Our Puppet Story – Patterns and Learnings

Martin Schütte

German Unix User Group

March 27 2014

1. **Intro**

2. **Vagrant**

3. **Puppet**
    - Intro
    - Dashboard & PE
    - Facter & Hiera
    - git
    - Problems
    - Misc

1. **Intro**

2. **Vagrant**

3. **Puppet**
    - Intro
    - Dashboard & PE
    - Facter & Hiera
    - git
    - Problems
    - Misc

About DECK36

- Small team of 7 engineers
- Longstanding expertise in designing, implementing and operating complex web systems
- Developing own data intelligence-focused tools and web services
- Offering our expert knowledge in Automation & Operation, Architecture & Engineering, Analytics & Data Logistics

About me

- System Automation Engineer
- Puppet Certified Professional 2013
- martin.schuette@deck36.de

The Problem

Jason Antman
@j_antman

We have the word "iff". Can we start using "inn", as in "this works inn the test environment"?

12:48 PM - 15 Nov 2013

Zvi 'Viz' Effron @CtrlZvi Nov 15
@j_antman @tom_forsyth And it's better known cousin "onn." As in, "Works onn my machine."

The Goal

Stable and reproducible environment for a Software.

- ... environment for new developer,
- ... test config changes,
- ... clean package build env,
- ... preconfigured demo box.

But also quickly deployable, and centrally managed
with current software versions.

1. Intro

2. **Vagrant**

3. Puppet
    - Intro
    - Dashboard & PE
    - Facter & Hiera
    - git
    - Problems
    - Misc

Vagrant

Configuration tool for VMs and Provisioning.

“Local cloud”

- Self service
- Instant provisioning
- Cost efficient
- Elastic
- Pay per use

Vagrant

VM Providers:

- VirtualBox: “default”, works offline, ressource hungry
- Docker: lightweight, requires Linux, good for testing
- AWS EC2: remote VMs, good for automation (Jenkins)

Provisioning:

- Shell script
- Puppet, apply manifest or run agent
- Chef, solo or client
- Ansible playbooks
- Salt states
- Docker containers

VeeWee definition

```
Veewee::Definition.declare({
  :iso_file  => "debian-wheezy-DI-b4-amd64-netinst.iso",
  :disk_size => '40560', :disk_format => 'VDI',
  :cpu_count => '2',     :memory_size => '3192',

  :boot_wait => "10",    :boot_cmd_sequence => [
     '<Esc>', 'install ',
     'preseed/url=http://%IP%:%PORT%/preseed.cfg ',
     'debconf/frontend=noninteractive ', '<Enter>'
  ],

  :postinstall_files => [
    "base.sh", "vagrant.sh", "customize-puppet.sh", ...
  ],
  ...
})
```

Vagrantfile

```
Vagrant.configure("2") do |config|
  config.vm.box = "graylog2"
  config.vm.box_url = "http://vagrantboxes.footballradar.com/wheezy64.box"

  config.vm.provider "virtualbox" do |v|
    v.memory = 1024
  end

  config.vm.provision :puppet do |puppet|
    puppet.manifest_file = "graylog2.pp"
    puppet.module_path = "modules"
  end

  config.vm.network :forwarded_port, guest: 9000, host: 9000
  config.vm.network :forwarded_port, guest: 80, host: 8080
  config.vm.network :forwarded_port, guest: 12201, host: 12201, protocol: 'udp'
  config.vm.network :forwarded_port, guest: 12201, host: 12201, protocol: 'tcp'
  config.vm.network :forwarded_port, guest: 12900, host: 12900

end
```

Multi-VM Vagrantfile

```
Vagrant.configure("2") do |config|
  # VM 1: appserver
  config.vm.define :app do |app|
    app.vm.hostname = "testbox.example.org"
    app.vm.network :forwarded_port, host: 8080, guest: 80
    app.vm.synced_folder ".", "/home/vagrant/files"
  end

  # VM 2: DB server
  config.vm.define :db do |db|
    db.vm.hostname = "db.example.org"
    db.vm.provider :virtualbox do |vb|
      vb.customize ["modifyvm", :id, "--cpus", "2"]
    end
  end

  # Box & Provisioning
  config.vm.box = "precise64"
  config.vm.provision :shell,
      :path => "vagrant_install_puppet_keys.sh"
  config.vm.provision :puppet_server,
      :puppet_server => "puppetmaster.example.org"
end
```

vagrant-aws

```
Vagrant.configure("2") do |config|
  config.vm.box = "dummy"

  config.vm.provider :aws do |aws, override|
    aws.access_key_id = "YOUR KEY"
    aws.secret_access_key = "YOUR SECRET KEY"
    aws.keypair_name = "KEYPAIR NAME"

    region  = "eu-west-1"
    aws.ami = "ami-20414854"

    aws.tags = {
      'Role' => 'TestVM',
      'Net'  => 'Devnet'
    }
  end
end
```

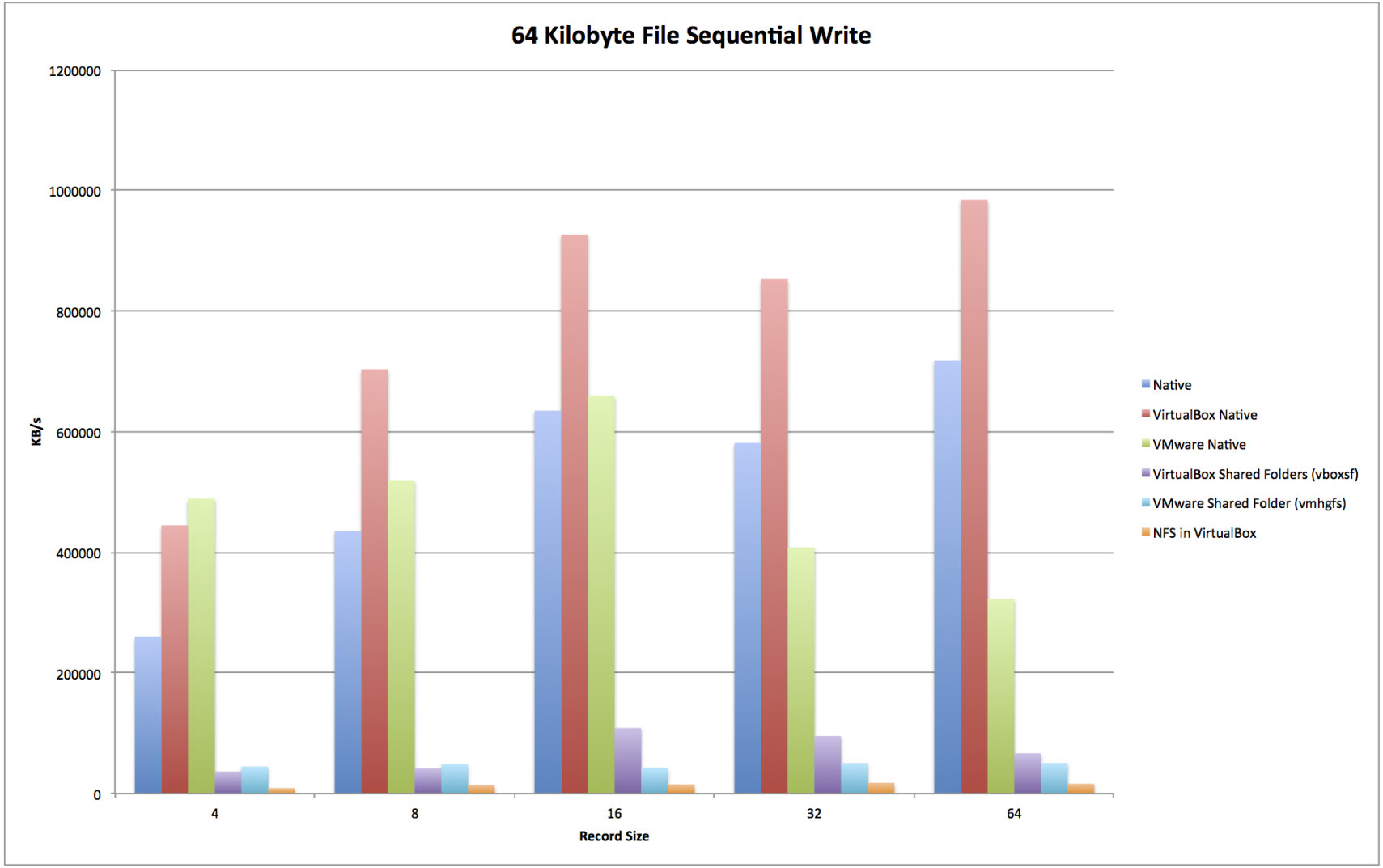
Synced Folders

Shared folders, mounted from host into guest.

Options:

- VirtualBox
- NFS
- SMB
- rsync

Synced Folders



src: Mitchell Hashimoto, Comparing Filesystem Performance in Virtual Machines

1. Intro

2. Vagrant

3. **Puppet**
    - Intro
    - Dashboard & PE
    - Facter & Hiera
    - git
    - Problems
    - Misc

Puppet

- Configuration Management
- Declarative: Resources and Dependencies

Puppet

Puppet Agent execution:

1. Create catalog:
    - read manifest
    - gather resources
    - ensure order
2. Apply for each resource:
    - query state
    - change to desired state

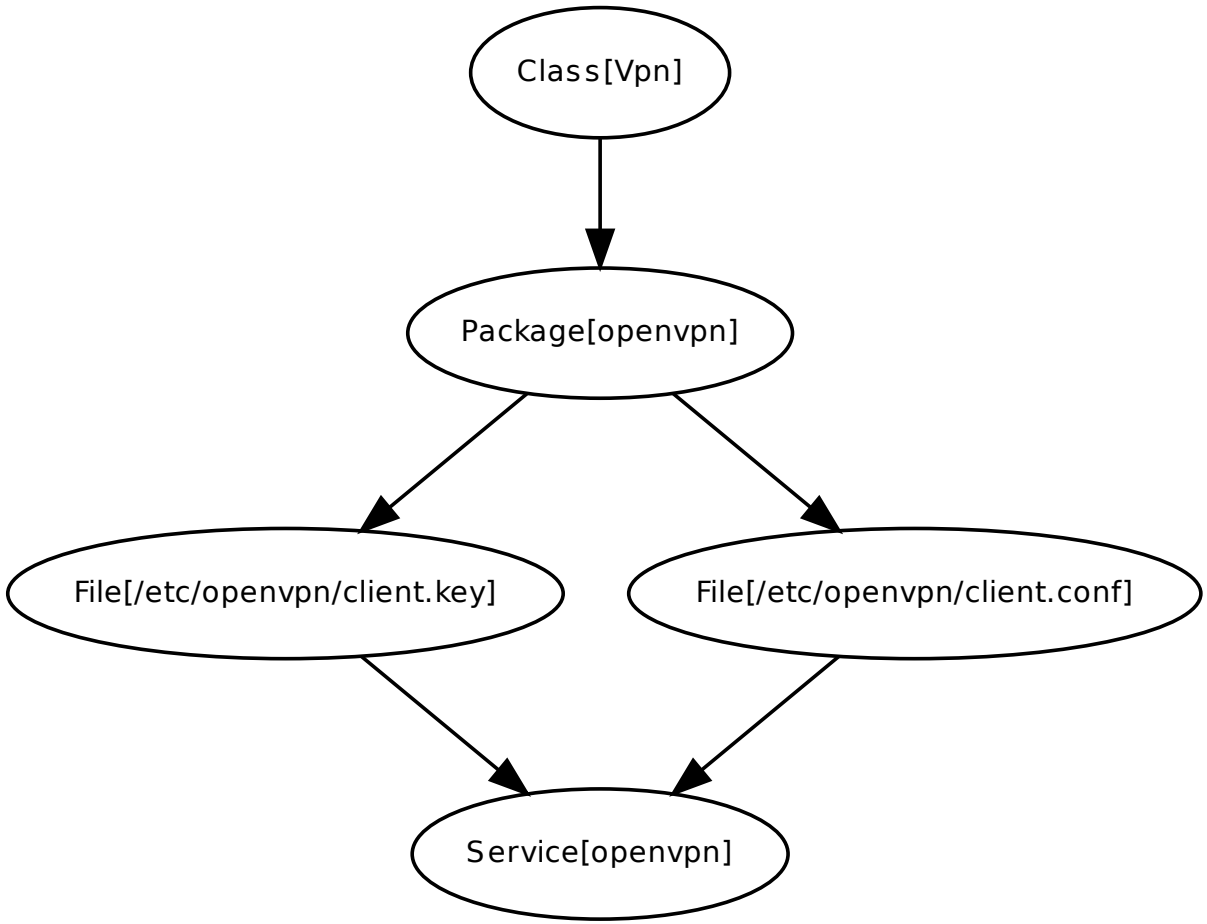
Syntax

```
class vpn($version = 'present', $ca_crt, $usr_crt, $usr_key) {
  package {
    'openvpn':
      ensure => $version;
  }

  file {
    "/etc/openvpn/client.key":
      ensure  => file,
      mode    => '0600',
      content => $usr_key;
      require => Package['openvpn'],
      notify  => Service['openvpn'];
    "/etc/openvpn/client.conf":
      ensure  => file,
      source  => "puppet:///modules/vpn/client.conf",
      require => Package['openvpn'],
      notify  => Service['openvpn'];
  }

  service { 'openvpn':
    ensure  => running,
    require => Package['openvpn'],
  }
}
```

Syntax

```
class vpn($version = 'present', $ca_crt, $usr_crt, $usr_key) {
  package {
    'openvpn':
      ensure => $version;
  }
  ->
  file {
    "/etc/openvpn/client.key":
      ensure  => file,
      mode    => '0600',
      content => $usr_key;
    "/etc/openvpn/client.conf":
      ensure => file,
      source => "puppet:///modules/vpn/client.conf";
  }
  ~>
  service { 'openvpn':
    ensure => running,
  }
}
```

Relationships

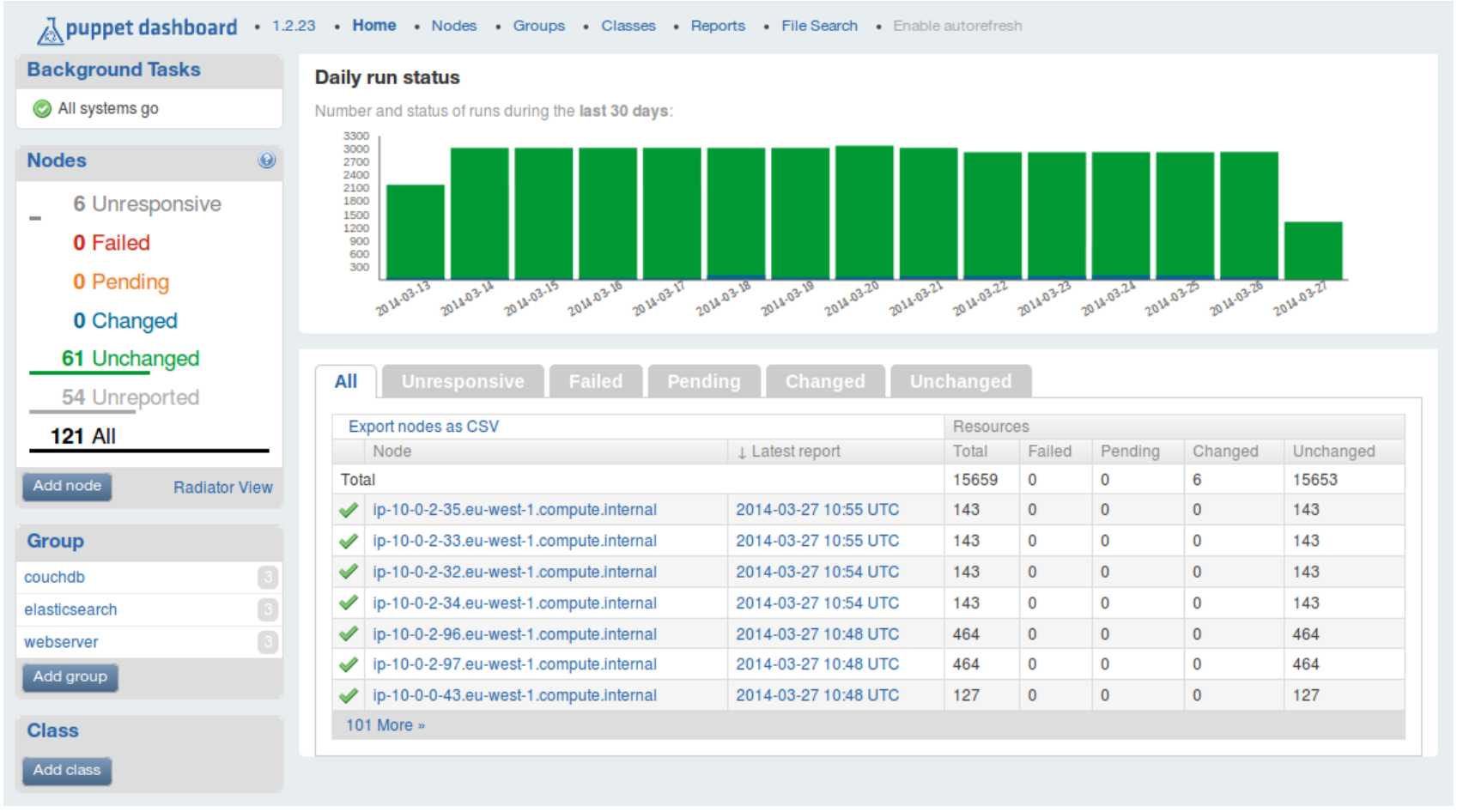


Puppet Module Layout

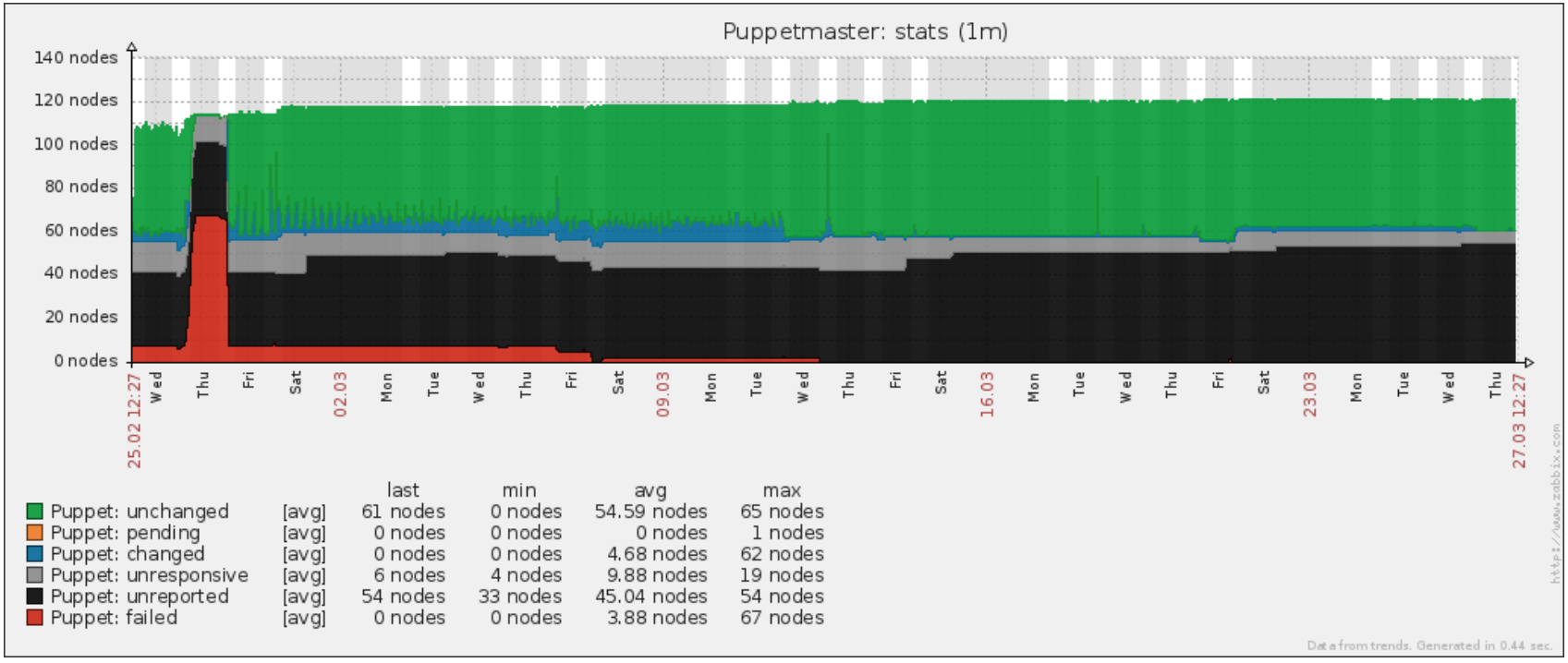
`module_name`

- `manifests` — Puppet code (classes/defines)
  - `init.pp`
  - `subclass.pp`
- `files` — static files
- `templates` — .erb templates
- `lib` — ruby plugins (custom types/facts)
- `tests` — usage examples for manifests
- `spec` — spec tests for libs

Puppet Dashboard



External Monitoring



stdlib facts.d

- simple data input
- e.g. `ec2metadata`, inventory lookup

custom_facts.sh

```
#! /bin/sh

which ec2metadata >/dev/null 2>&1 || exit 1

echo "ec2_ami_id=$(ec2metadata --ami-id)"
echo "ec2_instance_id=$(ec2metadata --instance-id)"
echo "ec2_instance_type=$(ec2metadata --instance-type)"
echo "ec2_public_ipv4=$(ec2metadata --public-ipv4)"
echo "ec2_public_hostname=$(ec2metadata --public-hostname)"
```

Hiera

- banish top scope variables
- use Hiera!
- structure with roles & profiles

node definitions vs. Hiera

site.pp

```
node "mydev.vagrantup.com" inherits basenode-vagrant {
    $vmEnv = "development"
    include sysadmin
    include ntp
    include vagrant
    include user::vagrant
    include mysqlserver
    include redisserver

    # ...
}
```

node definitions vs. Hiera

site.pp

```
hiera_include('include_classes', ['sysadmin'])

node default {
}
```

role_elasticsearch.yaml

```
include_classes:
  - elasticsearch
  - elasticsearch::plugins
  - zabbix::helper::elasticsearch
elasticsearch::clustername: "mycluster"
elasticsearch::client: false
elasticsearch::heapsize: "768m"
```

hiera.yaml

```
:hierarchy:
    - node/%{fqdn}
    - vm/netenv_role_%{puppet_netenv}_%{puppet_role}
    - vm/role_%{puppet_role}
    - vm/netenv_%{puppet_netenv}
    - domain_%{domain}
    - common

:backends:
    - yaml

:logger: console

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

Example lookup

```
fqdn = dev.pod1.org
domain = pod1.org
puppet_role = dev
puppet_netenv = vagrant
```

⇒ Lookup in:

1. `node/dev.pod1.org.yaml`
2. `vm/netenv_role_vagrant_dev.yaml`
3. `vm/role_dev.yaml`
4. `vm/netenv_vagrant.yaml`
5. `domain_pod1.org.yaml`
6. `common.yaml`

Hiera & Puppet 2.x compatibility

```
class vpn($version = hiera('vpn::version', 'present'),
          $ca_crt  = hiera('vpn::ca_crt'),
          $usr_crt = hiera('vpn::usr_crt'),
          $usr_key = hiera('vpn::usr_key')) {
  package {
    'openvpn':
      ensure => $version;
  }

  # ...
}
```

git workflow

- use git!
- use git hooks
- use per-user environments for easy testing
- repos for testing/production

git hook: Syntax Check

Git `pre-commit` hook with `puppet-lint` to syntax check Puppet, ERB templates, YAML files (`http://github.com/gini/puppet-git-hooks`)

Example Output:

```
$ git commit -m 'test' modules/graylog2/templates/server.conf.erb
-:5: syntax error, unexpected $undefined
...rd_sha2 = "; _erbout.concat(( @ root_pwd_sha2 ).to_s); _erbo...
...                                 ^
ERB syntax error in modules/graylog2/templates/server.conf.erb
```

git hook: E-Mail Notification

Git `post-receive` hook to notify team on push (`http://git.kernel.org/cgit/git/git.git/tree/contrib/hooks/post-receive-email?id=HEAD`)

Example E-Mail:

```
- Log -----------------------------------------
commit 5df04ee883b8de8a37bf0ac97eec068cd1f3a414
Author: N. N. <n.n@deck36.de>
Date:   Tue Jan 7 08:57:17 2014 +0000

    fixed path to csync2 executable

-----------------------------------------------

Summary of changes:
 modules/user/files/etc/sudoers.d/support |    2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
```

environments

- per user env + `production`

⇒ easy testing with `puppet agent -t --environment=user`

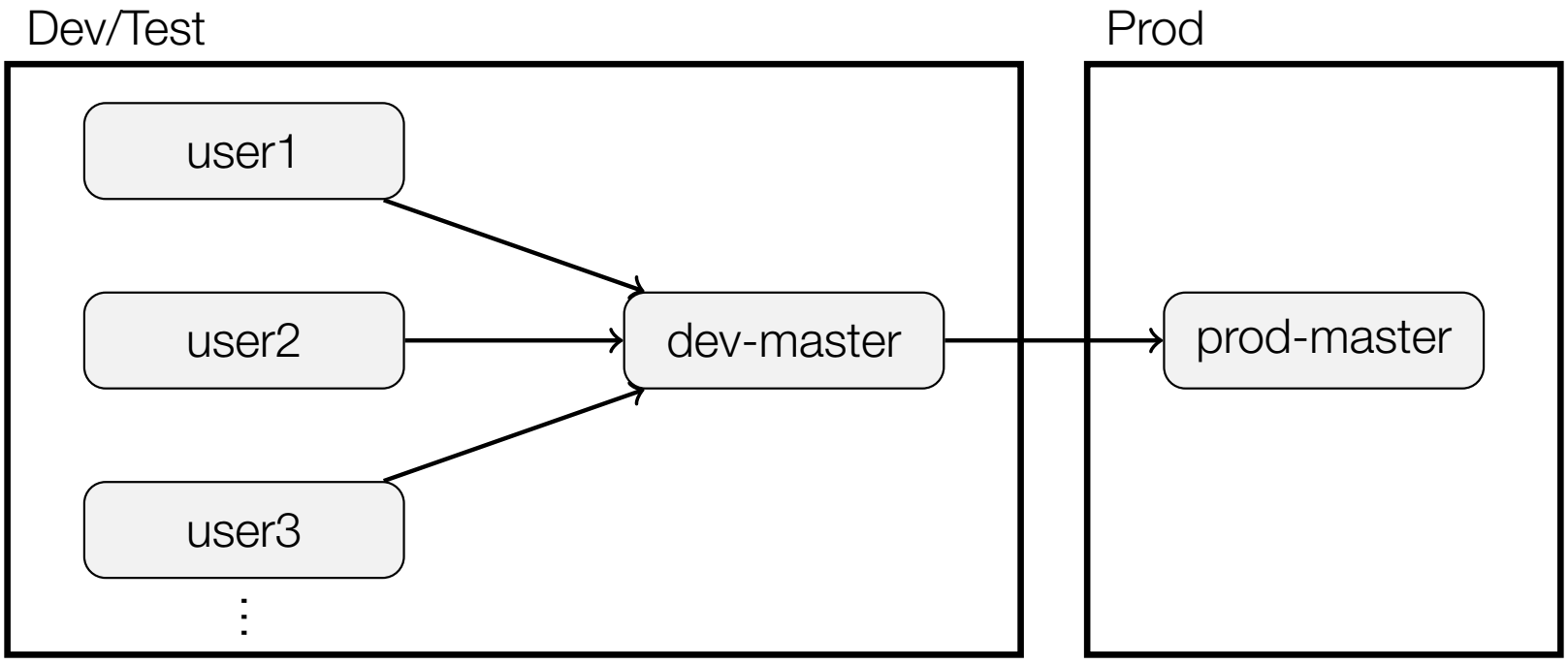
- two servers for testing/production

Config (in puppet < 3.5.0):

puppet.conf

```
[mschuette]
modulepath = $confdir/environments/mschuette/modules
manifest = $confdir/environments/mschuette/manifests/site.pp
pluginsync = true
```

environments



Puppet Problems

- some tasks require two agent runs
- `apt-get upgrade` and package dependencies
- beware of version mismatch between `apt` (or `yum`) and `package`
- scoping and namespaces
- `exec` is the new `eval`

Version ping-pong

modules/php/init.pp

```
class php($version = '5.3.10-1ubuntu3.10') {
    package { 'php5-common':
        ensure => $version,
    }
}
class php::curl($version) {
    require php
    package { 'php5-curl':
        ensure => $version,
    }
}
```

server.pp

```
class { 'php::curl':
    version => '5.5.5+dfsg-1+debphp.org~precise+2',
}
```

Namespace problems

```
# this does not work, cf. #PUP-1073

package { 'memcached':
    ensure   => present,
    provider => apt,
}

package { 'memcached':
    ensure   => present,
    provider => gem,
}
```

`exec` tricks

You can do (and break) everything with `exec`.

But of course you should not.

exec tricks

```
# no pkg provider for npm
exec { 'npm install -g less':
    creates => '/usr/lib/node_modules/npm/node_modules/less',
}

# hide change
exec { 'zabbix_update.sh':
    command     => 'false',
    onlyif      => "/opt/zabbix_update.sh $api_url && false",
    logoutput   => on_failure,
}
```

MCollective

“multissh deluxe”

AMQP client/server framework to

- orchestrate actions
- control puppet agents
- run commands
- query resources
- …

Hooks to other systems

- include in provisioning process
- provide normative data as facts
- register or update DNS name → Route 53
- register or update host in Zabbix monitoring → API

Versions

- Puppet 2.7: legacy
- Puppet 3.0: major upgrade, with Hiera support
- Puppet 3.x: current development, future parser

Questions?

```
class presentation {
    package { 'questions':
        ensure => 'answered',
    }
}
```

Links:

- Vagrant
- Puppet Language: Visual Index
- Puppet Type Reference
- Puppet Ask

Thank You