Introduction

Vagrant is a tool for building and managing virtual machine environments in a single workflow. With an easy-to-use workflow and focus on automation, Vagrant lowers development environment setup time, increases production parity, and makes the "works on my machine" excuse a relic of the past.

Vagrant, an open-source software product for **building and maintaining portable virtual development environments.** **Written in: Ruby**

The Hashicorp Repository Contains More Than 10,000 Boxes !

Alternatives: **Docker, CLI Tools, Terraform**

Why Vagrant ?

Every developer has faced problems when it comes to setting up a development environment. Usually the environment behaves as it should on one machine, while on another machine it behaves differently or does not function at all.

Vagrant changes the way how developers setup their work environments and maintain them. Vagrant makes it possible to create a configurable portable development environment easily. These environments need to be configured in a so-called **Vagrantfile** to get completely configured.

In this **Vagrantfile**  file, the developer specifies how the environment should be set up, configured, which software should be installed and which operating system should be used. This Vagrantfile can then be distributed among other developers who just need this file in order to set up the same development environment on their own machine. Vagrant will then follow every step as defined in the provided Vagrantfile and initialise the machine.

* Vagrant encourages automation to set up your development environments using shell scripts or configuration management software.
* Vagrant allows you to work with the same operating system that is running in production, whether your physical development machine is running Linux, Mac OS X, or Windows.

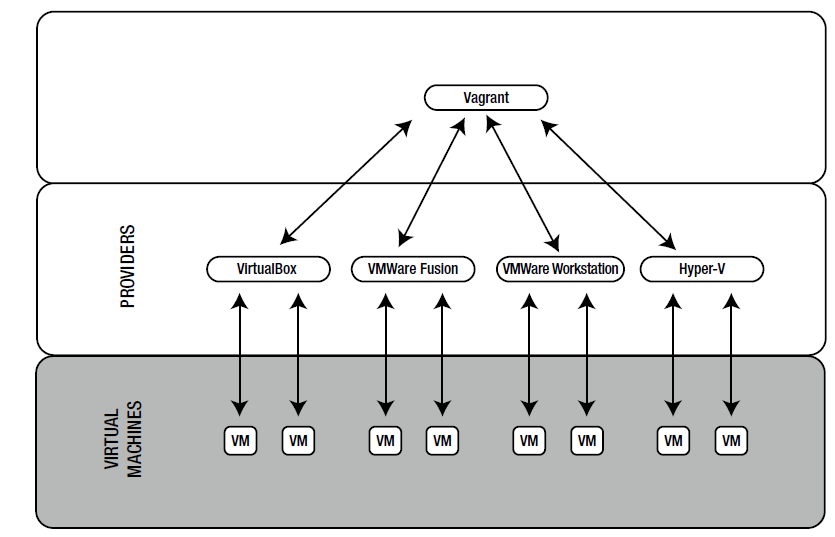
If you were to run the virtual development environment manually — without Vagrant’s help, that is — you would have to follow these steps:

1. Download the VM image.
2. Start the VM.
3. Configure the VM’s shared directories and network interfaces.
4. Maybe install some software within the VM.

With Vagrant, all these tasks (and many more) are automated. The command $ vagrant up can do the following (depending on the configuration file):

* Automatically download and install a VM, if necessary
* Boot the VM
* Configure various resources: RAM, CPUs, network connections, and shared folders
* Install additional software within the VM by using tools such as Puppet, Chef,
* Ansible, and Salt

Architrcture

Vagrant sits on top of existing and well-known virtualization solutions such as VirtualBox, VMWare Workstation, VMWare Fusion, and Hyper-V; and provides a unified and simple command-line interface to manage VMs. To work with Vagrant, you have to install at least one provider 

In Vagrant terminology, a **provider** is a software virtualization solution such as VirtualBox, VMWare Fusion, or VMWare Workstation.

Commands

$ vagrant init [url]

$ vagrant up

$ vagrant halt

$ vagrant destroy [--force]

$ vagrant reload

$ vagrant ssh

$ vagrant status

Working

**Set the Proxy on cmdline**

set http\_proxy=http://10.219.2.220:80

set https\_proxy=http://10.219.2.220:80

**Installation**

1. Download the free VirtualBox for your operating system from [the VirtualBox website](https://www.virtualbox.org/wiki/Downloads).
2. After download, just run the binary and install it.
3. Download [Vagrant](https://www.vagrantup.com/downloads.html).
4. Again, just run the binary to install it.

2.Vagrant is a command-line based tool. Once installation is complete, open a console window and create a new directory called 'vagrant\_intro’ to workwith new Vegrant baox

cd ~

mkdir vagrant\_intro

cd vagrant\_intro

3. To add a box, goto box repository: <https://app.vagrantup.com/boxes/search> Run this command:

$ vagrant box add <name>

$ vagrant box add ubuntu/trusty64

This will download the box named "hashicorp/precise64" from [HashiCorp's Vagrant Cloud box catalog](https://vagrantcloud.com/boxes/search) .

In the above command, you will notice that boxes are namespaced. Boxes are broken down into two parts - the **username and the box name** - separated by a slash

4.To create an environment, inside your folder run init command, it will create **‘vagrantfile’**

vagrant init ubuntu/trusty64

It will downloads the Ubuntu Box into our local mechine, we can check the downloaded Box by going this location **Windows** : C:\Users\<Username>\.vagrant.d\boxes **Linux/Mac: ~/.vagrant.d/boxes**

The generated 'Vagrantfile' is a Ruby file that controls your [one or more] virtual machines.

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.

5.Start the Environmet

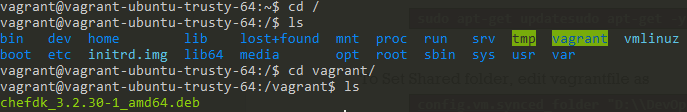
$ vagrant up

6. connect to Environment

$ vagrant ssh

To Set Shared folder, edit vagrantfile as

config.vm.synced\_folder "D:\\DevOps\\Instl\\VagrantBoxes\\SyncFolder", "/vagrant"

we named our syncd folder as “vagrant”, you can find the files in Syncfolder by going /vagrant/

Ubuntu Install

**1.Install**

sudo apt-get install vagrant

2.Complete log

satya@satya-Aspire-E5-523:~/Desktop/DevOps/vagrant$ mkdir centos

satya@satya-Aspire-E5-523:~/Desktop/DevOps/vagrant$ cd centos/

satya@satya-Aspire-E5-523:~/Desktop/DevOps/vagrant/centos$ vagrant box add centos/7

==> box: Loading metadata for box &apos;centos/7&apos;

box: URL: https://vagrantcloud.com/centos/7

This box can work with multiple providers! The providers that it

can work with are listed below. Please review the list and choose

the provider you will be working with.

1) hyperv

2) libvirt

3) virtualbox

4) vmware\_desktop

Enter your choice: 3

==> box: Adding box &apos;centos/7&apos; (v1804.02) for provider: virtualbox

box: Downloading: https://vagrantcloud.com/centos/boxes/7/versions/1804.02/providers/virtualbox.box

==> box: Successfully added box &apos;centos/7&apos; (v1804.02) for &apos;virtualbox&apos;!

satya@satya-Aspire-E5-523:~/Desktop/DevOps/vagrant/centos$ ls

satya@satya-Aspire-E5-523:~/Desktop/DevOps/vagrant/centos$ vagrant init centos/7A `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.

satya@satya-Aspire-E5-523:~/Desktop/DevOps/vagrant/centos$ vagrant up

Bringing machine &apos;default&apos; up with &apos;virtualbox&apos; provider...

==> default: Importing base box &apos;centos/7&apos;...

==> default: Matching MAC address for NAT networking...

==> default: Checking if box &apos;centos/7&apos; is up to date...

==> default: Setting the name of the VM: centos\_default\_1537980466397\_98016

==> 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: Remote connection disconnect. 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&apos;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: No guest additions were detected on the base box for this VM! Guest

default: additions are required for forwarded ports, shared folders, host only

default: networking, and more. If SSH fails on this machine, please install

default: the guest additions and repackage the box to continue.

default:

default: This is not an error message; everything may continue to work properly,

default: in which case you may ignore this message.

==> default: Rsyncing folder: /home/satya/Desktop/DevOps/vagrant/centos/ => /vagrant

satya@satya-Aspire-E5-523:~/Desktop/DevOps/vagrant/centos$

satya@satya-Aspire-E5-523:~/Desktop/DevOps/vagrant/centos$ vagrant ssh

CentOS Default username/passwd is **root/vagrant**

How to Download Vagrant Box Manually

Save: box.sh, run on terminal

/\* this is the box (and the version) that we want to download from: https://app.vagrantup.com/debian/boxes/jessie64 \*/

wget https://app.vagrantup.com/debian/boxes/jessie64/versions/8.9.0/providers/virtualbox.box -O debian-jessie64-8.9.0.box

/\* add the box to vagrant \*/

vagrant box add debian/jessie64 debian-jessie64-8.9.0.box

/\* update box version \*/

cd ~/.vagrant.d/boxes/debian-VAGRANTSLASH-jessie64/

mv 0 8.9.0

/\* create metadata\_url file \*/

echo -n "https://app.vagrantup.com/debian/boxes/jessie64" > metadata\_url

/\* show vagrant boxes \*/

vagrant box list

Public static ip for vagrant box

Since release 1.3.0: we can set Static IP on public networks.

you just have to put this configuration in your Vagrantfile (documentation):

config.vm.network "public\_network", ip: "192.168.0.200