Building dynamic networks with puppet

Thomas Uphill
thomas@narrabilis.com

http://goo.gl/nR9rti

latest slides

https://github.com/uphillian/puppetconf2013

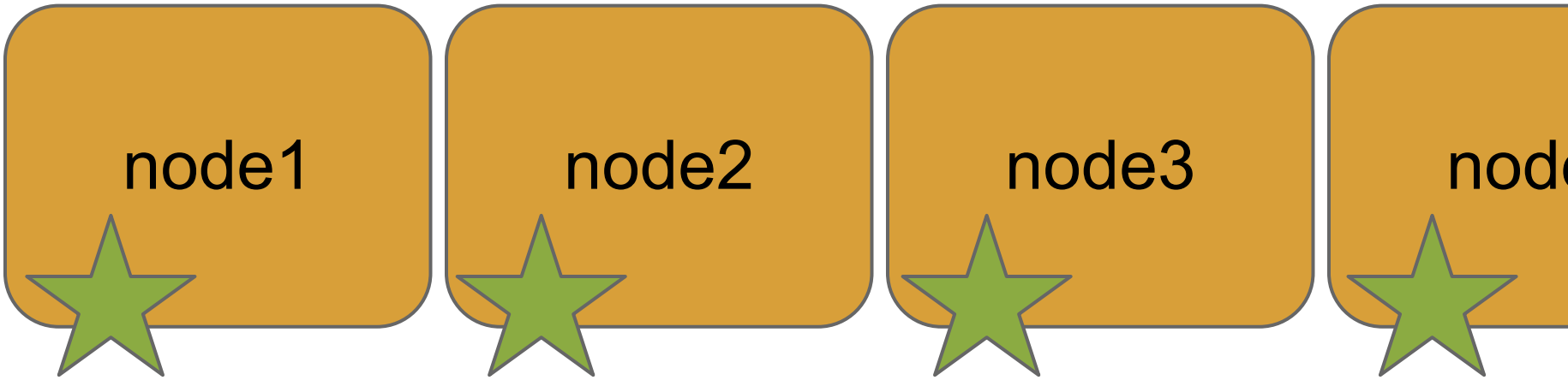
latest code

Me

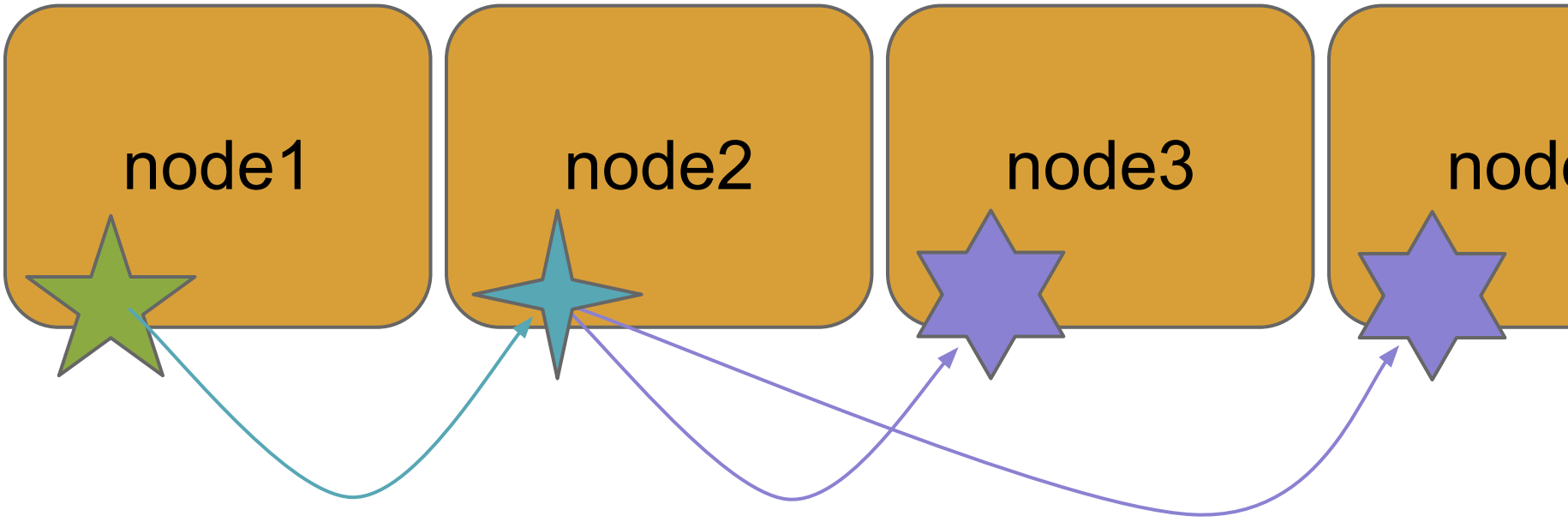
IAS School of Mathematics

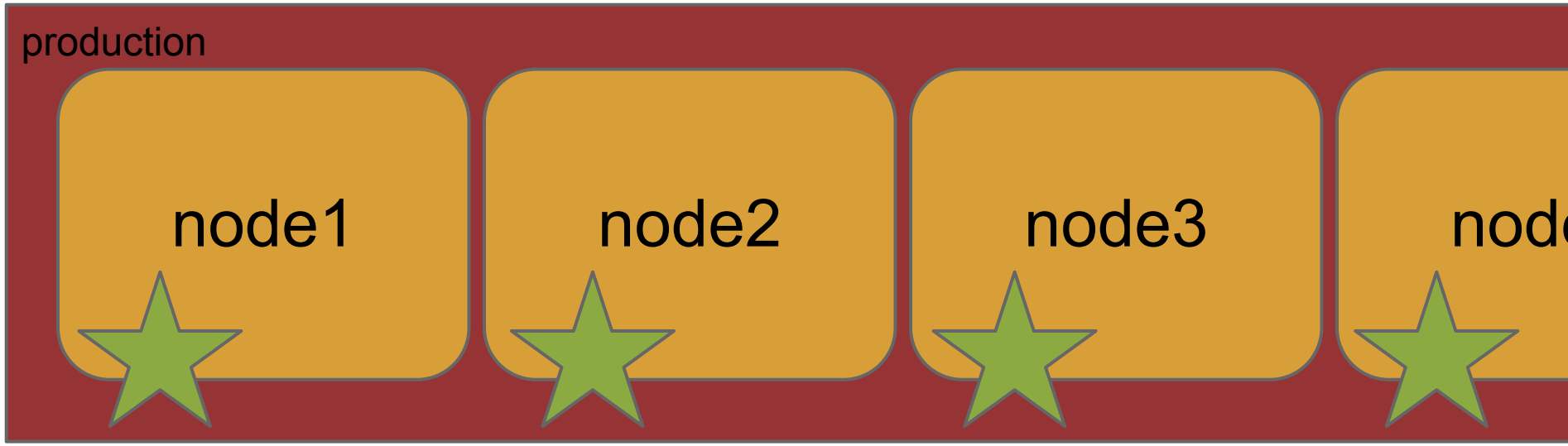
COSTCO WHOLESALE

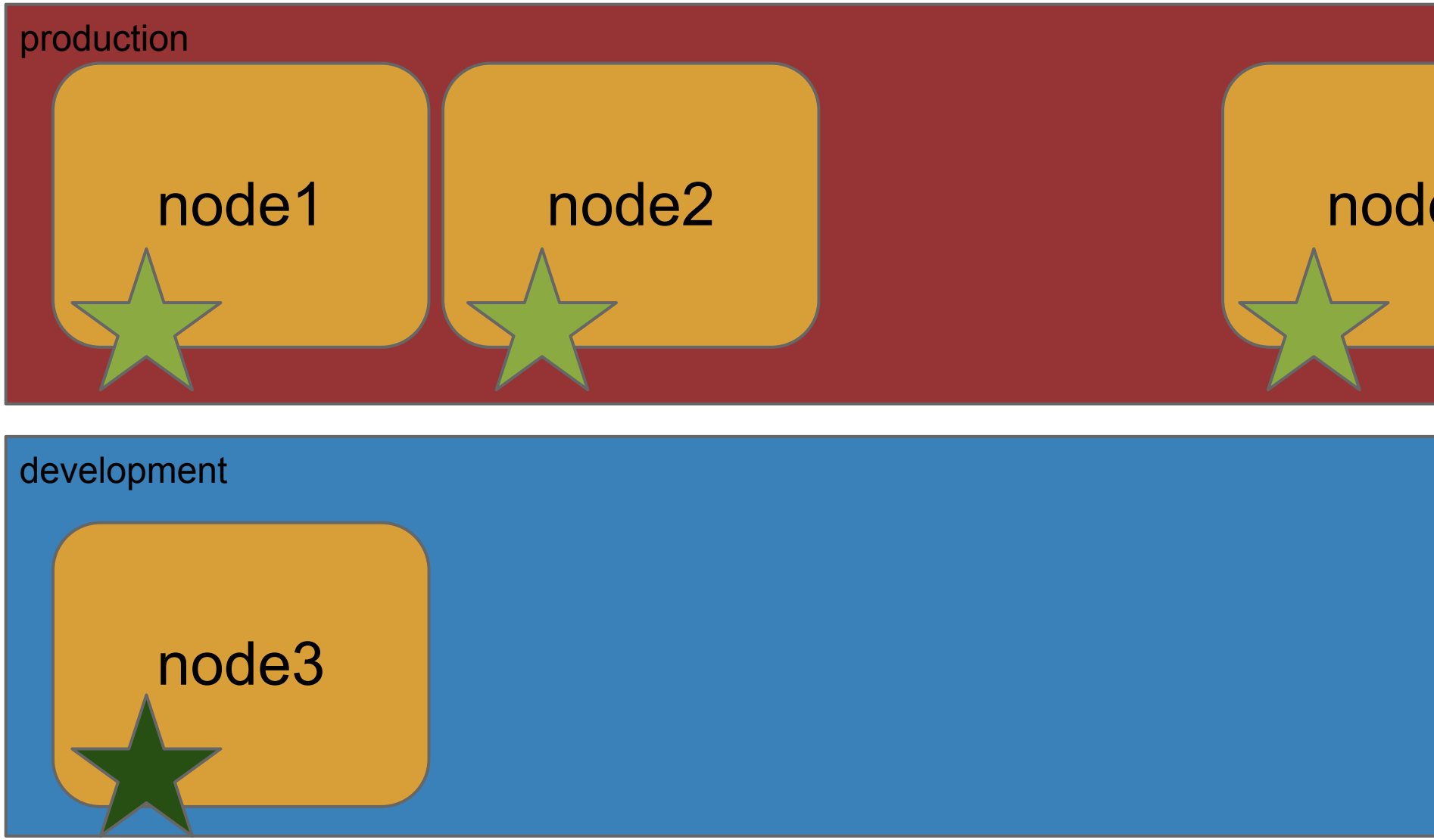
About this talk...



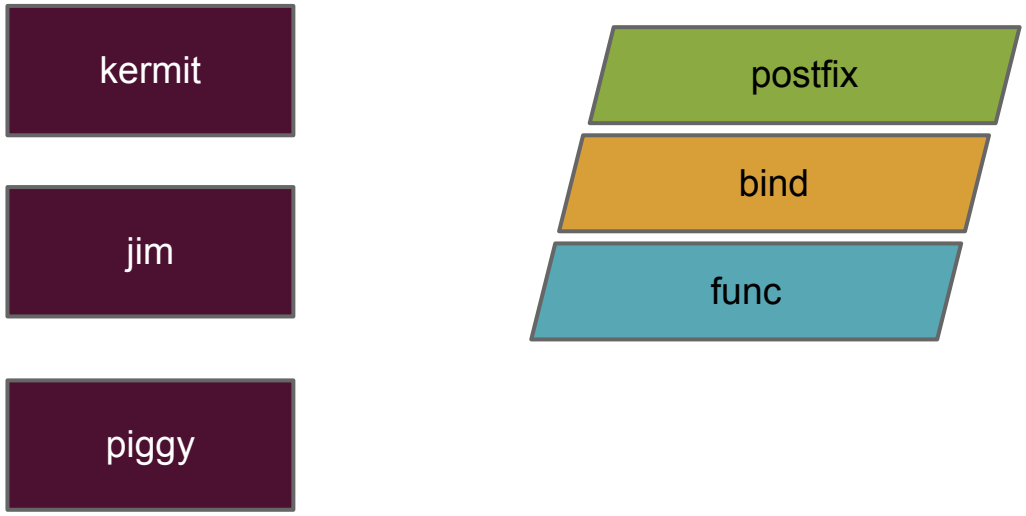
About this talk...



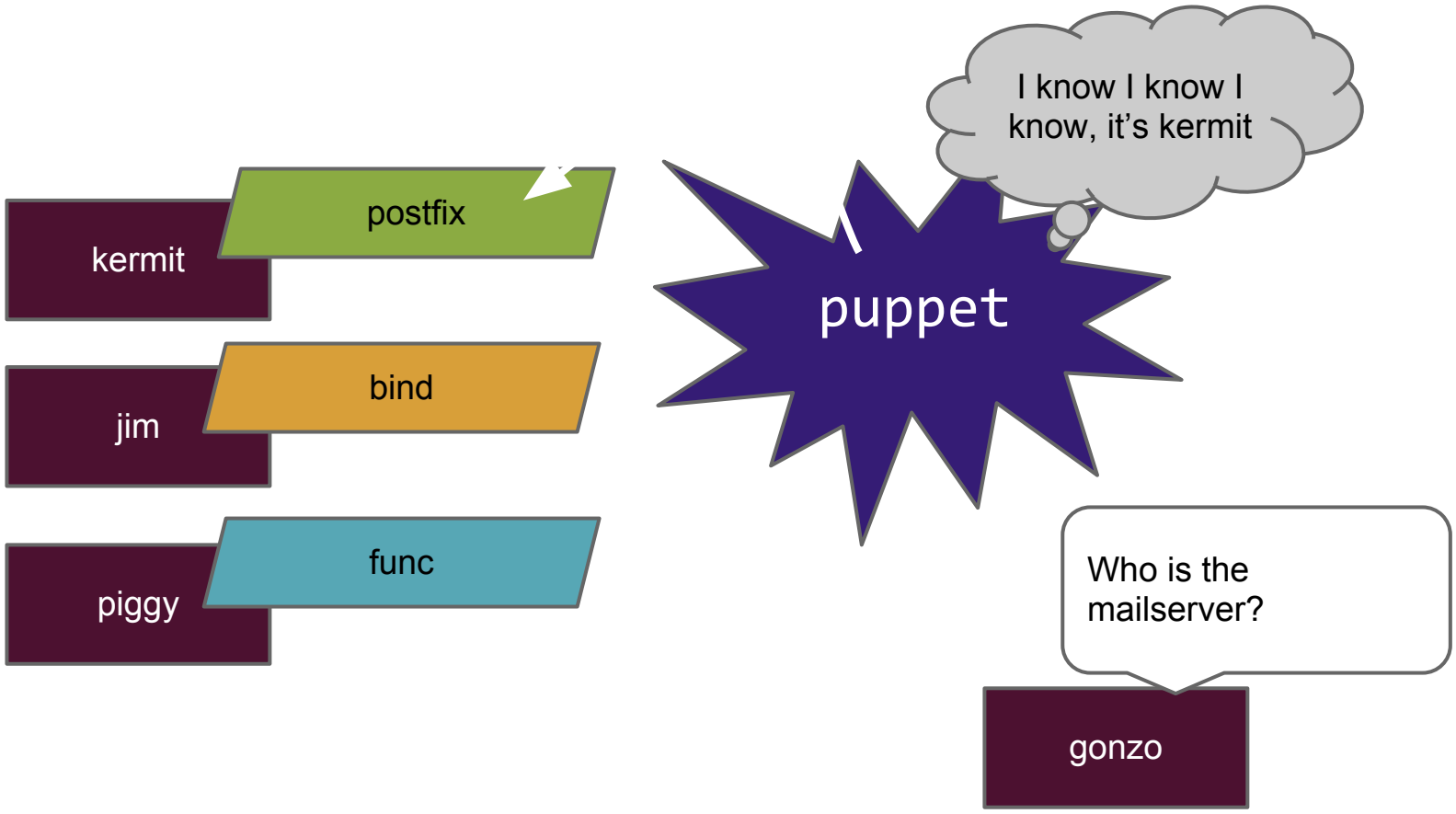

production
node1
node2
node3


production
node1
node2
development
node3

goal



goal


I know I know I know, it's kermit
postfix
kermit
puppet
bind
jim
func
piggy
Who is the mailserver?
gonzo

Outline

postfix

augeas
exported resources
tags

augeas

- manipulate specific parts of a file
- files as objects
- type in puppet

augeas

http://augeas.net/tour.html

http://projects.puppetlabs.com/projects/1/wiki/puppet_augeas

```
default=0
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title Springdale Linux  (2.6.32-358.6.2.el6.x86_64)
        root (hd0,0)
        kernel /vmlinuz-2.6.32-358.6.2.el6.x86_64 ro root=/dev/mapper/vg_jim-lv_root
rd_NO_LUKS LANG=en_US.UTF-8 rd_LVM_LV=vg_jim/lv_swap rd_NO_MD
SYSFONT=latarcyrheb-sun16 crashkernel=auto rd_LVM_LV=vg_jim/lv_root
KEYBOARDTYPE=pc KEYTABLE=us rd_NO_DM rhgb quiet
        initrd /initramfs-2.6.32-358.6.2.el6.x86_64.img
title Springdale-PUIAS Linux (2.6.32-358.el6.x86_64)
        root (hd0,0)
        kernel /vmlinuz-2.6.32-358.el6.x86_64 ro root=/dev/mapper/vg_jim-lv_root rd_NO_LUKS
LANG=en_US.UTF-8 rd_LVM_LV=vg_jim/lv_swap rd_NO_MD SYSFONT=latarcyrheb-sun16
crashkernel=auto rd_LVM_LV=vg_jim/lv_root  KEYBOARDTYPE=pc KEYTABLE=us rd_NO_DM
rhgb quiet
        initrd /initramfs-2.6.32-358.el6.x86_64.img
```

augeas


default=0
timeout=5
splashimage=(hd0,0
hiddenmenu
title Springdale Linux
root (hd0,0)
kernel /vmlin
rd_NO_LUKS LANG
SYSFONT=latarcyrh
KEYBOARDTYPE=p
initrd /initram
title Springdale-PUIA
root (hd0,0)
kernel /vmlin
LANG=en_US.UTF-
crashkernel=auto rd_
rhgb quiet
initrd /initram
[user@host] $ augtool
augtool> set /files/etc/grub.conf/default 1
augtool> rm /files/etc/grub.conf/title[2]
rm : /files/etc/grub.conf/title[2] 18
augtool> save
Saved 1 file(s)

augeas


```
default=0
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title Springdale Linux  (2.6.32-358.6.2.el6.x86_64)
        root (hd0,0)
        kernel /vmlinuz-2.6.32-358.6.2.el6.x86_64 ro root=/dev/mapper/vg_jim-lv_root
rd_NO_LUKS LANG=en_US.UTF-8 rd_LVM_LV=vg_jim/lv_swap rd_NO_MD
SYSFONT=latarcyrheb-sun16 crashkernel=auto rd_LVM_LV=vg_jim/lv_root
KEYBOARDTYPE=pc KEYTABL
        initrd /initramfs-2.6.32-35
title Springdale-PUIAS Linux (2.6
        root (hd0,0)
        kernel /vmlinuz-2.6.32-35
LANG=en_US.UTF-8 rd_LVM_LV
crashkernel=auto rd_LVM_LV=vg
rhgb quiet
        initrd /initramfs-2.6.32-35
```

```
default=1
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title Springdale Linux  (2.6.32-358.6.2.el6.x86_64)
        root (hd0,0)
        kernel /vmlinuz-2.6.32-358.6.2.el6.x86_64 ro root=/dev/mapper/vg_jim-lv_root
rd_NO_LUKS LANG=en_US.UTF-8 rd_LVM_LV=vg_jim/lv_swap rd_NO_MD
SYSFONT=latarcyrheb-sun16 crashkernel=auto rd_LVM_LV=vg_jim/lv_root
KEYBOARDTYPE=pc KEYTABLE=us rd_NO_DM rhgb quiet
        initrd /initramfs-2.6.32-358.6.2.el6.x86_64.img
```

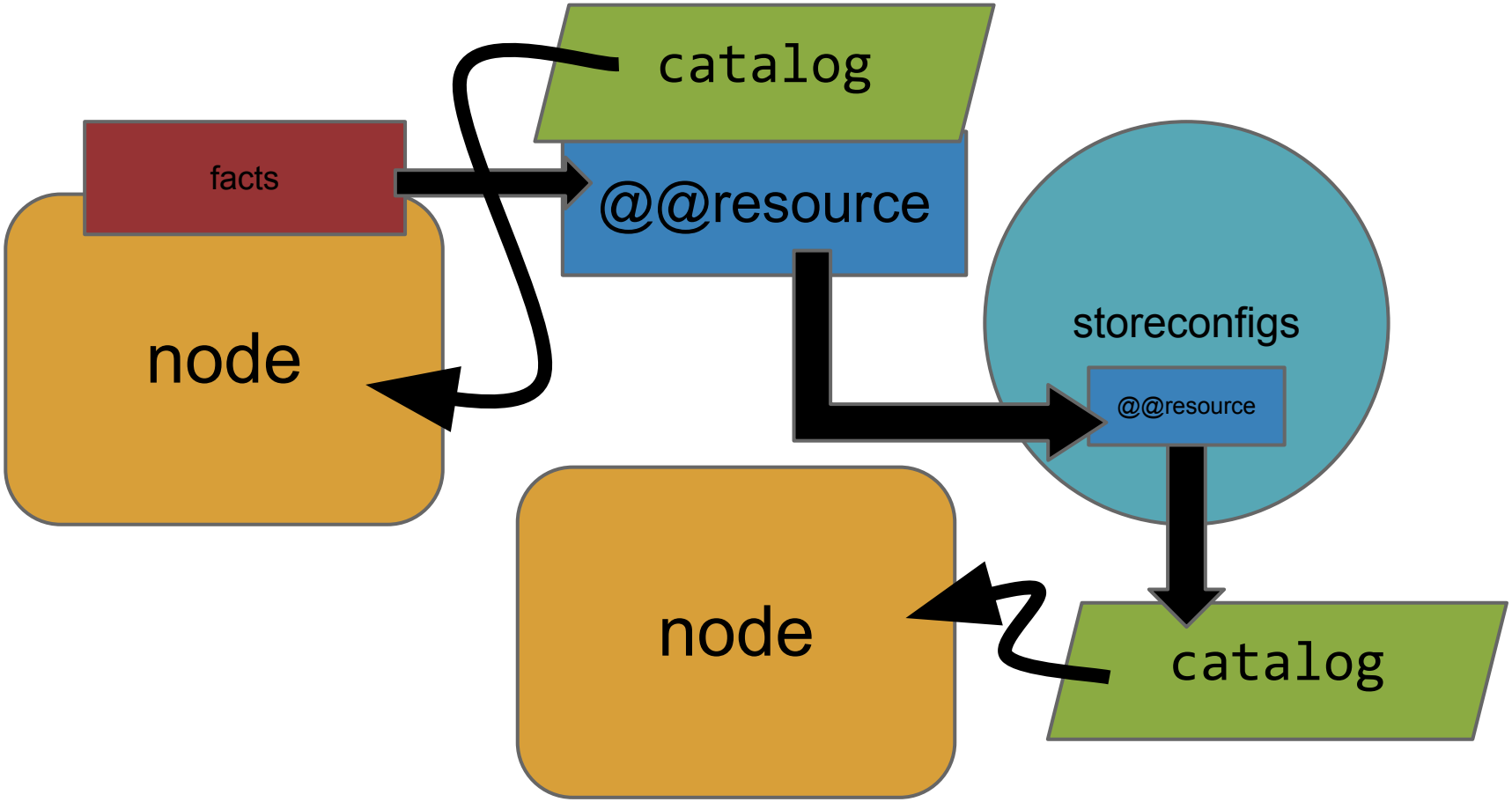
augeas


```
[user@host] $ augtool
augtool> set /files/etc/grub.conf/default 1
augtool> rm /files/etc/grub.conf/title[2]
rm : /files/etc/grub.conf/title[2] 18
augtool> save
Saved 1 file(s)
```

```
augeas { ‘remove second entry, set default’:
  context => ‘/files/etc/grub.conf’,
  changes => [
    ‘set default 1’,
    ‘rm title[2]’
  ],
}
```

exported resources

http://docs.puppetlabs.com/guides/exported_resources.html
http://docs.puppetlabs.com/puppet/2.7/reference/lang_exported.html



exported resources

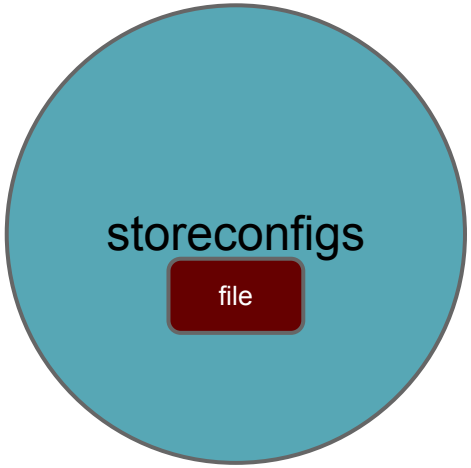
- any class/type can be exported
- classic example ssh keys

```
class ssh {
    @@sshkey { $hostname: type => dsa, key => $sshdsakey }
    Sshkey <<| |>>
}
```

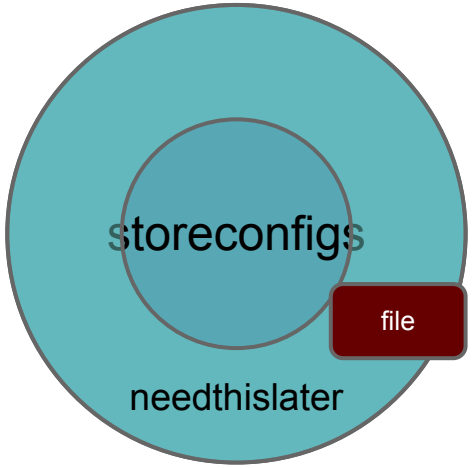
tags

http://projects.puppetlabs.com/projects/1/wiki/Using_Tags

```
file {'/some/file':
}
```




```
file {'/some/file':

}
```



postfix

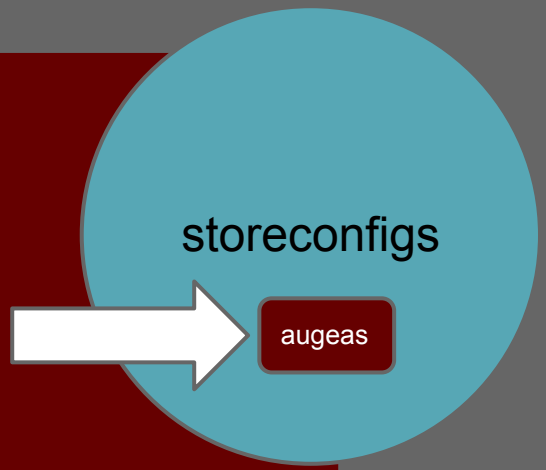
- only accepts mail from puppet nodes
- update itself

postfix
access.conf

<ipaddress> OK
<ipaddress> RELAY

postfix client

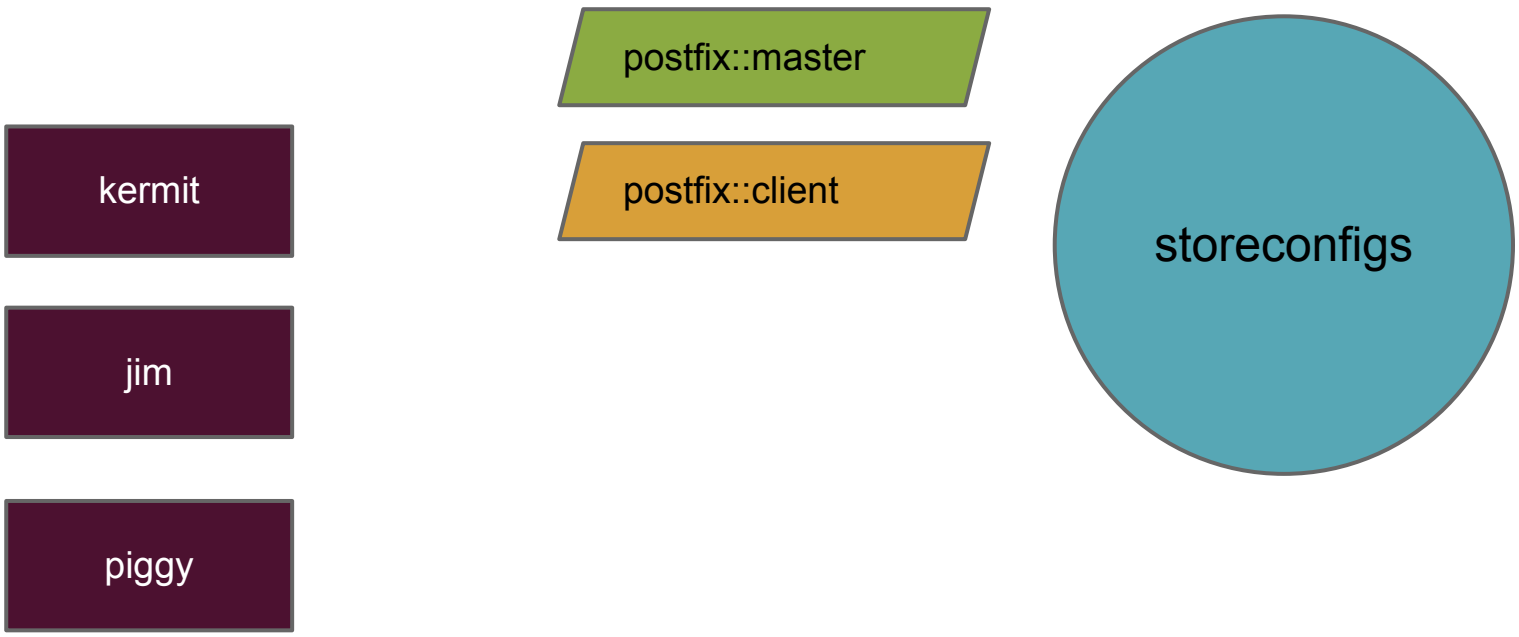
```
class postfix::client {
  # this is a client of the master, set our config file to use the master
  # add our name to the access list on the master
    @@augeas {"access by $::hostname":
            context => "/files/etc/postfix/access",
            changes => [
              "ins 0 after *[last()]",
              "set 0/pattern $::ipaddress",
              "set 0/action OK",
            ],
            onlyif => "match *[ pattern = '$::ipaddress'] size == 0",
       tag   => 'postfix_access',
     }
}
```



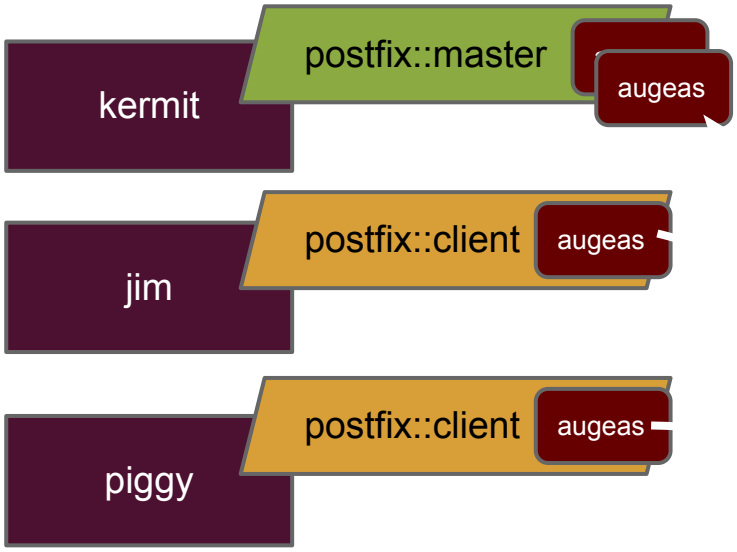
postfix
master

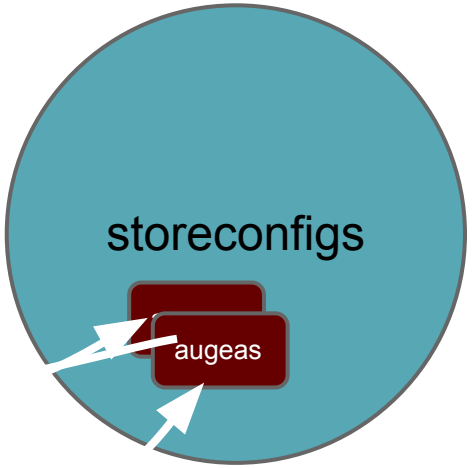
```
class postfix::master {
    Augeas <<| tag == 'postfix_access' |>>  {
      notify => Exec['postfix rebuild access']
    }

  exec { 'postfix rebuild access':
    path        => '/bin:/usr/bin',
    command     => '/usr/sbin/postmap /etc/postfix/access',
    refreshonly => true,
  }
}
```

postfix


kermit
jim
piggy
postfix::master
postfix::client
storeconfigs

postfix

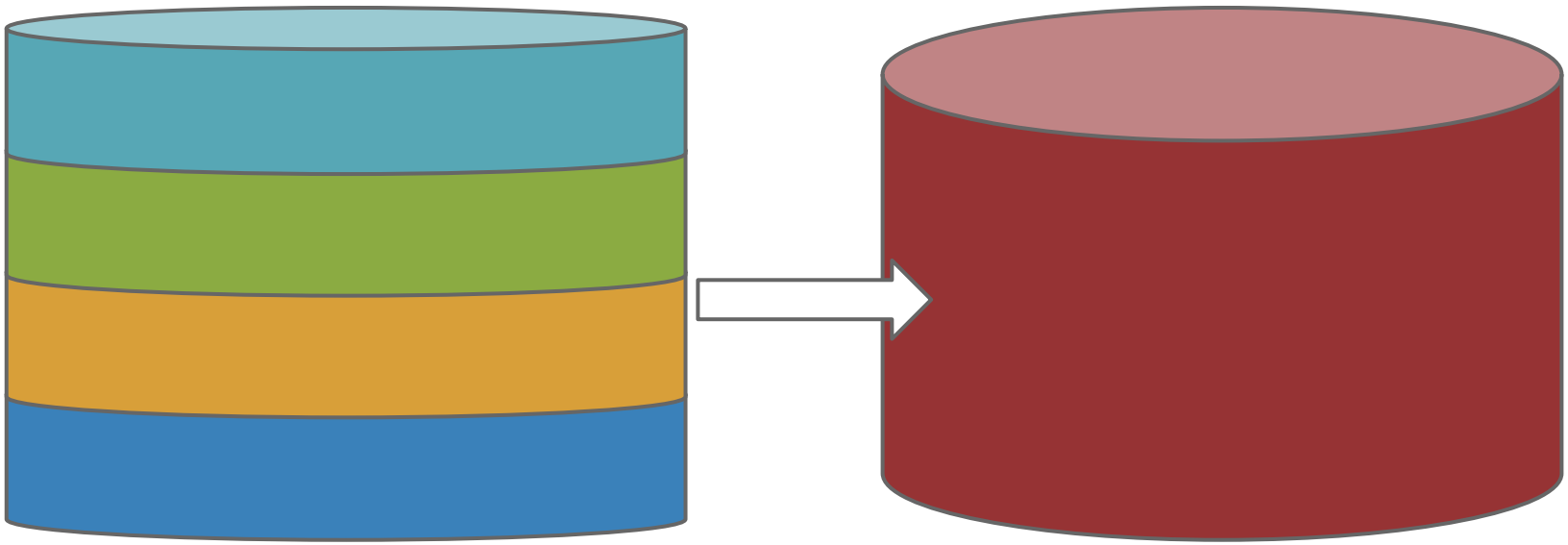

kermit
postfix::master
augeas
jim
postfix::client
augeas
piggy
postfix::client
augeas


storeconfigs
augeas

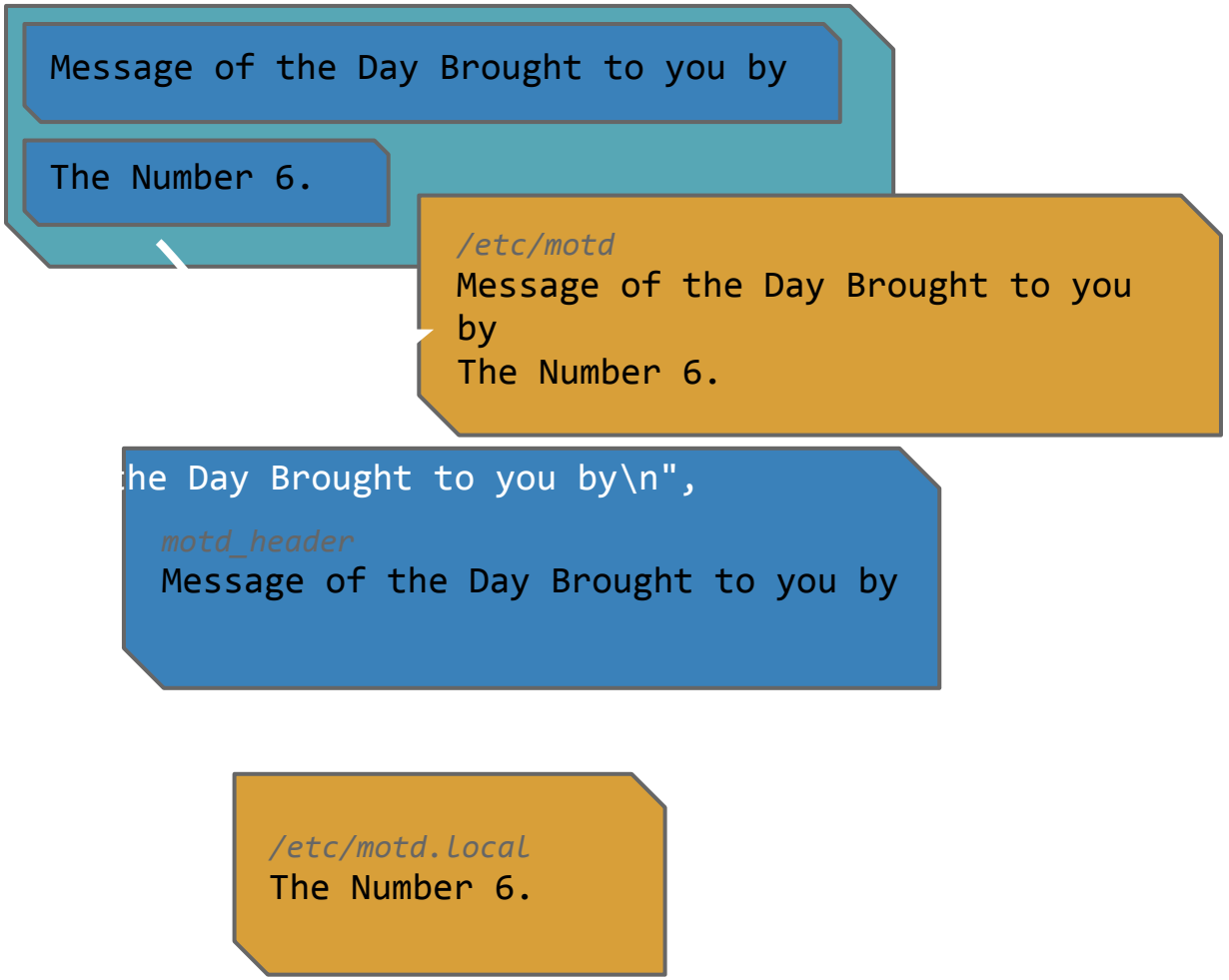
dns

concat
custom facts
firewall
exported resources
tags

concat



concat


Message of the Day Brought to you by
The Number 6.
/etc/motd
Message of the Day Brought to you by
The Number 6.
he Day Brought to you by\n",
motd_header
Message of the Day Brought to you by
/etc/motd.local
The Number 6.

custom facts

http://docs.puppetlabs.com/guides/custom_facts.html

modules/four/lib/facter/four.rb

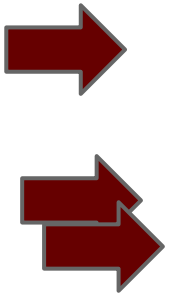
```
Facter.add("four") do
  setcode do
    2+2
  end
end
```

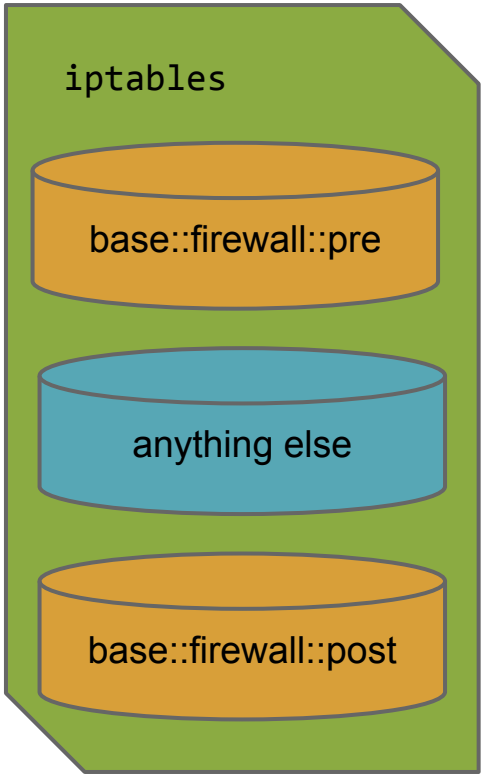
```
[root@jim ~]# puppet agent -t --pluginsync
info: Loading facts in /var/lib/puppet/lib/facter/four.rb
[root@jim ~]# facter -p four
4
```

firewall module

https://github.com/puppetlabs/puppetlabs-firewall

firewall module





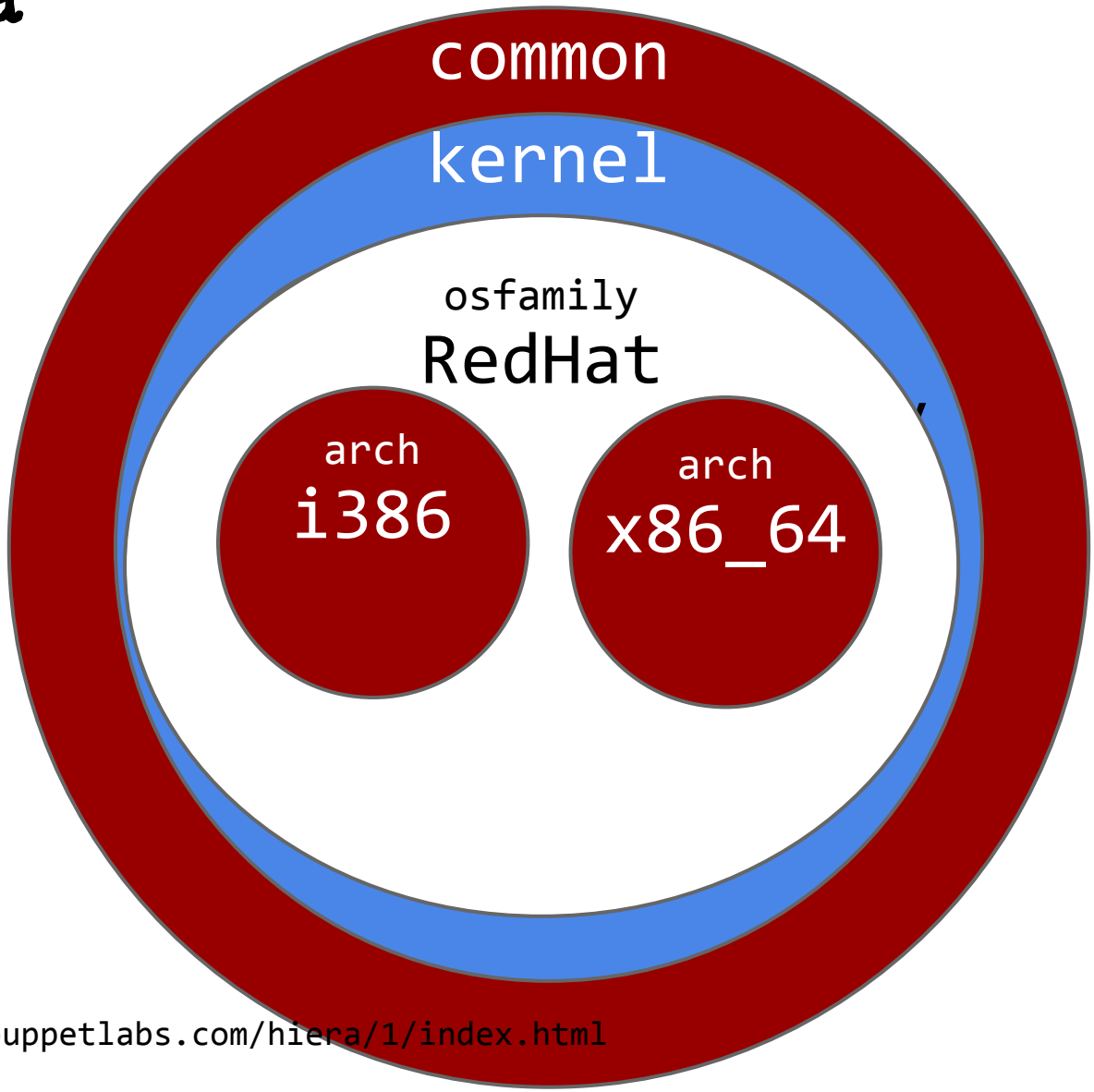
firewall module

firewall module

firewall module

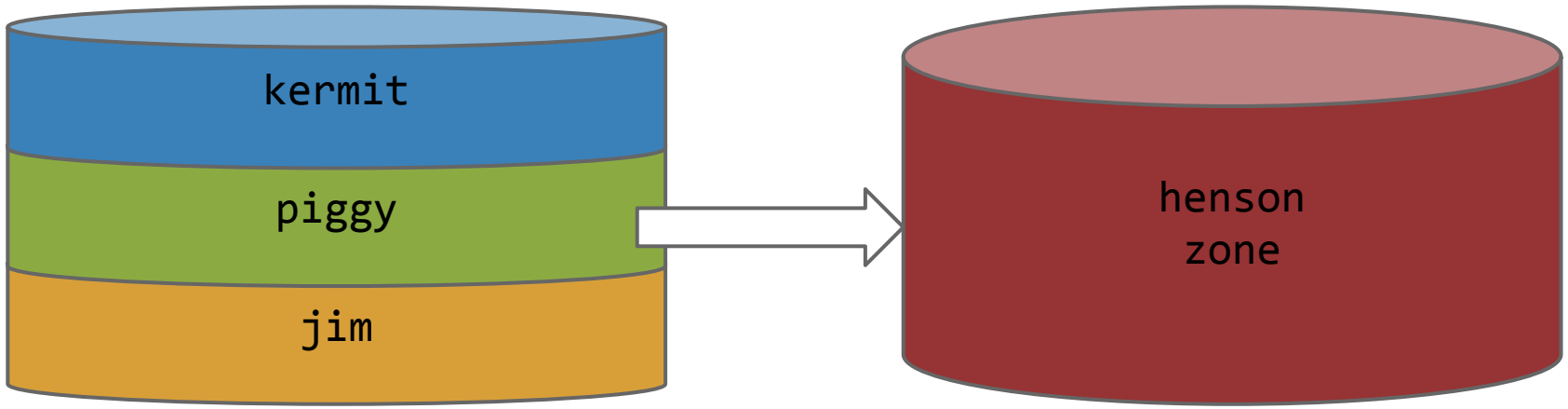
never assume

hiera

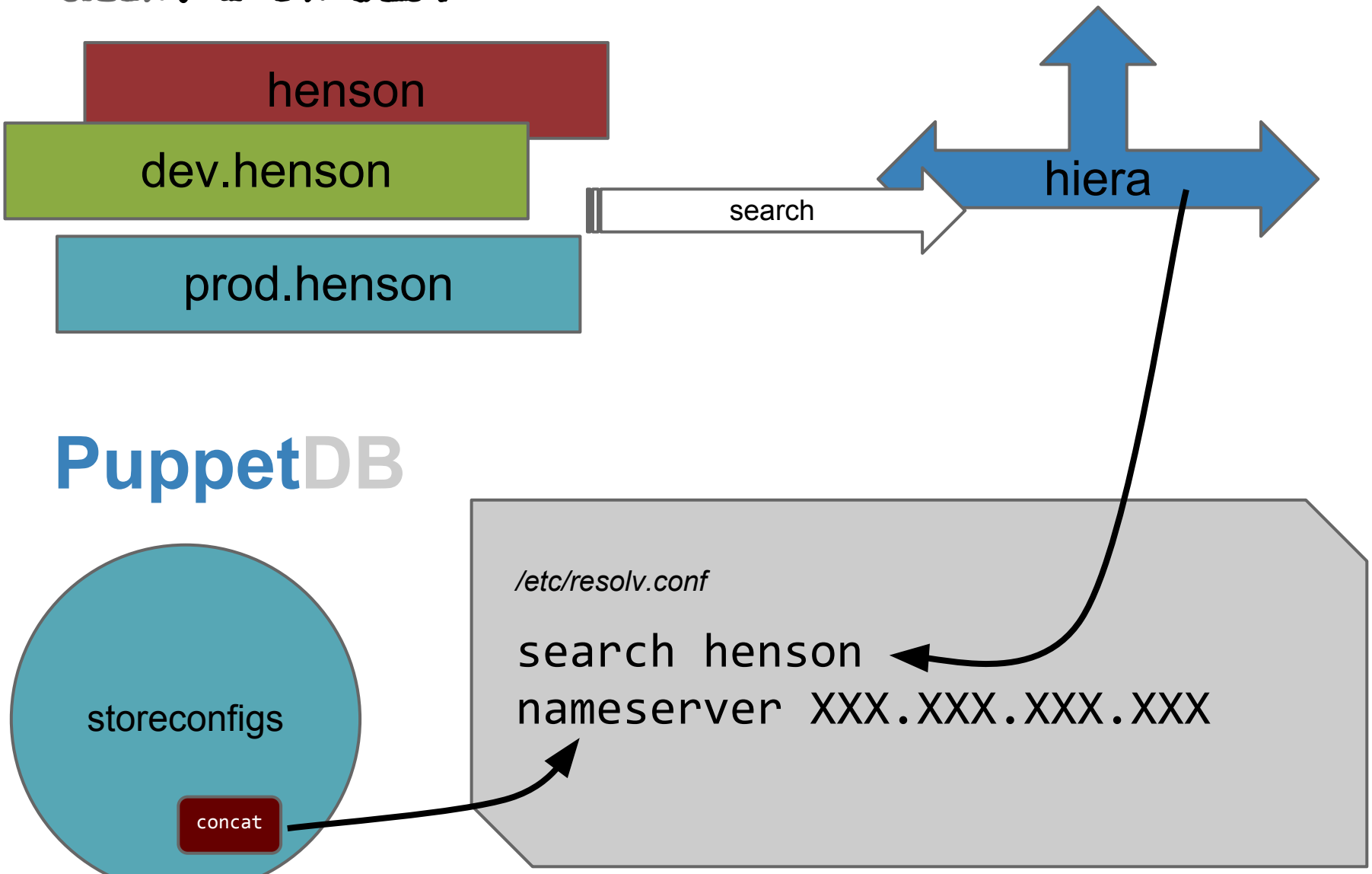


http://docs.puppetlabs.com/hiera/1/index.html

dns/resolv



dns/resolv

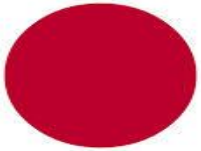

henson
dev.henson
prod.henson
search
hiera
PuppetDB
storeconfigs
concat
/etc/resolv.conf
search henson
nameserver XXX.XXX.XXX.XXX

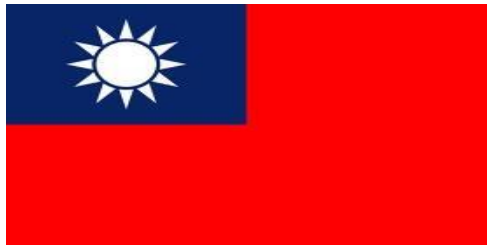
dns/resolv

zone (fact)

production

development



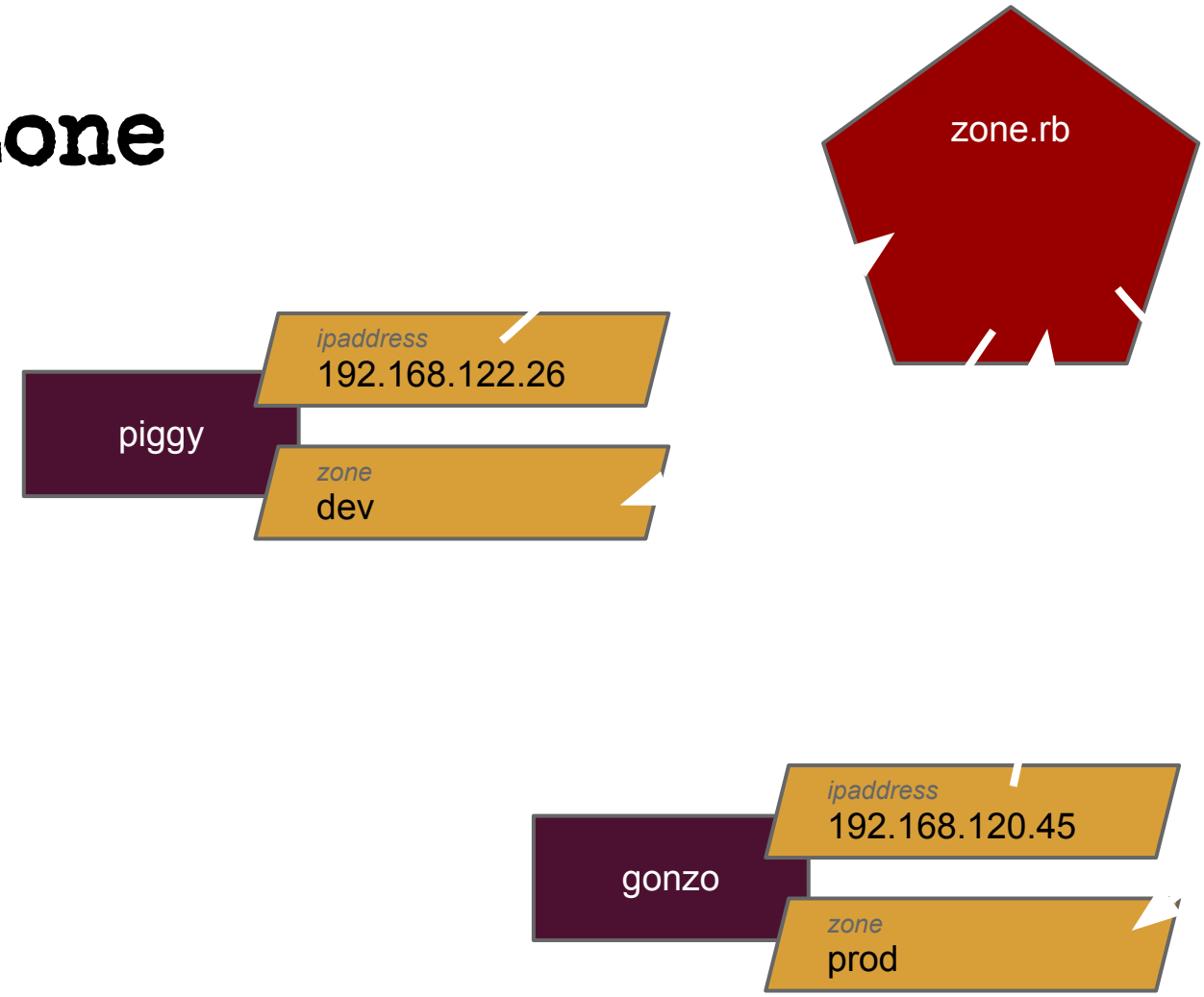


zone

- define custom fact based on ip address of the node

```
require('ipaddr')
ip = IPAddr.new(Facter.value('ipaddress'))

zones = {
 'prod' => IPAddr.new('192.168.120.0/23'),
 'dev'  => IPAddr.new('192.168.122.0/23'),
 'sbx'  => IPAddr.new('192.160.124.0/23'),
 }

zone = 'undef'


for z in zones.keys do
  if zones[z].include?(ip)
     zone = z
  end
end

Facter.add("zone") do
  setcode do zone end
end
```

zone


zone.rb
ipaddress
192.168.122.26
piggy
zone
dev
ipaddress
192.168.120.45
gonzo
zone
prod

zone

hiera.yaml

```
:hierarchy:
  - zones/%{    }
  - environments/%{environment}
  - global
```

zones/dev.yaml

```
---
dns::search: "dev.henson henson corp.henson"
```

zones/prod.yaml

```
---
dns::search: "corp.henson henson"
```

```
require 'ipaddr'
require 'puppet/util/ipcidr'
def reverse(dev)
  ip = IPAddr.new(Facter.value("network_#{dev}"))
  nm = Puppet::Util::IPCidr.new(Facter.value("network_#{dev}")).mask(Facter.value("netmask_#
{dev}"))
  cidr = nm.cidr
  # set fact for network in reverse vvv.www.uuu.in-addr.arpa
  Facter.add("reverse_#{dev}") do
    setcode do ip.reverse.to_s[2..-1] end
  end
  # set fact for network in cidr notation
  Facter.add("network_cidr_#{dev}") do
    setcode do cidr end
  end
end
# loop through the interfaces, defining the two facts for each
interfaces = Facter.value('interfaces').split(',')
interfaces.each do
  |eth| reverse(eth)
end
```

network_cidr_ethX
reverse_ethX

named.conf.erb

gonzo

ipaddress
192.168.120.45

```
options {
  listen-on port 53 { 127.0.0.1; <%= @ipaddress_eth0 -%>;};
  ...
  allow-query     { localhost; <%- interfaces.split(',').each do |eth| if has_variable?
("network_cidr_#{eth}") then -%><%= scope.lookupvar("network_cidr_#{eth}") -%>;<%- end end -%> };
  ...
};
```

network_cidr_eth0
192.168.120.0/23

zone
prod

```
zone "<%= @zone -%>.henson" IN {
  type master;
  file "zone.<%= @zone -%>.henson";
  allow-update { none; };
};

zone "<%=
  type master;
  file "reverse.<%= @reverse_eth0 -%>";
};
```

reverse_eth0
120.168.192.in-addr.arpa

named.conf.erb

```
/etc/named.conf
options {
  listen-on port 53 { 127.0.0.1; 192.168.120.45;};
  listen-on-v6 port 53 { ::1; };
  directory   "/var/named";
  dump-file   "/var/named/data/cache_dump.db";
        statistics-file "/var/named/data/named_stats.txt";
        memstatistics-file "/var/named/data/named_mem_stats.txt";
  allow-query     { localhost; 192.168.120.0/23;127.0.0.0/8; };
  recursion yes;
  …
};

zone "prod.henson" IN {
  type master;
  file "zone.prod.henson";
  allow-update { none; };
};

zone "120.168.192.in-addr.arpa" {
  type master;
  file "reverse.120.168.192.in-addr.arpa";
};
```

resolv.conf

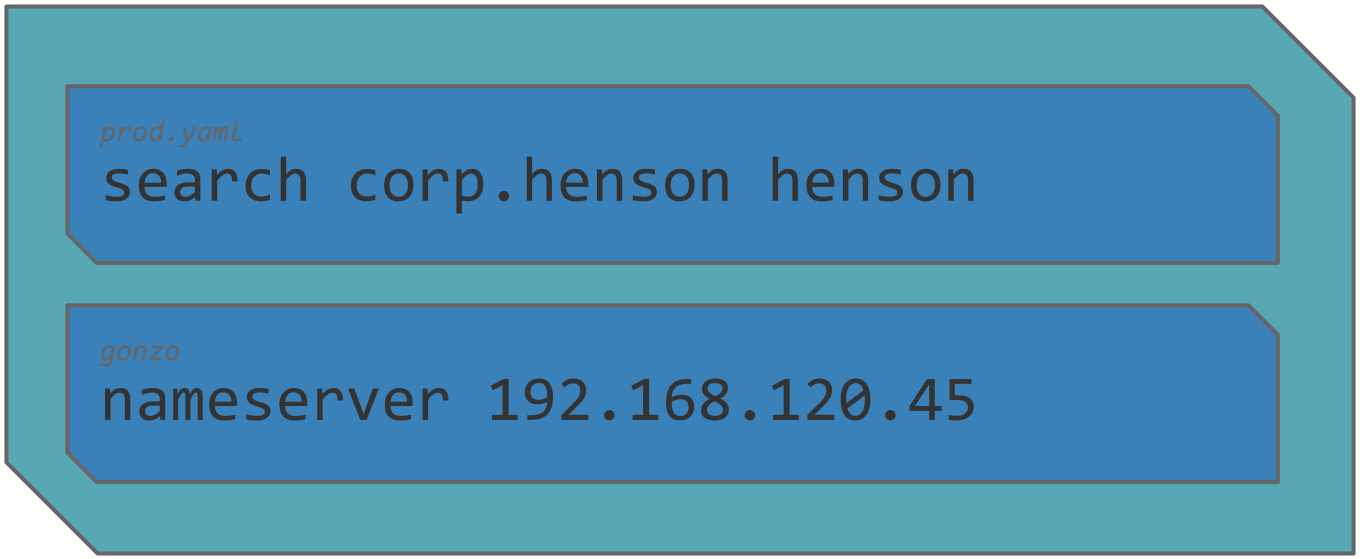
```
class dns::client {
 # pull settings from hiera, sensible defaults
  $domain = hiera('dns::domain','henson')
  $search = hiera('dns::search','henson')
 # include definition of concat for /etc/resolv.conf
  include dns::resolv
 # search the local search value
  concat::fragment{'resolv.conf search':
    target  => '/etc/resolv.conf',
    content => "search $search\n",
    order   => 07,}
 # pull in any nameservers
  Concat::Fragment <<| tag == 'resolv.conf' and tag == "$::zone" |>>
 ...
}
```

```
class dns::resolv {
 concat {'/etc/resolv.conf':
   mode => 0644,
 }
}
```

```
# export ourselves as a dnsserver
  @@concat::fragment {"resolv.conf nameserver $::hostname":
    target  => '/etc/resolv.conf',
    content => "nameserver $::ipaddress\n",
    order   => 10,
    tag     => ['resolv.conf',"$::zone"],
  }
```

resolv.conf

gonzo



zone (named)

```
class dns::server {
 # include zone.henson from everyone else.
  include dns::zones
  Concat::Fragment <<| tag == 'zone' and tag == 'henson' |>>
}
```

```
class dns::client {
  ...
  @@concat::fragment {"zone henson $::hostname":
    target  => '/var/named/zone.henson',
    content => "$::hostname A $::ipaddress\n",
    order   => 10,
    tag     => ['zone','henson'],}
  $lastoctet = regsubst($::ipaddress_eth0,'^([0-9]+)[.]([0-9]+)[.]([0-9]+)[.]([0-9]+)$','\4')
  @@concat::fragment {"zone reverse $::reverse_eth0 $::hostname":
    target  => "/var/named/reverse.$::reverse_eth0",
    content => "$lastoctet PTR $::fqdn\n",
    order   => 10,
    tag     => ['zone','henson'],}
}
```

zone (named)

```
class dns::server {
 # include zone.henson
  include dns::zones
  Concat::Fragment <<|
}
```

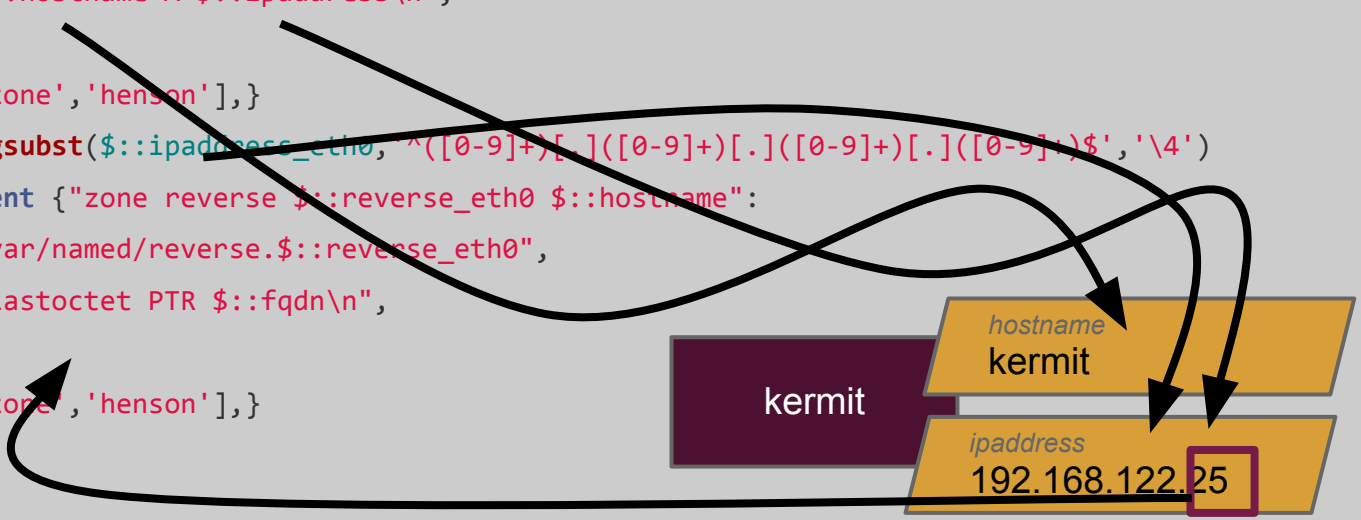
```
class dns::client {
  ...
  @@concat::fragment {"
    target  => '/var/na
    content => "$::host
    order   => 10,
    tag     => ['zone',
  $lastoctet = regsubst
  @@concat::fragment {"
    target  => "/var/na
    content => "$lastoc
    order   => 10,
    tag     => ['zone',
}
```

```
class dns::zones {
 concat {'/var/named/zone.henson':
   mode   => 0644,
   notify => Exec['named reload'],
}
concat::fragment {'zone.henson header':
   target  => '/var/named/zone.henson',
   source => "puppet:///modules/dns/$::zone/zone.henson",
   order   => 01,
 }
 concat {'/var/named/reverse.120.168.192.in-addr.arpa':
   mode   => 0644,
   notify => Exec['named reload'],
 }
 concat::fragment {'reverse.120.168.192.in-addr.arpa header':
   target => '/var/named/reverse.120.168.192.in-addr.arpa',
   source => 'puppet:///modules/dns/reverse/reverse.120.168.192.in-addr.arpa',
   order  => 01,
  }
...
}
```

zone (named)

```
class dns::server {
 # include zone.henson from everyone else.
  include dns::zones
  Concat::Fragment <<| tag == 'zone' and tag == 'henson' |>>
}
```

```
class dns::client {
  ...
  @@concat::fragment {"zone henson $::hostname":
    target  => '/var/named/zone.henson',
    content => "$::hostname A $::ipaddress\n",
    order   => 10,
    tag     => ['zone','henson'],}
  $lastoctet = regsubst($::ipaddress_eth0, '^([0-9]+)[.]([0-9]+)[.]([0-9]+)[.]([0-9]+)$','\4')
  @@concat::fragment {"zone reverse $::reverse_eth0 $::hostname":
    target  => "/var/named/reverse.$::reverse_eth0",
    content => "$lastoctet PTR $::fqdn\n",
    order   => 10,
    tag     => ['zone','henson'],}
}
```



zone (named)

gonzo

```
$ORIGIN henson.
$TTL 1D
@     IN SOA   root hostmaster (
                        2013060102 ; serial
                        8H         ; refresh
                        4H         ; retry
                        4W         ; expire
                        1D )       ; minimum
               NS       jim
               MX       10 jim
;
; just in case someone asks for localhost.henson
localhost      A        127.0.0.1
; CNAMES
puppet         CNAME    jim
jim            A        192.168.122.24
; exported resources below this point
```

```
gonzo A 192.168.120.45
```

```
jim A 192.168.122.24
```

```
piggy A 192.168.122.26
```

zone (named)

gonzo

```
$ORIGIN 120.168.192.in-addr.arpa.
$TTL 1D
@     IN SOA  jim.prod.henson. hostmaster.prod.henson. (
              2013060101  ; serial
              28800       ; refresh (8 hours)
              14400       ; retry (4 hours)
              2419200     ; expire (4 weeks)
              86400       ; minimum (1 day)
              )
; define the authoritative name server
              NS      jim.prod.henson.
; static
2              PTR     jim.prod.henson.
; exported resources below this point
```

```
45 PTR gonzo.prod.henson
```

dns/summary

- making a new nameserver is as simple as assigning a role to a node
  - installs packages
  - configures firewall
  - updates resolv.conf for every node in that zone
- each new node that gets added to puppet adds itself to the appropriate zone file
  - any changes cause a reload

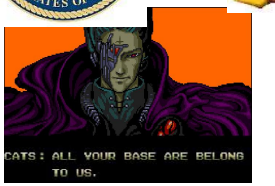
big idea

- nothing is hard coded
- puppet defines the infrastructure

Questions

thank you

Thanks


CATS: ALL YOUR BASE ARE BELONG TO US.


SPRINGDALE LINUX


Cat Herder


THE
BRANDED
ONES


COSTCO
LINUX
TEAM
777

references/resources

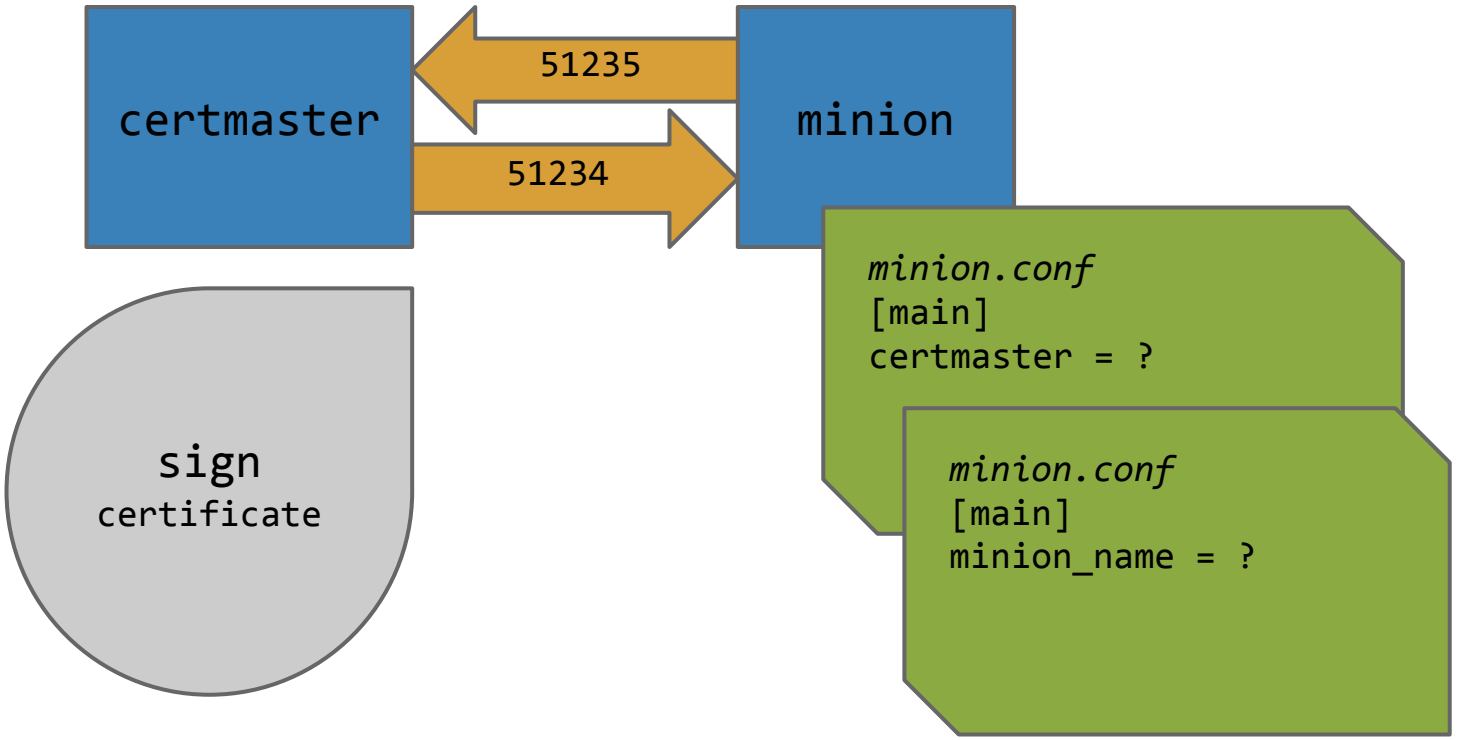
bonus

func

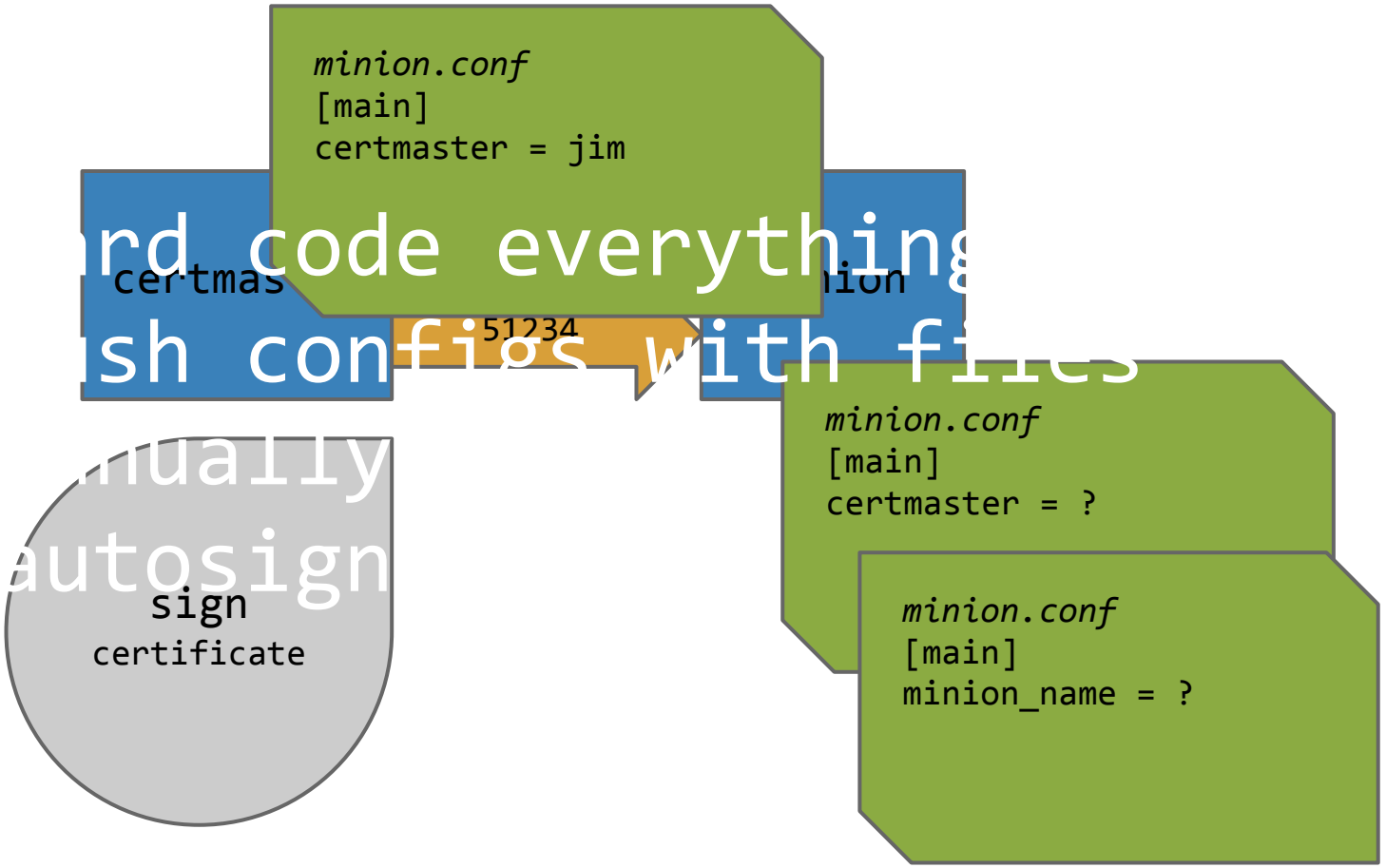
exported firewall and exec rules

func

func


certmaster
51235
51234
minion
minion.conf
[main]
certmaster = ?
minion.conf
[main]
minion_name = ?
sign
certificate

func


minion.conf
[main]
certmaster = jim
rd code everything
certmas
ion
51234
sh configs with files
nually
autosign
sign
certificate
minion.conf
[main]
certmaster = ?
minion.conf
[main]
minion_name = ?

func

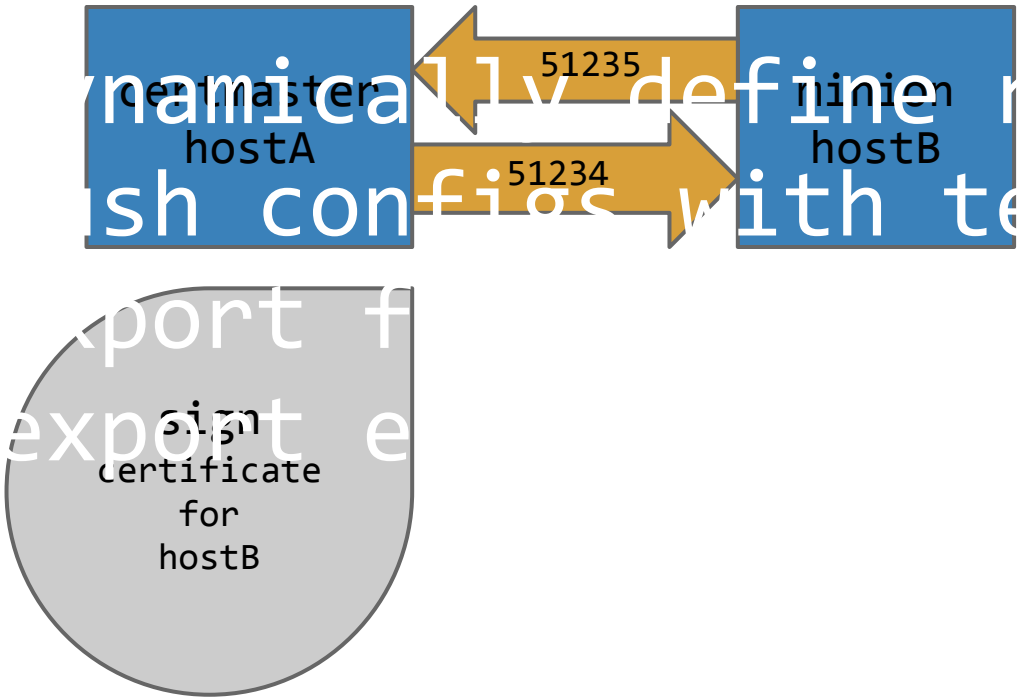
```
class func1::minion {
  package {'func': }
  service {'funcd':
    require => Package['func']'
  }
  file { '/etc/certmaster/minion.conf':
    source => 'puppet:///func1/etc/certmas
    mode   => 644,
    owner  => 0,
    group  => 0
  }
  firewall {'51234 ACCEPT func from jim':
    action => 'accept',
    source => '192.168.122.24',
    dport  => 51234
  }
}
```

ipaddress of jim

/etc/certmaster/minion.conf

```
# configuration for minions

[main]
certmaster = jim
certmaster_port = 51235
log_level = DEBUG
cert_dir = /etc/pki/certmaster
```

func (exported resources)

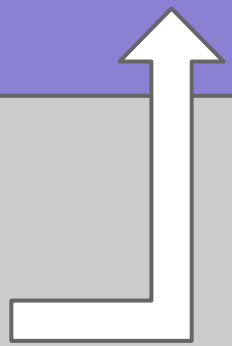


func (exported resources)

```
class func2::master {
  @@firewall { "51234 ACCEPT func from funcmaster
    action => 'accept',
    source => "$::ipaddress",
    dport  => '51234',
    tag    => 'func_master',
  }
  Firewall <<| tag == 'certmaster_minion' |>>
  # export the config file with our name to the minions
  @@file {'/etc/certmaster/minion.conf':
    mode    => 644,
    content => template('func2/etc/certmaster/minion.conf.erb'),
    tag     => 'func_master'
  }
  package { 'certmaster': }
  service { 'certmaster': require => Package['certmaster'] }
}
```

/etc/certmaster/minion.conf.erb

```
# configuration for minions

[main]
certmaster = <%= @fqdn %>
certmaster_port = 51235
log_level = DEBUG
cert_dir = /etc/pki/certmaster
```



func (exported resources)

```
class func2::minion {
  #allow connections from the func master
  Firewall <<| tag == 'func_master' |>>

  #allow connections from us to the
  @@firewall {"51235 ACCEPT certmas
    action => 'accept',
    source => "$::ipaddress",
    dport  => '51235',
    tag    => 'certmaster_minion',
  }
  package {'func': }
  service {'funcd': require => Packag

  File <<| tag == func_master |>>
}
```

/etc/certmaster/minion.conf.erb

```
# configuration for minions

[main]
certmaster = piggy
certmaster_port = 51235
log_level = DEBUG
cert_dir = /etc/pki/certmaster
```

func (exported resources)

firewall + exported resources

- cluster
  allow all traffic from cluster members
- amanda
  allow backup servers move/change
- nagios
  only allow your nagios to connect

func (exported resources)

```
class func2::minion {
...
@@exec {"sign certificate for $::fqdn":
    command => "certmaster-ca --sign $::fqdn",
    path    => '/usr/bin:/bin',
    creates => "/var/lib/certmaster/certmaster/certs/${::fqdn}.cert",
    tag     => 'certmaster_sign_minion',
  }
...
}
```

```
class func2::master {

...

# sign minion keys

  Exec <<| tag == 'certmaster_sign_minion' |>>

...
}
```