

Vagrant Up in 5 Easy Steps



Trevor Roberts Jr
@VMTrooper



Introduction

- What is Vagrant?
 - Created by Mitchell Hashimoto
 - Provisioning Tool for Dev & Test Environments
- Why use Vagrant?
 - Quick
 - Easily replicate production on a Dev box
- How do I get started?

Step 1: Select your Provider

- What is a provider?
 - Oracle VirtualBox (Free)
 - VMware Fusion or Workstation (\$)
 - Build your own (AWS, Rackspace, etc.)

Step 2: Install Vagrant

- <http://www.vagrantup.com/downloads.html>

Step 3: Select a box

- Vagrant Cloud:

<https://vagrantcloud.com/discover/featured>

- Build your own:

<http://www.packer.io/docs/post-processors/vagrant.html>

- Download and Use Others not on Vagrant Cloud:

<http://www.vagrantbox.es/>

We will use **hashicorp/precise64** since it is built for Virtualbox and VMware

Step 4: Vagrant Init & Up

Open a Terminal Window in the directory for your test lab:

- vagrant init hashicorp/precise64
- vagrant up (Virtualbox)
- vagrant up --provider=vmware_fusion (VMware)

vagrant up --provider=vmware_fusion or vagrant up --provider=vmware_workstation (VMware)

Step 5: Vagrant SSH

Login to the VM using vagrant CLI:

- `vagrant ssh`

Success!



But wait, there's more...

- Version Control
- Customize Your VM
- Multi-VM Configuration (Static & Dynamic)
- VM Provisioners
- Squid Proxy

Version Control

- Source Code Control for your Vagrantfile
 - Git, SVN, etc.
- Online collaboration
 - GitHub, BitBucket, etc.
- This presentation and example code can be found at:
<https://github.com/VMTrooper/VagrantBrownBag>

Customize Your VM

- Hostname
 - config.vm.hostname = "controller"
- IP Address
 - config.vm.network :private_network, ip: 178.16.172.200

NOTE: For the :public_network option, you will need to do some extra work involving DHCP

For :public_network, see <http://vmtrooper.com/vagrant-static-external-ip-addresses-with-the-vmware-fusion-provider/>

Customize Your VM

- IMPORTANT for Virtualbox!
 - Port forwarding must be used if you want to access a specific port on your VM from your desktop... because, reasons:
 - config.vm.network "forwarded_port", guest: 80, host: 8080

For :public_network, see <http://vmtrooper.com/vagrant-static-external-ip-addresses-with-the-vmware-fusion-provider/>

Customize Your VM

- Memory

```
# If using Fusion
config.vm.provider :vmware_fusion do |v|
    v.vmx["memsize"] = 1024
    v.vmx["numvcpus"] = 2
end

# If using VirtualBox
config.vm.provider :virtualbox do |vbox|
    vbox.customize ["modifyvm", :id, "--memory", 1024]
    vbox.customize ["modifyvm", :id, "--cpus", 2]
end
```

Multi-VM Configuration

- Share files between VMs.
 - The Vagrantfile's folder is mounted to `/vagrant` on each VM
- Use Ruby to make multi-VM deployments A LOT simpler (think loops, branching, variables!)
 - Compare sample code in `./multivmstatic` vs `./multivmdynamic`

NOTE:

`vagrant ssh` without a VM name will not work since Vagrant is aware you created multiple VMs.

Use the `vagrant status` command to get your vm names:

bash-3.2\$ `vagrant status`

Current machine states:

controller	running (virtualbox)
compute	running (virtualbox)

This environment represents multiple VMs. The VMs are all listed above with their current state. For more information about a specific VM, run `'vagrant status NAME'`.

THEN, use the command `vagrant ssh <vmname>` ex: `vagrant ssh compute`

Provisioners

- Configuration Management
 - Shell Scripts
 - Puppet (Standalone & Agent)
 - Chef (Solo & Client)
 - Ansible
 - SaltStack
 - CFEngine

Additional Info

- Vagrant Site: <http://docs.vagrantup.com/v2/>
- Google Groups: <https://groups.google.com/forum/#!forum/vagrant-up>
- IRC: #vagrant on Freenode
- GitHub (Check out bunchc, bodepd, ody, vmtrooper (me!), etc)
- For Fusion-related tips, I post them as I find them:
 - <http://vmtrooper.com/category/automation/vagrant/>
- Mitchell's Book

