and Managing the VM:

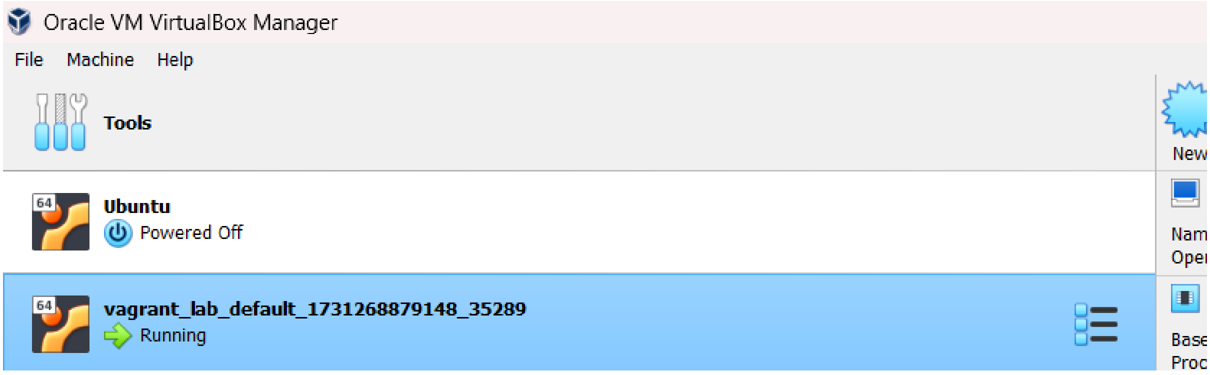
Start the VM:

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

```
vagrant up
```



```
C:\Users\Hp\vagrant_lab>vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Box 'ubuntu/trusty64' could not be found. Attempting to find and install...
    default: Box Provider: virtualbox
    default: Box Version: >= 0
==> default: Loading metadata for box 'ubuntu/trusty64'
    default: URL: https://vagrantcloud.com/api/v2/vagrant/ubuntu/trusty64
==> default: Adding box 'ubuntu/trusty64' (v20191107.0.0) for provider: virtualbox
    default: Downloading: https://vagrantcloud.com/ubuntu/boxes/trusty64/versions/20191107.0.0/providers/virtualbox/unknown/vagrant.box
Download redirected to host: cloud-images.ubuntu.com
    default:
==> default: Successfully added box 'ubuntu/trusty64' (v20191107.0.0) for 'virtualbox'!
==> default: Importing base box 'ubuntu/trusty64'...
==> default: Matching MAC address for NAT networking...
==> default: Checking if box 'ubuntu/trusty64' version '20191107.0.0' is up to date...
==> default: Setting the name of the VM: vagrant_lab_default_1731271244722_50044
==> 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: 22 (guest) => 2222 (host) (adapter 1)
==> 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: Warning: Connection reset. Retrying...
    default: Warning: Connection aborted. Retrying...
    default: Warning: Connection reset. Retrying...
    default: Warning: Connection aborted. Retrying...
    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: Checking for guest additions in VM...
    default: The guest additions on this VM do not match the installed version of
    default: VirtualBox! In most cases this is fine, but in rare cases it can
    default: prevent things such as shared folders from working properly. If you see
    default: shared folder errors, please make sure the guest additions within the
    default: virtual machine match the version of VirtualBox you have installed on
    default: your host and reload your VM.
    default:
    default: Guest Additions Version: 4.3.40
    default: VirtualBox Version: 7.0
==> default: Mounting shared folders...
    default: C:/Users/Hp/vagrant_lab => /vagrant

C:\Users\Hp\vagrant_lab>
```



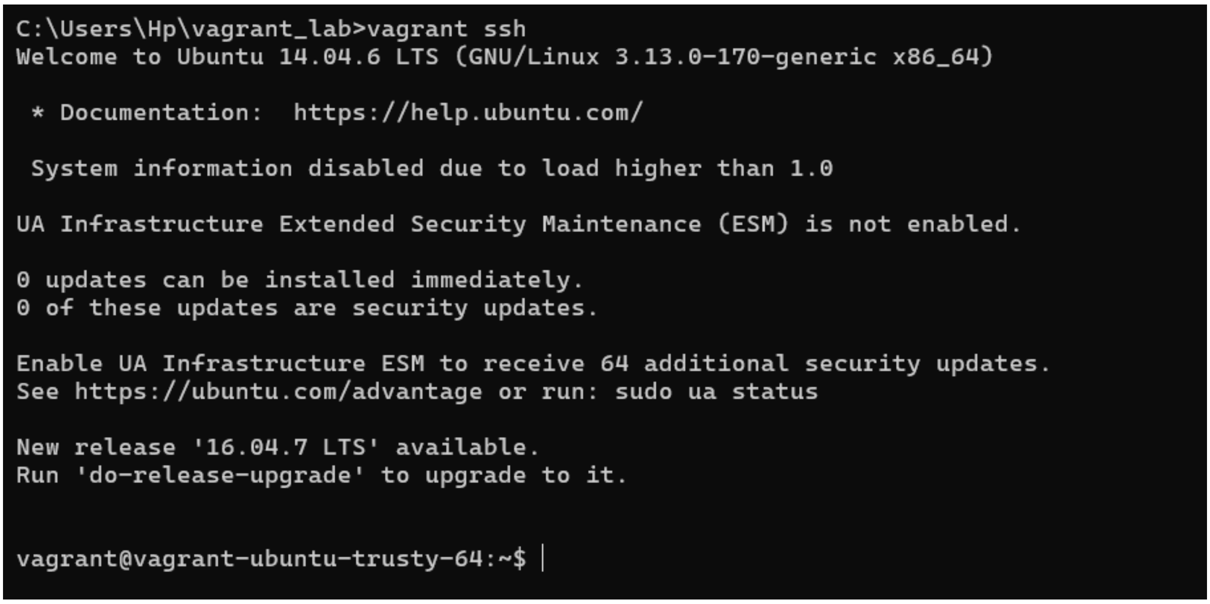
Ubuntu
Powered Off

vagrant_lab_default_1731268879148_35289
Running

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:

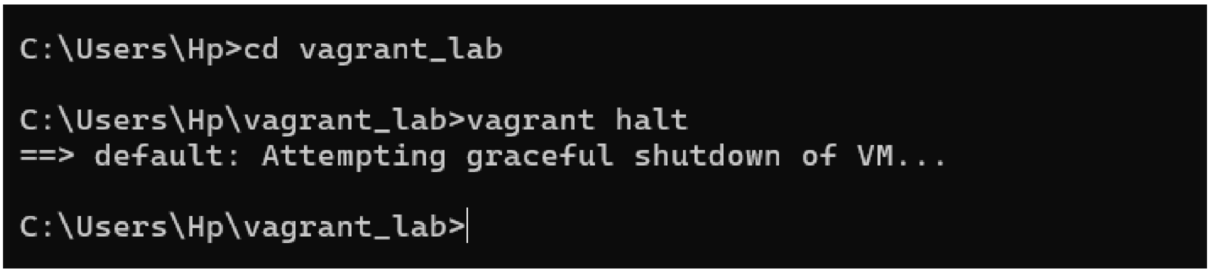
```
vagrant ssh
```



```
C:\Users\Hp\vagrant_lab>vagrant ssh
Welcome to Ubuntu 14.04.6 LTS (GNU/Linux 3.13.0-170-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

 System information disabled due to load higher than 1.0

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 '16.04.7 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


vagrant@vagrant-ubuntu-trusty-64:~$ 
```

- 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
```

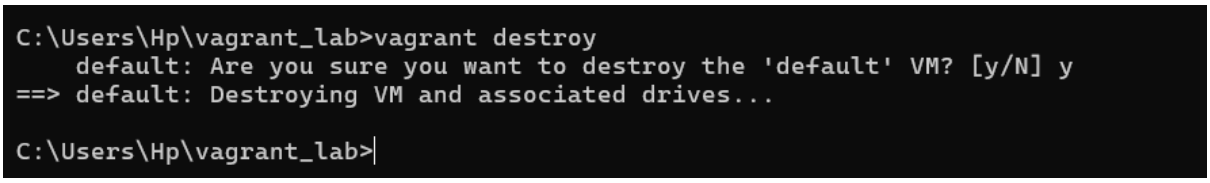


```
C:\Users\Hp>cd vagrant_lab

C:\Users\Hp\vagrant_lab>vagrant halt
==> default: Attempting graceful shutdown of VM...

C:\Users\Hp\vagrant_lab>
```

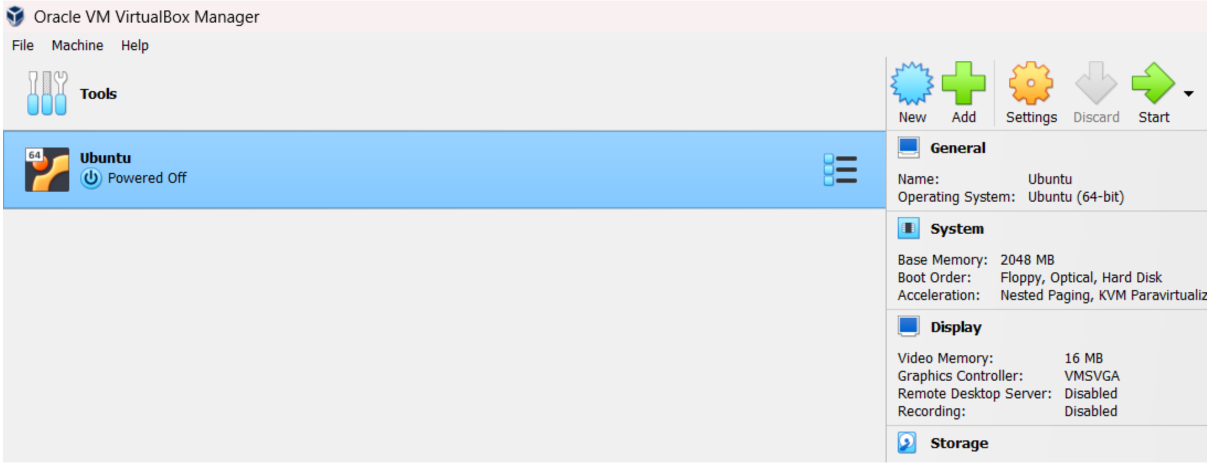
Destroy the VM (optional):

To remove the VM completely, use the following command:

```
vagrant destroy
```



```
C:\Users\Hp\vagrant_lab>vagrant destroy
    default: Are you sure you want to destroy the 'default' VM? [y/N] y
==> default: Destroying VM and associated drives...

C:\Users\Hp\vagrant_lab>
```



Ubuntu
Powered Off

General

Name: Ubuntu
Operating System: Ubuntu (64-bit)

System

Base Memory: 2048 MB
Boot Order: Floppy, Optical, Hard Disk
Acceleration: Nested Paging, KVM Paravirtualiz

Display

Video Memory: 16 MB
Graphics Controller: VMSVGA
Remote Desktop Server: Disabled
Recording: Disabled

Storage

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.