

# 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
```

```
suja1@HP-Victus MINGW64 ~  
$ vagrant --version  
Vagrant 2.4.1  
  
suja1@HP-Victus MINGW64 ~  
$ |
```

### 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
```

```
suja1@HP-Victus MINGW64 ~/OneDrive/Desktop  
$ mkdir vagrant_lab  
  
suja1@HP-Victus MINGW64 ~/OneDrive/Desktop  
$ cd 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.

```
suja1@HP-Victus 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.
```

## 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
```

## 4. Launching and Managing the VM:

Start the VM:

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

```
vagrant up
```

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
```

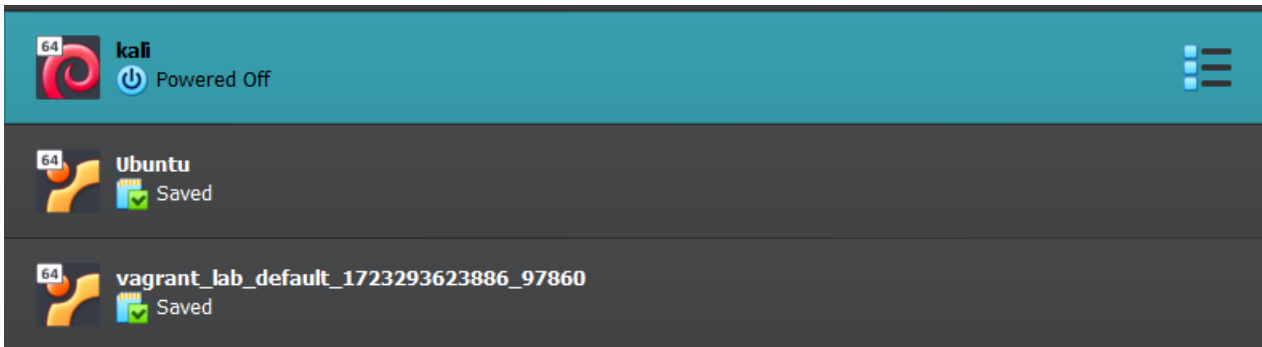
```
suja1@HP-Victus MINGW64 ~/OneDrive/Desktop/vagrant_lab
$ vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Importing base box 'ubuntu/xenial64'...
==> default: Matching MAC address for NAT networking...
==> default: Checking if box 'ubuntu/xenial64' version '20211001.0.1' is up to date...
==> default: Setting the name of the VM: vagrant_lab_default_1723426110053_34483
==> 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: 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: 5.1.38
default: VirtualBox Version: 7.0
==> default: Mounting shared folders...
default: /vagrant => C:/Users/suja1/OneDrive/Desktop/vagrant_lab

suja1@HP-Victus MINGW64 ~/OneDrive/Desktop/vagrant_lab
$ vagrant ssh
Welcome to Ubuntu 16.04.7 LTS (GNU/Linux 4.4.0-210-generic x86_64)

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

- 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
```

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.