
Puppet Open Source Master and Agent setup instructions:

As pre-requisites, install NTP

Because it acts as a certificate authority for agent nodes, the Puppet master server must maintain accurate system time to avoid potential problems when it issues agent certificates. Certificates can appear to be expired if there are time discrepancies.

```
$ timedatectl list-timezones
```

This will give you a list of the timezones available for your server. When you find the region/timezone setting that is correct for your server, set it with this command (substitute your preferred region and timezone):

```
$ sudo timedatectl set-timezone Asia/India
```

Install NTP via apt-get with these commands:

```
$ sudo apt-get update
$ sudo apt-get -y install ntp
```

Open `ntp.conf` for editing, that is located at `/etc/ntp.conf`

```
$ sudo vi /etc/ntp.conf
```

Add the time servers from the NTP Pool Project page to the top of the file (replace these with the servers of your choice):

```
server 2.in.pool.ntp.org
server 1.asia.pool.ntp.org
server 2.asia.pool.ntp.org
```

Start NTP to add the new time servers:

```
$ sudo service ntp restart
```

Install Puppet Server

This will install Puppet server application on the **Ubuntu server.**

```
$ cd ~ && wget https://apt.puppetlabs.com/puppetlabs-
release-pcl-trusty.deb
$ sudo dpkg -i puppetlabs-release-pcl-trusty.deb
$ sudo apt-get update
```

Install the `puppetserver` package:

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

By default the PuppetServer service will not be running.

For puppetserver installation on Centos / el,

```
$ cd ~ && wget https://yum.puppetlabs.com/puppet6/puppet6-release-el-7.noarch.rpm
$ cd ~ && wget https://yum.puppetlabs.com/puppet6-release-el-8.noarch.rpm
$ sudo rpm -ivh puppet6-release-el-8.noarch.rpm
$ sudo yum update
$ sudo yum install puppetserver
```

Configure Memory Allocation

By default, Puppet Server is configured to use 2 GB of RAM. You should customize this setting based on how much free memory your master server has, and how many agent nodes it will manage.

Open `/etc/default/puppetserver` using below command.,

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

Then find the `JAVA_ARGS` line, and use the `-Xms` and `-Xmx` parameters to set the memory allocation. For example, if you want to use 3 GB of memory, the line should look like this:

```
JAVA_ARGS="-Xms3g -Xmx3g"
```

Start Puppet Server

Now start Puppet Server with below command:

```
$ sudo systemctl restart puppetserver
```

Next, enable Puppet Server so that it starts when your master server boots:

```
$ export PATH=$PATH:/opt/puppetlabs/bin/puppet
$ sudo puppet resource service puppetserver ensure=running
enable=true
```

Check the puppet server version installed...

```
$ puppet -V
```

Install Puppet Agent

Perform these steps on all of your agent servers.

Enable the official Puppet Labs collection repository with these commands:

```
$ cd ~ && wget https://apt.puppetlabs.com/puppetlabs-release-pcl-trusty.deb
$ sudo dpkg -i puppetlabs-release-pcl-trusty.deb
```

Then install the `puppet-agent` package:

```
$ sudo apt-get update
$ sudo apt-get install puppet-agent
```

On CentOS system use below steps to download and install Puppet-agent,

```
$ wget http://yum.puppetlabs.com/puppetlabs-release-pcl-el-6.noarch.rpm
```

```
$ sudo yum install puppetlabs-release-pcl-el-6.noarch.rpm
```

```
$ sudo yum update
```

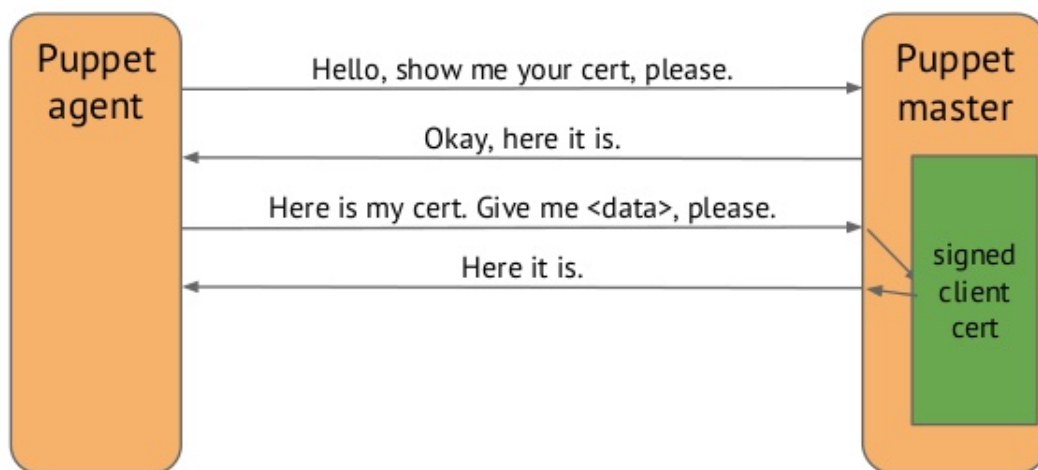
```
$ sudo yum install puppet-agent
```

Now that the Puppet agent is installed, start it with this command and make sure it start every time when the server is restarted.

```
$ sudo /opt/puppetlabs/bin/puppet resource service puppet ensure=running enable=true
```

Sign Certificates on Puppet Master

Puppet mechanics: SSL: the next handshakes



The first time Puppet runs on an agent node, it will send a certificate signing request to the Puppet master. Before Puppet Server will be able to communicate with and control the agent node, it must sign that particular agent node's certificate.

List Current Certificate Requests

On the Puppet master, run the following command to list all unsigned certificate requests:

```
$ sudo /opt/puppetlabs/bin/puppet cert list
```

If you just set up your first agent node, you will see one request. It will look something like the following, with the agent node's hostname:

Output:

```
"node1.autofact.com" (SHA256)
15:90:C2:FB:ED:69:A4:F7:B1:87:0B:BF:F7:DD:B5:1C:33:F7:76:
67:F3:F6:23:AE:07:4B:F6:E3:CC:04:11:4C
```

Note that there is no + in front of it. This indicates that it has not been signed yet.

Signing a certificate Request

To sign a certificate request, use the `puppet cert sign` command, with the hostname of the certificate you want to sign. For example, to sign `host1.nyc3.example.com`'s certificate, you would use the following command:

```
$ sudo /opt/puppetlabs/bin/puppet cert sign
node1.autofact.com
```

You will see the following output, which indicates that the certificate request has been signed:

```
Output:
Notice: Signed certificate request for node1.autofact.com
Notice: Removing file Puppet::SSL::CertificateRequest
node1.autofact.com at
'/etc/puppetlabs/puppet/ssl/ca/requests/node1.autofact.co
m.pem'
```

Incase puppet agent certificate signing request is not generated or not received on Puppet Master, delete existing public and private keys and retry running the agent using `'puppet agent -t'`. Delete the keys using below command,

```
$ find /etc/puppetlabs/puppet/ssl -name <nodename.pem> -delete
```

Incase of puppet agent is running on windows,

On Windows:

```
del "\\etc\\puppetlabs\\puppet\\ssl\\certs\\tomcat-server.pem" /f
```

The Puppet master can now communicate and control the node that the signed certificate belongs to.

If you want to sign all of the current requests, use the `--all` option, like so:

```
$ sudo /opt/puppetlabs/bin/puppet cert sign --all
```

Revoke Certificates

You may want to remove a host from Puppet, or rebuild a host then add it back to Puppet. To do this, you can use the `clean` action:

```
$ sudo /opt/puppetlabs/bin/puppet cert clean hostname
```

The specified host's associated certificates will be removed from Puppet.

View All Signed Requests

If you want to view all of the requests, signed and unsigned, run the following command:

```
sudo /opt/puppetlabs/bin/puppet cert list --all
```

You will see a list of all of the requests. Signed requests are preceded by a + and unsigned requests do not have the +.

Output:

```
+ "puppet" (SHA256)
5A:71:E6:06:D8:0F:44:4D:70:F0:BE:51:72:15:97:68:D9:67:16:
41:B0:38:9A:F2:B2:6C:BB:33:7E:0F:D4:53 (alt names:
"DNS:puppet", "DNS:puppet.autofact.com")
+ "node1.autofact.com" (SHA256)
F5:DC:68:24:63:E6:F1:9E:C5:FE:F5:1A:90:93:DF:19:F2:28:8B:
D7:BD:D2:6A:83:07:BA:FE:24:11:24:54:6A
```

With this your Puppet Server (Master) and Agent setup is complete.

*When the installation completes, you can find the final PE configuration file at
'/etc/puppetlabs/enterprise/conf.d/pe.conf' on puppet.c.prefab-pixel-185310.internal.*

You can find the installer log at /var/log/puppetlabs/installer/installer.log on 35.229.103.142.