



Backup provisioning with bacula and puppet

Dave Simons
OSBConf 2014-09

Dave Simons

- Linux and Open Source Consultant @ inuits.eu
- @miouhpi
- github.com/simonsd

Inuits.eu

- Open source consultancy company
- System administration
- Web development
- Some embedded development
- Offices in Belgium, Netherlands, Ukraine and Czech

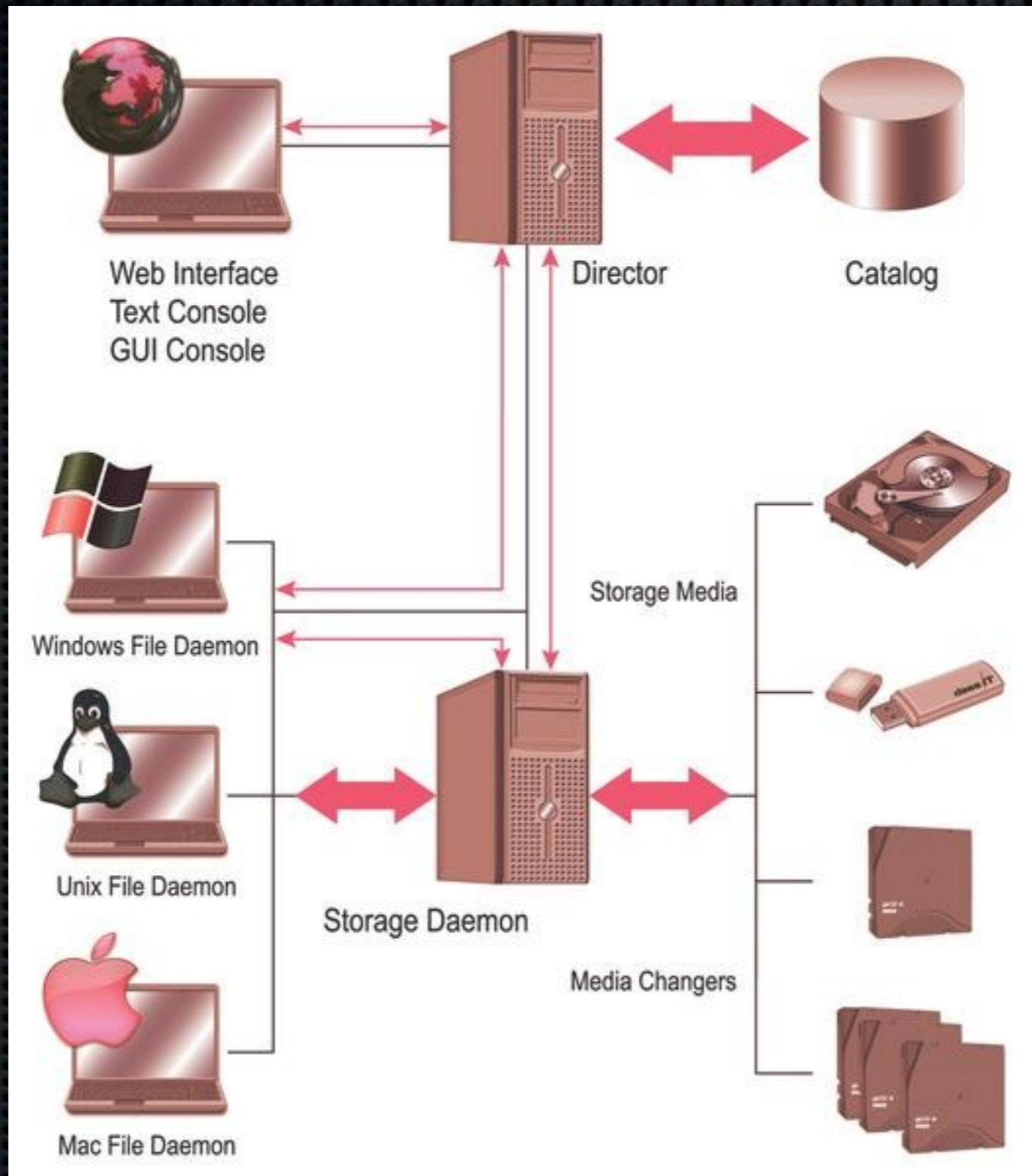
Bacula

- Networked backup system
- Written in C++
- Runs on UNIX, Mac and Windows
- Open source

components

- Director
- Storage daemon
- File daemon
- Catalog (database)
- Storage backend
- Firewall
- Interface

overview



Classic setup

- Install/configure director
- Install/configure catalog
- Install/configure storage daemon
- Install/configure storage backend
- Install/configure file daemon 1
- Install/configure file daemon 2
- Install/configure file daemon N

What's wrong with that?

Lots of things!

- Time consuming
- Error prone
- Repetitive (pronounced /bohr-ing/)

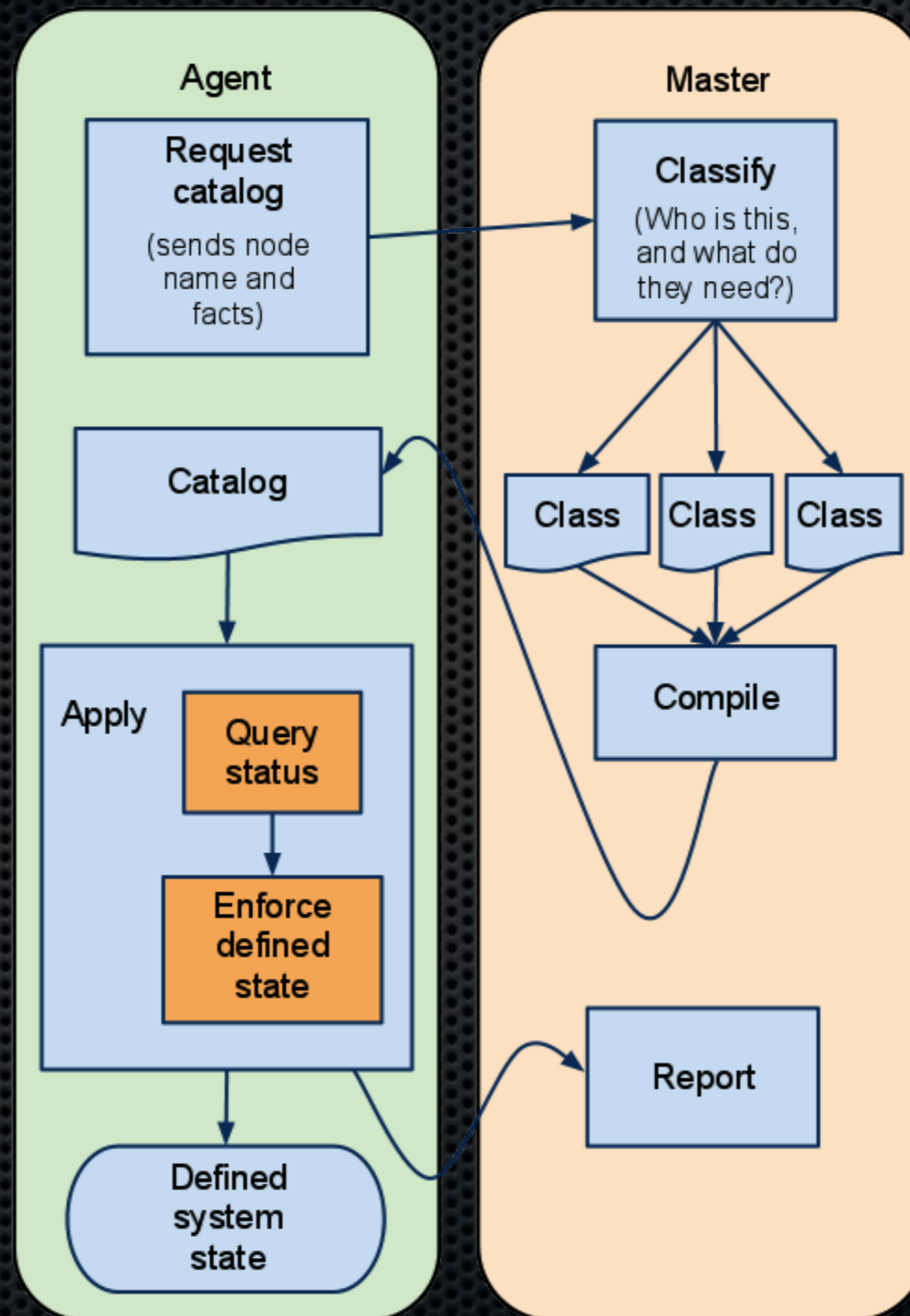
Puppet

- Configuration management tool
- Written in Ruby
- Server-client model
- Standalone
- Open source
- Runs on UNIX, Mac and Windows

components

- Puppet master
- Puppet agent
- Puppet apply
- Hiera
- Puppetdb
- Mcollective
- dashboard/foreman

overview



Puppet apply

- Subcommand
- Compiles catalog
- Executes catalog

hiera

- Datastore
- Abstract code/data
- YAML
- Adjustable hierarchy

```
---  
:backends:  
  - yaml  
  
:logger: console  
  
:hierarchy:  
  - clients/{environment}/{hostname}  
  - roles/{environment}/{role}  
  - roles/common/{role}  
  - hypervisors/{hypervisor}  
  - locations/{datacenter}  
  - environments/{environment}  
  - common  
  
:yaml:  
  :datadir: /etc/puppet/hieradata
```

hieradata

```
[simonsd@shinku][inuits]$ tree -L 2 hiera
```

```
hiera/
```

```
├── common.yaml
├── clients
│   ├── production
│   │   ├── client1.yaml
│   │   ├── client2.yaml
│   │   └── client3.yaml
├── environments
│   └── production.yaml
├── hypervisors
│   ├── hv1.example.com.yaml
│   ├── hv2.example.com.yaml
│   └── hv3.example.com.yaml
├── locations
│   ├── dc1.yaml
│   ├── dc2.yaml
│   └── dc3.yaml
```

Puppet language

- Ruby DSL
- Resources
 - Regular
 - Virtual
 - exported
- Classes
- Manifests
- Modules

resource types

- Cron
- Exec
- File
- Group
- User
- Mount
- Package
- Service

```
file{'/usr/local/bin/mysql-backup':  
  content => template('mysql-backup'),  
  owner   => 'root',  
  group   => 'root',  
  mode    => '0755',  
}
```

```
cron{'mysql-backup':  
  command => '/usr/local/bin/mysql-backup',  
  user     => 'root',  
  hour     => '00',  
  minute   => '15',  
  require  => File['/usr/local/bin/mysql-  
backup',  
}
```

```
user{'bacula':  
  ensure   => 'present',  
  uid      => '712',  
  gid      => 'bacula',  
  password => 'secret',  
  home     => '/home/bacula',  
  shell    => '/bin/sh',  
}
```

Puppet tree

```
[simonsd@shinku][inuits]$ tree -L 2 puppet/
```

```
puppet/
```

```
├── manifests
│   ├── classes
│   ├── defaults
│   ├── hosts
│   └── site.pp
├── modules
│   ├── activemq
│   ├── apache
│   ├── apc
│   ├── apt
│   ├── augeas
│   ├── bacula
│   ├── bash
│   ├── bind
│   ├── collectd
│   ├── concat
│   ├── crond
│   ├── customers
│   ├── dell
│   ├── drupal
│   ├── ejabberd
│   ├── elasticsearch
│   ├── fail2ban
│   ├── ffmpeg
│   ├── filemapper
│   ├── firewall
│   ├── foreman
│   ├── ganbatte
│   ├── gdash
│   ├── gitolite
│   ├── gitorious
│   ├── graphite
│   ├── grok
│   ├── icinga
│   ├── inifile
│   ├── inuits
│   └── ...
```

loading

- Manual imports

```
[simonsd@shinku][puppet]$ cat manifests/site.pp
import 'defaults/*.pp'
import 'classes/*.pp'
import 'hosts/*.pp'
```

- Autoloading magic

```
[simonsd@shinku][puppet]$ puppet apply
--modulepath=/etc/puppet/environments/production/modules site.pp
```

```
[simonsd@shinku][puppet]$ cat puppet.conf
```

```
<snip>
```

```
[master]
```

```
manifest      = $confdir/environments/$environment/manifests/site.pp
```

```
modulepath    = $confdir/environments/$environment/modules
```

```
<snip>
```

modules

```
[simonsd@shinku][puppet]$ tree example-module/  
example-module/  
├── manifests  
│   └── init.pp  
├── templates  
├── files  
├── lib  
├── facts.d  
├── tests  
└── spec
```

basic manifest

```
[simonsd@shinku][puppet]$ cat modules/bacula/manifests/bconsole.pp
```

```
class bacula::bconsole (
  $config_root      = $::bacula::params::config_root,
  $bconsole_pkgname = $::bacula::params::bconsole_pkgname,
  $director_server  = $::bacula::params::director_server,
  $director_port    = $::bacula::params::director_port,
  $director_password = $::bacula::params::director_password,
) inherits ::bacula::params {

  package{"$bconsole_pkgname":
    ensure => 'present',
  }

  file{"${config_root}/bconsole.conf":
    ensure  => 'file',
    owner   => 'root',
    group   => 'root',
    mode    => '0640',
    content => template('bacula/bconsole.conf.erb'),
    require => Package[$bconsole_pkgname],
  }

}
```

basic template

```
[simonsd@shinku][puppet]$ cat modules/bacula/templates/bconsole.conf.erb
### File managed with puppet ###
### Served by:          '<%= scope.lookupvar('::servername') %>'
### Module:            '<%= scope.to_hash['module_name'] %>'
### Template source:  '<%= template_source %>'

#
# List Directors we connect to
#

Director {
  name      = "<%= director_server %>"
  dirport   = "<%= director_port %>"
  address   = "<%= director_server %>"
  password  = "<%= director_password %>"
}
```

Virtual resources

```
@file{'/tmp/foo':  
  content => 'bar',  
  tag     => 'someproject',  
}
```

```
realize File['/tmp/foo']
```

```
File <| tag='someproject' |>
```

exported resources

```
@@nagios_service{'check_bacula_fd-bacula-fd.prod.example.com':  
  check_command      => 'check_nrpe_command!check_bacula_fd',  
  service_description => 'bacula-fd.prod.example.com',  
  host_name          => 'bacula-fd.prod.example.com',  
  target             => "${::icinga::targetdir}/services/bacula-fd.prod.example.com.cfg",  
  notify             => Service[${::icinga::service_server}],  
  tag                => 'bacula',  
}
```

```
Nagios_service <<| name='check_bacula_fd-bacula-fd.prod.example.com' |>>
```

```
Nagios_service <<| tag='bacula' |>>
```


puppet-bacula module tree

```
[simonsd@shinku][puppet]$ tree modules/bacula/
```

```
modules/bacula/  
├── manifests  
│   ├── bconsole.pp  
│   ├── catalog.pp  
│   ├── client.pp  
│   ├── common.pp  
│   ├── default  
│   │   ├── filesets.pp  
│   │   ├── jobdefs.pp  
│   │   ├── pools.pp  
│   │   └── schedules.pp  
│   ├── device.pp  
│   ├── director  
│   │   ├── config.pp  
│   │   ├── install.pp  
│   │   └── service.pp  
│   ├── director.pp  
│   ├── fd  
│   │   ├── config.pp  
│   │   ├── install.pp  
│   │   └── service.pp  
│   ├── fd.pp  
│   ├── fileset.pp  
│   ├── firewall  
│   │   ├── dir.pp  
│   │   ├── fd.pp  
│   │   └── sd.pp  
│   ├── firewall.pp  
│   ├── init.pp  
│   ├── jobdefs.pp  
│   ├── job.pp  
│   ├── messages.pp  
│   ├── params.pp  
│   ├── pool.pp  
│   ├── schedule.pp  
│   ├── sd  
│   │   ├── config.pp  
│   │   ├── install.pp  
│   │   └── service.pp  
│   ├── sd.pp  
│   └── storage.pp  
└── templates  
    ├── bacula-dir.conf.erb  
    └── bacula-fd.conf.erb
```

configure director

```
[simonsd@shinku][puppet]$ cat manifests/classes/bacula-dir.pp
class 'bacula-dir' {

  class {'::bacula::director':
    log_email    => 'logs@example.com',
    from_email => 'bacula@example.com',
  }

  bacula::catalog{"bacula-${::domain}":
    dbname      => hiera('bacula-database'),
    dbuser      => hiera('bacula-database'),
    dbpassword  => hiera('bacula-database'),
    dbhost      => hiera('bacula-database'),
  }

  @@percona::database{"bacula-${::domain}":
    ensure => 'present',
  }

  @@percona::rights{"bacula-${::domain}":
    ensure    => 'present',
    database => hiera('bacula-prod-database'),
    user      => hiera('bacula-prod-database'),
    password => hiera('bacula-prod-database'),
    host      => hiera('bacula-prod-database'),
    priv      => 'all',
  }

  @@nagios_service{"check_bacula_dir-${::fqdn}":
    check_command      => 'check_nrpe_command!check_bacula_dir',
    service_description => "Bacula-dir: ${::fqdn}",
    host_name          => ${::fqdn},
    target              => "${::icinga::targetdir}/services/${::fqdn}.cfg",
    notify              => Service[${::icinga::service_server}],
  }

}
```

configure storage daemon

```
[simonsd@shinku][puppet]$ cat manifests/classes/bacula-sd.pp
class 'bacula-sd' {

  class {'::bacula::sd':
    storage_dir => hiera('bacula-storage-dir', '/srv/backup/bacula'),
  }

  @@nagios_service{'check_bacula_sd-${::fqdn}':
    check_command      => 'check_nrpe_command!check_bacula_sd',
    service_description => "Bacula-sd: ${::fqdn}",
    host_name          => ${::fqdn},
    target              => "${::icinga::targetdir}/services/${::fqdn}.cfg",
    notify             => Service[${::icinga::service_server}],
  }

}
```

configure file daemon

```
[simonsd@shinku][puppet]$ cat manifests/classes/bacula-fd.pp
class 'bacula-fd' {
```

```
  class {'::bacula::fd':
    storage_server => hiera('bacula_default_storage_server', "bacula-sd.${::domain}"),
    director_server => hiera('bacula_director_server',      'bacula'),
    storage_dir     => hiera('bacula_storage_dir',          '/srv/backup/bacula'),
    catalog         => hiera('bacula_catalog',              "bacula-${::domain}"),
  }

  @@bacula::job{${::fqdn}:
    client           => hiera('bacula_hostname',           ${::fqdn}),
    jobdefs          => hiera('bacula_jobdef',              'DefaultJob'),
    fileset          => hiera('bacula_fileset',             false),
    messages         => hiera('bacula_messages',          'bacula-fd'),
    storage          => hiera('bacula_storage',             "${::fqdn}-storage"),
    full_pool        => hiera('bacula_fullpool',            "${::fqdn}-Full"),
    incremental_pool => hiera('bacula_incpool',             "${::fqdn}-Incremental"),
    bschedule        => hiera('bacula_schedule',           'default'),
    client_run_before_job => hiera('bacula_client_run_before_job', false),
    client_run_after_job  => hiera('bacula_client_run_after_job',  false),
  }

  @@bacula::pool{"${::fqdn}-Full":}
  @@bacula::pool{"${::fqdn}-Incremental":
    maximum_volumes => '7',
  }

  @@nagios_service{'check_bacula_fd-${::fqdn}':
    check_command      => 'check_nrpe_command!check_bacula_fd',
    service_description => "Bacula-fd: ${::fqdn}",
    host_name          => ${::fqdn},
    target             => "${::icinga::targetdir}/services/${::fqdn}.cfg",
    notify             => Service[${::icinga::service_server}],
  }
}
```

Other options

- Chef
- Ansible
- Salt
- Cfengine
- Bcfg2
- Bareos
- Rsync
- Rsnapshot
- Rdiff
- Backup ninja

Questions?