Managing Puppet Code

With R10k and Directory Environments

Chris Roddy

croddy@puppetlabs.com

Managing Puppet Code

- Evolution of Puppet environments
- Monolithic repositories
- R10k workflow
- Master configuration
- Q&A

Puppet Environments

What is an Environment?

> **Puppet environment:**
> An isolated collection of Puppet nodes, which a master can serve with completely different site manifests and modulepaths

Config-File Environments

Each environment configured explicitly in `puppet.conf`:

```
[main]
  certname = puppet.example.com
  modulepath = /etc/puppet/modules
  manifest = /etc/puppet/manifests/site.pp
  server = puppet.example.com

[agent]
  pluginsync = true
  environment = production

[master]
  reports = tagmail

[dev]
  modulepath = /etc/puppet/environments/dev/modules
  manifest = /etc/puppet/environments/dev/site.pp

[test]
  modulepath = /etc/puppet/environments/test/modules
  manifest = /etc/puppet/environments/test/site.pp

[preprod]
  modulepath = /etc/puppet/environments/preprod/modules
  manifest = /etc/puppet/environments/preprod/site.pp
```

Dynamic Environments

Environments configured generically using variables:

```
[main]
  certname = puppet.example.com
  modulepath = /etc/puppet/environments/$environment/modules
  manifest = /etc/puppet/environments/$environment/manifests/site.pp
  server = puppet.example.com

[agent]
  pluginsync = true
  environment = production

[master]
  reports = tagmail
```

Environment Contents

Environments all tend to look the same...

```
/etc/puppet/environments/
├── dev
|   ├── hieradata
|   ├── manifests
|   └── modules
|       ├── denyhosts
|       ├── firewall
|       ├── ngircd
|       ├── stdlib
|      ...
├── test
|   ├── hieradata
|   ├── manifests
|   └── modules
|       ├── denyhosts
|       ├── firewall
|       ├── ngircd
|       ├── stdlib
|      ...
└── production
     ├── hieradata
     ├── manifests
     └── modules
        ├── denyhosts
        ├── firewall
        ├── ngircd
        ├── stdlib
       ...
```

Directory Environments

Set `basemodulepath` and `environmentpath` in `puppet.conf`:

```
[main]
  certname = puppet.example.com
  basemodulepath = /etc/puppet/modules
  environmentpath = /etc/puppet/environments
  server = puppet.example.com

[agent]
  pluginsync = true
  environment = production

[master]
  reports = tagmail
```

Set `modulepath` and manifest in `environment.conf`:

```
modulepath = modules:$basemodulepath
```

All manifests in `manifests` will be loaded (by default).

Per-Environment Hiera data

`$environment` is interpolated in `hiera.yaml`.

```
---
:backends:
  - yaml

:hierarchy:
  - nodes/%{clientcert}
  - datacenters/%{datacenter}
  - platforms/%{operatingsystem}-%{operatingsystemrelease}
  - platforms/%{operatingsystem}
  - common

:yaml:
  :datadir: /etc/puppet/environments/%{environment}/hieradata
```

Future of Environments

- Enabling directory environments disables config-file environments
- Config-file environments are deprecated
- Directory environments will be the only option in Puppet 4

Monolithic Repositories

Puppet Super Overmind Monolith

All of `/etc/puppet` as a repository:

```
/etc/puppet/
├── environments
|   ├── dev
|   |   ├── hieradata
|   |   ├── manifests
|   |   └── modules
|   |       └── ...
|   ├── production
|   |   ├── hieradata
|   |   ├── manifests
|   |   └── modules
|   |       └── ...
|  ...
├── manifests
├── modules
|   └── ...
├── hiera.yaml
├── puppet.conf
└── ssl
    ├── ca
    ├── certificate_requests
    ├── certs
    ├── private
    ├── private_keys
    └── public_keys
```

...`private_keys`???

Environment Repository

Each branch as the contents of a single environment:

```
/etc/puppet/environments/production
├── hieradata
├── manifests
│   └── site.pp
└── modules
    ├── concat
    ├── denyhosts
    ├── firewall
    ├── git
    ├── inifile
    ├── ngircd
    ├── profiles
    ├── roles
    ├── stdlib
    ├── thttpd
    └── ...
```

Problems with Monolithic Repos

- History of local modules is intermixed and difficult to follow
- Importing Forge modules results in huge, useless changesets
- Rolling a single module forward or back is very difficult

The R10k Way

What is R10k?

R10k is a tool for deploying Puppet environments from version control.

What is R10k?

R10k is a tool for deploying Puppet environments from version control.

- Directory environments
- Private modules from Git
- Public modules from the Forge and Github
- **Environment metadata in a special "control" repository**

Control Repository

The control repository holds environment metadata, arranged for a single Puppet environment:

```
example-control/
├── dist
│   ├── profiles
│   └── roles
├── environment.conf
├── hieradata
│   └── ...
├── manifests
│   └── site.pp
└── Puppetfile
```

Puppetfile

The `Puppetfile` lists the desired modules and their versions.

```
# Modules from the Forge
mod 'puppetlabs/firewall','1.1.3'
mod 'puppetlabs/stdlib','4.3.2'
mod 'puppetlabs/vcsrepo','1.1.0'

# Modules from Git repositories
mod 'denyhosts',
  :git => 'git@github.com:cmroddy/puppetlabs-denyhosts.git',
  :ref => '3403309a'

mod 'ngircd',
  :git => 'git@github.com:cmroddy/puppet-ngircd.git',
  :ref => 'master'

mod 'thttpd',
  :git => 'git@github.com:cmroddy/puppet-sthttpd.git',
  :ref => 'v1.2.3'
```

Branches Become Environments

R10k will create an environment for each branch of the control repository.

```
croddy@devbox $ git branch
  dev
  preprod
* production
  test
```

Branches Become Environments

R10k will create an environment for each branch of the control repository.

```
croddy@devbox $ git branch
  dev
  preprod
* production
  test
```

After `r10k deploy environment -p…`

```
/etc/puppet/environments
├── dev
│   ├── dist
│   │   ├── profiles
│   │   └── roles
│   ├── environment.conf
│   ├── hieradata
│   │   └── ...
│   ├── manifests
│   │   └── site.pp
│   ├── modules
│   │   ├── denyhosts
│   │   ├── firewall
│   │   ├── ngircd
│   │   ├── stdlib
│   │  ...
│   └── Puppetfile
│
├── preprod
│   └── ...
├── production
│   └── ...
└── test
    └── ...
```

What is an Environment?

> **Puppet environment:**
> An isolated collection of **Puppet code**, configuration, and supporting materials, which a master can serve at **different points** in the development life cycle

Feature Branch Environments

```
croddy@devbox $ git branch
  dev
  preprod
  production
  test
  croddy_JIRA_567
* croddy_JIRA_1234
  jmorrison_JIRA_890
```

After `r10k deploy environment -p…`

```
/etc/puppet/environments/
├── croddy_JIRA_1234
├── croddy_JIRA_567
├── dev
├── jmorrison_JIRA_890
├── preprod
├── production
└── test
```

The 'production' Environment

The `production` Puppet environment is not the environment that contains your production nodes.

The 'production' Environment

The `production` Puppet environment is not the environment that contains your production nodes.

The `production` Puppet environment contains production-grade Puppet code that can manage nodes from any of your server environments.

Development Life Cycle

- Make a new branch based on `production`
- Update `Puppetfile` with new module versions
- Update Hiera data, Roles, and Profiles
- Test feature environment where appropriate
- Merge it up to QA, preprod, etc.

Development Life Cycle

- Unambigious collections of modules with easy merges
- Tests the same code in early environments as in production
- Site manifest, Roles, Profiles, modules, and Hiera data in a single Git hash

R10k Master Configuration

R10k Puppet Master Configuration

- Configure master for directory environments
- Initialize a control repository
- Install `r10k` gem and its dependencies
- Configure in `r10k.yaml`
- Set up a way to run R10k automatically

Directory Environments

Set `basemodulepath` and `environmentpath` in `puppet.conf`:

```
[main]
  certname = puppet.example.com
  basemodulepath = /etc/puppet/modules
  environmentpath = /etc/puppet/environments
  server = puppet.example.com

[agent]
  pluginsync = true
  environment = production

[master]
  reports = tagmail
```

(Legacy dynamic environments work, too)

Control Repository

- Initialize an empty Git repo
- Add a `Puppetfile`
- Copy in Hiera data, site manifest, and static modules
- Push it to a repo the master can access

R10k Gem

Install the `r10k` gem...

```
$ sudo gem install r10k
[sudo] password for croddy:
Fetching: multipart-post-1.2.0.gem (100%)
Successfully installed multipart-post-1.2.0
Fetching: faraday-0.8.9.gem (100%)
Successfully installed faraday-0.8.9
Fetching: faraday_middleware-0.9.1.gem (100%)

    .    .    .

Installing ri documentation for systemu-2.5.2
Parsing documentation for colored-1.2
Installing ri documentation for colored-1.2
Parsing documentation for cri-2.6.1
Installing ri documentation for cri-2.6.1
Parsing documentation for r10k-1.4.1
Installing ri documentation for r10k-1.4.1
Done installing documentation for multipart-post, faraday, faraday_middleware, multi_json, faraday_middleware
10 gems installed

$ which r10k
/usr/local/bin/r10k
```

Configure R10k

R10k needs to know how to find the control repository.

```
#/etc/r10k.yaml
---
:cachedir: /var/cache/r10k
:sources:
  puppet:
    basedir: /etc/puppet/environments
    remote: git@github.com:cmroddy/example-control.git
```

The user running R10k will usually need SSH keys.

Ready to Deploy

R10k fetches the control repo and creates environments for each branch.

```
$ r10k deploy environment -pv
[R10K::Task::Deployment::DeployEnvironments - INFO] Loading environments from all sources
[R10K::Task::Environment::Deploy - NOTICE] Deploying environment test
[R10K::Task::Puppetfile::Sync - INFO] Loading modules from Puppetfile into queue
[R10K::Task::Module::Sync - INFO] Deploying thttpd into /etc/puppet/environments/test/modules
[R10K::Task::Module::Sync - INFO] Deploying ngircd into /etc/puppet/environments/test/modules
[R10K::Task::Module::Sync - INFO] Deploying denyhosts into /etc/puppet/environments/test/modules
[R10K::Task::Module::Sync - INFO] Deploying vcsrepo into /etc/puppet/environments/test/modules
[R10K::Task::Module::Sync - INFO] Deploying stdlib into /etc/puppet/environments/test/modules
[R10K::Task::Module::Sync - INFO] Deploying inifile into /etc/puppet/environments/test/modules
[R10K::Task::Module::Sync - INFO] Deploying firewall into /etc/puppet/environments/test/modules
[R10K::Task::Module::Sync - INFO] Deploying concat into /etc/puppet/environments/test/modules
[R10K::Task::Module::Sync - INFO] Deploying make into /etc/puppet/environments/test/modules
[R10K::Task::Environment::Deploy - NOTICE] Deploying environment production
[R10K::Task::Puppetfile::Sync - INFO] Loading modules from Puppetfile into queue
[R10K::Task::Module::Sync - INFO] Deploying thttpd into /etc/puppet/environments/production/modules
[R10K::Task::Module::Sync - INFO] Deploying ngircd into /etc/puppet/environments/production/modules
[R10K::Task::Module::Sync - INFO] Deploying denyhosts into /etc/puppet/environments/production/modules
[R10K::Task::Module::Sync - INFO] Deploying vcsrepo into /etc/puppet/environments/production/modules
[R10K::Task::Module::Sync - INFO] Deploying stdlib into /etc/puppet/environments/production/modules
[R10K::Task::Module::Sync - INFO] Deploying inifile into /etc/puppet/environments/production/modules
[R10K::Task::Module::Sync - INFO] Deploying firewall into /etc/puppet/environments/production/modules
[R10K::Task::Module::Sync - INFO] Deploying concat into /etc/puppet/environments/production/modules
[R10K::Task::Module::Sync - INFO] Deploying make into /etc/puppet/environments/production/modules
[R10K::Task::Deployment::PurgeEnvironments - INFO] Purging stale environments from /etc/puppet/environments
    .    .    .
```

Automatic Deployment

- Cron job
- Git hook
- Mcollective
- Jenkins
- Master's `prerun_command`

zack/r10k

http://forge.puppetlabs.com/zack/r10k

- Manages the Gem and its dependencies
- Manages `r10k.yaml`
- Includes an Mcollective agent and a web hook
- More famous than R10k itself (on Google)

Gotchas

- `master` is an illegal environment name, so you must rename the branch
- R10k does not install dependencies, you must specify them
- `hiera.yaml` can't be consulted on a per-environment basis *(HI-46)*
- Agent-set environments require a special configuration in PE 3.7
- R10k has Subversion support, so your Git *coup d'état* might fail

?