

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.

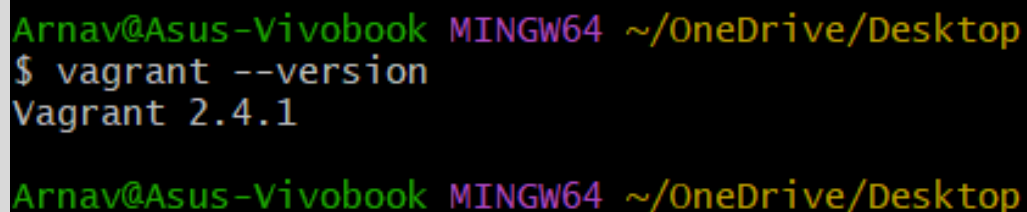
Step-by-Step Exercise:

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
```

A terminal window screenshot with a black background and colored text. The prompt 'Arnav@Asus-Vivobook' is in green, 'MINGW64' is in purple, and the path '~/OneDrive/Desktop' is in yellow. The command '\$ vagrant --version' is entered, and the output 'Vagrant 2.4.1' is displayed. The prompt and path are repeated on the next line.

```
Arnav@Asus-Vivobook MINGW64 ~/OneDrive/Desktop  
$ vagrant --version  
Vagrant 2.4.1  
Arnav@Asus-Vivobook MINGW64 ~/OneDrive/Desktop
```

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
```

```
Arnav@Asus-Vivobook MINGW64 ~  
$ cd "C:\Users\Arnav\OneDrive\Desktop"  
  
Arnav@Asus-Vivobook MINGW64 ~/OneDrive/Desktop  
$ mkdir vagrant-lab
```

Initialize Vagrant:

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

```
vagrant init
```

```
Arnav@Asus-Vivobook MINGW64 ~/OneDrive/Desktop/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.  
  
Arnav@Asus-Vivobook MINGW64 ~/OneDrive/Desktop/vagrant-lab  
$ ls  
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)

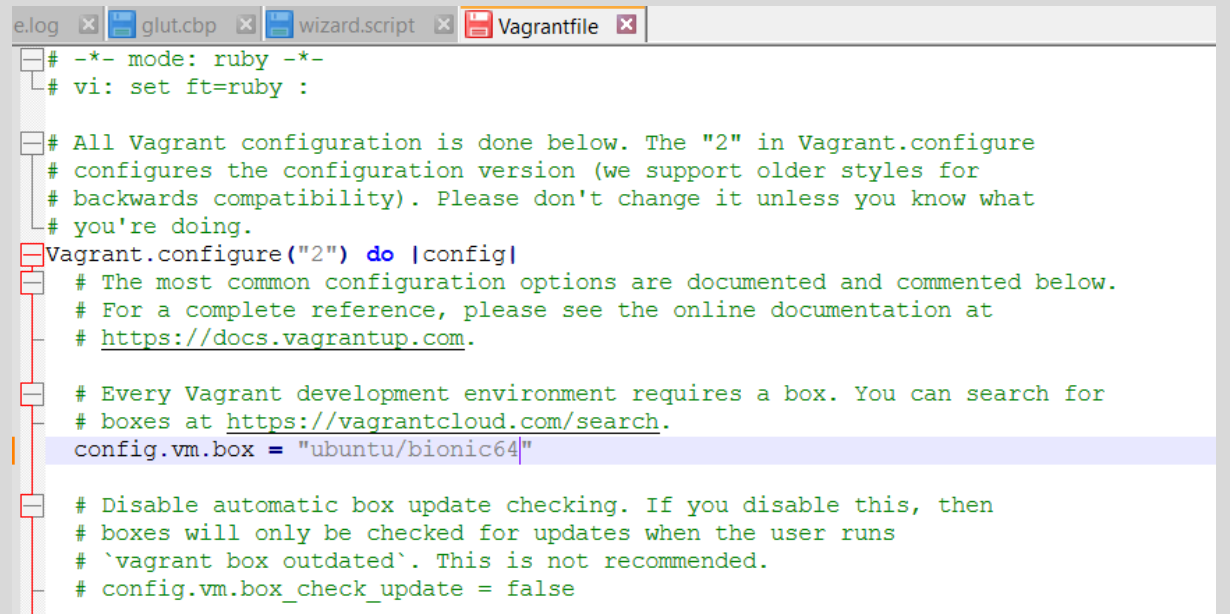
  config.vm.network "private_network", type: "dhcp" # Configures a private network

  config.vm.provider "virtualbox" do |vb|

    vb.memory = "1024" # Allocates 1GB of RAM to the VM

  end

end
```



```
# -*- 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/bionic64"

  # 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
```

4. Launching and Managing the VM:

Start the VM:

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

```
vagrant up
```

```

Arnav@Asus-Vivobook MINGW64 ~/OneDrive/Desktop/vagrant-lab
$ vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Box 'ubuntu/bionic64' could not be found. Attempting to find and install...
default: Box Provider: virtualbox
default: Box Version: >= 0
==> default: Loading metadata for box 'ubuntu/bionic64'
default: URL: https://vagrantcloud.com/api/v2/vagrant/ubuntu/bionic64
==> default: Adding box 'ubuntu/bionic64' (v20230607.0.1) for provider: virtualbox
default: Downloading: https://vagrantcloud.com/ubuntu/boxes/bionic64/versions/20230607.0.1/providers/virtualbox/unknown/vagrant
Download redirected to host: cloud-images.ubuntu.com
default:
==> default: Successfully added box 'ubuntu/bionic64' (v20230607.0.1) for 'virtualbox'!
==> default: Importing base box 'ubuntu/bionic64'...
==> default: Matching MAC address for NAT networking...
==> default: Checking if box 'ubuntu/bionic64' version '20230607.0.1' is up to date...
==> default: Setting the name of the VM: vagrant-lab_default_1723445832138_21439
Vagrant is currently configured to create VirtualBox synced folders with
the 'SharedFoldersEnableSymlinksCreate' option enabled. If the Vagrant
guest is not trusted, you may want to disable this option. For more
information on this option, please refer to the VirtualBox manual:

https://www.virtualbox.org/manual/ch04.html#sharedfolders

This option can be disabled globally with an environment variable:

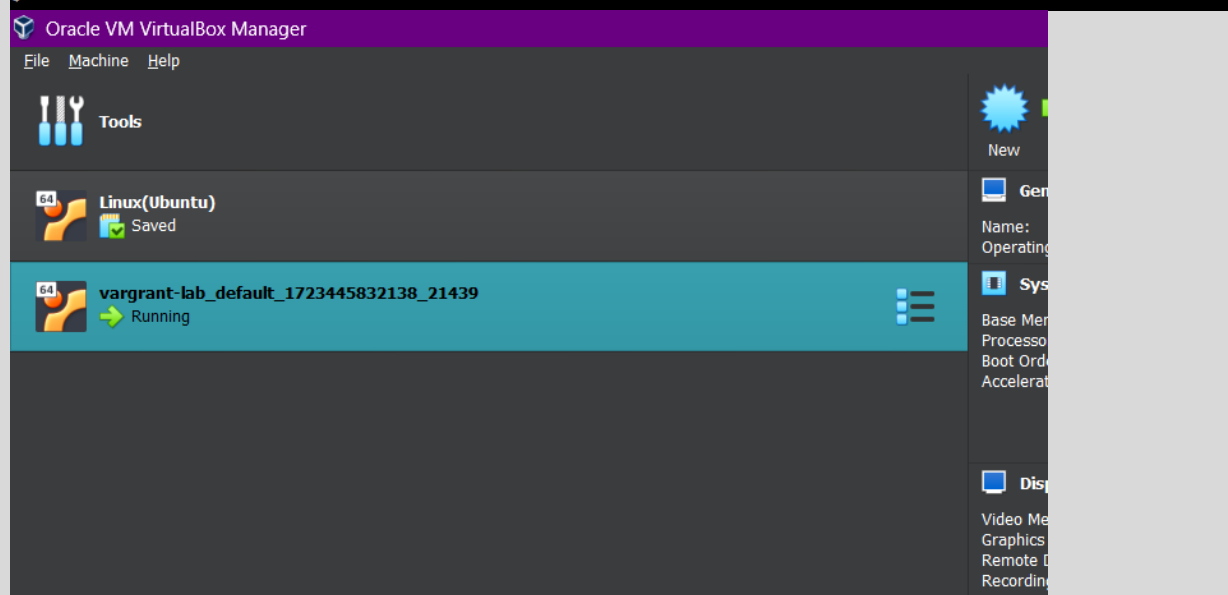
VAGRANT_DISABLE_VBOXSYMLINKCREATE=1

or on a per folder basis within the Vagrantfile:

config.vm.synced_folder '/host/path', '/guest/path', sharedFoldersEnableSymlinksCreate: false
==> 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: 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!
==> 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: 5.2.42
default: VirtualBox Version: 7.0
==> default: Mounting shared folders...
default: /vagrant => C:/Users/Arnav/OneDrive/Desktop/vagrant-lab

Arnav@Asus-Vivobook MINGW64 ~/OneDrive/Desktop/vagrant-lab
$

```



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

Arnav@Asus-Vivobook MINGW64 ~/OneDrive/Desktop/vagrant-lab
$ vagrant ssh
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 Mon Aug 12 06:59:06 UTC 2024

System load:  0.35               Processes:            101
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%

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

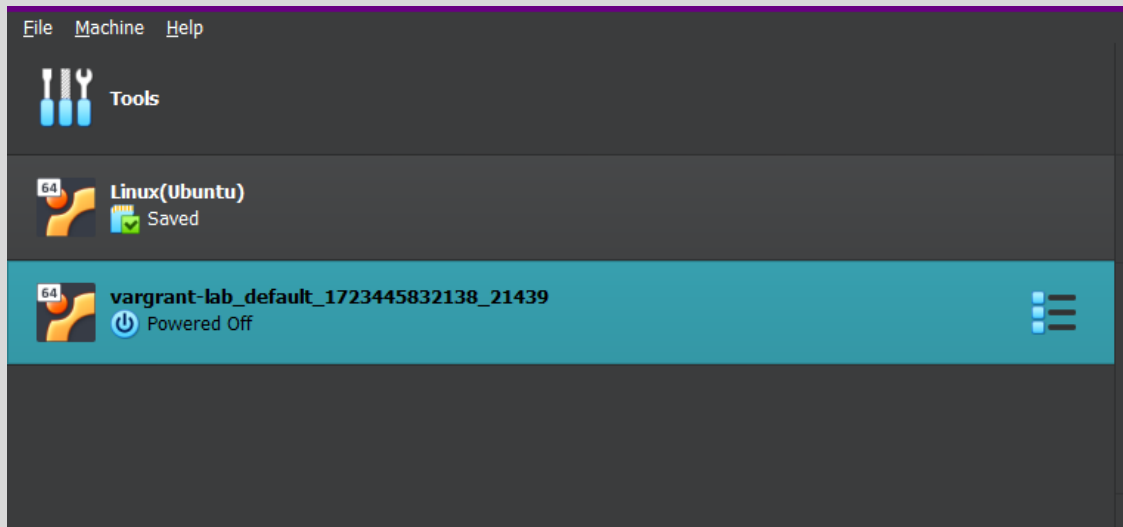
New release '20.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

vagrant@ubuntu-bionic:~$ ls -la
.  ..  .bash_logout  .bashrc  .cache  .gnupg  .profile  .ssh
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

```
Arnav@Asus-Vivobook MINGW64 ~/OneDrive/Desktop/vagrant-lab
$ vagrant halt
==> default: Attempting graceful shutdown of VM...
```



Destroy the VM (optional):

To remove the VM completely, use the following command:

vagrant destroy

```
Arnav@Asus-Vivobook MINGW64 ~/OneDrive/Desktop/vagrant-lab
$ vagrant destroy
   default: Are you sure you want to destroy the 'default' VM? [y/N] y
==> default: Destroying VM and associated drives...

Arnav@Asus-Vivobook MINGW64 ~/OneDrive/Desktop/vagrant-lab
$
```

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.