



Standardizing development environments

Vagrant + Puppet

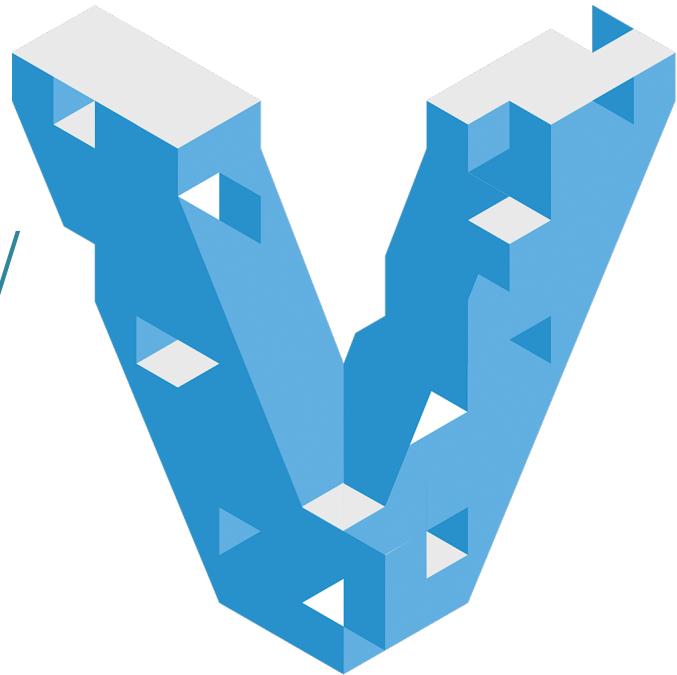
Open Repositories, June 2016

Liz Krznarich, Software Engineer/UI Design, ORCID

e.krznarich@orcid.org <http://orcid.org/0000-0001-6622-4910>

Vagrant: What is it?

- Tool for building portable/reproducible local environments using VMs
 - » Lightweight, headless VMs w/ config defined in code
- Awesome



VAGRANT

Vagrant: Why use it?

- Ensure devs have the same local environment
- Distribute a project *and* the environment needed to run it, all in one repo

-\(\cup\)-

IT WORKS
on my machine

Vagrant: Why use it?

- Loads of handy features:
 - » Disposable environments
 - » Nice CLI
 - » Synced folders
 - » Multi-machine
 - » Plugins
 - » Easy to get started!



Vagrant: Start using it!

1.  Install Vagrant

<https://www.vagrantup.com/downloads.html>

2.  Install VirtualBox

<https://www.virtualbox.org/wiki/Downloads>

3. Create a Vagrant file

```
Vagrant.configure("2") do |config|  
  
  # Base system to build from – see list at https://  
  # atlas.hashicorp.com/boxes/search  
  config.vm.box = "ubuntu/trusty64"  
  
  # Forward a port from host to guest so that you can run  
  # a server  
  config.vm.network "forwarded_port", guest: 80, host:  
  8080  
  
  # Sync a host directory to a location on the guest  
  config.vm.synced_folder "./my-project", "/var/www/html/  
  my-project"  
  
end
```

Vagrant: Terminology

- **Provider** Software providing your VM (Virtualbox, VMWare, etc)
- **Vagrantfile** Vagrant configuration file (goes in your project root)
- **Box** System specs for your VM
- **Provisioner** Tool that will perform setup tasks once the VM is running (Puppet, shell script, etc)

4. Fire it up!

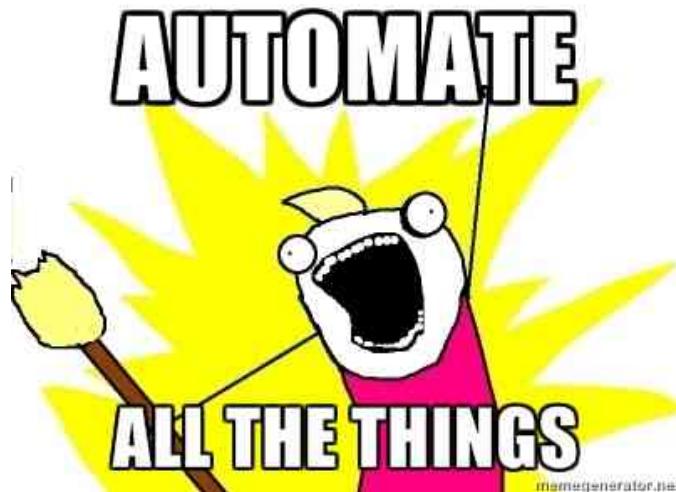
```
vagrant up
```

5. Access your machine

```
vagrant ssh
```

Great! But that's just a Linux box.
My project needs much more stuff.

You're in luck! Vagrant supports provisioners (like Puppet) that help to automate your setup.



Puppet: What is it?

Configuration management tool

- Standardize & declare system configuration - make it deployable across multiple machines
- Software, user/access mgmt, network config, databases...

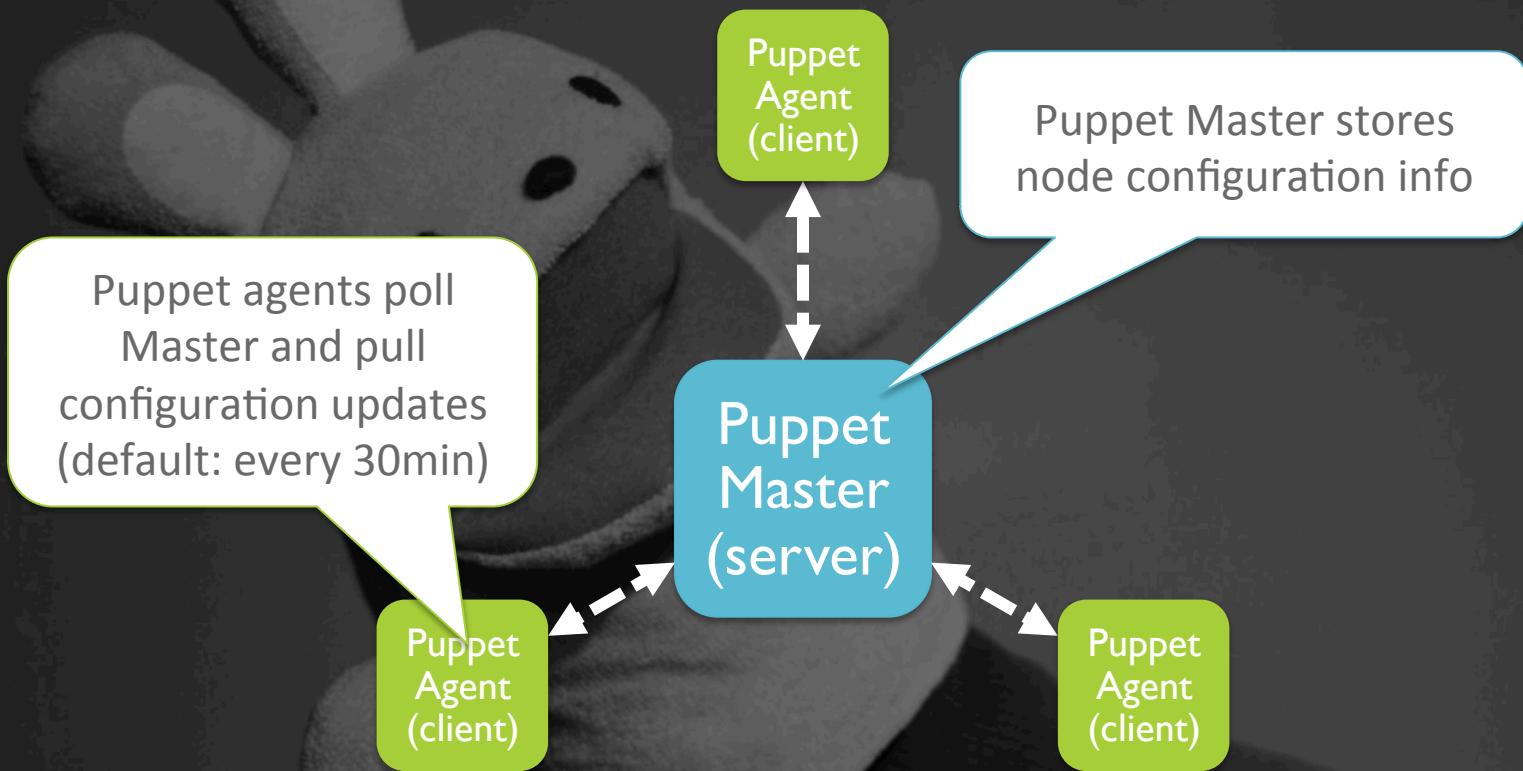


“Infrastructure as code”

Puppet: Nuts & Bolts

- **Open source** (also Enterprise version)
- **Written in Ruby** (but config is written in Puppet 'language')
- **Client-server architecture** (puppet master/puppet agent)

Puppet: Nuts & Bolts



Puppet: Why use it?

- Reproducible: Easily manage multiple machines/spin up new machines
- Config as code = works w/version control (easy to identify/fix problems)
- Keep users from mucking up configuration!
- And so much more...

Why Vagrant + Puppet?

- Specify config details that can't be included in a Vagrantfile
 - » Install applications
 - » Set up databases
 - » Add directories, users, etc
- Already using Puppet in production? Allow devs to 'Puppetize' in situ while writing new code



Puppet: Terminology

- **Manifests** Puppet config code goes here
- **Templates** Complex code that's tricky to write in manifests (Embedded Puppet or Embedded Ruby)
- **Modules** Extend the Puppet core / segment your Puppet code

I. Create a Puppet manifest (default.pp)

```
# specify default paths
Exec {
  path => [ "/usr/bin", "/bin", "/usr/sbin", "/sbin", "/usr/local/bin", "/usr/local/sbin" ]
}
# install packages
class install-packages {
  exec { 'apt-get update':
    command => 'apt-get update',
  }
  $packages = [ "apache2", "php5", "php5-mysql", "curl", "vim", "wget", "zip",
"unzip" ]
  package { $packages:
    ensure => "installed",
    require => Exec['apt-get update'],
  }
}
# start apache
class apache {
  service { "apache2":
    ensure => "running",
    require => Class["install-packages"],
  }
}
include install-packages
include apache
```

2. Enable the Puppet provisioner in your Vagrantfile

```
Vagrant.configure("2") do |config|
  config.vm.box = "ubuntu/trusty64"
  config.vm.network "forwarded_port", guest: 80, host: 8080
  config.vm.synced_folder "./my-project", "/var/www/html"

  # Enable the Puppet provisioner and specify the manifest
  # location
  config.vm.provision :puppet do |puppet|
    puppet.manifests_path = "puppet/manifests"
    puppet.options = [ '--verbose' ]
  end

end
```

3. Load the changes and run provisioner

```
vagrant reload --provision
```

4. Revel in the joy of reproducible configuration



Hot tips & minor pitfalls

- **Want to pull config from a remote Puppet master?**
Sure, Vagrant can do that!
- **Dependency headaches** Puppet doesn't execute in order; dependencies must always be declared
- **Installing a Ruby version higher than Puppet Ruby?**
Things get weird fast O_o
- ***nix VM on Windows using Synced Folders = #\$\$&*!**
(not specific to Puppet)

Get the code

<https://github.com/lizkrznarich/OR2016>

THANK YOU!

Questions? e.krznarich@orcid.org

<https://github.com/lizkrznarich>

<http://orcid.org/0000-0001-6622-4910>