DevOps

Puppet: For Configuration Mgmt.

Trainer: Abhijith V G – AWS, eCommerce, Mobile & DevOps Architect

Magento Certified

DEVELOPER PLUS

amazon web services | Certified

Developer - Associate

Puppet: What and Why of Configuration Management Tools

Problem Scenarios With Growth: Application

Puppet: What and Why of Configuration Management Tools

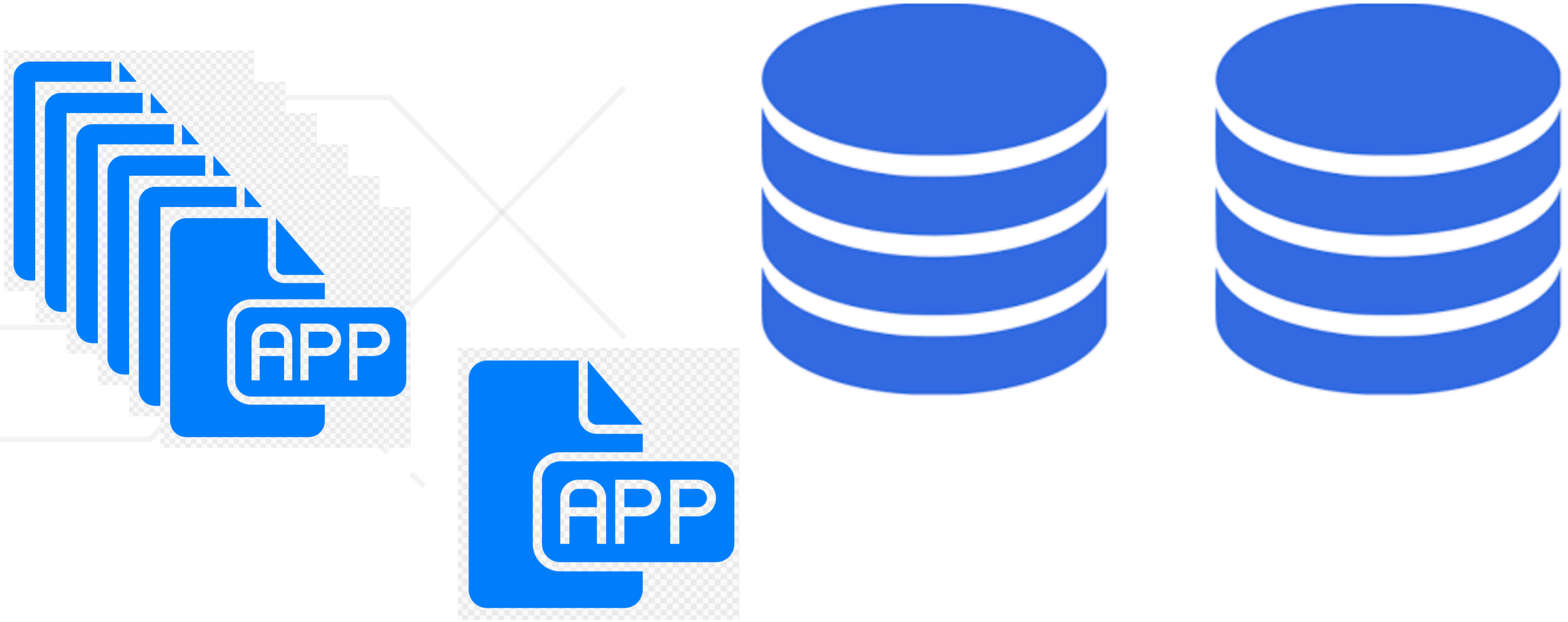
Problem Scenarios With Growth: Application+DB

Puppet: What and Why of Configuration Management Tools

Problem Scenarios With Growth: Application+DB+SlaveDB/ReadReplicas

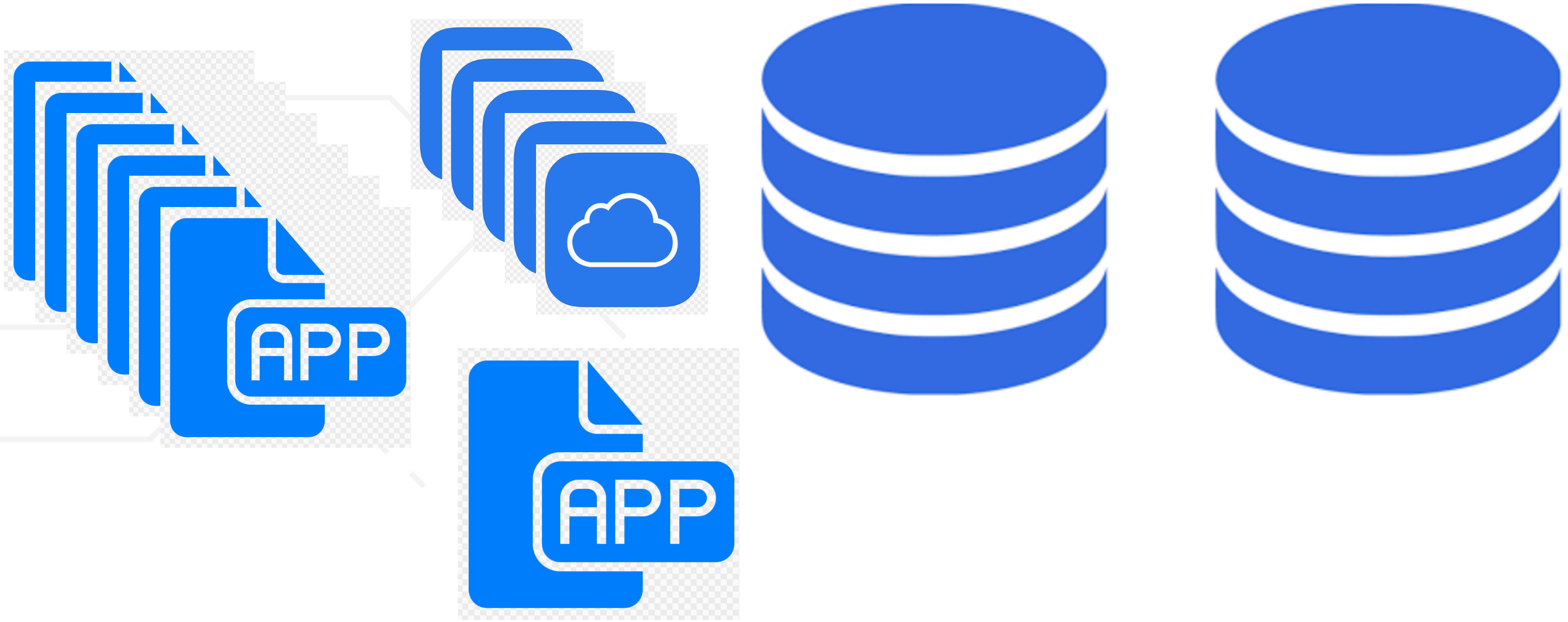
Puppet: What and Why of Configuration Management Tools

Problem Scenarios With Growth: LoadBalancer+Application(with Redundancy) DB+SlaveDB/ReadReplicas



Puppet: What and Why of Configuration Management Tools

Problem Scenarios With Growth: LoadBalancer+Application(with Redundancy) DB+SlaveDB/ReadReplicas+CDN



Puppet: What and Why of Configuration Management Tools

Problem Scenarios With Growth: LoadBalancer+Application(with Redundancy) DB+SlaveDB/ReadReplicas+CDN+CacheDBs

Puppet: What and Why of Configuration Management Tools

Problem Scenarios With Growth: LoadBalancer+Application(with Redundancy) DB+SlaveDB/ReadReplicas+CDN+CacheDBs+Monitoring?

Puppet: What and Why of Configuration Management Tools

Problem Scenarios With Growth: The Infrastructure + Application Mesh (Mess!)

Puppet: SysAdmin Challenges

SysAdmin (DevOps) Challenges

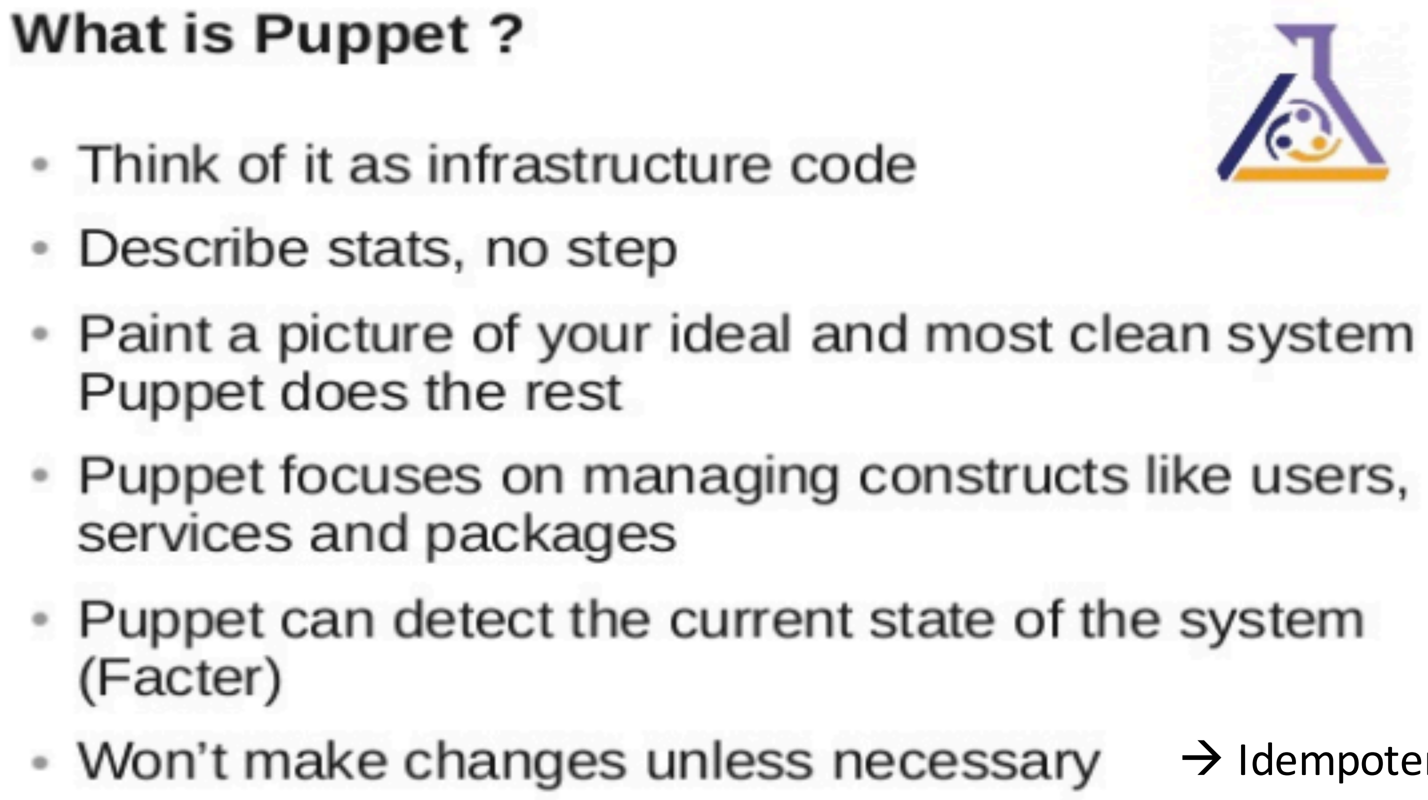
Deploying Servers with Consistent Configurations/Versions
Identifying and Fixing Inconsistencies
Continuous Application Deployment
Automating Everything
Rollback and Disaster Recovery

Puppet: Why Not Use Manual Scripting

Why not use scripts to automate?

1. Difficult to code
   - Expertise/Ordering/Dependencies/Tagging
2. Difficult to document/delegate
3. Must be coded for each OS/Version

Puppet: What is Puppet - How puppet works



What is Puppet ?

- Think of it as infrastructure code
- Describe stats, no step
- Paint a picture of your ideal and most clean system Puppet does the rest
- Puppet focuses on managing constructs like users, services and packages
- Puppet can detect the current state of the system (Facter)
- Won't make changes unless necessary

→ Idempotent

Puppet: What is Puppet – Common Terms

Puppet Labs - The Company behind Puppet

Puppet - The OpenSource version

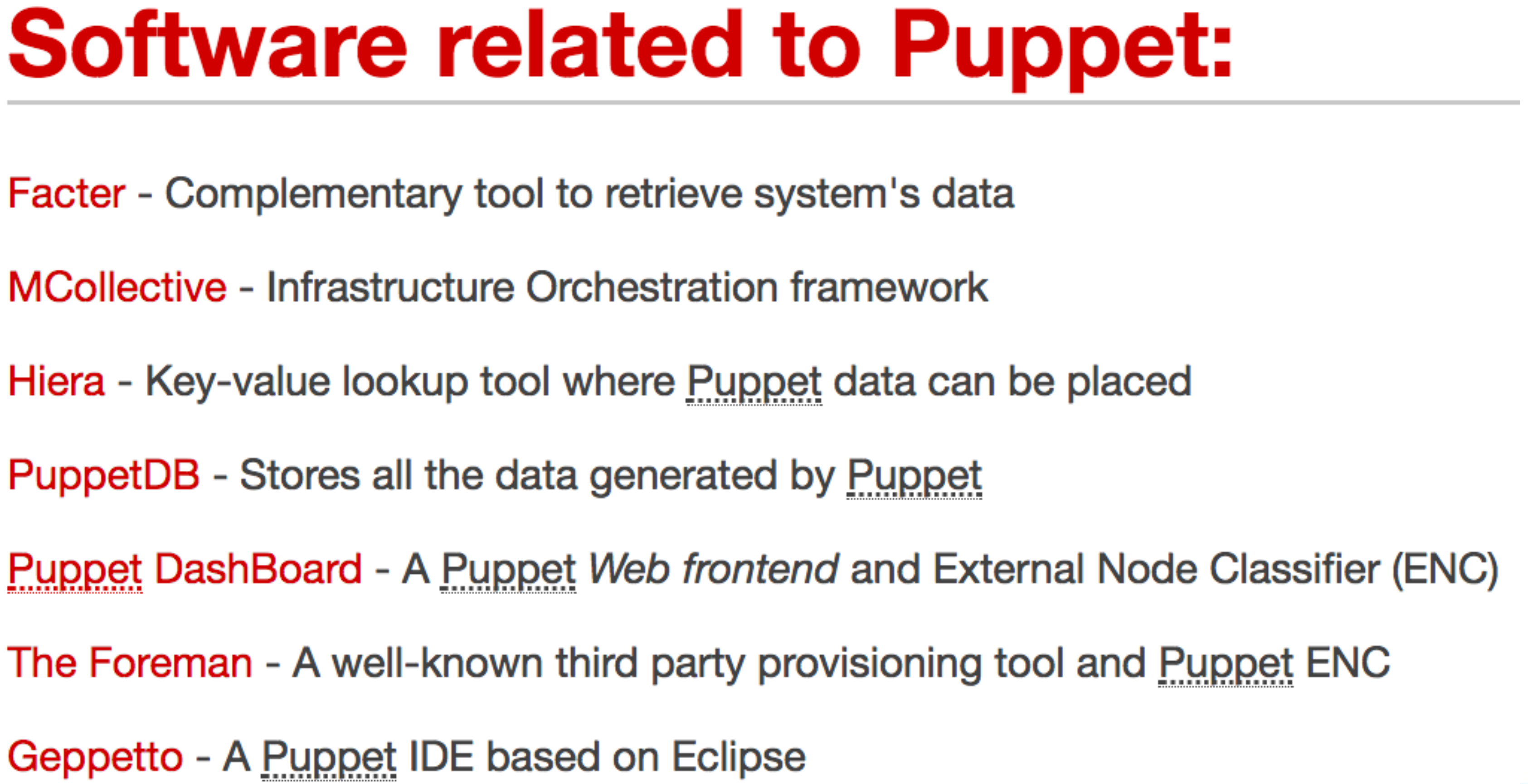
Puppet Enterprise - The commercial version

The Community - Active and vibrant

Puppet Documentation - Main and Official reference

Puppet Modules on: Module Forge and GitHub

Puppet: What is Puppet – Related Softwares



Software related to Puppet:

Facter - Complementary tool to retrieve system's data

MCollective - Infrastructure Orchestration framework

Hiera - Key-value lookup tool where Puppet data can be placed

PuppetDB - Stores all the data generated by Puppet

Puppet DashBoard - A Puppet *Web frontend* and External Node Classifier (ENC)

The Foreman - A well-known third party provisioning tool and Puppet ENC

Geppetto - A Puppet IDE based on Eclipse

Puppet: Who is Using Puppet

MOTOROLA
PayPal
Sun ORACLE
OfficeMax
THOMSON REUTERS
NHS
zynga
Square
Time Warner Cable
Symantec
Server Density
Dushkin McGraw-Hill
TransUnion
SPICEWORKS IT'S EVERYTHING IT
shopzilla
MORPH
smarsh
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
London Stock Exchange
intel
NYSE
intuit
Raytheon
salesforce
SONY
NASA
McAfee
REI
verizon
ISE INTERNATIONAL SECURITIES EXCHANGE
NetApp
NIKE
LOCKHEED MARTIN
Nestlé
Los Alamos NATIONAL LABORATORY
JOHN DEERE
Secure-24
MACQUARIE BANK
NICS
kpn

Puppet: General Information

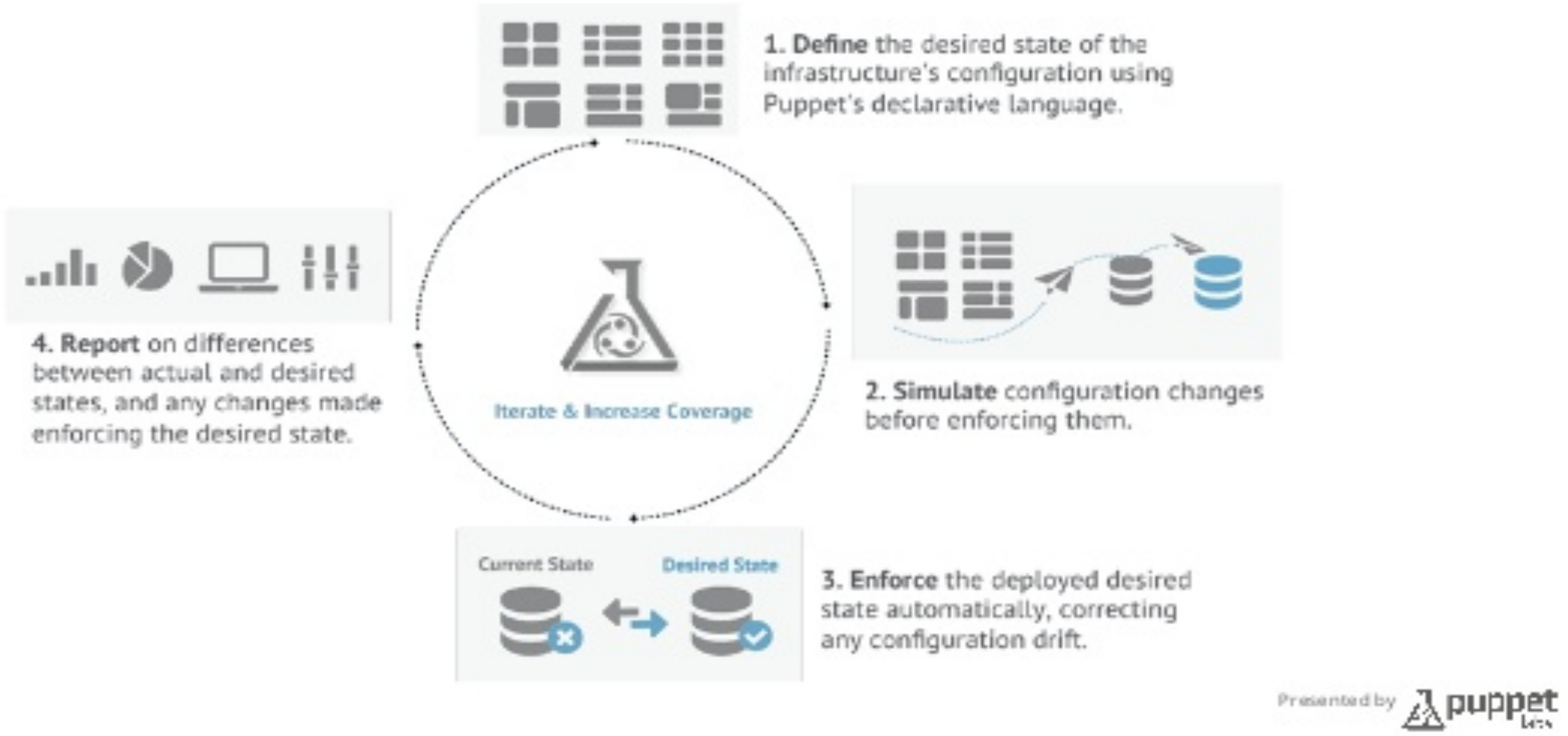
Puppet created in 2005 by Luke Kanies

Support for Linux, Unix, Windows

First commercial release in 2011

Open Source V 4.5
Puppet Enterprise 2016-1

Puppet: What is Puppet - How puppet works



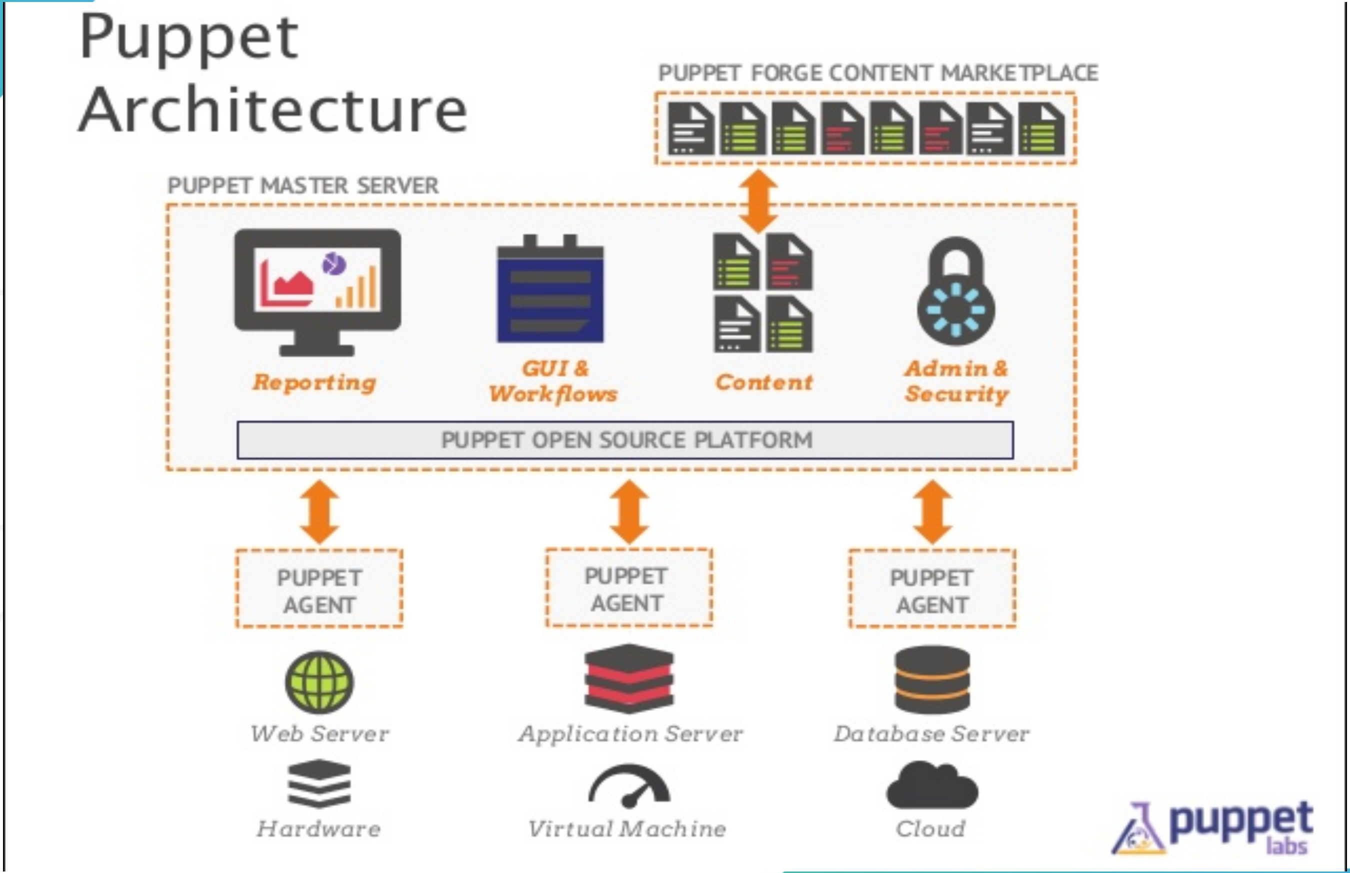
1. **Define** the desired state of the infrastructure's configuration using Puppet's declarative language.

2. **Simulate** configuration changes before enforcing them.

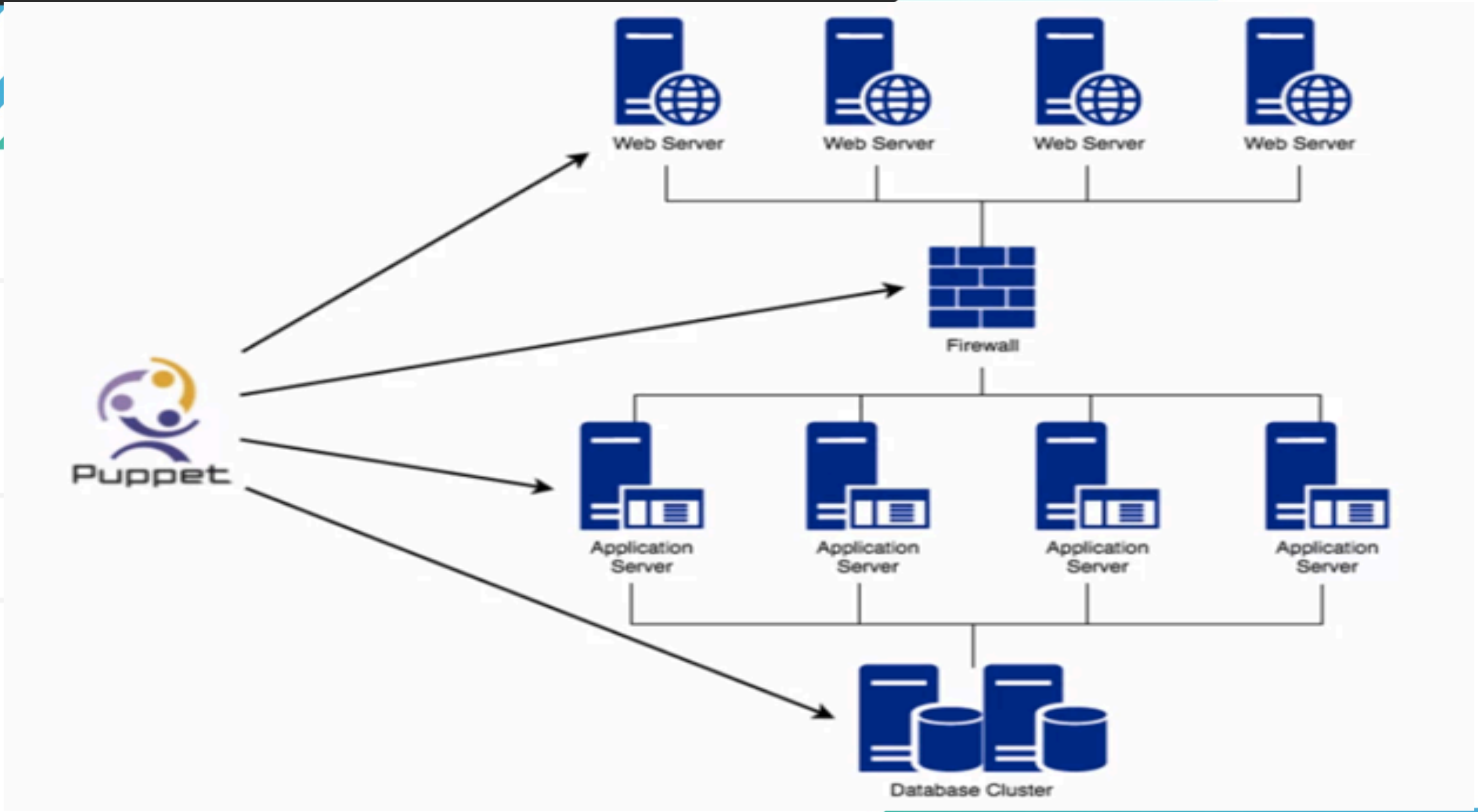
3. **Enforce** the deployed desired state automatically, correcting any configuration drift.

4. **Report** on differences between actual and desired states, and any changes made enforcing the desired state.

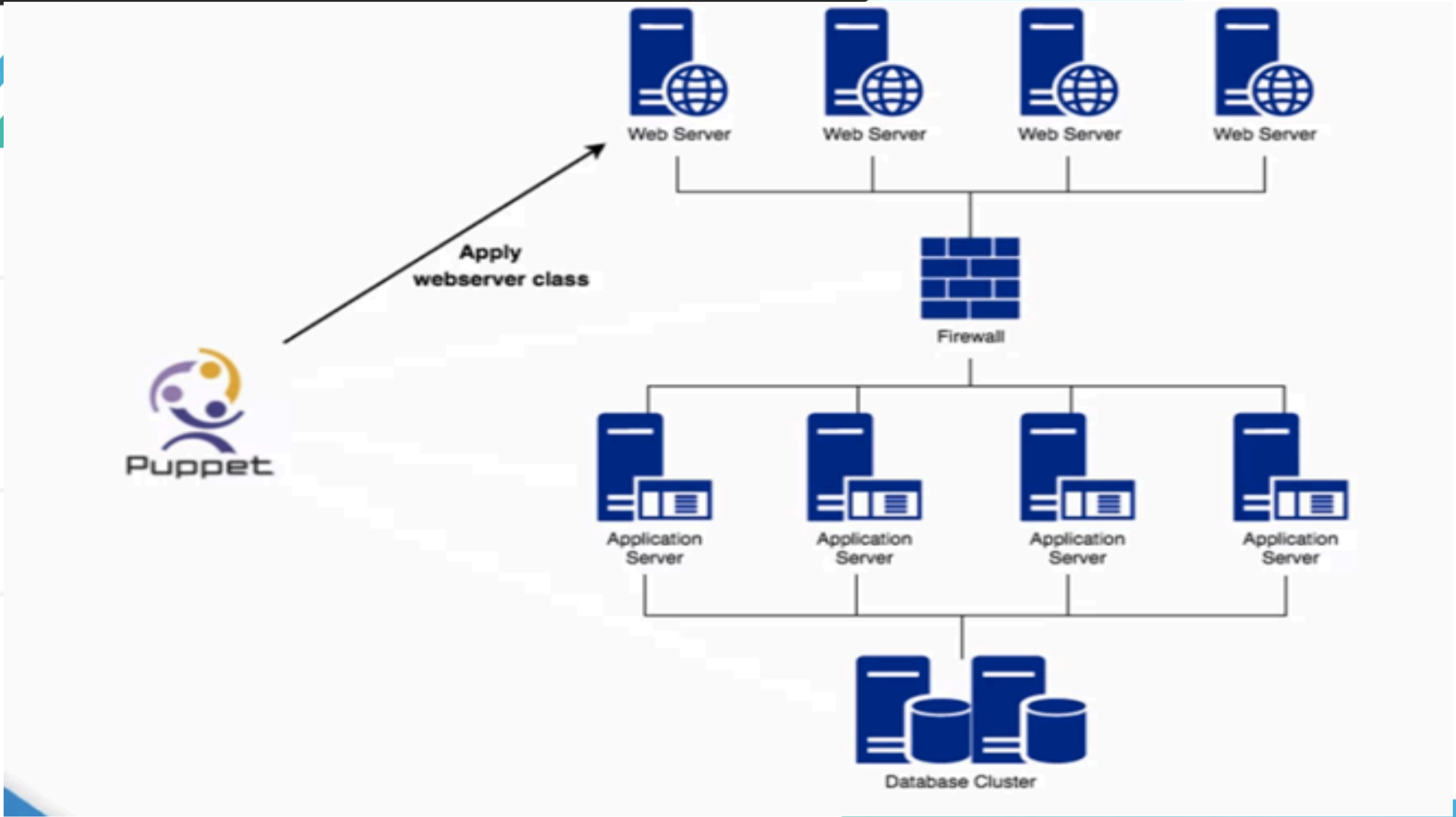
Puppet: What is Puppet - Puppet Architecture



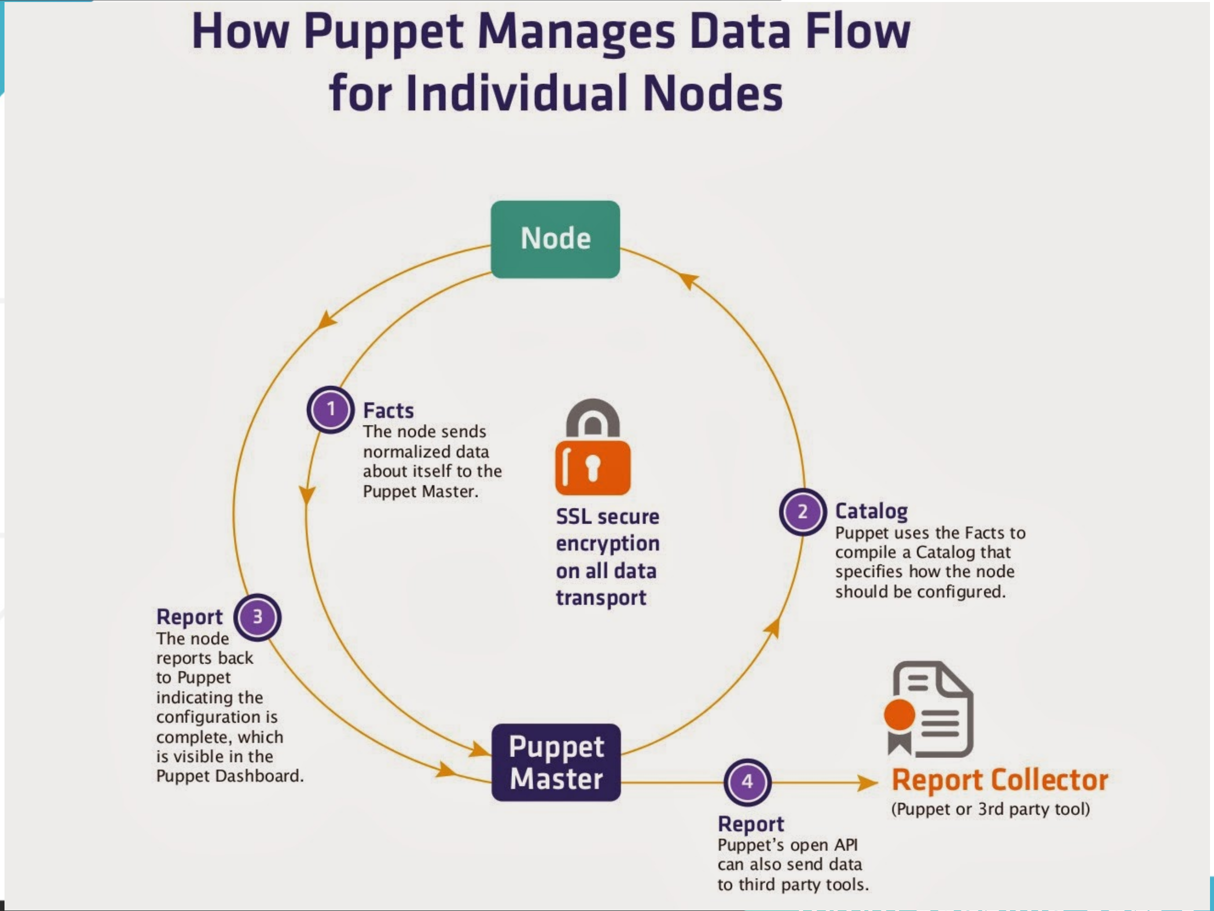
Puppet: What is Puppet - Master and Agents



Puppet: What is Puppet - Master and Agents



Puppet: What is Puppet - Master and Agents


How Puppet Manages Data Flow for Individual Nodes
Node
1
Facts
The node sends normalized data about itself to the Puppet Master.
SSL secure encryption on all data transport
2
Catalog
Puppet uses the Facts to compile a Catalog that specifies how the node should be configured.
3
Report
The node reports back to Puppet indicating the configuration is complete, which is visible in the Puppet Dashboard.
Puppet Master
4
Report
Puppet's open API can also send data to third party tools.
Report Collector
(Puppet or 3rd party tool)

Puppet: What is Puppet - Master and Agents Interaction

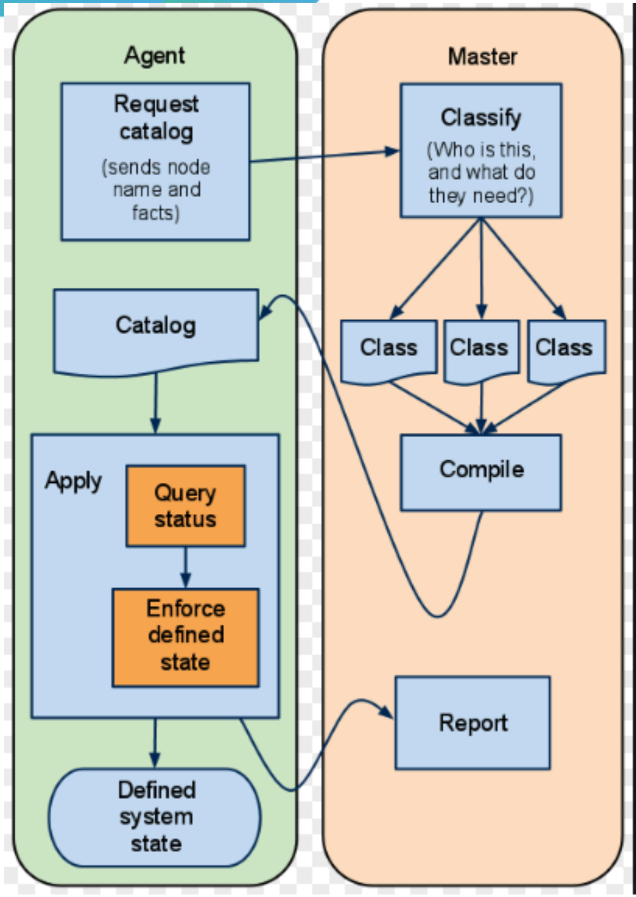
The Node (Puppet Agent) sends FACTS to Puppet Server

The Puppet Master compiles A CATALOG

All of the changes made are placed in REPORTS

Puppet Master and Nodes (Agents) communicate via SSL over TCP/8140

Nodes (Agents) pull configuration from Puppet Master



Puppet: What is Puppet – Configuration Items

Individual Configuration Items are called **resource declarations**

A Resource Declaration answers these…

What aspect of the system is to be managed

What state do we want it in

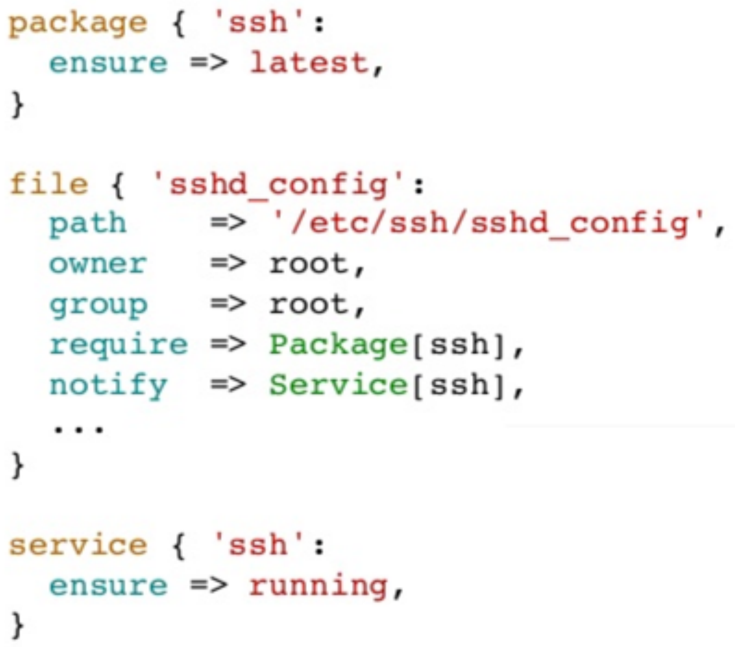
A Resource Declaration has these…

Type (can be Package, File or Service)

Title

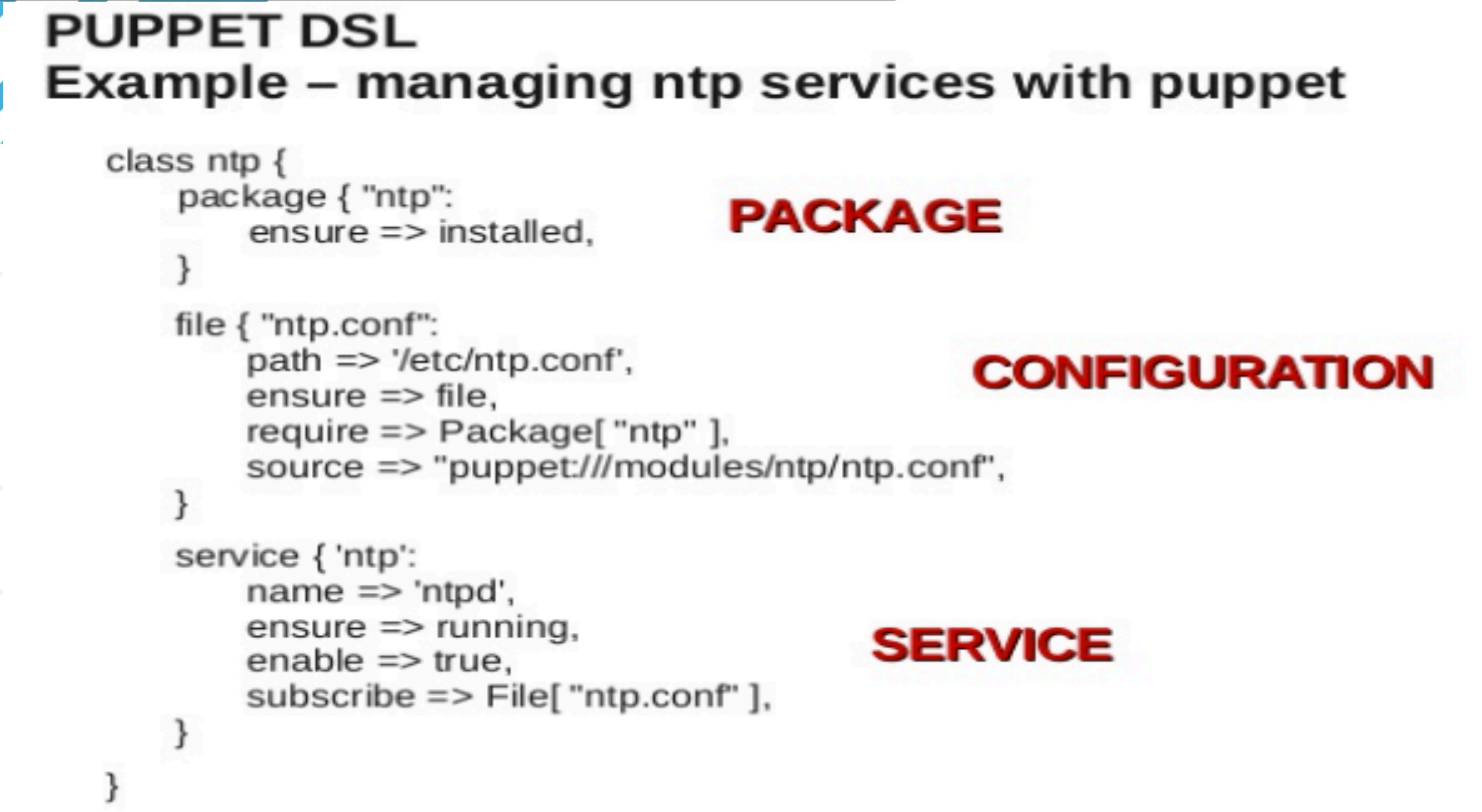
Attributes/Parameters

Provider (yum, apt etc)



```
package { 'ssh':
  ensure => latest,
}

file { 'sshd_config':
  path     => '/etc/ssh/sshd_config',
  owner    => root,
  group    => root,
  require  => Package[ssh],
  notify   => Service[ssh],
  ...
}

service { 'ssh':
  ensure => running,
}
```

Puppet: What is Puppet - Configuration Language



PUPPET DSL
Example – managing ntp services with puppet

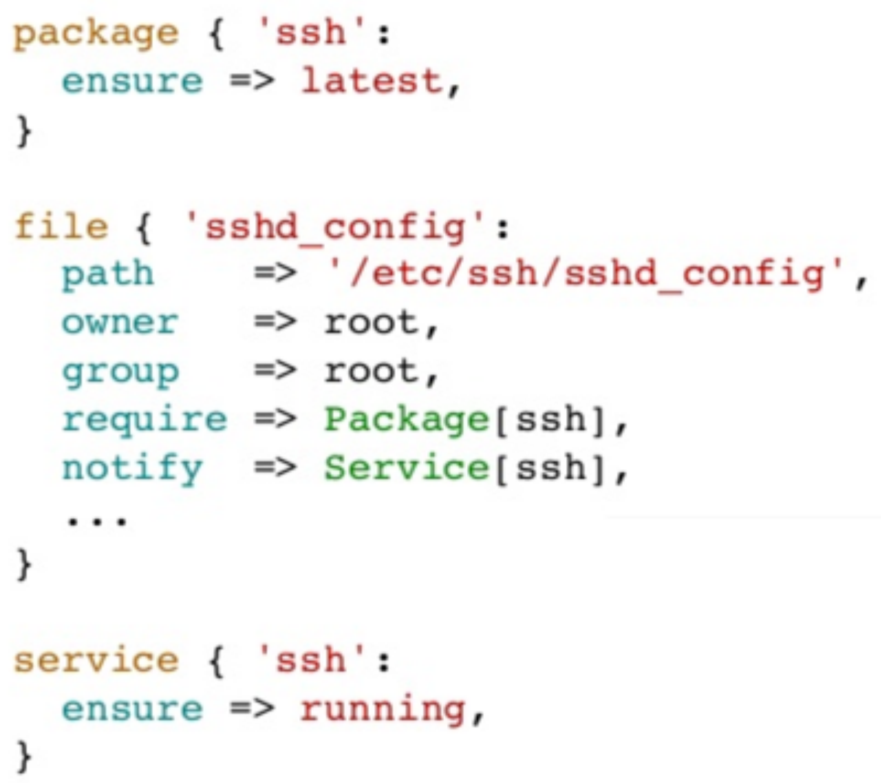
```
class ntp {
        package { "ntp":
                ensure => installed,
        }

        file { "ntp.conf":
                path => '/etc/ntp.conf',
                ensure => file,
                require => Package[ "ntp" ],
                source => "puppet:///modules/ntp/ntp.conf",
        }

        service { 'ntp':
                name => 'ntpd',
                ensure => running,
                enable => true,
                subscribe => File[ "ntp.conf" ],
        }
}
```

PACKAGE

CONFIGURATION

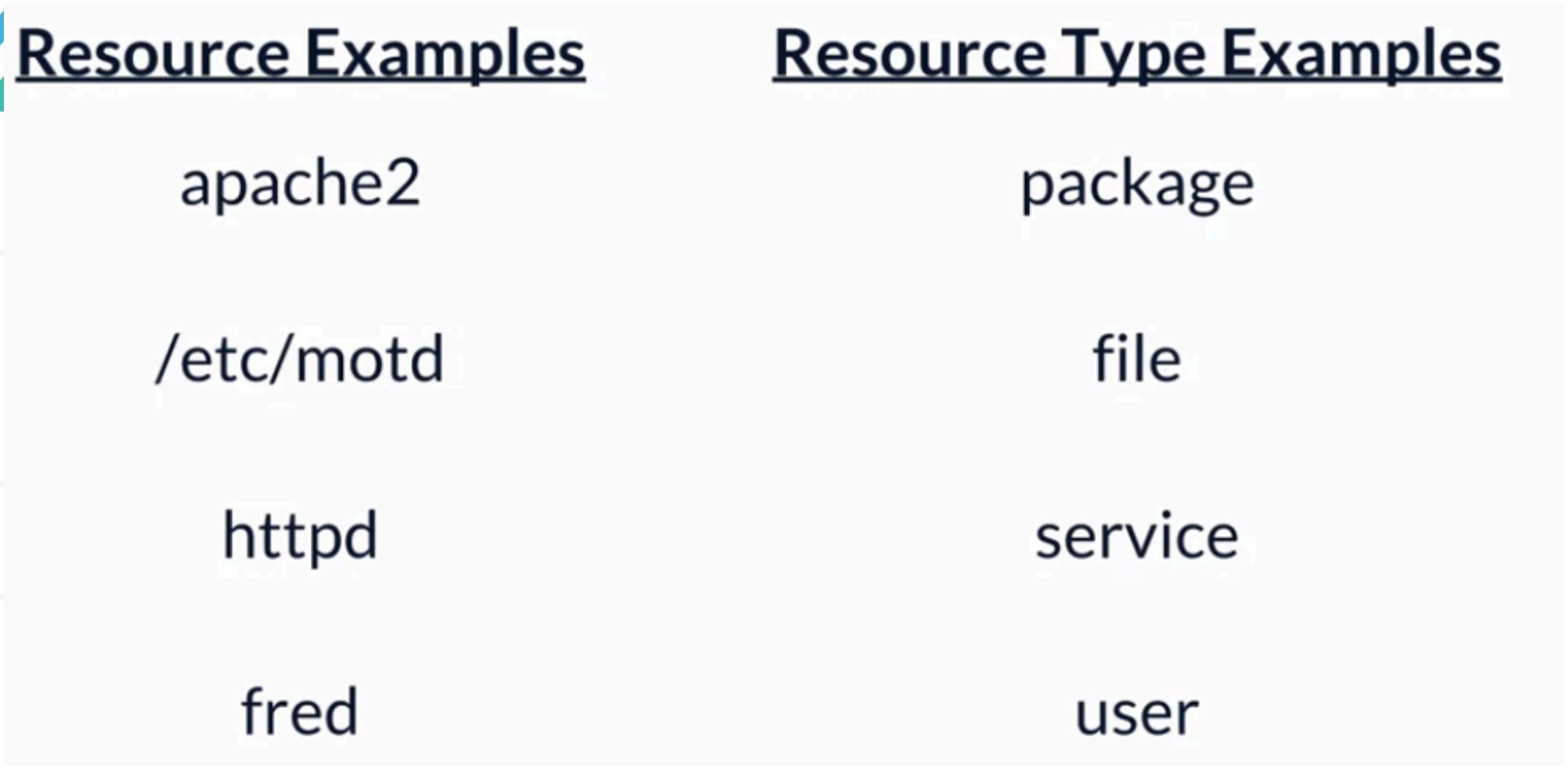
SERVICE

Puppet: What is Puppet - Configuration Language



```
package { 'ssh':
  ensure => latest,
}

file { 'sshd_config':
  path    => '/etc/ssh/sshd_config',
  owner   => root,
  group   => root,
  require => Package[ssh],
  notify  => Service[ssh],
  ...
}

service { 'ssh':
  ensure => running,
}
```

Puppet: What is Puppet - Resources



Resource Examples	Resource Type Examples
apache2	package
/etc/motd	file
httpd	service
fred	user

Puppet: What is Puppet - Resource Abstraction Layer

Puppet provides an abstraction layer between the platforms and your description of configuration. The resources defined in Puppet to configure your nodes are independent from the commands, formats and syntax required to configure these nodes.

Resource Abstraction Layer (RAL) refers to the components of Puppet that interact with the system.

The RAL provides an abstract concept of something you can manage, and it defines concrete ways of managing things.

The Puppet RAL is what allows you to write a manifest that works on several different platforms without having to remember if you should invoke apt-get install or yum install.

Puppet: What is Puppet - Transactional Layer

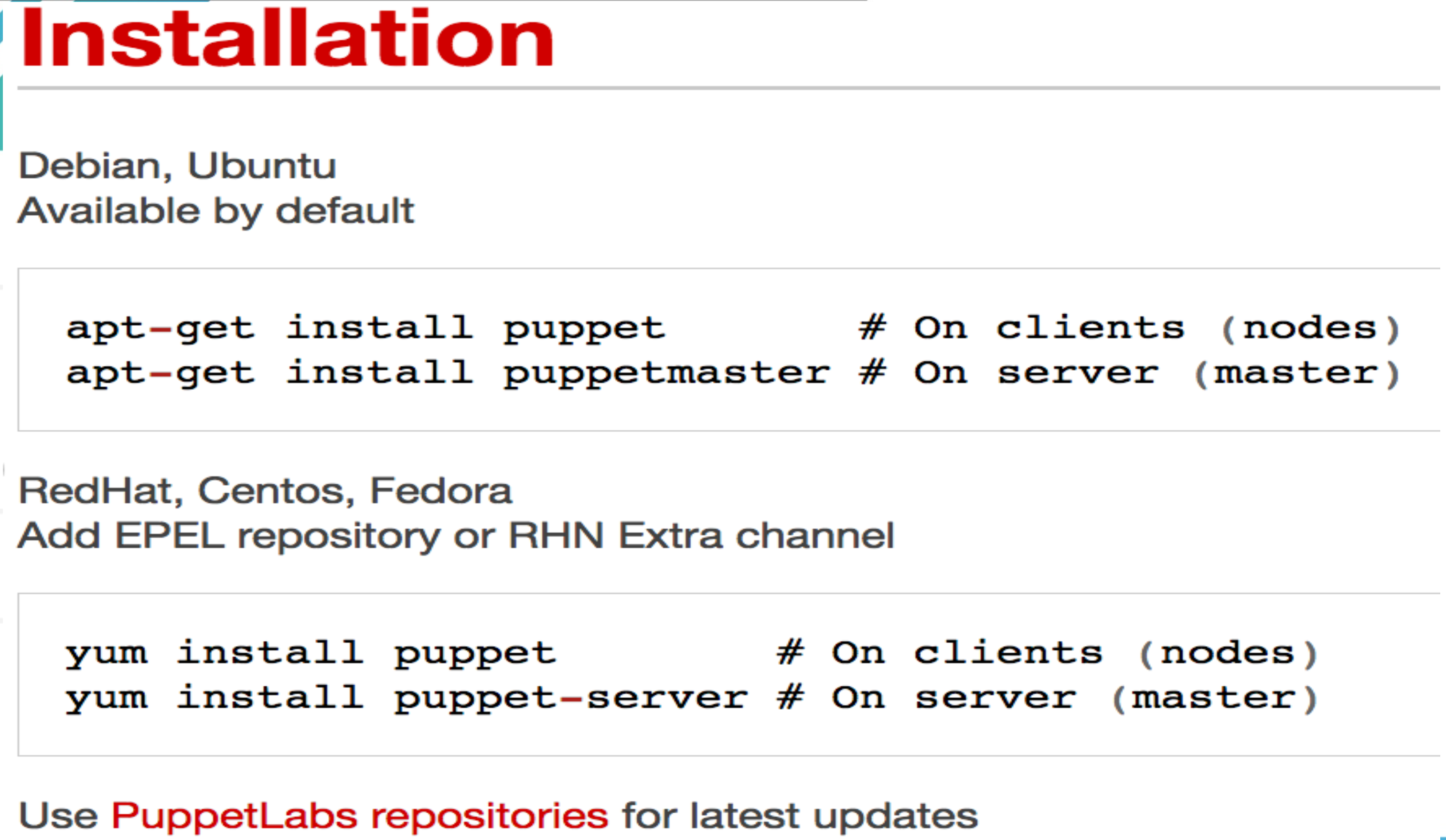
Puppet’s transactional layer is the engine of the Puppet client-server deployment. Configurations are created and can be executed repeatedly on the target hosts(nodes) as idempotent.

Puppet is not fully transactional, you cannot roll back them, but you can test transactions using noop to without applying them.

Puppet: Common Terms

- **Resources:** A resource describes something about the state of the system, such as a certain user or file should exist, or a package should be installed.
- **Manifests:** Puppet programs are called manifests. Their filenames use the .pp extension.
- **Classes:** Classes are code blocks that can be called in a code elsewhere. Using classes allows you reuse Puppet code, and can make reading manifests easier.
  - **Class Definition:** A class definition is where the code that composes a class lives. Defining a class makes the class available to be used in manifests, but does not actually evaluate anything.
  - **Class Declaration:** A class declaration occurs when a class is called in a manifest. A class declaration tells Puppet to evaluate the code within the class.
- **Modules:** A module is a collection of manifests and data (such as facts, files, and templates), and they have a specific directory structure. Modules are useful for organizing your Puppet code.

Puppet: Installation and Configuration - Installing Puppet



Installation

Debian, Ubuntu
Available by default

```
apt-get install puppet       # On clients (nodes)
apt-get install puppetmaster # On server (master)
```

RedHat, Centos, Fedora
Add EPEL repository or RHN Extra channel

```
yum install puppet        # On clients (nodes)
yum install puppet-server # On server (master)
```

Use PuppetLabs repositories for latest updates

Puppet: Installation and Configuration - Configuring Puppet Master and Agent

1. **Create Repos for Puppet Labs**
    1. **This is to setup software repositories to get the latest versions**
    2. **Update the repository**
2. **Install Puppet-Agent on Nodes**
3. **Install PuppetServer on the Puppet Server**
4. **Sign the Node's SSL Certificates on the Puppet Server**

Puppet: Installation of Puppet Server using Docker

YOU NEED TO DO THIS TODAY!

1. docker network create puppet
2. docker pull schogini/docker-puppetserver-ubuntu
3. docker run -ti --rm --network puppet --name puppet --hostname puppet schogini/docker-puppetserver-ubuntu
   1. service puppetserver start – *DON'T FORGET THIS*
   2. puppet cert clean puppetnode1 – *THIS CAN TAKE 5 – 10 mins*
   3. puppet cert list

Puppet: What is Puppet - Resources

The *puppet describe subcommand* queries types and providers for their built in documentation. This lets you look up the behavior of types, their properties and parameters, and so forth.

puppet describe file

File

Manages files, including their content, ownership, and permissions.The `file` type can manage normal files, directories, and symlinks; thetype should be specified in the `ensure` attribute.File contents can be managed directly with the `content` attribute, ordownloaded from a remote source using the `source` attribute; the lattercan also be used to recursively serve directories (when the `recurse`attribute is set to `true` or `local`). On Windows, note that filecontents are managed in binary mode;

Puppet: What is Puppet - Resource Type: File

```
puppet resource file /etc

file { '/etc':
      ensure => 'directory',
      ctime  => '2017-09-11 13:54:31 +0000',
      group  => '0',  mode   => '0755',
      mtime  => '2017-09-11 13:54:31 +0000',
      owner  => '0',  type   => 'directory',
}
```

Puppet: What is Puppet - Resource Type: Package

```
puppet resource package wget

package { 'wget':
    ensure => '1.15-1ubuntu1.14.04.2',
}
```

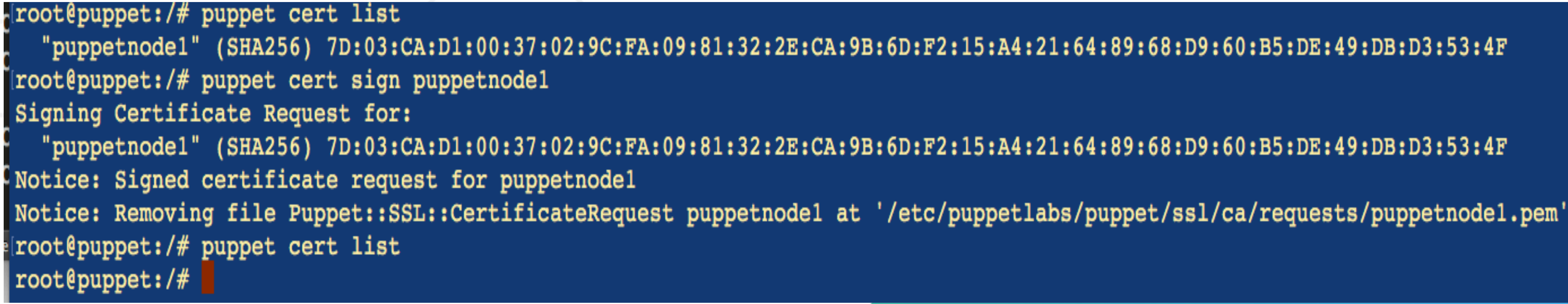
Puppet: What is Puppet - Resource Type: Package

```
puppet resource service puppet
```

```
service {'puppet':
      ensure => 'stopped',
      enable => 'true',
}
```

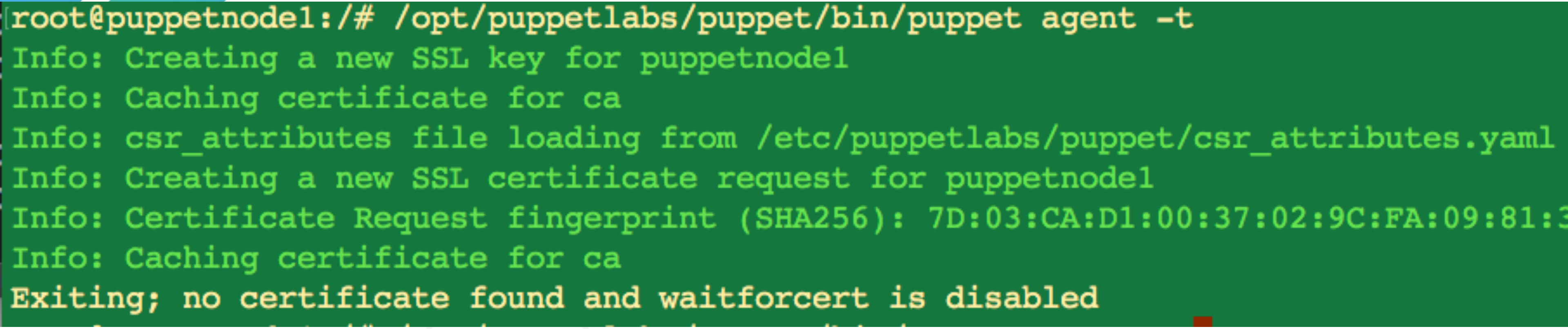
Puppet: Installation of Puppet Node using Docker

1. docker pull schogini/docker-puppetnode-ubuntu
2. docker run -ti --rm --network puppet --name puppetnode1 --hostname puppetnode1 schogini/docker-puppetnode-ubuntu
   - puppet agent –t
3. NOW BACK ON PUPPET SERVER
   - puppet cert sign puppetnode1

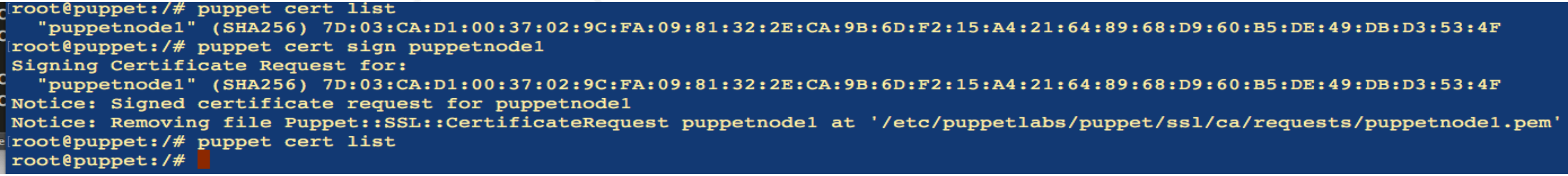


```
root@puppet:/# puppet cert list
  "puppetnode1" (SHA256) 7D:03:CA:D1:00:37:02:9C:FA:09:81:32:2E:CA:9B:6D:F2:15:A4:21:64:89:68:D9:60:B5:DE:49:DB:D3:53:4F
root@puppet:/# puppet cert sign puppetnode1
Signing Certificate Request for:
  "puppetnode1" (SHA256) 7D:03:CA:D1:00:37:02:9C:FA:09:81:32:2E:CA:9B:6D:F2:15:A4:21:64:89:68:D9:60:B5:DE:49:DB:D3:53:4F
Notice: Signed certificate request for puppetnode1
Notice: Removing file Puppet::SSL::CertificateRequest puppetnode1 at '/etc/puppetlabs/puppet/ssl/ca/requests/puppetnode1.pem'
root@puppet:/# puppet cert list
root@puppet:/#
```

Puppet: Flow of Connecting and Signing a Puppet Agent (Node)

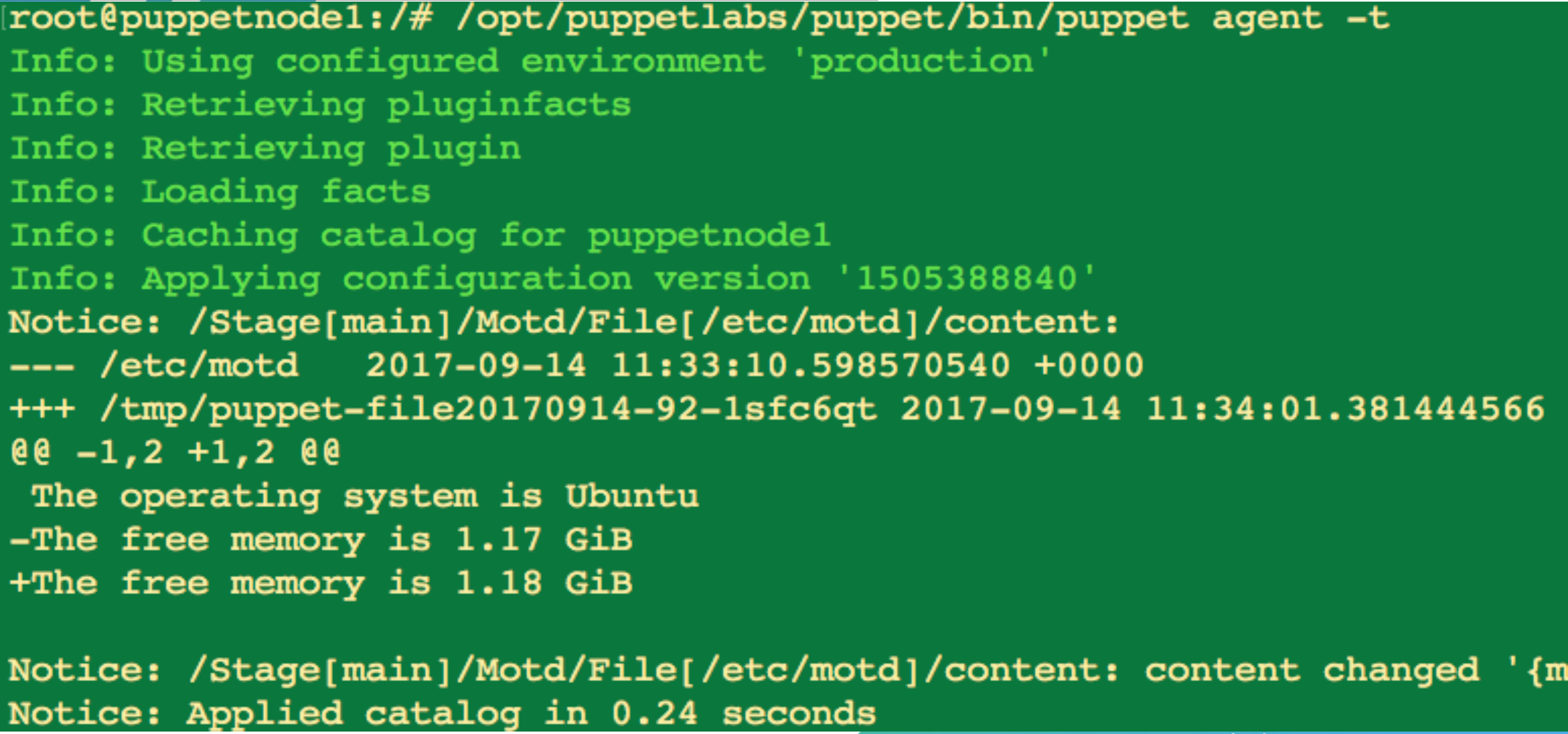


```
root@puppetnode1:/# /opt/puppetlabs/puppet/bin/puppet agent -t
Info: Creating a new SSL key for puppetnode1
Info: Caching certificate for ca
Info: csr_attributes file loading from /etc/puppetlabs/puppet/csr_attributes.yaml
Info: Creating a new SSL certificate request for puppetnode1
Info: Certificate Request fingerprint (SHA256): 7D:03:CA:D1:00:37:02:9C:FA:09:81:3
Info: Caching certificate for ca
Exiting; no certificate found and waitforcert is disabled
```



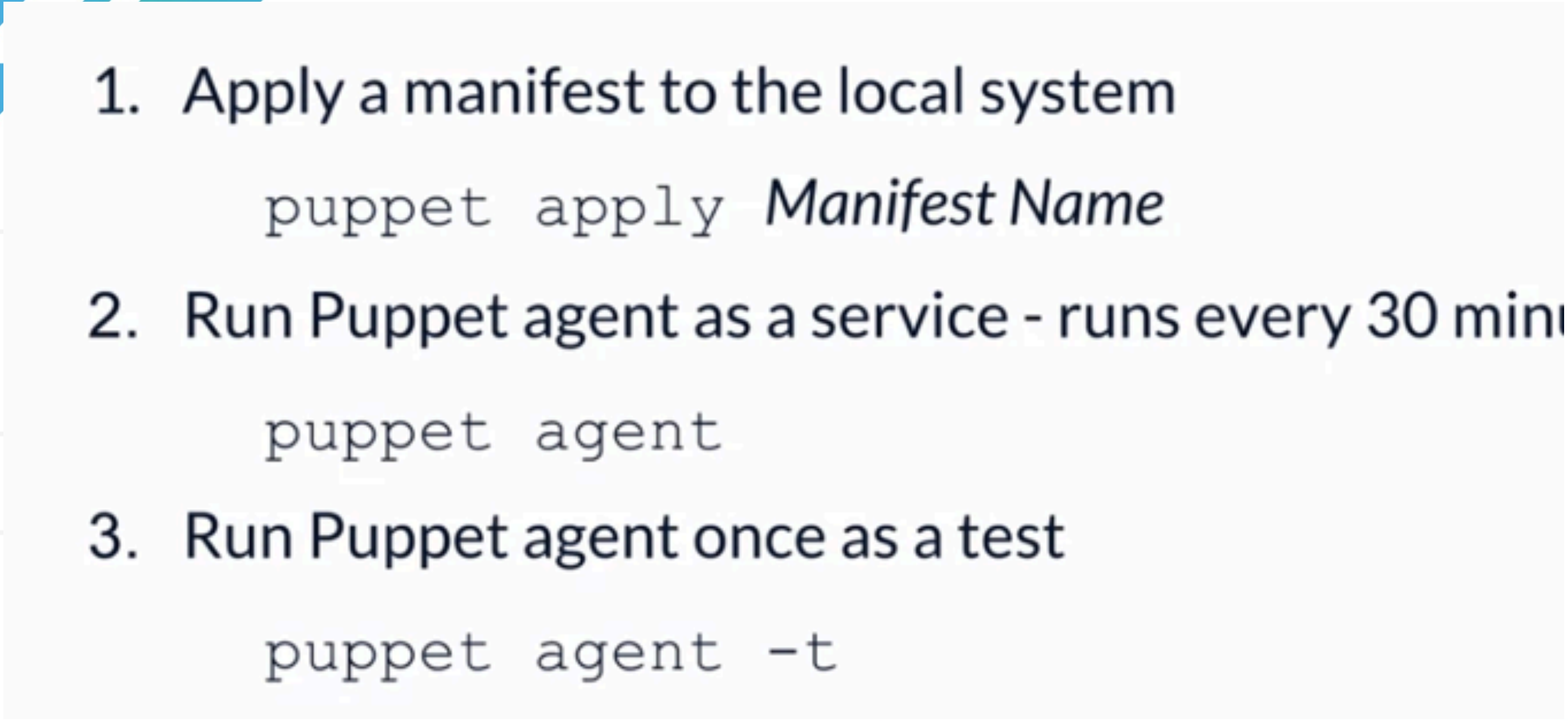
```
root@puppet:/# puppet cert list
  "puppetnode1" (SHA256) 7D:03:CA:D1:00:37:02:9C:FA:09:81:32:2E:CA:9B:6D:F2:15:A4:21:64:89:68:D9:60:B5:DE:49:DB:D3:53:4F
root@puppet:/# puppet cert sign puppetnode1
Signing Certificate Request for:
  "puppetnode1" (SHA256) 7D:03:CA:D1:00:37:02:9C:FA:09:81:32:2E:CA:9B:6D:F2:15:A4:21:64:89:68:D9:60:B5:DE:49:DB:D3:53:4F
Notice: Signed certificate request for puppetnode1
Notice: Removing file Puppet::SSL::CertificateRequest puppetnode1 at '/etc/puppetlabs/puppet/ssl/ca/requests/puppetnode1.pem'
root@puppet:/# puppet cert list
root@puppet:/#
```

Puppet: Flow of Connecting and Signing a Puppet Agent (Node)



```
root@puppetnode1:/# /opt/puppetlabs/puppet/bin/puppet agent -t
Info: Using configured environment 'production'
Info: Retrieving pluginfacts
Info: Retrieving plugin
Info: Loading facts
Info: Caching catalog for puppetnode1
Info: Applying configuration version '1505388840'
Notice: /Stage[main]/Motd/File[/etc/motd]/content:
--- /etc/motd   2017-09-14 11:33:10.598570540 +0000
+++ /tmp/puppet-file20170914-92-1sfc6qt 2017-09-14 11:34:01.381444566
@@ -1,2 +1,2 @@
 The operating system is Ubuntu
-The free memory is 1.17 GiB
+The free memory is 1.18 GiB

Notice: /Stage[main]/Motd/File[/etc/motd]/content: content changed '{m
Notice: Applied catalog in 0.24 seconds
```

Puppet: 3 Ways to Run Puppet (Applying Configuration Changes)



1. Apply a manifest to the local system

   ```
   puppet apply Manifest Name
   ```

2. Run Puppet agent as a service - runs every 30 min

   ```
   puppet agent
   ```

3. Run Puppet agent once as a test

   ```
   puppet agent -t
   ```

Puppet: Test Running of a Puppet Module

https://forge.puppet.com/puppetlabs/motd
(https://forge.puppet.com/)

ON SERVER

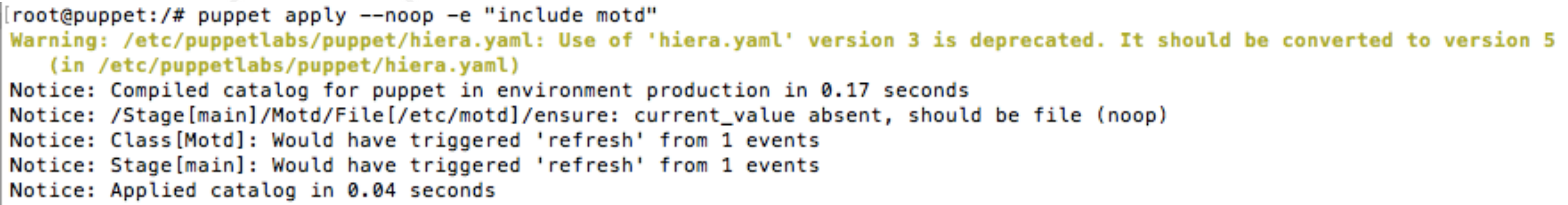
Install a puppet module

puppet module install puppetlabs-motd --version 1.5.1

Check what will happen when you apply a sample Puppet code

*puppet apply **--noop** -e "include motd"*

-e flag allows us to execute Puppet code directly via the command.
Use 'noop' mode where Puppet runs in a no-op or dry-run mode. This is useful for seeing what changes Puppet will make without actually executing the changes.



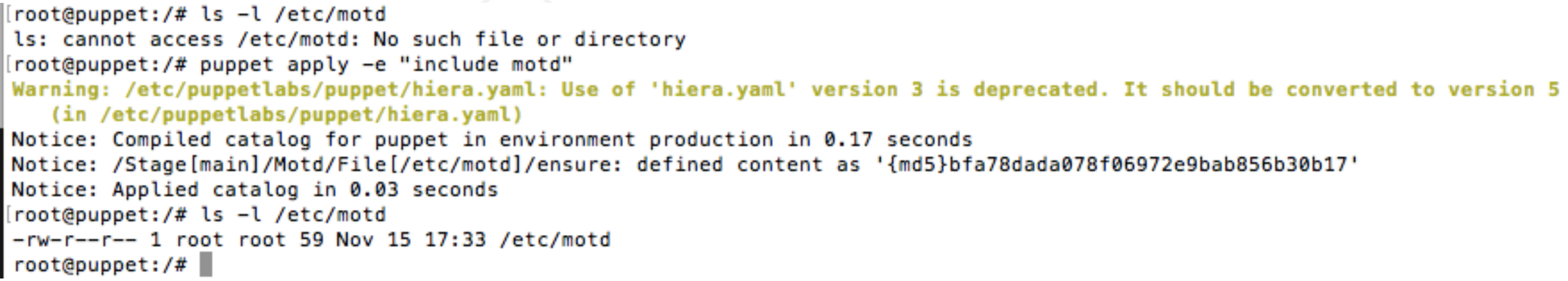
```
[root@puppet:/# puppet apply --noop -e "include motd"
Warning: /etc/puppetlabs/puppet/hiera.yaml: Use of 'hiera.yaml' version 3 is deprecated. It should be converted to version 5
   (in /etc/puppetlabs/puppet/hiera.yaml)
Notice: Compiled catalog for puppet in environment production in 0.17 seconds
Notice: /Stage[main]/Motd/File[/etc/motd]/ensure: current_value absent, should be file (noop)
Notice: Class[Motd]: Would have triggered 'refresh' from 1 events
Notice: Stage[main]: Would have triggered 'refresh' from 1 events
Notice: Applied catalog in 0.04 seconds
```

Puppet: Live Running of a Puppet module (without noop)

ON SERVER

Apply the Puppet code – no testing this time.

puppet apply -e "include motd”



```
root@puppet:/# ls -l /etc/motd
ls: cannot access /etc/motd: No such file or directory
root@puppet:/# puppet apply -e "include motd"
Warning: /etc/puppetlabs/puppet/hiera.yaml: Use of 'hiera.yaml' version 3 is deprecated. It should be converted to version 5
   (in /etc/puppetlabs/puppet/hiera.yaml)
Notice: Compiled catalog for puppet in environment production in 0.17 seconds
Notice: /Stage[main]/Motd/File[/etc/motd]/ensure: defined content as '{md5}bfa78dada078f06972e9bab856b30b17'
Notice: Applied catalog in 0.03 seconds
root@puppet:/# ls -l /etc/motd
-rw-r--r-- 1 root root 59 Nov 15 17:33 /etc/motd
root@puppet:/#
```

Puppet: HW-1

1. Create a new Docker Network
2. Install Puppet Server in Docker
3. Install Puppet Node in Docker
4. Sign the Certificates to Allow the node to communicate with the server
5. Confirm that the agent can run without errors
6. Install a Puppet Module motd
7. Apply the Puppet Module to the Puppet Server locally

Cognixia

THANK YOU