**Configure Hostname**

Before starting, you will need to configure /etc/hosts and /etc/hostname file on Server node and agent node, so they can able to communicate with each other.

On the Server node, open /etc/hosts and /etc/hostname file and make the following changes:

sudo nano /etc/hosts

Add the following line at the end of the file:

192.168.0.103 puppet-server

sudo nano /etc/hostname

Change the file as shown below:

puppet-server

Save and close the file when you are finished.

On the Agent node, open /etc/hosts and /etc/hostname file and make the following changes:

sudo nano /etc/hosts

Add the following line at the end of the file:

192.168.0.103 puppet-server

sudo nano /etc/hostname

Change the file as shown below:

puppet-agent

Save and close the file when you are finished.

**Install Puppet**

Puppet server is not available in Ubuntu 16.04 default repository. So you will need to add Puppet Lab repository on both Master and Agent node.

On each node, run the following command to download and install Puppet repository:

wget https://apt.puppetlabs.com/puppetlabs-release-pc1-xenial.deb  
sudo dpkg -i puppetlabs-release-pc1-xenial.deb  
sudo apt-get update -y

Next, install Puppet server package on Master node with the following command:

sudo apt-get install puppetserver -y

After installing the Puppet server, you will need to configure the memory allocation. You are recommended to customize the memory usage depends on how much memory your master node has. You can do this by editing /etc/default/puppetserver file:

sudo nano /etc/default/puppetserver

Change the lines as per your server capacity:

From

JAVA\_ARGS="-Xms2g -Xmx2g -XX:MaxPermSize=256m"

To

JAVA\_ARGS="-Xms512m -Xmx512m"

Save and close the file, then start Puppet server and enable it to start on boot time with the following command:

sudo systemctl start puppetserver  
sudo systemctl enable puppetserver

You can check the status of the Puppet server using the following command:

sudo systemctl status puppetserver

If everything when fine you should see the following output:

● puppetserver.service - puppetserver Service  
 Loaded: loaded (/lib/systemd/system/puppetserver.service; enabled; vendor preset: enabled)  
 Active: active (running) since Sat 2017-10-28 18:47:26 IST; 12min ago  
 Process: 887 ExecStart=/opt/puppetlabs/server/apps/puppetserver/bin/puppetserver start (code=exited, status=0/SUCCESS)  
 Main PID: 963 (java)  
 CGroup: /system.slice/puppetserver.service  
 └─963 /usr/bin/java -Xms256m -Xmx256m -Djava.security.egd=/dev/urandom -XX:OnOutOfMemoryError=kill -9 %p -cp /opt/puppetlabs/server/

**Install Puppet Agent**

Now, your Puppet server is up and running. It’s time to install Puppet agent on Agent node.

Before installing Puppet agent, make sure you have installed Puppet Lab repository on Agent node. Next, install Puppet agent by just running the following command:

sudo apt-get install puppet-agent -y

Once Puppet agent is installed, you will need to edit the puppet configuration file and set puppet master information.

You can do this with the following command:

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

Add the following lines:

[main]  
certname = puppet-agent  
server = puppet-server  
environment = IT

Save and close the file, then start Puppet agent service and enable it to start on boot time with the following command:

sudo systemctl start puppet  
sudo systemctl enable puppet

**Sign the Puppet Agent Certificate on Puppet Server**

When the Puppet runs Agent node first time, it sends a certificate signing request to the Puppet server. In Client-Server architecture, Puppet master server must approve a certificate request for each Agent node to control the Agent node.

On Puppet server, list all unsigned certificate requests with the following command:

sudo /opt/puppetlabs/bin/puppet cert list

You should see the one request with your agent node’s hostname:

"puppet-agent" (SHA256) 7C:28:E8:AF:09:23:55:19:AF:C1:EE:C3:66:F2:02:73:AD:7F:53:17:28:CE:B0:26:AE:C7:6C:67:16:05:6F:2E

Next, sign a certificate request using the following command:

sudo /opt/puppetlabs/bin/puppet cert sign puppet-agent

You should see the following output:

Signing Certificate Request for:  
 "puppet-agent" (SHA256) 7C:28:E8:AF:09:23:55:19:AF:C1:EE:C3:66:F2:02:73:AD:7F:53:17:28:CE:B0:26:AE:C7:6C:67:16:05:6F:2E  
Notice: Signed certificate request for puppet-agent  
Notice: Removing file Puppet::SSL::CertificateRequest puppet-agent at '/etc/puppetlabs/puppet/ssl/ca/requests/puppet-agent.pem'

The Puppet Master server is now able to communicate and control the Agent node. If you want to sign certificate request of multiple nodes at once, then run the following command:

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

Once the Puppet master has signed your Puppet Agent certificate, run the following command on Puppet Agent node to test it:

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

If everything is done correctly, you should see the following output:

Info: Using configured environment 'production'  
Info: Retrieving pluginfacts  
Info: Retrieving plugin  
Info: Caching catalog for puppet-agent  
Info: Applying configuration version '1509200872'  
Notice: Applied catalog in 0.09 seconds

**Configure Puppet Server to Install Apache on Agent Node**

Both Puppet Master and Agent node are now configured and are functional. It’s time to verify Puppet.

To do so, create a manifest file to install Apache web server on Agent node. Manifest is a data file that contains client configurations. By default, manifest file is located at /etc/puppetlabs/code/environments/production/manifests/ directory.

Before proceeding to create a manifest file, you will need to install the puppetlabs-apache module.

On the Puppet master node, run the following command to install the puppetlabs-apache module:

sudo /opt/puppetlabs/bin/puppet module install puppetlabs-apache

You should see the following output:

Notice: Preparing to install into /etc/puppetlabs/code/environments/production/modules ...  
Notice: Downloading from https://forgeapi.puppet.com ...  
Notice: Installing -- do not interrupt ...  
/etc/puppetlabs/code/environments/production/modules  
└─┬ puppetlabs-apache (v2.3.0)  
 ├── puppetlabs-concat (v4.1.0)  
 └── puppetlabs-stdlib (v4.20.0)

Next, create a manifest file on the Puppet master with the following command:

sudo nano /etc/puppetlabs/code/environments/production/manifests/site.pp

Add the following lines:

node 'puppet-agent' {  
 class { 'apache': } # use apache module  
 apache::vhost { 'localhost': # define vhost resource  
 port => '80',  
 docroot => '/var/www/html'  
 }  
}

The above configuration will install the Apache, configure a virtual host called localhost, listening on port 80, and with a document root /var/www/html on Agent node.

Now, on the Agent node, run the following command to retrieve all the configuration from manifest file:

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

If everything is successful, you should see the following output:

Notice: /Stage[main]/Apache/Apache::Vhost[default]/File[15-default.conf symlink]/ensure: created  
Info: /Stage[main]/Apache/Apache::Vhost[default]/File[15-default.conf symlink]: Scheduling refresh of Class[Apache::Service]  
Notice: /Stage[main]/Main/Node[puppet-agent]/Apache::Vhost[localhost]/Concat[25-localhost.conf]/File[/etc/apache2/sites-available/25-localhost.conf]/ensure: defined content as '{md5}05a8b8c6772009021086814bdf8c985e'  
Info: Concat[25-localhost.conf]: Scheduling refresh of Class[Apache::Service]  
Notice: /Stage[main]/Main/Node[puppet-agent]/Apache::Vhost[localhost]/File[25-localhost.conf symlink]/ensure: created  
Info: /Stage[main]/Main/Node[puppet-agent]/Apache::Vhost[localhost]/File[25-localhost.conf symlink]: Scheduling refresh of Class[Apache::Service]  
Info: Class[Apache::Service]: Scheduling refresh of Service[httpd]  
Notice: /Stage[main]/Apache::Service/Service[httpd]: Triggered 'refresh' from 1 events  
Notice: Applied catalog in 53.11 seconds

Congratulations! Apache is now installed and running on the Agent node.