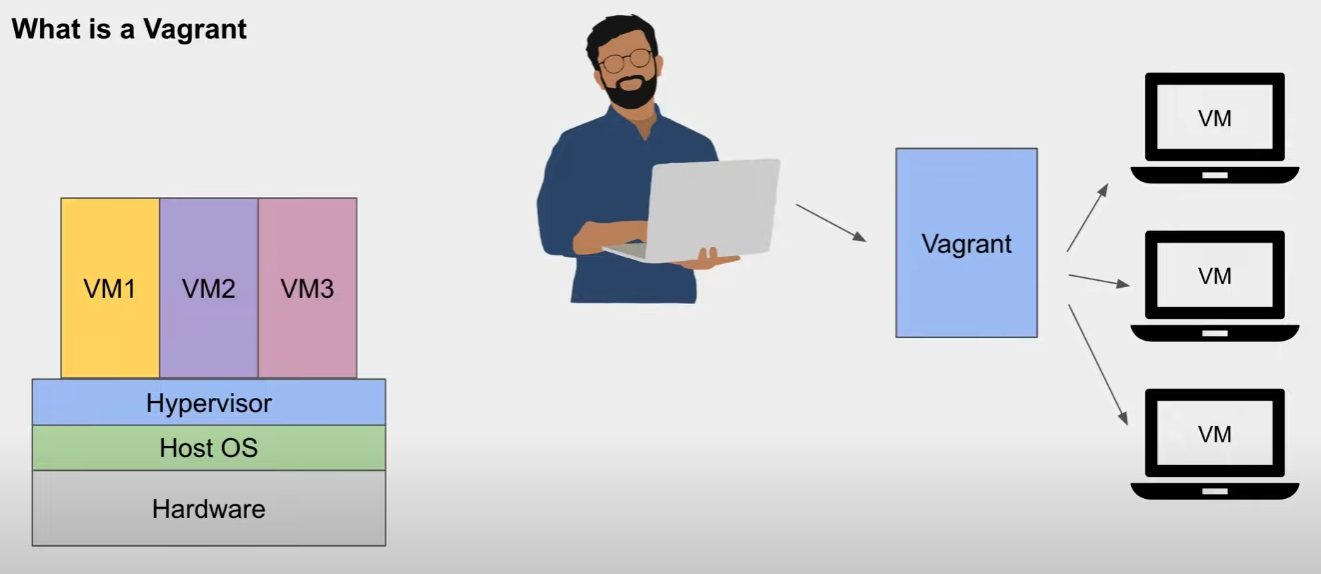
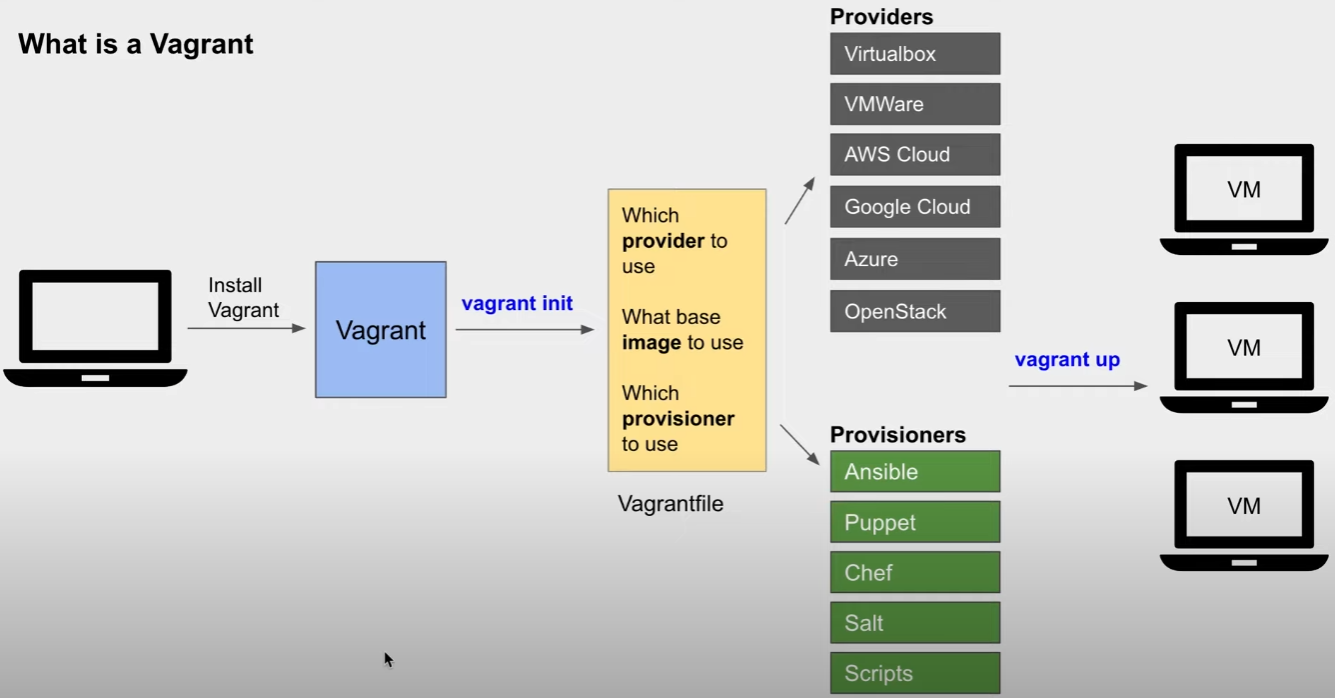
**Vagrant**  
  
**What is Vagrant?**  
It is an open-source tool written in Ruby, helps us to **automate the creation and management of Virtual Machines**.

we can specify the configuration of a virtual machine in a simple configuration file[Vagrantfile] and then create the VM with using simple vagrant command vagrant up .

It can control VMs from command line using **Vagrant commands** [Scroll down for all the Vagrant command].



**How Vagrant Works?**



**Provision**: Setting up and configuring a VM to make it ready for use. It can include installing required software, libraries, setting networks etc.

**Vagrantfile**: A Vagrantfile is a configuration file used by Vagrant to define and provision virtual machines. It's a script that tells Vagrant how to create and configure a virtual machine, and what software to install on it.

*The Vagrantfile is written in Ruby. You don't need to be an expert in Ruby to use Vagrant - most of the time you can simply copy and paste example Vagrantfiles or modify existing ones to suit your needs*  
Reference Link:

|  |
| --- |
| [1 | Everything a Beginner needs to know about Vagrant](https://www.youtube.com/watch?v=czMCO1w-xQU&t=233s&ab_channel=AutomationStepbyStep) |

**Vagrant - Getting Started | Install ＞ Setup ＞ Use**

* Step 1 - Install Vagrant <https://www.vagrantup.com/downloads>

Check vagrant is installed in command prompt using below command

|  |
| --- |
| vagrant --version |

* Step 2 - Select a VM Provider. Vagrant has direct support for VirtualBox, Hyper-V, Docker Install VirtualBox <https://www.virtualbox.org/wiki/Downloads>
* Step 3 - Create a new folder for Vagrant project
* Step 4 - On terminal or command line navigate to the folder and initiate vagrant project

|  |
| --- |
| vagrant init |

This will create a new Vagrantfile in the folder

Vagrantfile is a configuration file that defines the settings for your virtual machine

* Step 5 - Choose a box to use <https://app.vagrantup.com/boxes/search>

A box is a pre-configured virtual machine image that you can use as a starting point for your virtual machine

* Step 6 - Add configuration of the box in Vagrantfile

For example, you could use the "ubuntu/bionic64" box by adding the following line to your Vagrantfile:

config.vm.box = "ubuntu/bionic64"

We can also directly add configuration for the virtual machine using the following commands

vagrant init centos/7 (if vagrantfile does not already exists)

vagrant box add centos/7 (will add box to vagrant, but will not create Vagrant file)

* Step 7 - Start virtual machine using command vagrant up

This will create a new virtual machine using the box you selected and start it. The first time, Vagrant will download the box from the internet

* Step 8 - SSH into the virtual machine vagrant ssh

Reference Link:

|  |
| --- |
| [2 | Getting started with Vagrant Setup for beginners](https://www.youtube.com/watch?v=7DLfOGt8YvA&t=368s&ab_channel=AutomationStepbyStep) |

**Pre-Requisites Tools:**

|  |
| --- |
| 1. Vagrant  2. VS code  3. Oracle VM |

[Reference Note : <https://www.youtube.com/watch?v=gL37hU-U88E&ab_channel=becloudready>]

**Steps to Set Up Vagrant for Project**:  
**Create a Vagrant Directory**

**Steps to Create a VM using vagrant file**

|  |  |  |  |
| --- | --- | --- | --- |
| **Step 1**: Initialize the Vagrant Directory   |  | | --- | | vagrant init |   **Step 2**: Replace the Vagrant File  **Step 3**: Install the vagrant-hostmanager   |  | | --- | | $ vagrant plugin install vagrant-hostmanager |   **Step 4**: Now execute the Vagrantfile   |  | | --- | | vagrant up | |

**Vagrant Box Commands:**  
1. Adds a box to your local box repository

|  |
| --- |
| vagrant box add  Ex: vagrant box add ubuntu/focal64 |

2. Lists all boxes in your local box repository

|  |
| --- |
| vagrant box list |

3. Checks if any boxes in your local box repository are outdated

|  |
| --- |
| vagrant box outdated |

4. Updates a box to a new version

|  |
| --- |
| vagrant box update  Ex: vagrant box update ubuntu/focal64 |

5. Repackages a box with a new name and metadata

|  |
| --- |
| vagrant box repackage  Ex: vagrant box repackage ubuntu/focal64 --name my-new-box |

6. Removes outdated boxes from your local box repository

|  |
| --- |
| vagrant box prune |

7. Removes a box from your local box repository

|  |
| --- |
| vagrant box remove  Ex: vagrant box remove ubuntu/focal64 |

8. Location of VM boxes

|  |
| --- |
| Mac OS X and Linux: ~/.vagrant.d/boxes  Windows: C:/Users/USERNAME/.vagrant.d/boxes |

**Vagrant Commands:**  
1. To initialize the Directory

|  |
| --- |
| vagrant init Ex: vagrant init ubuntu/bionic64 |

2. To run the Vagrant file [Make sure that you are at the same path where vagrant file is to execute the command]

|  |
| --- |
| vagrant up |

3. To connect the server

|  |
| --- |
| vagrant ssh <Hostname or IP> Ex: vagrant ssh app01 |

4. Outputs OpenSSH valid configuration to connect to the VMs via SSH

|  |
| --- |
| vagrant ssh-config Ex: |

5. Stops the guest machine

|  |
| --- |
| vagrant halt |

6. Suspends the guest machine

|  |
| --- |
| vagrant suspend |

7. Resumes a suspended guest machine

|  |
| --- |
| vagrant resume |

8. Reloads the guest machine by restarting it

|  |
| --- |
| vagrant reload |

9. Stops and deletes all traces of the guest machine

|  |
| --- |
| vagrant destroy |

10. Shows the status of the current Vagrant environment

|  |
| --- |
| vagrant status |

11. Packages a running virtual environment into a reusable box

|  |
| --- |
| vagrant package  Ex: vagrant package --output mybox.box |

12. Runs any configured provisioners against the running VM.

|  |
| --- |
| vagrant provision |

13. Installs a Vagrant plugin

|  |
| --- |
| vagrant plugin install Ex: vagrant plugin install myplugin |

14. Lists all installed Vagrant plugins

|  |
| --- |
| vagrant plugin list |

15. Uninstalls a Vagrant plugin

|  |
| --- |
| --help Ex: vagrant --help  vagrant init --help |

**Vagrant Tips:**

1. To get help for any vagrant command

|  |
| --- |
| vagrant plugin uninstall Ex: vagrant plugin uninstall myplugin |

2. If using Virtual Box

|  |
| --- |
| vboxmanage list vms  vboxmanage list runningvms |

**Vagrant Plugins:**

Plugins are add-ons to enhance functionality of a tool  
Available Plugins: <https://github.com/hashicorp/vagrant/wiki/Available-Vagrant-Plugins>

1. to see subcommands and options

|  |
| --- |
| vagrant plugin –help |

2. to list all installed plugins

|  |
| --- |
| vagrant plugin list |

3. to install a plugin

|  |
| --- |
| vagrant plugin install <plugin name> Ex: vagrant plugin install vbinfo |

4. to use the plugin

|  |
| --- |
| vagrant vbinfo |

5. to update a plugin

|  |
| --- |
| vagrant plugin update  vagrant plugin update <plugin name> Ex: vagrant plugin update vbinfo |

6. to repair installed plugins in case of error

|  |
| --- |
| vagrant plugin repair |

7. to associate the plugin with a license file

|  |
| --- |
| vagrant plugin license [license\_file |

8. to remove plugin

|  |
| --- |
| vagrant plugin uninstall <plugin name> Ex: vagrant plugin uninstall vbinfo |

9. to delete all plugins

|  |
| --- |
| vagrant plugin expunge |

10. to reinstall all expunged plugins

|  |
| --- |
| vagrant plugin expunge --reinstall |

**Plugins location on host system:**

Win: C:\Users\＜username＞\.vagrant.d\

Mac: ~/.vagrant.d/

Linux: ~/.vagrant.d/plugins /opt/vagrant/plugins /usr/share/vagrant/plugins  
  
**Vagrant Manuals:**

|  |
| --- |
| # -\*- 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 = "geerlingguy/centos7"  # 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  # Create a forwarded port mapping which allows access to a specific port  # within the machine from a port on the host machine. In the example below,  # accessing "localhost:8080" will access port 80 on the guest machine.  # NOTE: This will enable public access to the opened port  # config.vm.network "forwarded\_port", guest: 80, host: 8080  # Create a forwarded port mapping which allows access to a specific port  # within the machine from a port on the host machine and only allow access  # via 127.0.0.1 to disable public access  # config.vm.network "forwarded\_port", guest: 80, host: 8080, host\_ip: "127.0.0.1"  # Create a private network, which allows host-only access to the machine  # using a specific IP.  # config.vm.network "private\_network", ip: "192.168.33.10"  # Create a public network, which generally matched to bridged network.  # Bridged networks make the machine appear as another physical device on  # your network.  # config.vm.network "public\_network"  # Share an additional folder to the guest VM. The first argument is  # the path on the host to the actual folder. The second argument is  # the path on the guest to mount the folder. And the optional third  # argument is a set of non-required options.  # config.vm.synced\_folder "../data", "/vagrant\_data"  # Disable the default share of the current code directory. Doing this  # provides improved isolation between the vagrant box and your host  # by making sure your Vagrantfile isn't accessable to the vagrant box.  # If you use this you may want to enable additional shared subfolders as  # shown above.  # config.vm.synced\_folder ".", "/vagrant", disabled: true  # Provider-specific configuration so you can fine-tune various  # backing providers for Vagrant. These expose provider-specific options.  # Example for VirtualBox:  #  # config.vm.provider "virtualbox" do |vb|  # # Display the VirtualBox GUI when booting the machine  # vb.gui = true  #  # # Customize the amount of memory on the VM:  # vb.memory = "1024"  # end  #  # View the documentation for the provider you are using for more  # information on available options.  # Enable provisioning with a shell script. Additional provisioners such as  # Ansible, Chef, Docker, Puppet and Salt are also available. Please see the  # documentation for more information about their specific syntax and use.  # config.vm.provision "shell", inline: <<-SHELL  # apt-get update  # apt-get install -y apache2  # SHELL  end |