**Puppet Configuration:**

port 8140.

**RHEL/CentOS 7**

# rpm -ivh http://yum.puppetlabs.com/puppetlabs-release-el-7.noarch.rpm

**RHEL/CentOS 6**

# rpm -ivh http://yum.puppetlabs.com/puppetlabs-release-el-6.noarch.rpm

### Installing and Upgrading Puppet on the Master Server :

# yum install puppet-server facter

**/etc/init.d/puppetmaster 🡪 Donot start now**

 Next, run the following command to upgrade Puppet to most newest version.

# puppet resource package puppet-server ensure=latest /etc/init.d/puppetmaster restart

Before starting edit puppet.conf to certname = server FQDN

### Installing and Upgrading Puppet on Agent Node

# yum install puppet # puppet resource package puppet ensure=latest # /etc/init.d/puppet restart

Before starting edit puppet.conf file 🡪 server = puppet server FQDN

Now lets connect our puppet agent to puppet master server for the first time. And see what happens.

[?](http://www.slashroot.in/puppet-tutorial-configuring-puppet-agent)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | [root@slashroot2 ~]# puppet agent --server slashroot1.slashroot.in --no-daemonize --verbose  info: Creating a new SSL key for slashroot2.slashroot.in  warning: peer certificate won't be verified in this SSL session  info: Caching certificate for ca  warning: peer certificate won't be verified in this SSL session  warning: peer certificate won't be verified in this SSL session  info: Creating a new SSL certificate request for slashroot2.slashroot.in  info: Certificate Request fingerprint (md5): 59:7A:AE:2C:7B:15:DA:E5:A8:14:7D:FF:1F:5B:7A:66  warning: peer certificate won't be verified in this SSL session  warning: peer certificate won't be verified in this SSL session  warning: peer certificate won't be verified in this SSL session  warning: peer certificate won't be verified in this SSL session  <div style="display:none; visibility:hidden;" id="cosk"></div>  notice: Did not receive certificate |

As shown in the above example you can see that, an SSL key is made for this agent machine and is waiting for the corresponding certificate to be signed by the puppet master server.

An Important fact to note here is a notice shown in the above command result, which says that "**notice: Did not receive certificate**".

**--server**in the above command specifies the puppet master server hostname

**--no-daemonize** tells the puppet agent to not to run as a daemon, and also output the messages to the screen. If you run puppet agent without this option, then you will not get the messages on the screen.

**Note:** If you do not specify the option --server, puppet agent will look for a host named "puppet". This is the main reason of keeping the puppet master hostname as puppet.

The ssl certificate signing is done only the first time an agent connects to the server.

The notice message(**notice: Did not receive certificate**)will keep on coming on the screen until the certificate request is signed by the puppet master.

## 

## How to Sign the SSL certificate from puppet Master?

Now as the client node (slashroot2) is waiting for its certificate to be signed, lets go and sign the certificate request from slashroot1(our puppet master server)

On your puppet master run the below command to show the certificate signing requests.

[root@slashroot1 ~]# puppetca --list

slashroot2.slashroot.in (59:7A:AE:2C:7B:15:DA:E5:A8:14:7D:FF:1F:5B:7A:66)

[root@slashroot1 ~]#

**#puppetca --list**command will show you the agent certificate requests that are waiting to be signed.

**#puppet cert list**command will also show you the same thing

Now lets sign the certificate by the following method.

[root@slashroot1 ~]# puppetca --sign slashroot2.slashroot.in

OR # puppet cert sign [agent FQDN]

notice: Signed certificate request for slashroot2.slashroot.in

notice: Removing file Puppet::SSL::CertificateRequest slashroot2.slashroot.in at '/var/lib/puppet/ssl/ca/requests/slashroot2.slashroot.in.pem'

Now from the above output you can clearly see that the puppet master server signed the certificate and also removed the old certificate signing request.

Now as soon as the certificate gets signed from the master server you will get the below message on the puppet agent's screen(because we ran puppet agent command with --no-daemonize option on our agent). 

notice: Did not receive certificate

warning: peer certificate won't be verified in this SSL session

notice: Did not receive certificate

warning: peer certificate won't be verified in this SSL session

info: Caching certificate for slashroot2.slashroot.in

notice: Starting Puppet client version 2.7.9

info: Caching certificate\_revocation\_list for ca

info: Caching catalog for slashroot2.slashroot.in

info: Applying configuration version '1355395673'

info: Creating state file /var/lib/puppet/state/state.yaml

notice: Finished catalog run in 0.14 seconds

Now what does that message mean?

It means that our puppet agent got a signed certificate and the certificate is cached. Also the agents tells us that its applying a configuration version number "1355395673" based on the catalog given by the master server.  
  
From now onwards we can restart and stop our puppet agent whenever required.

Note: Keep all the client nodes and the puppet server synchronized with one single ntp source. Because ssl connection rely heavily on time being synchronized.  
  
We ran the command **#puppet agent --server slashroot1.slashroot.in --no-daemonize --verbose**, just for showing the output on the screen as example.In normal cases you can add the puppet server address in the puppet.conf file of your agent machine.

So on our agent we will add server address in the [main] section as shown below.

**server=slashroot1.slashroot.in**  
  
After adding this server option in puppet.conf file simply restarting puppet agent will start it as a daemon. Which will periodically fetch data from the master server.

You can start/restart your puppet agent using the below commands.

**/etc/init.d/puppet start**

**How do I change the time interval for a client to fetch it’s configuration from the server ?**  
Add “runinterval = 3600 “ under [main] section in “/etc/puppet/puppet.conf” on the client.

Time is in seconds.

**Installing apache web server with puppet**  
Although puppet server configuration is stored in “/etc/puppet/puppet.conf”, client configurations are stored in files called manifests.

#On the puppet server run

[user@puppet ~]# sudo vim /etc/puppet/manifests/site.pp

node ‘vps.client.com’ {

package { ‘httpd’ :

ensure => installed,

}

}

The configuration is pretty self explanatory, the first line indicates that we need to install this configuration on a client machine with the hostname ‘vps.client.com’. If you want to apply the configuration to the puppet server then replace ‘vps.client.com’ with ‘default’ .  
Read node definitions for multiple node configurations.

The next two lines tell puppet that we need to ensure that the apache web server is installed. Puppet will check if apache is installed and if not, install it.

Think of a “package” as an object, “httpd” as the name of the object and “ensure => present” as the action to be performed on the object.

So if I wanted **puppet to install a mysql database server**, the configuration would be

node ‘vps.client.com’ {

package { ‘mysql-server’ :  
ensure => installed,  
}  
}

The puppet server will compile this configuration into a catalog and serve it to a client when a request is sent to it.

**How do I pull my configuration to a client immediately?**  
Puppet client’s usually pull configuration once every 30 minutes, But you can pull a configuration immediately buy running “service puppet restart or the following command.

[user@puppet ~]# sudo puppet agent --test

**What if I wanted puppet to add a user ‘Tom’?**  
Then the object would be user, the name of the object would be ‘tom’ and the action would be ‘present’.

node ‘vps.client.com’ {

user { ‘tomr’ :

ensure => present,

}

}

In puppet terms, these objects are known as **Resources**, the name of the objects are **Titles** and the actions are called **Attributes**.

Puppet has a number of these resources to help ease your automation, You can read about them at http://docs.puppetlabs.com/references/latest/type.html

**How to ensure a service is running with puppet?**  
Once you have package like apache installed, you will want to ensure that it is running. On the command line you can do this with the service command, However in puppet you will need to use the manifest file and add the configuration as follows.

node ‘vps.client.com’ {

package { ‘httpd’ :

ensure => installed,

}

->

service { ‘httpd’ : #Our resource and it’s title

ensure => running, #Action to be performed on resource or attribute

enable => true, # Start apache at boot

}

}

Now you must have noticed I have added an “->” symbol. This is because Puppet is not particular about ordering, But we want the service command to run only after apache is installed and not before, hence I have added the arrow symbol which tells Puppet to run only after “httpd” is installed.  
To know more about puppet ordering read.

**How to automate installation of predefined conf files?**  
You may want to have a customised apache conf file for this client, which will have the vhost entry and other specific parameters you choose. In this case we need to use the file resource.

Before we go into the configuration, you should know how **puppet serves files**. A **Puppet server provides access to custom files via mount points.** One such mount point by default is the modules directory.  
The **modules directory** is where you would add your modules. Modules make it easier to reuse configurations, rather than having to write configurations for every node we can store them as a module and call them whenever we like.

In order to write a module, you need to create a subdirectory inside the modules directory with the module name and create a manifest file called init.pp which should contain a class with the same name as the subdirectory.

[user@puppet ~]# cd /etc/puppet/modules

[user@puppet ~]# mkdir httpd

[user@puppet ~]# mkdir -p httpd/manifests httpd/files

[user@puppet ~]# vim httpd/manifests/init.pp

class httpd { #Same name as our Sub Directory

package { 'httpd':

ensure => present,

}

->

file {'/etc/httpd/conf/httpd.conf': #Path to file on the client we want puppet to administer

ensure => file, #Ensure it is a file,

mode => 0644, #Permissions for the file

source => 'puppet:///modules/httpd/httpd.conf', #Path to our customised file on the puppet server

}

->

service { 'httpd':

ensure => running,

enable => true,

subscribe => File['/etc/httpd/conf/httpd.conf'] # Restart service if any any change is made to httpd.conf

}

}

You need to add your custom httpd.conf file in the files subdirectory located at “/etc/puppet/modules/httpd/files/”

To understand the how the URI to the source attribute works read http://docs.puppetlabs.com/guides/file\_serving.html

Now call the module in our main manifest file.

[user@puppet ~]#sudo vim /etc/puppet/manifests/site.pp

node ‘vps.client.com’ {

include httpd

}

Incase you need a Web interface to  Manage your Linux Servers then read my tutorial [Using Foreman, an Opensource Frontend for Puppet](http://techarena51.com/index.php/using-foreman-opensource-frontend-puppet/)

**Update**: For more Automation and other System Administration/Devops Guides see <https://github.com/Leo-G/DevopsWiki>  
**Puppet FAQ**  
**How do I change the time interval for a client to fetch it’s configuration from the server ?**  
Add “runinterval = 3600 “ under [main] section in “/etc/puppet/puppet.conf” on the client.

Time is in seconds.

**How do I install modules from**[**puppet forge**](https://forge.puppetlabs.com/)**?**

[user@puppet ~]#sudo puppet module install "full module name"

#Example

[user@puppet ~]#sudo puppet module install puppetlabs-mysql

**rpm -Uvh http://yum.spacewalkproject.org/2.0/RHEL/6/x86\_64/spacewalk-repo-2.0-3.el6.noarch.rpm**

**rpm -Uvh http://mirror.muntinternet.net/pub/epel/6/i386/epel-release-6-8.noarch.rpm**

**wget http://www.jpackage.org/jpackage50.repo**

**/dev/mapper/group-volume1**