



# Puppet

Configuration management tool

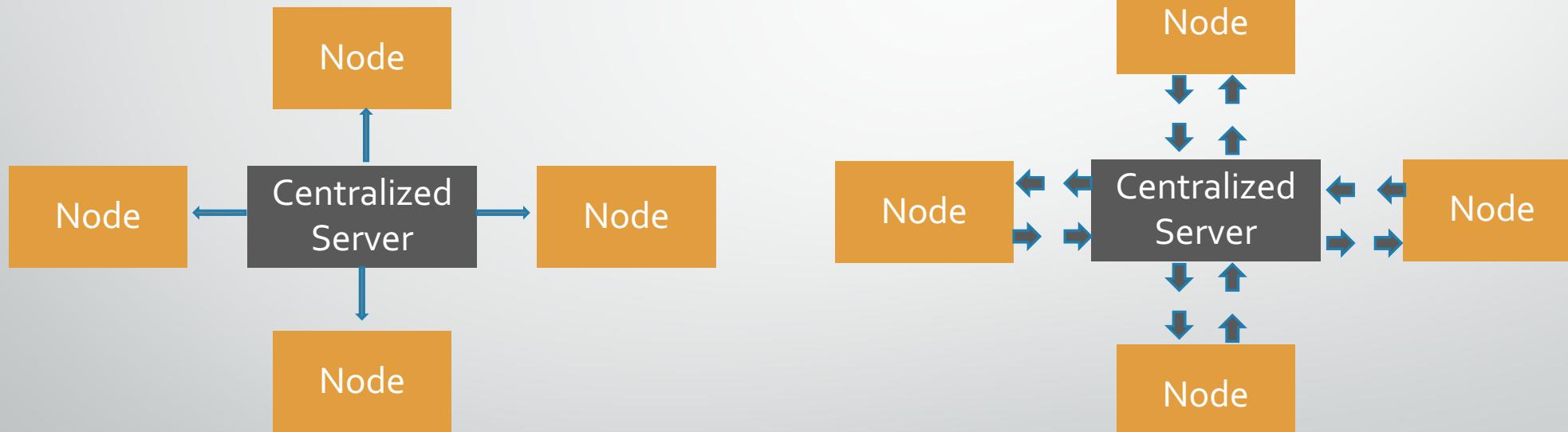
By :-

- Nitin Ramchandani
- Pinakin Abhiyankar
- Ronak Ghadiya
- Shalini Sejwani

# What is configuration management and Why is it needed

- Configuration management is the process of standardizing resource configurations and enforcing their state across IT infrastructure in an automated yet agile manner.
- Configuration Management handles changes systematically so that a system maintains its integrity over time.
- Configuration Management is the concept of turning your server infrastructure into code.

# There are two types of Configuration Management Approaches: Push and Pull



# What is puppet

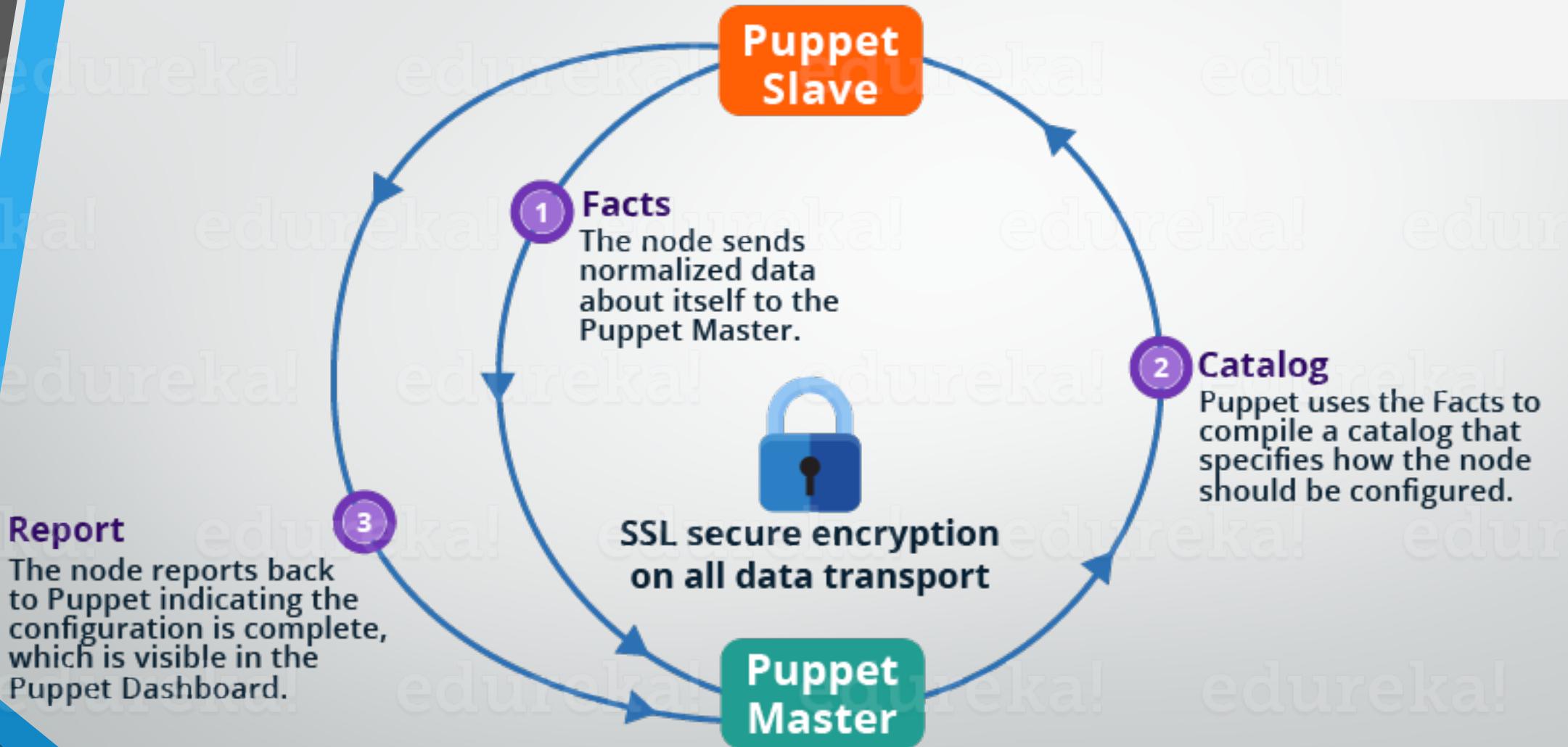
- Puppet is configuration tool
  - deploying
  - configuring
  - managing server
- It uses a Master-Slave architecture or master-less architecture
- Puppet uses Pull approach

# Puppet Primary Components

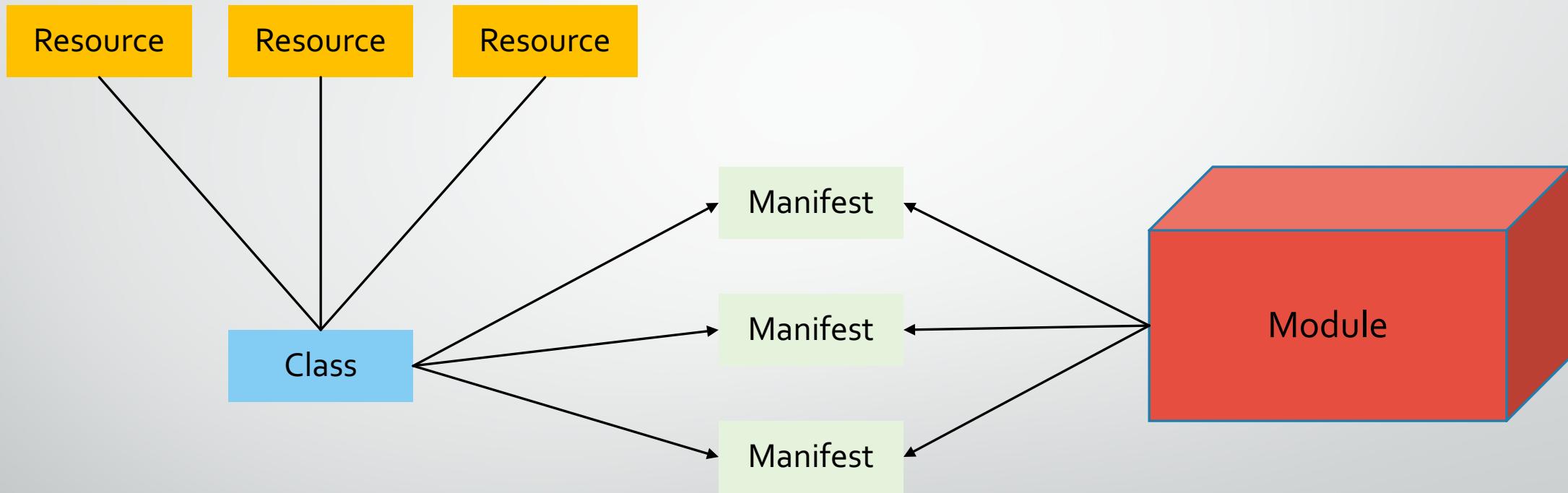
- Puppet Master - Received queries and status reports from the Puppet Agents; provides commands to Puppet Agents
- Puppet Agents – Queries Puppet Master; runs Master commands as needed, reports results back to master
- Reporting/Analytics – Visibility to into puppet agents including configuration logs, metrics on timing, resources and changes
- Puppet DB- Holds information about every node within the infrastructure

# Master-Slave Communication using SSL





# Components of Puppet



# Resource

- Resource describes some aspect of the system like a specific service or package.
- Puppet has many built-in resource types, like [files](#), [cron jobs](#), [services](#), etc.

```
# A resource declaration:  
file { '/etc/passwd':  
    ensure => file,  
    owner  => 'root',  
    group  => 'root',  
    mode   => '0600',  
}
```

Every resource has a **resource type**, a **title**, and a set of **attributes**:

```
<TYPE> { '<TITLE>':  
    <ATTRIBUTE> => <VALUE>,  
}
```

# Class

- Groups of resources can be organized into classes.
- While a resource may describe a single file or package. A class may describe everything needed to configure entire application

```
# A class with no parameters
class base::linux {
  file { '/etc/passwd':
    owner => 'root',
    group => 'root',
    mode   => '0644',
  }
  file { '/etc/shadow':
    owner => 'root',
    group => 'root',
    mode   => '0440',
  }
}
```

## Manifests:

- Puppet programs are called manifests. Manifests are collection of puppet resources. They use .pp extension.
- For example: You can write a Manifest in Puppet Master that creates a file and installs Apache server on all Puppet Slaves connected to the Puppet Master.

```
class { 'apache': }
  apache::vhost { 'example.com':
    port      => '80',
    docroot   => '/var/www/html'
  }
}
```

## Module

- A Puppet 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, because they allow you to split your code into multiple Manifests.
- The **Puppet Forge** is a repository of community approved modules.

	Puppet	Ansible	Saltstack	Chef
Scalability	Highly scalable	Highly scalable	Highly scalable	Highly scalable
Ease of Setup	Master-Agent Puppet server runs on master machine and client runs as agent on each client machine	Master-Node It has only master running on server machine, but no agents running on client machine. Client requires no special setup, hence is faster	Master-Agent Here Server is called master and Clients are called as minions which run as agents in client machine	Master – Agent Chef server runs on master machine and chef clients run as agents on each client machine
Availability	If the active master goes down, the other master takes its place	Runs with a single active node, called the primary instance. If primary goes down, Secondary instance takes its place	It can have multiple masters configured, if one goes down, agents connect with the other master	When there is failure in primary server, i.e., chef server, it has a backup server to take the place of primary server
Management	Not very easy to learn to manage configurations as it uses its own language called Puppet DSL	Easy to learn to manage the configurations. Good for real time application.	Easy to learn and manage configurations. Immediate remote execution	One needs to be a programmer to manage configurations as it offers configuration in Ruby DSL
Interoperability	Master works only on Linux/Unix but Agent also works on windows	Server works only on Linux/Unix. It supports windows as well.	Master works on Linux/Unix but minions can work on windows as well	Chef Server works in Linux/Unix but Client can be on Windows

# Comparison

	Puppet	Ansible	Chef	Salt Stack
<b>Commits:</b>	24752	29790	19184	80954
<b>Branches:</b>	11	48	194	17
<b>Contributors:</b>	442	2643	943	1778
<b>Release:</b>	303	151	507	145

# Companies that use puppet





# Demo