Puppet in the Enterprise

Thomas Uphill
thomas@narrabilis.com

Latest version

http://goo.gl/G0TLfJ

Example Files

https://github.com/uphillian/lisa2014

If you see something, say something!

Google Comments enabled

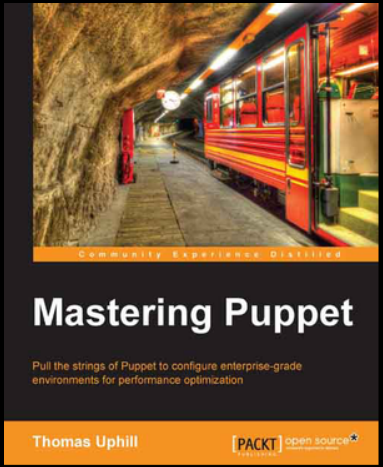
Watch in Presentation Mode!

Seriously, Trust Me

...Animations Are Good things

Me

PuppetConf 2013
Mastering Puppet
Puppet Cookbook 4*


Community Experience Distilled
Mastering Puppet
Pull the strings of Puppet to configure enterprise-grade environments for performance optimization
Thomas Uphill
[PACKT] open source

Email server

time shift

"Tend the flock, not the sheep"

-- Me

The Puppet Problem

The Puppet Problem

system administrators

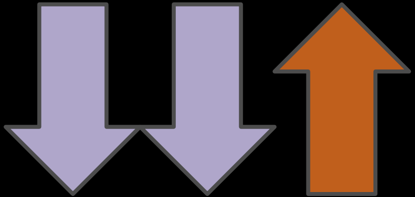
- ❑ scripts
- ❑ pipes/redirection
- ❑ **lazy**

developers

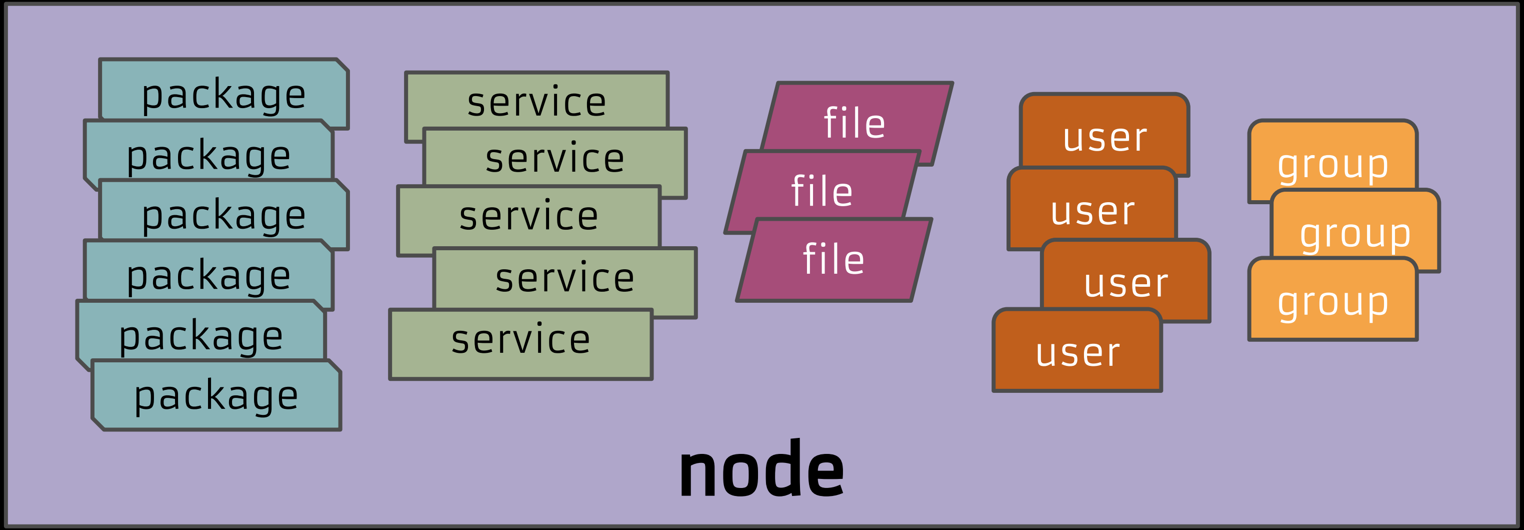
- ❑ objects
- ❑ code reuse
- ❑ **lazy**

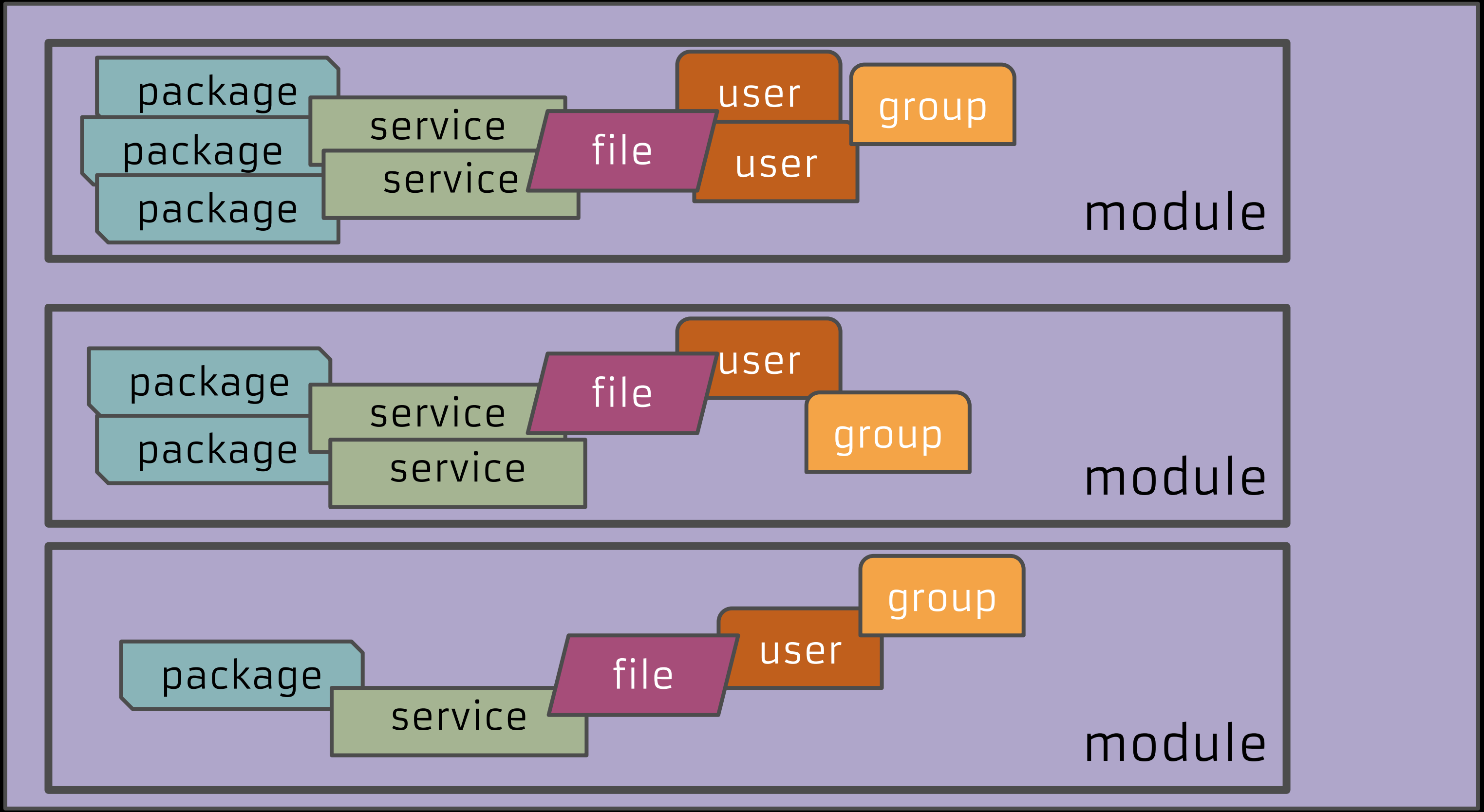
Puppet Problem

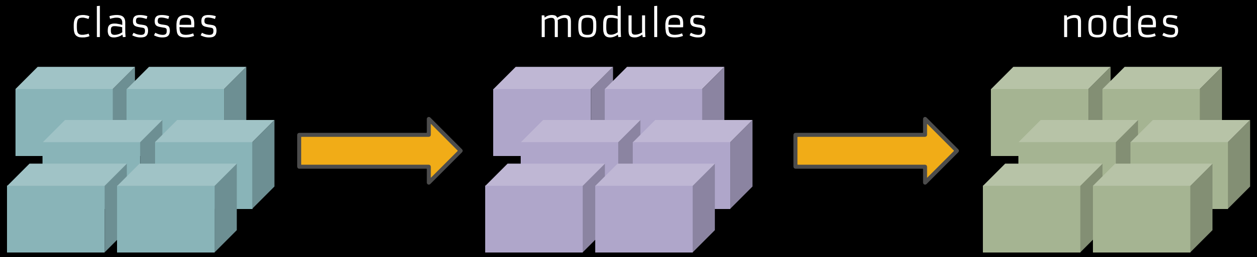
system administrators



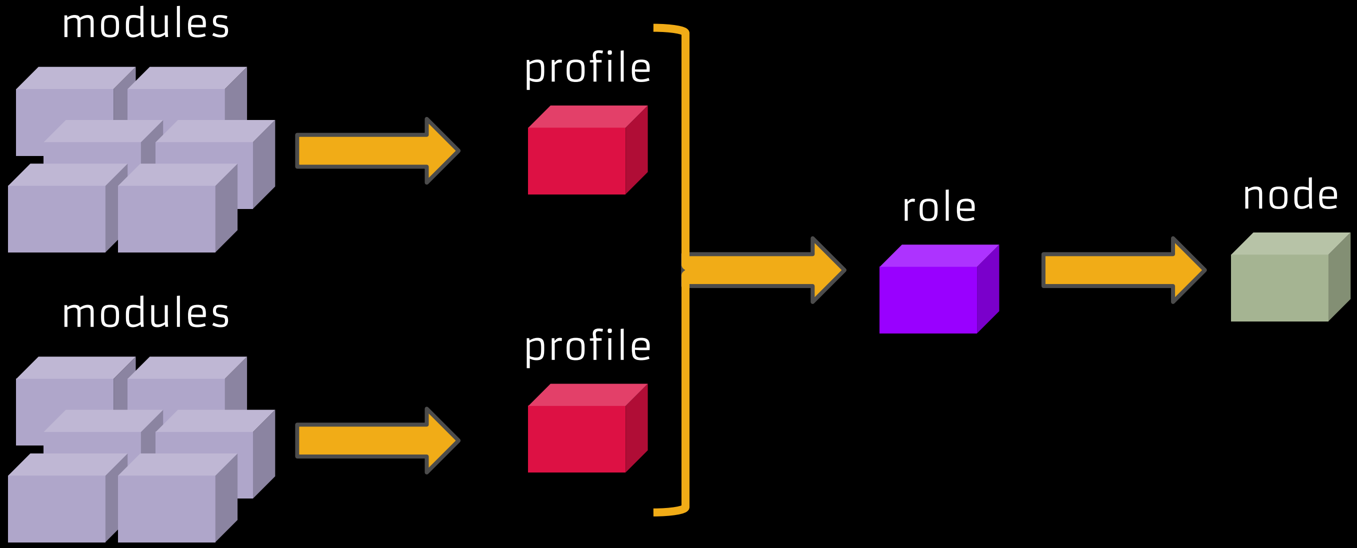
developers


package
package
package
package
package
package
service
service
service
service
service
file
file
file
user
user
user
user
group
group
group
node

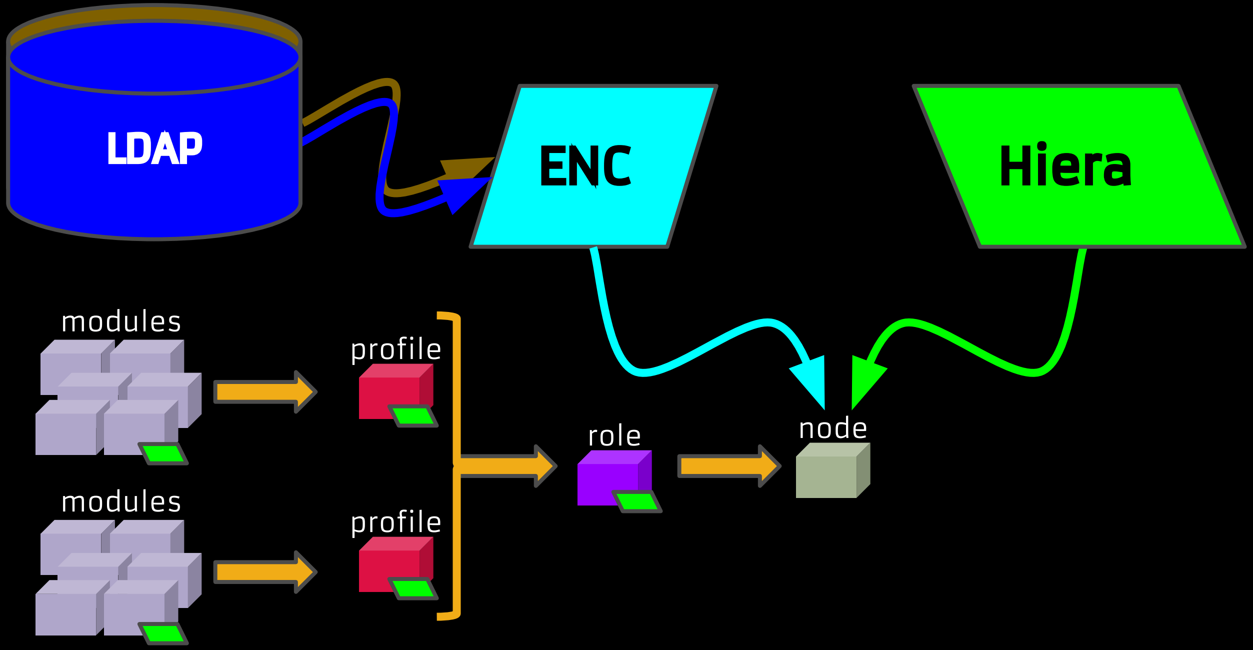

package
package
package
service
service
file
user
user
group
module
package
package
service
service
file
user
group
module
package
service
file
user
group
module


classes
modules
nodes

Roles and Profiles



Roles and Profiles and Exceptions



Goal

```
node thx
  class { 'r
}
```

```
if $::hostname == "thatweirdone" {
  include obscureclass95 }
else {
  include superobscure72
}
```

The Puppet Problem

- **Minimize** exceptions
- if else if else if else if else if else
- case
  - case
    - case
      - case

Hiera

Separating **data** from **code**

Techniques:

- Parameterized classes
- hiera_include
- fact based hierarchy

But first...custom facts

custom facts

Why?

- facts are loaded and defined early in catalog compilation
- facts can be used in hiera hierarchy
- facts can be used as selectors in case statements

custom facts

Two methods:

- external facts
  - `/etc/facter/facts.d/fact.txt`
  - `/etc/facter/facts.d/fact.{yaml|json}`
  - `/etc/facter/facts.d/`**`filename*`** *(chmod +x)*
- custom facts
  - `module/lib/facter/fact.rb`

custom facts

External facts:

❑

❑

❑

```
#!/bin/bash
otherfact=$(facter -p otherfact
factone=$((otherfact + 1))
echo factone=$factone
echo facttwo=other
```

http://bluehawk.monmouth.edu/~rclayton/web-pages/s11-503/recursion.jpg

custom facts

written in ruby
can access previously defined facts
puppet 3+ ⇒ automatically sync'ed

Parameterized Classes

- class
  (argu

```
class resol
  file {'/e
    content
    owner
    group
    mode
  }
}
```

```
class resolv ($nameserver = '8.8.8.8') {
  file {'/etc/resolv.conf':
    content => "nameserver $nameserver",
    owner   => '0',
    group   => '0',
    mode    => '0644',
  }
}
```

Parameterized Classes

When?

- Include **without** modification:
  - `include resolv`
  - `class {'resolv': }`
- Include **with** modification

  -
    ```
    class {'resolv':
        nameserver => 'value'
    }
    ```

parameterized class

```
class selinux ($config = 'enforcing') {
  case $config {
    'enforcing':  {
      exec {'selinux_enforcing':
        command => '/usr/sbin/setenforce 1',
      }
    }
    /permissive|disabled/: {
      exec {'selinux_permissive':
        command => '/usr/sbin/setenforce 0',
        }
    }
  }
  file {'/etc/selinux/config':
    content => "SELINUX=$config\nSELINUXTYPE=targeted\n",
  }
}
```

Automatic Parameter lookup

```
3.8') {

erver",
```

```
---
resolv::nameserver: 8.8.4.4
```

```
    gr
    mo
  }
}
```

```
include resolv
class {'resolv': }
```

```
nameserver 8.8.4.4
```

Hierarchy

```
:hierarchy:
  - "cunning/%{::cunning_fact}"
  - global
```

pst.yaml

```
---
resolv::nameserver: 8.8.4.1
```

where the

/etc/hieradata/cunning/
pst.yaml
cmt.yam
est.yam

```
nameserver 8.8.4.1
```

hiera_include

hiera_include('lookupkey','notfound')

- Lookup
- include
  'lookup
- if nothir '
- call hi

site.pp

```
node default {
  class {'base': }
  hiera_include('classes','notfound')
}
```

fact based hierarchy

hiera.yaml

```
---
:hierarchy:
  - "%{hostname}"
  - "%{operatingsystem}"
  - "is_virtual/%{is_virtual}"
  - common
```

fact based hierarchy

hi

--

:hi

- "%{h

"%{operatingsyste

/hieradata

common.yaml

site.pp

```
...
hiera_include('classes','notfound')
...
```

true.yaml

```
---
classes: 'virtual_mac
```

notfound/manifests/init.pp

```
class notfound {
  # nothing
}
```

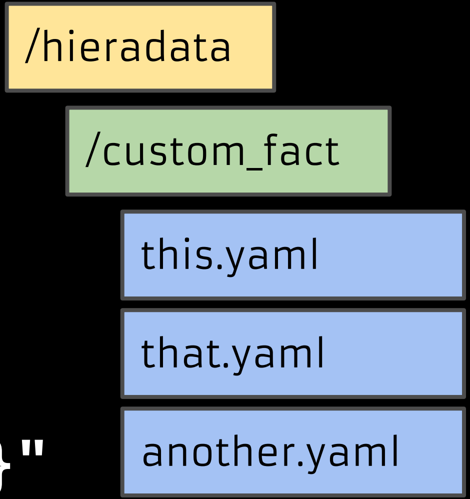
virtual_machine/manifests/init.pp

```
class virtual_machine {
  service {'tuned': ensure => running }
  exec {'tuned-adm virtual':
    command => 'tuned-adm profile virtual-guest',
  }
}
```

fact based hierarchy - custom fact

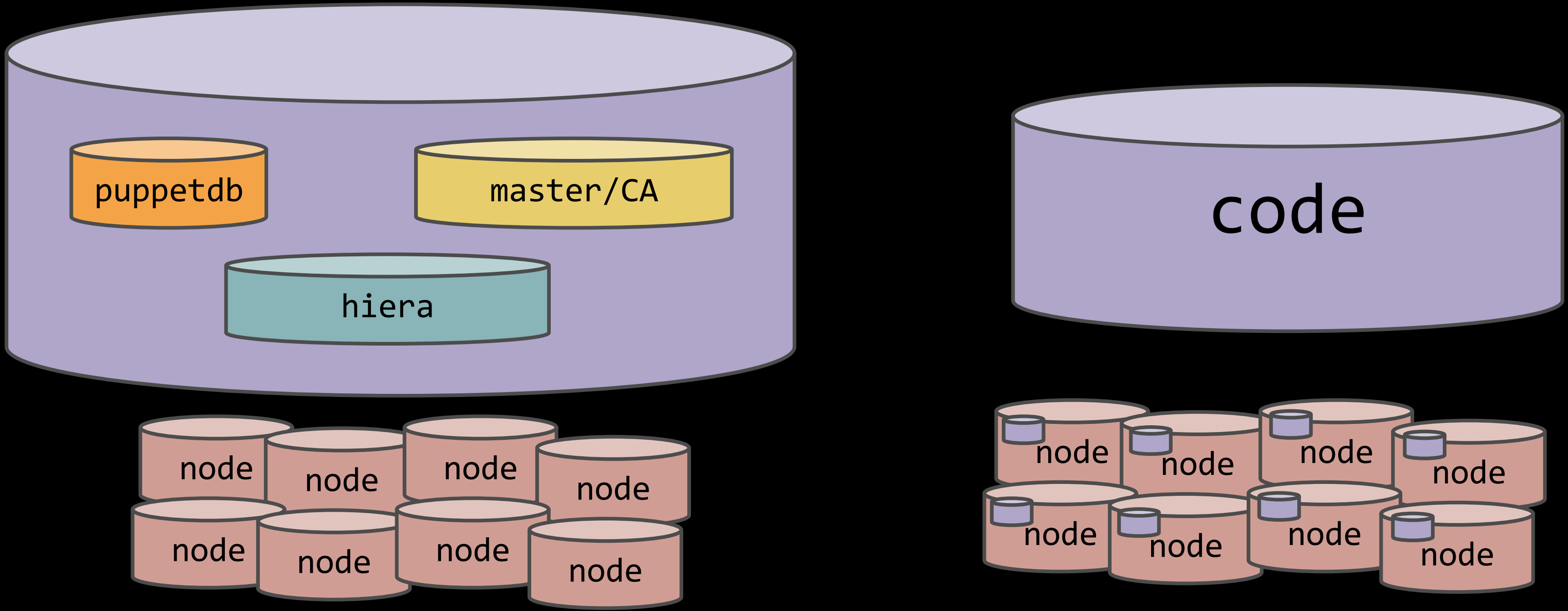
hiera.yaml

```
---
:hierarchy:
  - "%{hostname}"
  - "%{operatingsystem}"
  - "is_virtual/%{is_virtual}"
  - "custom_fact/%{custom_fact}"
  - common
```



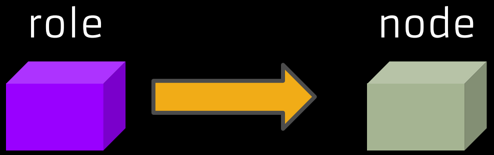
Centralized or Not?

Centralized/Decentralized



Decentralized

```
puppet apply
```



Centralized

```
puppet agent
```

Scaling

Scaling

What is the most important thing to remember about puppet?

- Puppet is a web service.
- Puppet is a web service on port 8140
- Puppet is an SSL web service on port 8140

Scaling

REST API

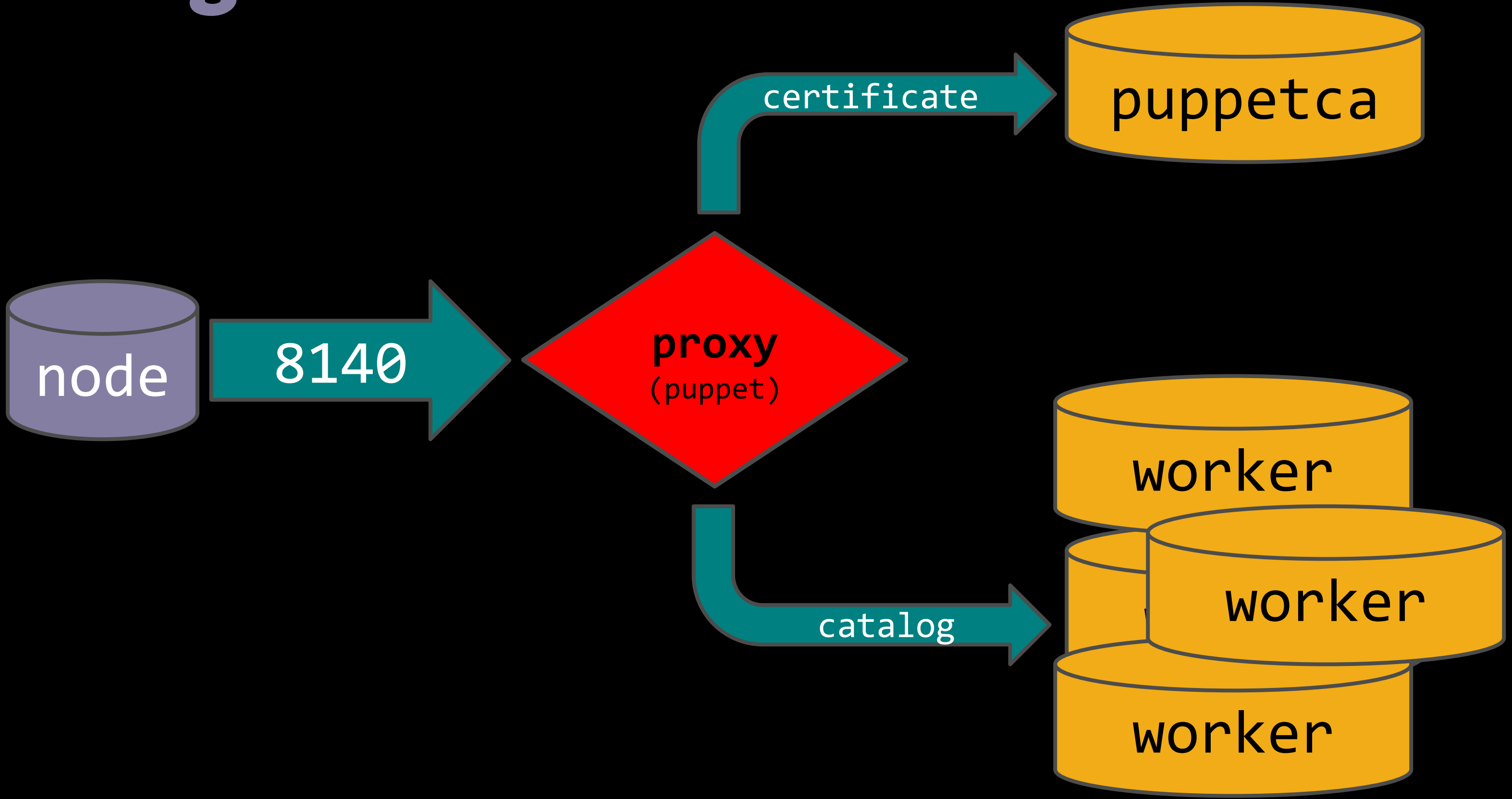
https://puppet:8140/environment/resource/key

`catalog` `file_metadata`

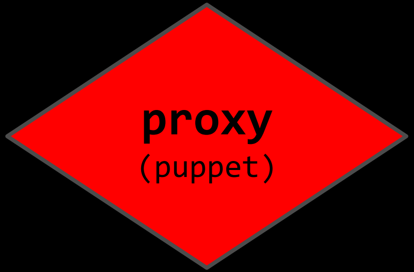
`certificate` `fact`

`file_content`

Scaling

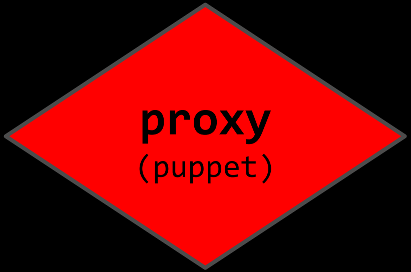

node
8140
proxy
(puppet)
certificate
puppetca
catalog
worker
worker
worker

Scaling



- ☐ apache
- ☐ nginx
- ☐ haproxy

Scaling/apache



```
<Proxy balancer://puppetca>
BalancerMember http://127.0.0.1:18140
</Proxy>
<Proxy balancer://puppetworker>
BalancerMember http://192.168.100.101:18140

BalancerMember http://192.168.100.102:18140
</Proxy>
Listen 8140
<VirtualHost *:8140>
  SSLEngine on

  …<setup ssl>...


  ProxyPassMatch ^/([^/]+/certificate.*)$ balancer://puppetca/$1
  ProxyPass / balancer://puppetworker/
</VirtualHost>
```

Scaling: does it actually work?

Demo 1:

VM

- proxy
- passenger
- puppetdb / postgresql

Infrastructure as Code
Software as a Service
Platform as a Service

buzzword something

development
continuous integration
refactoring

workflow

Workflow

Decentralized:

- create machine
- install puppet
- apply role
- download code
- puppet apply

Centralized:

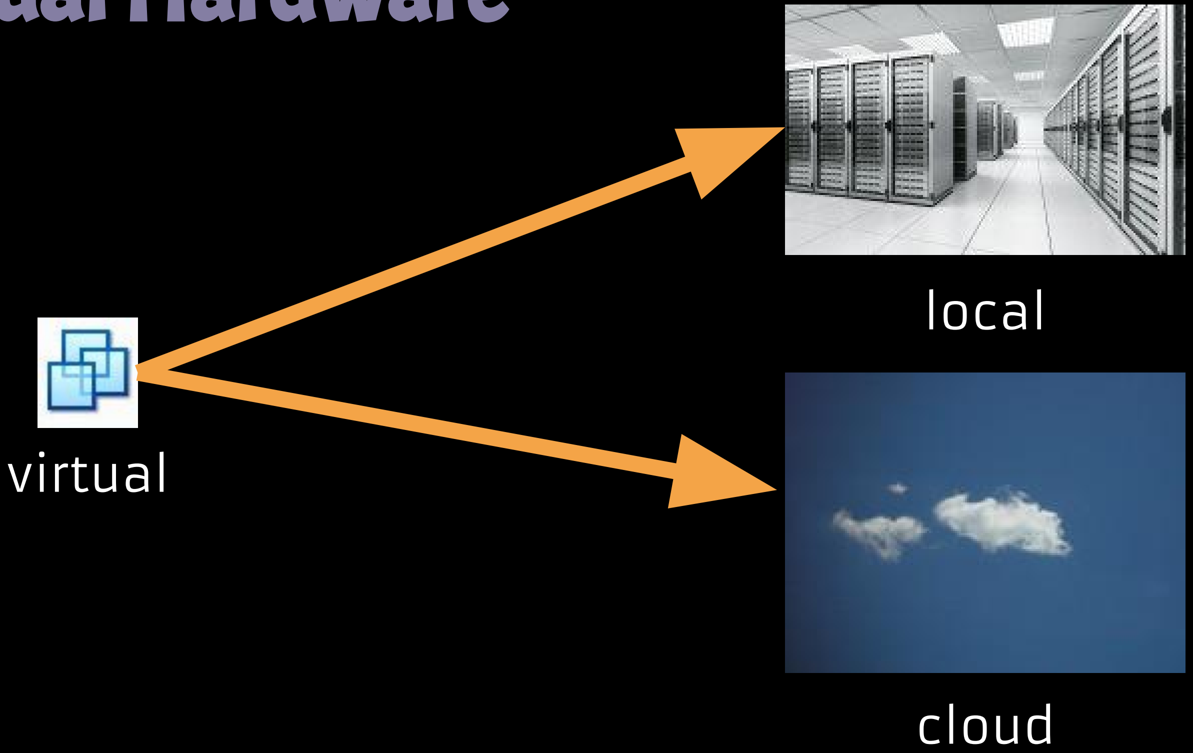
- create machine
- install puppet
- apply role
- puppet agent

Hardware

physical

virtual

Virtual Hardware



Bootstrapping

Decentralized:

- create machine
- **install puppet (bootstrap)**
- apply role
- download code
- puppet apply

Centralized:

- create machine
- **install puppet (bootstrap)**
- apply role
- puppet agent

Bootstrapping

Install Puppet

- `gem install puppet`
- `apt-get install puppet`
  - install puppetlabs apt source
- `yum install puppet`
  - install puppetlabs yum repo
- tar file
- port/brew install puppet

Bootstrapping

Apply role

- hiera

```
$role = hiera('role','undefined')
```

- ENC

```
CMDB lookup
LDAP lookup
```

- `node definition` ← doesn't scale well

  site.pp

```
node 'nodename' { include 'webserver' }
```

Bootstrapping

ensure puppet running

- agent: service
- apply: cron task

install puppet

```
package {'puppet': ensure => installed }
```

Workflow - creation

- Provision (VM/Physical)
- Bootstrap puppet
  - Assign role to node
- Apply puppet (agent or apply)
  - ensure puppet installed properly
  - ensure puppet running (service or cron task)
- Register node
  - monitoring / nagios

Workflow - deletion

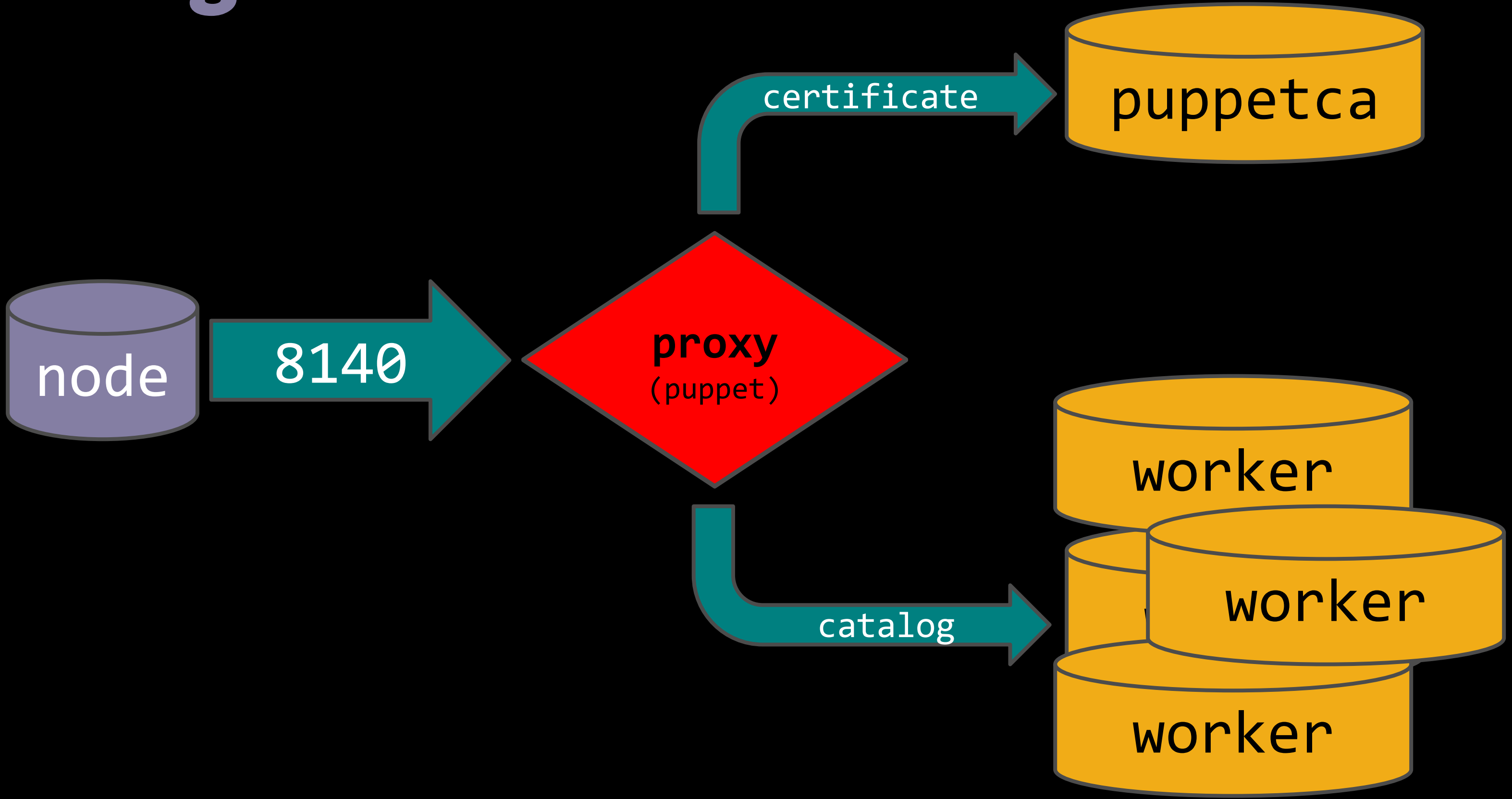
- Decommission (VM/Physical)
- Remove role assignment
  - hiera/enc/ldap
- Delete from Reports (foreman/console)
- De-register node
  - monitoring / nagios

Workflow

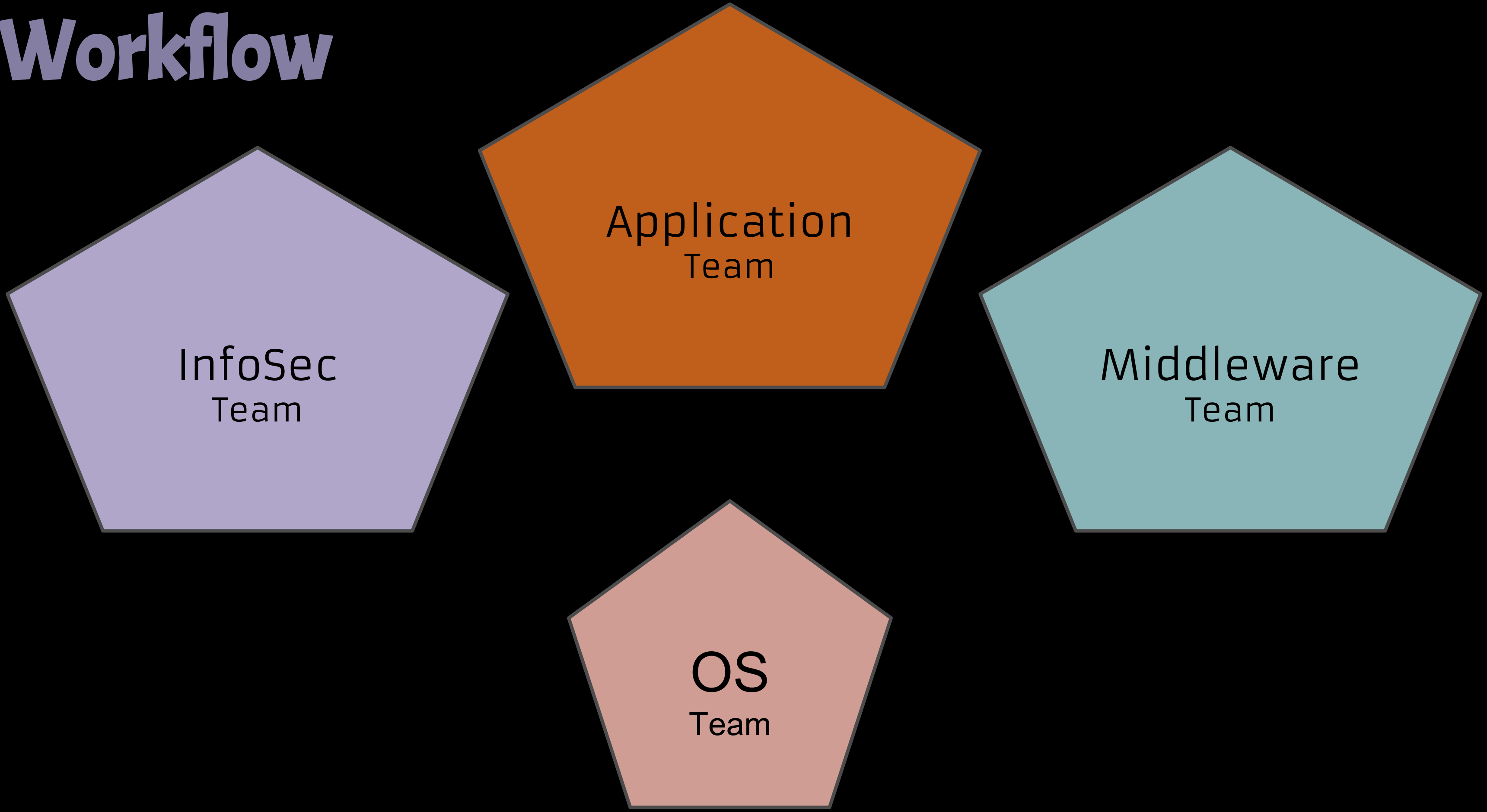
Maximize return on investment:

1. install puppet early
2. apply `bootstrap.pp` manifest
3. ***
4. profit

Scaling

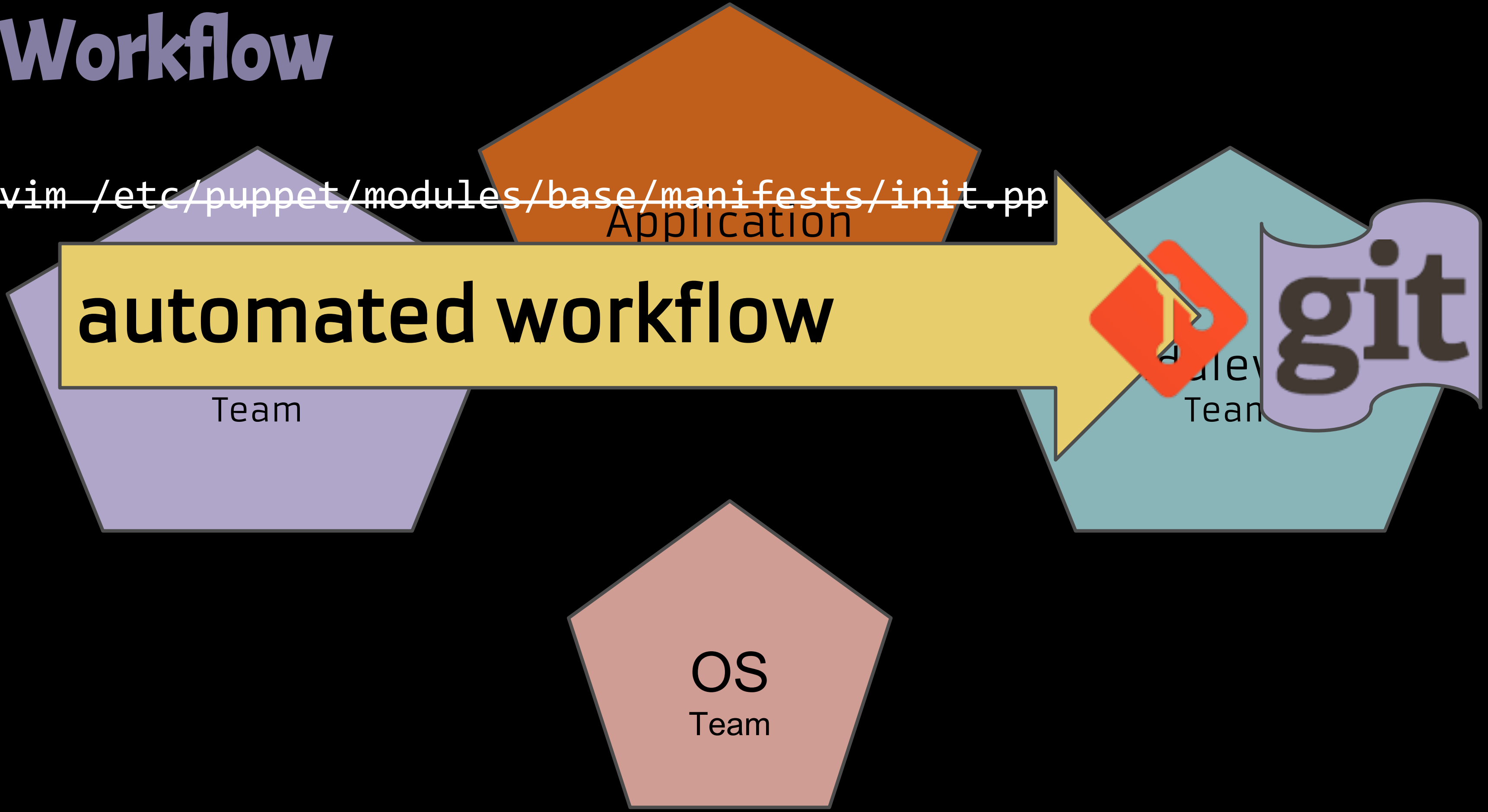

certificate
puppetca
node
8140
proxy
(puppet)
worker
worker
worker
catalog

Workflow



Workflow

~~vim /etc/puppet/modules/base/manifests/init.pp~~



Workflow (code)

- Push code to masters
- Branches
  - Code promotion
  - Environments
  - Purge old
- Hieradata

- defacto source code control for puppet
- integrates into workflow
- cheap branches
- hooks

Git Hooks

http://goo.gl/dg5TVw

- Branch is a reference
- references are hashes
- branches are cheap

- branch == environment
- directory environments *(3.6+)*

```
environmentpath = /etc/puppet/environments
environment.conf
  modulepath = relative:path:/absolute/path
  manifest = relative/path/site.pp
```

- directory with `environmentpath` is the environment

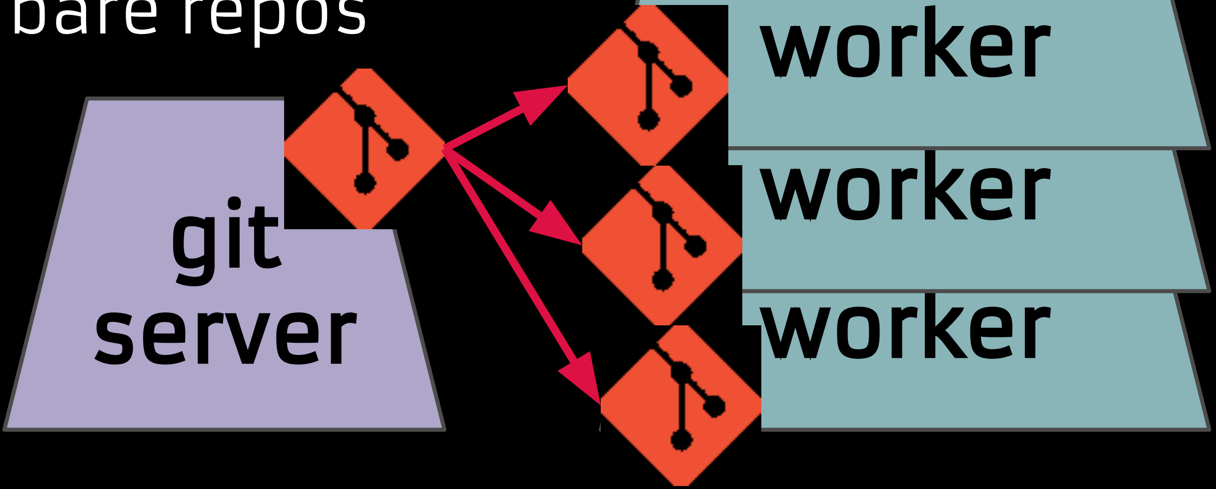
directory environments

```
$environmentpath/
                production/     branch
                          /modules
                          /manifests

                devel/     branch
                     /modules
                     /manifests
```

git

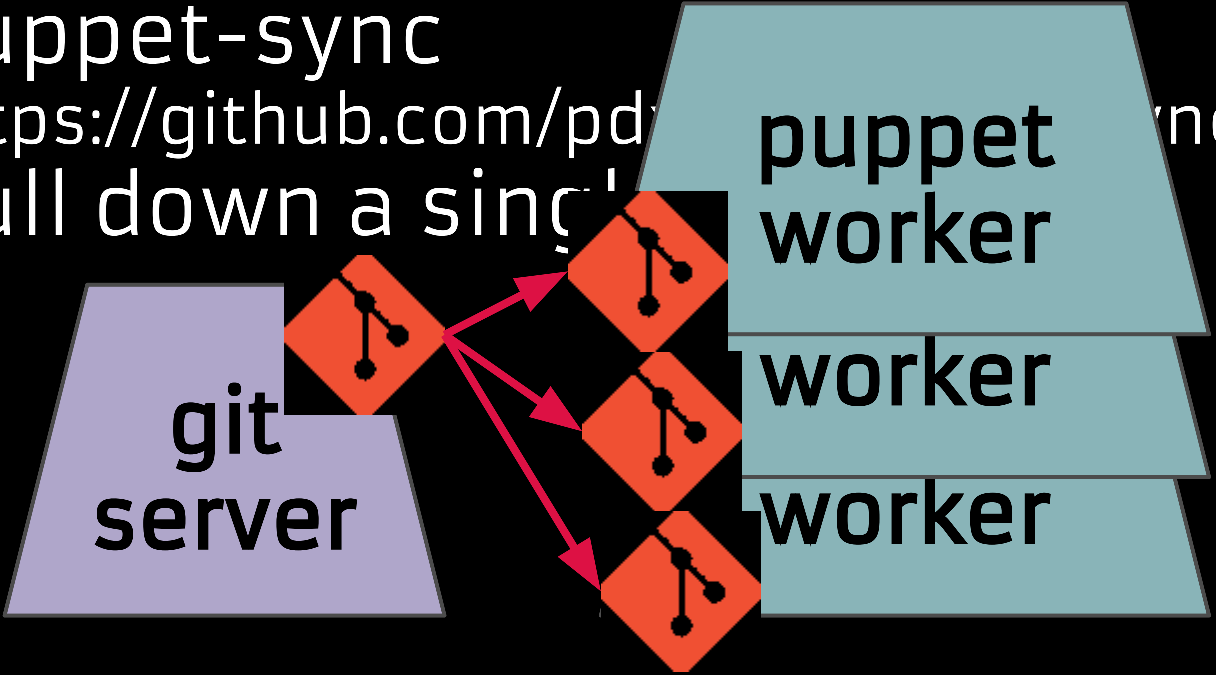
- repository 101
  - remote / origin
  - bare repos



- hooks
- many hooks, two useful here:
  - `pre-receive`
    separate who can do what
  - `post-receive`
    push code

puppet-sync

- puppet-sync https://github.com/pd ync
- pull down a sing



puppet-sync

```
#!/bin/bash
DEPLOY='/etc/puppet/environments'
REPO='hostname:/srv/git/repos/puppet.git'

read oldrev newrev ref
BRANCH=${ref/*\/*\/}

sudo -u puppet ssh worker "puppet-sync --branch $BRANCH --repository $REPO --deploy $DEPLOY"

exit=$?
exit $exit
```

puppet-sync

```
#!/bin/bash
DEPLOY='/etc/puppet/environments'
REPO='hostname:/srv/git/repos/puppet.git'
SSH_KEY='/var/lib/puppet/.ssh/puppet-sync.key'

read oldrev newrev ref
BRANCH=${ref/*\/*\/}

sudo -Hu puppet ansible workers \
  -a "puppet-sync --branch $BRANCH --repository $REPO --deploy $DEPLOY" \
  -o --private-key=$SSH_KEY

exit=$?
exit $exit
```

Up to here

- single git repository
- clone to each master (worker/CA)
- automated

Workflow

- multiple git repositories
  - librarian-puppet
  - r10k

Puppetfile

```
forge "https://forgeapi.puppetlabs.com"

mod 'puppetlabs/stdlib', '4.1.0'

mod 'puppetlabs/apache', '1.1.1'
  :git => "git://github.com/puppetlabs/puppetlabs-apache.git"

mod 'puppetlabs/apt', '1.1.1'
  :git => "git://github.com/puppetlabs/puppetlabs-apt.git"
  :ref => 'any/valid/gitref'
```

r10k

https://gith

- Uses Pu
- local ca
- Configu

r10k.yaml

```
:cachedir: '/var/cache/r10k'
:sources:
 :plops:
   remote: '/var/lib/git/puppet.git'
   basedir: '/etc/puppet/environments'
```

```
/etc/puppet/environments/$branch
                                /Puppetfile
                                /company/modulename
```

r10k

r10k.yaml

```
:cachedir: '/var/cach
:sources:
 :plops:
   remote: '/var/lib/gi
   basedir: '/etc/puppe
```

Puppetfile

```
mod 'puppetlabs/stdlib'
mod 'os',
  :git => '/var/lib/git
mod 'middleware',
  :git => '/var/lib/git
```

```
/etc/puppet/environments/$branch
                                /Puppetfil
                                /company
                                        mo ule1
                                        module2
                                        module3
                                /modules/
                                        stdlib
                                        os
                                        middleware
                                /environment.conf
```

dist, local, ours

```
modulepath = modules:company:/somewhere/else
manifest = manifests/site.pp
```

r10k

deploy using r10k

```
$ r10k deploy environment -p [environment]
```

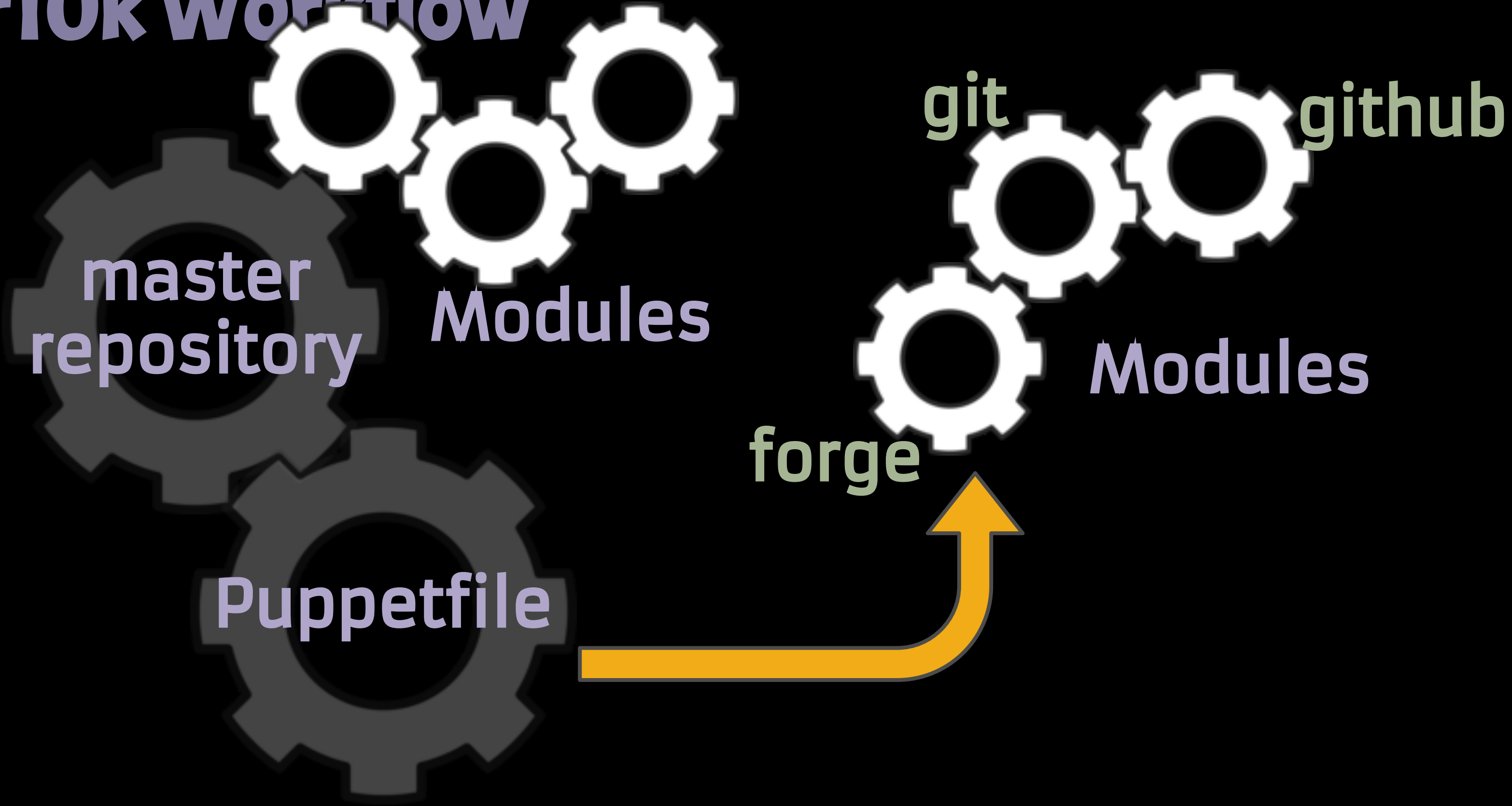
even better

```
$ sudo -u puppet !!
```

r10k post-receive (git hook)

```
read oldrev newrev refname
branch=${refname#*\/*\/}
sudo -u puppet \
  r10k deploy environment $branch -p
```

r10k Workflow



r10k Workflow

master repository

"He who controls the spice controls the universe"

Baron Vladimir Harkonnen

r10k Workflow

One repository per module

Modules included by Puppetfile

r10k repo controls everything

hiera

hieradata is in git also
githook pushes hiera code
hiera = exceptions
  add modules/profiles to a node

hieradata - multiple teams

multiple backends

- OS Team ⇒ JSON
- App Team ⇒ YAML
- WebGui ⇒ Database

You can still use an ENC too.

Goal

```
node thx1138 {
  class { 'role::drupal7': }
}
```

Real Goal

"If you are editing code in /etc/puppet, you are doing it wrong." - Me

Bootable ISO

tutorial.html
Demo 2, 3 and 4

Troubleshooting

http://goo.gl/b2NISc

Summary

Create a workflow/lifecycle for nodes

Create a workflow for code

hieradata

Separate data from code: hiera

create a class hierarchy: roles/profiles

centralize or decentralize: scale

KISS

Questions?

Comments?