ANSHIKA SRIVASTAVA

ROLL NUMBER – R2142220907

SAP ID – 500107049

DEVSECOPS BATCH B1 HONS.

EXPERIMENT – 1

Introduction to Vagrant and Vagrantfile
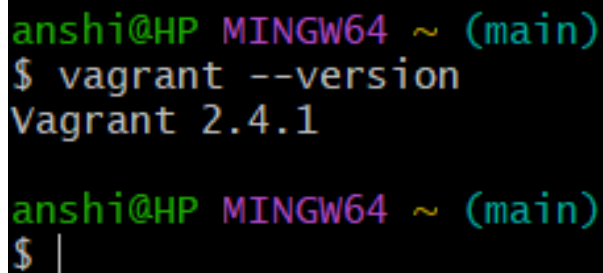
1. **Setting Up the Environment:**

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 commands to verify the installation

vagrant –version

```
anshi@HP MINGW64 ~ (main)
$ vagrant --version
Vagrant 2.4.1

anshi@HP MINGW64 ~ (main)
$ |
```

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

```
anshi@HP MINGW64 /d/Academics/Docker
$ mkdir vagrant_lab

anshi@HP MINGW64 /d/Academics/Docker
$ cd vagrant_lab

anshi@HP MINGW64 /d/Academics/Docker/vagrant_lab
$ |
```

3. Initialize Vagrant:

To initialize a new Vagrantfile in your project directory:

vagrant init

```
anshi@HP MINGW64 /d/Academics/Docker/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.

anshi@HP MINGW64 /d/Academics/Docker/vagrant_lab
$ ls
Vagrantfile

anshi@HP MINGW64 /d/Academics/Docker/vagrant_lab
$ |
```

4. Understanding the Vagrantfile:

Open the Vagrantfile in any text editor and modify the Vagrantfile to configure a basic virtual machine.

Default file

```
Vagrantfile ×

D: > Academics > Docker > vagrant_lab > 🔴 Vagrantfile
  1  # -*- mode: ruby -*-
  2  # vi: set ft=ruby :
  3
  4  # All Vagrant configuration is done below. The "2" in Vagrant.configure
  5  # configures the configuration version (we support older styles for
  6  # backwards compatibility). Please don't change it unless you know what
  7  # you're doing.
  8  Vagrant.configure("2") do |config|
  9    # The most common configuration options are documented and commented below.
 10    # For a complete reference, please see the online documentation at
 11    # https://docs.vagrantup.com.
 12
 13    # Every Vagrant development environment requires a box. You can search for
 14    # boxes at https://vagrantcloud.com/search.
 15    config.vm.box = "base"
 16
 17    # Disable automatic box update checking. If you disable this, then
 18    # boxes will only be checked for updates when the user runs
 19    # `vagrant box outdated`. This is not recommended.
 20    # config.vm.box_check_update = false
 21
 22    # Create a forwarded port mapping which allows access to a specific port
 23    # within the machine from a port on the host machine. In the example below,
 24    # accessing "localhost:8080" will access port 80 on the guest machine.
 25    # NOTE: This will enable public access to the opened port
 26    # config.vm.network "forwarded port", guest: 80, host: 8080
```
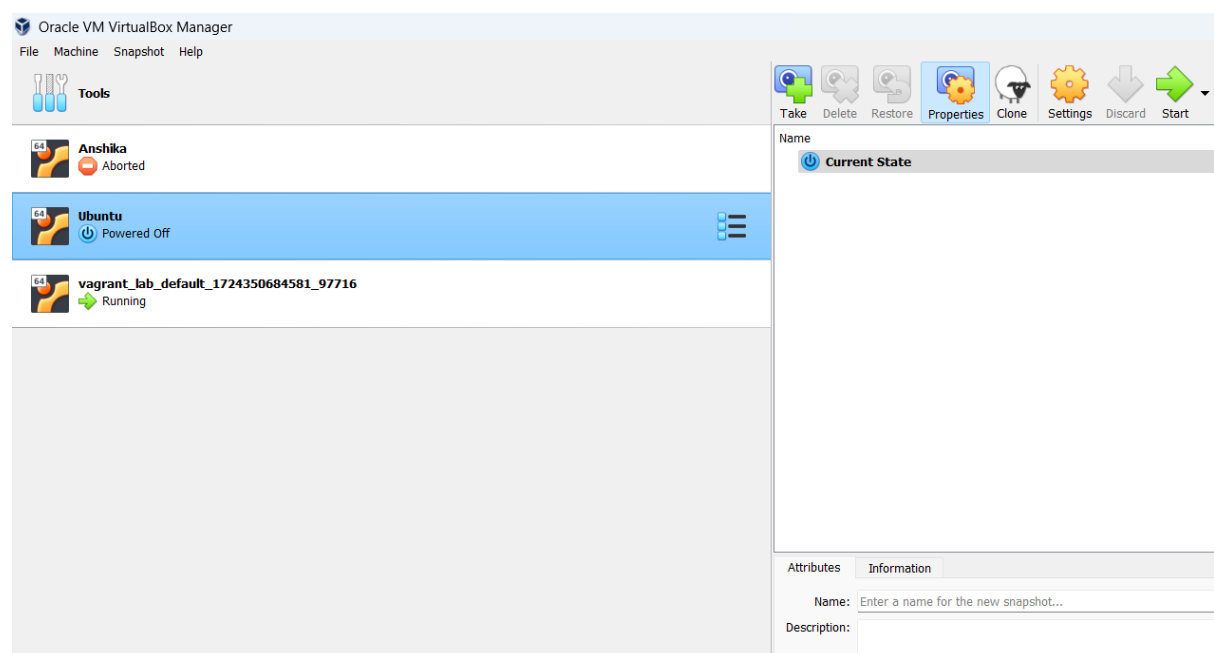
Modified file

```
Vagrantfile ×

D: > Academics > Docker > vagrant_lab > 🔴 Vagrantfile
  1  # -*- mode: ruby -*-
  2  # vi: set ft=ruby :
  3
  4  # All Vagrant configuration is done below. The "2" in Vagrant.configure
  5  # configures the configuration version (we support older styles for
  6  # backwards compatibility). Please don't change it unless you know what
  7  # you're doing.
  8  Vagrant.configure("2") do |config|
  9    # The most common configuration options are documented and commented below.
 10    # For a complete reference, please see the online documentation at
 11    # https://docs.vagrantup.com.
 12
 13    # Every Vagrant development environment requires a box. You can search for
 14    # boxes at https://vagrantcloud.com/search.
 15    config.vm.box = "ubuntu/trusty64"
 16
 17    # Disable automatic box update checking. If you disable this, then
 18    # boxes will only be checked for updates when the user runs
 19    # `vagrant box outdated`. This is not recommended.
 20    # config.vm.box_check_update = false
 21
 22    # Create a forwarded port mapping which allows access to a specific port
 23    # within the machine from a port on the host machine. In the example below,
 24    # accessing "localhost:8080" will access port 80 on the guest machine.
 25    # NOTE: This will enable public access to the opened port
 26    # config.vm.network "forwarded port", guest: 80, host: 8080
```

5. Launching and Managing the VM:

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

vagrant up

6. SSH into the VM

Use the following command to connect into running VM using SSH

vagrant ssh

Exit the SSH session by typing exit.

7. Stopping VM

Stop the VM with the following command

vagrant halt