Vagrant

What is Vagrant

Vagrant is open source tool to create Development environment and test the fixes/code/features before putting it on real environment
It Creates and configures lightweight, reproducible, and portable development environments
Most importantly it is an Open Source Software – Free!!
It automates virtual machine creation using Oracle’s Virtual Box
The package/product is written in Ruby
Avoiding all confusions, it is not a Virtual Machine solution, like Virtualbox, VMware.
It is just a “helper”, a automated solution to interact easily with VirtualBox from Oracle.
It sets-up a new VM based on the pre-configured Box that is specified on the Vagrantfile. The user does not have to open VirtualBox and configure the system, vagrant will deal with it.
Static IPs are assigned to the VM, and is accessible from outside
Necessary Ports can be enabled for access
The best part is, once the job is done, the environment can be destroyed or removed
The entire setup can be brought to life, in a few minutes, by clicking a single command

Advantages:

Reduces the Setup time
Very simple and straight forward
Self servicing, based on Pull from a Base Box(each VM is based on a base box)
Consistency can be maintaining by spawning multiple instances which are just identical.
Repeatability (A new VM can be created and crushed in few minutes)
Helps create a production-like VM in minutes
With an agile driven development approach where testing becomes handy, virtualization tools like Vagrant, offer a great helping hand.

Setup and Configuration:

1. **Download oracle virtualbox**
2. **Install vagrant.exe**
3. **Install gitbash**

mkdir vagrant

cd vagrant/

vagrant init creates a file called Vagrantfile

Open this file:

Config.vm.box = “base” box is a image

Vagrant will extract this images and create VM for us.

Box images are not isos – these are installed virtual machines

There are many boxes available we can directly install and use

Some boxes are official – provided by Ubuntu itself.

Vagrant is for Testing, training, not for production machines

Vagrant box add ubuntu/trustry64 downloads the box

And change the Vagrantfile as below:

config.vm.box = “ubuntu/trusty64”

vagrant can do following things on VMs

restart

stop

start

edit

vagrant halt   stops/powers it off

vagrant destroy   will delete the VM

vagrant reload   reboots VM

vagrant reload –provision   setting up the VM from scratch ex: setting web server on a freshly installed VM.

Vagrant suspend – hibernates VM

Virtualbox – is a hypervisor which virtualizes the hardware.

Vmware workstation/vmware server/hyperv/

Vagrant is command line tool to automate

vagrant will create a virtual machine

vagrant up will read current directory and set up a virtual machine

default config:

memory = 512 MB

Vagrant is developed by Hashi corp

Sites to download boxes:

Example: https://atlas.hashicorp.com/centos/boxes/7

Vagrant downloads boxes from https://vagrantcloud.com/

```
vagrant init centos-7.0-x86_64

vagrant box add centos-7.0-x86_64 https://github.com/tommy-muehle/puppet-vagrant-boxes/releases/download/1.1.0/centos-7.0-x86_64.box

vagrant up

vagrant reload
```

Vagrant files:

1. **Basic:**

```
# -*- 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://atlas.hashicorp.com/search.
  config.vm.box = "ubuntu/trusty64"

  # 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.
  # config.vm.network "forwarded_port", guest: 80, host: 8080

  # 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"

  # 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.

  # Define a Vagrant Push strategy for pushing to Atlas. Other push strategies
  # such as FTP and Heroku are also available. See the documentation at
  # https://docs.vagrantup.com/v2/push/atlas.html for more information.
  # config.push.define "atlas" do |push|
  #   push.app = "YOUR_ATLAS_USERNAME/YOUR_APPLICATION_NAME"
  # end

  # 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
End
```

2. Bridged network:

```
# -*- 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://atlas.hashicorp.com/search.
  config.vm.box = "ubuntu/trusty64"

  # 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.
  # config.vm.network "forwarded_port", guest: 80, host: 8080

  # 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", bridge: "wlo1"

  # 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"
  # 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.

  # Define a Vagrant Push strategy for pushing to Atlas. Other push strategies
  # such as FTP and Heroku are also available. See the documentation at
  # https://docs.vagrantup.com/v2/push/atlas.html for more information.
  # config.push.define "atlas" do |push|
  #   push.app = "YOUR_ATLAS_USERNAME/YOUR_APPLICATION_NAME"
  # end

  # 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
End
```

Note:

If host OS is linux (ubuntu/rhel) the wireless adapter will be wlo1

We need to bridge VM to the host to use the network on VM

By default VM gets NAT – (Network address translation) adapter

It gets private adapter.

VMs can talk to each other using NAT

If VM has to ping internet, we need to add Bridged adapter

```
$ cat Vagrantfile
Vagrant.configure("2") do |config|
 config.vm.box = "ubuntu/trusty64"
 config.vm.network "public_network", bridge: "Intel(R) Dual Band Wireless-AC 7265 #2"
 config.vm.define "myvm" do  |myvm|
 end
 end
```

To login as root

sudo su –

Set password for root using

passwd root

to enable ssh for root login

go to /etc/ssh/sshd_config

permitrootloging yes

and restart ssh using

```
service ssh restart
```

3, Define memory

```
# -*- 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://atlas.hashicorp.com/search.
  config.vm.box = "ubuntu/trusty64"

  # 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.
  # config.vm.network "forwarded_port", guest: 80, host: 8080

  # 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", bridge: "wlo1"

  # 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"
  # 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.

  # Define a Vagrant Push strategy for pushing to Atlas. Other push strategies
  # such as FTP and Heroku are also available. See the documentation at
  # https://docs.vagrantup.com/v2/push/atlas.html for more information.
  # config.push.define "atlas" do |push|
  #   push.app = "YOUR_ATLAS_USERNAME/YOUR_APPLICATION_NAME"
  # end

  # 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
End
```

4. **Memory and CPu:**

```
Vagrant.configure("2") do |config|
 config.vm.define "myvmmemcpu" do  |myvmmemcpu|
 config.vm.box = "ubuntu/trusty64"
 config.vm.network "public_network", bridge:
"Intel(R) Dual Band Wireless-AC 7265 #2"
 myvmmemcpu.vm.provider :virtualbox do |v|
 v.customize ["modifyvm", :id, "--memory", 4096]
 v.customize ["modifyvm", :id, "--cpus", 2]
 end
 end
 end
```

5. Multi box

```
Vagrant.configure("2") do |config|
  config.vm.define "web" do |web|
   web.vm.box = "ubuntu/trusty64"
    web.vm.hostname = 'web'
  end
  config.vm.define "db" do |db|
    db.vm.box = "nrel/CentOS-6.5-x86_64"
    db.vm.hostname = 'db'
  end
  config.vm.define "control" do |control|
    control.vm.box = "ubuntu/xenial64"
    control.vm.hostname = "control"
  end

end
```

config.vm.box = “ubuntu/trusty64”   global setting

we can ignore specifying the boxes – if we specify the global setting.

multiVM with public network:

create 3 VMs with names web,db,app

```
Vagrant.configure("2") do |config|
  config.vm.define "web" do |web|
   web.vm.box = "geerlingguy/centos7"
   web.vm.network "public_network", bridge: "Broadcom 802.11n Network Adapter"
    web.vm.hostname = 'web'
  end

  config.vm.define "db" do |db|
    db.vm.box = "geerlingguy/centos7"
    db.vm.network "public_network", bridge: "Broadcom 802.11n Network Adapter"
    db.vm.hostname = 'db'
  end
  config.vm.define "app" do |app|
    app.vm.box = "geerlingguy/centos7"
    app.vm.network "public_network", bridge: "Broadcom 802.11n Network Adapter"
    app.vm.hostname = "app"
  end
end
==========================================
```

5. Windows:

```
# -*- 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://atlas.hashicorp.com/search.
  config.vm.box = "designerror/windows-7"

  # 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.
  # config.vm.network "forwarded_port", guest: 80, host: 8080

  # 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"

  # 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.

  # Define a Vagrant Push strategy for pushing to Atlas. Other push strategies
  # such as FTP and Heroku are also available. See the documentation at
  # https://docs.vagrantup.com/v2/push/atlas.html for more information.
  # config.push.define "atlas" do |push|
  #   push.app = "YOUR_ATLAS_USERNAME/YOUR_APPLICATION_NAME"
  # end

  # 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
End
```

```
config.vm.network :forwarded_port, guest: 4000, host: 4000
```

network sample vagrantfile:

```
$ cat Vagrantfile
Vagrant.configure("2") do |config|
 config.vm.box = "ubuntu/trusty64"
 config.vm.network "public_network", bridge: "Intel(R) Dual Band
Wireless-AC 7265 #2"
 config.vm.define "myvm" do  |myvm|
 end
 end
```

6. Provisioning using vagrant

Synced folders:

By default /vagrant directory will be created on the VM

And is shared with host machine where the vagrant file is created.

Whatever date we put in host will also be available in /vagrant

We can have bootstrap.sh for provisioning

Example:

cat bootstrap.sh

```
#!/bin/bash
apt-get update
apt-get install apache2 -y
apt-get install vim
service apache2 restart
```

Give 777 permission

Line to be added in Vagrant file:

config.vm.provision :shell, path: “apache.sh”

Sample file:

```
# -*- 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://atlas.hashicorp.com/search.
  config.vm.box = "ubuntu/trusty64"

  # 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.
  # config.vm.network "forwarded_port", guest: 80, host: 8080

  # 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", bridge: "wlo1"

  # 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"

  # 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.

  # Define a Vagrant Push strategy for pushing to Atlas. Other push strategies
  # such as FTP and Heroku are also available. See the documentation at
  # https://docs.vagrantup.com/v2/push/atlas.html for more information.
  # config.push.define "atlas" do |push|
  #   push.app = "YOUR_ATLAS_USERNAME/YOUR_APPLICATION_NAME"
  # end

  # 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
config.vm.provision :shell, path: "bootstrap.sh"
```

```
#  config.vm.network :forwarded_port, guest: 80, host: 4567

End
```

To get the bridge of the windows Machine:

Go to

C:/Program Files/Oracle/VirtualBox

And execute

```
./VBoxManage.exe list  bridgedifs
Name:            Intel(R) Dual Band
Wireless-AC 7265 #2
GUID:            403cc955-931f-465b-b570-
d075d3b5fea3
DHCP:            Enabled
IPAddress:       192.168.0.100
NetworkMask:     255.255.255.0
IPV6Address:
fe80:0000:0000:0000:81d6:02f9:dc90:0215
IPV6NetworkMaskPrefixLength: 64
```

```
HardwareAddress: 48:45:20:10:f5:e5
MediumType:      Ethernet
Status:          Up
VBoxNetworkName: HostInterfaceNetworking-
Intel(R) Dual Band Wireless-AC 7265 #2

Name:            Intel(R) Ethernet
Connection (3) I218-LM
GUID:            2693ad08-d2fc-4edf-aefc-
c29b04b46306
DHCP:            Enabled
IPAddress:       10.0.0.20
NetworkMask:     255.255.255.0
IPV6Address:
fe80:0000:0000:0000:bca2:f0db:5732:094b
IPV6NetworkMaskPrefixLength: 64
HardwareAddress: 50:7b:9d:dd:a2:27
MediumType:      Ethernet
Status:          Down
```

```
VBoxNetworkName: HostInterfaceNetworking-Intel(R) Ethernet Connection (3) I218-LM
```

7.Port forwarding:

```
Vagrant.configure("2") do |config|
 config.vm.box = "ubuntu/trusty64"
 config.vm.network "public_network", bridge: "Intel(R) Dual Band Wireless-AC 7265 #2"
 config.vm.network "forwarded_port", guest: 80, host: 8080
 config.vm.define "myvmp" do  |myvmp|
 end
 end
```

Guest’s port 80 can be accessed on the host in the port 8080

http://127.0.0.1:8080/

To attach a disk to vagrant VM:

```
unless File.exist?(file_to_disk)
    config.vm.customize ['createhd', '--filename', file_to_disk, '--size', 500 * 1024]
  end
  config.vm.customize ['storageattach', :id, '--storagectl', 'SATA Controller', '--port', 1, '--device', 0, '--type', 'hdd', '--medium', file_to_disk]
```

for centOS:

```
Vagrant.configure("2") do |config|
 config.vm.box = "geerlingguy/centos7"
 config.vm.network "public_network", bridge: "Intel(R) Dual Band Wireless-AC 7265 #2"
 config.vm.define "centos2" do  |centos2|
 end
 end
```

```
Vagrant.configure("2") do |config|
  config.vm.define "web" do |web|
   web.vm.box = "geerlingguy/centos7"
   web.vm.network "public_network", bridge: "Broadcom 802.11n Network Adapter"
    web.vm.hostname = 'server'
  end
```