

St. Francis Institute of Technology, Mumbai-400 103
Department Of Information Technology

A.Y. 2023-2024
Class: TE-ITA/B, Semester: V

Subject: **DevOps Lab**

**Experiment – 11: To learn Pull based Software Configuration
Management and provisioning tools using Puppet.**

1. **Aim:** To install and Configure Pull based Software Configuration Management and provisioning tools using Puppet.
2. **Objectives:** Aim of this experiment is that, the students will learn:
 - To Synthesize software configuration and provisioning using Puppet
 - To Build and operate a scalable automation system.
3. **Outcomes:** After study of this experiment, the students will learn following:
 - Architecture of Puppet
 - Puppet Master Slave Communication
 - Configuring Puppet Master and Agent on Linux machines
4. **Prerequisite:** None
5. **Requirements:** AWS account, putty, Personal Computer, Windows operating system, Internet Connection, Microsoft Word.
6. **Pre-Experiment Exercise:**

Brief Theory: Refer shared material

7. **Laboratory Exercise**

A. Procedure:

a. Answer the following:

- What is Puppet? Enlist its features.

Ans

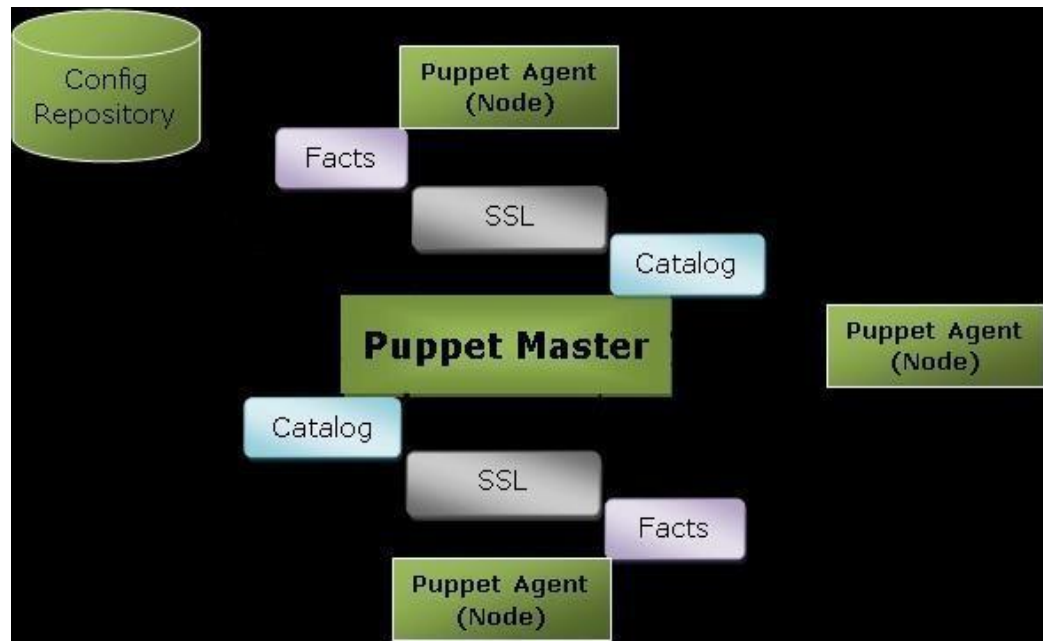
A puppet is a configuration management tool commonly used in system administration and IT operations to automate the provisioning, configuration, and management of infrastructure resources. Here are some key features of Puppet:

1. **Declarative Language:** Puppet uses a declarative language to define the desired state of a system. You specify what the system should look like, and Puppet ensures it reaches that state.
2. **Cross-Platform:** Puppet is platform-agnostic and can be used to manage resources across different operating systems, making it versatile for heterogeneous environments.
3. **Agent-Server Model:** Puppet follows a client-server architecture where Puppet agents run on managed nodes and communicate with a central Puppet master server to enforce configurations.
4. **Idempotence:** Puppet ensures that applying the same configuration multiple times has the same effect as applying it once. This minimizes unintended changes to the system.

5. **Resource Abstraction:** Puppet represents system resources (e.g., files, services, packages) as “resources” in its manifest files, allowing for consistent management across different systems.
 6. **Module-Based:** Configuration in Puppet is organized into reusable modules, which can be shared and applied to different systems, promoting code reusability.
 7. **Dependency Management:** Puppet manages the order in which resources are applied based on their dependencies, ensuring that configurations are applied correctly.
 8. **Reporting and Logging:** Puppet provides reporting and logging capabilities to track changes, detect discrepancies, and troubleshoot issues in the infrastructure.
 9. **Extensibility:** Puppet’s functionality can be extended using custom facts, functions, and resource types, allowing users to tailor it to their specific needs.
 10. **Community and Ecosystem:** Puppet has an active user community and a rich ecosystem of pre-built modules and resources available through the Puppet Forge.
 11. **Version Control Integration:** Puppet configurations can be managed using version control systems like Git, enabling collaboration, change tracking, and rollback capabilities.
 12. **Security:** Puppet supports encryption and secure communication between agents and the master server, helping protect sensitive data.
- Explain architecture of Puppet with a diagram.

Ans

Puppet uses master-slave or client-server architecture. Puppet client and server interconnected by SSL, which is a secure socket layer. It is a model-driven system.



Puppet Architecture

Puppet is a configuration management tool used for automating the provisioning and management of IT infrastructure. Its architecture consists of several key components:

Puppet Master:

The Puppet Master, or Puppet Server, is the central control node. It manages the configuration data and enforces it on managed nodes.

It hosts the Puppet Master daemon, which receives and compiles configurations, and serves them to Puppet Agents.

Puppet Agent:

Puppet Agents are the target systems or nodes that are being managed. These agents periodically contact the Puppet Master to request and apply configurations.

Puppet Agents run the Puppet agent daemon and apply configurations based on the instructions received from the Puppet Master.

Puppet DSL:

Puppet DSL (Domain-Specific Language) is a declarative language used to define the desired state of a system. Puppet manifests are written in this language and describe the configuration resources and relationships.

Catalog:

The Puppet Master compiles the configuration manifests into a catalog for each Puppet Agent. The catalog is a compiled representation of the desired system state.

Facter:

Facter is a system profiling tool that gathers information about the target nodes. Puppet Agents use this data to determine the system's current state.

PuppetDB (optional):

PuppetDB is a separate database that can be used to store facts and catalog data. It helps in querying and reporting on Puppet-managed infrastructure.

Hiera (optional):

Hiera is a key-value data lookup system that allows you to separate data from code in your Puppet manifests. It's useful for organizing and storing configuration data

b. Refer the shared material and do online research to answer following:

- Mention steps for creating 2 EC2 instances on AWS for creating master and slave machines. Attach screenshots for the same.
- Commands used to configure master and slave machines using putty
Commands to run on puppet Master (which is one of the EC2 instances)
- `sudo apt-get update` :This command will update the packages
- `wget https://apt.puppetlabs.com/puppet-release-bionic.deb` :This command will download the puppet folder
- `sudo dpkg -i puppet-release-bionic.deb` :This command will unzip the puppet folder
- `sudo apt-get install puppetmaster` : This command will install the puppet master
- `apt policy puppetmaster` : This command will verify puppet master after installation
- `sudo systemctl status puppet-master.service`: This command will check status of puppet master service
- `sudo nano /etc/default/puppet-master`: This command will fine tune some settings....
- Add this line in the puppet master file: `JAVA_ARGS="-Xms512m -Xmx512m"`
This command will change the memory allocation to 512MB
- `sudo systemctl restart puppet-master.service` : This command will restart puppet master after the recent changes
- `sudo ufw allow 8140/tcp` : This command will open TCP port for puppet to communicate
- `sudo nano /etc/hosts` : This command will open hosts file for entering master's IP address
- `sudo puppet cert list` : This command will show puppet agent's certificate received for signing
- `sudo puppet cert sign --all` : This command will sign the received certificate
Commands to run on slave node/ puppet agent (which is the other EC2 instance)
- `sudo apt-get update`: This command will update the packages

- `wget https://apt.puppetlabs.com/puppet-release-bionic.deb` This command will download the puppet folder
- `sudo dpkg -i puppet-release-bionic.deb` :This command will unzip the puppet folder
- `sudo apt-get install puppet` :This command will install the puppet agent
- `sudo nano /etc/hosts` :This command will open hosts file for entering master's IP address
- `sudo systemctl start puppet` :This command will start the puppet agent
- `sudo systemctl enable puppet` :This command will enable the puppet agent
- `sudo puppet agent --test` :This command will test communication between puppet master

8. Post-Experiments Exercise

A. Extended Theory:

Nil

B. Questions:

- Explain the two types of configuration management approaches.
- How does the connection between puppet master server and puppet agent nodes happen?

C. Conclusion:

- Write what was performed in the experiment.
- Write the significance of the topic studied in the experiment.

D. References:

<https://www.edureka.co/blog/puppet-tutorial/>

<https://www.simplilearn.com/puppet-tutorial-article>