From Quattor to Agile Infrastructure Deployment

Mike Kenyon
IT/SDC

26/09/2013

Agile Infrastructure Overview

- Foreman: Virtual machine provisioning up until…
- Puppet: configuration of machine
  - Facter: machine-specific data store
  - Hiera: “site-wide” configuration parameters
- (Much) more info

Puppet in a nutshell

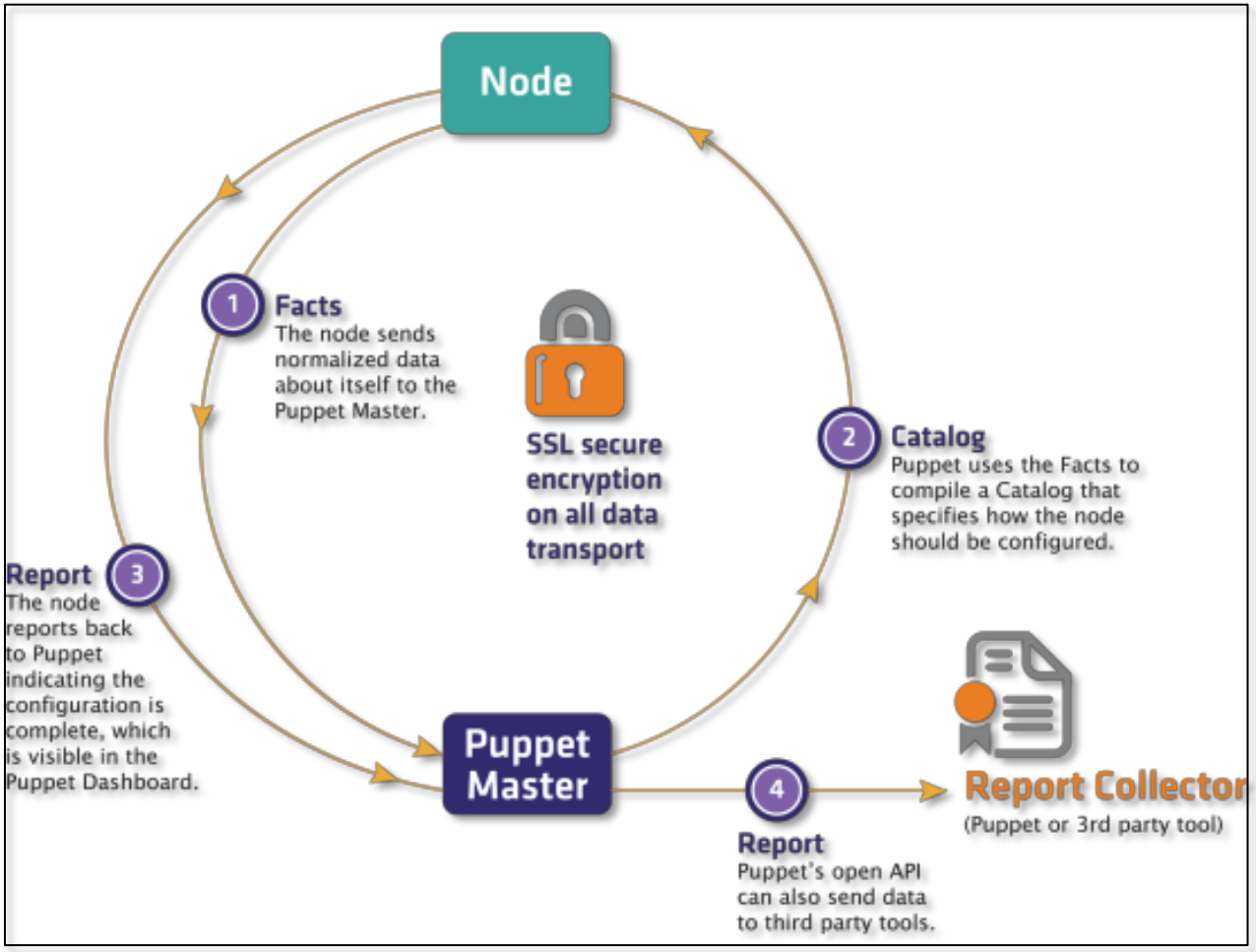
- Configuration management tool
- Declarative: Define **what** is to be configured, not **how** it is to be configured, e.g:
  - yum install <package> 
  - “ensure <package> is installed” 
- Configuration declared in modules (directory) containing manifest files (Ruby)

Puppet in a nutshell: Facter

- A node information store
  - Populated by (extensible) fact plugins
  - Facts available to Puppet manifests, e.g. $::hostname

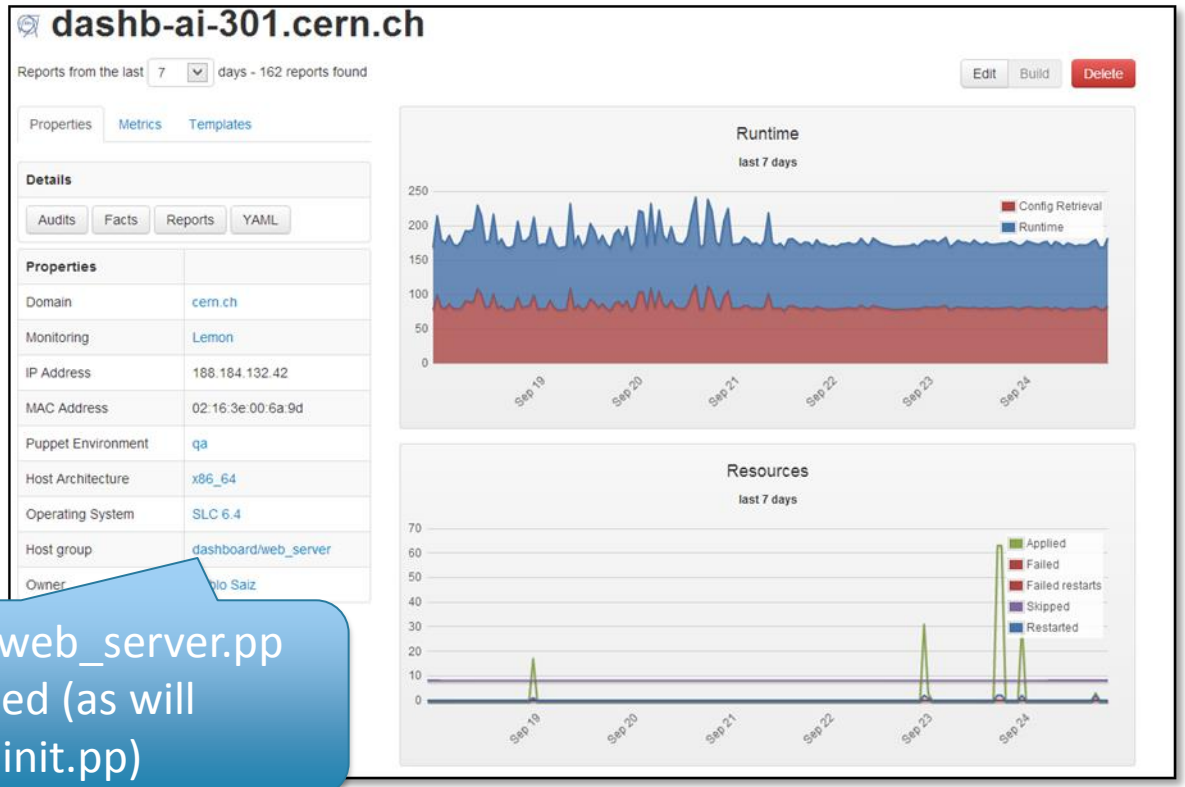
```
[root@dashb-ai-506 ~]# facter --puppet
facterversion => 1.7.2
filesystems => ext4,iso9660
fqdn => dashb-ai-506.cern.ch
hardwareisa => x86_64
hardwaremodel => x86_64
hostgroup => dashboard/nagios
hostgroup_0 => dashboard
hostgroup_1 => nagios
hostname => dashb-ai-506
id => root
interfaces => eth0,lo
ip6tables_version => 1.4.7
ipaddress => 128.142.202.56
is_essential => false
is_pe => false
is_virtual => true
javakeystore => /usr/lib/jvm/java-1.7.0-oracle-1.7.0.25.x86_64/jre/lib/security/cacerts
javaversion => /usr/lib/jvm/jre-1.7.0-oracle.x86_64/bin/java
k5logins => andreeva,dtuckett,ekaravak,lsargsya,mkenyon,psaiz,rbritoda,tuckett
kernel => Linux
kernelmajversion => 2.6
kernelrelease => 2.6.32-358.11.1.el6.x86_64
```

Puppet in a nutshell: Execution summary



Real-world examples: Web-server class

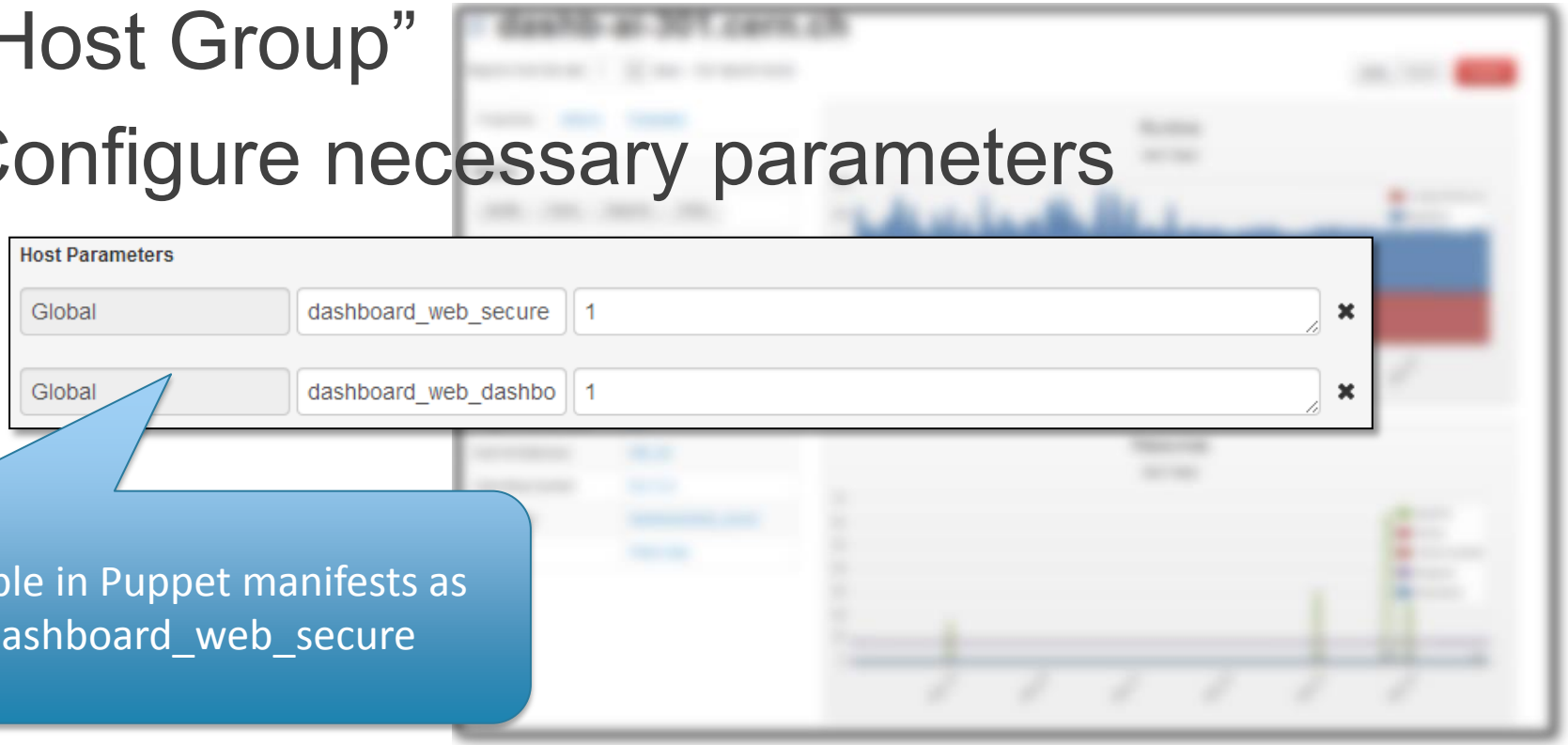
- Build a machine (another talk), assign a "Host Group"



Hence dashboard::web_server.pp class will be used (as will dashboard::init.pp)

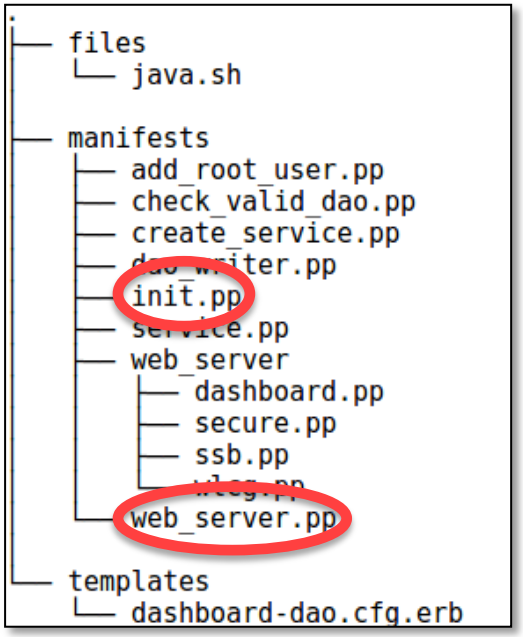
Real-world examples: Web-server class

- Build a machine (another talk), assign a "Host Group"
- Configure necessary parameters



Real-world examples: Web-server class

- Build a machine (another talk), assign a "Host Group"
- Configure necessary parameters
- Write a module…



Real-world examples: Web-server class

manifests/init.pp

```
class hg_dashboard {
  osrepos::ai121yumrepo{'dashboard-stable':
    descr    => 'ARDA Dashboard Stable Packages',
    baseurl  => "http://dashb-slc${major}-build.cern.ch/apt/RPMS.stable",}
...
  group{'cg':
    ensure => present,
    name   => 'cg',
    gid    => 2688,}
...
  package {'gparted': ensure=>'installed'}
...
  user { 'dboard':
    ensure      => present,
    uid         => 1711,}
...
```

Real-world examples: Web-server class

- Build
- Cont
- Write

manifests/web_server.pp

```
class hg_dashboard::web_server {
  include firewall
...
  firewall { '042 allow web':
    proto  => 'tcp',
    dport  => '80',
    action => 'accept',
  }
...
  exec { '/usr/sbin/setenforce Permissive':
     onlyif => '/usr/bin/test $(/usr/sbin/getenforce) != Permissive
  }

  if ($::dashboard_web_dashboard ) {
    if ($::dashboard_web_dashboard !~ /\#/) {
       include hg_dashboard::web_server::dashboard
    }
  }
...
  if ($::dashboard_web_secure ) {
       include hg_dashboard::web_server::secure}
```

Real-world examples: Web-server class

- Build
- Conf
- Write

manifests/web_server.pp

```
class hg_dashboard::web_server {
  include firewall
...
  firewall { '042 allow web':
    proto  => 'tcp',
    dport  => '80',
    action => 'accept',
  }
...
  exec { '/usr/sbin/setenforce Permissive':
     onlyif => '/usr/bin/test $(/usr/sbin/getenforce) != Permissive
  }

  if ($::dashboard_web_dashboard ) {
    if ($::dashboard_web_dashboard !~ /\#/) {
       include hg_dashboard::web_server::dashboard
    }
  }
...
  if ($::dashboard_web_secure ) {
       include hg_dashboard::web_server
```

```
class hg_dashboard::web_server::dashboard {
  file {'/etc/httpd/conf.d/dashboard.conf':
        ensure  => present,
        before  => File['httpd_vdir'],
        require => Package['dashboard-web']}
  package {'dashboard-web':ensure=>installed}
```

Real-world examples: Web-server class

- Build
- C
- V

```
file {'/etc/httpd/conf.d/httpd.conf':
     ensure  => present,
     source  => 'puppet:///modules/hg_dashboard/httpd.conf',
     require => Package['httpd'],
     notify  => Service['httpd']}
```

Puppet in a nutshell (revisited): Hiera

- Hierarchical Key:Value store for Puppet
- Keeps machine-specific data out of manifests
- Acts like a site-wide configuration file
- Can be encrypted for sensitive data
- Can also override some default settings:

```
osrepos_epel_priority: 10
apache::user: dboard
apache::group: cg
```

Using Hiera parameters

manifests/database.pp

```
class hg_dashboard::database {

  $vos = $::dashboard_dao
  $volist = split($vos, ",")
  package {"dashboard-dao":}
  if ($vos) {
    $dao = hiera("dashboard_dao_encrypted", {} )
     hg_dashboard::check_valid_dao {$volist:
         dao => $dao,
         before => Dao_writer["dao_writer"],
         }

     hg_dashboard::dao_writer {"dao_writer":
         dao_file => '/opt/dashboard/etc/dashboard-dao/dashboard-dao.cfg',
         volist => $vos,
         dao => $dao,
         }

     } else {
        fail("The 'dashboard-dao' parameter doesn't exist. Add it via Foreman.")
    }

  }
```


```
---
dashboard_dao_encrypted:
        atlas_vo_info:
           section_title: vo-info
           vo_name: ATLAS
           logo_filename: atlaslrg.png
```

Using Hiera parameters

manifests/dao_writer.pp

```
define hg_dashboard::dao_writer($dao_file='/opt/dashboard/etc/dashboard-....'
...
   if ! defined(Concat["${dao_file}"]) {
      concat{"${dao_file}":
         owner =>  $owner,
         group =>  $group,
         mode  =>  $mode}
...
      concat::fragment{"dao_file${dao_file}_header":
         target  => "${dao_file}",
         order   => '00',
         content => "#DAO file generated from hg_dashboard::dao_writer class."
...
     concat::fragment{"${vo}":
       target  => $dao_file,
       order   => '05',
       content => template('hg_dashboard/dashboard-dao.cfg.erb')}
```

Using Hiera parameters

Hiera data

```
---
dashboard_dao_encrypted:
        atlas_vo_info:
            section_title: vo-info
            vo_name: ATLAS
            logo_filename: atlaslrg.png
```

Dao.erb

```
##### ORACLE SPECIFIC CONFIGURATION

<% @vo.each do |vo_key| -%>
[<%= @dao[vo_key]['section_title'] -%>]
<% @dao[vo_key].sort_by {|key, value| key}.each do | subkey | -%>
<% if subkey[0] != 'section_title' -%>
<%= subkey[0]-%> = <%= subkey[1]%>
<% end -%>
<% end -%>

<% end -%>
```

[output]

```
#DAO file generated from hg_dashb[illegible]d::dao_writer class.
...
[vo-info]
logo_filename = atlaslrg.png
vo_name = ATLAS
```

Configuring Nagios with Puppet: Work in Progress

- Inputs:
  - VO-Feeds
  - POEM (nagios metric profiles)
  - Static files

```
▼<atp_site name="UKI-SCOTGRID-GLASGOW">
  <service hostname="svr018.gla.scotgrid.ac.uk" flavour="SRMv2"/>
  <service hostname="svr026.gla.scotgrid.ac.uk" flavour="CREAM-CE"/>
  <group name="Tier-3" type="CMS_Tier"/>
  <group name="T3_UK_ScotGrid_GLA" type="CMS_Site"/>
  <group name="T3_UK_ScotGrid_GLA" type="AllGroups"/>
  <group name="T3_UK_ScotGrid_GLA" type="Tier3s"/>
  <group name="T3_UK_ScotGrid_GLA" type="Tier3s+Tier2s"/>
  <group name="T3_UK_ScotGrid_GLA" type="Tier3s+Tier2s + Tier1s"/>
</atp_site>
```

```
{
    "atp_service_type_flavour": "OSG-SRMv2",
    "fqan": "/cms/Role=production",
    "metric": "org.cms.SRM-GetPFNFromTFC",
    "vo": "cms"
},
{
    "atp_service_type_flavour": "OSG-SRMv2",
    "fqan": "/cms/Role=production",
    "metric": "org.cms.SRM-VODel",
    "vo": "cms"
},
```

```
"org.cms.SRM-VODel" : {
   "docurl" : "https://twiki.cern.ch/twiki/bin/vi
   "parent" : "org.cms.SRM-AllCMS",
   "flags" : {
      "OBSESS" : 1,
      "VO" : 1,
      "PASSIVE" : 1
   },
   "metricset" : "org.cms.SRM"
},
```

Configuring VO-SAM-Nagios with Puppet: Work in Progress

- Puppet will detect if input sources have changed, if so:
  - Trigger Ruby script to generate Nagios configuration
- Significant reduction in code (~250 lines so far)

Finally: Not just for deployed services

- We use Puppet for internal productivity, e.g. development environments:

```
#  Development machine in the dashboard cluster
class hg_dashboard::devel {
  $developer = $::dashboard_developer
  package {'eclipse-pde':}
  package {'pylint':}
  package {'java-1.6.0-openjdk-devel':}
  package {'sqldeveloper':}
  package {'firefox':}
  package {'git':}
  package {'rpm-build':}

  include   hg_dashboard::web_server::secure
  include   hg_dashboard::web_server::dashboard

  if ($developer) {
    $b = split($developer,',')
    hg_dashboard::create_local_user {$b:}
    hg_dashboard::add_root_user {$b:}}

  firewall { '042 allow django web server':
    proto  => 'tcp',
    dport  => '8000',
    action => 'accept',}
}
```

Finally, finally…

- AI approach is **much** more intuitive than Quattor
  - High level of abstraction, resulting in…
  - Machine configuration more akin to software development, rather than system-administration
- We've only just scratched surface of Puppet
  - Explore what it can do for Nagios…