



Mardi du dev

Puppet

Quand les serveurs deviennent des marionnettes

Summary

- What is Puppet ?
- Why Puppet ?
- Resource Abstraction Layer
- Puppet language
- Modules
- Architecture
- Reporting

Summary

- * What is Puppet ?
- Why Puppet ?
- Resource Abstraction Layer
- Puppet language
- Modules
- Architecture
- Reporting

What is Puppet

- **S**oftware **C**onfiguration **M**anagement system
- Written in Ruby
- Free software (Apache 2.0)
- Current version 3.6
- PuppetLabs since 2005
- Cross platform (Linux, Unix, Windows)

Summary

- What is Puppet ?
- * **Why Puppet ?**
- Resource Abstraction Layer
- Puppet language
- Modules
- Architecture
- Reporting

Why Puppet ?

- Simplify the majority of the technical tasks
- The sysadmin work is written as code
- Configuration homogeneity
- Massive deployments/modifications become easy

Why Puppet ?

- Pets vs Cattle
- DevOps
- Continuous deployment/integration
- Other products : cfengine, chef, ansible

Summary

- What is Puppet ?
- Why Puppet ?
- * **Resource Abstraction Layer**
- Puppet language
- Modules
- Architecture
- Reporting

Resource Abstraction Layer

- High level resources :
 - Some **types** : package, service, file, user
 - Providers : fulfillment of resources
 - Package provider : yum, apt, pip, gem

- Available resources types :
 - Puppet built-in
 - 3rd party

Resource Abstraction Layer

puppet resource --type

```
augeas
computer
cron
exec
file
filebucket
group
host
interface
k5login
macauthorization
mailalias
maillist
mcx
mount
nagios_command
nagios_contact
nagios_contactgroup
nagios_hostdependency
nagios_hostescalation
nagios_hostextinfo
nagios_hostgroup
nagios_service
nagios_servicedependency
nagios_serviceescalation
nagios_serviceextinfo
nagios_servicegroup
nagios_timeperiod
notify
package
resources
router
schedule
scheduled_task
selboolean
```

Resource Abstraction Layer

puppet describe -s user

user

====

Manage users. This type is mostly built to manage system users, so it is lacking some features useful for managing normal users.

[..]

Parameters

allowdupe, attribute_membership, attributes, auth_membership, auths, comment, ensure, expiry, forcelocal, gid, groups, home, ia_load_module, iterations, key_membership, keys, managehome, membership, name, [...]

Providers

aix, directoryservice, hpuxuseradd, ldap, pw, user_role_add, useradd, windows_adsi

Resource Abstraction Layer

puppet resource user puppet

```
user { 'puppet':  
  ensure           => 'present',  
  comment          => 'Puppet',  
  gid              => '52',  
  home             => '/var/lib/puppet',  
  password         => '!!',  
  password_max_age => '-1',  
  password_min_age => '-1',  
  shell           => '/sbin/nologin',  
  uid             => '52',  
}
```

Summary

- What is Puppet ?
- Why Puppet ?
- Resource Abstraction Layer
- * **Puppet language**
- Modules
- Architecture
- Reporting

Puppet language

- Declarative, **Domain Specific Language**
 - Describe the desired state of the system by declaring resources
 - Ruby DSL is supported for complex logic
- Programs are called **manifests**
- A manifest is compiled into a **catalog**

Puppet language : concepts

- Nodes
- Classes
- Ordering (Run-stages)
- Variables, conditionnals
- Dependency relationship
- Tags
- Facts

Puppet language : nodes

- A block of code in one node's catalog

```
node 'myserver1' {  
  include classY  
}
```

```
node 'myserver2' {  
  include classX  
}
```


Puppet language : classes

- Block of code to group resources
- Inheritances
- Parameters and variables
- Singleton

Puppet language : define

- Blocks of Puppet code that can be evaluated multiple times with different parameters Inheritances
- Once defined, they act like a new resource type

Puppet language : define

```
define apache::vhost ($port, $docroot, $servername = $title, $vhost_name = '*')
{
  include apache # contains Package['httpd'] and Service['httpd']
  include apache::params # contains common config settings
  $vhost_dir = $apache::params::vhost_dir
  file { "${vhost_dir}/${servername}.conf":
    content => template('apache/vhost-default.conf.erb'),
    # This template can access all of the parameters and variables from above.
    owner   => 'www',
    group   => 'www',
    mode    => '644',
    require => Package['httpd'],
    notify  => Service['httpd'],
  }
}
```

Puppet language : variables/conditionnals

- Variable names are prefixed with a \$
- Assignment :

```
$content = "some content\n"
```

- Resolution :

```
file {'/tmp/testing':  
  ensure => file,  
  content => $content, }
```

Puppet language : variables/conditionnals

- Conditionnal statement : if, unless, case, selector

```
case $operatingsystem {  
  centos, redhat: { $service_name = 'ntpd' }  
  debian, ubuntu: { $service_name = 'ntp' }  
}
```

Puppet language : facts

- System information, available as « global variables » in manifests

```
[machine ~]$facter
architecture => x86_64
augeasversion => 0.9.0
facter version => 1.6.17
hardwareisa => x86_64
hardwaremodel => x86_64
is_virtual => true
kernel => Linux
kernelmajversion => 2.6
kernelrelease => 2.6.32-
279.19.1.el6.x86_64
kernelversion => 2.6.32
lsbdistcodename => Final
lsbdistdescription => CentOS
release 6.3 (Final)
lsbdistid => CentOS
lsbdistrelease => 6.3
```

Summary

- What is Puppet ?
- Why Puppet ?
- Resource Abstraction Layer
- Puppet language
- * **Modules**
- Architecture
- Reporting

Puppet language : modules

- Self-contained bundles of code and data
- **Nearly all** puppet manifests belong in module

<MODULE NAME>

- manifests
- files
- templates
- lib
- facts.d
- tests
- spec

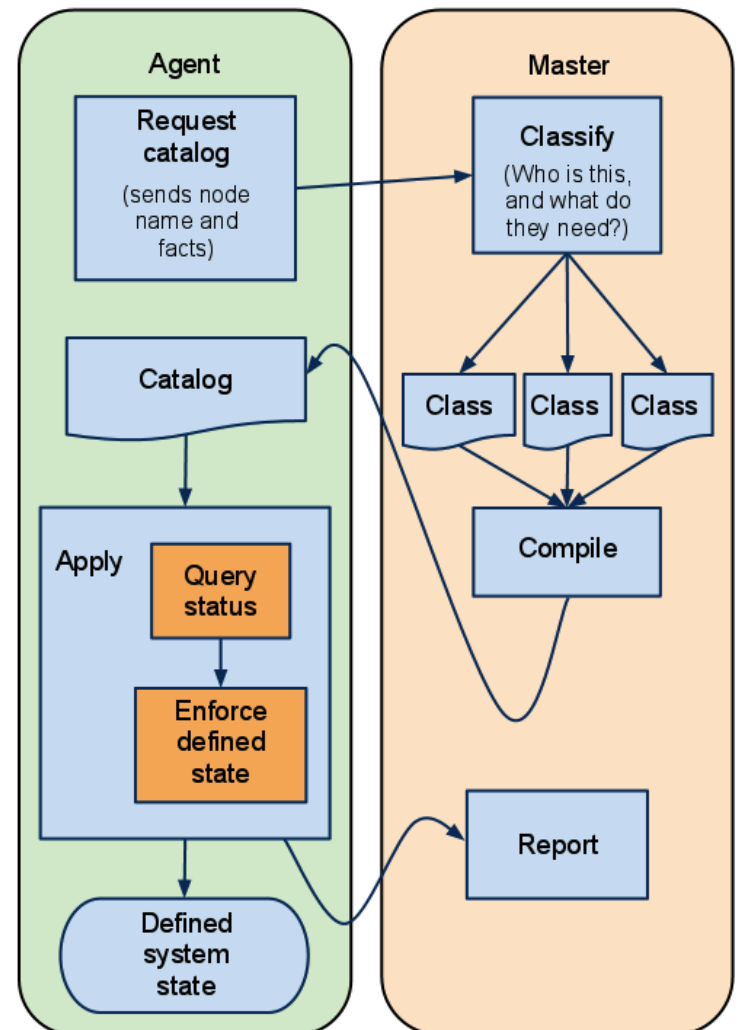
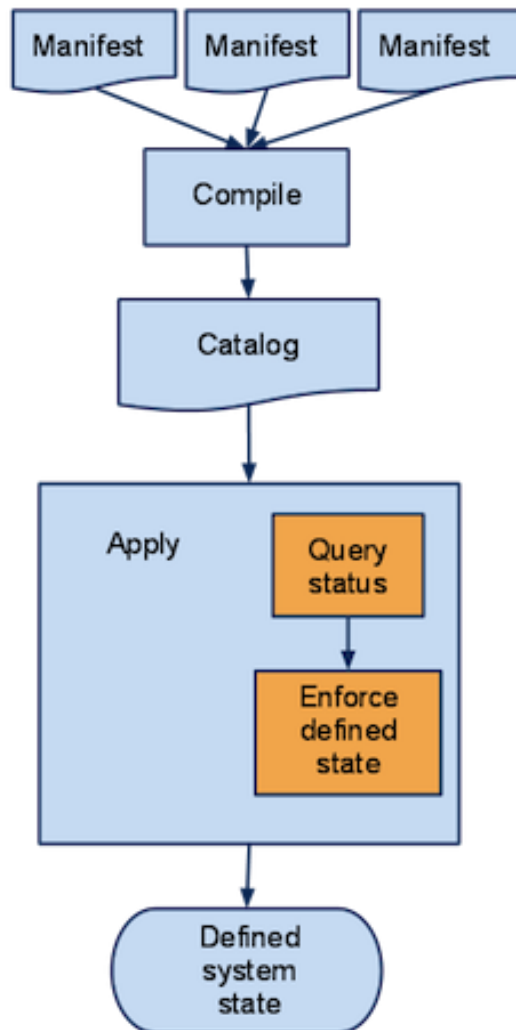
Summary

- What is Puppet ?
- Why Puppet ?
- Resource Abstraction Layer
- Puppet language
- Modules
- * **Architecture**
- Reporting

Architecture

- Pull-based agent/master mode
- REST API
- Master stores manifests
- Agent requests its catalog from the master
- Client-Master encryption (integrated PKI)

Standalone VS Agent-Server



Summary

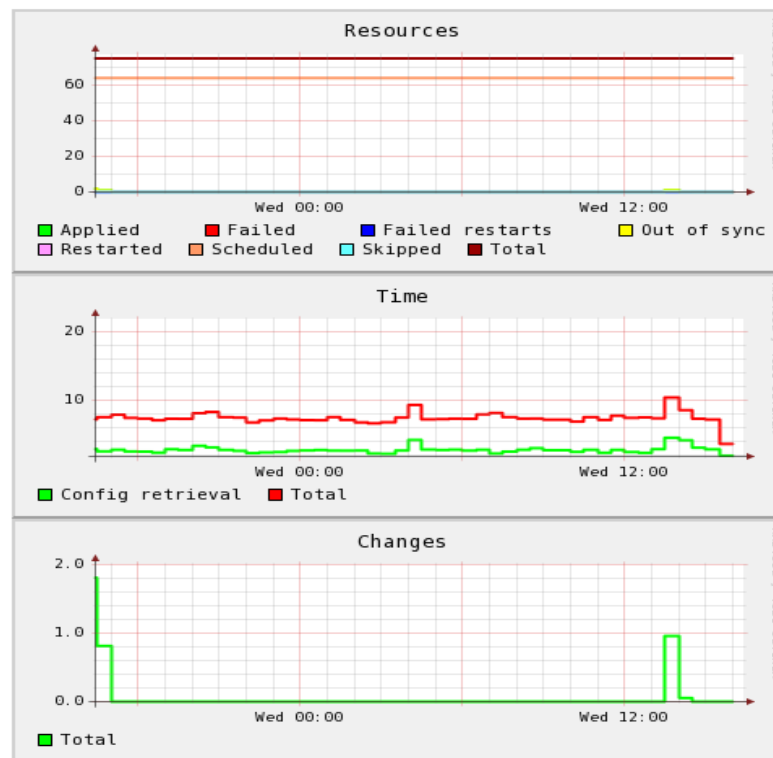
- What is Puppet ?
- Why Puppet ?
- Resource Abstraction Layer
- Puppet language
- Modules
- Architecture
- * **Reporting**

Reporting

- Agent can be configured to send reports at the end of every configuration run :
 - Logs
 - Resources change, exec time
- Local logs
- Puppet Dashboard for graphic reports

Reporting : nagios/RRD integration

Puppetd Stats	OK
Puppetd lag	OK
Puppetd process status	OK



Questions ?