Automation With Puppet

Keywords

Automation, Configuration Management, Infrastructure As Code (IAC)

References

- https://puppet.com/docs

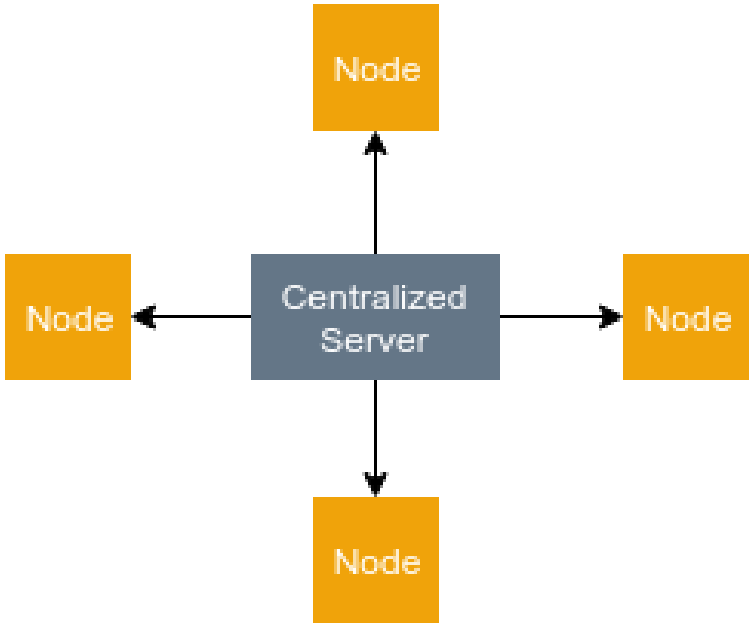
Configuration Management

What is Configuration Management?

- The process of standardizing resource configurations and enforcing their state across IT infrastructure in an automated yet agile manner.
- Automate the process of provisioning, configuration, and management of a server infrastructure.
- The concept of turning your server infrastructure into code (Infrastructure As Code).

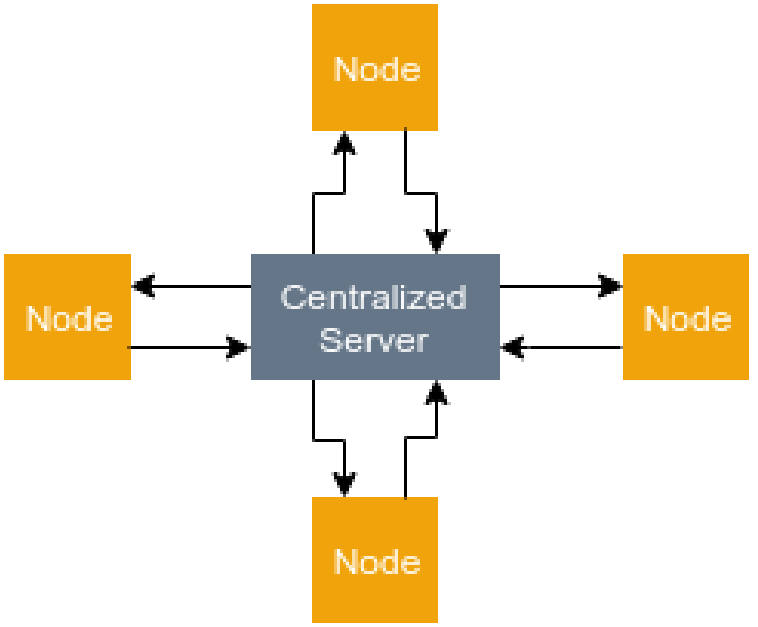
Configuration Management Type

Push Configuration



- The centralized Server pushes the configurations to the nodes.

Pull Configuration



- The nodes pull configuration from centralized server.
- Nodes dynamically update themselves with the configuration present in the server.

Puppet

Puppet

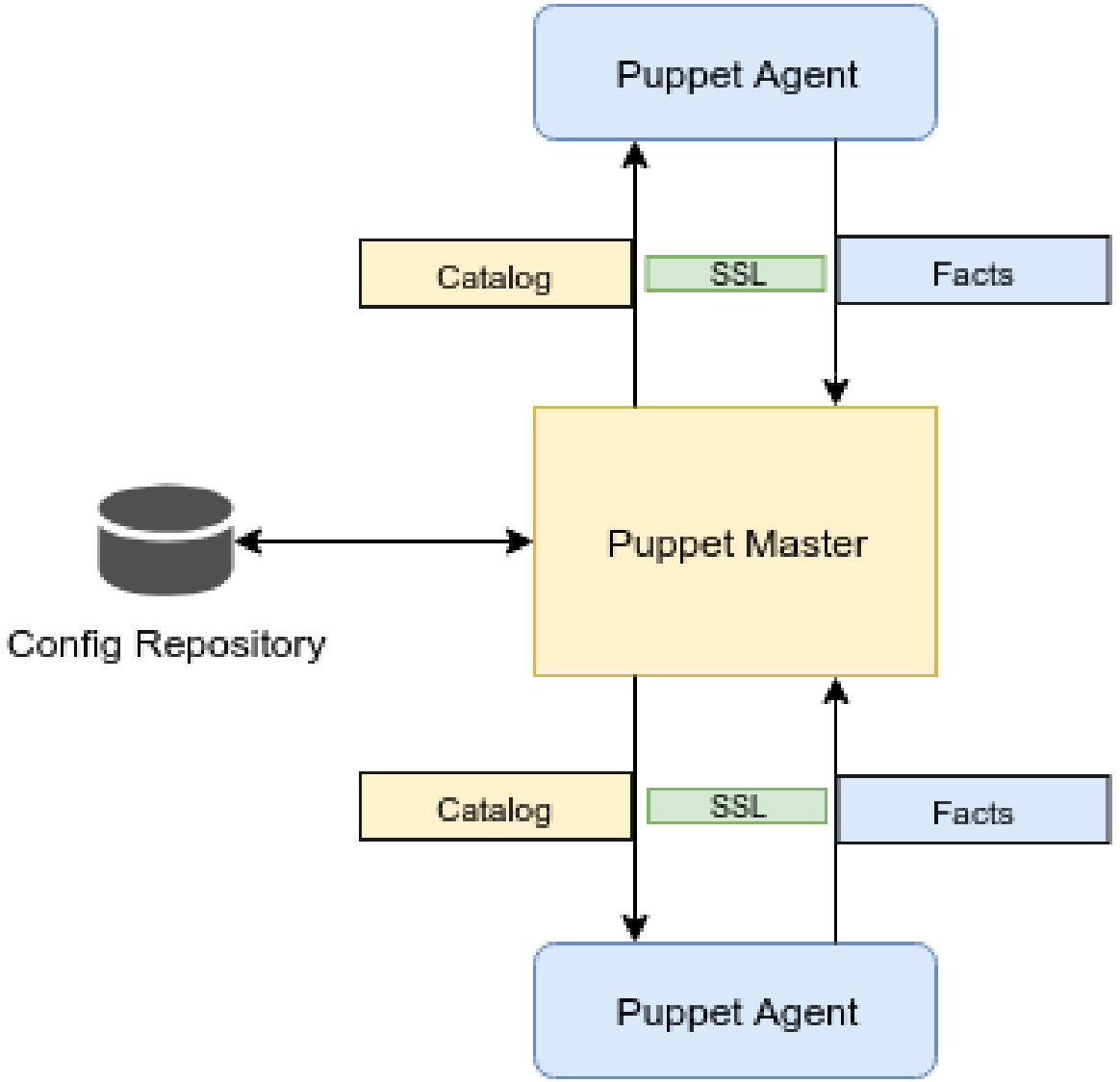
- A configuration management tool to automate the provisioning, configuration, and management of a server infrastructure.
- Runs on most Linux distributions, various UNIX platforms, and Windows.
- Includes its own declarative language to describe system configuration (Manifest)
- Using client-server paradigm to distribute compiled manifest (called catalog) via encrypted communication (SSL)

Puppet Variant

- Puppet Enterprise & Open Source Puppet
- Key differences:
  1. Web management tool
  2. Role-based access control
  3. Enterprise Support (24/7)
  4. Node Management
  5. Reporting

** https://puppet.com/products/why-puppet/puppet-enterprise-and-open-source-puppet*

Puppet Architecture



Puppet Architecture (1)

- **Puppet Master**

  Puppet Master is the key mechanism which handles all the configuration related stuff. It applies the configuration to nodes using the Puppet agent.

- **Puppet Agent**

  Puppet Agents are the actual working machines which are managed by the Puppet master. They have the Puppet agent daemon service running inside them.

- **Config Repository**

  This is the repo where all nodes and server-related configurations are saved and pulled when required

- **Facts**

  Facts are the details related to the node or the master machine, which are basically used for analyzing the current status of any node.

- **Catalog**

  All the manifest files or configuration which are written in Puppet are first converted to a compiled format called catalog and later those catalogs are applied on the target machine.

OS Supported by Puppet

- Linux: Debian, Ubuntu, Centos, RedHat, Fedora, Suse...
- Microsoft: Vista, Win 7, 8, 10, Server
- Mac
- Solaris
- IBM AIX
- Cisco: IOS-XR, NX-OS
- Ariesta EOS

Puppet Key Element

Puppet Manifest

All Puppet programs which are built with an intension of creating or managing any target host machine. It's written using Ruby programming language and saved with extension of .pp

```
1  node default { }
2
3  node 'linuxagent.example.com' {
4    include lampserver
5  }
6
7  node 'windowsagent.example.com' {
8    include iisserver
9  }
```

Puppet Resources

Resource is an unit of puppet mainly used for modeling and maintaining system configuration. Puppet resources list can be shown using command:

puppet describe --list

```
user { 'budi':
   ensure    => present,
   uid       => '552',
   shell     => '/bin/bash',
   home      => '/home/budi',
}

file { '/shared/readme.txt':
    ensure  => present,
    content => 'download instruction at doe.com/guides'
}
```

Puppet Class and Module

Puppet classes are defined as a collection of resources, which are grouped together in order to get a target node or machine in a desired state. These classes are defined inside Puppet manifest files which is located inside Puppet modules.

```
class basic_tools {
  $basic_packages = ['python-minimal', 'python-pip']
  package { $basic_packages:
    ensure  => 'installed'
  }
}

node 'default' {
  class { basic_tools: }
}
```

Fileserver

Distributing file from puppet master to agents

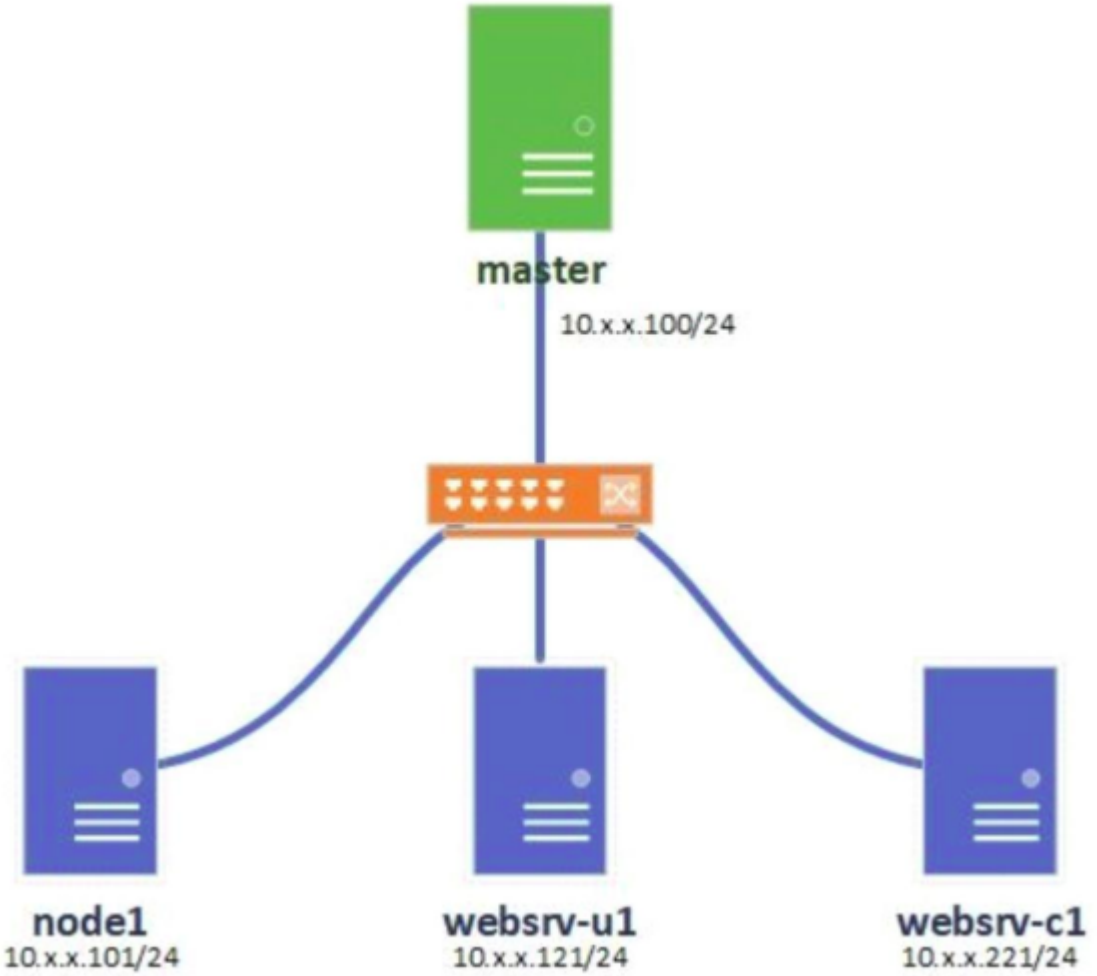
```
file { "/var/www/html/test.zip" :
  source  => "puppet:///modules/mymodule/test.zip",
}
```

Lab 1

Automation with Puppet

Lab 1 Topology

- Puppet architecture
- Installing puppet
- Puppet manifest
- Facter
- Conditional statement
- Node selection
- Using puppet module

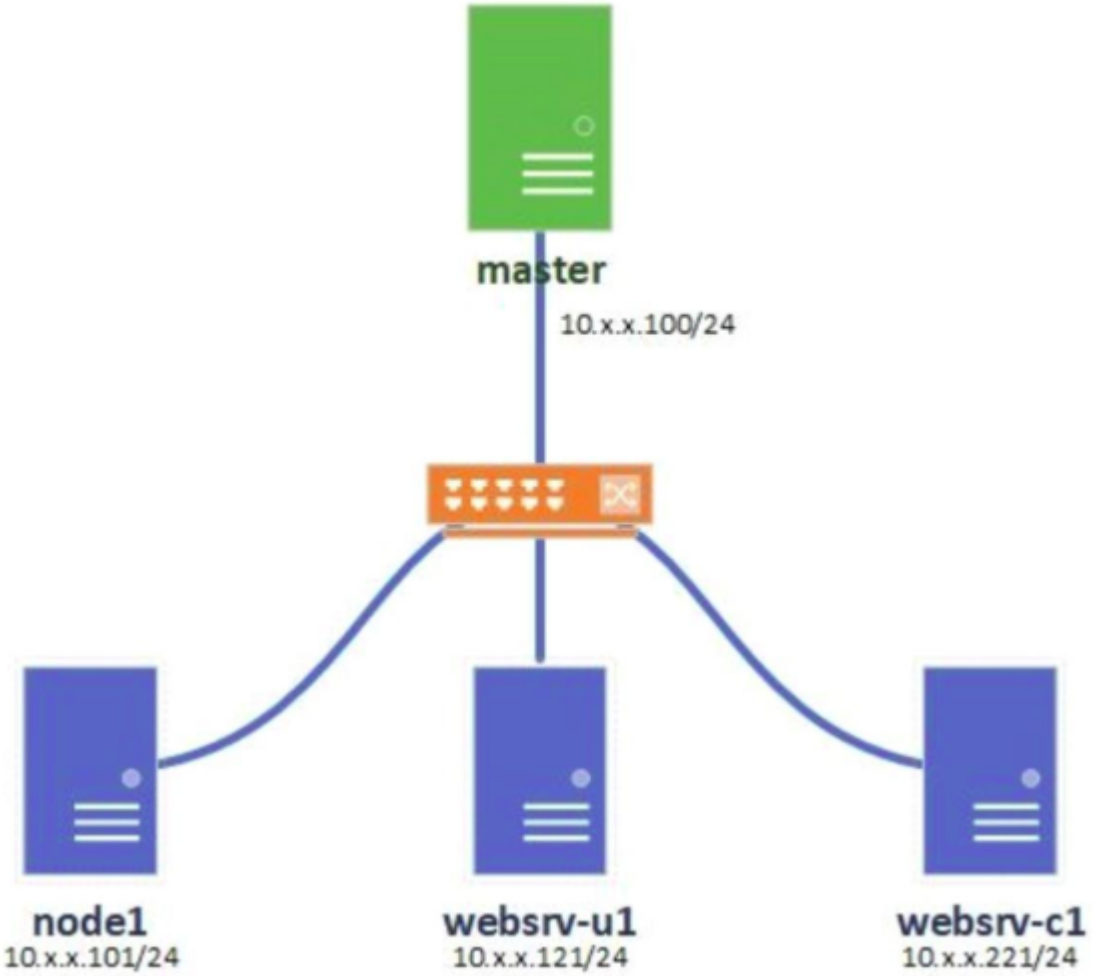


Lab 2

Automation with Puppet

Lab 2 Topology

- Create and using class
- Creating module
- File Server



NolSatu.id