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.

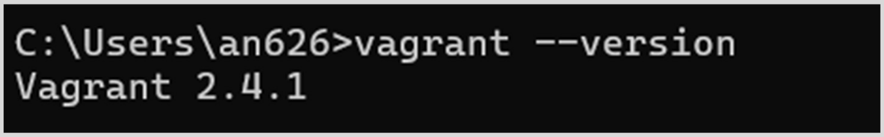
Step-by-Step Exercise:

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
```

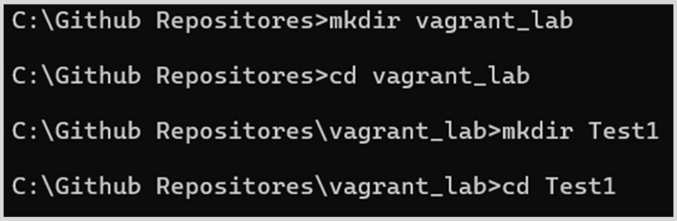


```
C:\Users\an626>vagrant --version
Vagrant 2.4.1
```

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
```

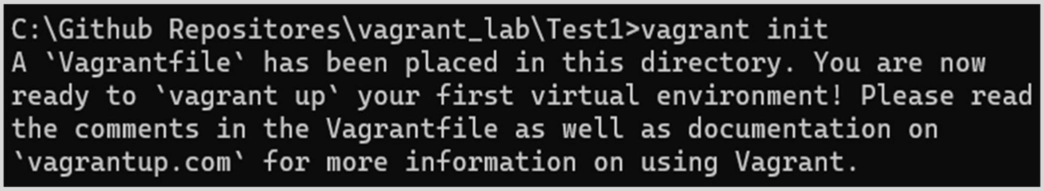


```
C:\Github Repositores>mkdir vagrant_lab

C:\Github Repositores>cd vagrant_lab

C:\Github Repositores\vagrant_lab>mkdir Test1

C:\Github Repositores\vagrant_lab>cd Test1
```

Initialize Vagrant:

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

```
vagrant init
```



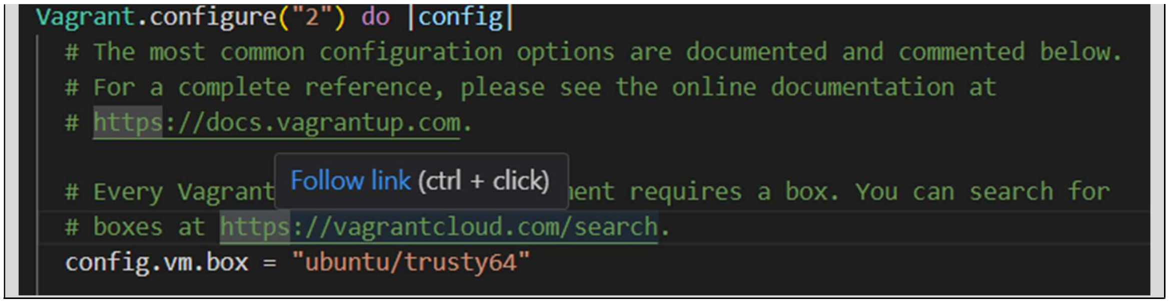
```
C:\Github Repositores\vagrant_lab\Test1>vagrant init
A `Vagrantfile` has been placed in this directory. You are now
ready to `vagrant up` your first virtual environment! Please read
the comments in the Vagrantfile as well as documentation on
`vagrantup.com` for more information on using Vagrant.
```

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
```



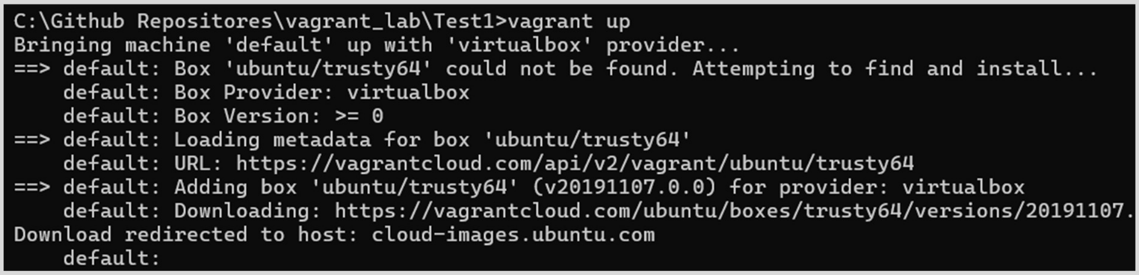
```
Vagrant.configure("2") do |config|
  # The most common configuration options are documented and commented below.
  # For a complete reference, please see the online documentation at
  # https://docs.vagrantup.com.

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

4. Launching and Managing the VM:

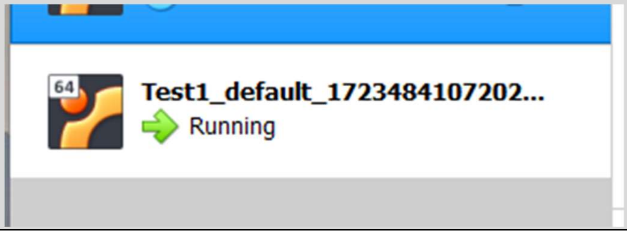
Start the VM:

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

```
vagrant up
```



```
C:\Github Repositores\vagrant_lab\Test1>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.
Download redirected to host: cloud-images.ubuntu.com
    default:
```

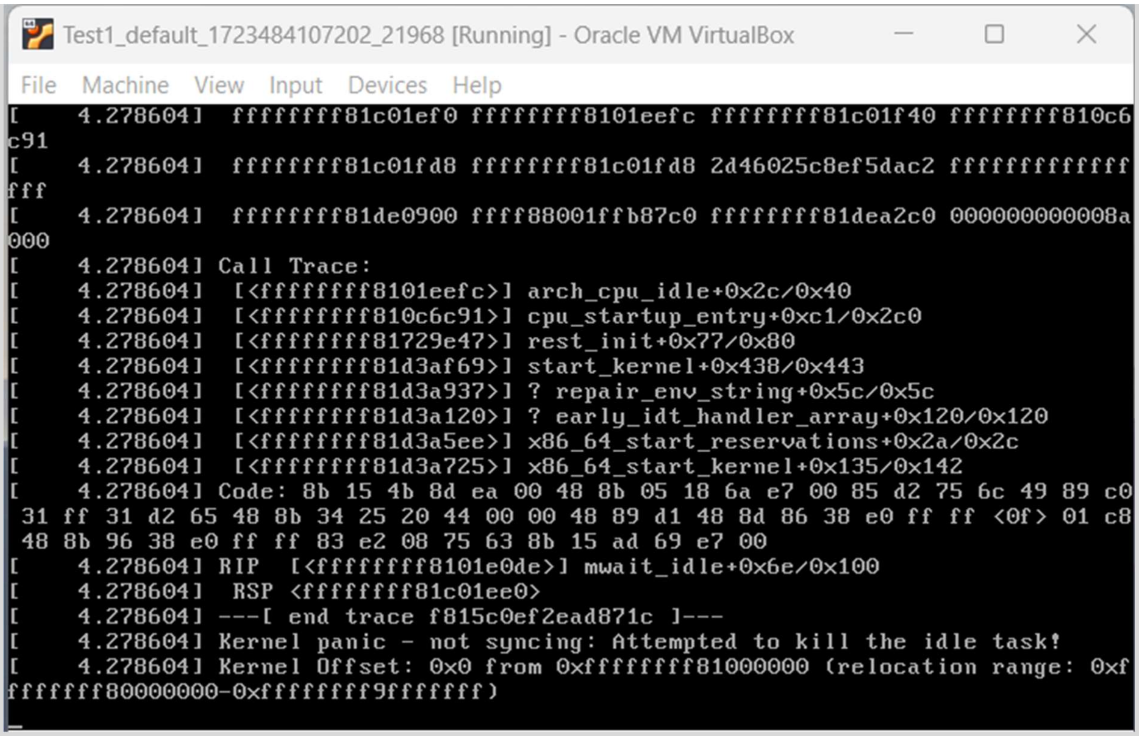


Test1_default_1723484107202...
Running

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:

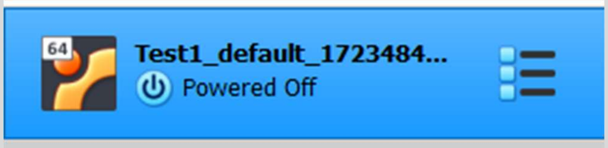
```
vagrant ssh
```


Test1_default_1723484107202_21968 [Running] - Oracle VM VirtualBox
File
Machine
View
Input
Devices
Help

```
[    4.278604]  ffffffff81c01ef0 ffffffff8101eefc ffffffff81c01f40 ffffffff810c6
c91
[    4.278604]  ffffffff81c01fd8 ffffffff81c01fd8 2d46025c8ef5dac2 fffffffffffff
fff
[    4.278604]  ffffffff81de0900 ffff88001ffb87c0 ffffffff81dea2c0 000000000008a
000
[    4.278604] Call Trace:
[    4.278604]  [<ffffffff8101eefc>] arch_cpu_idle+0x2c/0x40
[    4.278604]  [<ffffffff810c6c91>] cpu_startup_entry+0xc1/0x2c0
[    4.278604]  [<ffffffff81729e47>] rest_init+0x77/0x80
[    4.278604]  [<ffffffff81d3af69>] start_kernel+0x438/0x443
[    4.278604]  [<ffffffff81d3a937>] ? repair_env_string+0x5c/0x5c
[    4.278604]  [<ffffffff81d3a120>] ? early_idt_handler_array+0x120/0x120
[    4.278604]  [<ffffffff81d3a5ee>] x86_64_start_reservations+0x2a/0x2c
[    4.278604]  [<ffffffff81d3a725>] x86_64_start_kernel+0x135/0x142
[    4.278604] Code: 8b 15 4b 8d ea 00 48 8b 05 18 6a e7 00 85 d2 75 6c 49 89 c0
 31 ff 31 d2 65 48 8b 34 25 20 44 00 00 48 89 d1 48 8d 86 38 e0 ff ff <0f> 01 c8
 48 8b 96 38 e0 ff ff 83 e2 08 75 63 8b 15 ad 69 e7 00
[    4.278604] RIP  [<ffffffff8101e0de>] mwait_idle+0x6e/0x100
[    4.278604]  RSP <ffffffff81c01ee0>
[    4.278604] ---[ end trace f815c0ef2ead871c ]---
[    4.278604] Kernel panic - not syncing: Attempted to kill the idle task!
[    4.278604] Kernel Offset: 0x0 from 0xffffffff81000000 (relocation range: 0xf
fffffff80000000-0xffffffff9fffffff)
_
```

- 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:

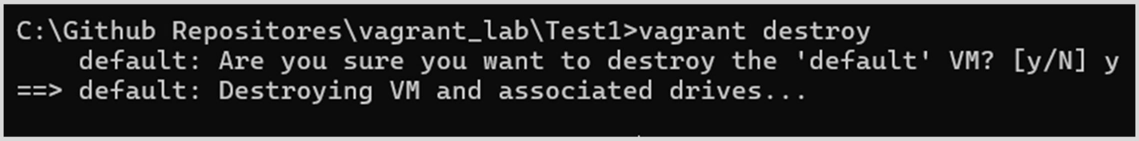
```
vagrant halt
```



Destroy the VM (optional):

To remove the VM completely, use the following command:

```
vagrant destroy
```



```
C:\Github Repositores\vagrant_lab\Test1>vagrant destroy
    default: Are you sure you want to destroy the 'default' VM? [y/N] y
==> default: Destroying VM and associated drives...
```

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.