**120A3051**

**Shreya Idate**

**Batch: E3**

**Experiment No: 11**

**AIM**: To install and configure Pull based Software Configuration Management and provisioning tools using Puppet.

**THEORY**: Puppet is a tool that helps you manage and automate the configuration of servers.

When you use Puppet, you define the desired state of the systems in your infrastructure that you want to manage. You do this by writing infrastructure code in Puppet's Domain-Specific Language (DSL) — Puppet Code — which you can use with a wide array of devices and operating systems. Puppet code is declarative, which means that you describe the desired state of your systems, not the steps needed to get there. Puppet then automates the process of getting these systems into that state and keeping them there. Puppet does this through Puppet primary server and a Puppet agent. The Puppet primary server is the server that stores the code that defines your desired state. The Puppet agent translates your code into commands and then executes it on the systems you specify, in what is called a Puppet run.

# Puppet Master

Puppet master handles all the configuration related process in the form of puppet codes. It is a Linux based system in which puppet master software is installed. The puppet master must be in Linux. It usesthe puppet agent to apply the configuration to nodes.

This is the place where SSL certificates are checked and marked.

# Puppet Slave or Agent

Puppet agents are the real working systems and used by the Client. It is installed on the client machine and maintained and managed by the puppet master. They have a puppet agent service running inside them.

The agent machine can be configured on any operating system such as Windows, Linux, Solaris, orMac OS. Config Repository

Config repository is the storage area where all the servers and nodes related configurations are stored, and we can pull these configurations as per requirements.

# Facts

Facts are the key-value data pair. It contains information about the node or the master machine. It represents a puppet client states such as operating system, network interface, IP address, uptime, and whether the client

machine is virtual or not.

These facts are used for determining the present state of any agent. Changes on any target machine are made based on facts. Puppet's facts are predefined and customized.

Catalog

The entire configuration and manifest files that are written in Puppet are changed into a compiled format. This compiled format is known as a catalog, and then we can apply this catalog to the target machine.

The above image performs the following functions:

* First of all, an agent node sends facts to the master or server and requests for a catalog.
* The master or server compiles and returns the catalog of a node with the help of some information accessed by the master.
* Then the agent applies the catalog to the node by checking every resource mentioned in the catalog. If it identifies resources that are not in their desired state, then makes the necessary adjustments to fix them. Or, it determines in no-op mode, the adjustments would be required toreconcile the catalog.
* And finally, the agent sends a report back to the master.

Puppet Master-Slave Communication

Puppet master-slave communicates via a secure encrypted channel through the SSL (Secure Socket Layer). Let's see the below diagram to understand the communication between the master and slave with this channel:

# Benefits of Puppet:

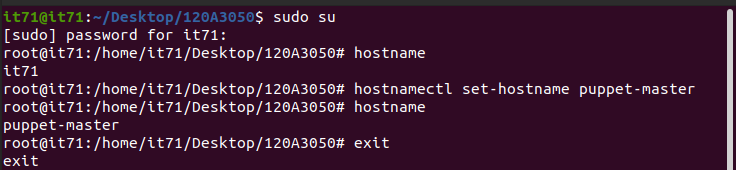
There are many benefits to implementing a declarative configuration tool like Puppet into your environment — most notably consistency and automation.

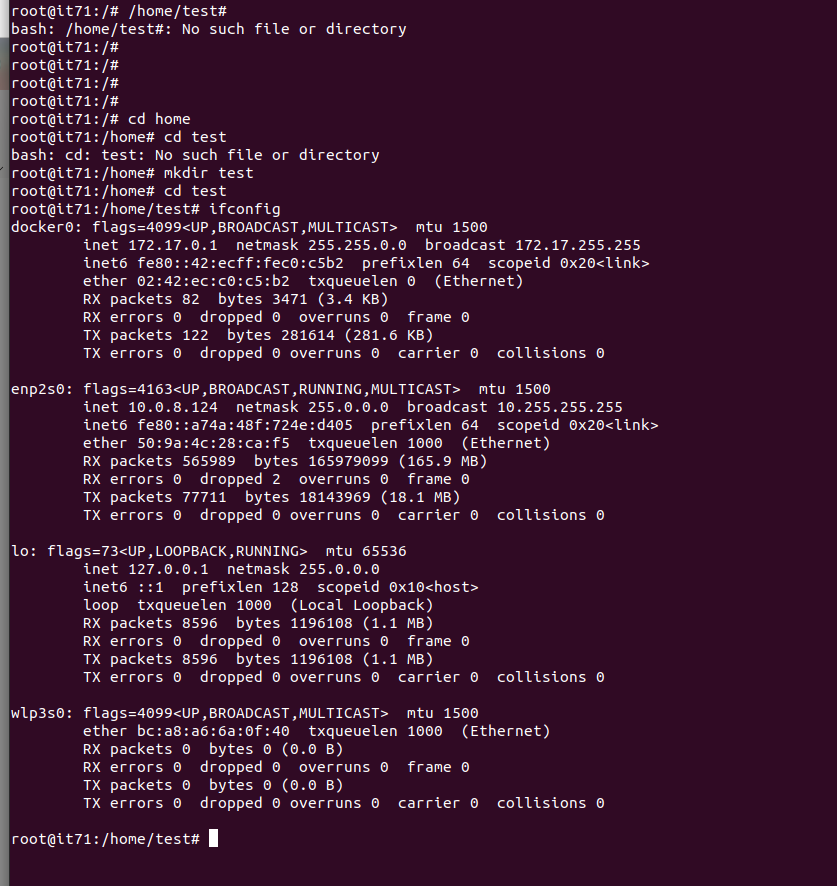
* Consistency - When you use configuration management, you are able to validate that Puppet applied the desired state you wanted. You can then assume that state has been applied, helping you to identify why your model failed and what was incomplete, and saving you valuable time in the process. Most importantly, once you figure it out, you can add the missing part to your model and ensure that you never have to deal with that same problem again.
* Automation - When you manage a set of servers in your infrastructure, you want to keep them in a certain state. If you only have to manage homogeneous 10 servers, you can do so with a script or by manually going into each server. In this case, a tool like Puppet may not provide much extra value. But if you have 100 or 1,000 servers, a mixed environment, or you have plans to scale your infrastructure in the future, it is difficult to do this manually. This is where Puppet can help you — to save you time and money, to scale effectively, and to do so securely.
* Open source
* Translate if required
* Cross platform
* Allows resource abstraction
* Clean and easily learnable

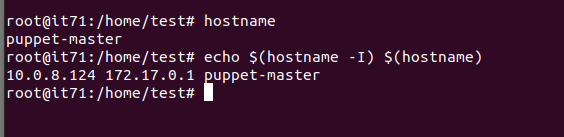
# Master

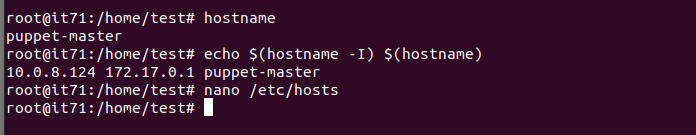
# Installation of Puppet Master:

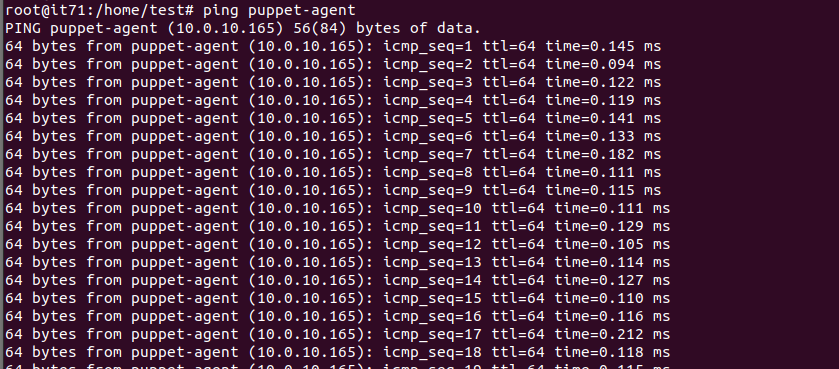
1. **Setting hostname of Master :**





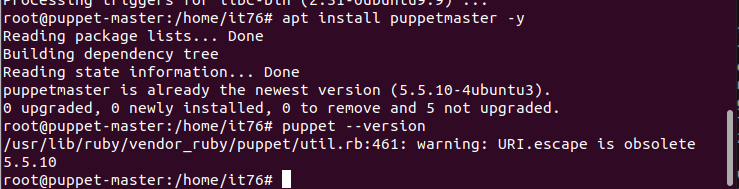


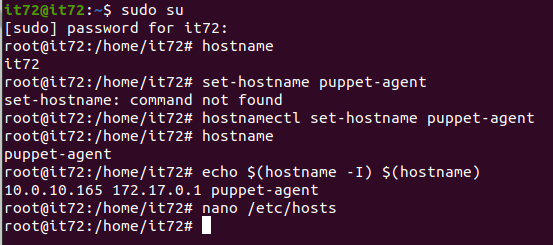
****

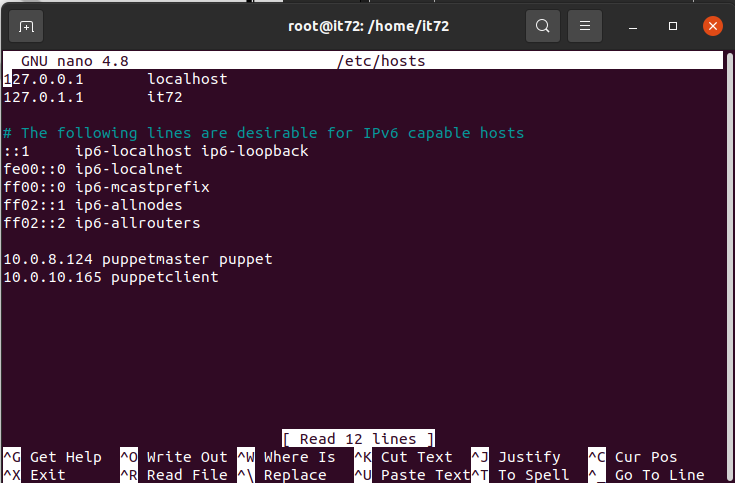


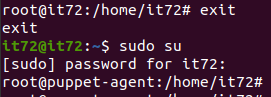
**Agent:**

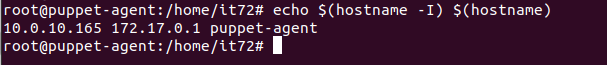
**Installation of Puppet**

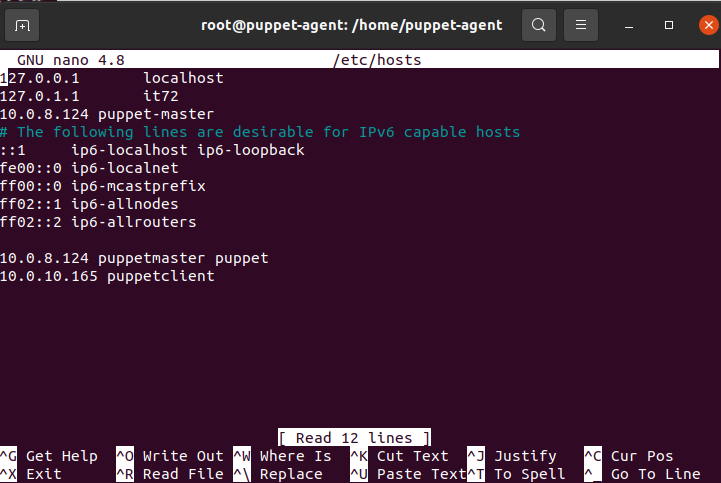


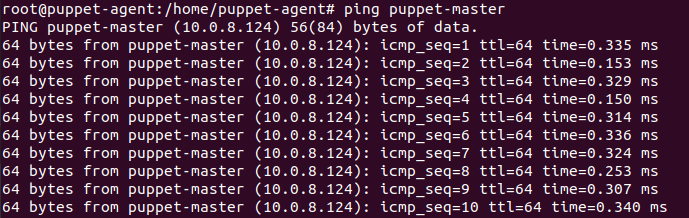
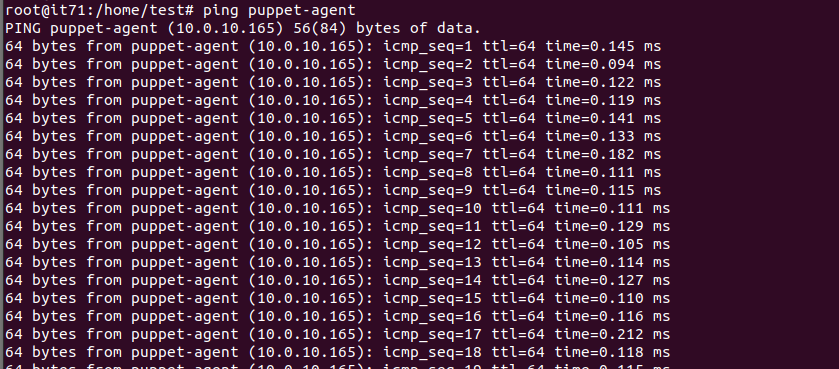










**Ping puppet master** **Ping Puppet Agent**

# Conclusion: In this experiment we learnt to install and configure Pull based Software Configuration Management and provisioning tools using Puppet.