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 executesit 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 agentto 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 andmaintained 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 wecan 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 apuppet 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 madebased 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. Thiscompiled 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'ssee 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 :

```
it71@it71:~/Desktop/120A3050$ sudo su
[sudo] password for it71:
root@it71:/home/it71/Desktop/120A3050# hostname
it71
root@it71:/home/it71/Desktop/120A3050# hostnamectl set-hostname puppet-master
root@it71:/home/it71/Desktop/120A3050# hostname
puppet-master
root@it71:/home/it71/Desktop/120A3050# exit
exit
```

```
root@it71:/# /home/test#
bash: /home/test#: No such file or directory
root@it71:/#
root@it71:/#
root@it71:/#
root@it71:/#
root@it71:/# cd home
root@it71:/home# cd test
bash: cd: test: No such file or directory
root@it71:/home# mkdir test
root@it71:/home# cd test
root@it71:/home/test# ifconfig
docker0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 172.17.0.1  netmask 255.255.0.0  broadcast 172.17.255.255
        inet6 fe80::42:ecff:fec0:c5b2  prefixlen 64  scopeid 0x20<link>
        ether 02:42:ec:c0:c5:b2  txqueuelen 0  (Ethernet)
        RX packets 82  bytes 3471 (3.4 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 122  bytes 281614 (281.6 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp2s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.8.124  netmask 255.0.0.0  broadcast 10.255.255.255
        inet6 fe80::a74a:48f:724e:d405  prefixlen 64  scopeid 0x20<link>
        ether 50:9a:4c:28:ca:f5  txqueuelen 1000  (Ethernet)
        RX packets 565989  bytes 165979099 (165.9 MB)
        RX errors 0  dropped 2  overruns 0  frame 0
        TX packets 77711  bytes 18143969 (18.1 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 8596  bytes 1196108 (1.1 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 8596  bytes 1196108 (1.1 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlp3s0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether bc:a8:a6:6a:0f:40  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@it71:/home/test#
```

```
root@it71:/home/test# hostname
puppet-master
root@it71:/home/test# echo $(hostname -I) $(hostname)
10.0.8.124 172.17.0.1 puppet-master
root@it71:/home/test#
```

```
root@it71:/home/test# hostname
puppet-master
root@it71:/home/test# echo $(hostname -I) $(hostname)
10.0.8.124 172.17.0.1 puppet-master
root@it71:/home/test# nano /etc/hosts
root@it71:/home/test#
```

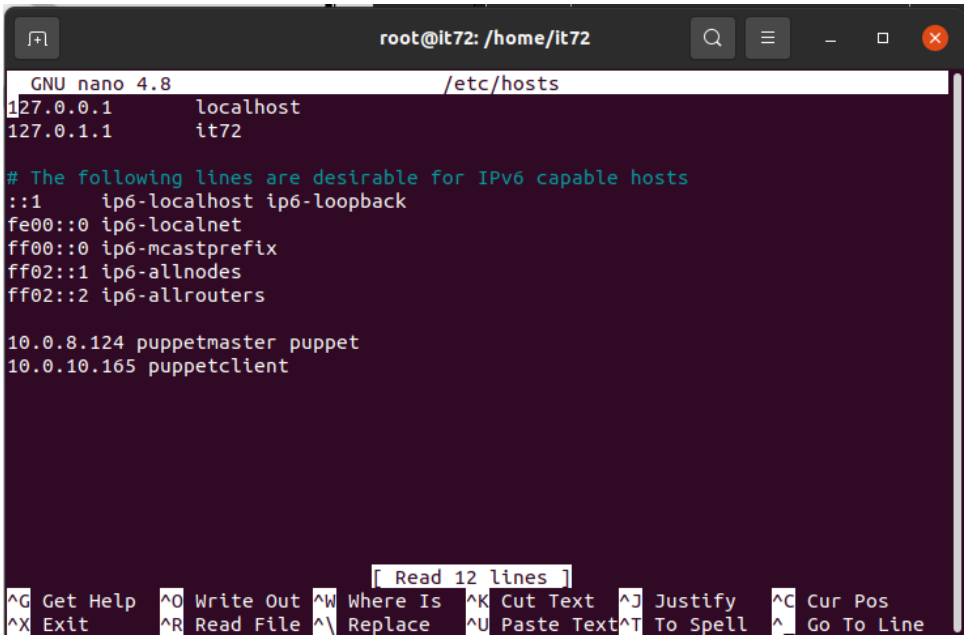
```
root@it71:/home/test# ping puppet-agent
PING puppet-agent (10.0.10.165) 56(84) bytes of data.
64 bytes from puppet-agent (10.0.10.165): icmp_seq=1 ttl=64 time=0.145 ms
64 bytes from puppet-agent (10.0.10.165): icmp_seq=2 ttl=64 time=0.094 ms
64 bytes from puppet-agent (10.0.10.165): icmp_seq=3 ttl=64 time=0.122 ms
64 bytes from puppet-agent (10.0.10.165): icmp_seq=4 ttl=64 time=0.119 ms
64 bytes from puppet-agent (10.0.10.165): icmp_seq=5 ttl=64 time=0.141 ms
64 bytes from puppet-agent (10.0.10.165): icmp_seq=6 ttl=64 time=0.133 ms
64 bytes from puppet-agent (10.0.10.165): icmp_seq=7 ttl=64 time=0.182 ms
64 bytes from puppet-agent (10.0.10.165): icmp_seq=8 ttl=64 time=0.111 ms
64 bytes from puppet-agent (10.0.10.165): icmp_seq=9 ttl=64 time=0.115 ms
64 bytes from puppet-agent (10.0.10.165): icmp_seq=10 ttl=64 time=0.111 ms
64 bytes from puppet-agent (10.0.10.165): icmp_seq=11 ttl=64 time=0.129 ms
64 bytes from puppet-agent (10.0.10.165): icmp_seq=12 ttl=64 time=0.105 ms
64 bytes from puppet-agent (10.0.10.165): icmp_seq=13 ttl=64 time=0.114 ms
64 bytes from puppet-agent (10.0.10.165): icmp_seq=14 ttl=64 time=0.127 ms
64 bytes from puppet-agent (10.0.10.165): icmp_seq=15 ttl=64 time=0.110 ms
64 bytes from puppet-agent (10.0.10.165): icmp_seq=16 ttl=64 time=0.116 ms
64 bytes from puppet-agent (10.0.10.165): icmp_seq=17 ttl=64 time=0.212 ms
64 bytes from puppet-agent (10.0.10.165): icmp_seq=18 ttl=64 time=0.118 ms
```

Agent:

Installation of Puppet

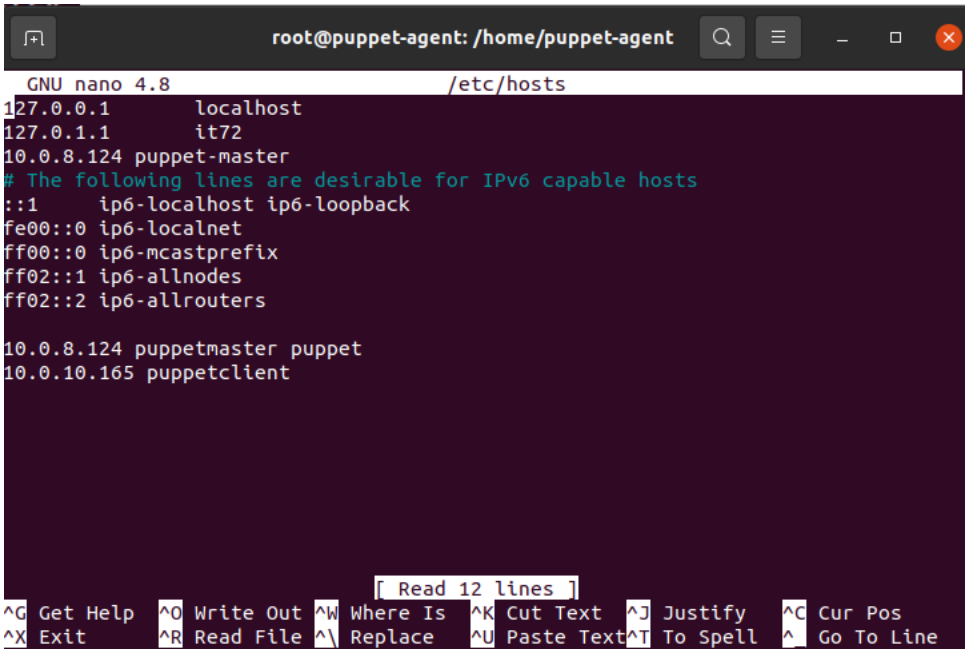
```
root@puppet-master:/home/it76# apt install puppetmaster -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
puppetmaster is already the newest version (5.5.10-4ubuntu3).
0 upgraded, 0 newly installed, 0 to remove and 5 not upgraded.
root@puppet-master:/home/it76# puppet --version
/usr/lib/ruby/vendor_ruby/puppet/util.rb:461: warning: URI.escape is obsolete
5.5.10
root@puppet-master:/home/it76#
```

```
it72@it72:~$ sudo su
[sudo] password for it72:
root@it72:/home/it72# hostname
it72
root@it72:/home/it72# set-hostname puppet-agent
set-hostname: command not found
root@it72:/home/it72# hostnamectl set-hostname puppet-agent
root@it72:/home/it72# hostname
puppet-agent
root@it72:/home/it72# echo $(hostname -I) $(hostname)
10.0.10.165 172.17.0.1 puppet-agent
root@it72:/home/it72# nano /etc/hosts
root@it72:/home/it72#
```




```
root@it72:/home/it72# exit
exit
it72@it72:~$ sudo su
[sudo] password for it72:
root@puppet-agent:/home/it72#
```

```
root@puppet-agent:/home/it72# echo $(hostname -I) $(hostname)
10.0.10.165 172.17.0.1 puppet-agent
root@puppet-agent:/home/it72#
```



Ping puppet master

```
root@puppet-agent:/home/puppet-agent# ping puppet-master
PING puppet-master (10.0.8.124) 56(84) bytes of data.
64 bytes from puppet-master (10.0.8.124): icmp_seq=1 ttl=64 time=0.335 ms
64 bytes from puppet-master (10.0.8.124): icmp_seq=2 ttl=64 time=0.153 ms
64 bytes from puppet-master (10.0.8.124): icmp_seq=3 ttl=64 time=0.329 ms
64 bytes from puppet-master (10.0.8.124): icmp_seq=4 ttl=64 time=0.150 ms
64 bytes from puppet-master (10.0.8.124): icmp_seq=5 ttl=64 time=0.314 ms
64 bytes from puppet-master (10.0.8.124): icmp_seq=6 ttl=64 time=0.336 ms
64 bytes from puppet-master (10.0.8.124): icmp_seq=7 ttl=64 time=0.324 ms
64 bytes from puppet-master (10.0.8.124): icmp_seq=8 ttl=64 time=0.253 ms
64 bytes from puppet-master (10.0.8.124): icmp_seq=9 ttl=64 time=0.307 ms
64 bytes from puppet-master (10.0.8.124): icmp_seq=10 ttl=64 time=0.340 ms
```

Ping Puppet Agent

```
root@it71:/home/test# ping puppet-agent
PING puppet-agent (10.0.10.165) 56(84) bytes of data.
64 bytes from puppet-agent (10.0.10.165): icmp_seq=1 ttl=64 time=0.145 ms
64 bytes from puppet-agent (10.0.10.165): icmp_seq=2 ttl=64 time=0.094 ms
64 bytes from puppet-agent (10.0.10.165): icmp_seq=3 ttl=64 time=0.122 ms
64 bytes from puppet-agent (10.0.10.165): icmp_seq=4 ttl=64 time=0.119 ms
64 bytes from puppet-agent (10.0.10.165): icmp_seq=5 ttl=64 time=0.141 ms
64 bytes from puppet-agent (10.0.10.165): icmp_seq=6 ttl=64 time=0.133 ms
64 bytes from puppet-agent (10.0.10.165): icmp_seq=7 ttl=64 time=0.182 ms
64 bytes from puppet-agent (10.0.10.165): icmp_seq=8 ttl=64 time=0.111 ms
64 bytes from puppet-agent (10.0.10.165): icmp_seq=9 ttl=64 time=0.115 ms
64 bytes from puppet-agent (10.0.10.165): icmp_seq=10 ttl=64 time=0.111 ms
64 bytes from puppet-agent (10.0.10.165): icmp_seq=11 ttl=64 time=0.129 ms
64 bytes from puppet-agent (10.0.10.165): icmp_seq=12 ttl=64 time=0.105 ms
64 bytes from puppet-agent (10.0.10.165): icmp_seq=13 ttl=64 time=0.114 ms
64 bytes from puppet-agent (10.0.10.165): icmp_seq=14 ttl=64 time=0.127 ms
64 bytes from puppet-agent (10.0.10.165): icmp_seq=15 ttl=64 time=0.110 ms
64 bytes from puppet-agent (10.0.10.165): icmp_seq=16 ttl=64 time=0.116 ms
64 bytes from puppet-agent (10.0.10.165): icmp_seq=17 ttl=64 time=0.212 ms
64 bytes from puppet-agent (10.0.10.165): icmp_seq=18 ttl=64 time=0.118 ms
```

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