

Getting started with Vagrant

Guillermo Guerrero Ibarra

Simpple

guillermo@guerreroibarra.com

http://guillermo.guerreroibarra.com

February 18, 2015

Overview

1 Intro

- Why Vagrant?

2 Vagrant

- Installation
- How it works
- Configuration

3 Provisioning

- Ways to provision
- Cookbooks
- Berkshelf

4 Interesting links

- Vagrant auto generators

Why Vagrant?

- Project dependencies:
 - Database
 - Programming language
 - Development tools
- Isolation
- Easy to use
- Multiplatform
- Compatible with multiple virtualization software:
 - VirtualBox / VMware / Parallels
 - Snapshots

Vagrant installation

Download and install the following packages:

- Virtual box → <https://www.virtualbox.org>
- Vagrant → <https://www.vagrantup.com>
- Chefdk → <https://downloads.chef.io/chef-dk>
- Berkshelf → <http://berkshelf.com>
 - \$ vagrant plugin install vagrant-berkshelf
 - \$ vagrant plugin install vagrant-triggers

How it works

Vagrant commands:

- Init → Vagrantfile
- Up (First time Vagrant executes provision)
- Halt
- Suspend / Resume
- Provision → Shell / Chef
- Destroy

Vagrant triggers:

- On up
- On destroy
- On provision
- ...

Vagrant configuration

\$ vagrant init → Vagrantfile

```
1 Vagrant.configure(2) do |config|
2   config.vm.box = 'ubuntu/trusty64'
3
4   config.ssh.forward_agent = true
5   config.vm.network 'forwarded_port', :guest => 3000, :host
      => 9000
6   config.vm.network 'forwarded_port', :guest => 3306, :host
      => 9306
7
8   config.vm.network 'private_network', :ip => '10.0.2.15'
9
10  config.vm.provider 'virtualbox' do |vb|
11    vb.memory = '1024'
12  end
13 end
```

Ways to provision

Shell

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

Chef

```
1 config.vm.provision :chef_solo do |chef|
2     chef.add_recipe 'apt'
3     chef.add_recipe 'build-essential'
4     chef.add_recipe 'git'
5     chef.add_recipe 'nodejs'
6 end
```

Others → puppet, knife, chef zero...

Cookbooks

- Supermarket → <https://supermarket.chef.io>
- Recipe order is important
- Chef.json parameters in Vagrantfile

Chef

```
1 config.vm.provision :chef_solo do |chef|
2   chef.add_recipe 'apt'
3   chef.add_recipe 'build-essential'
4   chef.add_recipe 'git'
5   chef.add_recipe 'nodejs'
6   chef.add_recipe 'rvm::system'
7   chef.add_recipe 'rvm::vagrant'
8   chef.add_recipe 'app::packages'
9   chef.add_recipe 'app::mysql'
10  chef.add_recipe 'app::gemset'
11  chef.add_recipe 'app::environment'
12  chef.add_recipe 'app::post-install'
13 end
```


Berkshelf → <http://berkshelf.com>

Berksfile

```
1 source "https://supermarket.getchef.com"
2
3 cookbook 'apt'
4 cookbook 'git'
5 cookbook 'build-essential'
6 cookbook 'nodejs'
7 cookbook 'mysql'
8 cookbook 'rvm'
9 cookbook 'openssl'
0 cookbook 'app', path: './site-cookbooks/app'
```

Custom cookbooks

- My example → <https://github.com/ryanfox1985/vagrant-rails-dev>
- Recipes are in ruby
- Templates files support erb (parameters)

Cookbooks folders

```
1 site --cookbooks/app
2   recipes
3     list_of_recipies.rb
4   templates
5     defaults_files
6   Berksfile
7   metadata.rb
```

Vagrant auto generators

For noobs:

- PuPHPet → <https://puphpet.com>
- Rove → <http://rove.io>

Questions?

