MediaWiki-Vagrant
What is Vagrant?

Vagrant is a lightweight, reproducible, and portable development environment.

Say goodbye to the "works on my machine" excuse [...]
What is MediaWiki-Vagrant?

MediaWiki-Vagrant is a (slightly less) lightweight, (but very) reproducible, and portable development environment for MediaWiki related projects.
Lightweight

Clone the MW-Vagrant repo, setup, and go.

$ git clone ...
$ ./setup.sh
$ vagrant up
Reproducible

Puppet **provisioner** transitions the state from a stock Ubuntu image to a fully functional MediaWiki environment.
Portable

VirtualBox provider achieves consistent virtualization on all host platforms.

Lighter weight providers may be possible for Linux users (LXC, Docker, etc.).
Roles

MediaWiki-Vagrant provides a flexible role system.

$ vagrant roles list
$ vagrant roles info [role]
Roles

Easily switch functionality on/off.

$ vagrant roles enable <role>
$ vagrant roles disable <role>
A Brief Look Under the Hood

Ruby
Puppet
Hiera
MediaWiki

…

“Please, make them stop…”
All-beef patty, lettuce, tomatoes…
config.vm.provider :virtualbox do |vb, override|
  override.vm.box = 'trusty-cloud'
  override.vm.box_url = 'https://.../trusty.box'
end

config.vm.provision :puppet do |puppet|
  puppet.manifests_path = 'puppet/manifests'
  puppet.hiera_config_path = 'puppet/hiera.yaml'
end

Bootstraps the guest environment.
Puppet

Declares what things are possible.

class mediawiki($dir, $branch, $server_url, ...) {
    git::clone {'mediawiki/core': ...
    php::composer::install { $dir: ...
}

define mediawiki::wiki($wiki_name, $db_name, ...) {
    # ...
}
Hiera

Declares the **desired state** of all the things.

```yaml
---
classes:
  - '::role::mediawiki'
  - '::role::mobilefrontend'

mediawiki::wiki_name: devwiki
mediawiki::dir: /vagrant/mediawiki
mediawiki::db_name: wiki
```
Roles

Tie it all together and make it easy.

$ vagrant roles enable mobilefrontend
$ vagrant provision
...
(make coffee, flame bd808 on irc, ... profit?)
So why care about internals?

Just a little bit of **know-how** and some elbow grease ...

- Customize your own setup
- Write new roles for your team
- Help us
Local Customizations

- MediaWiki configuration
- Vagrant configuration
- Puppet configuration
<?php
$wgShowSQLErrors = true;
$wgDebugDumpSql = true;
$wgDebugComments = true;
$wgDebugPrintHttpHeaders = true;
$wgDebugToolbar = true;
Vagrant configuration

$ vagrant config --list
# disable NFS sharing if you have problems
$ vagrant config nfs_shares no

$ vagrant forward-port --list
# access mysql server from host computer
$ vagrant forward-port 3306 3306
Vagrant configuration

Vagrantfile-extra.rb

```ruby
Vagrant.configure('2') do |config|
  config.vm.synced_folder '../extra', '/var/www/extra'
end
```

$ vagrant reload

$ open http://127.0.0.1:8080/extra/
Puppet configuration

- Customize the ~vagrant dotfiles
- Install software
- Make starting over painless

VMs should be treated like cattle, not pets.
class local::dotfiles {
  File {
    owner => 'vagrant',
    group => 'vagrant',
  }
  file { '/home/vagrant/.gitconfig':
    source => 'puppet:///modules/local/dotfiles/gitconfig',
  }
}
Puppet configuration

puppet/modules/local/files/dotfiles/gitconfig

[alias]
  exec = !/usr/bin/env  # executes relative to working copy root
git = !git           # for `git git foo` typos

r = review --verbose --no-rebase
rd = review --download
pfr = push origin HEAD:refs/for/master  # push for review
Puppet configuration

puppet/hieradata/local.yaml

---
classes:
  - local::dotfiles
Puppet configuration

$ vagrant provision

===> default: Running provisioner: lsb_check...
===> default: Running provisioner: puppet...
===> default: Running Puppet with site.pp...
===> default: Notice: Compiled catalog for mediawiki-vagrant.dev in environment production in 7.95 seconds
===> default: Info: Applying configuration version '1416616517.d7e81ba'
===> default: Info: mount[files]: allowing mediawiki-vagrant access
===> default: Notice: /Stage[main]/Local::Dotfiles/File[/home/vagrant/.gitconfig]/ensure: defined content as '{md5}91e12f05e80054596248f7df71b37db'
===> default: Notice: Finished catalog run in 16.44 seconds

$ vagrant ssh -- cat .gitconfig
Running Tests

“All code is guilty until proven innocent.” [citation needed]

Red, green, refactor.
Running Unit Tests

Helping with TDD.

$ vagrant run-tests [tests]
Running Unit Tests

Helping with TDD.

$ vagrant run-tests [tests]
Running Unit Tests

Helping with TDD.

$ vagrant ssh
v@mwv:~ $ cd /vagrant/mediawiki/tests/phpunit
v@mwv:phpunit $ make phpunit
Running Unit Tests

Helping with TDD.

v@mwv:phpunit $ make help
v@mwv:phpunit $ make list-groups
v@mwv:phpunit $ make [test group]
Running Browser Tests

Helping with BDD.

v@mwv:~ $ env | grep '\(MEDIAWIKI\|BROWSER\)'
MEDIAWIKI_API_URL=http://127.0.0.1/w/api.php
MEDIAWIKI_USER=Selenium_user
MEDIAWIKI_PASSWORD=vagrant
MEDIAWIKI_URL=http://127.0.0.1/wiki/
BROWSER=firefox
Running Browser Tests

Helping with BDD.

Option 1: headless mode
Option 2: X forwarding
Option 3: host side, targeting the guest
Running Browser Tests

Option 1: Use **headless** mode.

```bash
v@mwv $ cd ../../MobileFrontend/tests/browser
v@mwv $ export HEADLESS=true
v@mwv $ bundle exec cucumber feature/login.feature
```
Running Browser Tests

Option 2: Use X forwarding.

$ vagrant ssh -- -X
v@mwv $ cd ../MobileFrontend/tests/browser
v@mwv $ bundle exec cucumber features/login.feature
Running Browser Tests

Option 3: Run them on the **host**, targeting the guest wiki.

```
$ cd .../MobileFrontend/tests/browser
$ bundle install
$ export MEDIAWIKI_URL=http://127.0.0.1:8080/wiki/
$ export MEDIAWIKI_USER=Selenium_user
  (...)
$ bundle exec cucumber features/login.feature
```
Vagrant Sharing

Share **HTTP** or **SSH** access to your VM with others.

- Help with troubleshooting
- Quick feedback
- Demo a feature
- Requires a (free) Vagrant Cloud account
Vagrant Sharing

Share **HTTP** access to your VM with others.

```bash
$ vagrant share
=> default: Detecting network information for machine...
=> default: Checking authentication and authorization...
=> default: Creating Vagrant Share session...
=> default: URL: http://big-alpaca-5574.vagrantshare.com
```
Vagrant Sharing

Share **SSH** access to your VM with others.

```
$ vagrant share --ssh
=> default: Generating new SSH key...
    default: Please enter a password to encrypt the key: (password)
    default: Repeat the password to confirm: (password)
=> default: You're sharing with SSH access. This means that another
=> default: user simply has to run
=> default: `vagrant connect --ssh rapid-otter-4306`
```
Questions?
Links

- https://www.vagrantup.com/
- http://puppetlabs.com/
- https://docs.puppetlabs.com/hiera/1/
- https://www.virtualbox.org/
- https://www.mediawiki.org/wiki/MediaWiki-Vagrant
- https://www.mediawiki.org/wiki/MediaWiki
Credits

- "MediaWiki-logo" Licensed under Public domain via Wikimedia Commons
- "MediaWiki Vagrant" by TMg. Licensed under CC-BY-SA-3.0-de via Wikimedia Commons.
- "Vagrant" by Fco.plj - Own work. Licensed under CC BY-SA 3.0 via Wikimedia Commons.
- "Puppet Labs Logo" by This file was extracted from: cresapdx.com/2011/01/puppet-labs-case-study/. Licensed under Fair use via Wikipedia. Image cropped to “flask” icon only by BDavis (WMF) for use in this presentation.
- "WeirdTalesv36n1pg068 Shocked Woman" by Weird_Tales_volume_36_number_01.djvu; Weird Tales, Inc. derivative work: AdamBMorgan -Weird_Tales_volume_36_number_01.djvu. Licensed under Public domain via Wikimedia Commons.

- Syntax highlighting for some slides via http://markup.su/highlighter/ by lusever.

Copyright © 2014, Bryan Davis, Dan Duvall, and the Wikimedia Foundation.
This work is licensed under a Creative Commons Attribution-ShareAlike 3.0 Unported License.