



VAGRANT

Development environments made easy
QC-LUG - 14 Jan 15

INTRODUCTION

- Chad Darnielle
- Email: cdarn@netins.net
- UNIX/Linux system administrator at a Global Fortune 500 company
- Started working with Linux starting with Red Hat 6.0 Hedwig (26 Apr 1999), kernel 2.2.5-15



WHY VAGRANT?

- Vagrant is computer software for creating and configuring virtual development environments. It can be seen as a wrapper around virtualization software such as VirtualBox, KVM, VMware and around configuration management software such as Ansible, Chef, Salt, or Puppet.

Wikipedia

WHY VAGRANT?

Preferred way to work on a web application?

Old Way

- Install and configure all of the software you need (e.g., Apache, PHP, MySQL, etc.) on your local development machine
- Multiple projects could be handled easily via virtual hosts and database isolation

New Way

- Modern web applications have a lot more moving parts, technology choices, and complexity

WHY VAGRANT?

Preferred way to work on a web application?

New Way

- New dynamic languages like JavaScript, Python, and Ruby
- New database choices like MySQL, PostgreSQL, Redis, Riak, Cassandra, etc.
- Additional considerations (e.g., multiple web servers, application servers , and back-end services) like Apache, Nginx, Unicorn, Thin, RabbitMQ, Solr, etc.

WHY VAGRANT?

Problems with continuing with the Old Way

- Keeping all of these technology combinations installed & properly configured quickly gets out of hand
- Software installs were often done manually. Some of the software is not trivial to install especially on Mac OS X and Windows if even available
- Configuration is difficult. Developers unlikely to know how production is configured. Misconfigured software can lead to functionality that works locally but not in production

WHY VAGRANT?

Problems with continuing with the Old Way


- Multiple projects are difficult to impossible because each project requires slightly different configurations or require completely different backend systems. The result is brittle setup of poorly configured dependencies or systems that run a lot of services that aren't necessary for individual web applications
- Difficult to keep development environments in sync with other team members or bring new team members on board because everyone has a separate environment. Result is confusion and long onboarding times

WHY VAGRANT?

Problems with continuing with the Old Way

- Difficult for multiple developers to use different operating systems. Some server software is very difficult or impossible to run on certain operating systems.

PROJECT INFORMATION

- Original author Mitchell Hashimoto
- Written in ruby programming language 
- MIT license
- Multiple providers and provisioners
- Supports plugins
- **Current stable release - 1.7.2 January 2015**
- Initial release - 0.1.0 March 2010
- October 2010 Engine Yard announced sponsorship of the project
- 1.0 March 2012
- Get full change history and project milestones below

THE TAO OF VAGRANT

- In the world of Vagrant, developers check out repository from a version control, run `vagrant up`, and have a full development environment with no human interactions
- Developers continue to work on their own machines with their own editors, browsers, and other tools
- Vagrant should be transparent and should match production as much as possible
- If something goes wrong or they want to start over from a clean state you run `vagrant destroy`, then `vagrant up`
- At the end of the day, Vagrant can suspend, halt, or destroy the development environment keeping the overall system clean.

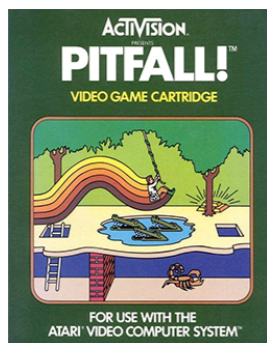


THE TERMINOLOGY OF VAGRANT

- **Providers** - Virtualization technology (VirtualBox, VMware[_desktop|_fusion], Docker, Hyper-V, AWS, etc.)
- **Provisioners** - Software that sets up the Vagrant VM (Shell scripts, Chef, Puppet, Ansible, Salt, etc.)
- **Boxes** - Base images upon which Vagrant environments are built

GETTING STARTED

- Recommended provider: **VirtualBox** because it's free and it's the most widely used provider
- VMware provider requires a license (\$79)
- VirtualBox & Vagrant run on Windows, Linux, and OS X
- VirtualBox - <https://www.virtualbox.org/wiki/Downloads>
- Vagrant - <https://www.vagrantup.com/downloads.html>



- **Avoid the VirtualBox / hardware pitfall!**
- VirtualBox requires VT-x or AMD-V CPU instructions if you are going to run x86_64 BOXES on top of a x86_64 host

DEMO TIME

- <https://docs.vagrantup.com/v2/getting-started/>

Overview

- Setup a project
- Install a BOX
- Connect to the BOX via SSH
- Synced Folders
- Provisioning via SHELL script
- Networking
- Share via HASHICORP ATLAS
- Teardown
- Rebuild

WRAPPING UP

- Main website
<http://vagrantup.com>
- GitHub
<https://github.com/mitchellh/vagrant>
- Discover Vagrant Boxes
<https://atlas.hashicorp.com/boxes/search>
- Google Group
<http://groups.google.com/group/vagrant-up>
- IRC
#vagrant on Freenode
- O'Reilly book - Vagrant: Up and Running
Print ISBN: 978-1-4493-3583-0 | ISBN 10:1-4493-3583-7
Ebook ISBN:978-1-4493-3582-3 | ISBN 10:1-4493-3582-9

WRAPPING UP

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