Mentioned in this presentation

- http://virtualbox.org — virtual machine
- Vagrant — http://vagrantup.com
- Ansible http://docs.ansible.com
- http://phansible.com — generator for Vagrantfile and Ansible playbook

Vagrant quick ref

- vagrant init — create template Vagrantfile in current directory
- vagrant up — boot machine
  - And sets it up if this is our first time
- vagrant suspend — sleep
- vagrant halt — shut down
- vagrant reload — reboot
- vagrant ssh — log in to VM using SSH
- vagrant destroy — deletes our machine and its disks

Pre-existing environments:

- vdd (https://www.drupal.org/project/vdd)
- Undine (https://www.drupal.org/project/undine)
  - Based on Acquia Cloud (unofficial)
- https://www.drupal.org/project/vagrant
- http://vampd.io
- http://www.drupalvm.com

Easy, repeatable development environments using virtualization and provisioning

CHASE CATHCART

WRIGHT STATE UNIVERSITY

DRUPALCAMP OHIO

OCTOBER 23, 2015

How do you do things currently?

1. Edit live via WebFTP
2. drush run-server
3. Local full stack: LAMP, MAMP, WAMP, WIMP, ...
4. LAMP in a virtual machine

How does it go when you deploy a code change?

- "It worked on my machine."
- Are you matching your local configuration to production (including software versions)?

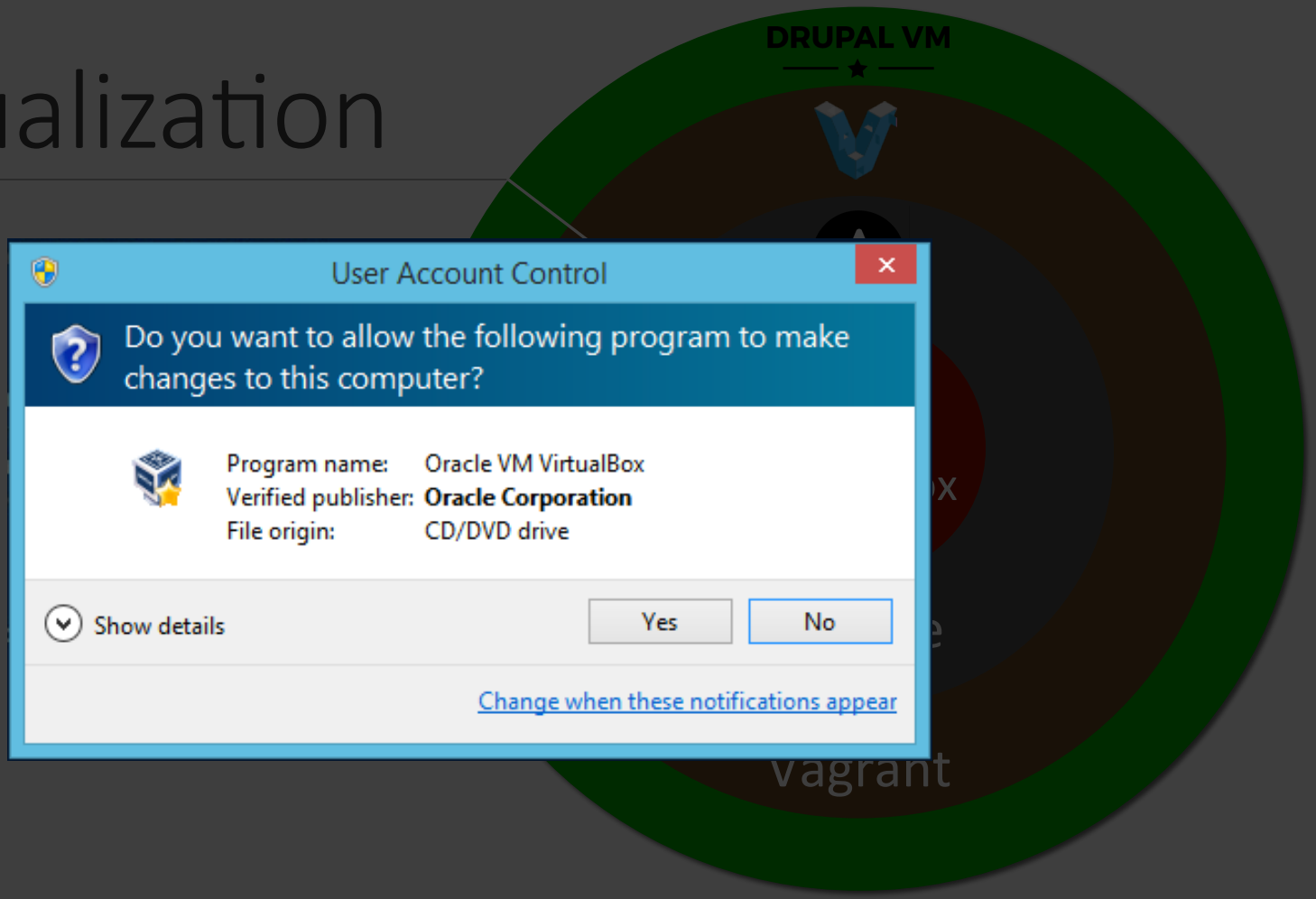
Keep Calm-o-Matic

Objectives

1. Isolate development environments for different servers from one another and from your local machine where possible
2. Make reproduction repeatable, quick, and easy
3. Reproduce our production environments, as closely as possible, locally

Virtualization

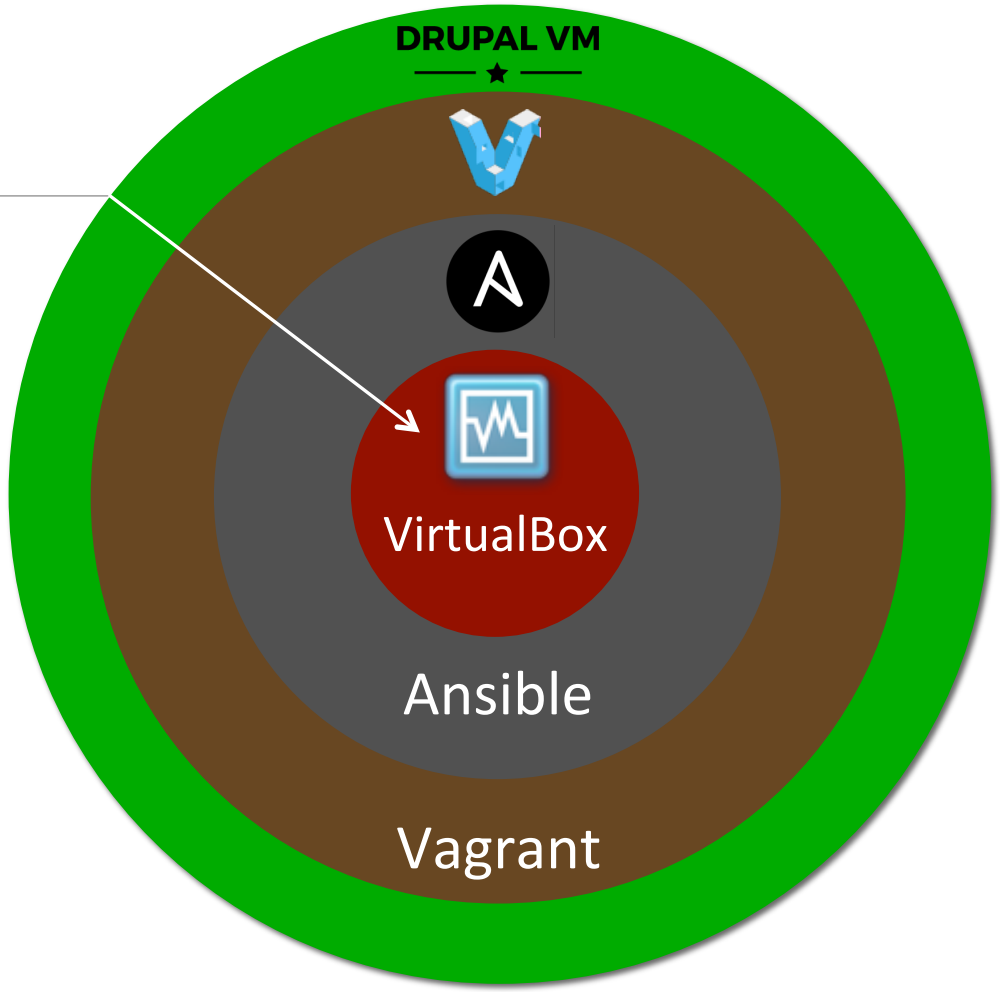
VirtualBox



virtualbox.org

Virtualization

VirtualBox



virtualbox.org

Provisioning

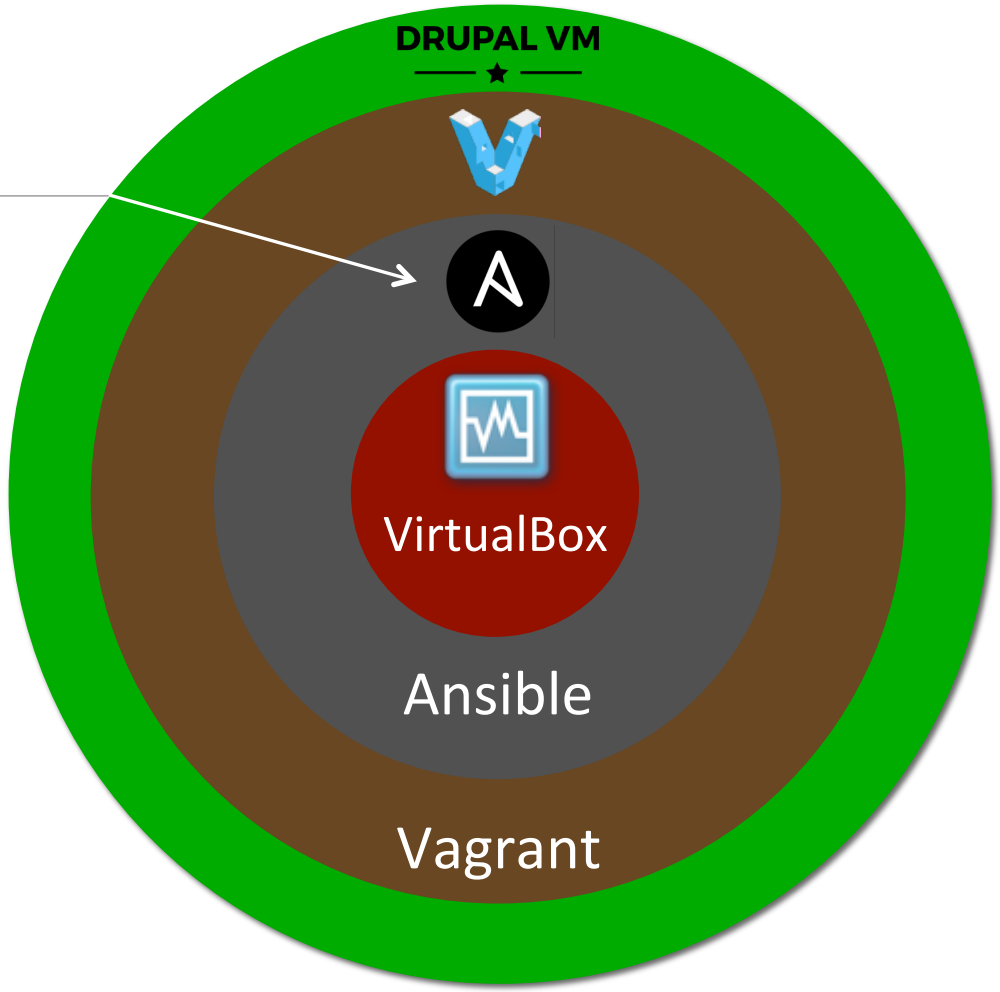
aka “IT Automation”

Ansible

- Others: chef, puppet, shell scripts

Runs a series of tasks

- Playbook



docs.ansible.com

User interface

Vagrant

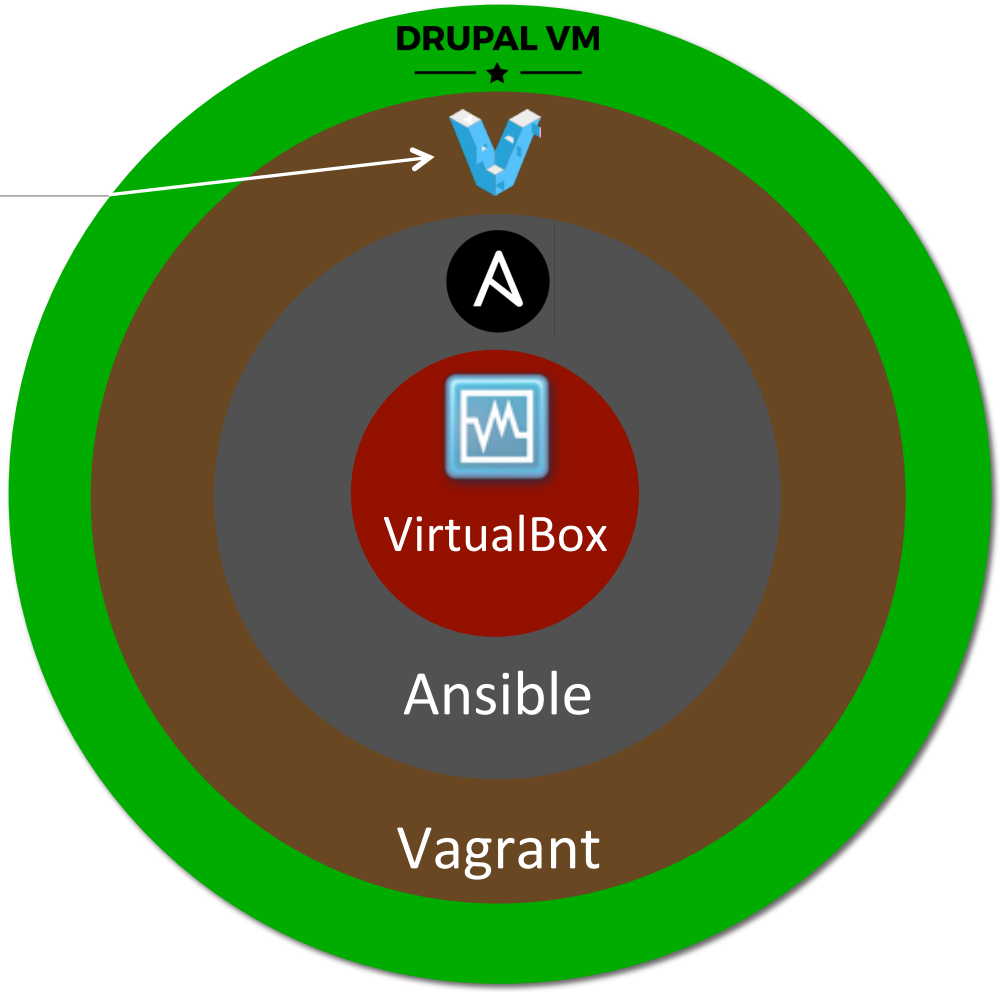
Vagrantfile: system configuration recipe

- Create VM, set parameters
- Install a base "box"—lightweight OS image
- Pass off to the provisioner once online

Provides commands for managing the VM

- vagrant up — boot machine, checking against the Vagrantfile
  - And sets it up if this is our first time
- vagrant suspend— sleep
- vagrant halt — shut down
- vagrant reload — reboot
- vagrant ssh — log in to VM using SSH

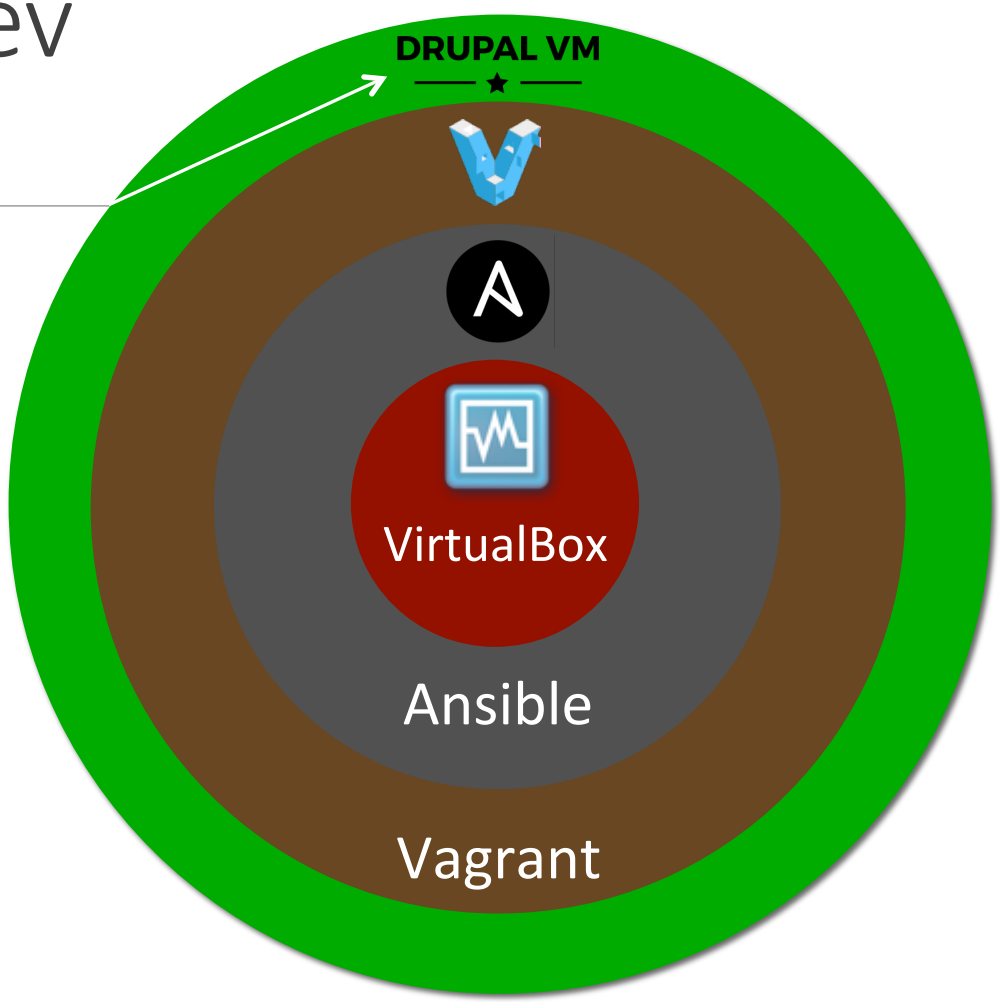
vagrantup.com



Off-the-shelf dev environments

Drupal VM
- @geerlingguy

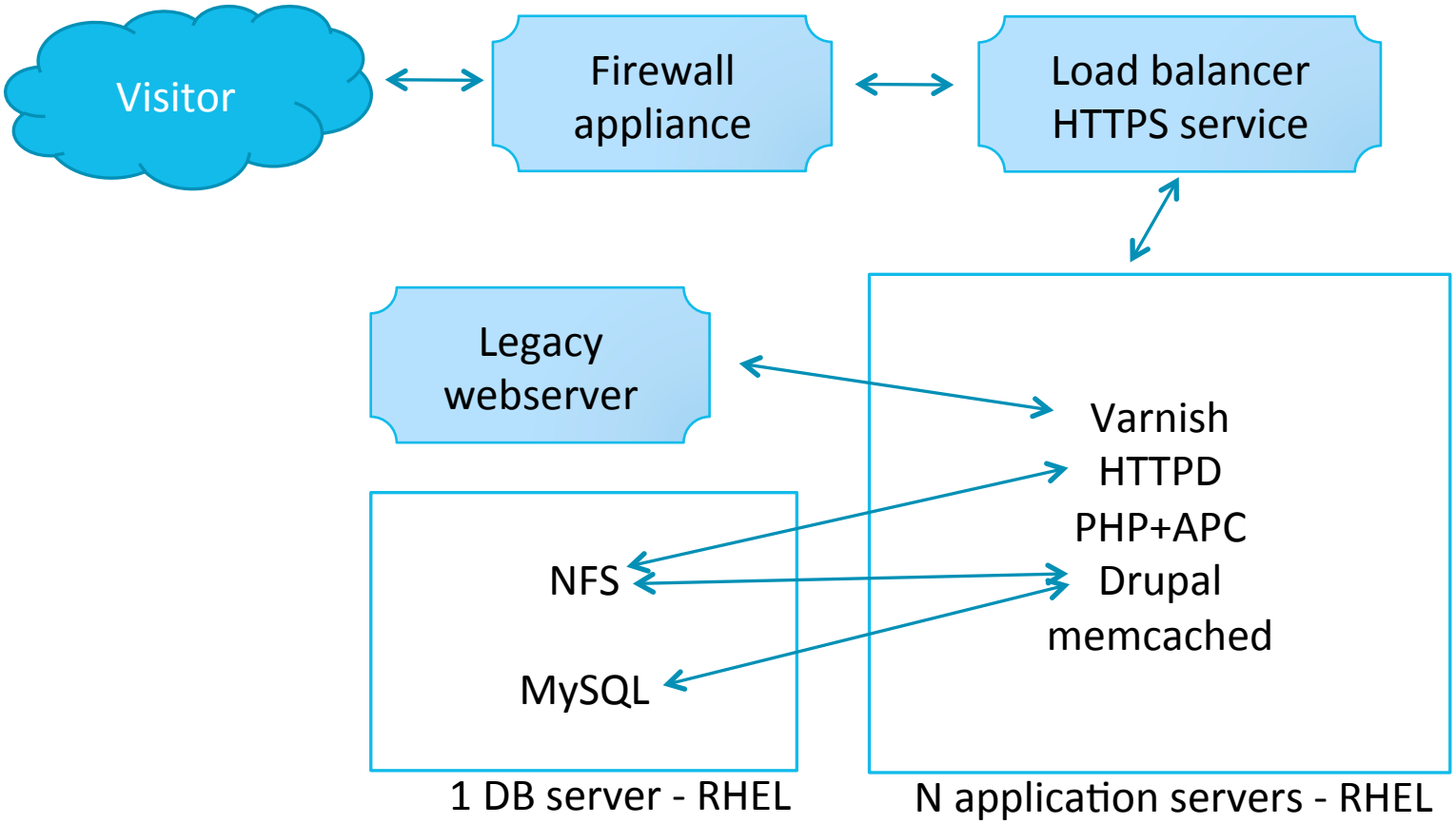
Others:
- Vagrant Drupal Development (VDD)
- Undine
  - Based on Acquia Cloud (unofficial)
- drupal.org/project/vagrant
- vampd.io



Objectives

1. Isolate development environments for different servers from one another and from your local machine where possible
2. Make reproduction repeatable, quick, and easy
3. Reproduce our production environments, as closely as possible, locally

Reproducing your production environment



Building your own environment

Vagrantfile

Provisioner file

- Playbook
- Something besides Ansible

Helpers

- Phansible (phansible.com) — Vagrantfile + playbook generator
- PuPHPet
- Vagrantbox.es and vagrantcloud.com — Boxes
- Ansible Galaxy — Contributed roles (collections of tasks and more)

Considerations

Which things matter?

- Versions, definitely (A.B.c)
- Equivalences can be ok (e.g., Red Hat => CentOS)
- You're not going to match every single thing up

Build out in stages

Windows

Technical things:

Networking

- Port forwarding or a private LAN?
- Access other devices (e.g., mobiles)
- VPN :-/

SELinux

Storage

- VirtualBox shared filesystem
- NFS
- Rsync

Steps to bliss, summarized

1. Install VirtualBox, Vagrant, Ansible
2. Make or download your development environment
   - It may have additional dependencies to install
3. 'vagrant up'
4. ...
5. Profit!
6. Screw something up? 'vagrant destroy'

Easy, repeatable development environments using virtualization and provisioning

CHASE CATHCART

WRIGHT STATE UNIVERSITY

DRUPALCAMP OHIO

OCTOBER 23, 2015