

10. Vagrant ssh

This will SSH into a running Vagrant machine and give you access to a shell.

→ Once the vm is up login to the vm by giving “vagrant ssh” command.

```
vagrant@localhost:~$ Virtualbox on your host claims: 4.3.10
VirtualboxService inside the vm claims: 5.1.18
Going on, assuming VirtualboxService is correct...
Restarting VM to apply changes...
==> default: Attempting graceful shutdown of VM...
==> 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: Machine booted and ready!
==> default: Checking for guest additions in VM...
==> default: Mounting shared folders...
default: /vagrant => C:/vagrant-vms/centos-vm

Eswar@LAPTOP-5SKPP7K3 MINGW64 /c/vagrant-vms/centos-vm
$ pwd
/c/vagrant-vms/centos-vm

Eswar@LAPTOP-5SKPP7K3 MINGW64 /c/vagrant-vms/centos-vm
$ vagrant ssh
Welcome to your Vagrant-built virtual machine.
[vagrant@localhost ~]$
```

→ Follow the same procedure in ubuntu-vm folder, Open another git bash prompt.

- 1) cd /c/vagrant-vms/ubuntu-vm
- 2) vagrant init ubuntu/trusty64
- 3) vagrant up

```
MINGW64/c/vagrant-vms/ubuntu-vm
Eswar@LAPTOP-5SKPP7K3 MINGW64 ~
$ cd /c/vagrant-vms/ubuntu-vm/

Eswar@LAPTOP-5SKPP7K3 MINGW64 /c/vagrant-vms/ubuntu-vm
$ vagrant init ubuntu/trusty64
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.

Eswar@LAPTOP-5SKPP7K3 MINGW64 /c/vagrant-vms/ubuntu-vm
$ vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Importing base box 'ubuntu/trusty64'...
```

```
vagrant@vagrant-ubuntu-trusty-64: ~
* Documentation: https://help.ubuntu.com/
System information as of Tue Apr  4 04:38:41 UTC 2017
System load: 0.5          Processes:      79
Usage of /: 3.6% of 39.34GB  Users Logged in:   0
Memory usage: 25%          IP address for eth0: 10.0.2.15
Swap usage:  0%
Graph this data and manage this system at:
  https://landscape.canonical.com/
Get cloud support with Ubuntu Advantage Cloud Guest:
  http://www.ubuntu.com/business/services/cloud
0 packages can be updated.
0 updates are security updates.

New release '16.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

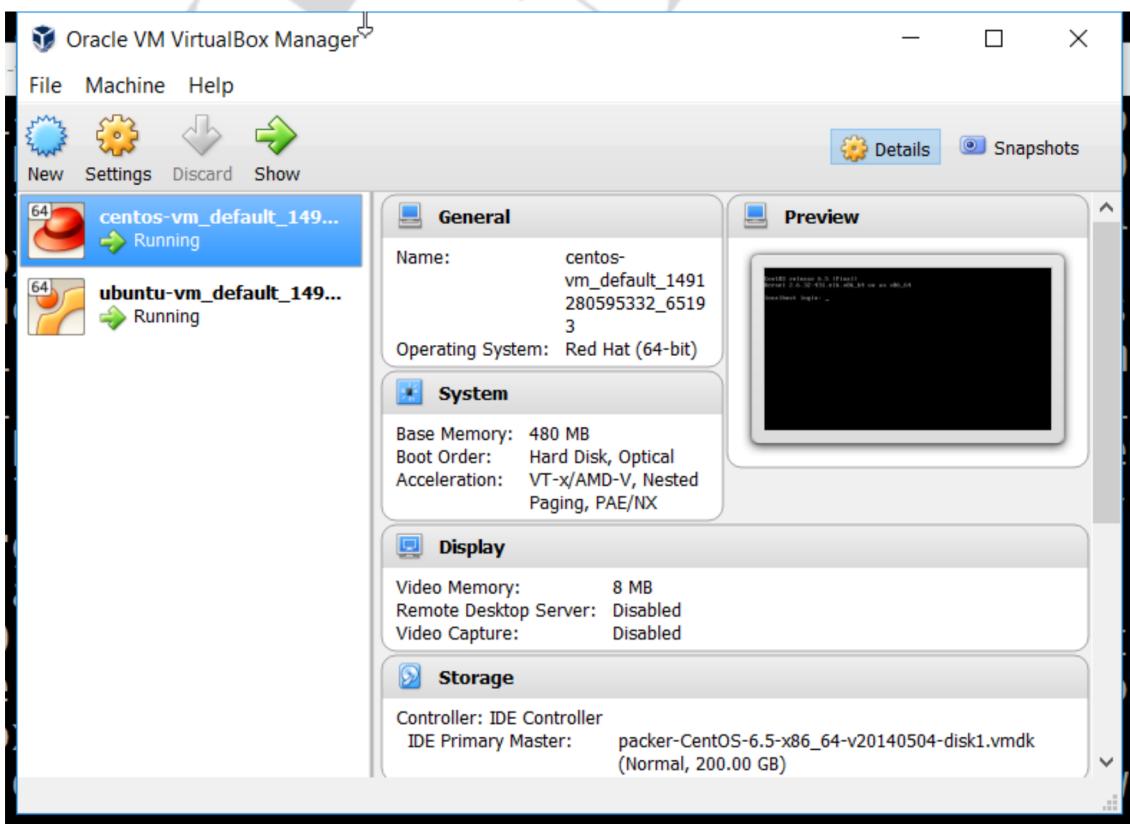
vagrant@vagrant-ubuntu-trusty-64:~$ |
```

11.Verification.

We are using the default hypervisor for vagrant which is Virtualbox. We will open up our virtualbox interface and check our running VM which we just created. This is not a required step but just to show you the power of vagrant we insist you to take a look.

Do not make any changes to a vm from virtualbox interface and only manage it with vagrant.

→ Open virtualbox software and you will see two vm's running there.



12. Teardown.

We now have a fully functional virtual machine which we can use for our any DevOps, Ops or development task. Once we are done with our experiments we may need to suspend, shutdown or delete the vm. With Vagrant, you *suspend*, *halt*, or *destroy* the guest machine. Each of these options have pros and cons. Choose the method that works best for you.

Halting.

→ Hit “exit” command to come out of the vm shell => hit “vagrant halt” to shut it down.

```
MINGW64:/c/vagrant-vms/ubuntu-vm
Usage of /: 3.6% of 39.34GB   Users logged in: 0
Memory usage: 25%           IP address for eth0: 10.0.2.15
Swap usage: 0%
Graph this data and manage this system at:
https://landscape.canonical.com/
Get cloud support with Ubuntu Advantage Cloud Guest:
http://www.ubuntu.com/business/services/cloud
0 packages can be updated.
0 updates are security updates.
>New release '16.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

vagrant@vagrant-ubuntu-trusty-64:~$ exit
logout
Connection to 127.0.0.1 closed.          I
Eswar@LAPTOP-5SKPP7K3 MINGW64 /c/vagrant-vms/ubuntu-vm
$ vagrant halt
```

```
MINGW64:/c/vagrant-vms/centos-vm
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: Machine booted and ready!
==> default: Checking for guest additions in VM...
==> default: Mounting shared folders...
default: /vagrant => C:/vagrant-vms/centos-vm

Eswar@LAPTOP-5SKPP7K3 MINGW64 /c/vagrant-vms/centos-vm
$ pwd
/c/vagrant-vms/centos-vm

Eswar@LAPTOP-5SKPP7K3 MINGW64 /c/vagrant-vms/centos-vm
$ vagrant ssh
Welcome to your Vagrant-built virtual machine.
[vagrant@localhost ~]$ exit
Logout
Connection to 127.0.0.1 closed.

Eswar@LAPTOP-5SKPP7K3 MINGW64 /c/vagrant-vms/centos-vm
$ vagrant halt
==> default: Attempting graceful shutdown of VM...
```

Suspending the virtual machine by calling `vagrant suspend` will save the current running state of the machine and stop it. When you are ready to begin working again, just run `vagrant up`, and it will be resumed from where you left off. The main benefit of this method is that it is super-fast, usually taking only 5 to 10 seconds to stop and start your work. The downside is that the virtual machine still eats up your disk space, and requires even more disk space to store all the state of the virtual machine RAM on disk.

Halting the virtual machine by calling `vagrant halt` will gracefully shut down the guest operating system and power down the guest machine. You can use `vagrant up` when you are ready to boot it again. The benefit of this method is that it will cleanly shut down your machine, preserving the contents of disk, and allowing it to be cleanly started again. The downside is that it'll take some extra time to start from a cold boot, and the guest machine still consumes disk space.

Destroying the virtual machine by calling `vagrant destroy` will remove all traces of the guest machine from your system. It'll stop the guest machine, power it down, and remove all of the guest hard disks. Again, when you are ready to work again, just issue `vagrant up`. The benefit of this is that *no cruft* is left on your machine. The disk space and RAM consumed by the guest machine is reclaimed and your host machine is left clean. The downside is that `vagrant up` to get working again will take some extra time since it has to reimport the machine and re-provision it.

13. Vagrant box commands

In above exercise, we have seen mentioning the box name in the vagrantfile lets us pull-down the box from vagrant cloud and spin a vm for us when executing vagrant up command.

But sometimes you may need to download the box in advance, may keep record of all the downloaded box or may even delete a box.

We can use vagrant box command to manage our boxes.

Listing local boxes.

- ❖ Enter vagrant box list command to see the downloaded box in your system.

```
imran@DevOps:~$ vagrant box list
aerospike/centos-6.5          (virtualbox, 3.11.0.2)
bento/centos-6.7              (virtualbox, 2.2.7)
debian/contrib-jessie64        (virtualbox, 8.6.2)
debian/jessie64                (virtualbox, 8.6.1)
designerror/windows-7          (virtualbox, 1.0)
ferventcoder/win2008r2-x64-nocm (virtualbox, 0.1.0)
hashicorp/precise64            (virtualbox, 1.1.0)
nrel/CentOS-6.5-x86_64         (virtualbox, 1.2.0)
ubuntu/trusty64                (virtualbox, 20161004.0.0)
ubuntu/trusty64                (virtualbox, 20161020.0.3)
ubuntu/xenial64                (virtualbox, 20170119.1.0)
```

Adding a box

We can also download a box and can be used later in our vagrantfile.

Boxes in vagrant cloud and different providers like virtualbox, libvirt, vmware desktop etc.

We may also need to select the provider of the box.

```
imran@DevOps:~$ vagrant box add centos/7
==> box: Loading metadata for box 'centos/7'
    box: URL: https://atlas.hashicorp.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) libvirt
2) virtualbox
3) vmware_desktop

Enter your choice: 2
==> box: Adding box 'centos/7' (v1704.01) for provider: virtualbox
    box: Downloading: https://atlas.hashicorp.com/centos/boxes/7/versions/1
providers/virtualbox.box
    box: Progress: 3% (Rate: 2696k/s, Estimated time remaining: 0:04:01)
```

Other box options.

Visualpath Training & Consulting.

Flat no: 205, Nilgiri Block, Aditya Enclave, Ameerpet, Hyderabad, Phone No: - +91-970 445 5959, 961 824 5689 E-Mail ID : online.visualpath@gmail.com, Website : www.visualpath.in.

```

imran@DevOps:~$ vagrant box
Usage: vagrant box <subcommand> [<args>]

Available subcommands:
  add
  list
  outdated
  remove
  repackage
  update

For help on any individual subcommand run `vagrant box <subcommand> -h`
```

14. Useful Vagranfile settings.

We may need to modify settings of the vm created by vagrant through Vagrantfile.

Few of the generally used settings are mentioned below. To know list of all the settings please refer vagrant documentation.

Vagrant reload

After changing any vm setting in the vagrantfile, you may need to reload its changes to an already running vm.

Vagrant reload command will reboot the vm and load all the latest changes, you may need to provide a --provision option to force the provisioners to run.

Memory/RAM modification.

→ Open your vagrantfile and find config.vm.provider setting.

```

# 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
#
```

→ Uncomment below mentioned three lines of code and change vb.memory as per requirement.

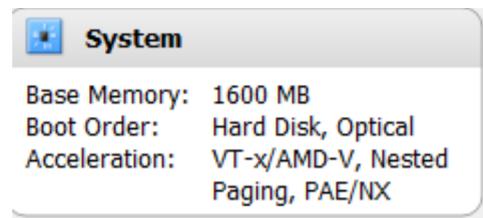
```

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 = "1600"
end
```

→ Run vagrant up if the vm is halted or vagrant reload if the vm is already running. Verify the Base memory setting from virtualbox.

Visualpath Training & Consulting.

Flat no: 205, Nilgiri Block, Aditya Enclave, Ameerpet, Hyderabad, Phone No: +91-970 445 5959, 961 824 5689 E-Mail ID : online.visualpath@gmail.com, Website : www.visualpath.in.



→ Login to the vm and run “free -m” command to check total memory size of a linux vm.

```
Eswar@LAPTOP-5SKPP7K3 MINGW64 /c/vagrant-vm/centos-vm
$ vagrant ssh
Last login: Mon Apr 10 04:10:46 2017 from 192.168.1.8
Welcome to your Vagrant-built virtual machine.
[vagrant@localhost ~]$ free -m
              total        used        free      shared  buffers  cached
Mem:       1562         122       1439          0         7        53
-/+ buffers/cache:       61       1500
Swap:      4095          0       4095
[vagrant@localhost ~]$
```

15. Bridge Networking.

Vagrant vm will by default have NAT Interface, we can create an extra bridge interface.

Benefit of bridge interface: Our vm will get IP from Router/Gateway/DHCP/Wifi router of our network. So our VM will be able to talk to any instance in network and vice versa.

Procedure to create bridge network interface:

- Open the Vagrantfile of an instance on which you want to create bridge interface.
- Enable Public network setting in Vagrantfile by uncommenting below mentioned line.
- Uncomment config.vm.network “public_network”

```
# 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
```

- Uncomment config.vm.network “public_network”.

```
# 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
```

→ If the VM is powered off you can hit “vagrant up”.

→ If the vm is already running you need to reload vm with new settings.

```
MINGW64:/c/vagrant-vm/centos-vm
```

```
Eswar@LAPTOP-5SKPP7K3 MINGW64 /c/vagrant-vm/centos-vm  
$ pwd  
/c/vagrant-vm/centos-vm
```

```
Eswar@LAPTOP-5SKPP7K3 MINGW64 /c/vagrant-vm/centos-vm  
$ ls  
Vagrantfile
```

```
Eswar@LAPTOP-5SKPP7K3 MINGW64 /c/vagrant-vm/centos-vm  
$ vagrant reload --provision
```

→ Observe it creates one more extra interface “Adapter 2: bridged”.

```
=> default: Clearing any previously set network interfaces..  
=> default: Preparing network interfaces based on configuration  
    default: Adapter 1: nat  
    default: Adapter 2: bridged  
    default: Forwarding ports
```

→ Login to vm and check its IP, you will see eth1 as bridged interface and will get IP from your networks

```
[vagrant@localhost ~]$ ifconfig
eth0      Link encap:Ethernet HWaddr 08:00:27:15:D5:19
          inet addr:10.0.2.15 Bcast:10.0.2.255 Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe15:d519/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:829 errors:0 dropped:0 overruns:0 frame:0
          TX packets:597 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:84251 (82.2 KiB) TX bytes:75364 (73.5 KiB)

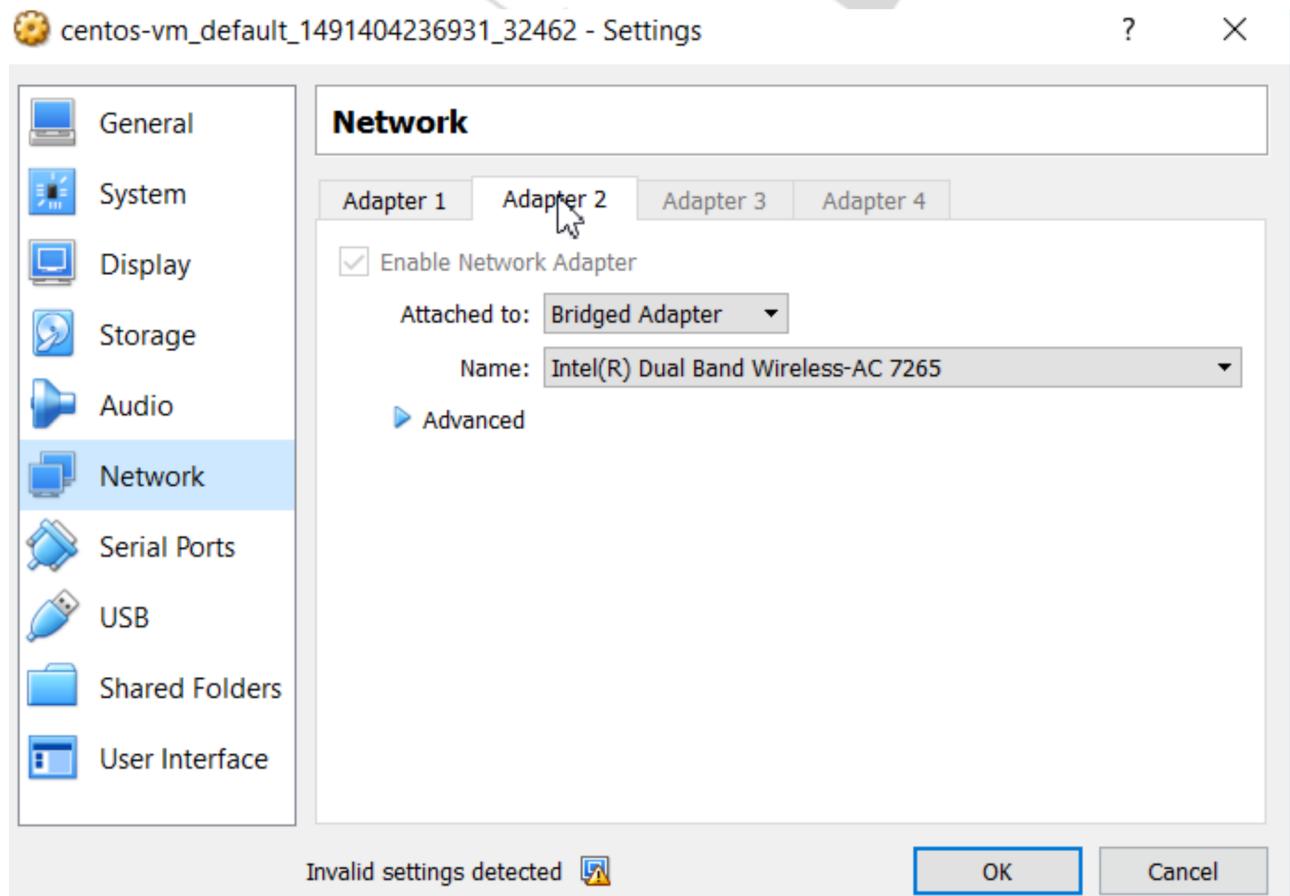
eth1      Link encap:Ethernet HWaddr 08:00:27:69:03:23
          inet addr:192.168.1.10 Bcast:192.168.1.255 Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe69:323/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:13 errors:0 dropped:0 overruns:0 frame:0
          TX packets:28 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2253 (2.2 KiB) TX bytes:2516 (2.4 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING MTU:16436 Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b) TX bytes:0 (0.0 b)
```

Router/Gateway/DHCP/Wifi router.

→ Verify from Virtualbox as well.

Select your VM => Settings => Network => Adapter 2



→ Connecting or ssh to vm with bridge IP.

Visualpath Training & Consulting.

Flat no: 205, Nilgiri Block, Aditya Enclave, Ameerpet, Hyderabad, Phone No: - +91-970 445 5959, 961 824 5689 E-Mail ID : online.visualpath@gmail.com, Website : www.visualpath.in.

Vagrant will auto create a user on all vagrant vm's.

Username => vagrant, Password => vagrant .

```
Eswar@LAPTOP-5SKPP7K3 MINGW64 /c/vagrant-vm/centos-vm
$ ssh vagrant@192.168.1.10
vagrant@192.168.1.10's password:
Last login: Mon Apr 10 04:08:51 2017 from 10.0.2.2
Welcome to your Vagrant-built virtual machine.
[vagrant@localhost ~]$ |
```

16. Provisioning

Provisioning helps execute command or script as soon as the vm comes up.

Provisioners in Vagrant allow you to automatically install software, alter configurations, and more on the machine as part of the `vagrant up` process.

This is useful since boxes typically are not built *perfectly* for your use case. Of course, if you want to just use `vagrant ssh` and install the software by hand, that works. But by using the provisioning systems built-in to Vagrant, it automates the process so that it is repeatable. Most importantly, it requires no human interaction, so you can `vagrant destroy` and `vagrant up` and have a fully ready-to-go work environment with a single command. Powerful.

Vagrant gives you multiple options for provisioning the machine, from simple shell scripts to more complex, industry-standard configuration management systems.

Setting up apache/httpd service in a centos vm.

→ Find config.vm.provision setting.

```
# Enable provisioning with a shell script. Additional provisioners such as
# Puppet, Chef, Ansible, Salt, and Docker 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
nd
```

→ Update the file with below mentioned content.

```
# Enable provisioning with a shell script. Additional provisioners such as
# Puppet, Chef, Ansible, Salt, and Docker are also available. Please see the
# documentation for more information about their specific syntax and use.
config.vm.provision "shell", inline: <<-SHELL
  yum install httpd
  service httpd start
  service iptables stop
SHELL
```

→ Reload the vm and check the provisioning.

```
=> vagrant: Verifying : apr-util-1.3.9-3.el6_0.1.x86_64 6/6
=> default: 
=> default: 
=> default: Installed:
=> default:   httpd.x86_64 0:2.2.15-59.el6.centos
=> default: 
=> default: Dependency Installed:
=> default:   apr.x86_64 0:1.3.9-5.el6_2
=> default:   apr-util.x86_64 0:1.3.9-3.el6_0.1
=> default:   apr-util-ldap.x86_64 0:1.3.9-3.el6_0.1
=> default:   httpd-tools.x86_64 0:2.2.15-59.el6.centos
=> default:   mailcap.noarch 0:2.1.31-2.el6
=> default: Complete!
=> default: Starting httpd:
=> default: httpd: Could not reliably determine the server's fully qualified domain name,
g localhost.localdomain for ServerName
=> default: [ OK ]
=> default: iptables: Setting chains to policy ACCEPT: filter
=> default: [ OK ]
=> default: iptables: Flushing firewall rules:
=> default: [ OK ]
=> default: iptables: Unloading modules:
=> default: [ OK ]
```

swar@LAPTOP-5SKPP7K3 MINGW64 /c/vagrant-vm/centos-vm

→ Login to vm and verify httpd & iptables service status.

```
$ vagrant ssh
Last login: Mon Apr 10 04:15:02 2017 from 10.0.2.2
Welcome to your Vagrant-built virtual machine.
[vagrant@localhost ~]$ sudo service httpd status
httpd (pid 2639) is running...
[vagrant@localhost ~]$ sudo service iptables status
iptables: Firewall is not running.
[vagrant@localhost ~]$
```

→ Verify apache default page by providing vm's bridge ip in browser.

If you are a member of the general public:
The fact that you are seeing this page indicates that the website you just visited is either experiencing problems or is undergoing routine maintenance.
If you would like to let the administrators of this website know that you've seen this page instead of the page you expected, you should send them e-mail. In general, mail sent to the name "webmaster" and directed to the website's domain should reach the appropriate person.
For example, if you experienced problems while visiting www.example.com, you should send e-mail to "webmaster@example.com".

If you are the website administrator:
You may now add content to the directory /var/www/htd1/. Note that until you do so, people visiting your website see this page and not your content. To prevent this page from ever being used, follow the instructions in the file /etc/httpd/conf.d/welcome.conf.
You are free to use the images below on Apache and CentOS Linux powered HTTP servers. Thanks for using Apache and CentOS!

Powered by APACHE CentOS

About CentOS:
The Community ENTerprise Operating System (CentOS) Linux is a community-supported enterprise distribution derived from sources freely provided to the public by Red Hat. As such, CentOS Linux aims to be functionally compatible with Red Hat Enterprise Linux. The CentOS Project is the organization that builds CentOS. We mainly change packages to remove upstream vendor branding and artwork.
For information on CentOS please visit the [CentOS website](#).

Note:
CentOS is an Operating System and it is used to power this website; however, the webserver is owned by the domain owner and not the CentOS Project. If you have issues with the content of this site, contact the owner of the domain, not the CentOS Project.
Unless this server is on the [centos.org](#) domain, the CentOS Project doesn't have anything to do with the content on this webserver or any e-mails that directed you to this site.
For example, if this website is www.example.com, you would find the owner of the example.com domain at the following WHOIS server.
<http://www.internic.net/whois.html>

17. External Script

The shell provisioner can also take an option specifying a path to a shell script on the host machine. Vagrant will then upload this script into the guest and execute it. An example:

```
Vagrant.configure("2") do |config|
  config.vm.provision "shell", path: "script.sh"
end
```

Relative paths, such as above, are expanded relative to the location of the root Vagrantfile for your project. Absolute paths can also be used, as well as shortcuts such as ~ (home directory) and .. (parent directory).

18. Synced Folders

Synced folders enable Vagrant to sync a folder on the host machine to the guest machine, allowing you to continue working on your project's files on your host machine, but use the resources in the guest machine to compile or run your project.

By default, Vagrant will share your project directory (the directory with the [Vagrantfile](#)) to /vagrant.

Configuration

Synced folders are configured within your Vagrantfile using the `config.vm.synced_folder` method. Usage of the configuration directive is very simple:

Visualpath Training & Consulting.

Flat no: 205, Nilgiri Block, Aditya Enclave, Ameerpet, Hyderabad, Phone No: - +91-970 445 5959, 961 824 5689 E-Mail ID : online.visualpath@gmail.com, Website : www.visualpath.in.

```
Vagrant.configure("2") do |config|
  # other config here

  config.vm.synced_folder "src/", "/srv/website"
end
```

The first parameter is a path to a directory on the host machine. If the path is relative, it is relative to the project root. The second parameter must be an absolute path of where to share the folder within the guest machine. This folder will be created (recursively, if it must) if it does not exist.

19. Defining Multiple Machines

Multiple machines are defined within the same project Vagrantfile using the `config.vm.define` method call. This configuration directive is a little funny, because it creates a Vagrant configuration within a configuration. An example shows this best:

```
Vagrant.configure("2") do |config|
  config.vm.provision "shell", inline: "echo Hello"

  config.vm.define "web" do |web|
    web.vm.box = "apache"
  end

  config.vm.define "db" do |db|
    db.vm.box = "mysql"
  end
end
```

As you can see, `config.vm.define` takes a block with another variable. This variable, such as `web` above, is the *exact* same as the `config` variable, except any configuration of the inner variable applies only to the machine being defined. Therefore, any configuration on `web` will only affect the `web` machine.

20. Controlling Multiple Machines

The moment more than one machine is defined within a Vagrantfile, the usage of the various `vagrant` commands changes slightly. The change should be mostly intuitive.

Commands that only make sense to target a single machine, such as `vagrant ssh`, now *require* the name of the machine to control. Using the example above, you would say `vagrant ssh web` or `vagrant ssh db`.

Other commands, such as `vagrant up`, operate on *every* machine by default. So if you ran `vagrant up`, Vagrant would bring up both the web and DB machine. You could also optionally be specific and say `vagrant up web` or `vagrant up db`.

21. Forwarded Ports

Network identifier: `forwarded_port`

Vagrant forwarded ports allow you to access a port on your host machine and have all data forwarded to a port on the guest machine, over either TCP or UDP.

For example: If the guest machine is running a web server listening on port 80, you can make a forwarded port mapping to port 8080 (or anything) on your host machine. You can then open your browser to `localhost:8080` and browse the website, while all actual network data is being sent to the guest.

Defining a Forwarded Port

The forwarded port configuration expects two parameters, the port on the guest and the port on the host. Example:

```
Vagrant.configure("2") do |config|
  config.vm.network "forwarded_port", guest: 80, host: 8080
end
```

This will allow accessing port 80 on the guest via port 8080 on the host.

22. Few Sample Vagrantfile's

Multivm Vagrantfile, every vm with its own custom setting.

```
Vagrant.configure("2") do |config|
  config.vm.define "web" do |web|
    web.vm.box = "ubuntu/trusty64"
    web.vm.hostname = 'web'

    web.vm.network "public_network", bridge: "wlo1"
```

Visualpath Training & Consulting.

Flat no: 205, Nilgiri Block, Aditya Enclave, Ameerpet, Hyderabad, Phone No: - +91-970 445 5959, 961 824 5689 E-Mail ID : online.visualpath@gmail.com, Website : www.visualpath.in.

```

web.vm.provision :shell, path: "bootstrap.sh"

web.vm.network :forwarded_port, guest: 80, host: 4567

` web.vm.provider :virtualbox do |v|
  v.customize ["modifyvm", :id, "--memory", 512]
end

end

config.vm.define "db" do |db|
  db.vm.box = "ubuntu/trusty64"
  db.vm.hostname = 'db'

  db.vm.network "public_network", bridge: "wlo1"

  db.vm.provider :virtualbox do |v|
    v.customize ["modifyvm", :id, "--memory", 512]
  end
end
end

```

Vagrantfile to setup three nodes with custom synced directory and private network.

```

# README

#
# Getting Started:

# 1. vagrant plugin install vagrant-hostmanager
# 2. vagrant up
# 8. vagrant ssh
#
# This should put you at the control host
# with access, by name, to other vms

Vagrant.configure(2) do |config|
  config.hostmanager.enabled = true

```

```

# config.vm.box = "nrel/CentOS-6.5-x86_64"

config.vm.define "puppetmaster", primary: true do |h|
  h.vm.box = "nrel/CentOS-6.5-x86_64"
  h.vm.network "private_network", ip: "192.168.8.10"
  h.vm.provider :virtualbox do |vb|
    vb.customize ["modifyvm", :id, "--memory", "2048"]
  end
  h.vm.hostname = 'puppetmaster'
  h.vm.synced_folder "puppet_repo", "/etc/puppet"
end

config.vm.define "lamp" do |h|
  h.vm.box = "nrel/CentOS-6.5-x86_64"
  h.vm.network "private_network", ip: "192.168.8.11"
  h.vm.hostname = 'lamp'
end

config.vm.define "lamptest" do |h|
  h.vm.box = "ubuntu/trusty64"
  h.vm.network "private_network", ip: "192.168.8.12"
  h.vm.hostname = 'lamptest'
end

```

Vagrantfile to setup 5 nodes.

Provisioning shell script to do SSH key exchange from autoserver to other 4 nodes.

```

# README

#
# Getting Started:

```

Visualpath Training & Consulting.

Flat no: 205, Nilgiri Block, Aditya Enclave, Ameerpet, Hyderabad, Phone No: - +91-970 445 5959, 961 824 5689 E-Mail ID : online.visualpath@gmail.com, Website : www.visualpath.in.

```

# 1. vagrant plugin install vagrant-hostmanager
# 2. vagrant up
# 3. vagrant ssh
#
# This should put you at the autoserver host
# with access, by name, to other vms
Vagrant.configure(2) do |config|
  config.hostmanager.enabled = true

  config.vm.box = "ubuntu/trusty64"

  config.vm.define "autoserver", primary: true do |h|
    h.vm.network "private_network", ip: "192.168.2.35"

    h.vm.hostname = 'autoserver'
    h.vm.synced_folder "../autoserver_home", "/home/vagrant/repo"
    h.vm.provision :shell, :inline => <<'EOF'
if [ ! -f "/home/vagrant/.ssh/id_rsa" ]; then
  ssh-keygen -t rsa -N "" -f /home/vagrant/.ssh/id_rsa
fi
cp /home/vagrant/.ssh/id_rsa.pub /vagrant/autoserver.pub

cat << 'SSHEOF' > /home/vagrant/.ssh/config
Host *
  StrictHostKeyChecking no
  UserKnownHostsFile=/dev/null
SSHEOF

chown -R vagrant:vagrant /home/vagrant/.ssh/
EOF
end

config.vm.define "nginx0101" do |h|
  h.vm.network "private_network", ip: "192.168.2.36"
  h.vm.hostname = 'nginx0101'

```

Visualpath Training & Consulting.

Flat no: 205, Nilgiri Block, Aditya Enclave, Ameerpet, Hyderabad, Phone No: - +91-970 445 5959, 961 824 5689 E-Mail ID : online.visualpath@gmail.com, Website : www.visualpath.in.

```

h.vm.provision :shell, inline: 'cat /vagrant/autoserver.pub >>
/home/vagrant/.ssh/authorized_keys'

end

config.vm.define "apache01" do |h|
  h.vm.network "private_network", ip: "192.168.2.37"
  h.vm.hostname = 'apache01'

  h.vm.provision :shell, inline: 'cat /vagrant/autoserver.pub >>
/home/vagrant/.ssh/authorized_keys'

end

config.vm.define "apache02" do |h|
  h.vm.network "private_network", ip: "192.168.2.38"
  h.vm.hostname = 'apache02'

  h.vm.provision :shell, inline: 'cat /vagrant/autoserver.pub >>
/home/vagrant/.ssh/authorized_keys'

end

config.vm.define "mysql01" do |h|
  h.vm.network "private_network", ip: "192.168.2.39"
  h.vm.hostname = 'mysql01'

  h.vm.provision :shell, inline: 'cat /vagrant/autoserver.pub >>
/home/vagrant/.ssh/authorized_keys'

end

```

Summary:

- ✓ Virtualization gave us the power of deploying multiple app in one physical server each having their own OS.
- ✓ Hypervisor gives us power and interface to create multiple virtual server also known as VM on one physical machine.
- ✓ There are two types of hypervisor, type1 which is installed directly on Hardware and type 2 which gets installed on a host OS.
- ✓ Virtualbox is type 2 hypervisor which can be installed on Linux and windows machine's.
- ✓ We can create & manage VM's and their resources like virtual hard disk, Memory, Vcpu, network etc through virtualbox interface.
- ✓ Creating & Managing lot vm from virtualbox interface is time consuming and mundane task.

Visualpath Training & Consulting.

Flat no: 205, Nilgiri Block, Aditya Enclave, Ameerpet, Hyderabad, Phone No: - +91-970 445 5959, 961 824 5689 E-Mail ID : online.visualpath@gmail.com, Website : www.visualpath.in.

- ✓ Vagrant is a command line tool that helps us create & manage virtual machine lifecycle automatically.
- ✓ Vagrantcloud hosts many of vm images AKA boxes which can be used by anyone using Vagrant tool. We don't do OS installation anymore with Vagrant, we just pull vagrant box images and run them.
- ✓ Vagrantfile is the single source of truth for your VM's in Vagrant. We can mention box name, its memory, network and other settings while spinning a VM in the Vagrantfile.
- ✓ Vagrantfile gives us the feature of provisioning by which we can run any automation script to configure VM after vm comes up.
- ✓ Multiple VM's can be managed by a single Vagrantfile.

Conclusion:

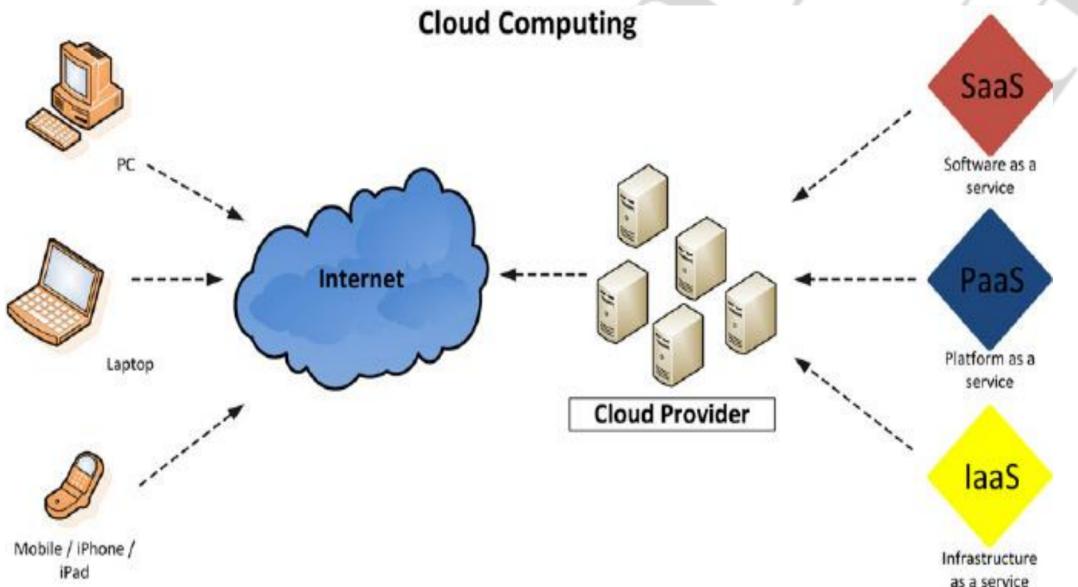
Vagrant is a great tool for our day to day DevOps tasks. Maybe you have written a script for deployment or maybe you are learning any new devops tool, to test all these things you need VM's. With Vagrant, we can quickly set up multiple vm's and start practicing. It's part of our daily toolkit. We can also use Ansible, Chef or Puppet code in provisioning part of the Vagrantfile, once we learn these tools in later chapters you can go ahead and try those. Vagrant can also be used to provision instances on AWS cloud.

If you are working in DevOps you will encounter with local virtual machines, always use Vagrant to maintain your local vm setup.

V. Amazon Web Services

1. What is cloud computing?

Cloud computing is the on-demand delivery of compute power, database storage, applications, and other IT resources through a cloud services platform via the internet with pay-as-you-go pricing.



What are different types of Clouds?

We have three types of cloud infrastructure.

Private Cloud:

The cloud infrastructure is provisioned for exclusive use by a single organization comprising multiple consumers (e.g., business units). It may be owned, managed, and operated by the organization, a third party, or some combination of them, and it may exist on or off premises.

Public Cloud:

The cloud infrastructure is provisioned for open use by the general public. It may be owned, managed, and operated by a business, academic, or government organization, or some combination of them. It exists on the premises of the cloud provider.

Hybrid Cloud:

The cloud infrastructure is a composition of two or more distinct cloud infrastructures (private, community, or public) that remain unique entities, but are bound together by standardized or proprietary technology that enables data and application portability (e.g., cloud bursting for load balancing between clouds).

Visualpath Training & Consulting.

Flat no: 205, Nilgiri Block, Aditya Enclave, Ameerpet, Hyderabad, Phone No: +91-970 445 5959, 961 824 5689 E-Mail ID : online.visualpath@gmail.com, Website : www.visualpath.in.