Managing Drupal & Subsites with Puppet

```puppet
drupal::site { 'mysite':
    ensure => present,
}
```
Setting the Scene
Starting a new project

Think about your workflow when setting up a workspace for a new project

Some existing approaches:

- `mkdir -p sites/newsite.com/{files,modules,themes,...}
- **Drawbacks**
  - Sure everything will just work the same...
  - Oh wait, *what* version of PHP did you say you’re running?
  - Ok, let’s try downgrading the `date` module...
Setting the Scene

Starting a new project

Think about your workflow when setting up a workspace for a new project

Some existing approaches:

- Start from scratch:
  - Step 1: boot new virtual machine
  - Step 2: install apache, mysql, php
  - Step 3: ???
  - Step 4: Profit!

- Drawbacks:
I forgot to bump up the PHP `memory_limit`!
Setting the Scene
Starting a new project

Think about your workflow when setting up a workspace for a new project

Some existing approaches:

```bash
#!/bin/sh

yum install apache php php-gd mysql
tar -xvzf drush-7.x-5.9.tar.gz
echo "alias drush='php ~/drush/drush.php'" >> ~/.bash_profile
source ~/.bash_profile
cd /var/www/html
drush dl drupal
drush site-install standard --account-name=admin --account-pass=admin \
  --db-url=mysql://drupal:p@ssword@localhost/drupal
echo "Don't forget to make php.ini updates!"
```
Overview

A better workflow:

```bash
binford2k:~ ben$ vagrant up clientXXX
binford2k:~ ben$ vagrant ssh
[root@clientXXX ~]# puppet puppet agent -t
[...]
notice: /Stage[main]/Node[default]/Drupal::Site[mysite.example.com]/ensure: created
notice: /Stage[main]/Node[default]/Drupal_module[token]/ensure: created
notice: /Stage[main]/Apache/Service[httpd]/ensure: ensure changed 'stopped' to 'running'
notice: /Stage[main]/Apache/Service[httpd]: Triggered 'refresh' from 2 events
notice: Finished catalog run in 17.02 seconds
```
Overview

Three main topics of my talk

- Puppet enables a more efficient workflow
- Quick primer on using the Puppet language
- Using Puppet to manage Drupal sites
Puppet Workflow
class someclient {
    include apache, mysql, drupal

    drupal::site ('someclient.com':
        ensure => present,
    )

    user ('backup':
        ensure    => present,
        home      => '/var/spool/backups',
        managehome => true,
    )
}
Declarative Model

Describe the state you want.

Convergence
- gid => 'sysadmin'
- home => '/mnt/home/elmo'

Node State
- user { 'elmo':
  - ensure => present,
  - gid => 'staff',
  - home => '/home/elmo',
}

Desired State
- user { 'elmo':
  - ensure => present,
  - gid => 'sysadmin',
  - home => '/mnt/home/elmo',
  ...
}
Maintaining State

- You provision a node.
- Puppet configures it.
- Puppet maintains the desired state.
Puppet Language Primer
Puppet Resources

- Resources are building blocks.
- They can be combined to make larger components.
- Together they can model the expected state of your system.
Resource Abstraction Layer
Provides a consistent model for resources across supported platforms.

<table>
<thead>
<tr>
<th>File</th>
<th>Package</th>
<th>Service</th>
<th>User</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ruby</td>
<td>Apt</td>
<td>Redhat</td>
<td>Useradd</td>
</tr>
<tr>
<td>Yum</td>
<td>Launchd</td>
<td>SMF</td>
<td>Ldap</td>
</tr>
<tr>
<td>Gems</td>
<td></td>
<td>Debian</td>
<td>Netinfo</td>
</tr>
<tr>
<td>Deb</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RPM</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Puppet Classes

Classes define a collection of resources that are managed together as a single unit.

```plaintext
# /etc/puppetlabs/puppet/modules/ssh/manifests/init.pp

class ssh {
  package { 'openssh':
    ensure => present,
  }

  file { '/etc/ssh/sshd_config':
    ensure => file,
    owner => 'root',
    group => 'root',
    mode => '0644',
    source => 'puppet:///modules/ssh/sshd_config',
    require => Package['openssh'],
  }

  service { 'sshd':
    ensure => running,
    enable => true,
    subscribe => File['/etc/ssh/ssh_config'],
  }
}
```
Modules

Modules are directories that contain configuration.

- Designed to encapsulate everything related to a given configuration
- They have a hierarchy convention that enable the following:
  - auto-loading of classes
  - file-serving for templates and files
  - auto-delivery of custom Puppet extensions
  - easy sharing with others
Node Definitions

Multiple classes are declared together to represent a role.

For example, to build a web application on oscar.example.com:

```yaml
node 'oscar.example.com'
  include ssh
  include apache
  include mysql
  include web-app
```

```
Node
web1.example.com
- include ssh
- include ha_proxy_mbr
- include apache
- include web_app

Node
lb.example.com
- include ssh
- include ha_proxy

Node
db.example.com
- include ssh
- include mysql

Node
web2.example.com
- include ssh
- include ha_proxy_mbr
- include apache
- include web_app
```
Learning More

Many resources available for learning

- Learning Puppet tutorial
  - [http://docs.puppetlabs.com/learning/](http://docs.puppetlabs.com/learning/)
  - Seriously, go there

- Training classes
  - [http://puppetlabs.com/training/](http://puppetlabs.com/training/)
    - Fundamentals
    - Advanced
    - Extending Puppet

- IRC and all that jazz
  - #puppet
  - [http://groups.google.com/group/puppet-users](http://groups.google.com/group/puppet-users)
Managing Drupal With Puppet
Managing Drupal

A Drupal install consists of:

- Running services
  - A webserver
  - A database

- Packages installed
  - services and dependencies
  - PHP, extensions, and dependencies
  - Drupal itself
  - Drush (for sanity’s sake)

- Configuration
  - docroot
  - settings.php
  - php.ini
  - sites/*
Managing Drupal

But don’t reinvent the wheel

- Module enables management of multisite Drupal
- Manages package and configuration of default instance
- Resource types
  - subsites
  - variables
  - modules
  - themes
- Configuration
  - customize many aspects of site
Managing Drupal

Getting the module

http://forge.puppetlabs.com/binford2k/drupal

[root@training ~] puppet module search drupal
Searching http://forge.puppetlabs.com ...
NAME              DESCRIPTION                    AUTHOR        KEYWORDS
binford2k-drupal  This is a module that allows.. @binford2k  cms www drupal web ...

[root@classroom modules]# puppet module install binford2k/drupal
Preparing to install into /etc/puppetlabs/puppet/modules ...
Downloading from http://forge.puppetlabs.com ...
Installing -- do not interrupt ...
/etc/puppetlabs/puppet/modules
  └─┬ binford2k-drupal (v0.0.2)
      ├─┬ puppetlabs-apache (v0.6.0)
      │   └── puppetlabs-firewall (v0.3.0)
      └── puppetlabs-mysql (v0.6.1)
Managing Drupal

Using the module to manage Drupal

```yaml
node 'clientXYZ.dynamic.vm' {
  include apache, mysql

  host { 'mysite.example.com':
    ensure => present,
    ip => $::ipaddress,
  }

  class { 'drupal':
    managedatabase => true,
  }

drupal::site { 'mysite.example.com':
  database => 'mysite',
  dbuser => 'myuser',
  admin_password => 'derple',
  managevhost => true,
  managedatabase => true,
}

drupal_module { 'mysite.example.com::token':
  ensure => present,
}
}
```
Managing Drupal
Using the module to manage Drupal

MOTHER OF GOD
Managing Drupal
See it in action

BEST DEMO

EVER
Conclusion
Recap

Three main topics of my talk

- Puppet enables a more efficient workflow
  - describe what you want
  - let Puppet figure out how to make it happen
  - you do something that builds value

- Quick primer on using the Puppet language
  - http://docs.puppetlabs.com/learning/
  - http://puppetlabs.com/training/

- Using Puppet to manage Drupal sites
  - puppet module install binford2k/drupal
Questions?
Thanks for coming!