

A close-up photograph of a man with a full, dark beard and long, wavy hair. He is wearing a red turtleneck sweater and is looking off to his left with a thoughtful expression. The background is blurred, showing what appears to be an indoor event or conference.

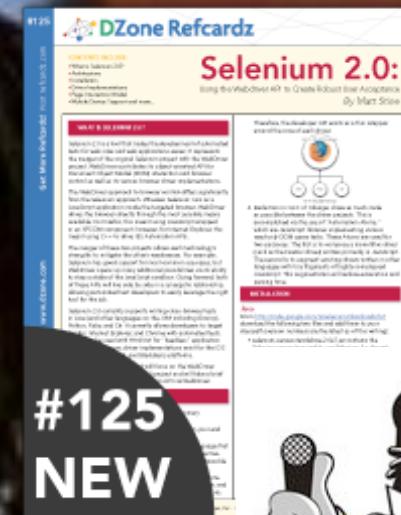
# VAGRANT

*Virtualized Development  
Environments Made Easy*

Matt Stine

# BASH-3.2\$ WHOAMI

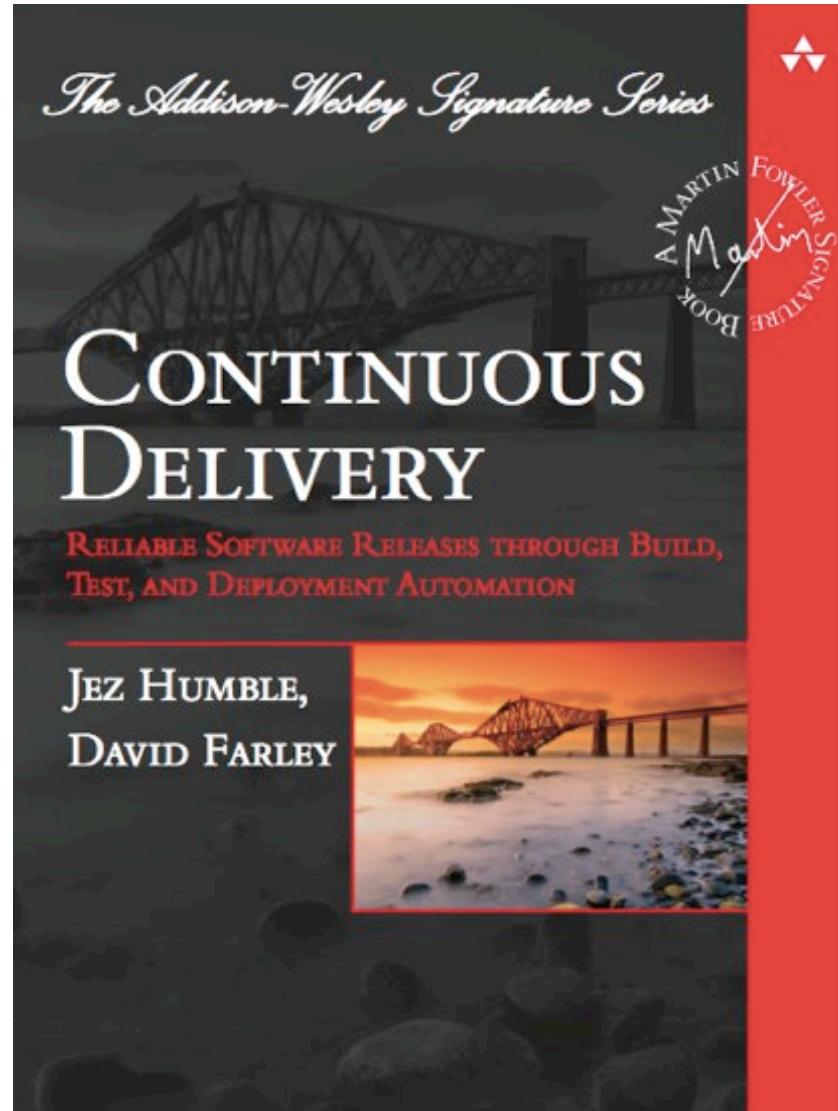
- Technical Architect - AutoZone
  - Speaker (JavaOne, SpringOne/2GX, CodeMash, NFJS, RWX, PAX, UberConf)
  - Author (GroovyMag, NFJS the Magazine, Selenium 2.0 Refcard)
  - Founder of the Memphis/Mid-South Java User Group
  - Former Agile Zone Leader @ DZone



# WHY VAGRANT?



# CONTINUOUS DELIVERY: DIFFERENCES MATTER



# THE GOAL



# **STRATEGIES OF THE PAST**



# VAGRANT



# **INSTALLATION / PREREQUISITES**





# RUBY



<http://www.ruby-lang.org/en/>



# RVM

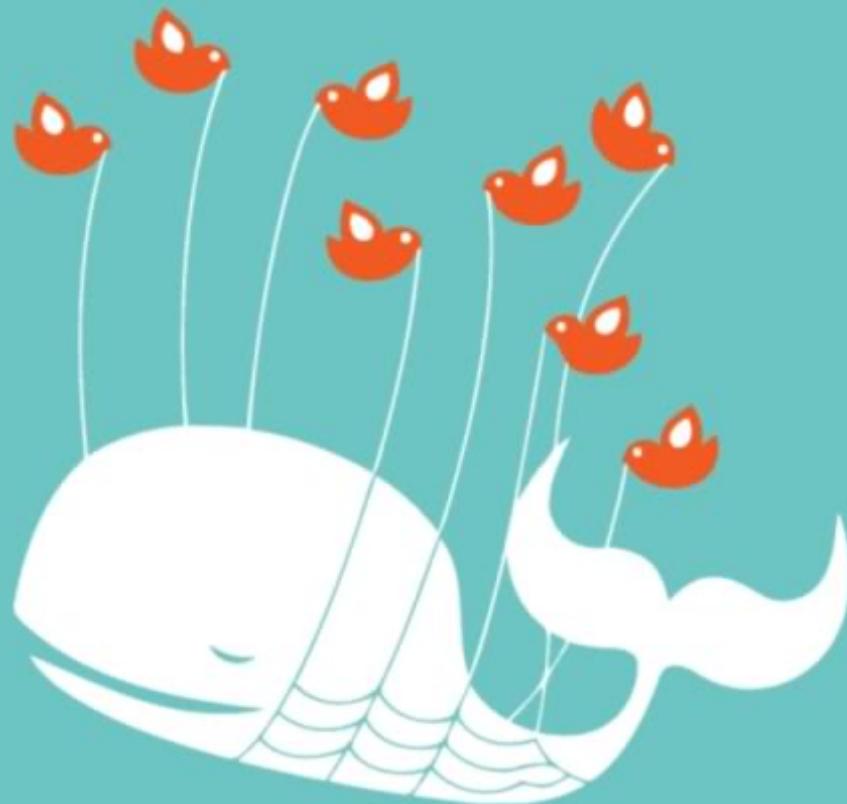
*Ruby Version Manager*

<http://beginrescueend.com/>



# WINDOWS?

*Fail Whale*



<http://vagrantup.com/docs/getting-started/setup/windows.html>



# VIRTUALBOX

<https://www.virtualbox.org/>



# **YOUR FIRST VAGRANT**



# BOXES



# **BASE BOXES**



```
vagrant box add lucid32  
http://files.vagrantup.com/  
lucid32.box
```



vagrant init  
lucid32



vagrant up



# DEMO



# VEEWEE

*Base Boxes Made Easy*

<https://github.com/jedi4ever/veewee>



# VEEWEE TEMPLATES

```
Last login: Sat Nov 19 16:33:53 on ttys003
→ ~ vagrant basebox templates
The following templates are available:
vagrant basebox define '<boxname>' 'archlinux-i386'
vagrant basebox define '<boxname>' 'archlinux-x86_64'
vagrant basebox define '<boxname>' 'CentOS-4.8-i386'
vagrant basebox define '<boxname>' 'CentOS-5.6-i386'
vagrant basebox define '<boxname>' 'CentOS-5.6-i386-netboot'
vagrant basebox define '<boxname>' 'CentOS-5.6-x86_64-netboot'
vagrant basebox define '<boxname>' 'CentOS-6.0-i386'
vagrant basebox define '<boxname>' 'CentOS-6.0-i386-netboot'
vagrant basebox define '<boxname>' 'CentOS-6.0-x86_64'
vagrant basebox define '<boxname>' 'CentOS-6.0-x86_64-netboot'
vagrant basebox define '<boxname>' 'Debian-5.0.8-amd64-netboot'
vagrant basebox define '<boxname>' 'Debian-5.0.8-i386-netboot'
vagrant basebox define '<boxname>' 'Debian-6.0.2-amd64-netboot'
vagrant basebox define '<boxname>' 'Debian-6.0.2-i386-netboot'
vagrant basebox define '<boxname>' 'Fedora-14-amd64'
vagrant basebox define '<boxname>' 'Fedora-14-amd64-netboot'
vagrant basebox define '<boxname>' 'Fedora-14-i386'
vagrant basebox define '<boxname>' 'Fedora-14-i386-netboot'
```



# VEEWEE SETUP

- Define a new box
- Optionally modify config files/scripts
- Get ISO in place
- Build the box
- Validate the box
- Export the box
- Add the box to your Vagrant setup
- Use it!



# VEEWEE DEMO



# COMMANDS



**vagrant box add**

*Add a box to your Vagrant installation*

**vagrant box remove**

*Delete a box from your Vagrant installation*

**vagrant box list**

*Lists boxes known by your Vagrant installation*

**vagrant package**

*Package a Vagrant box for redistribution*



## vagrant destroy

"Nuke" a Vagrant box - handle with care!

## vagrant halt

Forces a Vagrant box to halt - pull the plug.

## vagrant init

Initializes the current directory as the root for a Vagrant project.

## vagrant provision

Runs the provisioning (Chef/Puppet) scripts for the Vagrant box.



## vagrant resume

Start a halted Vagrant box.

## vagrant ssh

Quick and easy way to login to a Vagrant box.

## vagrant ssh-config

Gives you the goods to drop into .ssh/config

## vagrant status

Is the my Vagrant box on or off?



`vagrant suspend`

*Take a snapshot and put your Vagrant box  
to sleep.*

`vagrant up`

*Build the box and boot it up!*



# **VAGRANTFILE ANATOMY**



# HELLO, VAGRANTFILE

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



# **CONFIG OPTION SETS**



# CONFIG.SSH

config.ssh.host

config.ssh.max\_tries

config.ssh.private\_key\_path

config.ssh.timeout

config.ssh.forward\_agent

config.ssh.forward\_x11



# CONFIG.VM

config.vm.box

config.vm.box\_url

config.vm.box\_ovf

config.vm.base\_mac

config.vm.customize

config.vm.define

config.vm.forward\_port

config.vm.network

config.vm.provision

config.vm.share\_folder



# CONFIG.VM

config.vm.box

config.vm.box\_url

config.vm.box\_ovf

config.vm.base\_mac

config.vm.customize

config.vm.define

config.vm.forward\_port

config.vm.network

config.vm.provision

config.vm.share\_folder

```
config.vm.customize do |vml|
  vml.memory_size = 512
  vml.name = "My Project VM"
end
```



# CONFIG.VM

config.vm.box

config.vm.box\_url

config.vm.box\_ovf

config.vm.base\_mac

config.vm.customize

config.vm.define

config.vm.forward\_port

config.vm.network

config.vm.provision

config.vm.share\_folder

```
config.vm.forward_port("web", 80, 8080)
config.vm.forward_port("ftp", 21, 4567)
config.vm.forward_port("ssh", 22, 2222,
| :auto => true)
```



# CONFIG.VM

config.vm.box

config.vm.box\_url

config.vm.box\_ovf

config.vm.base\_mac

```
config.vm.share_folder("a", "/mstine_home", "/Users/mstine")
```

```
config.vm.share_folder("b", "/other", "../other")
```

```
config.vm.share_folder("c", "/forty-two", ".",
| :owner => "my-user", :group => "my-group")
```

config.vm.forward\_port

config.vm.network

config.vm.provision

config.vm.share\_folder



# CONFIG.PACKAGE

config.package.name

config.package.extension



# CONFIG.NFS

config.nfs.map\_uid

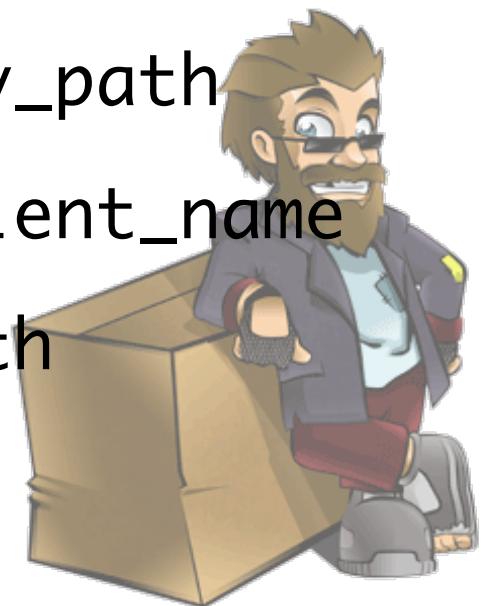
config.nfs.map\_gid

<http://vagrantup.com/docs/nfs.html>



# CONFIG.CHEF

```
if :chef_solo  
    config.chef.cookbooks_path  
elsif :chef_server  
    config.chef.chef_server_url  
    config.chef.validation_key_path  
    config.chef.validation_client_name  
    config.chef.client_key_path
```



# CONFIG.CHEF

:chef\_solo and :chef\_server

config.chef.json

config.chef.provisioning\_path

config.chef.run\_list



# CONFIG.PUPPET

config.puppet.manifest\_path

config.puppet.pp\_path



# **CONFIG.PUPPET\_SERVER**

`config.puppet_server.puppet_server`

`config.puppet_server.puppet_node`



# PROVISIONING



# WHY?



# PROVISIONING OPTIONS

- Chef Solo
- Chef Solo (Remote)
- Chef Server
- Puppet
- Puppet Server
- Shell
- Your own crypted provisioner





# PUPPET



# PUPPET WHOAMI

- Framework/platform for infrastructure automation
- Three core components:
  - Deployment
  - Configuration Language/Resource Abstraction Layer (RAL)
  - Transactional Layer
- Core principles:
  - Idempotency
  - Cross-platform
  - Model/graph-based
- Utilizes a custom, external DSL for configuration
- <http://projects.puppetlabs.com/projects/puppet/wiki>



# SETTING THE MANIFESTS PATH

```
Vagrant::Config.run do |config|
  config.vm.provision :puppet do |puppet|
    puppet.manifests_path = "manifests"
    puppet.manifest_file = "base.pp"
  end
end
```



# CONFIGURING THE MANIFESTS

```
class apache {  
    package { "apache2":  
        ensure => present,  
    }  
  
    service { "apache2":  
        ensure => running,  
        require => Package["apache2"],  
    }  
}  
  
include apache
```



# MODULES

```
Vagrant::Config.run do |config|
  config.vm.provision :puppet, :module_path => "modules"
end
```





**CHEF**



# CHEF WHOAMI

- Infrastructure automation framework focused on 3 core areas:
  - Provisioning
  - Configuration
  - Integration
- Core principles:
  - Idempotency
  - Thick Client/Thin Server
  - Ordering MATTERS
  - Reasonability
- Utilizes a Ruby internal DSL for configuration
- <http://wiki.opscode.com/display/chef/Home>



# SETTING COOKBOOKS PATH

```
Vagrant::Config.run do |config|
  config.vm.provision :chef_solo do |chef|
    chef.cookbooks_path = "cookbooks"
  end
end
```



# CONFIGURING MAIN COOKBOOK

```
# vagrant_main cookbook
# This cookbook includes and sets up a server with java, apache,
# and tomcat.
#
require_recipe "java"
require_recipe "tomcat"
require_recipe "apache2"
```

```
Vagrant::Config.run do |config|
  config.vm.provision :chef_solo do |chef|
    chef.add_recipe("vagrant_main")
  end
end
```



# JSON CONFIGURATION

```
Vagrant::Config.run do |config|
  config.vm.provision :chef_solo do |chef|
    chef.json = {
      :universal_answer => 42,
      :have_my_towel => true,
      :earth => "mostly harmless"
    }
  end
end
```



# ROLES

```
Vagrant::Config.run do |config|
  config.vm.provision :chef_solo do |chef|
    chef.roles_path = "roles"
    chef.add_role("web")
  end
end
```



# CASE STUDY





FIND A  
SPEAKER

FIND A  
FLUFFBOX

COMING  
SOON

TOP 20  
RENTALS

Welcome to Fluffbox!

Hi joeuser! [Logout](#)

[Get a Free Rental](#) [How Fluffbox Works](#)

[Search](#)

## Easy NFJS Speaker Rentals

Rent and Return at over 42,000 Locations



### FIND A SPEAKER

RWX Speakers available  
now.



### FIND A FLUFFBOX

42,000 Fluffbox locations  
near you.

# TECH STACK

- Web application developed with Grails 1.3.4
- Deploys to Apache Tomcat 6
- MySQL 5.1 Database
- Deployment orchestration using Capistrano



# MULTI-MACHINE CONFIGURATIONS



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

  config.vm.define :db do |db_config|
    db_config.vm.box = "lucid32"
  end
end
```



```
..--multimachine — zsh — 110x37
--multimachine — zsh
box.

Guest Additions Version: 4.1.0
VirtualBox Version: 4.1.4
[web] Matching MAC address for NAT networking...
[web] Clearing any previously set forwarded ports...
[web] Forwarding ports...
[web] -- ssh: 22 => 2222 (adapter 1)
[web] Creating shared folders metadata...
[web] Running any VM customizations...
[web] Booting VM...
[web] Waiting for VM to boot. This can take a few minutes.
[web] VM booted and ready for use!
[web] Mounting shared folders...
[web] -- v-root: /vagrant
[db] Fixed port collision 'ssh'. Now on port 2200.
[db] Importing base box 'lucid32'...
[db] The guest additions on this VM do not match the install version of
VirtualBox! This may cause things such as forwarded ports, shared
folders, and more to not work properly. If any of those things fail on
this machine, please update the guest additions and repack the
box.

Guest Additions Version: 4.1.0
VirtualBox Version: 4.1.4
[db] Matching MAC address for NAT networking...
[db] Clearing any previously set forwarded ports...
[db] Forwarding ports...
[db] -- ssh: 22 => 2200 (adapter 1)
[db] Creating shared folders metadata...
[db] Running any VM customizations...
[db] Booting VM...
[db] Waiting for VM to boot. This can take a few minutes.
[db] VM booted and ready for use!
[db] Mounting shared folders...
[db] -- v-root: /vagrant
➔ basic-multimachine git:(master) []
```

# **HOST-ONLY NETWORKING**



```
Vagrant::Config.run do |config|
  config.vm.define :web do |web_config|
    web_config.vm.box = "lucid32"
    web_config.vm.network "33.33.33.10"
  end

  config.vm.define :db do |db_config|
    db_config.vm.box = "lucid32"
    db_config.vm.network "33.33.33.11"
  end
end
```



# AVOID RRR'S

- Ranges Reserved by R- 192.168.0.x
- 10.0.0.x
- 33.33.33.x usually safe!



# WEB SERVER PROVISIONING NEEDS

- Run apt-get update to make sure we get the latest available packages for Ubuntu
- Install the JDK
- Install Apache Tomcat
- Set up a “fluffbox” user that will perform our automated deployments using Capistrano
- Add a public key to fluffbox’s authorized\_keys file to allow Capistrano to login without a password
- Add fluffbox to the groups necessary for writing to Tomcat’s webapps directory and for sudo usage
- Set up the base of the directory structure used by Capistrano to do its work



# DATABASE SERVER PROVISIONING NEEDS

- Run apt-get update to make sure we get the latest available packages for Ubuntu
- Install MySQL
- Create a fluffbox database
- Create a fluffbox user that can connect from our web/application server



# ASSEMBLE COMMUNITY COOKBOOKS

- apt
- java
- tomcat
- openssl
- database
- mysql



# CREATING COOKBOOKS

`knife cookbook create web_base`

`knife cookbook create db_base`



# **WEB\_BASE COOKBOOK CODE**



```
gem_package "libshadow" do
  action :install
end

user "fluffbox" do
  comment "Fluffbox Deployer"
  home "/home/fluffbox"
  shell "/bin/bash"
  password "$1$Z4H/cwr8$AAUjbPhfUy6xp0bPKMNQX."
  action :create
end

directory "/home/fluffbox" do
  owner "fluffbox"
  group "fluffbox"
  mode "0755"
  action :create
end
```

```
directory "/home/fluffbox/.ssh" do
  owner "fluffbox"
  group "fluffbox"
  mode "0755"
  action :create
end

directory "/u" do
  owner "fluffbox"
  group "fluffbox"
  mode "0755"
  action :create
end

cookbook_file "/home/fluffbox/.ssh/authorized_keys" do
  source "fluffbox/authorized_keys"
  mode "0644"
end
```

```
group "tomcat6" do
  members ['tomcat6', 'fluffbox']
end

group "admin" do
  members ['vagrant', 'fluffbox']
end
```

# **DB\_BASE COOKBOOK**

## **CODE**



```
gem_package "mysql" do
  action :install
end

# create connection info as an external ruby hash
mysql_connection_info = {:host => "localhost",
  :username => 'root',
  :password => node['mysql']['server_root_password']}

# create a mysql database
mysql_database 'fluffbox' do
  connection mysql_connection_info
  action :create
end

mysql_database_user 'fluffbox' do
  connection mysql_connection_info
  password 'fluffbox'
  database_name 'fluffbox'
  host '33.33.33.11'
  action :grant
end
```

# **FINAL VAGRANTFILE CODE**



```
Vagrant::Config.run do |config|
  config.vm.define :web do |web_config|
    web_config.vm.box = "lucid32"
    web_config.vm.network "33.33.33.11"

    web_config.vm.provision :chef_solo do |chef|
      chef.cookbooks_path = "cookbooks"

      chef.add_recipe "apt"
      chef.add_recipe "java"
      chef.add_recipe "tomcat"
      chef.add_recipe "web_base"
    end
  end
end
```

```
config.vm.define :db do |db_config|
  db_config.vm.box = "lucid32"
  db_config.vm.network "33.33.33.10"

  db_config.vm.provision :chef_solo do |chef|
    chef.cookbooks_path = "cookbooks"

    chef.add_recipe "apt"
    chef.add_recipe "openssl"
    chef.add_recipe "mysql::server"
    chef.add_recipe "db_base"

    chef.json.merge!({
      :mysql => {
        :server_root_password => "supersecure",
        :bind_address => "33.33.33.10"
      }
    })
  end
end
end
```

**FIRE IT UP AND  
DEMO THE  
DEPLOYMENT!**



# SUMMARY

- Why Vagrant?
- Installation/Prerequisites
- Your first Vagrant
- Boxes/Base Boxes/VeeWee
- Commands
- Vagrantfile Anatomy
- Provisioning
- Case Study



# CODE SAMPLES

<https://github.com/mstine/fluffbox-rwx> (vagrant branch)

<https://github.com/mstine/vagrant-nfjsmag>





# PLEASE FILL OUT YOUR EVALUATIONS!

Matt Stine

[matt.stine@gmail.com](mailto:matt.stine@gmail.com)

<http://www.mattstine.com>

Twitter: mstine