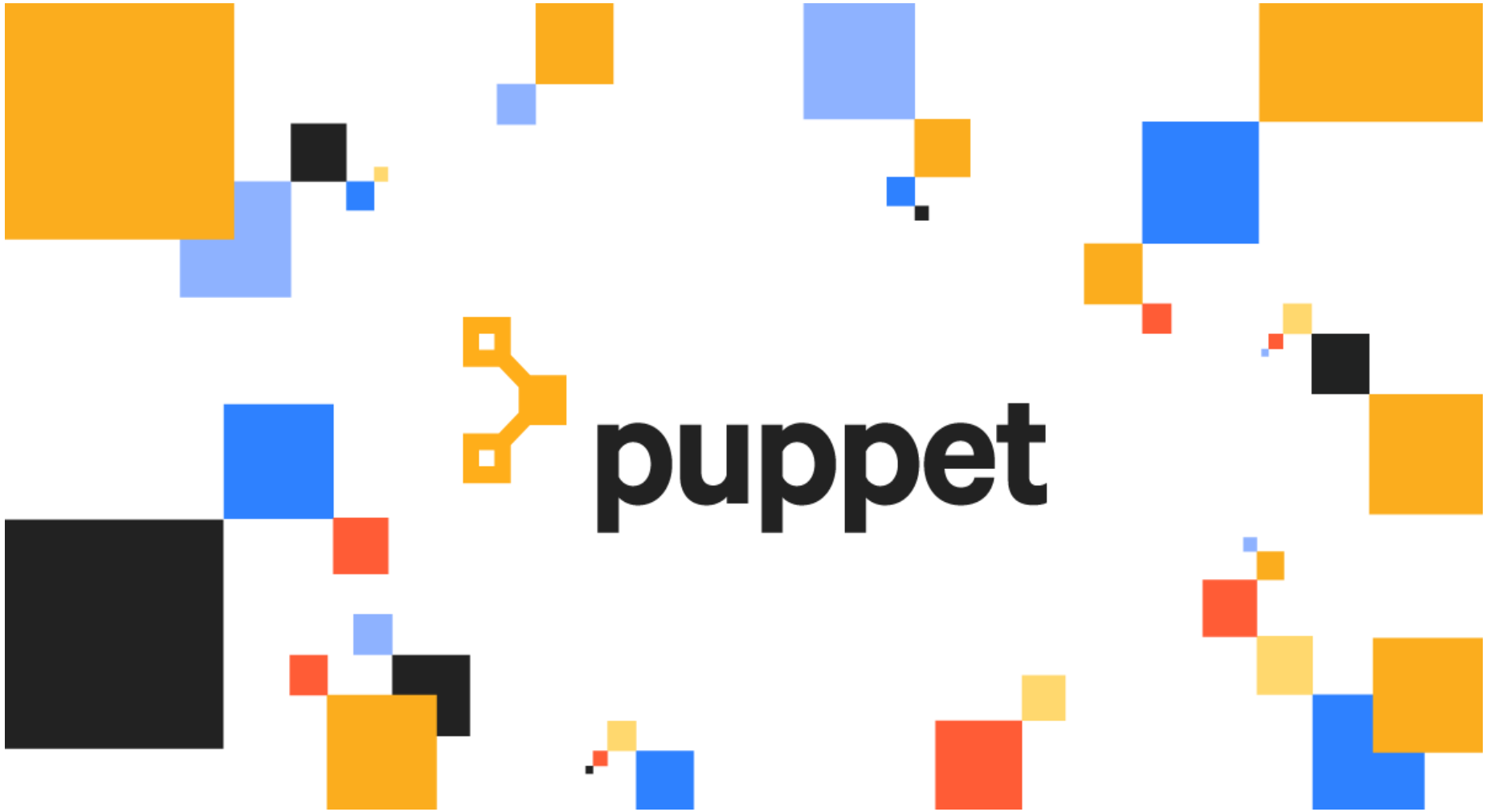
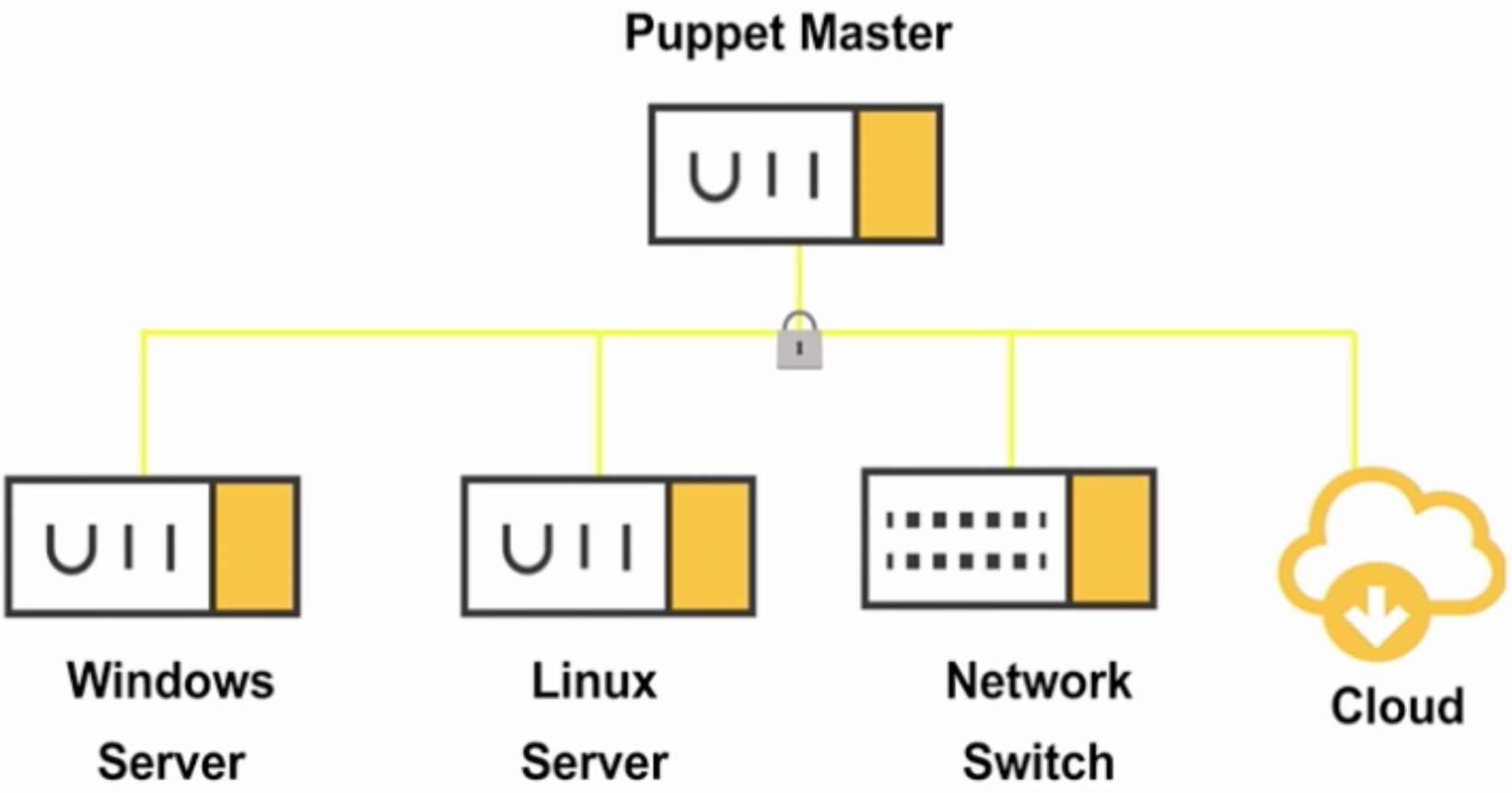

puppet

Puppet

- ✓ Puppet is an open source DevOps systems management tool for centralizing and automating the configuration management process.
- ✓ It is used to configure, manage, deploy, and orchestrate various applications and services across the whole infrastructure of an organization.
- ✓ Puppet is specially designed to manage the configuration of Linux and Windows systems. It is written in Ruby and uses its unique **D**omain **S**pecific **L**anguage (DSL) to describe system configuration.


Puppet Master
Windows Server
Linux Server
Network Switch
Cloud

Why do we use Puppet?

- Puppet provides more platform support. For example, the following are the platforms we can manage using Puppet:
  - Debian/Ubuntu
  - Microsoft Windows
  - Red Hat/CentOS/Fedora
  - MacOS X
- Puppet has better documentation
- Unlike other configuration management tools, Puppet, after deploying a configuration on any system, keeps verifying those configurations in certain intervals. These intervals can be modified as well
- Puppet is being used by some of the major organizations in the industry, namely, Google, Red Hat, etc.
- Puppet helps DevOps professionals and System Administrators work faster and smarter

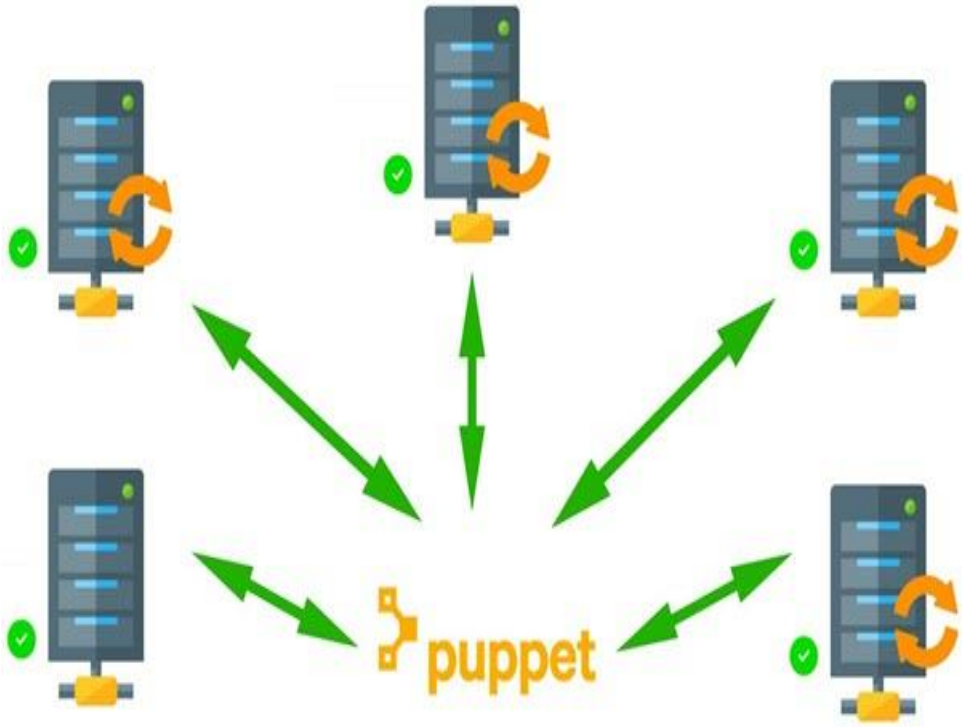
What are the Puppet versions?

- **Open Source Puppet**: It is a basic version of Puppet configuration management tool, which is also known as Open Source Puppet. It is available directly from Puppet's website and is licensed under the Apache 2.0 system.
- **Puppet Enterprise**: Commercial version that offers features like compliance reporting, orchestration, role-based access control, GUI, API and command line tools for effective management of nodes.

What Puppet can do?

For example, you have an infrastructure with about 100 servers. As a system admin, it's your role to ensure that all these servers are always up to date and running with full functionality.

To do this, you can use Puppet, which allows you to write a simple code which can be deployed automatically on these servers. This reduces the human effort and makes the development process fast and effective.



Deployment models of configuration management tools

There are two deployment models for configuration management tools :

Push-based deployment model: initiated by a master node.

Pull-based deployment model: initiated by agents.

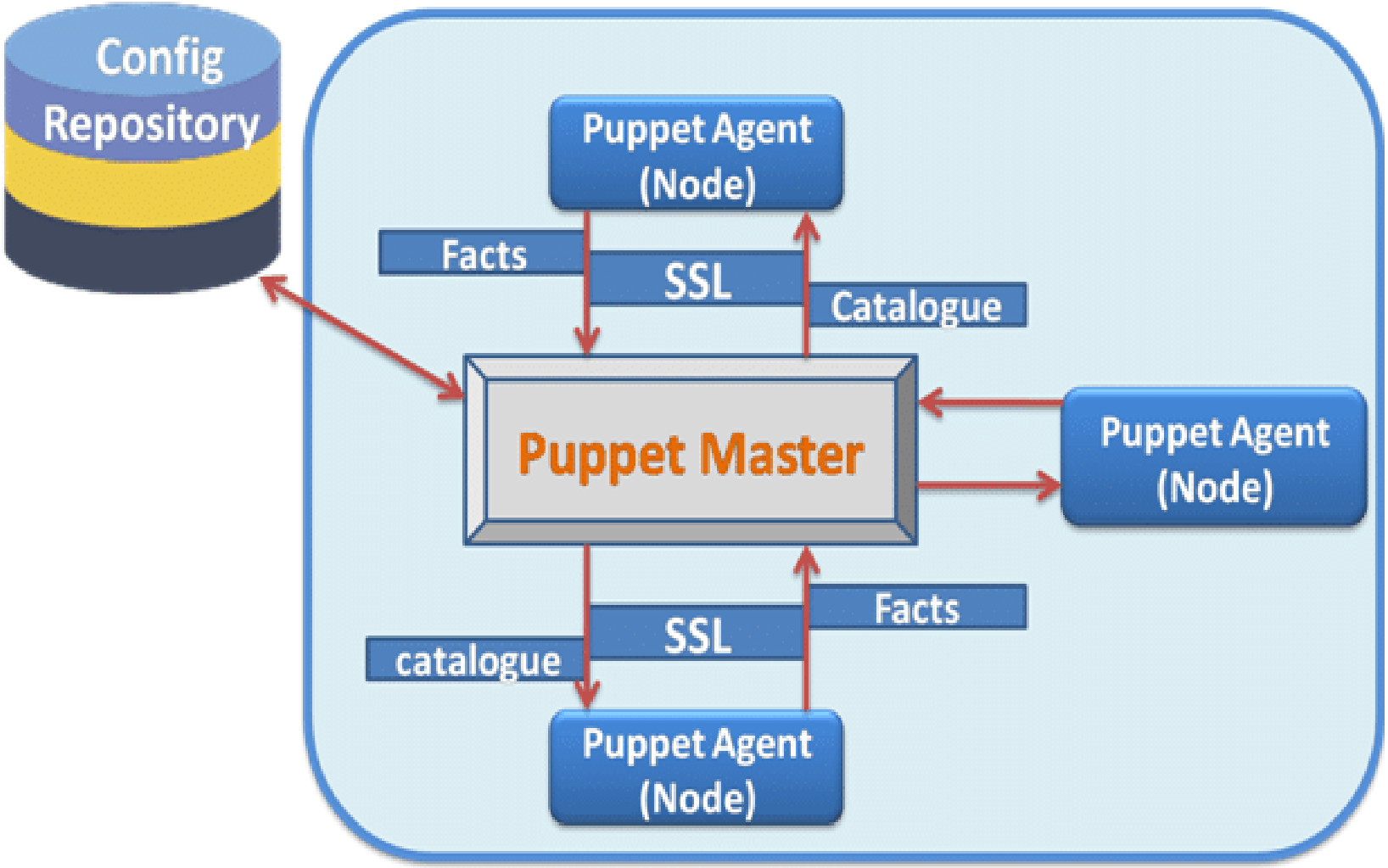
Push-based deployment model:

In this deployment model master server pushes the configurations and software to the individual agents. After verifying a secure connection, the master runs commands remotely on the agents. For example, Ansible and Salt Stack.

Pull-based deployment model.

In this deployment model, individual servers contact a master server, verify and establish a secure connection, download their configurations and software and then configure themselves accordingly — for example, Puppet and Chef.

Puppet architecture



Puppet architecture Description

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. On the basis of facts, changes are done on any target machine. There are pre-defined and custom facts in Puppet.

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.

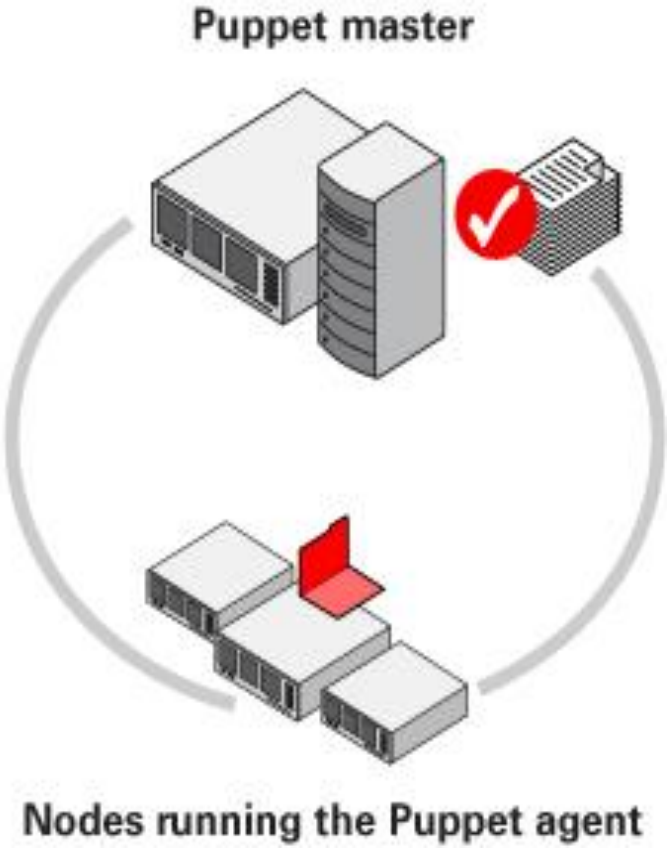
Puppet Architecture – SSL Connection

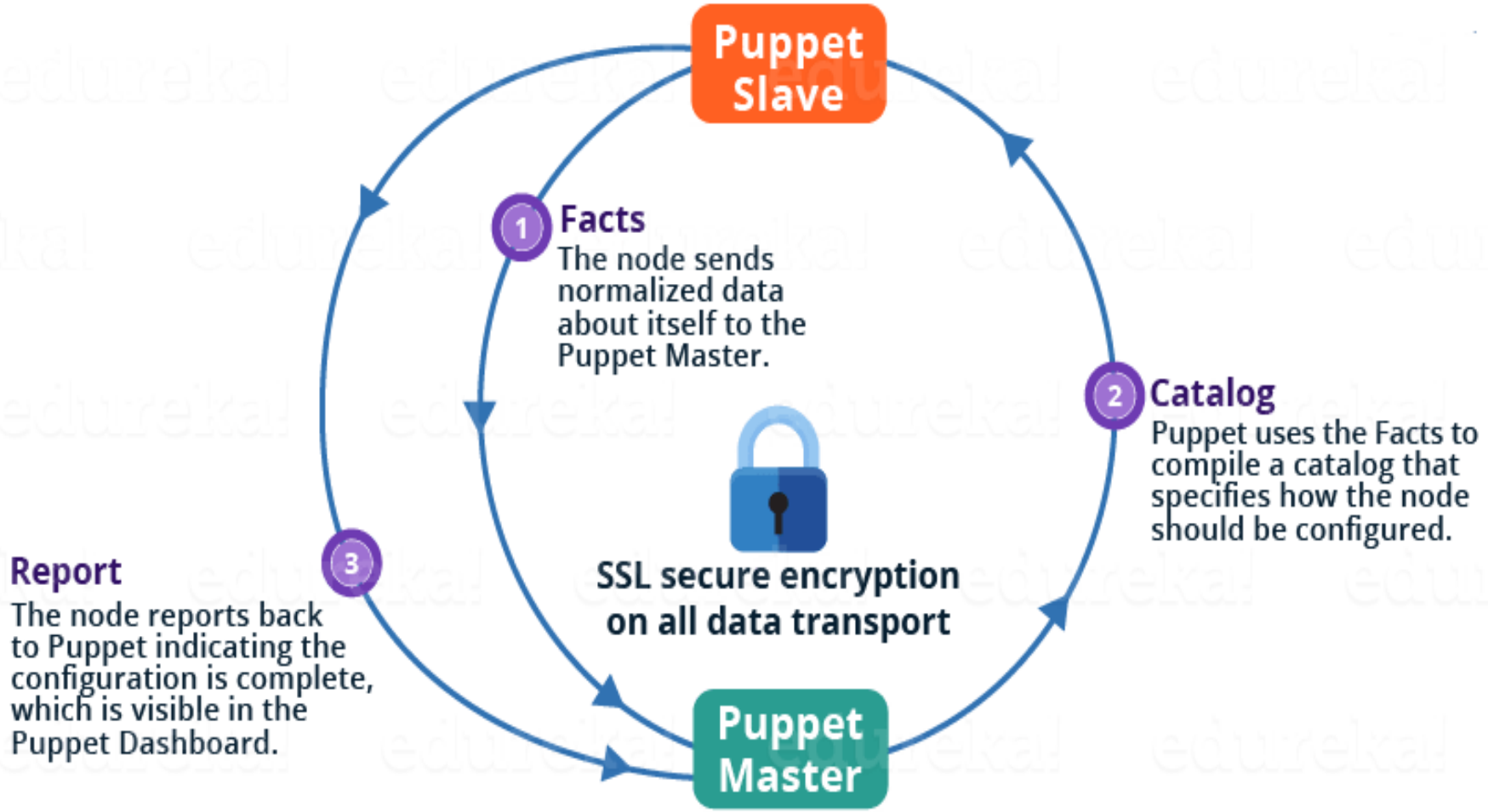
Because Puppet nodes have to interact with the master, all the information which is communicated between the master node and the slave nodes are encrypted using SSL certificates, The certificate signing process is as follows:

How Puppet works?

Puppet is based on a Pull deployment model, where the agent nodes check in regularly after every **1800** seconds with the master node to see if anything needs to be updated in the agent. If anything needs to be updated the agent pulls the necessary puppet codes from the master and performs required actions.

1. Node that is running the Puppet agent collects data about itself using facts
2. Agent sends facts to Puppet master
3. Master compiles a catalog based on data for how the node should be configured
4. Master sends catalog back to agent
5. Agent configures itself and reports back to master




Puppet Slave
1
Facts
The node sends normalized data about itself to the Puppet Master.
2
Catalog
Puppet uses the Facts to compile a catalog that specifies how the node should be configured.
SSL secure encryption on all data transport
3
Report
The node reports back to Puppet indicating the configuration is complete, which is visible in the Puppet Dashboard.
Puppet Master