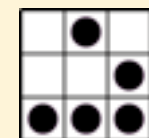


Use datacenter tools to make your dev life easier

Jochen Lillich



freistil.it

About me

Jochen Lillich

@geewiz

*** 1970**

Linux Sysadmin & Trainer

IT Manager at WEB.DE, 1&1

Founder & CTO at freistil IT

freistil IT

<http://freistil.it>

@freistil

“Ops for Devs”

DrupalCONCEPT: Drupal IaaS

Challenge...

Local development environments

Manual configuration effort

Mismatched software versions

Long onboarding times

...accepted!

Minimize setup time

Make Development = Production

Reproduce setup consistently on demand

Vagrant



“Vagrant uses Oracle’s VirtualBox to build configurable, lightweight, and portable virtual machines dynamically.”

<http://vagrantup.com>

Instant VM

```
$ gem install vagrant
```

```
$ vagrant box add precise64 http://files.vagrantup.com/precise64.box
```

```
$ cd ~/myvm
```

```
$ vagrant init precise64
```

```
$ vagrant up
```

```
$ vagrant ssh
```

Box

Pre-installed OS

Available from <http://files.vagrantup.com>

Box – Management

```
$ vagrant box -h
```

```
$ vagrant package -h
```

Box – DIY

VeeWee by Patrick Dubois

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

Vagrantfile

Ruby DSL for VM configuration

Can be used for many VMs

Vagrantfile

```
Vagrant::Config.run do |config|  
  config.vm.box = "precise64"  
  config.vm.boot_mode = :gui  
  config.vm.forward_port 80, 8080  
end
```

**“Well, I’ve been doing that for ages with snapshots/
images/other...”**



“No matter how complex the realities of your business, Chef makes it easy to deploy servers and scale applications throughout your entire infrastructure.”

<http://www.opscode.com/chef/>

CM with Chef

Ruby DSL for system administration tasks

Makes only necessary changes (Idempotence)

Infrastructure as Code

Cookbooks

Nodes

Recipes

Attributes

Resources

Roles

Providers

Resources

Package

Command, Script

Service

Cron

File, Directory

Git

Template

...

Example

```
package "apache2" do
  action :install
end

service "apache2" do
  action [:enable, :start]
end
```

Example (cont.)

```
template "/etc/apache2/apache2.conf" do
  source "apache2.conf.erb"
  owner "root"
  group "root"
  mode 0644
  notifies "reload", "service[apache2]"
end
```

Community sites

[**http://wiki.opscode.com**](http://wiki.opscode.com)

[**http://community.opscode.com**](http://community.opscode.com)

[**http://github.com/opscode-cookbooks**](http://github.com/opscode-cookbooks)



+



Installation

```
$ gem install librarian  
$ cd ~/myvm  
$ librarian-chef init  
$ $EDITOR Cheffile  
$ librarian-chef install
```

Chefffile

```
site "http://community.opscode.com/api/v1"  
cookbook 'apache2'  
cookbook 'php'  
cookbook 'mysql'
```

roles/drupal.rb

```
name "drupal"
description "Drupal dev VM"

run_list(
  "recipe[apache2]",
  "recipe[apache2::mod_expires]",
  "recipe[apache2::mod_rewrite]",
  "recipe[apache2::mod_php5]",
  "recipe[php]",
  "recipe[php::module_mysql]",
  "recipe[mysql::client]",
  "recipe[mysql::server]"
)
```

Vagrantfile

```
config.vm.provision :chef_solo do |chef|  
  chef.cookbooks_path = "cookbooks"  
  chef.roles_path = "roles"  
  chef.add_role "drupal-dev"  
end
```

Kitchen magic

\$ vagrant provision

**Now do some
awesome stuff!**

And finally...

\$ vagrant destroy

Tips

Keep all building blocks local:

Cache packages

Store your own base boxes

Use a central cookbook repository

Conclusion

Combining Vagrant and Chef gives us:

Dedicated, consistent, disposable dev setups

Efficient onboarding of new devs

Thank you!

Please rate my session:
<http://bit.ly/dc12chef>