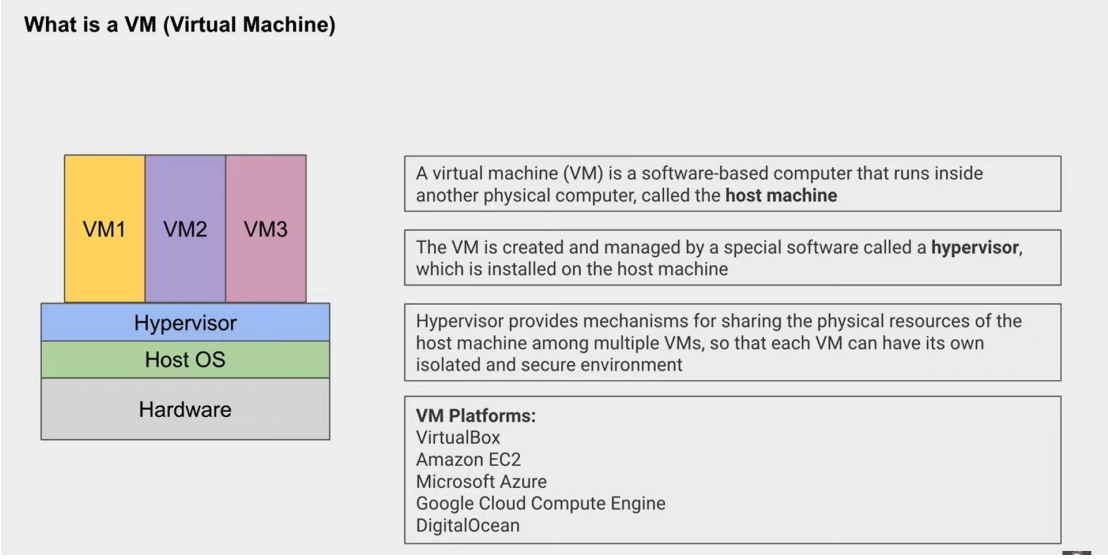
Vagrant

What is a VM (Virtual Machine):

A virtual machine (VM) is a software-based computer that runs inside another physical computer, called the host machine

The VM is created and managed by a special software called a hypervisor, which is installed on the host machine

Hypervisor provides mechanisms for sharing the physical resources of the host machine among multiple VMs, so that each VM can have its own isolated and secure environment



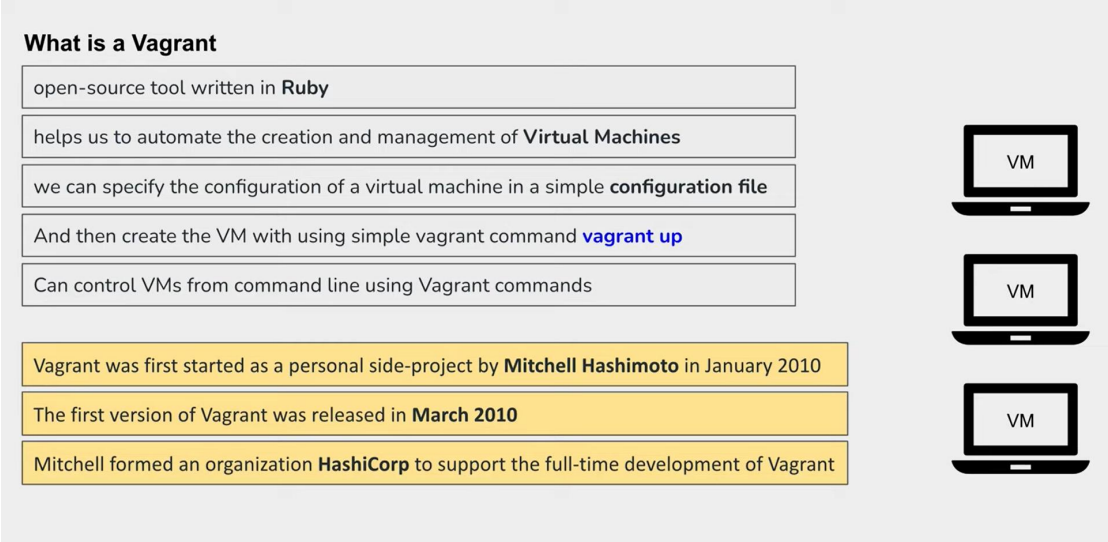
VM Platforms:

VirtualBox
Amazon EC2
Microsoft Azure
Google Cloud Compute Engine
DigitalOcean

What is Vagrant:

- 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
- And then create the VM with using simple vagrant command vagrant up

- Can control VMs from command line using Vagrant commands
- Vagrant was first started as a personal side-project by Mitchell Hashimoto in January 2010
- The first version of Vagrant was released in March 2010
- Mitchell formed an organization HashiCorp to support the full-time development of Vagrant

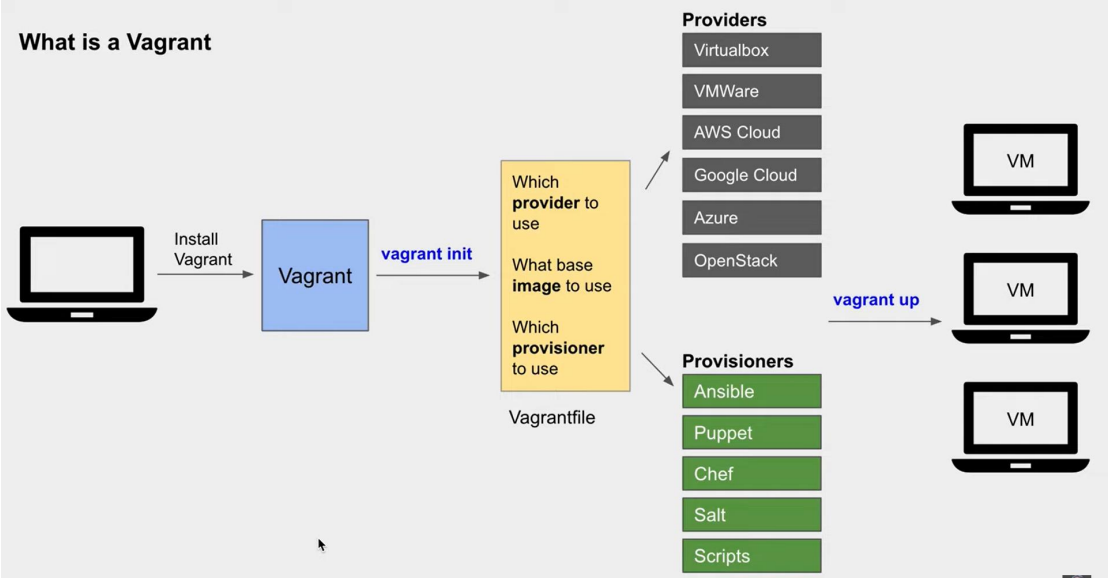


Terms to Know

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

Vagrant Setup



Vagrant - Getting Started | Install > Setup > Use

Step 1 - Install Vagrant https://www.vagrantup.com/downloads Check vagrant is installed vagrant --version

Step 2 - Select a VM Provider. Vagrant has direct support for VirtualBox, Hyper-V, Docker
Install VirtualBox https://www.virtualbox.org/wiki/Downl...

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

Vagrant Box - 7 Commands

vagrant box add
Adds a box to your local box repository
vagrant box add ubuntu/focal64

vagrant box list
Lists all boxes in your local box repository
vagrant box list

vagrant box outdated
Checks if any boxes in your local box repository are outdated
vagrant box outdated

vagrant box update
Updates a box to a new version
vagrant box update ubuntu/focal64

vagrant box repackage
Repackages a box with a new name and metadata
vagrant box repackage ubuntu/focal64 --name my-new-box

vagrant box prune
Removes outdated boxes from your local box repository
vagrant box prune

vagrant box remove
Removes a box from your local box repository
vagrant box remove ubuntu/focal64

Location of VM boxes
Mac OS X and Linux: ~/.vagrant.d/boxes
Windows: C:/Users/USERNAME/.vagrant.d/boxes

Vagrant Commands

vagrant init
Initializes a new Vagrant environment by creating a Vagrantfile
vagrant init centos/7

vagrant up
Creates and configures the guest machine

vagrant ssh
Logs in to the guest machine via SSH

vagrant ssh-config
Outputs OpenSSH valid configuration to connect to the VMs via SSH

vagrant halt
Stops the guest machine

vagrant suspend
Suspends the guest machine

vagrant resume
Resumes a suspended guest machine

vagrant reload
Reloads the guest machine by restarting it

vagrant destroy
Stops and deletes all traces of the guest machine

vagrant status
Shows the status of the current Vagrant environment

vagrant package
Packages a running virtual environment into a reusable box
vagrant package --output mybox.box

vagrant provision
Runs any configured provisioners against the running VM.

vagrant plugin install
Installs a Vagrant plugin
vagrant plugin install myplugin

vagrant plugin list
Lists all installed Vagrant plugins

vagrant plugin uninstall
Uninstalls a Vagrant plugin
vagrant plugin uninstall myplugin

Useful TIPS

--help To get help for any Vagrant command e.g. vagrant --help or vagrant init --help
vboxmanage list vms If using Virtualbox
vboxmanage list runningvms If using Virtualbox

References:
Vagrant - https://developer.hashicorp.com/vagra...
VirtualBox - https://www.virtualbox.org/wiki/Downl...
Vagrant Boxes Search - https://app.vagrantup.com/boxes/search

Example Code for 1 VM:

```
config.vm.box = "ubuntu/bionic64"

  config.vm.provision "shell", inline: <<-SHELL
    sudo apt-get update
    sudo apt-get install apache2 -y
  SHELL
```

Example Code for Multi VM:

```
#App server
config.vm.define "app" do |app|
config.vm.provision :shell, path: "configweb.sh"
app.vm.hostname = "appserver.srg.com"
app.vm.box = "centos/8"
app.vm.network :private_network, ip: "10.0.0.101"
end

#DB server
config.vm.define "db" do |db|
db.vm.hostname = "dbserver.srg.com"
db.vm.box = "centos/8"
db.vm.network :private_network, ip: "10.0.0.102"
end
```