

Lab 10. Puppet Master and Agent



In this lab, we will install Puppet server and agent on Ubuntu 20.04 server. We will setup both master and agent on the same node in this lab. Setup can be done on multiple machines using same steps.

Getting Started

First, you will need to update all packages on Puppet master and Puppet client systems. You can update them by running the following command:

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

Once all the packages are up-to-date, you can proceed to the next step.

Setup Hostname Resolution

Next, you will need to setup hostname on both nodes. So each node can communicate with each other by hostname. You can do it by editing `/etc/hosts` file. Add the following lines on the node:

```
echo "127.0.0.1 puppetmaster puppet" >> /etc/hosts
echo "127.0.0.1 puppetclient" >> /etc/hosts

cat /etc/hosts
```

Save and close the file when you are finished. Then, you can proceed to the next step.

Install Puppet Server

First, you will need to install the Puppet server on the master node. By default, the Puppet package is not available in the Ubuntu 20.04 default repository. So you will need to install the Puppet repository in your server.

First, download the latest version of Puppet with the following command:

```
wget https://apt.puppetlabs.com/puppet5-release-xenial.deb
```

Once the package is downloaded, install it by running the following command:

```
dpkg -i puppet5-release-xenial.deb
```

Once the installation is completed, update the repository and install the Puppet server by running the following command:

```
apt-get update -y

apt-get install puppetserver -y
```

Start the Puppet service with the following command (it will take some time):

```
service puppetserver restart
```

Next, you can verify the status of the Puppet service with the following command:

```
service puppetserver status
```

You should see the following command:

```
* puppetserver is running
```

Once you are finished, you can proceed to the next step.

Install and Configure Puppet Agent

At this point, the Puppet server is installed and configured. Now, you will need to install the Puppet agent on the client node.

First, download and install the Puppet repository with the following command:

```
wget https://apt.puppetlabs.com/puppet5-release-xenial.deb  
  
dpkg -i puppet5-release-xenial.deb
```

Next, update the repository and install the Puppet agent by running the following command:

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

After installing Puppet agent, you will need to edit the Puppet configuration file and define the Puppet master information:

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

Add the following lines:

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

Connect with Puppet Master

On the Puppet agent node, test the Puppet master and agent communication with the following command:

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

```
root@SandboxHost-637503348082775909:~# nano /etc/puppetlabs/puppet/puppet.conf  
root@SandboxHost-637503348082775909:~# /opt/puppetlabs/bin/puppet agent --test  
Info: Creating a new SSL key for puppetclient  
Info: csr_attributes file loading from /etc/puppetlabs/puppet/csr_attributes.yaml  
Info: Creating a new SSL certificate request for puppetclient  
Info: Certificate Request fingerprint (SHA256): 1A:1E:D3:B0:4A:62:07:2A:2E:93:10  
:0D:7C:3B:10:E8:A1:C0:C9:65:DF:87:67:2F:46:69:EF:10:55:E9:91:B7  
Exiting; no certificate found and waitforcert is disabled  
root@SandboxHost-637503348082775909:~#
```

Note that about connection will fail because we have to accept connection request from agent on the master node. At this point, the Puppet agent is installed and configured. Now, you can proceed to the next step.

Sign Puppet Agent Certificate

Puppet uses a client-server architecture so you will need to approve a certificate request for each agent node before it can configure it.

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

```
/opt/puppetlabs/bin/puppet ca list
```

Now, sign the certificate with the following command:

```
/opt/puppetlabs/bin/puppet ca sign puppetclient
```

Connect with Puppet Master

On the Puppet agent node, test the Puppet master and agent communication with the following command:

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

Note: You will get a master `hostname` mismatch error because we set hostname as `puppetmaster`. We need to use exact hostname of the lab environment which is read-only. Copy complete red highlighted hostname displayed by puppet in error message carefully as shown below:

```
root@sandboxhost-637503275972121024:~# puppet agent -t
Info: Loading certificate for sandboxhost-637503275972121024
Error: request https://sandboxhost-637503275972121024:8140/puppet-ca/v1/certificate/sandboxhost-637503275972121024 failed: SSL_connect returned: errno=0 state=error: certificate verify failed
Error: Could not request certificate: Server hostname 'sandboxhost-637503275972121024' did not match server certificate; expected one of [sandboxhost-637503275972121024,caas-b604373a191646c49946bfae5294910a-637503275972121024,caas-b604373a191646c49946bfae5294910a]
```

Edit the Puppet configuration file and update the Puppet master information:

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

Update the following lines:

```
[main]
certname = update-lab-environment-hostname
server = update-lab-environment-hostname
```

```
File Edit Tabs Help
GNU nano 4.8 /etc/puppetlabs/puppetserver/config.yaml
# This file can be used to override the default puppet settings.
# See the following links for more details on what settings are available:
# - https://puppet.com/docs/puppet/latest/config_important_settings.html
# - https://puppet.com/docs/puppet/latest/config_about_settings.html
# - https://puppet.com/docs/puppet/latest/config_file_main.html
# - https://puppet.com/docs/puppet/latest/configuration.html
[master]
vardir = /opt/puppetlabs/server/data/puppetserver
logdir = /var/log/puppetlabs/puppetserver
rundir = /var/run/puppetlabs/puppetserver
pidfile = /var/run/puppetlabs/puppetserver/puppetserver.pid
codedir = /etc/puppetlabs/code

[main]
certname = sandboxhost-637503275972121824.caas-ba64323af9f646c49946bfae5294910a
server = sandboxhost-637503275972121824.caas-ba64323af9f646c49946bfae5294910a
```

Edit the /etc/hosts configuration file:

```
echo "127.0.0.1 update-lab-environment-hostname" >> /etc/hosts

cat /etc/hosts
```

```
File Edit Tabs Help
root@SandboxHost-637503365645530879:~# cat /etc/hosts
127.0.0.1 localhost
127.0.0.1 SandboxHost-637503365645530879
127.0.0.1 sandboxhost-637503365645530879.caas-24463489e1f944de8386e780fa5cc50a

# 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
127.0.0.1 puppetmaster puppet
127.0.0.1 puppetclient
```

Run following command in terminal 2 test the Puppet master and agent communication:

```
puppet agent -t --waitforcert 500000
```

Run following commands in terminal 1 to sign the certificate(if required):

```
puppet ca list

puppet ca sign "your-cert-name"
```

Note:

Now, Puppet master will be able to communicate and control the agent node.

On the Puppet agent node, test the Puppet master and agent communication with the following command:

```
puppet agent -t
```

If everything is fine, you should get the following output:

```
Info: Using configured environment 'production'
Info: Retrieving pluginfacts
Info: Retrieving plugin
Info: Retrieving locales
Info: Caching catalog for puppetclient
Info: Applying configuration version '1599300398'
Notice: Applied catalog in 0.02 seconds
```

puppet apply vs puppet agent -t

Specifying the manifest for Puppet apply

The puppet apply command uses the manifest you pass to it as an argument on the command line:

```
puppet apply /etc/puppetlabs/code/environments/production/manifests/example.pp
```

You can pass Puppet apply either a single .pp file or a directory of .pp files. Puppet apply uses the manifest you pass it, not an environment's manifest.

Specifying the manifest for Puppet master

Puppet master uses the main manifest set by the current node's environment, whether that manifest is a single file or a directory of .pp files.

By default, the main manifest for an environment is `<ENVIRONMENTS`

`DIRECTORY>/<ENVIRONMENT>/manifests`, for example

`/etc/puppetlabs/code/environments/production/manifests`. You can configure the manifest per-environment, and you can also configure the default for all environments.

To check which manifest your Puppet master uses for a given environment, run:

```
puppet config print manifest --section master --environment <ENVIRONMENT>
```

You can also run following command to print central manifest location that master uses:

```
puppet master --configprint manifest
```

Note: Use `puppet agent -t` when you want to run manifest from master node and apply changes to agent node and run `puppet apply` command when you have put the manifest onto the machine itself.

Manifest directory behavior

When the main manifest is a directory, Puppet parses every .pp file in the directory in alphabetical order and evaluates the combined manifest. It descends into all subdirectories of the manifest directory and loads files in depth-first order. For example, if the manifest directory contains a directory named 01, and a file named 02.pp, it parses the files in 01 before it parses 02.pp.

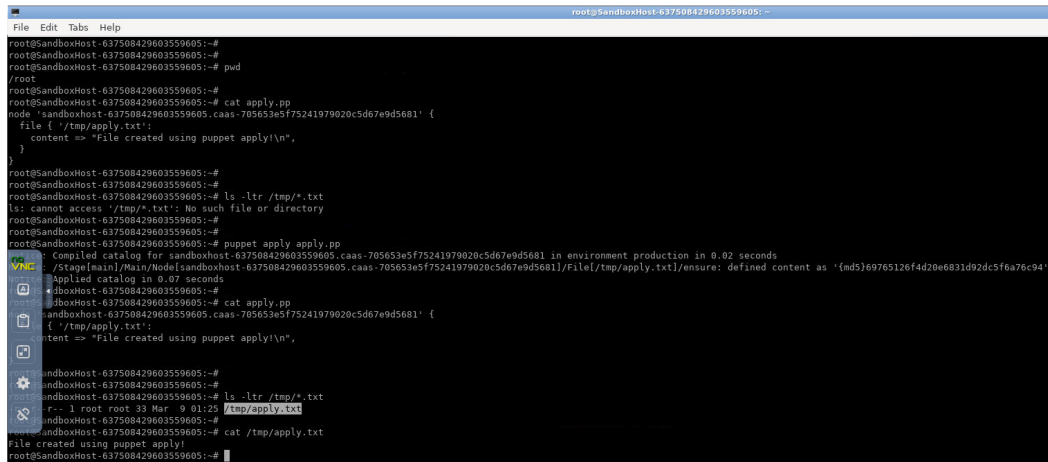
Puppet treats the directory as one manifest, so, for example, a variable assigned in the file 01_all_nodes.pp is accessible in node_web01.pp.

Task 1: Create New File (puppet apply -t)

Create new file '/tmp/apply.txt' by writing puppet manifest and apply to agent node:

Hint:

```
node 'hostname' {  
  
    # write here  
}
```



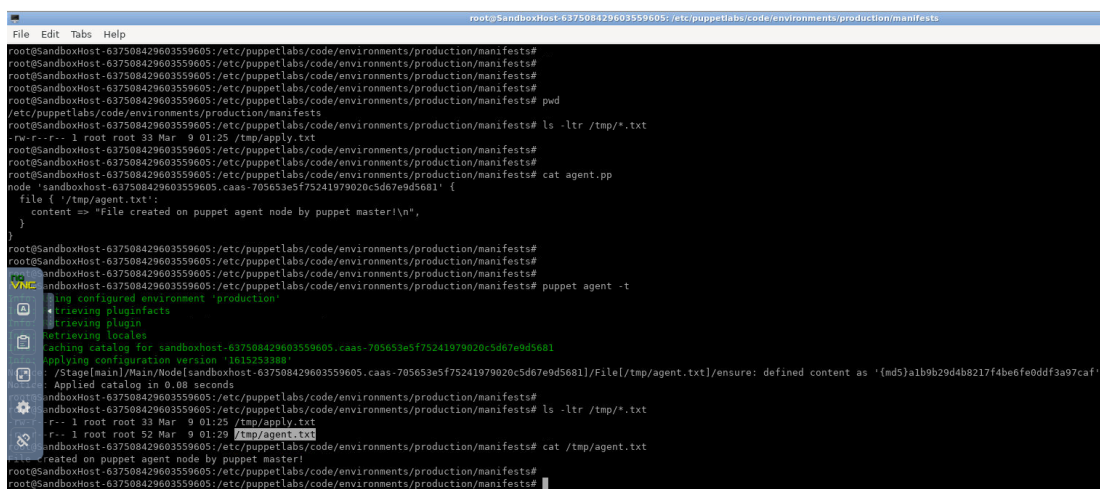
A terminal window showing the execution of a Puppet agent on a node. The user runs `cat apply.pp` and `node 'sandboxhost-637508429603559605.caas-705653e5f75241979020c5d67e9d5681' { file { '/tmp/apply.txt': content => "File created using puppet apply!\n", }`. The agent then applies the catalog, and the user runs `ls -ltr /tmp/.txt` to verify the file creation. The output shows the file `/tmp/apply.txt` was created.

Task 2: Create New File (puppet agent -t)

Create new file `/tmp/agent.txt` by writing puppet manifest and apply to agent node:

Hint: Create `agent.pp` file in `/etc/puppetlabs/code/environments/production/manifests` directory and run puppet agent command.

Verify that manifest was applied by master on agent node and file that has been created: `cat /tmp/agent.txt`



A terminal window showing the execution of a Puppet agent on a node. The user runs `cat /tmp/agent.txt` and `node 'sandboxhost-637508429603559605.caas-705653e5f75241979020c5d67e9d5681' { file { '/tmp/agent.txt': content => "File created on puppet agent node by puppet master!\n", }`. The agent then applies the catalog, and the user runs `ls -ltr /tmp/.txt` to verify the file creation. The output shows the file `/tmp/agent.txt` was created.

Conclusion

Congratulations! you have successfully installed and configured Puppet master and agent on Ubuntu server. You can now easily add multiple agents and manage them easily with Puppet.