Puppet Configuration Management Tool

DevOps Training

CONFIGURATION MANAGEMENT

Managing large infrastructures by hand clearly won't scale.

- A configuration management system lets us establish a single, authoritative source for system configurations
- Abstracted from the OS
- Changes can be deployed to 1000 machines just as easily as 1. Policy compliance is enforced.

INFRASTRUCTURE AS CODE

Resource Abstraction Language (RAL):

- Define particular system resources (eg: files, packages, users) independently of the underlying OS
- Describe the desired state of the system in your RAL.
- It's up to the config management system to bring everything into line.

INFRASTRUCTURE AS CODE

Resource Abstraction Language (RAL):

- Define particular system resources (eg: files, packages, users) independently of the underlying OS
- Describe the desired state of the system in your RAL.
- It's up to the config management system to bring everything into line.

OPTIONS

CFEngine : RAL is based on C fast and light weight

Chef: RAL written in Ruby and used in deployments

Ansible: – Script-based approach with “Playbooks”.

Salt : Agent (minion) runs on each host .”configuration directives”

PUPPET

Why Puppet? –

Quite a high level RAL in the form of manifests.

Very active community. (more than 7000 open modules available)

Ability to define environments.

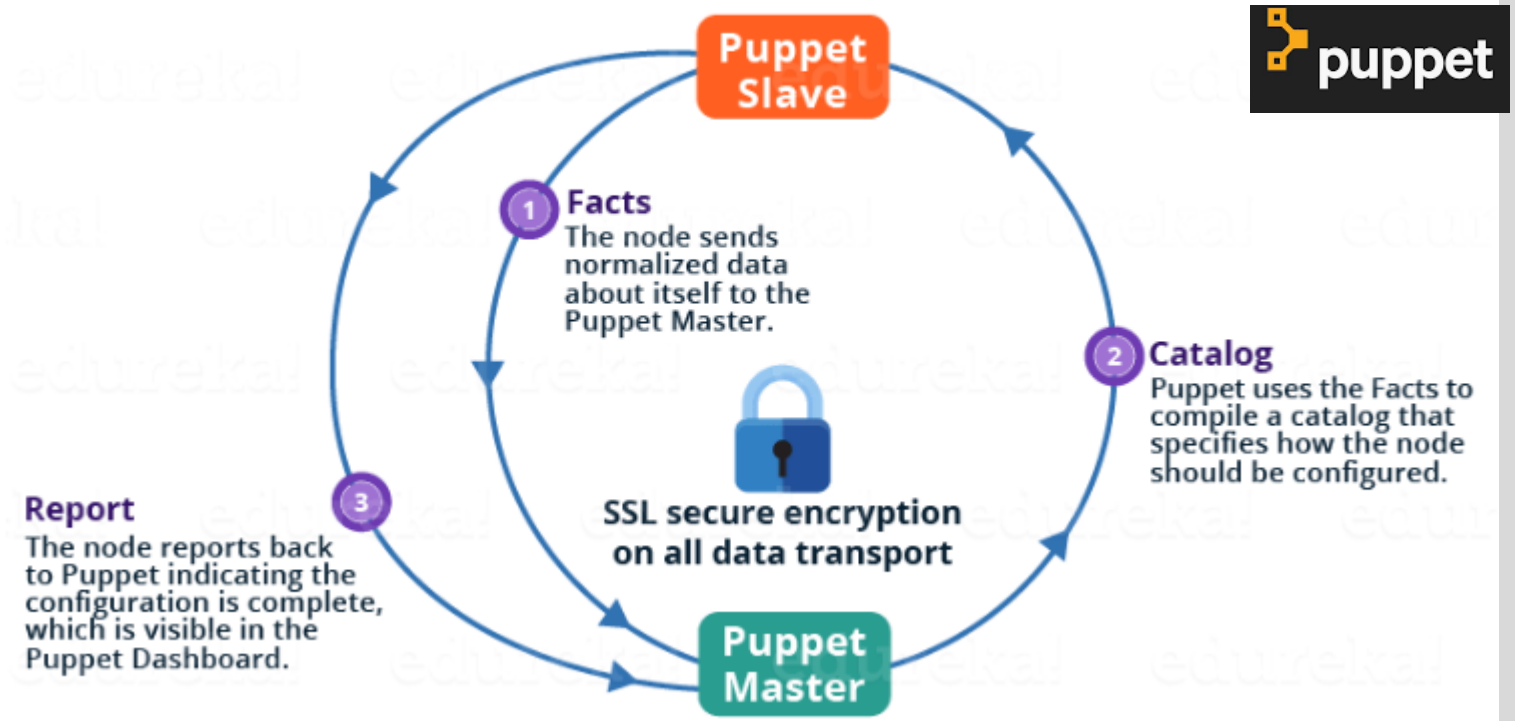
Strong dependency management.

Large surrounding toolset.

Easy language [Puppet DSL : Syntax] & Open Source

PUPPET ARCHITECTURE

Puppet uses a Master-Slave architecture. The diagram below depicts the same:



PUPPET MASTER/SLAVE COMMUNICATION

PUPPET MAIN COMPONENTS

Puppetmaster: –

- Distributes manifests to agents on demand.

- Environments give the ability to serve out different versions of code to your agents.

Puppet agent: –

- Implements manifests (locally, or from puppetmaster).

Facter: –

- Gathers metadata about the host, which can be used in manifests.

PUPPET MAIN COMPONENTS (CONT)

Puppetdb: - Collects data generated by puppet agents, for use in other puppet manifests.

Hiera: – Define hierarchical data for use in puppet manifests.

Puppet ecosystem: –

Foreman – Stats, audits, server lifecycle control. –

MCollective – Parallel job execution framework.

Blueprint – Generate manifests from existing system.

Augeas – Edit legacy config files

PUPPET MANIFEST EXAMPLE

```
group { 'heanet':
    ensure => present,
}

user { 'heanet':
    ensure      => present,
    home        => '/home/heanet',
    gid         => 'heanet',
    comment     => 'HEAnet role account',
    shell       => '/bin/bash',
    password    => '*',
    require     => Group[heanet],
}

file { '/home/heanet/':
    ensure  => directory,
    owner   => heanet,
    group   => heanet,
    mode    => '0755',
    require => User[heanet],
}
```

PUPPET MODULES

Puppet manifests, files and templates can be packaged into a module.

- Modules are intended to be self-contained and reusable: – Ntp server module. – Apache server module.
- The puppet community publishes an extensive collection of modules on the puppet forge: – http://forge.puppetlabs.com

PUPPET GOOD PRACTICES

Infrastructure is now code, so.

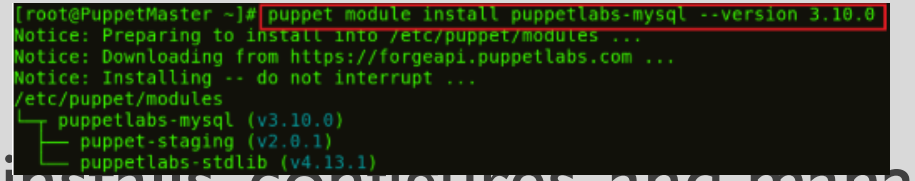
- Put your manifests under version control.
- Never allow changes to be made on the live puppetmaster. –
- Automate/code deployment procedures.

PUPPET HANDSON 1

Step 1: In Puppet Master install MySQL and PHP modules.

COMMAND :

puppet module install puppetlabs-mysql –version 3.10.0

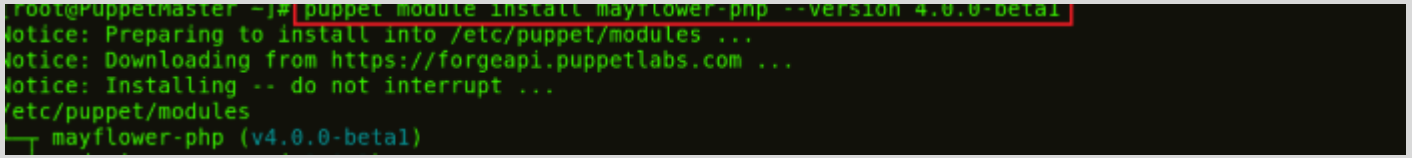


This MySQL module installs, configures, and manages the MySQL service. This module manages both the installation and configuration of MySQL, as well as extending Puppet to allow management of MySQL resources, such as databases, users, and grants.

PUPPET HANDSON 1 (CONT)

2: Command

puppet module install mayflower-php –version 4.0.0-beta1

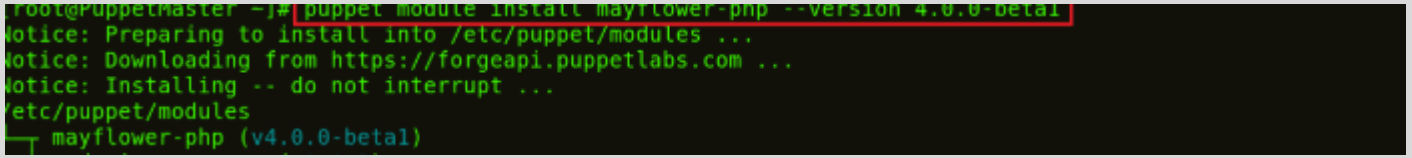


This module is used for managing PHP, in particular php-fpm. PHP-FPM (FastCGI Process Manager) is an alternative PHP FastCGI implementation with some additional features useful for sites of any size, especially busier sites

PUPPET HANDSON 1 (CONT)

2: Command

puppet module install mayflower-php –version 4.0.0-beta1



This module is used for managing PHP, in particular php-fpm. PHP-FPM (FastCGI Process Manager) is an alternative PHP FastCGI implementation with some additional features useful for sites of any size, especially busier sites

PUPPET HANDSON 1(CONT)

Step 2: In Puppet Manifests include MySQL server and PHP.

Execute this: vi /etc/puppet/manifests/site.pp

You can use any other editor as well like vim, gedit etc. In this site.pp file add the following:

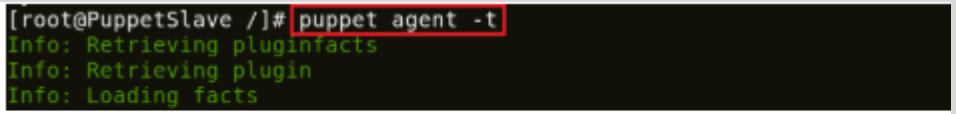
```
1include '::mysql::server'
2 include '::php'
```

Save and quit.

PUPPET HANDSON 1 (CONT)

Step 3: Puppet Slaves pulls its configuration from the Master periodically (after every 30 minutes). It will evaluate the main manifest and apply the module that specifies MySQL and PHP setup. If you want to try it out immediately, you need to run the following command on every Slave node:

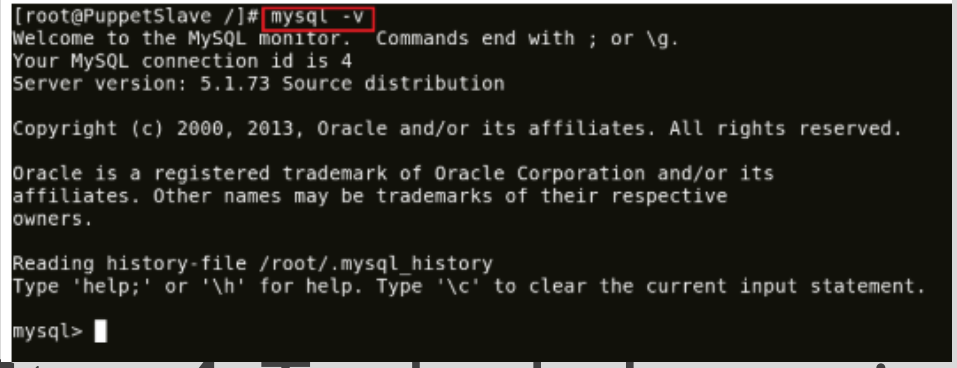
Execute this: puppet agent –t



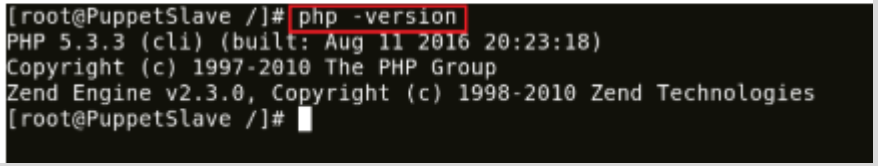
So MySQL and PHP is installed successfully on the Slave node.

PUPPET HANDSON 1(CONT)

1) mysql -v



2) php -version



Step 4: To check the version of MySQL and PHP installed:

THANK YOU!

training@laksans.com