Automate node classification using Puppet's external node classifier (ENC)

Table of Contents

Introduction

An **External Node Classifier (ENC)** is an external tool or script that integrates with Puppet to classify nodes based on their facts or other dynamic information. Instead of defining node configurations directly in `site.pp`, an ENC allows you to use external data sources (such as databases, APIs, or custom scripts) to classify nodes.

By using an ENC, you can automate the process of node classification, dynamically assign roles, environments, and other configurations based on external inputs, and ensure that your Puppet infrastructure scales efficiently.

Problem Statement

In large-scale infrastructures with hundreds or thousands of nodes, manually classifying each node and assigning roles via `site.pp` or `puppet.conf` becomes impractical. An ENC allows for automation by dynamically classifying nodes based on their attributes, such as their hostname, IP address, or any other custom facts.

Prerequisites

Completion of all previous lab guides (up to Lab Guide-06) is required before proceeding with Lab Guide-07.

Software Requirements

- **Puppet Master** installed and running.
- **External Node Classifier (ENC)** script (written in Ruby, Python, or another language).

- **Puppet 3.8.7** for compatibility.
- **Node Facts** (e.g., hostname, role, environment).

Hardware Requirements

- Puppet Master machine: Minimum 1GB RAM, 2 CPUs, and 10GB Disk.
- Puppet Agent machine(s): Minimum 512MB RAM, 1 CPU, 5GB Disk.

Implementation Steps

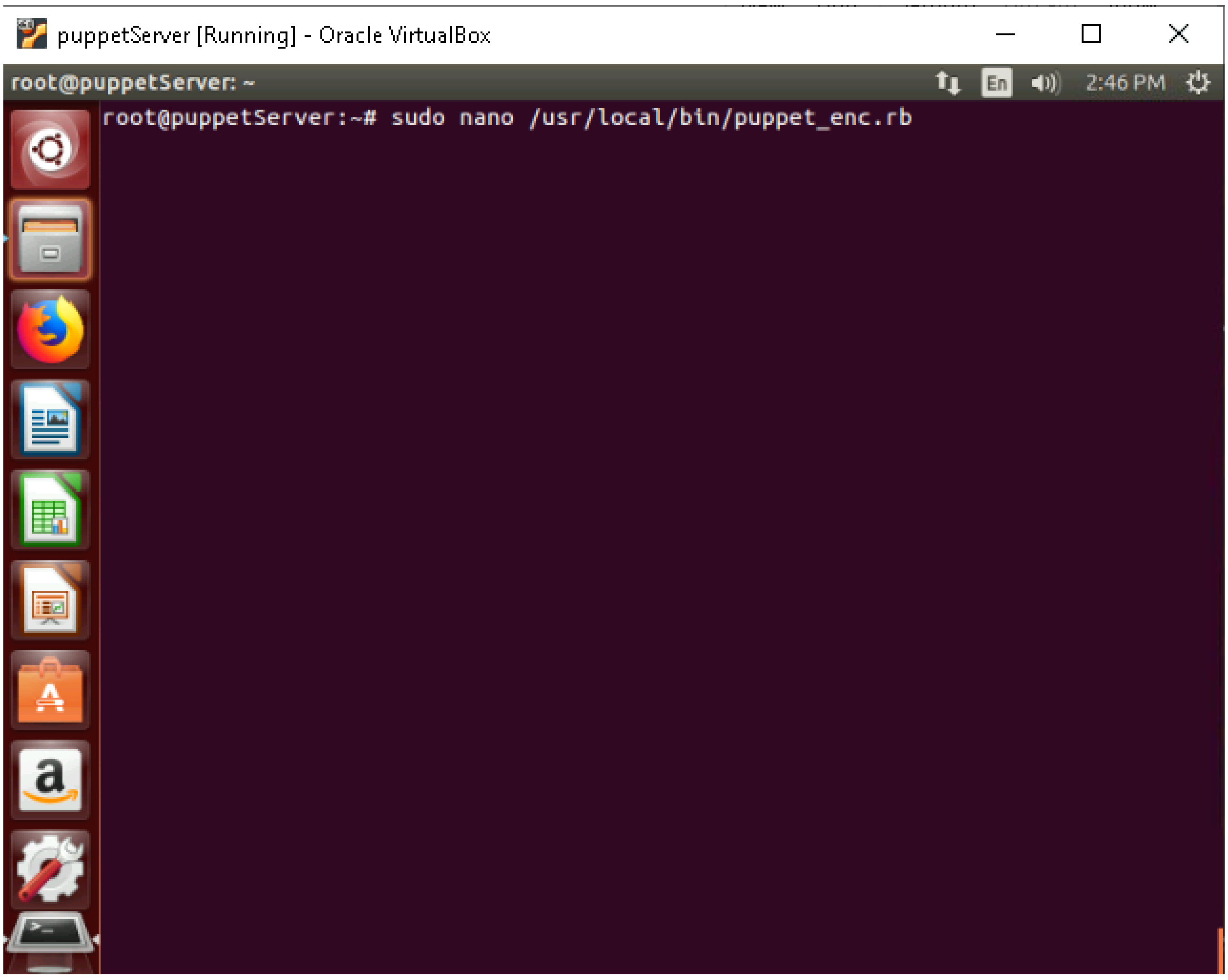
Step 1: Set Up an External Node Classifier (ENC)

1. **Create a Script for ENC**:

   The ENC script is a simple program that returns a JSON or YAML document describing the node's classification. This script will query your data source (e.g., an API, a database, or a set of facts) to determine the correct classification for each node.

   Example in **Ruby** (stored in `/usr/local/bin/puppet_enc.rb`):

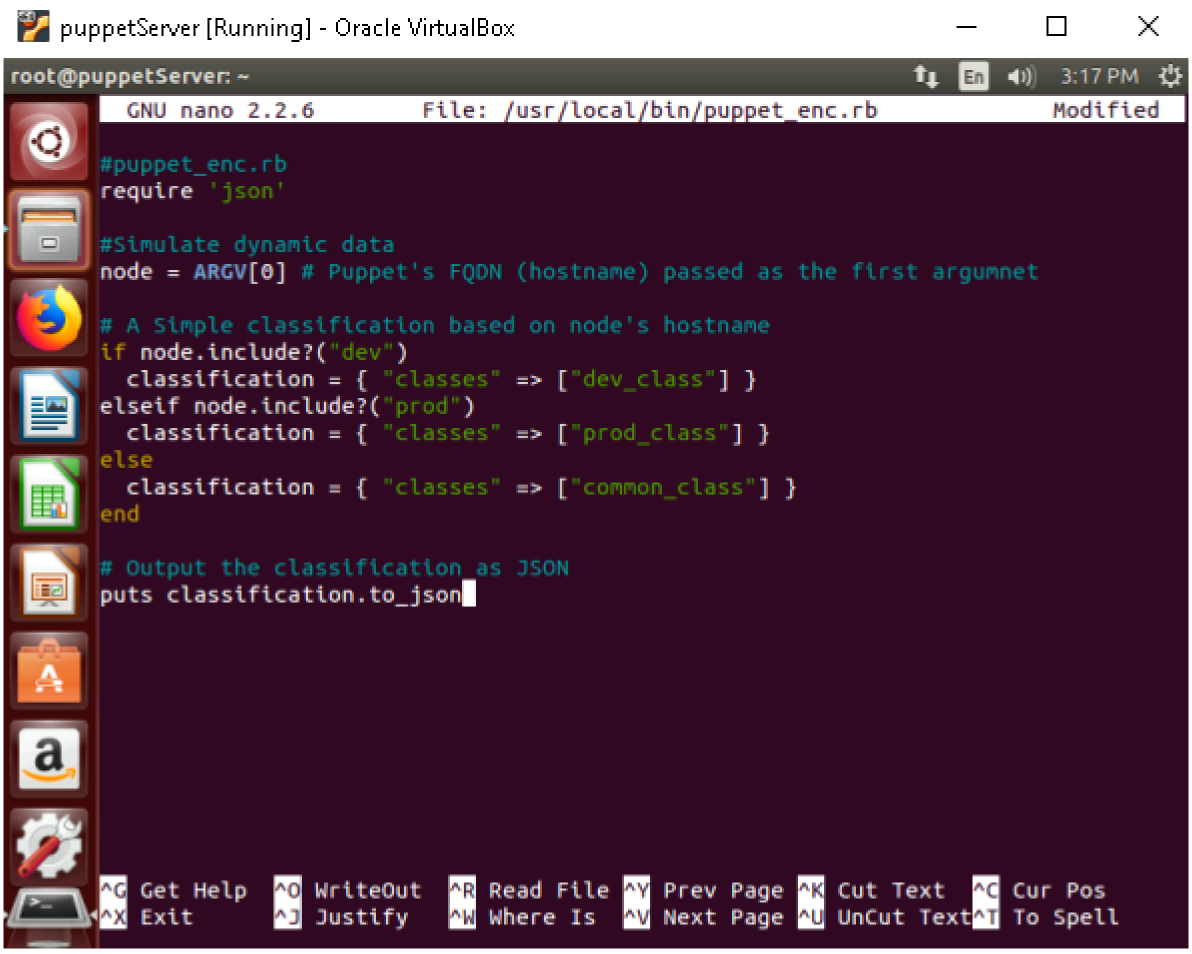
   ```
   sudo nano /usr/local/bin/puppet_enc.rb
   ```



```
root@puppetServer:~# sudo nano /usr/local/bin/puppet_enc.rb
```

Add the following content to the script:

```
#!/usr/bin/env ruby

# puppet_enc.rb
require 'json'

# Simulate dynamic data
node = ARGV[0]  # Puppet's FQDN (hostname) passed as the first argument

# A simple classification based on node's hostname
if node.include?("dev")
  classification = { "classes" => ["dev_class"] }
elsif node.include?("prod")
  classification = { "classes" => ["prod_class"] }
else
  classification = { "classes" => ["common_class"] }
end

# Output the classification as JSON
puts classification.to_json
```

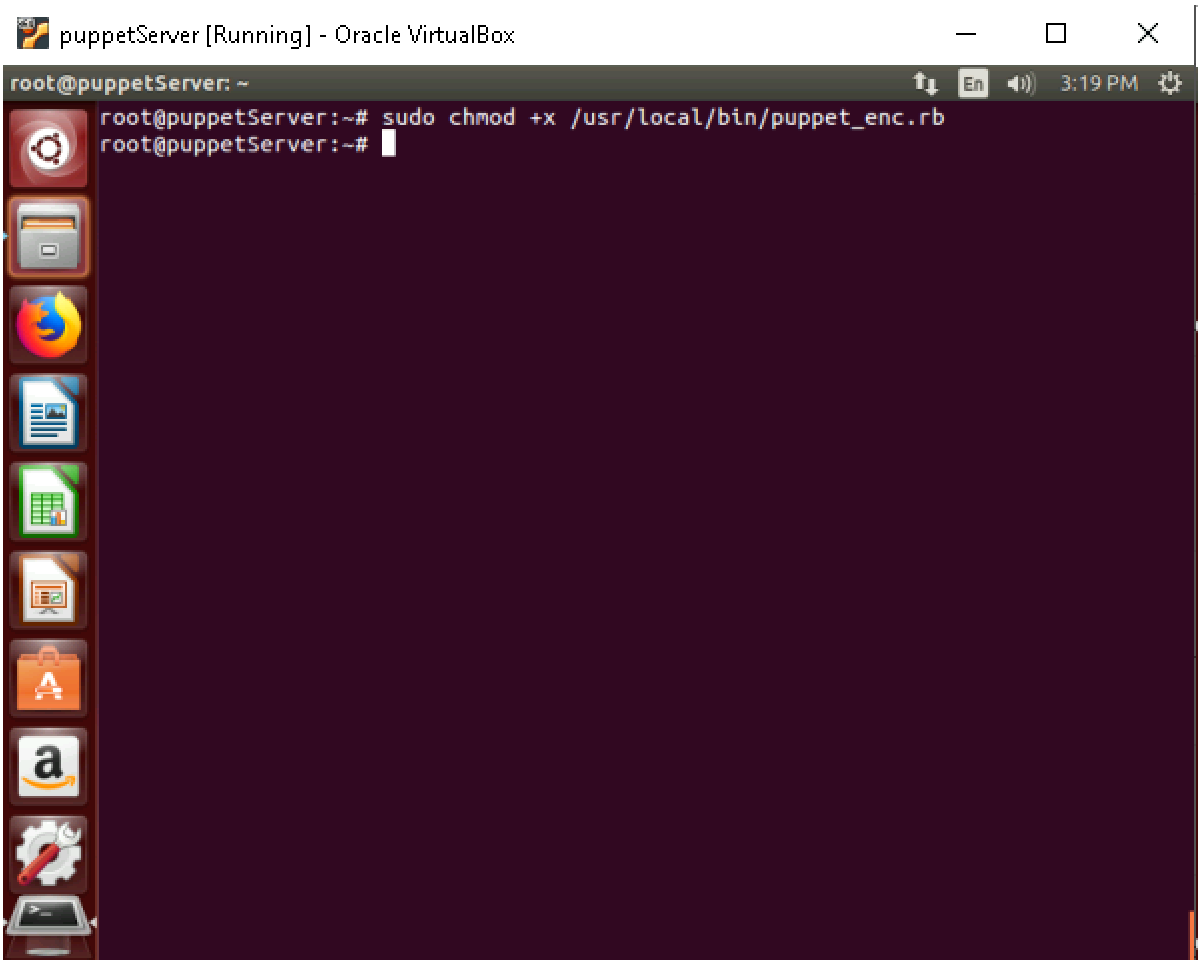

puppetServer [Running] - Oracle VirtualBox

```
root@puppetServer: ~                                            3:17 PM
  GNU nano 2.2.6          File: /usr/local/bin/puppet_enc.rb           Modified

#puppet_enc.rb
require 'json'

#Simulate dynamic data
node = ARGV[0] # Puppet's FQDN (hostname) passed as the first argumnet

# A Simple classification based on node's hostname
if node.include?("dev")
  classification = { "classes" => ["dev_class"] }
elseif node.include?("prod")
  classification = { "classes" => ["prod_class"] }
else
  classification = { "classes" => ["common_class"] }
end

# Output the classification as JSON
puts classification.to_json

^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

This script dynamically assigns classes based on the node's hostname. For instance, nodes with `dev` in the hostname will be assigned the `dev_class` class, and those with `prod` will be assigned `prod_class`.

2. **Make the Script Executable**:

```
sudo chmod +x /usr/local/bin/puppet_enc.rb
```



```
root@puppetServer:~# sudo chmod +x /usr/local/bin/puppet_enc.rb
root@puppetServer:~#
```

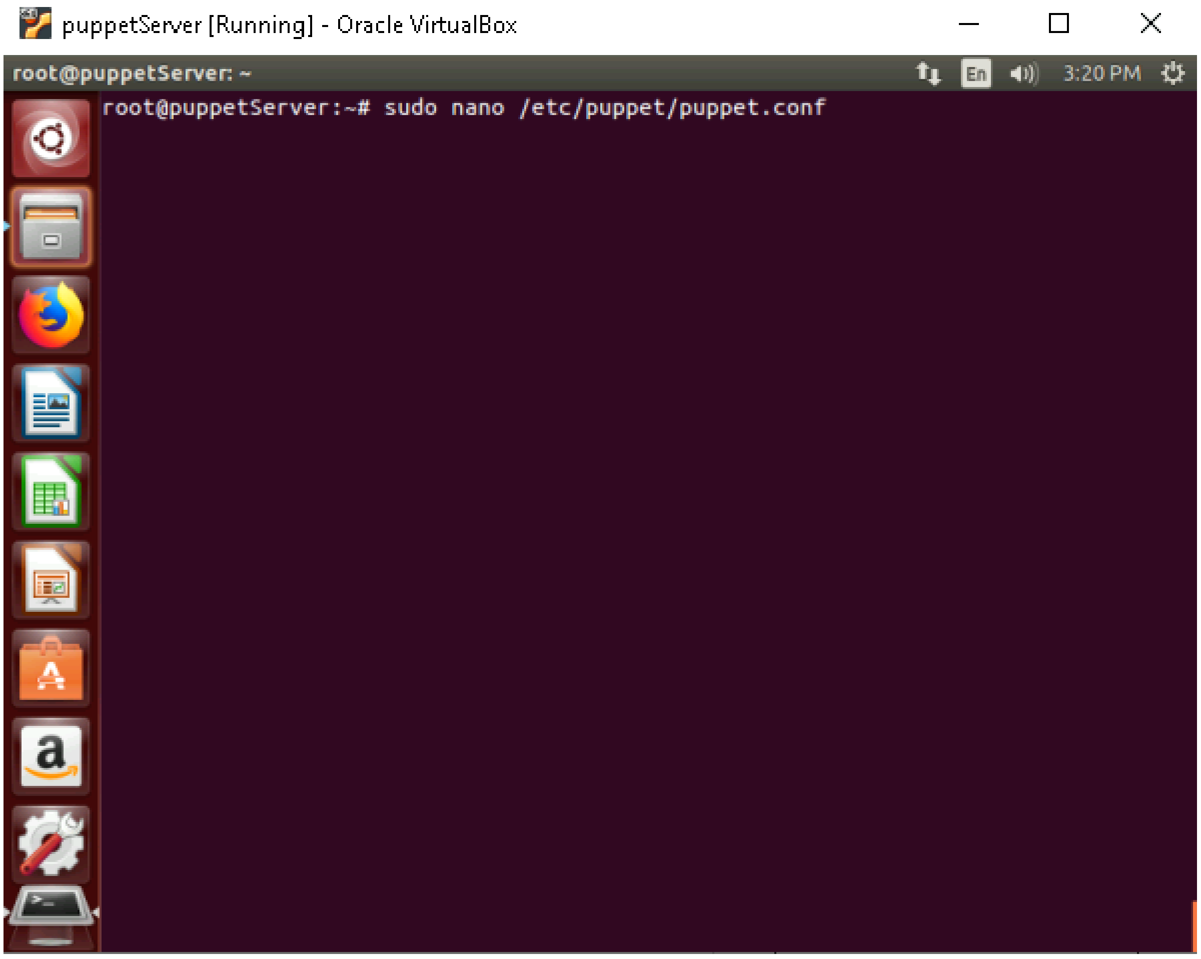
Step 2: Configure Puppet to Use the ENC

1. **Modify the Puppet Master Configuration**:

   Edit the `/etc/puppet/puppet.conf` file to tell Puppet to use your external node classifier script.

   Open `/etc/puppet/puppet.conf`:

```
sudo nano /etc/puppet/puppet.conf
```

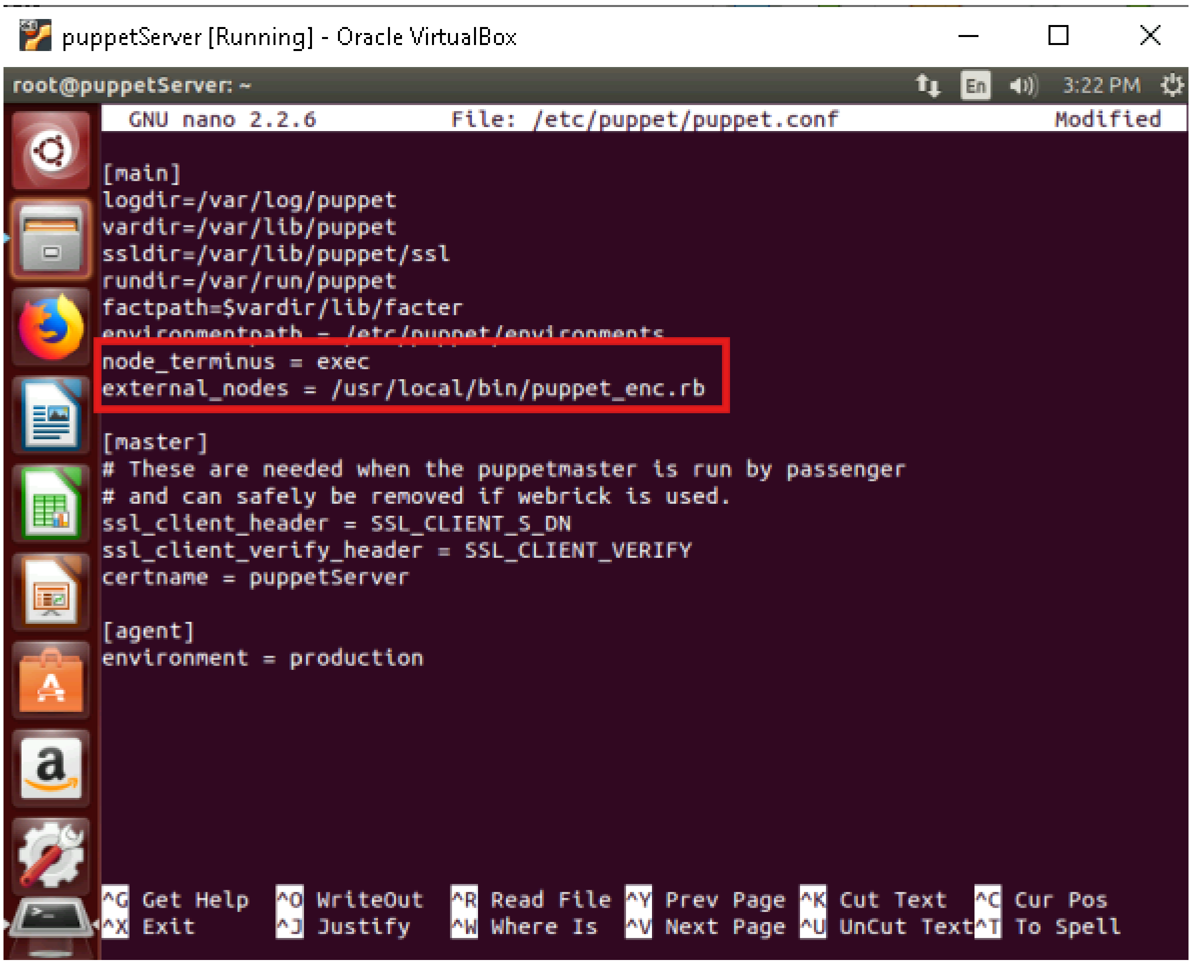


```
root@puppetServer:~# sudo nano /etc/puppet/puppet.conf
```

2. **Configure the External Node Classifier**:

   Add the following lines under the `[main]` section to point Puppet to the ENC script:

```
[main]
node_terminus = exec
external_nodes = /usr/local/bin/puppet_enc.rb
```



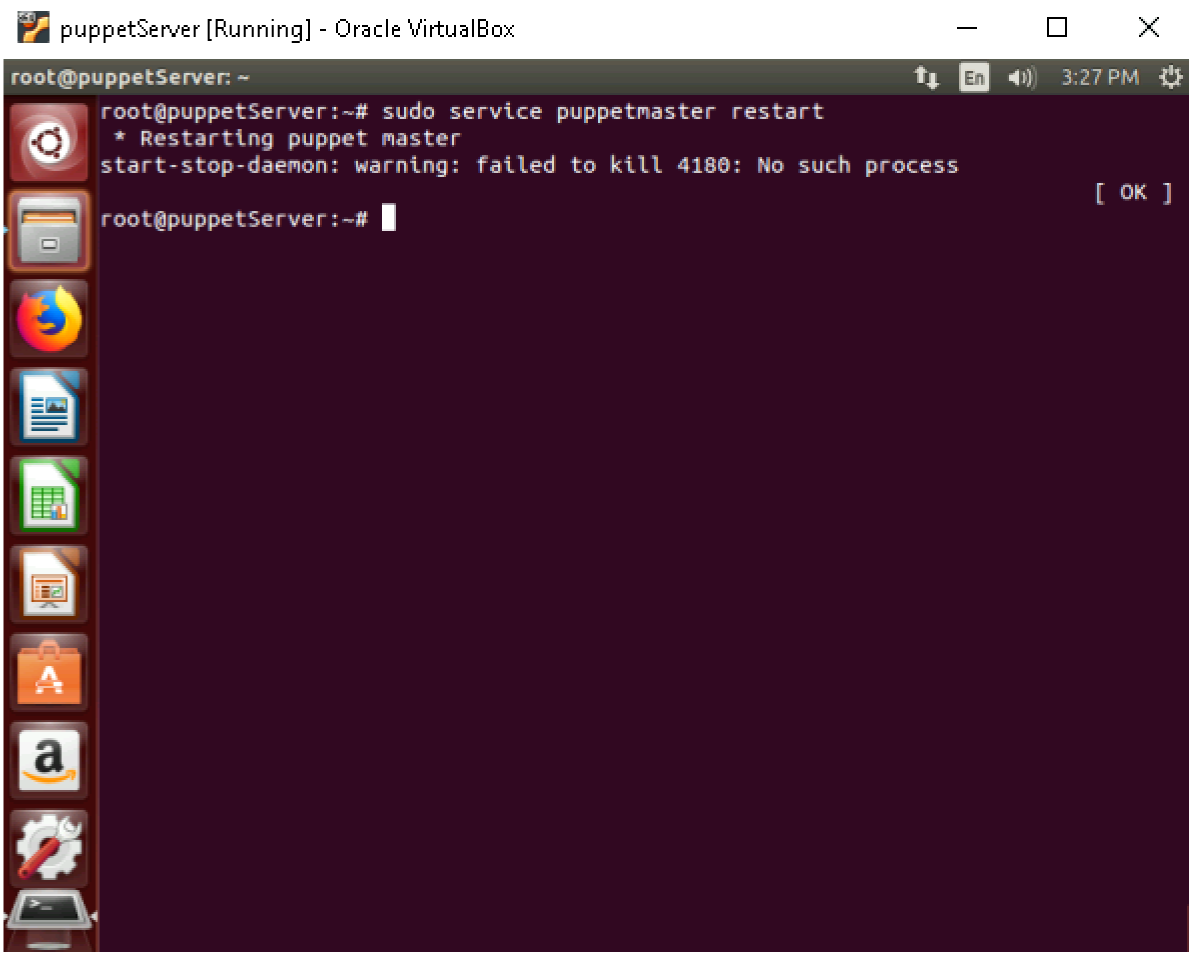
```
  GNU nano 2.2.6            File: /etc/puppet/puppet.conf                Modified

[main]
logdir=/var/log/puppet
vardir=/var/lib/puppet
ssldir=/var/lib/puppet/ssl
rundir=/var/run/puppet
factpath=$vardir/lib/facter
environmentpath = /etc/puppet/environments
node_terminus = exec
external_nodes = /usr/local/bin/puppet_enc.rb

[master]
# These are needed when the puppetmaster is run by passenger
# and can safely be removed if webrick is used.
ssl_client_header = SSL_CLIENT_S_DN
ssl_client_verify_header = SSL_CLIENT_VERIFY
certname = puppetServer

[agent]
environment = production

^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

- **Explanation**:
  - `node_terminus = exec`: This tells Puppet to use an external script for node classification.
  - `external_nodes = /usr/local/bin/puppet_enc.rb`: Specifies the path to the ENC script.

3. **Restart Puppet Server**:

   Restart the Puppet Server for the changes to take effect:

```
sudo service puppetserver restart
```



```
root@puppetServer:~# sudo service puppetmaster restart
 * Restarting puppet master
start-stop-daemon: warning: failed to kill 4180: No such process
                                                                    [ OK ]
root@puppetServer:~#
```

Step 3: Write the ENC Script

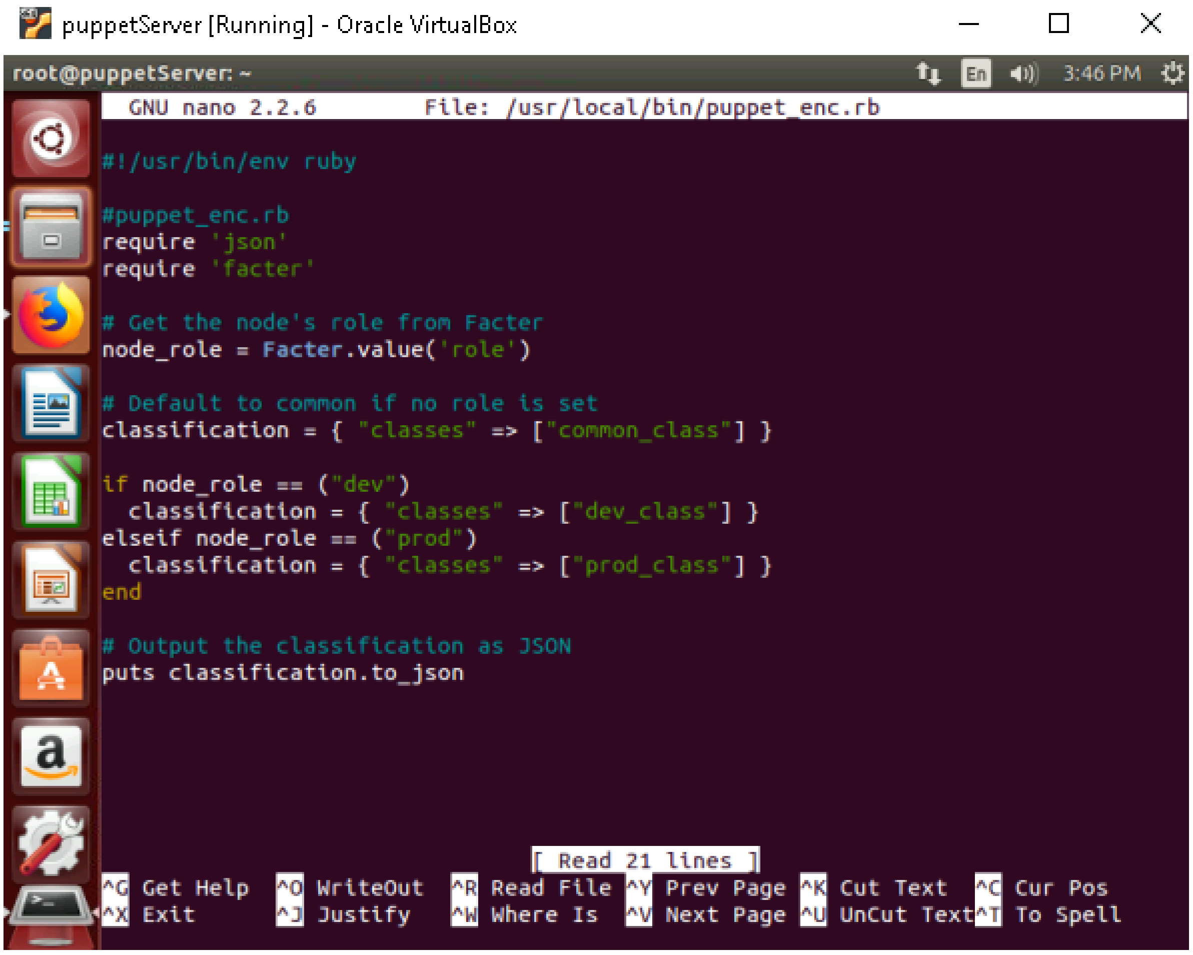
If you want a more complex ENC, you can modify the script to pull node data from a database, an API, or use custom facts.

For example, you could fetch node information based on custom facts from the Puppet database or other APIs to dynamically assign roles, environments, and classes.

Here's an enhanced version of the Ruby ENC script that checks custom facts (e.g., role or environment) for classification:

```
#!/usr/bin/env ruby

#puppet_enc.rb
require 'json'
require 'facter'

# Get the node's role from Facter
node_role = Facter.value('role')

# Default to common if no role is set
classification = { "classes" => ["common_class"] }

if node_role == "dev"
  classification["classes"] = ["dev_class"]
```

```
elsif node_role == "prod"
  classification["classes"] = ["prod_class"]
end

# Output the classification as JSON
puts classification.to_json
```



```
  GNU nano 2.2.6          File: /usr/local/bin/puppet_enc.rb

#!/usr/bin/env ruby

#puppet_enc.rb
require 'json'
require 'facter'

# Get the node's role from Facter
node_role = Facter.value('role')

# Default to common if no role is set
classification = { "classes" => ["common_class"] }

if node_role == ("dev")
  classification = { "classes" => ["dev_class"] }
elseif node_role == ("prod")
  classification = { "classes" => ["prod_class"] }
end

# Output the classification as JSON
puts classification.to_json

                                [ Read 21 lines ]
^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

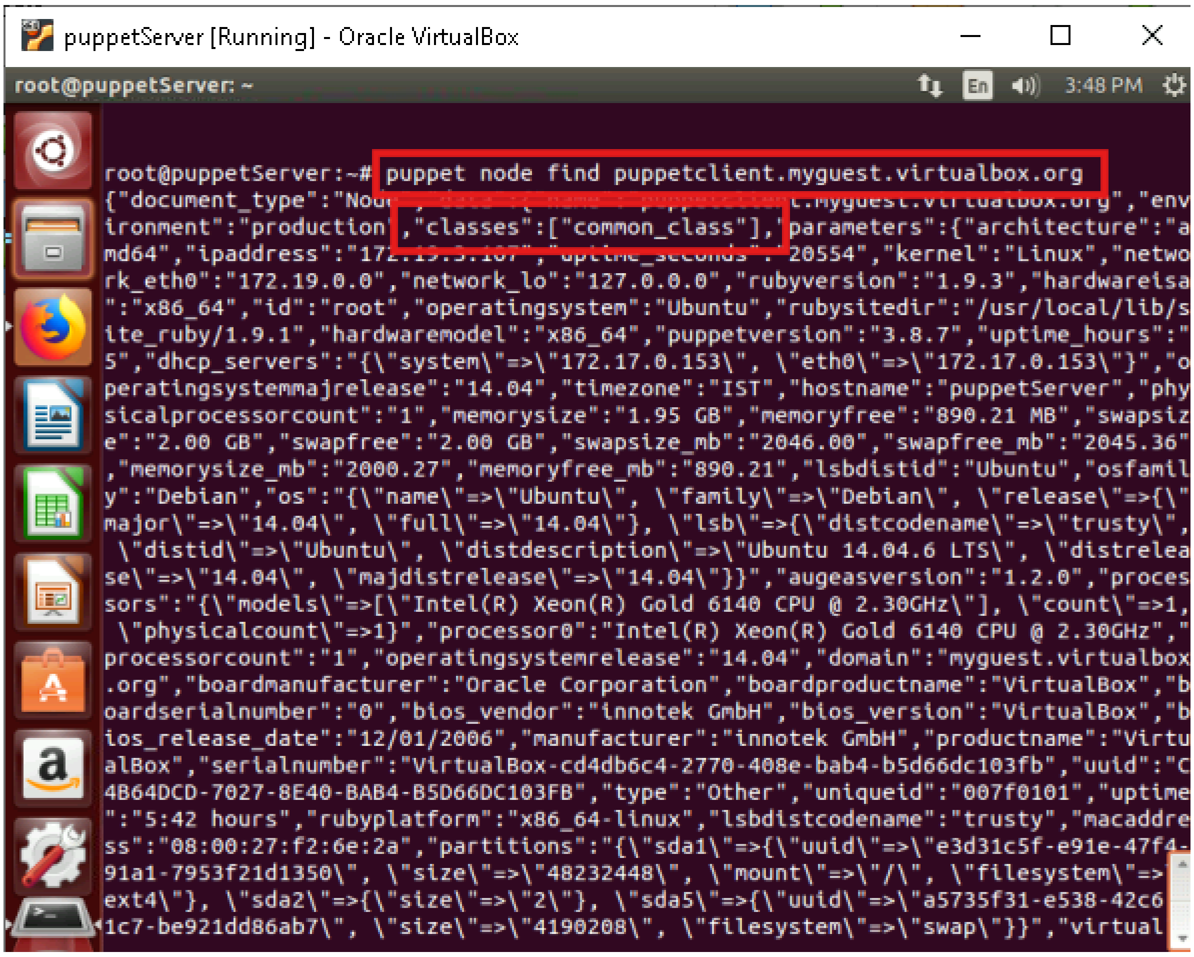
- **Explanation**:
  - The script uses the **Facter** gem to retrieve the `role` fact of the node and classifies the node accordingly.

Step 4: Test the Node Classification

1. **Verify Node Classification on the Puppet Master**:

   You can test the node classification by running the following command:

   ```
   puppet node find <node_fqdn>
   ```



```
root@puppetServer:~# puppet node find puppetclient.myguest.virtualbox.org
{"document_type":"Nod[illegible]t.myguest.virtualbox.org","env
ironment":"production","classes":["common_class"],"parameters":{"architecture":"a
md64","ipaddress":"172.[illegible]","uptime_seconds":"20554","kernel":"Linux","netwo
rk_eth0":"172.19.0.0","network_lo":"127.0.0.0","rubyversion":"1.9.3","hardwareisa
":"x86_64","id":"root","operatingsystem":"Ubuntu","rubysitedir":"/usr/local/lib/s
ite_ruby/1.9.1","hardwaremodel":"x86_64","puppetversion":"3.8.7","uptime_hours":"
5","dhcp_servers":"{\"system\"=>\"172.17.0.153\", \"eth0\"=>\"172.17.0.153\"}","o
peratingsystemmajrelease":"14.04","timezone":"IST","hostname":"puppetServer","phy
sicalprocessorcount":"1","memorysize":"1.95 GB","memoryfree":"890.21 MB","swapsiz
e":"2.00 GB","swapfree":"2.00 GB","swapsize_mb":"2046.00","swapfree_mb":"2045.36"
,"memorysize_mb":"2000.27","memoryfree_mb":"890.21","lsbdistid":"Ubuntu","osfamil
y":"Debian","os":"{\"name\"=>\"Ubuntu\", \"family\"=>\"Debian\", \"release\"=>{\"
major\"=>\"14.04\", \"full\"=>\"14.04\"}, \"lsb\"=>{\"distcodename\"=>\"trusty\",
 \"distid\"=>\"Ubuntu\", \"distdescription\"=>\"Ubuntu 14.04.6 LTS\", \"distrelea
se\"=>\"14.04\", \"majdistrelease\"=>\"14.04\"}}","augeasversion":"1.2.0","proces
sors":"{\"models\"=>[\"Intel(R) Xeon(R) Gold 6140 CPU @ 2.30GHz\"], \"count\"=>1,
 \"physicalcount\"=>1}","processor0":"Intel(R) Xeon(R) Gold 6140 CPU @ 2.30GHz","
processorcount":"1","operatingsystemrelease":"14.04","domain":"myguest.virtualbox
.org","boardmanufacturer":"Oracle Corporation","boardproductname":"VirtualBox","b
oardserialnumber":"0","bios_vendor":"innotek GmbH","bios_version":"VirtualBox","b
ios_release_date":"12/01/2006","manufacturer":"innotek GmbH","productname":"Virtu
alBox","serialnumber":"VirtualBox-cd4db6c4-2770-408e-bab4-b5d66dc103fb","uuid":"C
4B64DCD-7027-8E40-BAB4-B5D66DC103FB","type":"Other","uniqueid":"007f0101","uptime
":"5:42 hours","rubyplatform":"x86_64-linux","lsbdistcodename":"trusty","macaddre
ss":"08:00:27:f2:6e:2a","partitions":"{\"sda1\"=>{\"uuid\"=>\"e3d31c5f-e91e-47f4-
91a1-7953f21d1350\", \"size\"=>\"48232448\", \"mount\"=>\"/\", \"filesystem\"=>
ext4\"}, \"sda2\"=>{\"size\"=>\"2\"}, \"sda5\"=>{\"uuid\"=>\"a5735f31-e538-42c6
1c7-be921dd86ab7\", \"size\"=>\"4190208\", \"filesystem\"=>\"swap\"}}","virtual
```

- Replace `<node_fqdn>` with the fully qualified domain name of the node you want to classify.
- **Expected Output**: The `puppet node find` command should return information about the node, including the classes assigned to it (e.g., `dev_class`, `prod_class`).

References

- Puppet ENC Overview
- Facter Documentation
