SIES Graduate School of Technology
Sri Chandrasekarendra Saraswati Vidyapuram
Sector v, Nerul, Navimumbai-400706

Name : Kunal Rajesh Kumbhare
Prn : 120A3024
Branch : IT
3rd Year (5th semester)

EXPERIMENT-11

AIM:

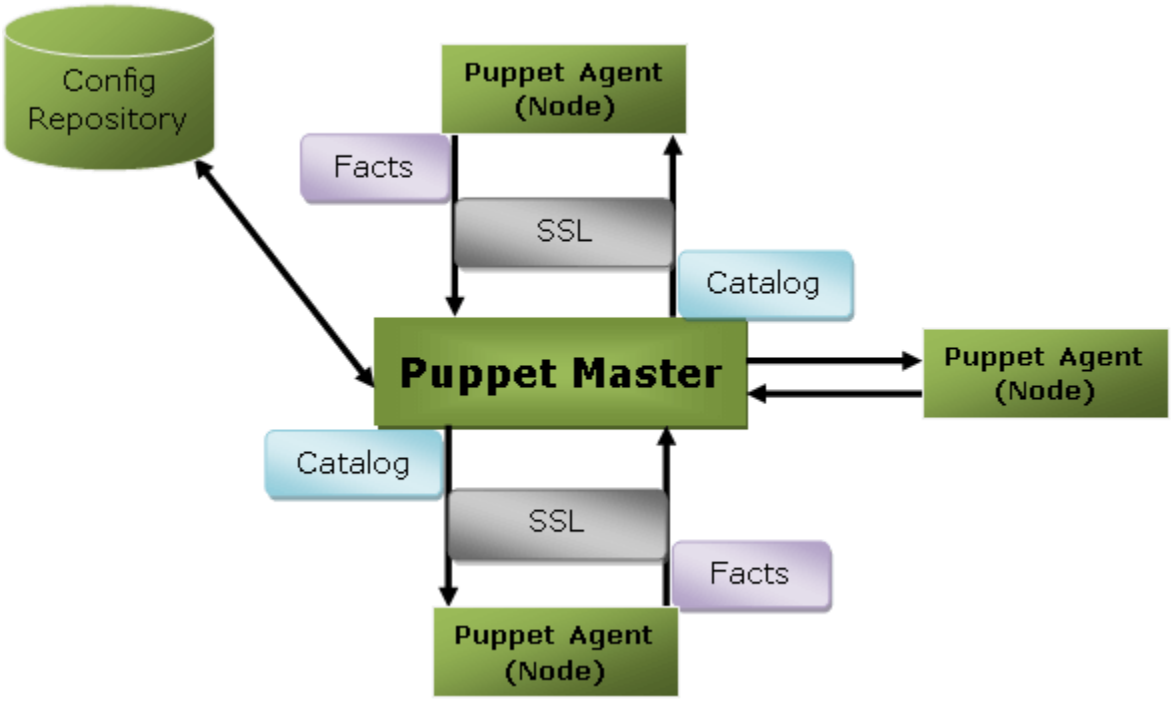
a. To install and configure Pull based Software Configuration Management and provisioning tools using Puppet
b. To learn Software Configuration Management and provisioning using Puppet Blocks (Manifest, Modules, Classes, Function)
c. To provision a LAMP/MEAN Stack using Puppet Manifest

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 Architecture

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.



Here, the client is referred to as a Puppet agent/slave/node, and the server is referred to as a Puppet master.

Advantages of puppet

Puppet is one the most popular configuration management tools around and has plenty of advantages.

1. Puppet is open source

2. Puppet allows resource abstraction

3. Puppet does a transaction only if needed

4. Puppet boosts manageability and productivity

5. Puppet is cross-platform

6. Puppet's language is clean and easily learnable

7. Puppet has cron-like support

8. Puppet has override mechanism

9. Puppet has an active community

Disadvantages of Puppet

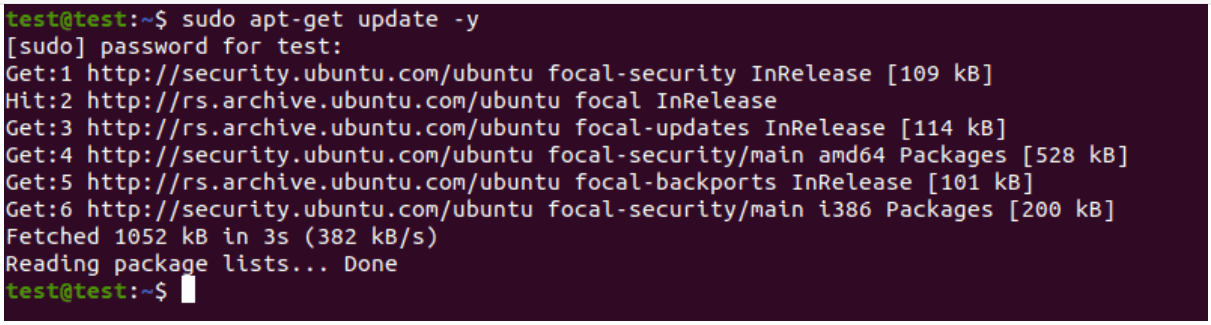
For all its advantages, Puppet has a few drawback as well.

1. Ruby can be complex to understand
2. If one wants to extend Puppet, one has to work with Ruby (which Puppet is written in) and it may not be easy as Ruby is not hugely popular.
3. Rapid releases and evolution
4. Puppet releases new versions quite fast and it can be quite a task to keep up with the new features and breaking changes (if any).
5. Puppet doesn't have comprehensive reporting features
6. It is not possible to look at comprehensive reports on the transactions that Puppet carries out. Those features are still upcoming and not very mature.
7. Puppet may not be suitable for smaller setups and businesses
8. Smaller setups have seen more success with Chef and Ansible and do not prefer Puppet to avoid complexity.
9. Puppet can be difficult for those new to programming
10. This is not exactly a con but people who do system administration may not be versed with programming. They can find puppet a bit daunting to start with.

Step 1: Update Package List

Before you start the installation process, update the list of available packages:

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



Step 2: Set Up Hostname Resolution

With Puppet, master and client nodes communicate using hostnames. Before installing Puppet, you need to set up a unique hostname on each node.

1. Open the **hosts** file on each node by using:

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

2. Paste the following lines at the end of each **hosts** file:

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

Where:

- **[puppet master ip]** is the IP address of the master node, and
- **[puppet client ip]** is the IP address of the client node.

```
  GNU nano 4.8                                                    /etc/hosts
127.0.0.1       localhost
127.0.1.1       test

# 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

192.168.0.1 puppetmaster puppet
192.168.0.2 puppetclient
```

3. Press **Ctrl + X** to close the file, and then type **Y** and press **Enter** to save the changes you made.

Step 3: Install Puppet Server on Master Node

1. Download the latest Puppet version on the master node:

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

```
test@test:~$ wget https://apt.puppetlabs.com/puppet6-release-focal.deb
--2021-03-11 16:38:27--  https://apt.puppetlabs.com/puppet6-release-focal.deb
Resolving apt.puppetlabs.com (apt.puppetlabs.com)... 13.33.68.96, 13.33.68.13, 13.33.68.7
Connecting to apt.puppetlabs.com (apt.puppetlabs.com)|13.33.68.96|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 11752 (11K) [application/x-debian-package]
Saving to: ‘puppet6-release-focal.deb’

puppet6-release-focal.deb           100%[=============================================

2021-03-11 16:38:28 (1,14 MB/s) - ‘puppet6-release-focal.deb’ saved [11752/11752]

test@test:~$
```

2. Once the download is complete, install the package by using:

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

```
test@test:~$ sudo dpkg -i puppet6-release-focal.deb
Selecting previously unselected package puppet6-release.
(Reading database ... 158229 files and directories currently installed.)
Preparing to unpack puppet6-release-focal.deb ...
Unpacking puppet6-release (6.0.0-14focal) ...
Setting up puppet6-release (6.0.0-14focal) ...
test@test:~$
```

3. Update the package repository:

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

4. Install the Puppet server with the following command:

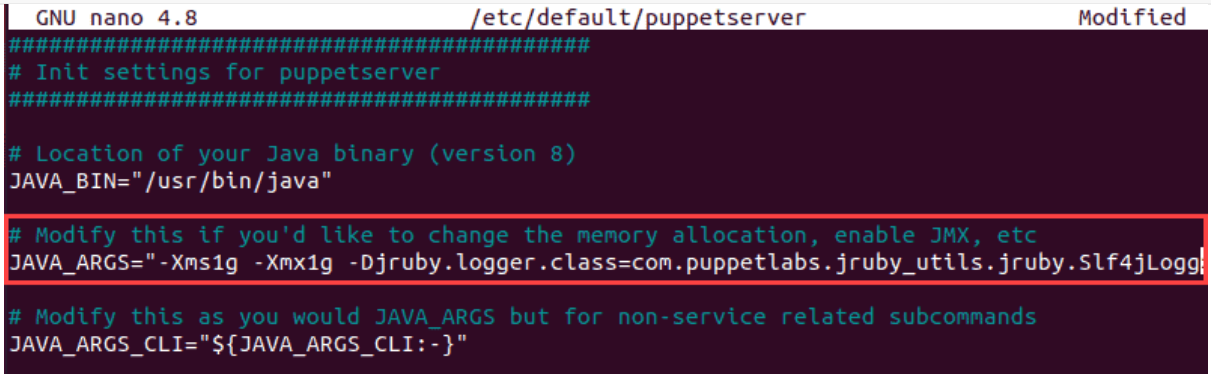
```
sudo apt-get install puppetserver -y
```

5. Open the **puppetserver** file by using:

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

6. In the **puppetserver** file, modify the following line to change the memory size to 1GB:

```
JAVA_ARGS="-Xms1g -Xmx1g -Djruby.logger.class=com.puppetlabs.jruby_utils.
jruby.Slf4jLogger"
```



7. Press **Ctrl + X** to close the **puppetserver** file. Type **Y** and press **Enter** to save the changes you made.

8. Start the Puppet service and set it to launch on system boot by using:

```
sudo systemctl start puppetserver
sudo systemctl enable puppetserver
```

```
test@test:~$ sudo systemctl start puppetserver
test@test:~$ sudo systemctl enable puppetserver
Synchronizing state of puppetserver.service with SysV service script with /lib/systemd/sy
stemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable puppetserver
Created symlink /etc/systemd/system/multi-user.target.wants/puppetserver.service → /lib/s
ystemd/system/puppetserver.service.
test@test:~$
```

9. Check if the Puppet service is running with:

```
sudo systemctl status puppetserver
```

```
test@test:~$ sudo systemctl status puppetserver
● puppetserver.service - puppetserver Service
     Loaded: loaded (/lib/systemd/system/puppetserver.service; enabled; vendor preset: e>
     Active: active (running) since Thu 2021-03-11 16:51:30 CET; 1min 45s ago
   Main PID: 6721 (java)
      Tasks: 40 (limit: 4915)
     Memory: 702.8M
     CGroup: /system.slice/puppetserver.service
             └─6721 /usr/bin/java -Xms1g -Xmx1g -Djruby.logger.class=com.puppetlabs.jrub>

Mar 11 16:50:14 test systemd[1]: Starting puppetserver Service...
Mar 11 16:50:57 test systemd[1]: /lib/systemd/system/puppetserver.service:23: PIDFile= r>
Mar 11 16:51:25 test systemd[1]: /lib/systemd/system/puppetserver.service:23: PIDFile= r>
Mar 11 16:51:26 test systemd[1]: /lib/systemd/system/puppetserver.service:23: PIDFile= r>
Mar 11 16:51:27 test systemd[1]: /lib/systemd/system/puppetserver.service:23: PIDFile= r>
Mar 11 16:51:30 test systemd[1]: Started puppetserver Service.
lines 1-15/15 (END)
```

Step 4: Install Puppet Agent on Client Node

1. Download the latest version of Puppet on a client node:

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

2. Once the download is complete, install the package by using:

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

3. Update the package repository one more time:

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

4. Install the Puppet agent by using:

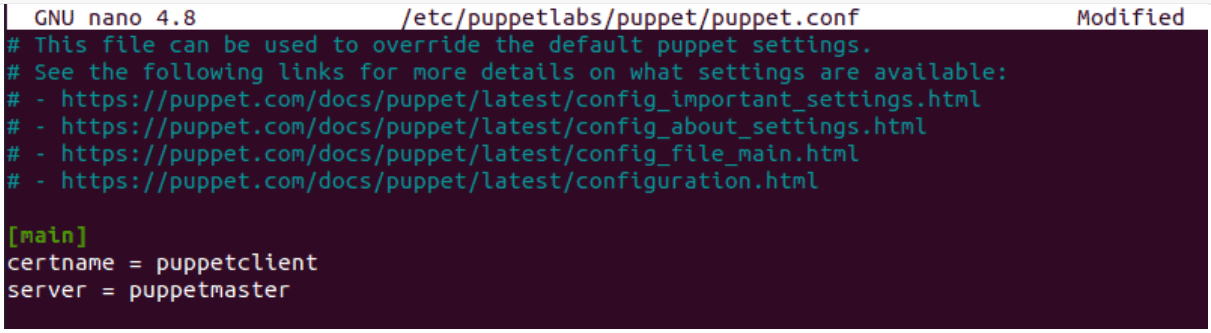
```
sudo apt-get install puppet-agent -y
```

5. Open the Puppet configuration file:

```
sudo nano /etc/puppetlabs/puppet/puppet.conf
```

6. Add the following lines to the end of the Puppet configuration file to define the Puppet master information:

```
[main]
certname = puppetclient
server = puppetmaster
```



7. Press **Ctrl + X** to close the Puppet configuration file, then type **Y** and press **Enter** to save the changes.

8. Start the Puppet service and set it to launch on system boot by using:

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

9. Check if the Puppet service is running with:

```
sudo systemctl status puppet
```

```
test@test:~$ sudo systemctl status puppet
● puppet.service - Puppet agent
     Loaded: loaded (/lib/systemd/system/puppet.service; enabled; vendor preset: enabled)
     Active: active (running) since Thu 2021-03-11 17:17:32 CET; 1min 24s ago
   Main PID: 4394 (puppet)
      Tasks: 2 (limit: 2316)
     Memory: 52.9M
     CGroup: /system.slice/puppet.service
             └─4394 /opt/puppetlabs/puppet/bin/ruby /opt/puppetlabs/puppet/bin/puppet ag>
```

Step 5: Sign Puppet Agent Certificate

1. Using the Puppet master node, list all the available certificates:

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

```
test@test:~$ sudo /opt/puppetlabs/bin/puppetserver ca list --all
Signed Certificates:
    test.ccbill-hq.local       (SHA256)  23:63:47:3B:D5:D6:15:CB:2D:55:B0:A9:CD:A5:89:6D:
AC:C9:B5:73:F2:FE:F0:F3:87:2B:EF:DB:12:45:92:D5    alt names: ["DNS:puppet", "DNS:test.cc
bill-hq.local"]    authorization extensions: [pp_cli_auth: true]
    puppetclient               (SHA256)  80:31:32:29:95:5B:51:5A:FA:50:62:A9:10:5B:3B:63:
01:2A:F8:24:C7:9F:B4:69:27:9D:70:75:15:56:C0:9D
test@test:~$
```

2. Sign the certificates with:

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

3. Use the following command to test the communication between the master and client nodes:

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

Installation of Puppet Master:

Setting hostname of Master and Agent:

```
test@ubuntu:~$ sudo su
[sudo] password for test:
root@ubuntu:/home/test# hostname
ubuntu
```

```
root@ubuntu:/home/test# hostnamectl set-hostname puppet-master
root@ubuntu:/home/test# hostname
puppet-master
```

```
root@ubuntu:/home/test# exit
exit
test@ubuntu:~$ sudo su
root@puppet-master:/home/test#
```

```
root@ubuntu:/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
        ether 02:42:77:63:54:c9  txqueuelen 0  (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

ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.101.128  netmask 255.255.255.0  broadcast 192.168.10
        inet6 fe80::f1b7:800d:b3bf:9ae0  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:e4:70:2a  txqueuelen 1000  (Ethernet)
        RX packets 18460  bytes 27176939 (27.1 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 5014  bytes 352951 (352.9 KB)
        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 345  bytes 25853 (25.8 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 345  bytes 25853 (25.8 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

```
root@puppet-master:/home/test# hostname
puppet-master
root@puppet-master:/home/test# echo $(hostname -I) $(hostname)
192.168.101.128 172.17.0.1 puppet-master
root@puppet-master:/home/test#
```

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

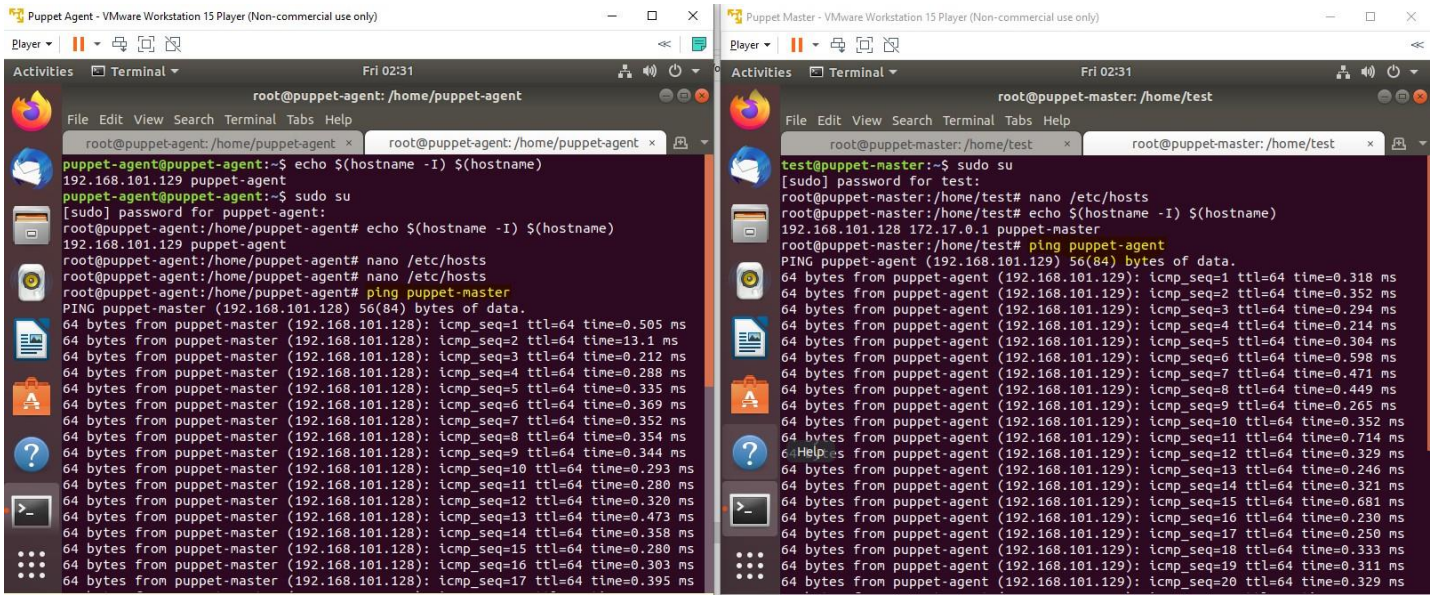
```
root@puppet-master:/home/test# nano /etc/hosts
```

```
GNU nano 2.9.3                          /etc/hosts

27.0.0.1       localhost
27.0.1.1       ubuntu

 The following lines are desirable for IPv6 capable hosts
:1     ip6-localhost ip6-loopback
e00::0 ip6-localnet
f00::0 ip6-mcastprefix
f02::1 ip6-allnodes
f02::2 ip6-allrouters
```

```
 GNU nano 2.9.3                         /etc/hosts

127.0.0.1       localhost
127.0.1.1       ubuntu
192.168.101.129 puppet-agent
# 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
```

```
root@puppet-agent:/home/puppet-agent# nano /etc/hosts
```

```
  GNU nano 2.9.3                          /etc/hosts

127.0.0.1       localhost
127.0.1.1       ubuntu
192.168.101.128 puppet-master

# 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
```



Run apt update and upgrade your master

```
root@puppet-master:/home/test# apt-get update && apt-get upgrade -y
Hit:1 https://download.docker.com/linux/ubuntu bionic InRelease
Hit:2 http://us.archive.ubuntu.com/ubuntu bionic InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:4 http://security.ubuntu.com/ubuntu bionic-security InRelease
Hit:5 http://us.archive.ubuntu.com/ubuntu bionic-backports InRelease
Reading package lists... 10%
```

```
root@puppet-master:/home/test# apt install puppetmaster -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  augeas-lenses debconf-utils facter fonts-lato hiera javascript-common
  libaugeas0 libboost-log1.65.1 libboost-program-options1.65.1
  libboost-regex1.65.1 libcpp-hocon0.1.6 libfacter3.10.0 libjs-jquery
  libleatherman-data libleatherman1.4.0 libruby2.5 libyaml-cpp0.5v5 puppet
  puppet-master rake ruby ruby-augeas ruby-deep-merge ruby-did-you-mean
  ruby-json ruby-minitest ruby-net-telnet ruby-power-assert ruby-selinux
  ruby-shadow ruby-test-unit ruby2.5 rubygems-integration
Suggested packages:
  augeas-doc mcollective-common puppet-common apache2 | lighttpd | httpd
  augeas-tools ruby-rrd ruby-hocon ri ruby-dev bundler
The following NEW packages will be installed:
  augeas-lenses debconf-utils facter fonts-lato hiera javascript-common
  libaugeas0 libboost-log1.65.1 libboost-program-options1.65.1
  libboost-regex1.65.1 libcpp-hocon0.1.6 libfacter3.10.0 libjs-jquery
  libleatherman-data libleatherman1.4.0 libruby2.5 libyaml-cpp0.5v5 puppet
```

```
root@puppet-master:/home/test# puppet --version
5.4.0
root@puppet-master:/home/test# service puppetmaster status
● puppet-master.service - Puppet master
   Loaded: loaded (/lib/systemd/system/puppet-master.service; enabled; vendor p
   Active: active (running) since Fri 2021-10-08 02:37:07 PDT; 9min ago
     Docs: man:puppet-master(8)
 Main PID: 44451 (puppet)
    Tasks: 3 (limit: 2294)
   CGroup: /system.slice/puppet-master.service
           └─44451 /usr/bin/ruby /usr/bin/puppet master

Oct 08 02:37:06 puppet-master puppet-master[44421]: Signed certificate request
Oct 08 02:37:07 puppet-master puppet-master[44421]: puppet-master.localdomain h
Oct 08 02:37:07 puppet-master puppet-master[44421]: Signed certificate request
Oct 08 02:37:07 puppet-master puppet-master[44421]: Removing file Puppet::SSL::
Oct 08 02:37:07 puppet-master puppet-master[44421]: Removing file Puppet::SSL::
Oct 08 02:37:07 puppet-master puppet-master[44421]: The WEBrick Puppet master s
Oct 08 02:37:07 puppet-master puppet-master[44421]:    (location: /usr/lib/ruby
Oct 08 02:37:07 puppet-master puppet-master[44451]: Reopening log files
Oct 08 02:37:07 puppet-master puppet-master[44451]: Starting Puppet master vers
Oct 08 02:37:07 puppet-master systemd[1]: Started Puppet master.
lines 1-19/19 (END)
```

```
root@puppet-master:/home/puppet-master# puppet config set dns_alt_names 'puppet master sies,puppet-master.sies.edu'
root@puppet-master:/home/puppet-master# cat /etc/puppetlabs/puppet/puppet.conf
cat: /etc/puppetlabs/puppet/puppet.conf: No such file or directory
root@puppet-master:/home/puppet-master# cat /etc/puppet/puppet.conf
[main]
ssldir = /var/lib/puppet/ssl
dns_alt_names = puppet master sies,puppet-master.sies.edu

[master]
vardir = /var/lib/puppet
cadir  = /var/lib/puppet/ssl/ca
dns_alt_names = puppet
```

```
root@puppet-master:/home/puppet-master# nano /etc/puppet/puppet.conf
```

```
root@puppet-master: /etc/puppet/code/environments/production/manifests
File  Edit  View  Search  Terminal  Help
  GNU nano 2.9.3                                  /etc/puppet/puppet.conf

[main]
ssldir = /var/lib/puppet/ssl
dns_alt_names = puppet master sies,puppet-master.sies.edu
server = puppet-master.sies.edu
[master]
vardir = /var/lib/puppet
cadir  = /var/lib/puppet/ssl/ca
dns_alt_names = puppet
```

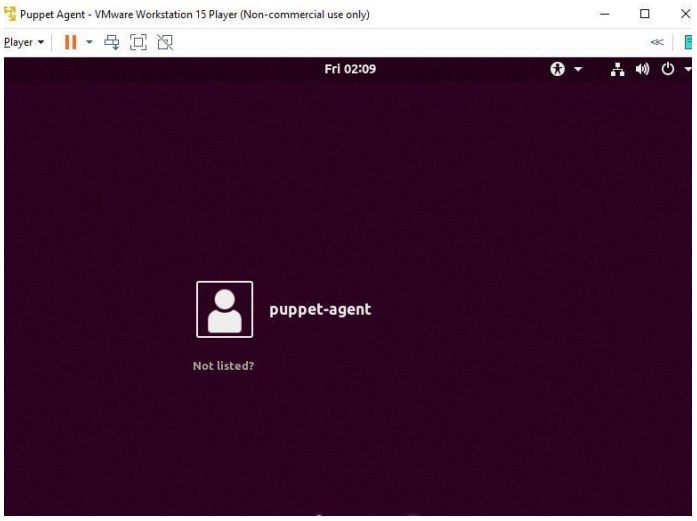
```
root@puppet-master:/home/puppet-master# service puppetmaster restart
root@puppet-master:/home/puppet-master# service puppetmaster status
● puppet-master.service - Puppet master
   Loaded: loaded (/lib/systemd/system/puppet-master.service; enabled; vendor pr
   Active: active (running) since Mon 2021-10-11 22:24:27 PDT; 2s ago
     Docs: man:puppet-master(8)
  Process: 60306 ExecStart=/usr/bin/puppet master (code=exited, status=0/SUCCESS
 Main PID: 60316 (puppet)
    Tasks: 3 (limit: 2294)
   CGroup: /system.slice/puppet-master.service
           └─60316 /usr/bin/ruby /usr/bin/puppet master

Oct 11 22:24:09 puppet-master.sies.edu systemd[1]: Starting Puppet master...
Oct 11 22:24:26 puppet-master.sies.edu puppet-master[60306]: The WEBrick Puppet
Oct 11 22:24:26 puppet-master.sies.edu puppet-master[60306]:    (location: /usr/
Oct 11 22:24:26 puppet-master.sies.edu puppet-master[60316]: Reopening log files
Oct 11 22:24:26 puppet-master.sies.edu puppet-master[60316]: Starting Puppet mas
Oct 11 22:24:27 puppet-master.sies.edu systemd[1]: Started Puppet master.
```

Troubleshoot puppet master

```
root@puppet-master:/home/puppet-master# apt install net-tools
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  net-tools
0 upgraded, 1 newly installed, 0 to remove and 11 not upgraded.
Need to get 194 kB of archives.
After this operation, 803 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu bionic/main amd64 net-tools amd64 1.60+git20161116.90da8a0-1ubuntu1 [194 kB]
Fetched 194 kB in 1s (145 kB/s)
Selecting previously unselected package net-tools.
(Reading database ... 165442 files and directories currently installed.)
Preparing to unpack .../net-tools_1.60+git20161116.90da8a0-1ubuntu1_amd64.deb ...
Unpacking net-tools (1.60+git20161116.90da8a0-1ubuntu1) ...
Setting up net-tools (1.60+git20161116.90da8a0-1ubuntu1) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
root@puppet-master:/home/puppet-master# netstat -anpl | grep 8140
tcp        0      0 0.0.0.0:8140            0.0.0.0:*               LISTEN      60316/ruby
tcp6       0      0 :::8140                 :::*                    LISTEN      60316/ruby
root@puppet-master:/home/puppet-master# sudo ufw allow 8140/tcp
Rules updated
Rules updated (v6)
root@puppet-master:/home/puppet-master# service puppetmaster restart
root@puppet-master:/home/puppet-master# netstat -anpl | grep 8140
tcp        0      0 0.0.0.0:8140            0.0.0.0:*               LISTEN      61025/ruby
tcp6       0      0 :::8140                 :::*                    LISTEN      61025/ruby
root@puppet-master:/home/puppet-master# service puppetmaster status
● puppet-master.service - Puppet master
   Loaded: loaded (/lib/systemd/system/puppet-master.service; enabled; vendor pr
   Active: active (running) since Mon 2021-10-11 22:27:00 PDT; 33s ago
```

Installation of Puppet Agent:

Setting hostname of Agent:



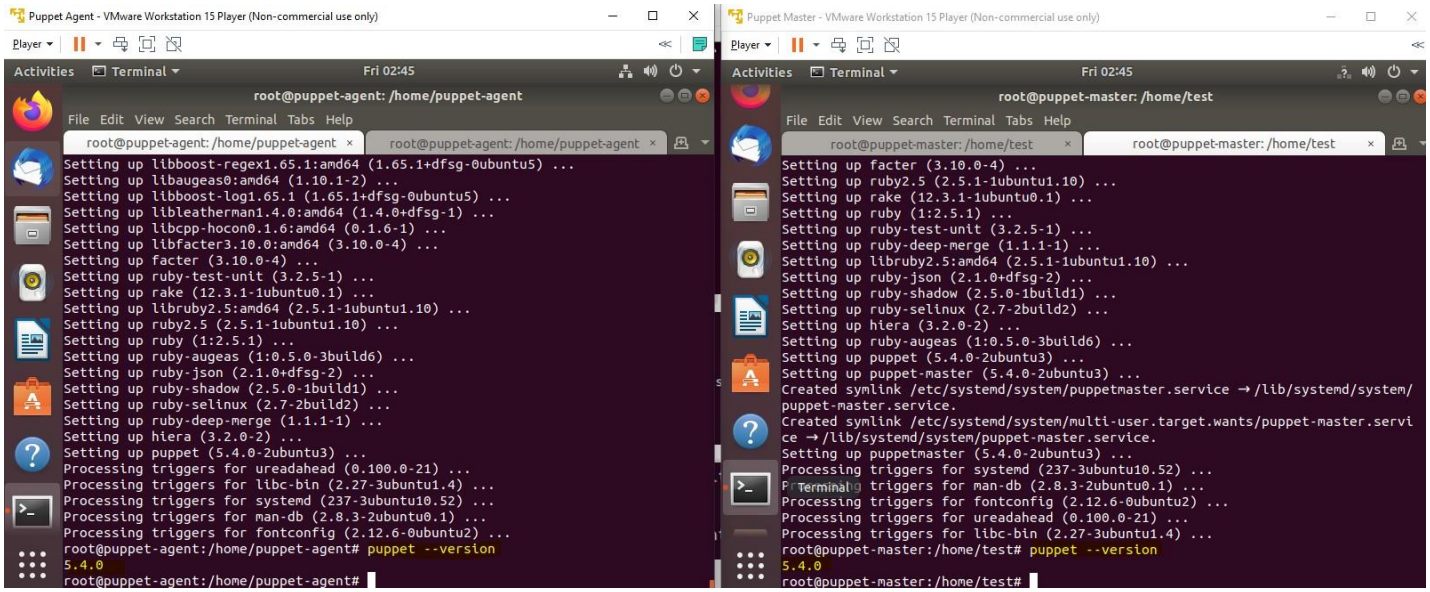

```
puppet-agent@ubuntu:~$ sudo su
[sudo] password for puppet-agent:
root@ubuntu:/home/puppet-agent# hostname
ubuntu
root@ubuntu:/home/puppet-agent# hostnamectl set-hostname puppet-agent
root@ubuntu:/home/puppet-agent# hostname
puppet-agent
root@ubuntu:/home/puppet-agent#
```

```
root@ubuntu:/home/puppet-agent# exit
exit
puppet-agent@ubuntu:~$ sudo su
root@puppet-agent:/home/puppet-agent#
```

Run apt update and upgrade your agent

```
root@puppet-agent:/home/puppet-agent# apt-get update && apt-get upgrade -y
Hit:1 http://us.archive.ubuntu.com/ubuntu bionic InRelease
Get:2 http://us.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:3 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Fetched 252 kB in 2s (105 kB/s)
Reading package lists... 26%
```

```
root@puppet-agent:/home/puppet-agent# apt install puppet
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  augeas-lenses debconf-utils facter fonts-lato hiera javascript-common
  libaugeas0 libboost-log1.65.1 libboost-program-options1.65.1
  libboost-regex1.65.1 libcpp-hocon0.1.6 libcurl4 libfacter3.10.0
  libjs-jquery libleatherman-data libleatherman1.4.0 libruby2.5
  libyaml-cpp0.5v5 rake ruby ruby-augeas ruby-deep-merge ruby-did-you-mean
  ruby-json ruby-minitest ruby-net-telnet ruby-power-assert ruby-selinux
  ruby-shadow ruby-test-unit ruby2.5 rubygems-integration
Suggested packages:
  augeas-doc mcollective-common puppet-common apache2 | lighttpd | httpd
  augeas-tools ruby-rrd ruby-hocon ri ruby-dev bundler
The following NEW packages will be installed:
  augeas-lenses debconf-utils facter fonts-lato hiera javascript-common
  libaugeas0 libboost-log1.65.1 libboost-program-options1.65.1
  libboost-regex1.65.1 libcpp-hocon0.1.6 libcurl4 libfacter3.10.0
  libjs-jquery libleatherman-data libleatherman1.4.0 libruby2.5
  libyaml-cpp0.5v5 puppet rake ruby ruby-augeas ruby-deep-merge
  ruby-did-you-mean ruby-json ruby-minitest ruby-net-telnet ruby-power-assert
```




```
root@puppet-agent:/home/puppet-agent# puppet --version
5.4.0
```

Configure puppet master entry in puppet agent by adding foll. line into puppet.conf:

```
root@puppet-agent:/home/puppet-agent# nano /etc/puppet/puppet.conf
```

```
root@puppet-agent: /home/puppet-agent
File  Edit  View  Search  Terminal  Help
  GNU nano 2.9.3                                  /etc/puppet/puppet.conf

[main]
ssldir = /var/lib/puppet/ssl
server = puppet-master.sies.edu
[master]
vardir = /var/lib/puppet
cadir  = /var/lib/puppet/ssl/ca
dns_alt_names = puppet
```

Enable your agent and set server:

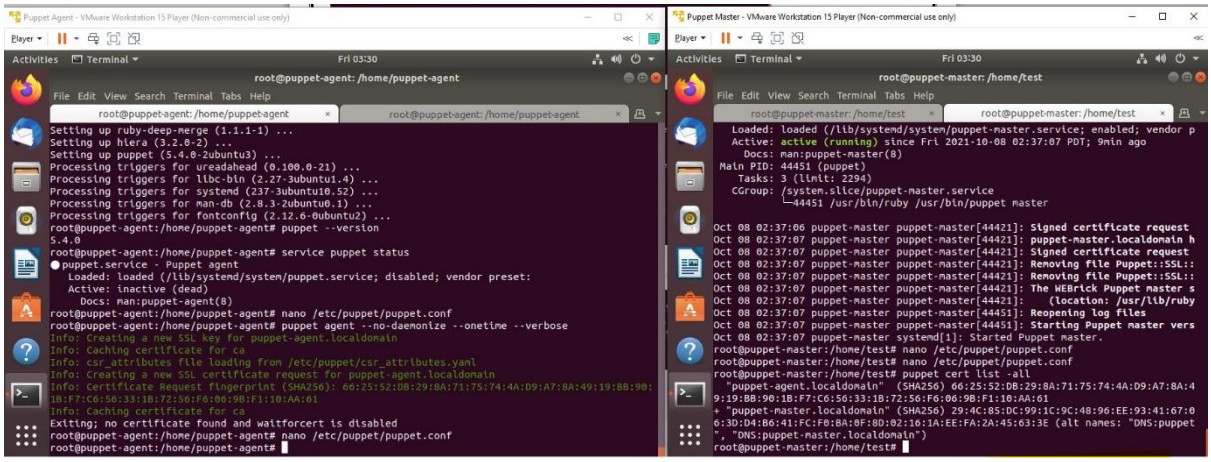
```
root@puppet-agent:/home/puppet-agent# puppet agent --enable
root@puppet-agent:/home/puppet-agent# puppet agent --server puppet-master
root@puppet-agent:/home/puppet-agent#
```

Invoke or start puppet agent and request for certificate:

```
root@puppet-agent:/home/puppet-agent# puppet --version
5.4.0
root@puppet-agent:/home/puppet-agent# service puppet status
● puppet.service - Puppet agent
   Loaded: loaded (/lib/systemd/system/puppet.service; disabled; vendor preset:
   Active: inactive (dead)
     Docs: man:puppet-agent(8)
root@puppet-agent:/home/puppet-agent# nano /etc/puppet/puppet.conf
root@puppet-agent:/home/puppet-agent# puppet agent --no-daemonize --onetime --verbose
Info: Creating a new SSL key for puppet-agent.localdomain
Info: Caching certificate for ca
Info: csr_attributes file loading from /etc/puppet/csr_attributes.yaml
Info: Creating a new SSL certificate request for puppet-agent.localdomain
Info: Certificate Request fingerprint (SHA256): 66:25:52:DB:29:8A:71:75:74:4A:D9:A7:8A:49:19:BB:90:
1B:F7:C6:56:33:1B:72:56:F6:06:9B:F1:10:AA:61
Info: Caching certificate for ca
Exiting; no certificate found and waitforcert is disabled
```

List all certificates on master

```
root@puppet-master:/home/test# puppet cert list -all
  "puppet-agent.localdomain"  (SHA256) 66:25:52:DB:29:8A:71:75:74:4A:D9:A7:8A:4
9:19:BB:90:1B:F7:C6:56:33:1B:72:56:F6:06:9B:F1:10:AA:61
+ "puppet-master.localdomain" (SHA256) 29:4C:85:DC:99:1C:9C:48:96:EE:93:41:67:0
6:3D:D4:B6:41:FC:F0:BA:0F:8D:02:16:1A:EE:FA:2A:45:63:3E (alt names: "DNS:puppet
", "DNS:puppet-master.localdomain")
```



Or troubleshoot:

```
root@puppet-agent:/home/puppet-agent# service puppet status
● puppet.service - Puppet agent
   Loaded: loaded (/lib/systemd/system/puppet.service; disabled; vendor preset:
   Active: inactive (dead)
     Docs: man:puppet-agent(8)
root@puppet-agent:/home/puppet-agent# service puppet restart
root@puppet-agent:/home/puppet-agent# service puppet status
● puppet.service - Puppet agent
   Loaded: loaded (/lib/systemd/system/puppet.service; disabled; vendor preset:
   Active: active (running) since Mon 2021-10-11 22:26:41 PDT; 22s ago
     Docs: man:puppet-agent(8)
  Process: 49882 ExecStart=/usr/bin/puppet agent (code=exited, status=0/SUCCESS)
 Main PID: 49890 (puppet)
    Tasks: 2 (limit: 2294)
   CGroup: /system.slice/puppet.service
           └─49890 /usr/bin/ruby /usr/bin/puppet agent

Oct 11 22:26:34 puppet-agent.sies.edu systemd[1]: Starting Puppet agent...
Oct 11 22:26:40 puppet-agent.sies.edu puppet-agent[49890]: Reopening log files
Oct 11 22:26:41 puppet-agent.sies.edu systemd[1]: Started Puppet agent.
Oct 11 22:26:42 puppet-agent.sies.edu puppet-agent[49890]: Could not request cer
```

```
root@puppet-agent:/home/puppet-agent# puppet resource service puppet ensure=running enable=true
Notice: /Service[puppet]/enable: enable changed 'false' to 'true'
service { 'puppet':
  ensure => 'running',
  enable => 'true',
}
root@puppet-agent:/home/puppet-agent# systemctl restart puppet
root@puppet-agent:/home/puppet-agent# systemctl enable puppet
Synchronizing state of puppet.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable puppet
```

```
root@puppet-agent:/home/puppet-agent# puppet agent --no-daemonize --onetime --verbose
Exiting; no certificate found and waitforcert is disabled
root@puppet-agent:/home/puppet-agent# puppet agent -t
Exiting; no certificate found and waitforcert is disabled
root@puppet-agent:/home/puppet-agent# puppet agent --no-daemonize --onetime --verbose
Exiting; no certificate found and waitforcert is disabled
```

List all certificates on master:

```
root@puppet-master:/home/test# puppet cert list -all
  "puppet-agent.localdomain"  (SHA256) 66:25:52:DB:29:8A:71:75:74:4A:D9:A7:8A:4
9:19:BB:90:1B:F7:C6:56:33:1B:72:56:F6:06:9B:F1:10:AA:61
+ "puppet-master.localdomain" (SHA256) 29:4C:85:DC:99:1C:9C:48:96:EE:93:41:67:0
6:3D:D4:B6:41:FC:F0:BA:0F:8D:02:16:1A:EE:FA:2A:45:63:3E (alt names: "DNS:puppet
", "DNS:puppet-master.localdomain")
```

Sign certificate on puppet master:

```
root@puppet-master:/home/test# puppet cert list -all
  "puppet-agent.localdomain"  (SHA256) 66:25:52:DB:29:8A:71:75:74:4A:D9:A7:8A:4
9:19:BB:90:1B:F7:C6:56:33:1B:72:56:F6:06:9B:F1:10:AA:61
+ "puppet-master.localdomain" (SHA256) 29:4C:85:DC:99:1C:9C:48:96:EE:93:41:67:0
6:3D:D4:B6:41:FC:F0:BA:0F:8D:02:16:1A:EE:FA:2A:45:63:3E (alt names: "DNS:puppet
", "DNS:puppet-master.localdomain")
root@puppet-master:/home/test# puppet cert sign puppet-agent
Error: Could not find CSR for: "puppet-agent".
root@puppet-master:/home/test# puppet cert sign puppet-agent.localdomain
Signing Certificate Request for:
  "puppet-agent.localdomain" (SHA256) 66:25:52:DB:29:8A:71:75:74:4A:D9:A7:8A:49
:19:BB:90:1B:F7:C6:56:33:1B:72:56:F6:06:9B:F1:10:AA:61
Notice: Signed certificate request for puppet-agent.localdomain
Notice: Removing file Puppet::SSL::CertificateRequest puppet-agent.localdomain
at '/var/lib/puppet/ssl/ca/requests/puppet-agent.localdomain.pem'
```

Or troubleshoot if any error while listing certificates on master and signing certificate on puppet master:

```
root@puppet-master:/home/puppet-master# puppet ca list
Warning: 'puppet ca' is deprecated and will be removed in a future release
   (location: /usr/lib/ruby/vendor_ruby/puppet/application/face_base.rb:244:in `main')
  puppet-agent.sies.edu  (SHA256) C9:56:FE:4E:FB:C0:D8:19:15:3D:19:9B:05:FE:77:01:C5:A4:87:24:3C:61:9E:08:2E:95:A1:7B:89:14:D1:3A
root@puppet-master:/home/puppet-master# puppet cert sign puppet-agent.sies.edu
Signing Certificate Request for:
  "puppet-agent.sies.edu" (SHA256) C9:56:FE:4E:FB:C0:D8:19:15:3D:19:9B:05:FE:77:01:C5:A4:87:24:3C:61:9E:08:2E:95:A1:7B:89:14:D1:3A
Notice: Signed certificate request for puppet-agent.sies.edu
Notice: Removing file Puppet::SSL::CertificateRequest puppet-agent.sies.edu at '/var/lib/puppet/ssl/ca/requests/puppet-agent.sies.edu.pem'
```

Enable agent and Invoke agent separately:

```
root@puppet-agent:/home/puppet-agent# service puppet status
● puppet.service - Puppet agent
   Loaded: loaded (/lib/systemd/system/puppet.service; enabled; vendor preset: e
   Active: active (running) since Mon 2021-10-11 22:30:14 PDT; 4min 57s ago
     Docs: man:puppet-agent(8)
 Main PID: 50039 (puppet)
    Tasks: 2 (limit: 2294)
   CGroup: /system.slice/puppet.service
           └─50039 /usr/bin/ruby /usr/bin/puppet agent

Oct 11 22:30:12 puppet-agent.sies.edu systemd[1]: Starting Puppet agent...
Oct 11 22:30:14 puppet-agent.sies.edu puppet-agent[50039]: Reopening log files
Oct 11 22:30:14 puppet-agent.sies.edu systemd[1]: Started Puppet agent.
Oct 11 22:32:14 puppet-agent.sies.edu puppet-agent[50039]: Did not receive certi
Oct 11 22:34:14 puppet-agent.sies.edu puppet-agent[50039]: Did not receive certi
root@puppet-agent:/home/puppet-agent# puppet agent --enable
root@puppet-agent:/home/puppet-agent# puppet agent --no-daemonize --onetime --verbose
Info: Using configured environment 'production'
Info: Retrieving pluginfacts
Info: Retrieving plugin
Info: Retrieving locales
Info: Caching catalog for puppet-agent.sies.edu
Info: Applying configuration version '1634016976'
Notice: Applied catalog in 0.01 seconds
```

Configure a sample modules on puppet master to install curl on Agent:

Check if curl is already installed on agent:

```
root@puppet-agent:/home/puppet-agent# curl

Command 'curl' not found, but can be installed with:

apt install curl

root@puppet-agent:/home/puppet-agent#
```

Go to puppet master, create foll. Dir. structures

```
root@puppet-master:/home/test# cd /etc/puppet/code/
root@puppet-master:/etc/puppet/code# mkdir -p environments/production/manifests
root@puppet-master:/etc/puppet/code# ls
environments
root@puppet-master:/etc/puppet/code# cd environments/production/manifests/
root@puppet-master:/etc/puppet/code/environments/production/manifests# pwd
/etc/puppet/code/environments/production/manifests
root@puppet-master:/etc/puppet/code/environments/production/manifests#
```

Add the following puppet code inside site.pp:

```
root@puppet-master:/etc/puppet/code/environments/production/manifests# nano site.pp
```

```
root@puppet-master: /etc/puppet/code/environments/production/manifests
File  Edit  View  Search  Terminal  Help
  GNU nano 2.9.3                        site.pp

node 'puppet-agent.sies.edu' {
      package {
         'curl':
              name => 'curl',
              ensure => installed,
      }
}
```

Validate the code:

```
root@puppet-master:/etc/puppet/code/environments/production/manifests# puppet parser validat
e site.pp
```

On agent, fetch the catalog:

```
root@puppet-agent:/home/puppet-agent# puppet agent -t
Info: Using configured environment 'production'
Info: Retrieving pluginfacts
Info: Retrieving plugin
Info: Retrieving locales
Info: Caching catalog for puppet-agent.sies.edu
Info: Applying configuration version '1634017364'
Notice: /Stage[main]/Main/Node[puppet-agent.sies.edu]/Package[curl]/ensure: created
Notice: Applied catalog in 9.90 seconds
```

Verify if curl has been installed:

```
root@puppet-agent:/home/puppet-agent# curl
curl: try 'curl --help' or 'curl --manual' for more information
```

Software Management Configuration Using Puppet

Puppet is a tool that helps you manage and automate the configuration of servers.Puppet provides the flexibility to integrate Reports with third-party tools using Puppet APIs.

Four types of Puppet building blocks are:

- Resources
- Classes
- Manifest
- Modules

Puppet Resources:

Puppet Resources are the building blocks of Puppet.

Resources are the inbuilt functions that run at the back end to perform the required operations in puppet.

Puppet Classes:

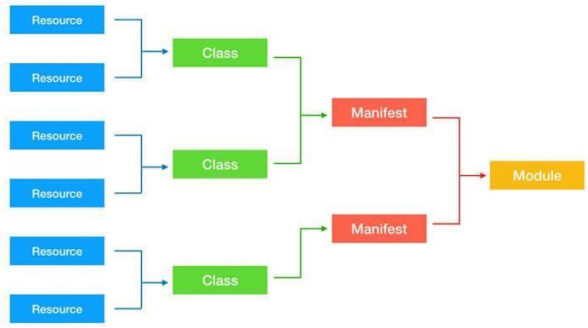
A combination of different resources can be grouped together into a single unit called class.

Puppet Manifest:

Manifest is a directory containing puppet DSL files. Those files have a .pp extension. The .pp extension stands for puppet program. The puppet code consists of definitions or declarations of Puppet Classes.

Puppet Modules:

Modules are a collection of files and directories such as Manifests, Class definitions. They are the re-usable and sharable units in Puppet. For example, the MySQL module to install and configure MySQL or the Jenkins module to manage Jenkins, etc.



Types of Puppet resources

In general, a system consists of files, users, services, processes, packages, etc. In Puppet, these are called resources. Resources are the fundamental building blocks in

Puppet. All the operations on puppet agents are performed with the help of puppet resources.

Puppet resources are the readymade tools that are used to perform various tasks and operations on any supported platform. We can use a single puppet resource to perform a specific task, or we can use multiple puppet resources together to perform some complex application configurations deployments.

Resources can have different types. Puppet uses **resources** and **resource types** in order to describe a system's configuration.

There are three kinds of resource types:

1. Puppet core or built-in resource types.
2. Puppet defined resource types.
3. Puppet custom resource types.

Puppet core or built-in resource types:

Core or built-in resource types are the pre-built puppet resource types shipped with puppet software. All of the core or built-in Puppet resource types are written and maintained by Puppet team.

Puppet defined resource types:

Defined resource types are lightweight resource types written in Puppet declarative language using a combination of existing resource types.

Puppet custom resource types:

Custom resource types are completely customized resource types written in Ruby.

Let's explore about puppet resource types …

In the terminal, type the following command to display a list of Puppet relevant subcommands:

Puppet --help

```
File  Edit  View  Search  Terminal  Help
[osboxes@puppetselfcontained ~]$ puppet --help

Usage: puppet <subcommand> [options] <action> [options]

Available subcommands:

  Common:
    agent              The puppet agent daemon
    apply              Apply Puppet manifests locally
    config             Interact with Puppet's settings.
    help               Display Puppet help.
    lookup             Interactive Hiera lookup
    module             Creates, installs and searches for modules on the Puppet Forge.
    resource           The resource abstraction layer shell
```

In our case, we are interested in the subcommand "**resource**" which we will use to find the information about inbuilt puppet resource types.

In the terminal, type any of the following commands to display a list of **actions** associated with the puppet subcommand "**resource**":

Puppet help resource
Puppet resource --help

```
File  Edit  View  Search  Terminal  Help
[osboxes@puppetselfcontained ~]$ puppet help resource

puppet-resource(8) -- The resource abstraction layer shell
========

SYNOPSIS
--------
Uses the Puppet RAL to directly interact with the system.


USAGE
-----
puppet resource [-h|--help] [-d|--debug] [-v|--verbose] [-e|--edit]
  [-p|--param <parameter>] [-t|--types] [-y|--to_yaml] <type>
  [<name>] [<attribute>=<value> ...]
```

In this case, we have the **resource** as subcommand and **–types** as action.

Puppet has 49 inbuilt core resource types.

In the terminal, type the following command to display a list of available inbuilt puppet resource types:

puppet resource –types

```
File  Edit  View  Search  Terminal  Help
[osboxes@puppetselfcontained ~]$ puppet resource --types
augeas
cron
exec
file
filebucket
group
host
mount
notify
package
resources
schedule
scheduled_task
selboolean
selmodule
service
ssh_authorized_key
sshkey
stage
tidy
user
whit
yumrepo
zfs
```

Each type supports a list of **attributes**. These attributes provide a detailed description that Puppet uses to manage the resource.

To find out all the attributes associated with the puppet resource type, use the following command:

puppet describe <resource type name>
Parameters will list all the available attributes for that resource type.

puppet describe package

```
File  Edit  View  Search  Terminal  Help
[osboxes@puppetselfcontained ~]$ puppet describe package

package
=======
Manage packages.  There is a basic dichotomy in package
support right now:  Some package types (such as yum and apt) can
retrieve their own package files, while others (such as rpm and sun)
cannot.  For those package formats that cannot retrieve their own files,
you can use the `source` parameter to point to the correct file.
Puppet will automatically guess the packaging format that you are
using based on the platform you are on, but you can override it
using the `provider` parameter; each provider defines what it
requires in order to function, and you must meet those requirements
to use a given provider.
You can declare multiple package resources with the same `name`, as long
as they specify different providers and have unique titles.
Note that you must use the _title_ to make a reference to a package
resource; `Package[<NAME>]` is not a synonym for `Package[<TITLE>]` like
it is for many other resource types.
**Autorequires:** If Puppet is managing the files specified as a
package's `adminfile`, `responsefile`, or `source`, the package
resource will autorequire those files.


Parameters
----------
```

```
File  Edit  View  Search  Terminal  Help


Parameters
----------

- **adminfile**
    A file containing package defaults for installing packages.
    This attribute is only used on Solaris. Its value should be a path to a
    local file stored on the target system. Solaris's package tools expect
    either an absolute file path or a relative path to a file in
    `/var/sadm/install/admin`.
    The value of `adminfile` will be passed directly to the `pkgadd` or
    `pkgrm` command with the `-a <ADMINFILE>` option.

- **allow_virtual**
    Specifies if virtual package names are allowed for install and
    uninstall.
    Valid values are `true`, `false`, `yes`, `no`.
    Requires features virtual_packages.

- **allowcdrom**
    Tells apt to allow cdrom sources in the sources.list file.
    Normally apt will bail if you try this.
    Valid values are `true`, `false`.

- **category**
```

It's hard for a new person to understand and relate many unmanaged puppet code files. This is where we need some grouping to tie together operations. The aim is to solve a single problem, such as all operations required to configure ssh on a server or ntp service or a complete web server or database server from scratch.

What are Puppet Classes?

Puppet classes are the collection of puppet resources bundled together as a single unit.

Puppet introduced classes to make the structure re-usable and organized.

First, we need to define a class using class definition syntax; classes must be unique and can be declared only once with the same name:

```
class <class-name> {
<Resource declarations>
}
```

Example:

```
class ntpconfig {
   file {
      "/etc/ntp.conf":
    ensure=> "present", content=> "server 0.centos.pool.ntp.org iburst\n",
   }
}
```

So far we have only defined the class, but we have not used it anywhere. Meaning this code that we have written will never get executed unless we declare this class elsewhere.

Class Declaration

To use a defined class in code, use the **include** keyword.

```
class ntpconfig {
   file {
      "/etc/ntp.conf":
     ensure=> "present",
     content=> "server 0.centos.pool.ntp.org iburst\n",
   }
}
include ntpconfig
```

Let's understand this with a real case scenario.

Demo install NTP

First, make sure the NTP package is not already present on the server, the following command will return nothing if the telnet is not present on the server:

```
rpm -qa | grep -i ntp
```

```
File  Edit  View  Search  Terminal  Help
[root@puppetselfcontained etc]# rpm -qa | grep -i ntp
ntp-4.2.6p5-28.el7.centos.x86_64
python-ntplib-0.3.2-1.el7.noarch
fontpackages-filesystem-1.44-8.el7.noarch
ntpdate-4.2.6p5-28.el7.centos.x86_64
[root@puppetselfcontained etc]#
```

As we can see, the NTP package is already present on the server. Let's remove the existing NTP package:

yum remove ntp

After removing the package, ensure that ntp.conf file is not existing:

ls -lrt /etc/ntp.conf

```
File  Edit  View  Search  Terminal  Help
[root@puppetselfcontained etc]# ls -lrt /etc/ntp.conf
ls: cannot access /etc/ntp.conf: No such file or directory
[root@puppetselfcontained etc]#
```

Verify the ntp service does not exist by running the following command:

systemctl status ntp

```
File  Edit  View  Search  Terminal  Help
[root@puppetselfcontained demo]# systemctl status ntpd
Unit ntpd.service could not be found.
[root@puppetselfcontained demo]#
```

Create a new .pp file to save the code. From the command line:

vi demontp.pp

Change to insert mode by pressing i from the keyboard.

Type the following code to create a new file:

```
# Class Definition
class ntpconfig {
   # Installing NTP Package
  package {"ntp":
    ensure=> "present",
    }
    # Configuring NTP configuration file
  file {"/etc/ntp.conf":
    ensure=> "present",
    content=> "server 0.centos.pool.ntp.org iburst\n",
    }
    # Starting NTP services
  service {"ntpd":
    ensure=> "running",
    }
}
```

After done with editing : press esc

To save the file, press :wq!

Next step is to **check** whether the code has any syntax errors. Execute the following command:

```
puppet parser validate demontp.pp
```

Make sure that you switched to the **root** to be able to complete the test without any error, by executing the command :

```
su root
```

Test is the next step in the code creation process. Execute the following command to perform a smoke test:

```
Puppet applies demontp.pp --noop
```

Last step is to **run** the puppet in real mode and verify the output.

```
puppet apply demontp.pp
```

Puppet didn't perform anything because the demo class was just **defined** but not **declared.**

So, until you declare the puppet class, the code will not get applied.

Let's **declare** the demo class inside the same code using **include class name** at the end of the code:

```
# Class Definition
class ntpconfig {
   # Installing NTP Package
  package {"ntp":
    ensure=> "present",
    }
    # Configuring NTP configuration file
  file {"/etc/ntp.conf":
    ensure=> "present",
    content=> "server 0.centos.pool.ntp.org iburst\n",
    }
    # Starting NTP services
  service {"ntpd":
    ensure=> "running",
    }
}

# Class Declaration
include ntpconfig
```

Again **check** whether the code has any syntax errors. Execute the following command:

```
puppet parser validate demontp.pp
```

Make sure that you switched to the **root** to be able to complete the test without any error, by executing the command :

su root

Testing is the next step in the code creation process. Execute the following command to perform a smoke test:

Puppet apply demontp.pp --noop

Last step is to **run** the puppet in real mode and verify the output.

puppet apply demontp.pp

This time the code gets applied because the class was defined and then declared.

```
File  Edit  View  Search  Terminal  Help
[root@puppetselfcontained demo]# puppet apply demontp.pp
Notice: Compiled catalog for puppetselfcontained.example.com in environment production
in 0.69 seconds
Notice: /Stage[main]/Ntpconfig/Package[ntp]/ensure: created
Notice: /Stage[main]/Ntpconfig/File[/etc/ntp.conf]/content: content changed '{md5}dc9e5
754ad2bb6f6c32b954c04431d0a' to '{md5}83fd3914bd6bb10f2b717c277d995b75'
Notice: /Stage[main]/Ntpconfig/Service[ntpd]/ensure: ensure changed 'stopped' to 'runni
ng'
Notice: Applied catalog in 1.88 seconds
[root@puppetselfcontained demo]#
```

Ensure that ntp.conf is now existing:

ls -lrt /etc/ntp.conf

Verify the ntp service has been started by running the following command:

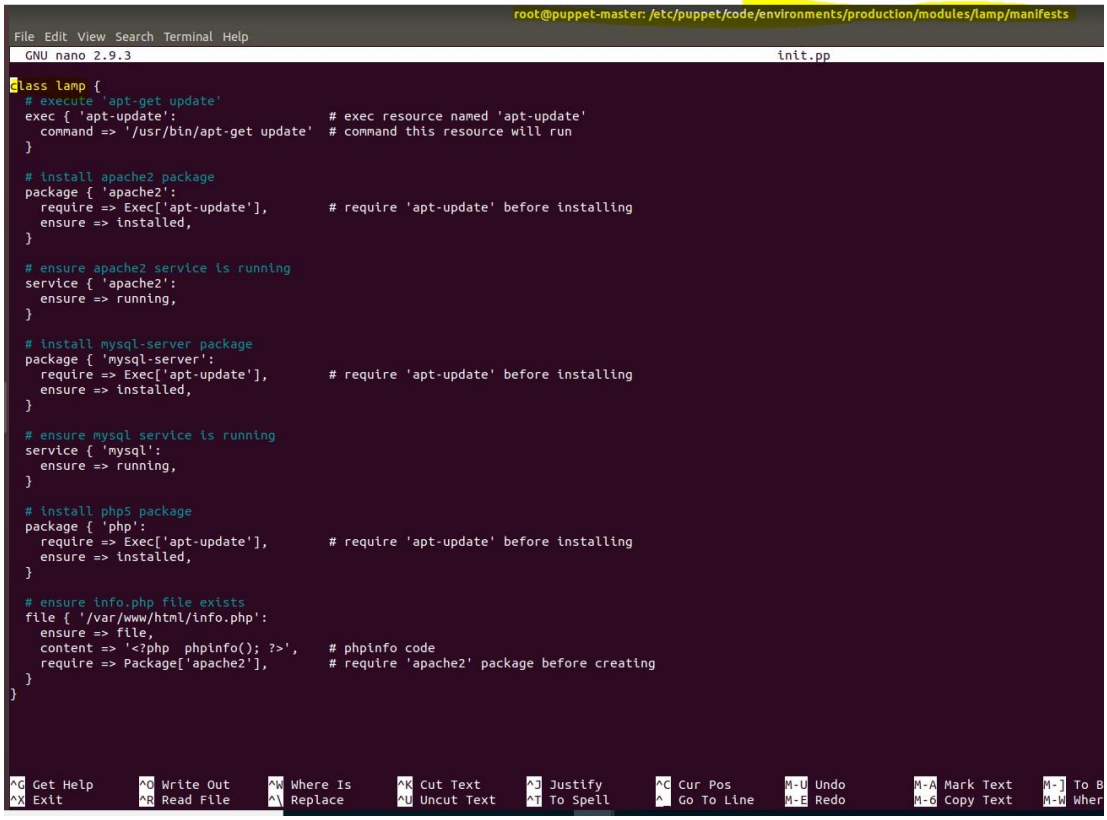
systemctl status ntpd

```
File  Edit  View  Search  Terminal  Help
[root@puppetselfcontained demo]# systemctl status ntpd
● ntpd.service - Network Time Service
   Loaded: loaded (/usr/lib/systemd/system/ntpd.service; disabled; vendor preset: disab
led)
   Active: active (running) since Sat 2019-03-23 04:14:35 EDT; 15s ago
  Process: 21487 ExecStart=/usr/sbin/ntpd -u ntp:ntp $OPTIONS (code=exited, status=0/SU
CCESS)
```

Creating a puppet module:

Create the foll. dir structures to add a LAMP stack module and create a lamp class inside init.pp:

```
root@puppet-master:/etc/puppet/code/environments/production# mkdir -p modules/lamp/manifests
root@puppet-master:/etc/puppet/code/environments/production# cd modules/lamp/manifests/
root@puppet-master:/etc/puppet/code/environments/production/modules/lamp/manifests# nano init.pp
```



Include lamp class inside site.pp:

Before fetching the catalog on agent, verify first whether the following packages are installed or not:

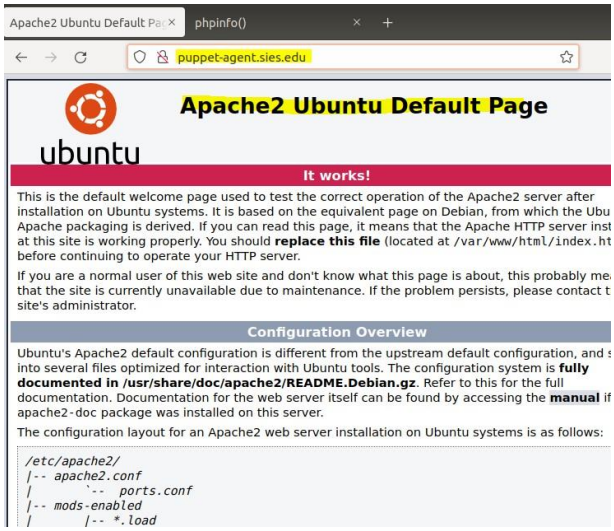
```
puppet-agent@puppet-agent:~$ mysql

Command 'mysql' not found, but can be installed with:

sudo apt install mysql-client-core-5.7
sudo apt install mariadb-client-core-10.1

puppet-agent@puppet-agent:~$ php

Command 'php' not found, but can be installed with:

sudo apt install php7.2-cli
sudo apt install hhvm

puppet-agent@puppet-agent:~$ apache

Command 'apache' not found, did you mean:

  command 'apache2' from deb apache2-bin

Try: sudo apt install <deb name>
```

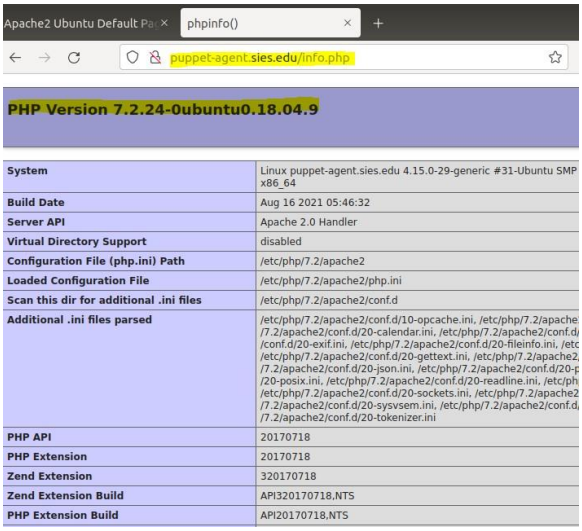
Fetch the catalog:

```
root@puppet-agent:/home/puppet-agent# puppet agent -t
Info: Using configured environment 'production'
Info: Retrieving pluginfacts
Info: Retrieving plugin
Info: Retrieving locales
Info: Caching catalog for puppet-agent.sies.edu
Info: Applying configuration version '1634018123'
Notice: /Stage[main]/Lamp/Exec[apt-update]/returns: executed successfully
Notice: /Stage[main]/Lamp/Package[apache2]/ensure: created
Notice: /Stage[main]/Lamp/Package[mysql-server]/ensure: created
Notice: /Stage[main]/Lamp/Package[php]/ensure: created
Notice: /Stage[main]/Lamp/File[/var/www/html/info.php]/ensure: defined content as '{md5}d9c0c977ee96604e48b81d795236619a'
Notice: Applied catalog in 197.64 seconds
```

Again verify the packages:

```
puppet-agent@puppet-agent:~$ apache2
[Mon Oct 11 22:56:10.293017 2021] [core:warn] [pid 61192] AH00111: Config variable ${APACHE_R
apache2: Syntax error on line 80 of /etc/apache2/apache2.conf: DefaultRuntimeDir must be a va
puppet-agent@puppet-agent:~$ php --version
PHP 7.2.24-0ubuntu0.18.04.9 (cli) (built: Aug 16 2021 05:46:32) ( NTS )
Copyright (c) 1997-2018 The PHP Group
Zend Engine v3.2.0, Copyright (c) 1998-2018 Zend Technologies
    with Zend OPcache v7.2.24-0ubuntu0.18.04.9, Copyright (c) 1999-2018, by Zend Technologies
puppet-agent@puppet-agent:~$ service apache2 status
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset:
  Drop-In: /lib/systemd/system/apache2.service.d
           └─apache2-systemd.conf
   Active: active (running) since Mon 2021-10-11 22:58:38 PDT; 26s ago
  Process: 68733 ExecStop=/usr/sbin/apachectl stop (code=exited, status=0/SUCCES
  Process: 68738 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCC
 Main PID: 68742 (apache2)
    Tasks: 6 (limit: 2294)
   CGroup: /system.slice/apache2.service
           ├─68742 /usr/sbin/apache2 -k start
           ├─68751 /usr/sbin/apache2 -k start
           ├─68752 /usr/sbin/apache2 -k start
           ├─68753 /usr/sbin/apache2 -k start
           ├─68754 /usr/sbin/apache2 -k start
           └─68755 /usr/sbin/apache2 -k start

Oct 11 22:58:38 puppet-agent.sies.edu systemd[1]: Starting The Apache HTTP Serve
Oct 11 22:58:38 puppet-agent.sies.edu systemd[1]: Started The Apache HTTP Server
puppet-agent@puppet-agent:~$ mysql
ERROR 1045 (28000): Access denied for user 'puppet-agent'@'localhost' (using password: NO)
```





Creating index.html file inside LAMP module:

```
root@puppet-master:/etc/puppet/code/environments/production/modules/lamp# mkdir files
```

```
root@puppet-master:/etc/puppet/code/environments/production/modules/lamp# ls
files  manifests
```

```
root@puppet-master:/etc/puppet/code/environments/production/modules/lamp# ls files/index.html
files/index.html
```

root@puppet-master: /etc/puppet/code/environments/production/modules/lamp

File Edit View Search Terminal Help

GNU nano 2.9.3 files/index.html

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<title> Login Page </title>
<style>
Body {
  font-family: Calibri, Helvetica, sans-serif;
  background-color: pink;
}
button {
       background-color: #4CAF50;
       width: 100%;
        color: orange;
        padding: 15px;
        margin: 10px 0px;
        border: none;
        cursor: pointer;
         }
 form {
        border: 3px solid #f1f1f1;
    }
 input[type=text], input[type=password] {
        width: 100%;
        margin: 8px 0;
        padding: 12px 20px;
        display: inline-block;
        border: 2px solid green;
        box-sizing: border-box;
    }
 button:hover {
        opacity: 0.7;
    }
  .cancelbtn {
        width: auto;
```

Add a file resource to create an index.html inside init.pp:

```
root@puppet-master:/etc/puppet/code/environments/production/modules/lamp/manifests# cat init.pp
class lamp {
  # execute 'apt-get update'
  exec { 'apt-update':                    # exec resource named 'apt-update'
    command => '/usr/bin/apt-get update'  # command this resource will run
  }

  # install apache2 package
  package { 'apache2':
    require => Exec['apt-update'],        # require 'apt-update' before installing
    ensure => installed,
  }

  # ensure apache2 service is running
  service { 'apache2':
    ensure => running,
  }

  # install mysql-server package
  package { 'mysql-server':
    require => Exec['apt-update'],        # require 'apt-update' before installing
    ensure => installed,
  }

  # ensure mysql service is running
  service { 'mysql':
    ensure => running,
  }

  # install php5 package
  package { 'php':
    require => Exec['apt-update'],        # require 'apt-update' before installing
    ensure => installed,
  }

  # ensure info.php file exists
  file { '/var/www/html/info.php':
    ensure => file,
    content => '<?php  phpinfo(); ?>',    # phpinfo code
    require => Package['apache2'],        # require 'apache2' package before creating
  }
  # change index.html
  file { '/var/www/html/index.html':
    ensure => present,
    source => 'puppet:///modules/lamp/index.html',
    require => File[/var/www/html/index.html],
  }
}
```

Fetch the catalog at agent: puppet agent –t

Test the webpage inside browser at agent using following URL: puppet-agent.sies.edu/index.html

Conclusion: Thus, we have successfully installed puppet master and understand about its software management configuration and perform Lampstack using puppet manifest.

Postlab

Chef:

Chef is an open source technology developed by Opscode. Adam Jacob, co-founder of Opscode is known as the founder of Chef. This technology uses Ruby encoding to develop basic building blocks like recipe and cookbooks. Chef is used in infrastructure automation and helps in reducing manual and repetitive tasks for infrastructure management.

Chef have got its own convention for different building blocks, which are required to manage and automate infrastructure.

Why Chef?

Chef is a configuration management technology used to automate the infrastructure provisioning. It is developed based on Ruby DSL language. It is used to streamline the task of configuration and managing the company's server. It has the capability to get integrated with any of the cloud technology.

In DevOps, we use Chef to deploy and manage servers and applications in-house and on the cloud.

Features of Chef

Following are the most prominent features of Chef −

- Chef uses popular Ruby language to create a domain-specific language.
- Chef does not make assumptions on the current status of a node. It uses its mechanisms to get the current status of machine.
- Chef is ideal for deploying and managing the cloud server, storage, and software.

Advantages of Chef

Chef offers the following advantages −

- **Lower barrier for entry** − As Chef uses native Ruby language for configuration, a standard configuration language it can be easily picked up by anyone having some development experience.

- **Excellent integration with cloud** – Using the knife utility, it can be easily integrated with any of the cloud technologies. It is the best tool for an organization that wishes to distribute its infrastructure on multi-cloud environment.

Disadvantages of Chef

Some of the major drawbacks of Chef are as follows –

- One of the huge disadvantages of Chef is the way cookbooks are controlled. It needs constant babying so that people who are working should not mess up with others cookbooks.
- Only Chef solo is available.
- In the current situation, it is only a good fit for AWS cloud.
- It is not very easy to learn if the person is not familiar with Ruby.
- Documentation is still lacking.

Key Building Blocks of Chef

Recipe

It can be defined as a collection of attributes which are used to manage the infrastructure. These attributes which are present in the recipe are used to change the existing state or setting a particular infrastructure node. They are loaded during Chef client run and comparted with the existing attribute of the node (machine). It then gets to the status which is defined in the node resource of the recipe. It is the main workhorse of the cookbook.

Cookbook

A cookbook is a collection of recipes. They are the basic building blocks which get uploaded to Chef server. When Chef run takes place, it ensures that the recipes present inside it gets a given infrastructure to the desired state as listed in the recipe.

Resource

It is the basic component of a recipe used to manage the infrastructure with different kind of states. There can be multiple resources in a recipe, which will help in configuring and managing the infrastructure. For example –

- **package** – Manages the packages on a node
- **service** – Manages the services on a node
- **user** – Manages the users on the node
- **group** – Manages groups
- **template** – Manages the files with embedded Ruby template
- **cookbook_file** – Transfers the files from the files subdirectory in the cookbook to a location on the node
- **file** – Manages the contents of a file on the node

- **directory** – Manages the directories on the node
- **execute** – Executes a command on the node
- **cron** – Edits an existing cron file on the node

Attribute

They are basically settings. They can be thought of as a key value pair of anything which one wants to use in the cookbook. There are several different kinds of attributes that can be applied, with a different level of precedence over the final settings that the node operates under.

File

It's a subdirectory within the cookbook that contains any static file which will be placed on the nodes that uses the cookbooks. A recipe then can be declared as a resource that moves the files from that directory to the final node.

Templates

They are similar to files, but they are not static. Template files end with the .ebr extension, which means they contain embedded Ruby. They are mainly used to substitute an attribute value into the files to create the final file version that will be placed on the node.

Metadata.rb

It is used to manage the metadata about the package. This includes details like the name and details of the package. It also includes things such as dependency information that tells which cookbooks this cookbook needs to operate. This allows the Chef server to build the run-list of the node correctly and ensures that all of the pieces are transferred correctly.

Default Cookbook Structure

```
C:\chef\cookbooks\nginx>tree
Folder PATH listing for volume Local Disk
Volume serial number is BE8B-6427
C: ├───attributes
├───definitions
├───files
│   └───default
├───libraries
├───providers
├───recipes
├───resources
└───templates
    └───default
```

What is SaltStack?

Salt is a very powerful automation framework. Salt architecture is based on the idea of executing commands remotely. All networking is designed around some aspect of remote execution. This could be as simple as asking a **Remote Web Server** to display a static Web page, or as complex as using a shell session to interactively issue commands against a remote server. Salt is an example of one of the more complex types of remote execution.

Need for SaltStack

SaltStack is built for speed and scale. This is why it is used to manage large infrastructures with tens of thousands of servers at LinkedIn, WikiMedia and Google.

Imagine that you have multiple servers and want to do things to those servers. You would need to login to each one and do those things one at a time on each one and then you might want to do complicated things like installing software and then configuring that software based on some specific criteria.

Benefits of SaltStack

Being simple as well as a feature-rich system, Salt provides many benefits and they can be summarized as below −

- **Robust** − Salt is powerful and robust configuration management framework and works around tens of thousands of systems.
- **Authentication** − Salt manages simple SSH key pairs for authentication.
- **Secure** − Salt manages secure data using an encrypted protocol.
- **Fast** − Salt is very fast, lightweight communication bus to provide the foundation for a remote execution engine.
- **Virtual Machine Automation** − The Salt Virt Cloud Controller capability is used for automation.
- **Infrastructure as data, not code** − Salt provides a simple deployment, model driven configuration management and command execution framework.

Introduction to ZeroMQ

Salt is based on the **ZeroMQ** library and it is an embeddable networking library. It is lightweight and a fast messaging library. The basic implementation is in **C/C++** and native implementations for several languages including **Java** and **.Net** is available.

ZeroMQ is a broker-less peer-peer message processing. ZeroMQ allows you to design a complex communication system easily.

ZeroMQ comes with the following five basic patterns −

- **Synchronous Request/Response** − Used for sending a request and receiving subsequent replies for each one sent.
- **Asynchronous Request/Response** − Requestor initiates the conversation by sending a Request message and waits for a Response message. Provider waits for the incoming Request messages and replies with the Response messages.
- **Publish/Subscribe** − Used for distributing data from a single process (e.g. publisher) to multiple recipients (e.g. subscribers).
- **Push/Pull** − Used for distributing data to connected nodes.
- **Exclusive Pair** − Used for connecting two peers together, forming a pair.

Ansible

Ansible is a radically simple IT automation platform that makes your applications and systems easier to deploy. Avoid writing scripts or custom code to deploy and update your applications — automate in a language that approaches plain English, using SSH, with no agents to install on remote systems.

The primary **features of Ansible** are as follows −

- Simple and easy to learn
- Written in Python
- Agentless
- YAML-based Playbooks
- Ansible galaxy