Vagrant & Puppet

Deploying Development Environments ... Fast

James Turnbull

Me

- FOSS contributor
- Author & Geek
- Work at Puppet Labs
- Talks funny

How did we get here?

All opinions my own. May be subject to change with added alcohol.

Problems

Development != Production

Production != Development

aka

“It works on my box”

Problems

No quick starts for new employees

Problems

Not repeatable

Problems

Testing is expensive

Especially when Development should be agile

Problems

Not every developer is a sysadmin

Not every developer wants to be a sysadmin


WEAK

Solutions

Virtualisation

Virtualisation

- VMWare
- Xen
- Virtual Box
- KVM

Not Quite Right…

Virtualisation needs hardware

Hardware costs money and takes time

Still needs to be managed

… and still may not be “realistic”

Solutions

Cloud

Cloud

- Amazon AWS
- Rackspace Cloud
- VMWare CloudFoundry
- Insertnameofcloudproviderhere

Still Not Quite Right…

Cloud costs money

Cloud requires bandwidth and access

Cloud instances also need management

… and still may not be “realistic”

Solutions

Configuration Management

Configuration Management

- Puppet
- Chef
- Cfengine
- Evil Proprietary Software (sic)

Also Not Quite Right…

Still requires a platform

Requires skills and knowledge

Solution?

Virtualisation + Configuration Management?

Solution!

Vagrant

Vagrant

- Open Source Software
- VirtualBox-based
- Written in Ruby
- Free!

VirtualBox

- Shared folders / NFS
- Configurable networking
- All the features of VirtualBox

Vagrant

- Integrated with Configuration Management
- Build your own boxes
- Run up and tear down – fast!
- Easy, fast, command line based
- Extensible… even by idiots like me

Installing Vagrant

1. Install VirtualBox

   http://www.virtualbox.org/wiki/Downloads

2. Install Vagrant

   $ gem install vagrant

Starting Vagrant

```
$ vagrant box add lucid32 http://.../lucid32.box
$ vagrant up
$ vagrant ssh
```

Many boxes…

http://www.vagrantbox.es/

- Ubuntu
- Debian
- SLES
- CentOS
- Solaris…

The Vagrantfile

```
Vagrant::Config.run do |config|
  config.vm.box = "lucid32"
end
```

Shared Folders

```
Vagrant::Config.run do |config|
  config.vm.share_folder("folder", "/folder", "/local/folder")
  config.vm.share_folder("v-root", "/vagrant", ".", :nfs => true)
end
```

Configurable Networking

```
Vagrant::Config.run do |config|
  config.vm.forward_port("web", "80", "8080"
  config.vm.network("66.66.66.1")
end
```

Puppet real quick

- Open Source
- Configuration Management
- Configuration Language
- One machine … many machines

Integration With Puppet Standalone

```
Vagrant::Config.run do |config|
  config.vm.provision :puppet do |puppet|
    puppet.manifests_path = “manifests”
    puppet.manifest_file = “dev.pp”
    puppet.module_path = “modules”
  end
end
```

Integration With Puppet Server

```
Vagrant::Config.run do |config|
  config.vm.provision :puppet_server do |puppet|
    puppet.puppet_server = “puppet.example.com”
  end
end
```

Other configuration management

- Chef
- Shell scripts
- Add your own: http://vagrantup.com/docs/provisioners/others.html

Starting one box

```
Vagrant::Config.run do |config|
  config.vm.box = “dev”
end

$ vagrant up
```

Starting many boxes

```
Vagrant::Config.run do |config|
  config.vm.define :web do |web_config|
    web_config.vm.box = "web"
    web_config.vm.forward_port("http", 80, 8080)
  end

  config.vm.define :app do |app_config|
    db_config.vm.box = ”app"
    db_config.vm.forward_port(”app", 8081, 8081)
  end
end
```

Stopping boxes

$ vagrant halt

Starting over…

$ vagrant destroy

And then like magic…

$ vagrant up

Very Fast Demonstration

http://www.flickr.com/photos/alexkoch/4056654358

In the end…

- Testing looks like Production
- Early testing actually means something
- Really, really fast
- Really, really easy
- Very DevOps … impress your friends

Vagrant & Puppet

http://vagrantup.com/

Questions?