Building with Virtual Development EnvironmentsA

Oscar Merida
July 2015

What will we look at?A

- Benefits of using a Virtual MachineA
- Setting up a virtual environmentA
- Updating your workflowA

Who uses…A

- XAMP, MAMP?A
- Acquia Dev Desktop?A
- Homebrew, Ports, etc?A
- Shared development environment?A
- Production?!A

Early Tools were CrudeA

- “One-size-fits-all”A
- Difficult to customize per site, or may not support multiple projectsA
- Tricky to updateA
- Inconsistent across individualsA

Projects are special snowflakesA

- Different PHP versions / extensionsA
- Different databases / versionsA
- What about additional servicesA
  - Redis?A
  - Memcached?A

Common pitfallsA

- Mismatches between development, staging, QA, & production environmentsA
  - bugs that “only happen on live”A
- Setup can be tedious and fragileA
- Setup is poorly documentedA

VirtualizationA

- Use a Virtual Machine (VM) identical to production environment.A
- Automate setting up the VM, aka “Provisioning”A
- Share setup more easily with collaborators.A

A nice bonusA

- Documenting how to setup a VM provides step-by-step guide to setting any instance.A
  - Every step and setting is in a configuration file.A
- Great for client handoffA

Virtualization?A

- “Hardware virtualization or platform virtualization refers to the creation of a virtual machine that acts like a real computer with an operating system. Software executed on these virtual machines is separated from the underlying hardware resources. For example, a computer that is running Microsoft Windows may host a virtual machine that looks like a computer with the Ubuntu Linux operating system; Ubuntu-based software can be run on the virtual machine.”A
- http://en.wikipedia.org/wiki/VirtualizationA

VirtualboxA

- Free & Open Source Virtualization softwareA
  - https://www.virtualbox.orgA
- Runs on Windows, Linux, Mac (+more)A
  - https://www.virtualbox.org/wiki/DownloadsA
- There's also VMWare, Parallels, and others…A

HardwareA

- Memory helps — a lotA
  - need enough to dedicate to the Guest OSA
- Disk spaceA
  - use files to represent and persist virtual hard disk storageA

VagrantA

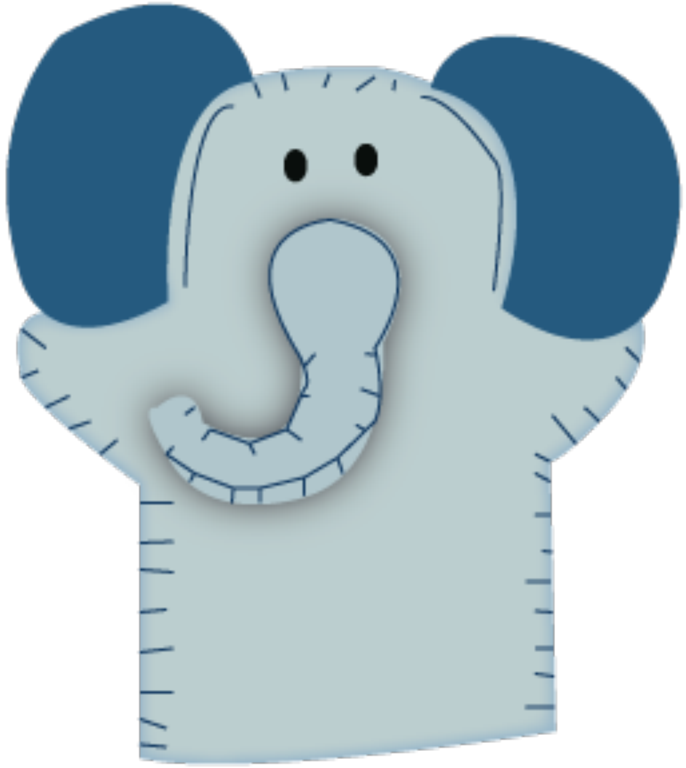
- "A tool for building complete development environments"A
- https://www.vagrantup.com/A
- A Vagrantfile + provisioner automatically configures a VM Box for you.A
  - … but someone has to make itA

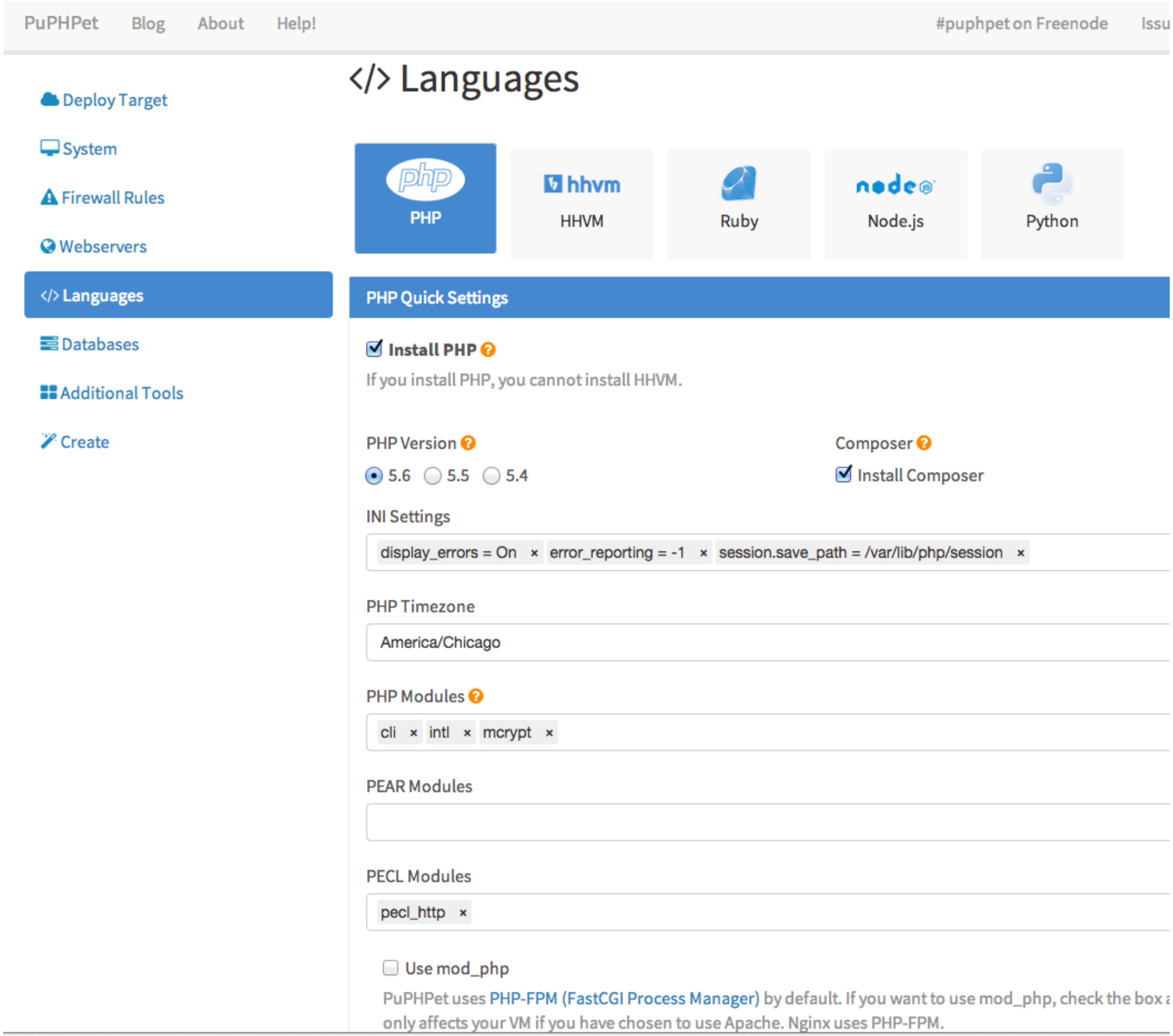
OK, let’s get to the good stuff already!A

- Assuming you’ve installed Vagrant & Virtualbox…A
- What’s the easiest way to get a VM working?A

PuPHPet (puffet)A

- Online GUI to create Virtual Machines A
- Choose base OS, PHP, Mysql, and moreA
- Download a zip file with Vagrantfile and supporting filesA





</> Languages

PHP Quick Settings

☑ **Install PHP**

If you install PHP, you cannot install HHVM.

PHP Version

● 5.6 ○ 5.5 ○ 5.4

Composer

☑ Install Composer

INI Settings

display_errors = On × error_reporting = -1 × session.save_path = /var/lib/php/session ×

PHP Timezone

America/Chicago

PHP Modules

cli × intl × mcrypt ×

PEAR Modules

PECL Modules

pecl_http ×

□ Use mod_php

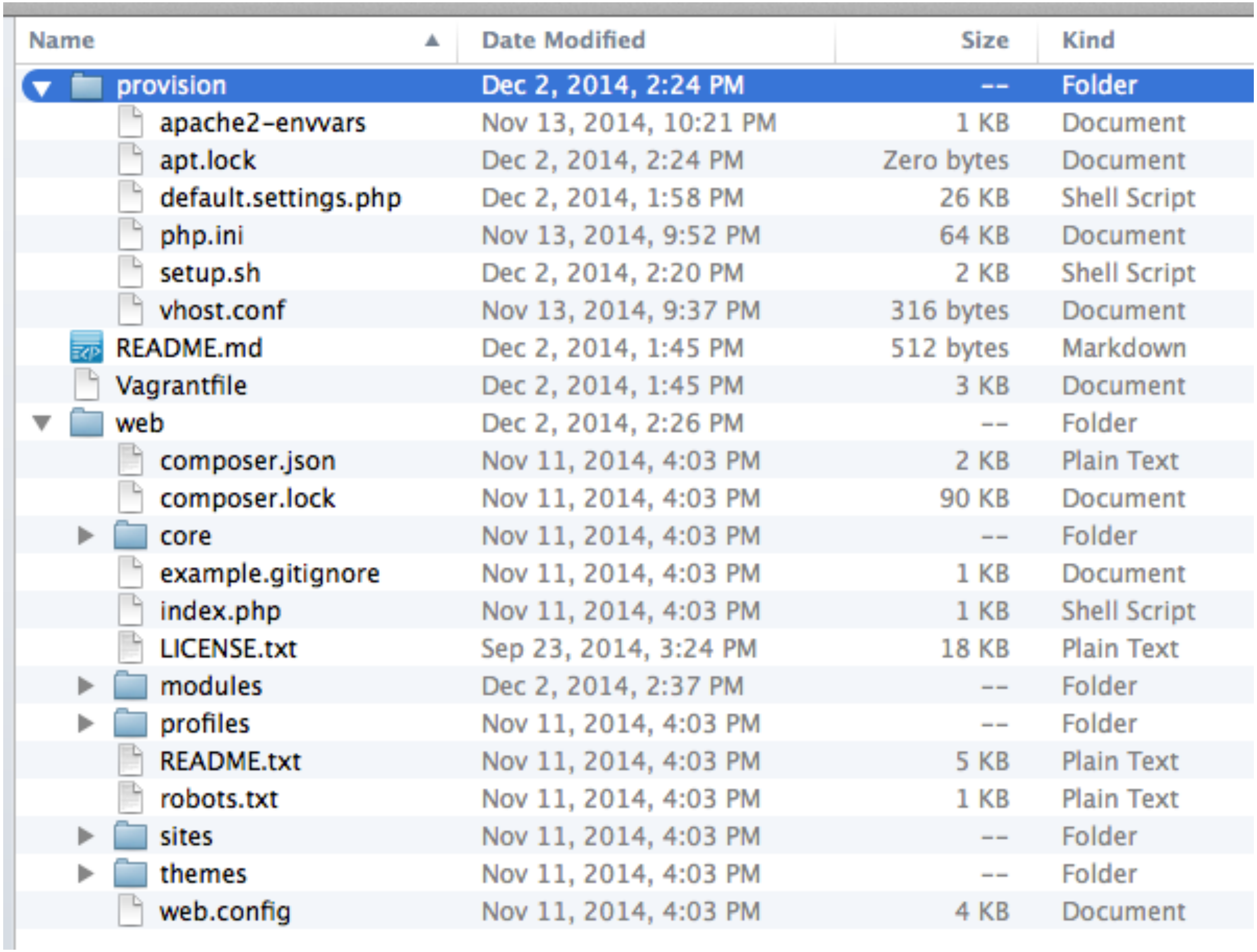
PuPHPet uses PHP-FPM (FastCGI Process Manager) by default. If you want to use mod_php, check the box a
only affects your VM if you have chosen to use Apache. Nginx uses PHP-FPM.

Using your VagrantfileA

- **Tip: extract the zip file to your project's root directoryA**
- Basic vagrant commands:A
  - vagrant up - start up the VMA
  - vagrant ssh - SSH to the VMA

Other Vagrant commandsA

- vagrant suspend - put the VM to sleep, use this between work sessions.A
- vagrant stop - turn off the VM.A
- vagrant destroy - delete the VM, CAUTION!A



Name ▲	Date Modified	Size	Kind
▼ provision	Dec 2, 2014, 2:24 PM	--	Folder
apache2-envvars	Nov 13, 2014, 10:21 PM	1 KB	Document
apt.lock	Dec 2, 2014, 2:24 PM	Zero bytes	Document
default.settings.php	Dec 2, 2014, 1:58 PM	26 KB	Shell Script
php.ini	Nov 13, 2014, 9:52 PM	64 KB	Document
setup.sh	Dec 2, 2014, 2:20 PM	2 KB	Shell Script
vhost.conf	Nov 13, 2014, 9:37 PM	316 bytes	Document
README.md	Dec 2, 2014, 1:45 PM	512 bytes	Markdown
Vagrantfile	Dec 2, 2014, 1:45 PM	3 KB	Document
▼ web	Dec 2, 2014, 2:26 PM	--	Folder
composer.json	Nov 11, 2014, 4:03 PM	2 KB	Plain Text
composer.lock	Nov 11, 2014, 4:03 PM	90 KB	Document
▶ core	Nov 11, 2014, 4:03 PM	--	Folder
example.gitignore	Nov 11, 2014, 4:03 PM	1 KB	Document
index.php	Nov 11, 2014, 4:03 PM	1 KB	Shell Script
LICENSE.txt	Sep 23, 2014, 3:24 PM	18 KB	Plain Text
▶ modules	Dec 2, 2014, 2:37 PM	--	Folder
▶ profiles	Nov 11, 2014, 4:03 PM	--	Folder
README.txt	Nov 11, 2014, 4:03 PM	5 KB	Plain Text
robots.txt	Nov 11, 2014, 4:03 PM	1 KB	Plain Text
▶ sites	Nov 11, 2014, 4:03 PM	--	Folder
▶ themes	Nov 11, 2014, 4:03 PM	--	Folder
web.config	Nov 11, 2014, 4:03 PM	4 KB	Document

Ooooh, A VM…A

- working with filesA
- viewing your development site in a browserA
- running drushA

Mapped foldersA

- By default Vagrant maps your project's folder on the Host OS toA /vagrant on the Guest OS.A
- For exampleA
  - If your project is /home/omerida/drupal8
  - The same files are in /vagrant on your VMA

File permissions gotchaA

- Need to insure web server can write to /sites/default/filesA
  - mount with “anyone can write” file permissionsA
  - run apache as the vagrant userA
  - add apache to the vagrant groupA
- https://www.drupal.org/node/2055947A

This means…A

- Work on PHP, CSS, image, and other files locally with your favorite IDE, text editor, and other programs.A
- … and the changes are automatically reflected in your Virtual EnvironmentA
- I've found default shared maps fine for many PHP projects.A
- For Drupal, consider using NFS but that involves more setup.A

Browsing your siteA

- Your Vagrantfile will setup networking so that the VM is accessible from the Guest OS, Usually a private address like 192.168.33.10A
- Add this IP to your /etc/hosts fileA
    - `192.168.33.10   project.dev`
    - On windows Hosts file is trickier to find.A
    - You'll need admin privileges to editA
    - Use vagrant-hostsupdater plugin to automate.A
- Go to "http://project.dev" in your browserA

Running drushA

- vagrant ssh to login to the VMA
- > cd /webA
- > drush statusA
- Can install drush from your Guest OS package repositoryA
- Or setup drush aliases to work on remote instance.A

Vagrant Drupal DevelopmentA

- https://www.drupal.org/project/vddA
- Fully configured Linux based VMA
  - Ubuntu 12.04 LTSA
- http://www.drupalvm.com/A
  - Ubuntu 14.04A
  - Optional components - Solr, Selenium, moreA

Collaboration tipsA

- One person responsible for initial setup and configurationA
- Commit your Vagrantfile + related files to your code repositoryA
- Standardize on Virtualbox & Vagrant versionsA

ProvisionersA

- You might outgrow PuPHPet (or phansible.com)A
- Various options for automating configurationA
  - PuppetA
  - ChefA
  - AnsibleA
  - Shell ScriptsA

Other Use CasesA

- Will my code run with PHP 5.6? with PHP7?A
- What if I switch Mysql for Percona, MariaDB, etcA
- Redis or memcached for caching?A
- How do I setup Varnish for my site?A

Testing Drupal 8A

- Get Drupal 8 running:A
  - https://github.com/omerida/drupal8-vmA
  - https://github.com/SandyS1/d8ansibleA

Thank You!A

- Twitter: @omeridaA
- Editor-in-Chief php[architect] magazineA
  - plus Conferences, Training, and booksA
  - www.phparch.comA
- Any Questions?A