Using Vagrant for Magento development

Alexander Turiak, @HexBrain

$ whoami

- Magento developer since 2011
- (Tries to be) Active in Magento community
- Co-founded HexBrain in 2013

Key points

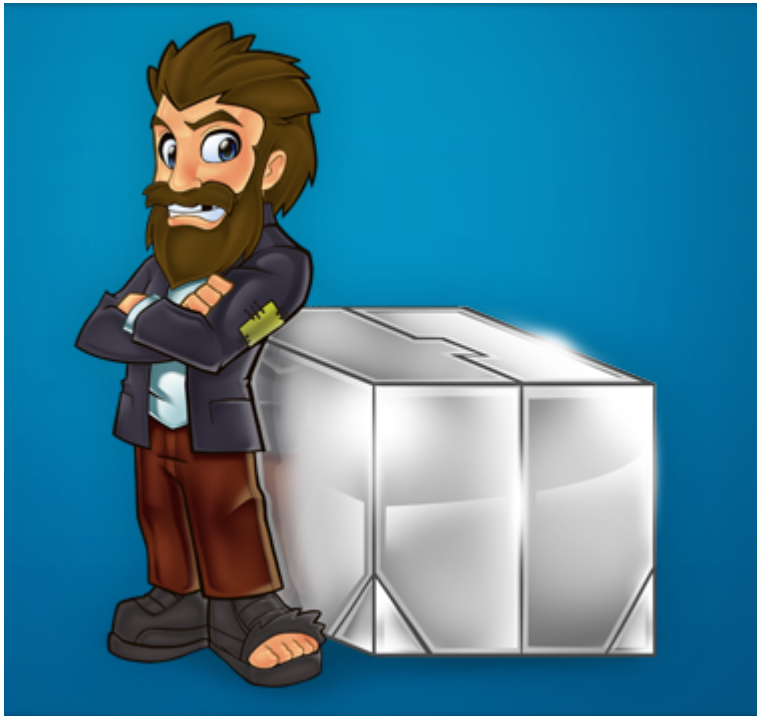
- What is Vagrant (boxes, virtual machines, provisioners)
- Why should you use Vagrant
- Magento specific use cases

What is Vagrant?

Is a tool for creating, deploying, sharing and managing virtual environments for solving development and testing tasks

Created by Mitchell Hashimoto in 2010

A wrapper around virtualization software and configuration management software




VAGRANT

Main issue it solves is “Works on my machine” providing environment consistency which can be shared same way as you share code e.g. using git

What’s in the “box”?

Box is the Vagrant specific package format, each box is tied to a specific provider

Providers

Out of the box Vagrant supports Virtual Box

Other providers (supported via plugins):

- VMware (commercial)
- AWS, Digital Ocean, Rackspace, Open Stack
- Docker

Provisioners

- Shell
- Puppet
- Chef
- Ansible

```
Vagrant.configure("2") do |config|
  config.vm.provision "shell", path: "script.sh"
end
```

Synced folders

- NFS
- SMB
- rsync

```
config.vm.synced_folder ".", "/vagrant", type: "rsync", rsync__exclude: ".git/"
```

Quick setup

```
vagrant box add precise32 http://files.vagrantup.com/precise32.box
vagrant init precise32
vagrant up
```

PuPHPet or Puppet

PuPHPet — is a GUI configuration tool for Vagrant. It uses Puppet for packages installation and configuration

https://puphpet.com/

What is Puppet?

It’s a tool which allows management of software packages, their configuration and the operating system itself

Just like Vagrant it’s written in Ruby.
Alternatives: Ansible (Python), Chef (Ruby)

Vagrant & PuPHPet

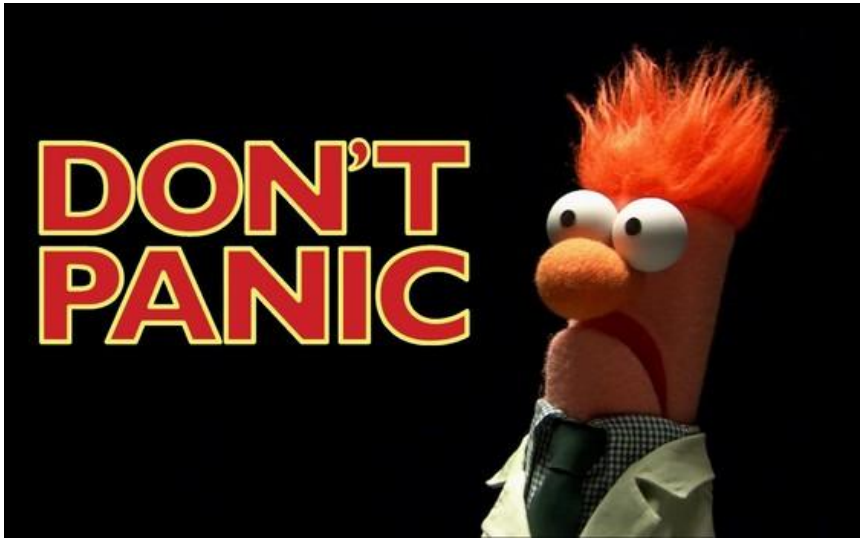
How does it work?

```
configValues = YAML.load_file("#{dir}/puphpet/config.yaml")
```

PuPHPet. Have you stuck?

What to do in case it doesn’t support adding a package you need?

This is Puppet



Adding NodeJS

1. puphpet/puppet/Puppetfile

```
mod 'nodejs', :git => 'https://github.com/willdurand/puppet-nodejs.git'
```

2. puphpet/puppet/manifest.pp

```
class { 'nodejs':
  version    => 'stable',
  target_dir => '/bin',
}
```

3. vagrant provision

Life after `vagrant up`

1. /etc/hosts
2. vagrant ssh
3. DB connect
4. Xdebug

What about Magento?

- n98-magerun
- modman
- modgit

vagrant ssh -c "cd /var/www/magento && n98-magerun.phar cache:clear"

Testing Varnish

To test several sites for Varnish implementation it's convenient to setup a box running Varnish on separate ports and route the traffic further to web server instances

Magento 2

There is a box built by Rolando Granadino most commonly known as @beeplogic https://github.com/rgranadino/mage2_vagrant which allows to quickly install Magento 2 with all it dependencies along with sample data ported by Marius Strajeru from Magento 1

Docker

It uses feature of Linux systems called containers allowing to spawn resources like file system, memory etc. of host machine to build separate isolated services ultimately having complex cloud like infrastructures locally

Released in 2013 by Solomon Hykes, allows deploying applications into isolated containers

Frequently used with lightweight CoreOS based on Linux kernel

Vagrant + Docker

http://docs.vagrantup.com/v2/provisioning/docker.html

Allows to use power of Vagrant with efficiency of Docker

https://github.com/cmuench/Magento-Vagrant-Puppet-Nginx

https://github.com/amacgregor/MageVagrant

https://github.com/EcomDev/vagrant-magento

https://github.com/r8/vagrant-lamp

Pros and cons

+ Configuration flexibility
+ Portability
+ Setup and reconfiguration speed
+ Testing code with different versions of software (PHP 5.4, 5.5, 5.6, HHVM)

- Overhead
- Disk space for each virtual machine (2-5 GB)

Links

http://vagrantup.com/

http://vagrantcloud.com/

http://puppetlabs.com/

http://puphpet.com/

http://docker.io/

Thank you for attention!

Mulțumesc!