**PUPPET 4.10.0**

**Pre-Requisite**

RAM - 3 GB

**Starting internet in centos7**

1. vi /etc/sysconfig/network-scripts/ifcfg-enp0s3

TYPE=Ethernet

BOOTPROTO=dhcp

DEFROUTE=yes

PEERDNS=yes

PEERROUTES=yes

IPV4\_FAILURE\_FATAL=no

IPV6INIT=yes

IPV6\_AUTOCONF=yes

IPV6\_DEFROUTE=yes

IPV6\_PEERDNS=yes

IPV6\_PEERROUTES=yes

IPV6\_FAILURE\_FATAL=no

IPV6\_ADDR\_GEN\_MODE=stable-privacy

NAME=enp0s3

UUID=b51d94a8-944e-4721-9d24-6eabce667c66

DEVICE=enp0s3

ONBOOT=yes

1. service network restart
2. yum install net-tools

**INSTALLING PUPPET 4.10.0**

1. Get the PuppetLabs repository rpm and install it

sudo rpm -ivh <https://yum.puppetlabs.com/puppetlabs-release-pc1-el-7.noarch.rpm>

1. Install the Puppet server using below command

sudo yum -y install puppetserver

1. edit the below file

vi /etc/sysconfig/puppetserver

make changes in line JAVA\_ARGS="-Xms2g -Xmx2g // REMOVE FROM LAST “ ”.

1. systemctl start puppetserver
2. systemctl enable puppetserver
3. /opt/puppetlabs/bin/puppet –version
4. ps -ef | grep puppetserver
5. Configure the puppet master server "puppet"

**vi /etc/puppetlabs/puppet/puppet.conf** is the Puppet Configuration file, edit the file to define the friendly dns hostnames and certificate name for the puppet master node in the [main] section

[main]

dns\_alt\_names = paras.master.net

certname=paras.master.net

**Setup a FQDN for your server**

1. vi /etc/hosts

10.202.62.54 paras.master.net

1. vi /etc/hostname

paras.master.net

1. reboot
2. systemctl stop firewalld ; systemctl disable firewalld

**INSTALLING FOREMAN 1.14**

1. Configure EPEL repositories

rpm -ivh <https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm>

1. Configure Foreman repositories

rpm -ivh <http://yum.theforeman.org/releases/1.14/el7/x86_64/foreman-release.rpm>

1. Download Foreman installer

yum -y install foreman-installer

1. Run the Foreman installer to start installing Foreman

foreman-installer

1. Once the installation is completed, you will see an output like below where you would find the initial username and password to access the Foreman



1. Do following settings in windows host file ( in desktop )

C:\Windows\System32\Drivers\etc\hosts

10.202.62.54 paras.master.net

**Configuring Master Agent relationship in PUPPET 4.10.0**

1. Installation of Puppet master is performed from point 4.
2. Install Agent configuration on Agent Node.
3. Enable the official Puppet Labs collection repository with this command:

sudo rpm -ivh <https://yum.puppetlabs.com/puppetlabs-release-pc1-el-7.noarch.rpm>

1. Install the puppet-agent package

sudo yum -y install puppet-agent

1. Configure the puppet agent on the client node "Agent"

Again **, vi /etc/puppetlabs/puppet/puppet.conf** is the Puppet Configuration file, edit the file to define the puppet master node "puppet" in the [agent] section.

We need to write there as that is empty file.

[main]

server = paras.master.net

[agent]

certname = paras.agent.net

1. service puppet restart
2. It will automatically send certificate to be signed from master.
3. Go to master node and run following commands.
4. sudo /opt/puppetlabs/bin/puppet cert list –all

"paras.agent.net" (SHA256) 98:90:E3:A4:7F:9A:F5:69:05:A7:5D:E5:FF:A2:1C:AF:E3:0C:66:97:A8:EE:48:C7:73:C4:41:D8:09:F6:E2:FE

+ "paras.master.net" (SHA256) 33:D1:80:C8:35:78:95:77:B5:9C:C3:38:66:08:4B:E5:FE:0B:F1:EA:E3:5E:1C:49:98:58:07:04:B4:CC:45:FD (alt names: "DNS:puppet", "DNS:paras.master.net")

+ = cert sign.

1. puppet cert sign "paras.agent.net"

Signing Certificate Request for:

"paras.agent.net" (SHA256) 98:90:E3:A4:7F:9A:F5:69:05:A7:5D:E5:FF:A2:1C:AF:E3:0C:66:97:A8:EE:48:C7:73:C4:41:D8:09:F6:E2:FE

Notice: Signed certificate request for paras.agent.net

Notice: Removing file Puppet::SSL::CertificateRequest paras.agent.net at '/etc/puppetlabs/puppet/ssl/ca/requests/paras.agent.net.pem'

1. sudo /opt/puppetlabs/bin/puppet cert list –all

+ "paras.agent.net" (SHA256) CD:66:B5:0D:80:7B:74:EE:F7:70:11:8F:CC:83:82:24:17:99:54:54:DB:51:C2:7F:72:1C:3F:58:4C:A8:37:E1

+ "paras.master.net" (SHA256) 33:D1:80:C8:35:78:95:77:B5:9C:C3:38:66:08:4B:E5:FE:0B:F1:EA:E3:5E:1C:49:98:58:07:04:B4:CC:45:FD (alt names: "DNS:puppet", "DNS:paras.master.net")

**Setting Up Environment DEV in PUPPET.10.0**

1. By default puppet runs on “production” environment.
2. cd /etc/puppetlabs/puppet/

auth.conf autosign.conf foreman.yaml hiera.yaml node.rb puppet.conf ssl/

1. cd /etc/puppetlabs/code/
2. ls

environments modules

1. cd environments
2. ls

common development production

1. mkdir dev
2. ls

common dev development production

1. ls production

environment.conf hieradata manifests modules

1. On agent node do the following

puppet agent –t

Info: Using configured environment 'production'

Info: Retrieving pluginfacts

Info: Retrieving plugin

Info: Caching catalog for paras.agent.net

Info: Applying configuration version '1494224672'

Notice: Applied catalog in 0.01 seconds

1. Open vi /etc/puppetlabs/puppet/puppet.conf and add the new environment

[main]

server = paras.master.net

**environment = dev**

[agent]

certname = paras.agent.net

1. service puppet restart
2. puppet agent –t and will get following error

Warning: Unable to fetch my node definition, but the agent run will continue:

Warning: Find /puppet/v3/node/paras.agent.net?environment=dev&configured\_environment=dev&transaction\_uuid=b58cd... resulted in 404 with the message: {"message":"Not Found: Could not find environment 'dev'","issue\_kind":"RUNTIME\_ERROR"}

Info: Retrieving pluginfacts

Error: /File[/opt/puppetlabs/puppet/cache/facts.d]: Could not evaluate: Could not retrieve information from environment dev source(s) puppet:///pluginfacts

Info: Retrieving plugin

Error: /File[/opt/puppetlabs/puppet/cache/lib]: Could not evaluate: Could not retrieve information from environment dev source(s) puppet:///plugins

Error: Could not retrieve catalog from remote server: Find /puppet/v3/catalog/paras.agent.net?environment=dev&facts\_format=pson&facts=%257B%2522name%2522%25... resulted in 404 with the message: {"message":"Not Found: Could not find environment 'dev'","issue\_kind":"RUNTIME\_ERROR"}

Warning: Not using cache on failed catalog

Error: Could not retrieve catalog; skipping run

1. Hence go to GUI of PUPPET ( FOREMAN ) and set environment manually.
2. Go to Configure > Environments > New Puppet Environment > Hosts > All Hosts > Edit agent > Set Environment
3. Go to agent CLI node.
4. puppet agent –t

Info: Using configured environment 'dev'

Info: Retrieving pluginfacts

Info: Retrieving plugin

Info: Caching catalog for paras.agent.net

Info: Applying configuration version '1494224953'

Notice: Applied catalog in 0.02 seconds

**Installing Tomcat (httpd) on PUPPET 4.10.0**

1. The client configurations are stored in files called manifests, in this file, user defines system resources and their state, either using Puppet’s declarative language or a Ruby DSL (domain-specific language)
2. Creating first manifest

The client configurations are stored in files called manifests, in this file, user defines system resources and their state, either using Puppet’s declarative language or a Ruby DSL (domain-specific language)

vi /etc/puppetlabs/code/environments/production/manifests/site.pp

Add below content in to the manifest file

node '**paras.agent.net**' {

              package { '**httpd**' :

              ensure => **installed**,

                       }

}

If you see the configuration file, it is very easy to understand. First line tells that we need to install this configuration on a client machine “paras.agent.net”. You can also define multiple nodes in the node statement by comma-separated list of node names.Next two lines tell the puppet that we need to ensure httpd is installed on a client machine.

1. Applying configuration on agent node.

Puppet clients pull the configurations from the puppet master once every 30 minutes, but you can pull it immediately by either restarting puppet client or running below command.

puppet agent --test

You can use below manifest incase If you want to start the apache after the installation and also want it to start automatically on system startup.

node '**paras.agent.net**' {

              package { '**httpd**' : # Package Name

              ensure => **installed**, # Install the package

                       } **->** # Order of the execution, service will be started after the installation

              service {'**httpd**': # Name of the service

              ensure => **running**, # Start the apache service

              enable => **true**, # Start on system boot

                       }

}

**MODULES IN PUPPET 4.10.0**

Individual Puppet manifest files may be good enough for simple tasks. but, when you have a workflow to automate, it is better to create and use a Puppet module instead.

A Puppet module is just a collection of manifests with data files that those manifests require, It allows us to reuse configurations, rather than having to write configurations for every node; we can store them as a module and call them whenever we require.

To write your own module, you need to create a subdirectory under the modules directory with the module name. Then, create a manifest file called init.pp, that should contain a class with the same name as the subdirectory you created just now.

Lets say, you want to place custom apache configuration file (Virtual Hosts), which will have specific options you choose. For this purpose, we will create the module and install the configurations on the client machine (paras.agent.netl).

1. On the Puppet master, make sure you are in the modules directory:  **cd /etc/puppet/modules/**. Then run **mkdir -p httpd\_paras/manifests** and **mkdir -p httpd\_paras/files** to create the new module directory and manifests with its data directory.

cd /etc/puppetlabs/code/modules/

mkdir -p httpd\_paras/manifests

mkdir -p httpd\_paras/files

vi httpd\_paras/manifests/init.pp

Add below content into init.pp file.

class **httpd\_paras** {

    package { '**httpd**':

      ensure => **present**,

    } ->

    file { "**/var/www/paras**": # Creating Document Root

      ensure => "directory",

      owner  => "apache",

      group  => "apache",

      mode   => ‘750’,

    } ->

 file { '**/var/www/paras/index.html**': # Creating Index file

     ensure  => file,

     content => "Index HTML Is Managed By Puppet",

     mode    => '0644',

   } ->

    file { '**/etc/httpd/conf.d/custom\_paras.conf**': # Path to the file on client machine

      ensure => file,

      mode   => '0600',

      source => '**puppet:///modules/httpd\_paras/custom\_paras.conf**', # Path to the custom file on puppet server

    } **~>**

    service { 'httpd':

      ensure => running,

      enable => true,

    }

}

**~> (notification arrow)** — Causes the resource on the left to be applied first, and sends a refresh event to the resource on the right if the left resource changes, meaning that the apache service is to be restarted and enabled to start automatically if there is any changes in the custom apache configuration file.

1. Create a custom apache configuration file, which is having an apache alias. This custom conf file should be created under **files** directory of the created module (**httpd\_paras**).

echo "alias /paras /var/www/paras" > /etc/puppet/modules/httpd\_paras/files/custom\_paras.conf

**Note**: If you see the above command, “**custom\_paras.conf**” file is created under the “**/etc/puppetlabs/code/modules/httpd\_paras/files**“, but where in “**init.pp**” of the “**httpd\_paras**” module; the path to the source file does not have parent directory “**files**” in the puppet url, it is because the Puppet fileserver takes those “**puppet:///**” url and inserts an implicit “**files**” directory[ so you don’t need to specify that in the url]  
url like,

puppet:///modules/httpd\_paras/custom\_paras.conf

gets translated on the filesystem into,

/etc/puppetlabs/code/modules/httpd\_paras/**files**/custom\_paras.conf

1. Its time to call the new module in our main manifest file,

vi /etc/puppetlabs/code/environments/production/manifests/site.pp

Remove everything in that file because its been entered in init.pp file otherwise will give duplication error.

Add the following and save.

node 'paras.agent.net' {

include **httpd\_paras** # This should match with the name of module directory

}

1. On your client machine, run following command to pull the configurations from puppet server.

puppet agent --test

1. Now access your client web server (<http://paras.agent.net>)

**INCLUDING TWO .PP FILES (MANIFEST FILES) IN SINGLE MODULE**

1. Under manifest create two files httpd\_test.pp and service\_paras.pp.

vi /etc/puppetlabs/code/modules/httpd\_paras/manifests/httpd\_test.pp

vi /etc/puppetlabs/code/modules/httpd\_paras/manifests/service\_paras.pp

1. In httpd\_test.pp declare class httpd\_paras::httpd\_test.pp and under it write as follows.

**class httpd\_paras::httpd\_test {**

**package { 'httpd':**

**ensure => present,**

**} ->**

**file { "/var/www/paras": # Creating Document Root**

**ensure => "directory",**

**owner => "apache",**

**group => "apache",**

**mode => '750',**

**} ->**

**file { '/var/www/paras/index.html': # Creating Index file**

**ensure => file,**

**content => "Index HTML Is Managed By Puppet",**

**mode => '0644',**

**} ->**

**file { '/etc/httpd/conf.d/custom\_paras.conf': # Path to the file on client machine**

**ensure => file,**

**mode => '0600',**

**source => 'puppet:///modules/httpd\_paras/custom\_paras.conf', # Path to the custom file on puppet server**

**}**

**}**

1. In service\_paras.pp declare class httpd\_paras::service\_paras.pp and under it write as follows

**class httpd\_paras::service\_paras {**

**service { 'httpd':**

**ensure => running,**

**enable => true,**

**}**

**}**

1. Now in init.pp file remove all and call classes formed above.

class httpd\_paras {

include httpd\_paras::httpd\_test

include httpd\_paras::service\_paras

}

1. On site.pp it will remain same as we have to specify same module.
2. Now go on agent node and run command.

puppet agent –test