

Experiment 12 - Puppet Setup

Roll No.	24
Name	Iyer Sreekesh Subramanian
Class	D15-A
Subject	DevOps Lab
LO Mapped	<p>LO1: To understand the fundamentals of DevOps engineering and be fully proficient with DevOps terminologies, concepts, benefits, and deployment options to meet your business requirements</p> <p>LO6: To Synthesize software configuration and provisioning using Ansible/Puppet.</p>

Aim: To install and Configure Pull based Software Configuration Management and provisioning tools using Puppet.

Introduction:

What is Puppet?

Puppet is a Configuration Management tool that is used for deploying, configuring and managing servers. It performs the following functions:

- Defining distinct configurations for each and every host, and continuously checking and confirming whether the required configuration is in place and is not altered (if altered Puppet will revert back to the required configuration) on the host.
- Dynamic scaling-up and scaling-down of machines.
- Providing control over all your configured machines, so a centralized (master-server or repo-based) change gets propagated to all, automatically.

Puppet uses a Master-Slave architecture in which the Master and Slave communicate through a secure encrypted channel with the help of SSL.

Puppet Key Metrics

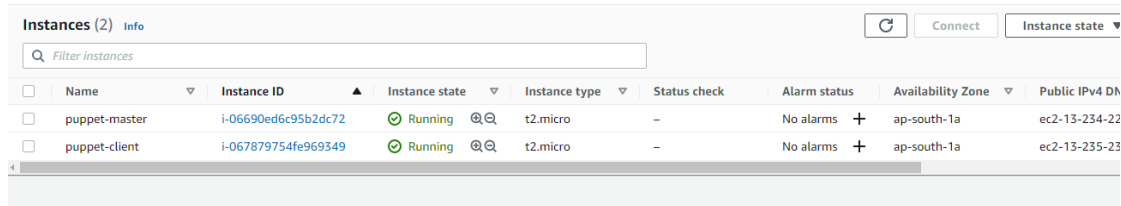
- **Large installed base:** Puppet is used by more than 30,000 companies worldwide including Google, Red Hat, Siemens, etc. along with several universities like Stanford and Harvard law school. An average of 22 new organizations per day uses Puppet for the first time.
- **Large developer base:** Puppet is so widely used that lots of people develop for it. Puppet has many contributors to its core source code.
- **Long commercial track record:** Puppet has been in commercial use since 2005, and has been continually refined and improved. It has been deployed in very large infrastructures (5,000+ machines) and the performance and scalability lessons learned from these projects have contributed to Puppet's development.
- **Documentation:** Puppet has a large user-maintained wiki with hundreds of pages of documentation and comprehensive references for both the language and its resource types. In addition, it's actively discussed on several mailing lists and has a very popular IRC channel, so whatever your Puppet problem, it's easy to find the answer.
- **Platform support:** Puppet Server can run on any platform that supports ruby for ex: CentOS, Microsoft Windows Server, Oracle Enterprise Linux etc. It not only supports the new operating systems but it can also run on relatively old and out-of-date OS and Ruby versions as well.

Installation:

Prerequisites:

→ 2 Linux Machines on AWS Ec2

Create 2 Ubuntu 20.04 EC2 instances on AWS and take remote access to them using SSH.



Instances (2) Info									
Filter instances									
<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4	DN
<input type="checkbox"/>	puppet-master	i-06690ed6c95b2dc72	Running	t2.micro	-	No alarms	ap-south-1a	ec2-13-234-22	
<input type="checkbox"/>	puppet-client	i-067879754fe969349	Running	t2.micro	-	No alarms	ap-south-1a	ec2-13-235-23	

Steps:

Follow instructions only on the mentioned machine(s)

1. On both master and client, Update Package List.

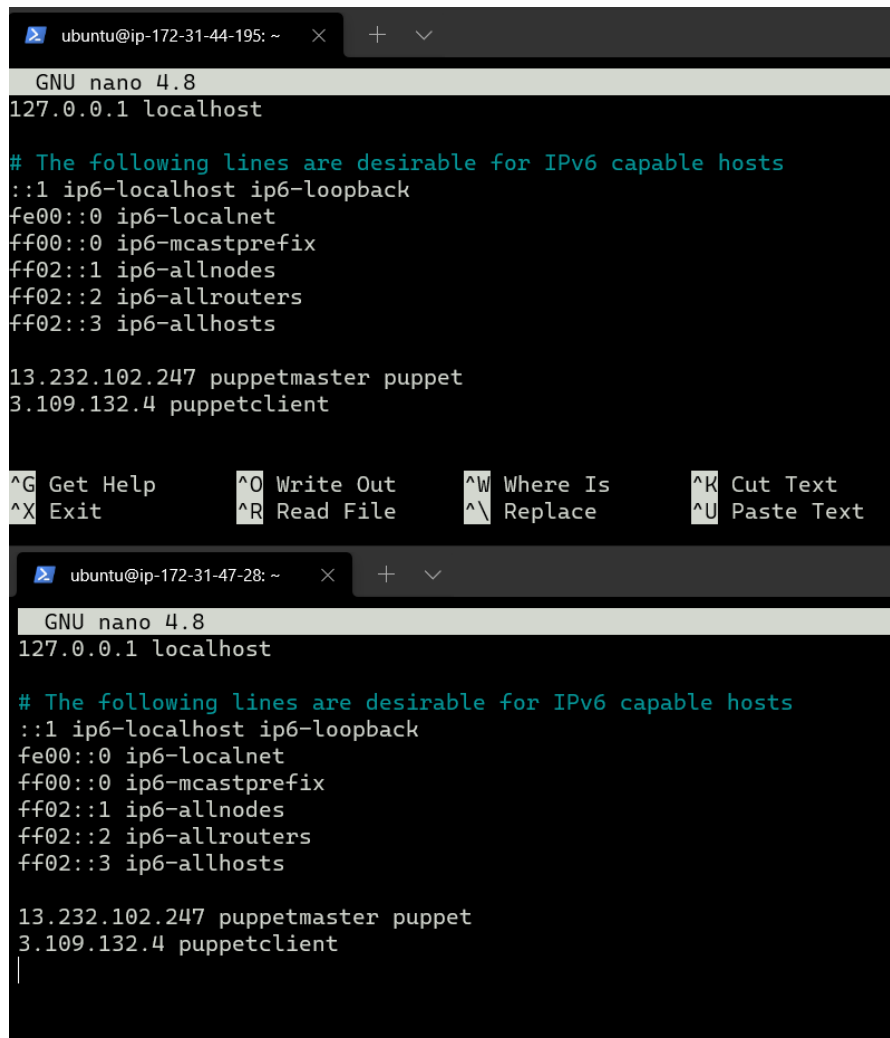
```
sudo apt-get update -y
```

2. On both master and client, set up the hostname resolution.

```
sudo nano /etc/hosts
```

Now, paste the following lines at the end of each file. These IP addresses are the Public IPv4 addresses of your instances. puppetmaster is your server and puppetclient is the client.

```
[puppet master ip] puppetmaster puppet
[puppet client ip] puppetclient
```



The image shows two terminal windows side-by-side, both running the nano 4.8 editor. The top window's title bar is 'ubuntu@ip-172-31-44-195: ~' and the bottom window's is 'ubuntu@ip-172-31-47-28: ~'. Both windows display the same text in the editor:

```
GNU nano 4.8
127.0.0.1 localhost

# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts

13.232.102.247 puppetmaster puppet
3.109.132.4 puppetclient
```

The bottom window shows an additional line at the end of the file, which is a blank line.

Follow these instructions only on the master machine.

3. Download the latest Puppet Version

```
wget https://apt.puppetlabs.com/puppet6-release-focal.deb
```

4. Once the download is complete, install the package by using dpkg

```
sudo dpkg -i puppet6-release-focal.deb
```

5. Update the package repository:

```
sudo apt-get update -y
```

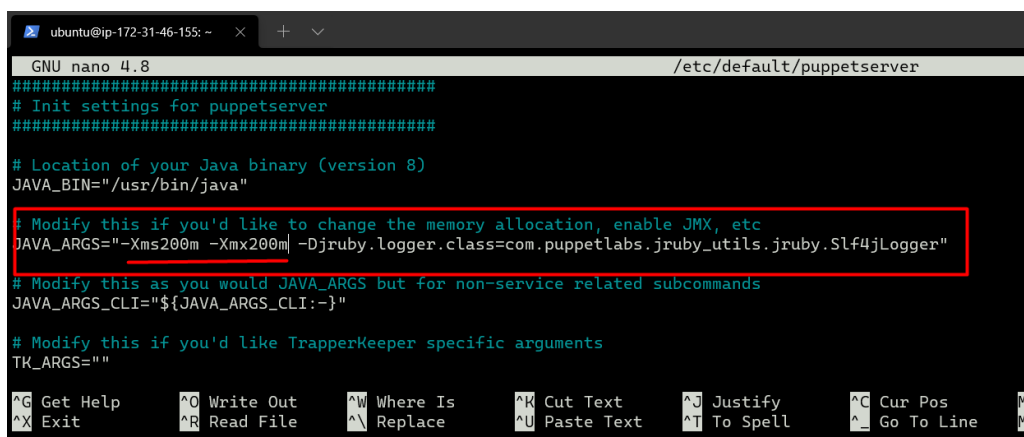
6. Install the puppet server.

```
sudo apt-get install puppetserver -y
```

Configuration:

7. Open the puppetserver file, under which you need to change the memory size from 2GB to 300MB since we don't want it to overload our instance which has only 1 GB of RAM.

```
sudo nano /etc/default/puppetserver
```



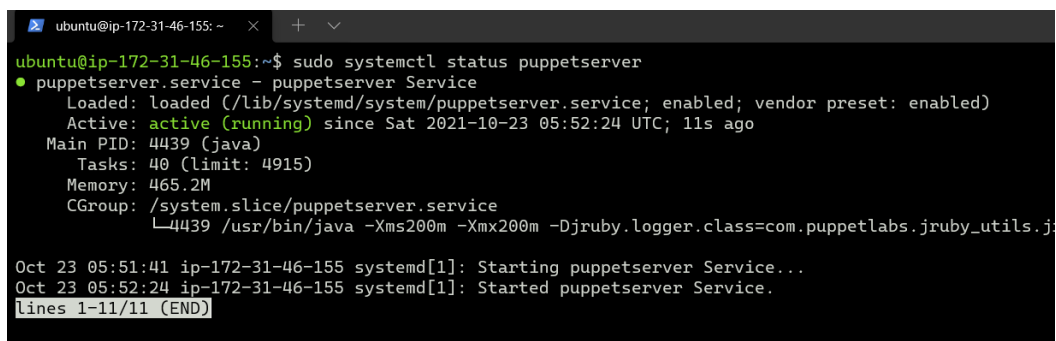
```
ubuntu@ip-172-31-46-155: ~  
GNU nano 4.8 /etc/default/puppetserver  
#####  
# Init settings for puppetserver  
#####  
# Location of your Java binary (version 8)  
JAVA_BIN="/usr/bin/java"  
  
# Modify this if you'd like to change the memory allocation, enable JMX, etc  
JAVA_ARGS="-Xms200m -Xmx200m -Djruby.logger.class=com.puppetlabs.jruby_utils.jruby.Slf4jLogger"  
# Modify this as you would JAVA_ARGS but for non-service related subcommands  
JAVA_ARGS_CLI="${JAVA_ARGS_CLI:-}"  
# Modify this if you'd like TrapperKeeper specific arguments  
TK_ARGS=""  
  
^G Get Help    ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos  
^X Exit        ^R Read File  ^\ Replace    ^U Paste Text ^T To Spell   ^_ Go To Line
```

8. Start and enable the Puppet Service

```
sudo systemctl restart puppetserver  
sudo systemctl enable puppetserver
```

9. Verify the status of the service.

```
sudo systemctl status puppetserver
```



```
ubuntu@ip-172-31-46-155: ~  
ubuntu@ip-172-31-46-155:~$ sudo systemctl status puppetserver  
● puppetserver.service - puppetserver Service  
   Loaded: loaded (/lib/systemd/system/puppetserver.service; enabled; vendor preset: enabled)  
   Active: active (running) since Sat 2021-10-23 05:52:24 UTC; 11s ago  
     Main PID: 4439 (java)  
       Tasks: 40 (limit: 4915)  
      Memory: 465.2M  
     CGroup: /system.slice/puppetserver.service  
             └─4439 /usr/bin/java -Xms200m -Xmx200m -Djruby.logger.class=com.puppetlabs.jruby_utils.j  
  
Oct 23 05:51:41 ip-172-31-46-155 systemd[1]: Starting puppetserver Service...  
Oct 23 05:52:24 ip-172-31-46-155 systemd[1]: Started puppetserver Service.  
lines 1-11/11 (END)
```

Now that our server is up and running, let's move to the client machine.
Perform these commands only on the **client** machine.

10. Download the latest version of Puppet on the client.

```
wget https://apt.puppetlabs.com/puppet6-release-focal.deb
```

11. Once the download is complete, use dpkg to install the package.

```
sudo dpkg -i puppet6-release-focal.deb
```

12. Update the package repository again.

```
sudo apt-get update -y
```

13. Install the agent using -

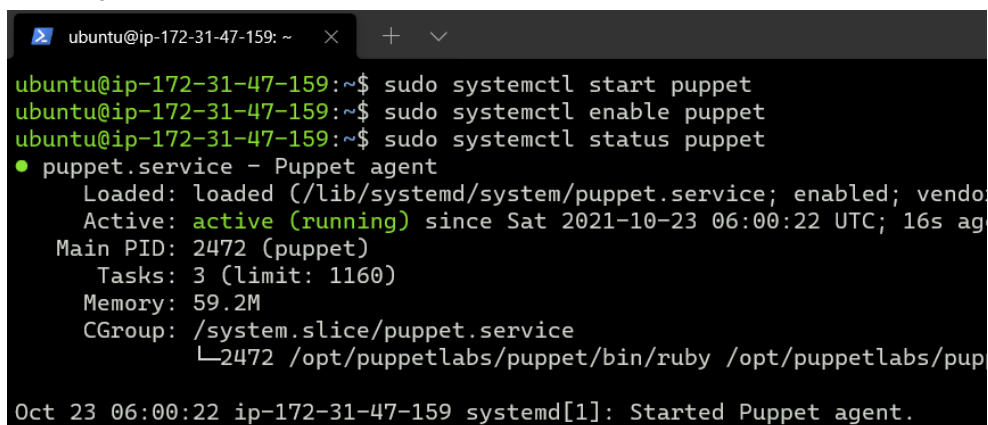
```
sudo apt-get install puppet-agent -y
```

14. Start the puppet service and enable it -

```
sudo systemctl start puppet  
sudo systemctl enable puppet
```

15. Check Puppet Service status

```
sudo systemctl status puppet
```

A terminal window screenshot showing the execution of Puppet-related commands. The prompt is 'ubuntu@ip-172-31-47-159: ~'. The commands and their outputs are: 'sudo systemctl start puppet', 'sudo systemctl enable puppet', and 'sudo systemctl status puppet'. The status output shows 'puppet.service - Puppet agent' is loaded, active (running), and has a main PID of 2472. It also shows tasks, memory usage, and CGroup information. At the bottom, a log message states: 'Oct 23 06:00:22 ip-172-31-47-159 systemd[1]: Started Puppet agent.'

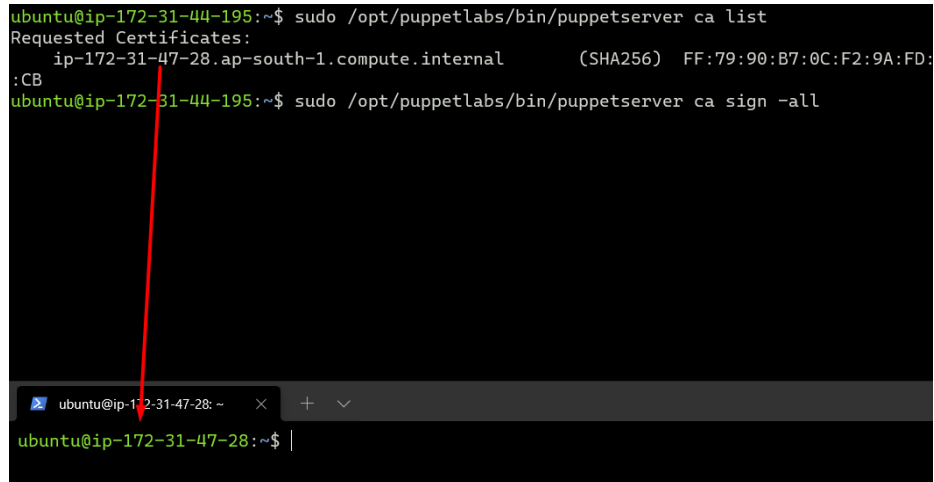
```
ubuntu@ip-172-31-47-159: ~  
ubuntu@ip-172-31-47-159:~$ sudo systemctl start puppet  
ubuntu@ip-172-31-47-159:~$ sudo systemctl enable puppet  
ubuntu@ip-172-31-47-159:~$ sudo systemctl status puppet  
● puppet.service - Puppet agent  
   Loaded: loaded (/lib/systemd/system/puppet.service; enabled; vendor preset: enabled)  
   Active: active (running) since Sat 2021-10-23 06:00:22 UTC; 16s ago  
     Main PID: 2472 (puppet)  
        Tasks: 3 (limit: 1160)  
       Memory: 59.2M  
      CGroup: /system.slice/puppet.service  
              └─2472 /opt/puppetlabs/puppet/bin/ruby /opt/puppetlabs/puppet/bin/puppetd  
  
Oct 23 06:00:22 ip-172-31-47-159 systemd[1]: Started Puppet agent.
```

Now that our server and puppet agent both are running, we can proceed to sign the Agent Certificate.

Perform these commands only on the master machine.

16. On the master node, list requested certificates

```
sudo /opt/puppetlabs/bin/puppetserver ca list
```

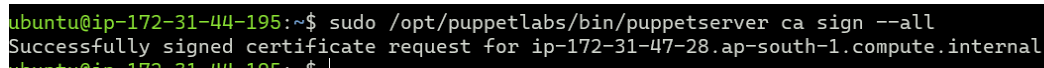


```
ubuntu@ip-172-31-44-195:~$ sudo /opt/puppetlabs/bin/puppetserver ca list
Requested Certificates:
ip-172-31-47-28.ap-south-1.compute.internal      (SHA256) FF:79:90:B7:0C:F2:9A:FD:
:CB
ubuntu@ip-172-31-44-195:~$ sudo /opt/puppetlabs/bin/puppetserver ca sign --all

ubuntu@ip-172-31-47-28: ~
```

17. Once you spot the requested certificate from the client-side, you can sign it using -

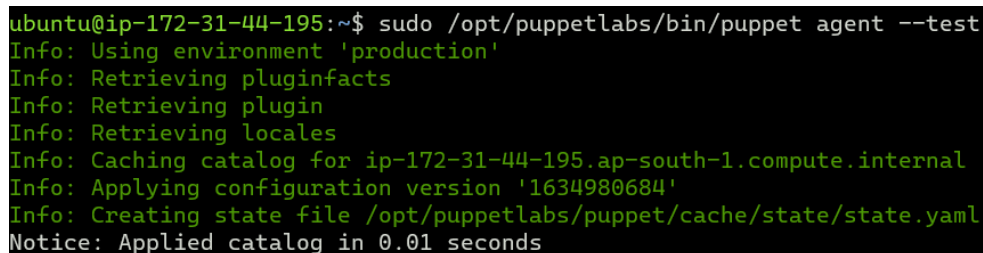
```
sudo /opt/puppetlabs/bin/puppetserver ca sign --all
```



```
ubuntu@ip-172-31-44-195:~$ sudo /opt/puppetlabs/bin/puppetserver ca sign --all
Successfully signed certificate request for ip-172-31-47-28.ap-south-1.compute.internal
ubuntu@ip-172-31-44-195:~$
```

18. To test the connection, you can use the following command

```
sudo /opt/puppetlabs/bin/puppet agent --test
```



```
ubuntu@ip-172-31-44-195:~$ sudo /opt/puppetlabs/bin/puppet agent --test
Info: Using environment 'production'
Info: Retrieving pluginfacts
Info: Retrieving plugin
Info: Retrieving locales
Info: Caching catalog for ip-172-31-44-195.ap-south-1.compute.internal
Info: Applying configuration version '1634980684'
Info: Creating state file /opt/puppetlabs/puppet/cache/state/state.yaml
Notice: Applied catalog in 0.01 seconds
```

If all went good, you'll get the notice of applying catalog, which means that puppet was configured properly on both machines.

Conclusion

Thus, we studied puppet, installed it on our machines and set up a master-worker cluster on it.