

Write custom facts and use them in your Puppet manifests

Table of Contents

1. [Introduction](#)
 2. [Problem Statement](#)
 3. [Prerequisites](#)
 - [Software Requirements](#)
 - [Hardware Requirements](#)
 4. [Implementation Steps](#)
 - [Step 1: Write a Custom Fact](#)
 - [Step 2: Verify the Custom Fact](#)
 - [Step 3: Use the Custom Fact in a Puppet Manifest](#)
 - [Step 4: Test the Custom Fact](#)
 5. [References](#)
-

Introduction

Custom facts allow you to add node-specific data to Facter. For example, you can create a fact to indicate the role of a node (e.g., `webserver`, `database`) or custom attributes like installed software versions. This data can then be used in Puppet manifests to make configurations more dynamic and flexible.

Problem Statement

Default facts provided by Facter (like `os`, `hostname`, and `memory`) might not cover all the data needed for node configuration. Custom facts solve this by allowing administrators to define and retrieve node-specific data that aligns with their infrastructure requirements.

Prerequisites

Completion of all previous lab guides (up to Lab Guide-07) is required before proceeding with Lab Guide-08.

Software Requirements

- Puppet 3.8.7
 - Ruby (required for writing custom facts)
-

Hardware Requirements

- Puppet Master: Minimum 1GB RAM, 2 CPUs, 10GB Disk
 - Puppet Agent: Minimum 512MB RAM, 1 CPU, 5GB Disk
-

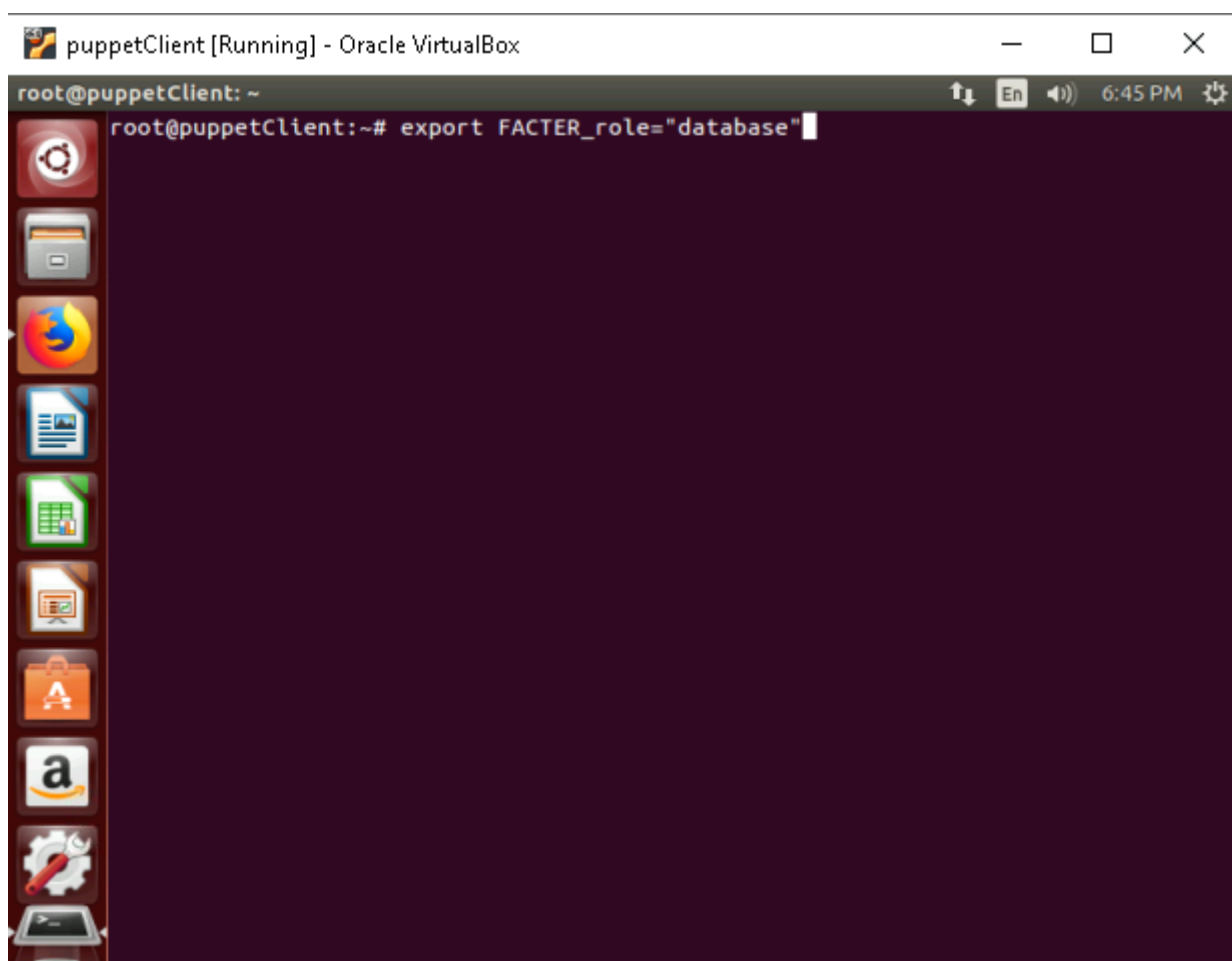
Implementation Steps

Step 1: Write a Custom Fact

1. Set the Role on a Node:

One can manually add the facts using the `export FACTER_{fact's name}` syntax to set the custom fact value for the current session on the Puppet Agent.

```
export FACTER_role="database"
```



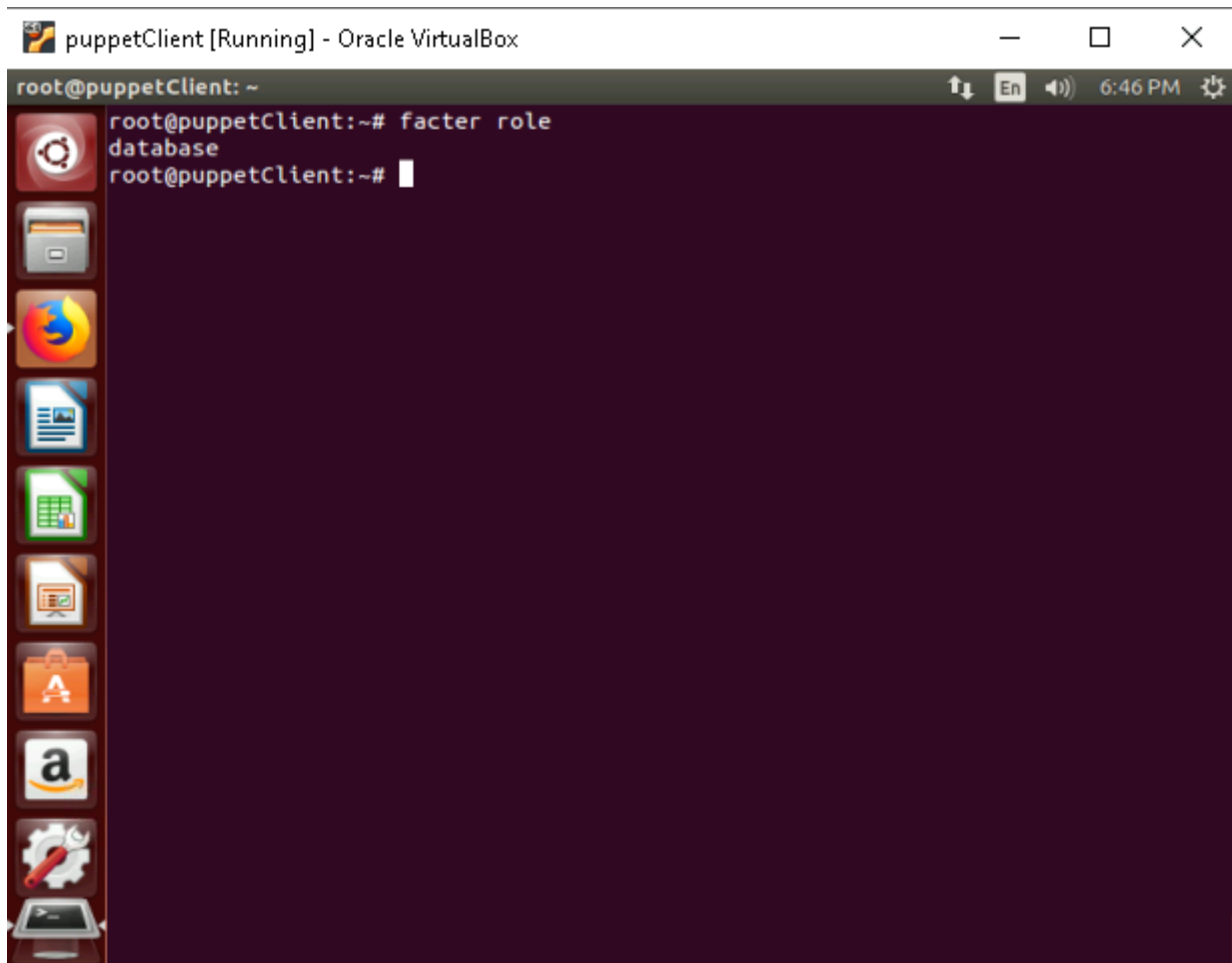
Note: After setting a custom fact, it is important to restart the Puppet Agent to ensure the new fact is recognized.

Step 2: Verify the Custom Fact

1. Verify the Custom Fact on the Puppet Agent is Set Correctly:

Run the following command on the Puppet Agent to verify the custom fact is set correctly on the agent to 'database'.

```
facter role
```



```
puppetClient [Running] - Oracle VirtualBox
root@puppetClient: ~
root@puppetClient:~# factor role
database
root@puppetClient:~#
```

- **Expected Output:**

```
^^^
database
^^^
```

2. Debugging Custom Facts:

If the fact does not appear, check the Factor logs for errors:

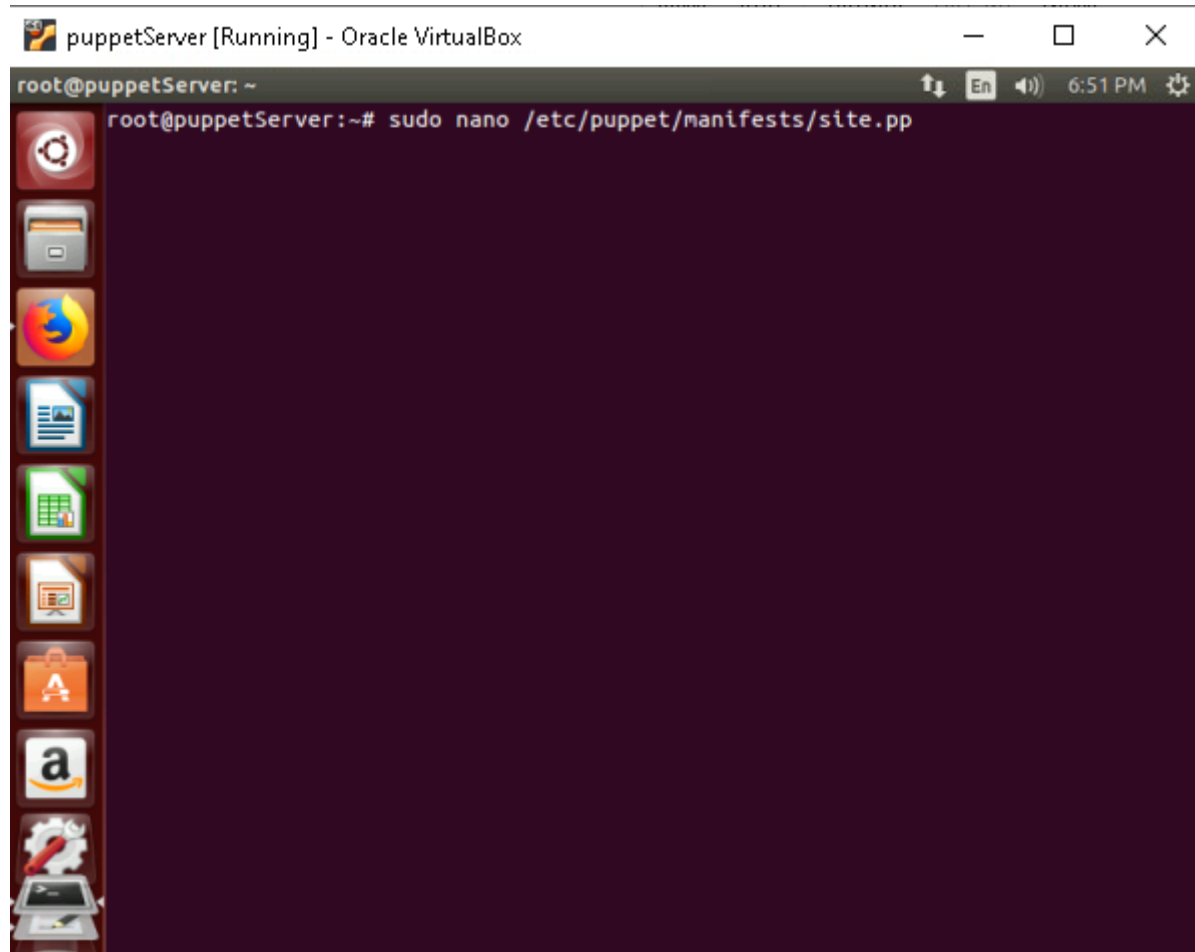
```
factor --debug
```

Step 3: Use the Custom Fact in a Puppet Manifest

1. Write a Puppet Manifest:

Modify or create a manifest file to use the custom fact. For example purposes, we will use the `site.pp` manifest file located at `/etc/puppet/manifests/` on the Puppet Master.

```
sudo nano /etc/puppet/manifests/site.pp
```



Add the following code:

```
node 'puppetclient.myguest.virtualbox.org' {  
  
    notify { "The role of this node is: ${role}": }  
  
    if $role == 'database' {  
        package { 'sqlite3': # SQLite package name  
            ensure => installed,  
        }  
  
        file { '/tmp/sqlite-demo.txt': # Example of using SQLite for debugging  
            ensure => file,  
            content => "SQLite has been installed successfully on this node.\n",  
            require => Package['sqlite3'],  
        }  
    }  
    elsif $role == 'webserver' {  
        package { 'nginx':  
            ensure => installed,  
        }  
        service { 'nginx':  
            ensure    => running,  
            enable    => true,  
            require   => Package['nginx'],  
        }  
    }  
}
```

```

    } else {
      notify { "Role not recognized: ${role}":
        message => "No valid role found for this node.",
      }
    }
  }
}

```

```

root@puppetServer: ~
GNU nano 2.2.6 File: /etc/puppet/manifests/site.pp

node 'puppetclient.myguest.virtualbox.org' {
  notify { "The role of this node is: ${role}": }

  if $role == 'database' {
    package { 'sqlite3': # SQLite package name
      ensure => installed,
    }

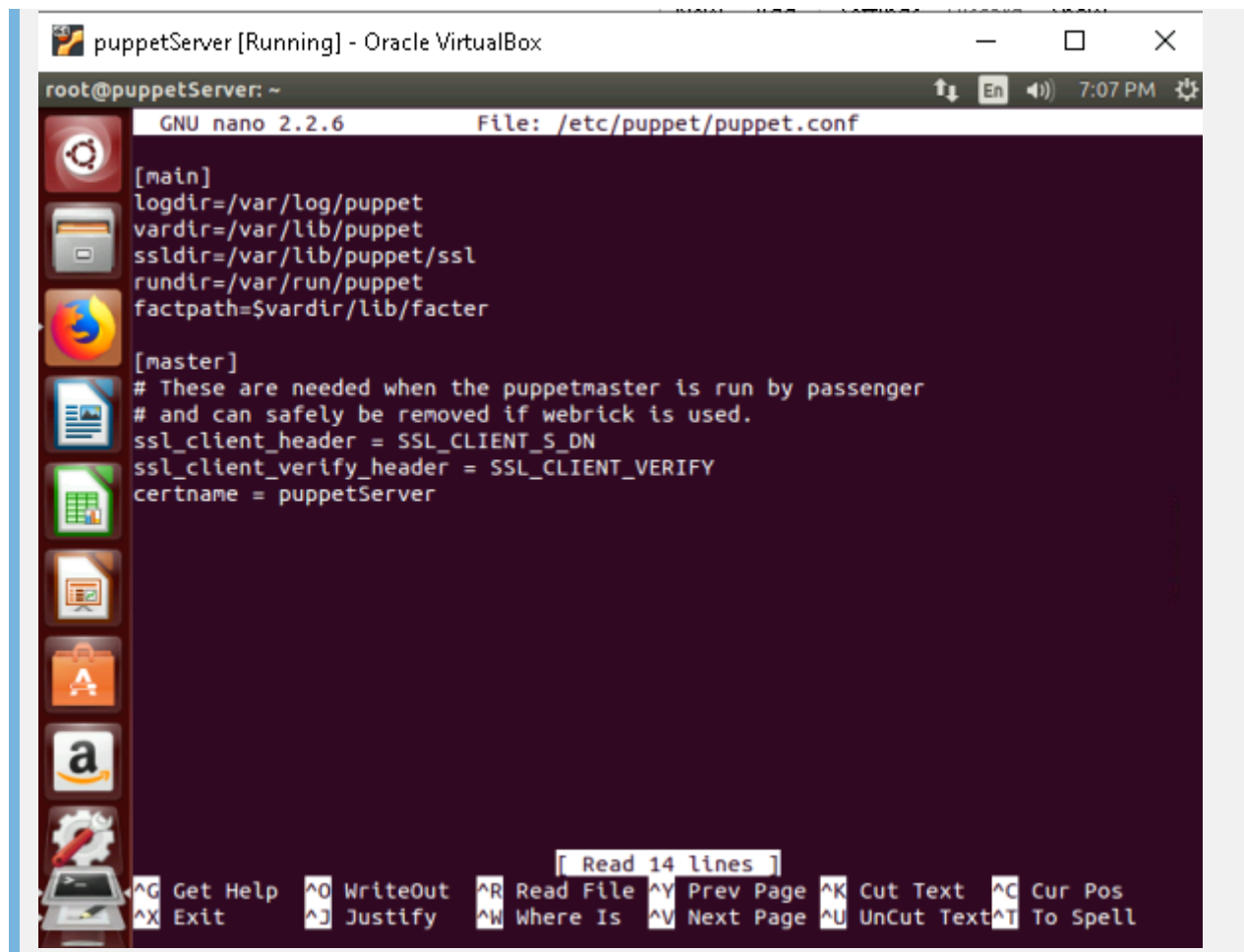
    file { '/tmp/sqlite-demo.txt': # Example of using SQLite for debugging
      ensure => file,
      content => "SQLite has been installed successfully on this node.\n",
      require => Package['sqlite3'],
    }
  } elsif $role == 'webserver' {
    package { 'nginx':
      ensure => installed,
    }
    service { 'nginx':
      ensure => running,
      enable => true,
      require => Package['nginx'],
    }
  } else {
    notify { "Role not recognized: ${role}":
      message => "No valid role found for this node.",
    }
  }
}

```

◦ **Explanation:**

- The **notify** resource displays the role of the node.
- The **if** statement checks the role of the node and installs the corresponding package and creates a file if the role is **database**.
- The **elsif** statement installs the **nginx** package and starts the service if the role is **webserver**.
- The **else** statement notifies the user if the role is not recognized.

Note: Make sure to remove ENC configurations from `/etc/puppet/puppet.conf` if you have configured them previously and even the environment configurations if you have configured them and then **restart the puppet server**.



```
puppetServer [Running] - Oracle VirtualBox
root@puppetServer: ~
GNU nano 2.2.6 File: /etc/puppet/puppet.conf

[main]
logdir=/var/log/puppet
vardir=/var/lib/puppet
ssl_dir=/var/lib/puppet/ssl
rundir=/var/run/puppet
factpath=$vardir/lib/facter

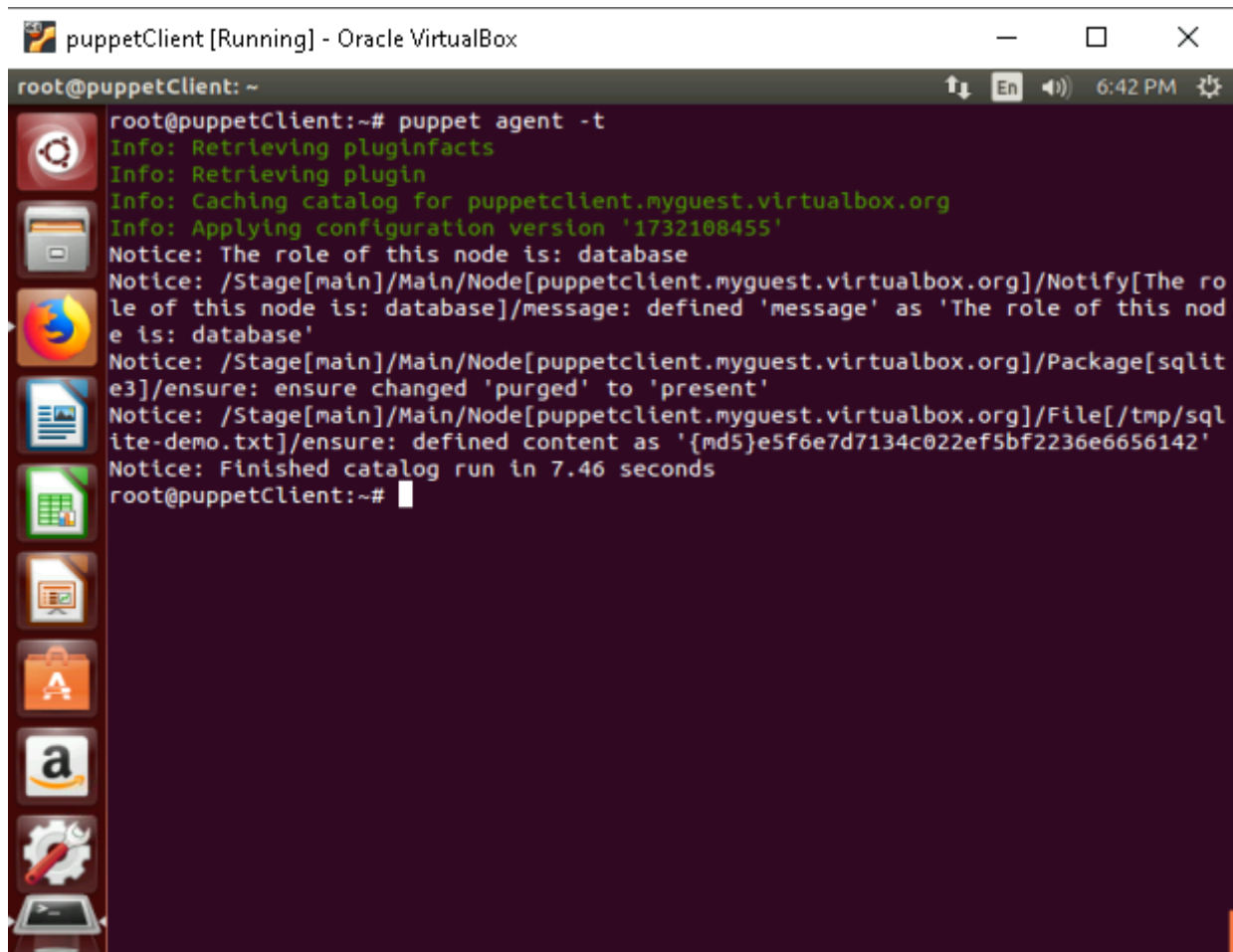
[master]
# These are needed when the puppetmaster is run by passenger
# and can safely be removed if webrick is used.
ssl_client_header = SSL_CLIENT_S_DN
ssl_client_verify_header = SSL_CLIENT_VERIFY
certname = puppetServer

[ Read 14 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
```

2. Apply the Manifest:

On the Puppet Agent, run the following command to apply the manifest:

```
puppet agent -t
```



```
puppetClient [Running] - Oracle VirtualBox
root@puppetClient: ~
root@puppetClient:~# puppet agent -t
Info: Retrieving pluginfacts
Info: Retrieving plugin
Info: Caching catalog for puppetclient.myguest.virtualbox.org
Info: Applying configuration version '1732108455'
Notice: The role of this node is: database
Notice: /Stage[main]/Main/Node[puppetclient.myguest.virtualbox.org]/Notify[The role of this node is: database]/message: defined 'message' as 'The role of this node is: database'
Notice: /Stage[main]/Main/Node[puppetclient.myguest.virtualbox.org]/Package[sqlite3]/ensure: ensure changed 'purged' to 'present'
Notice: /Stage[main]/Main/Node[puppetclient.myguest.virtualbox.org]/File[/tmp/sqlite-demo.txt]/ensure: defined content as '{md5}e5f6e7d7134c022ef5bf2236e6656142'
Notice: Finished catalog run in 7.46 seconds
root@puppetClient:~#
```

◦ **Expected Output:**

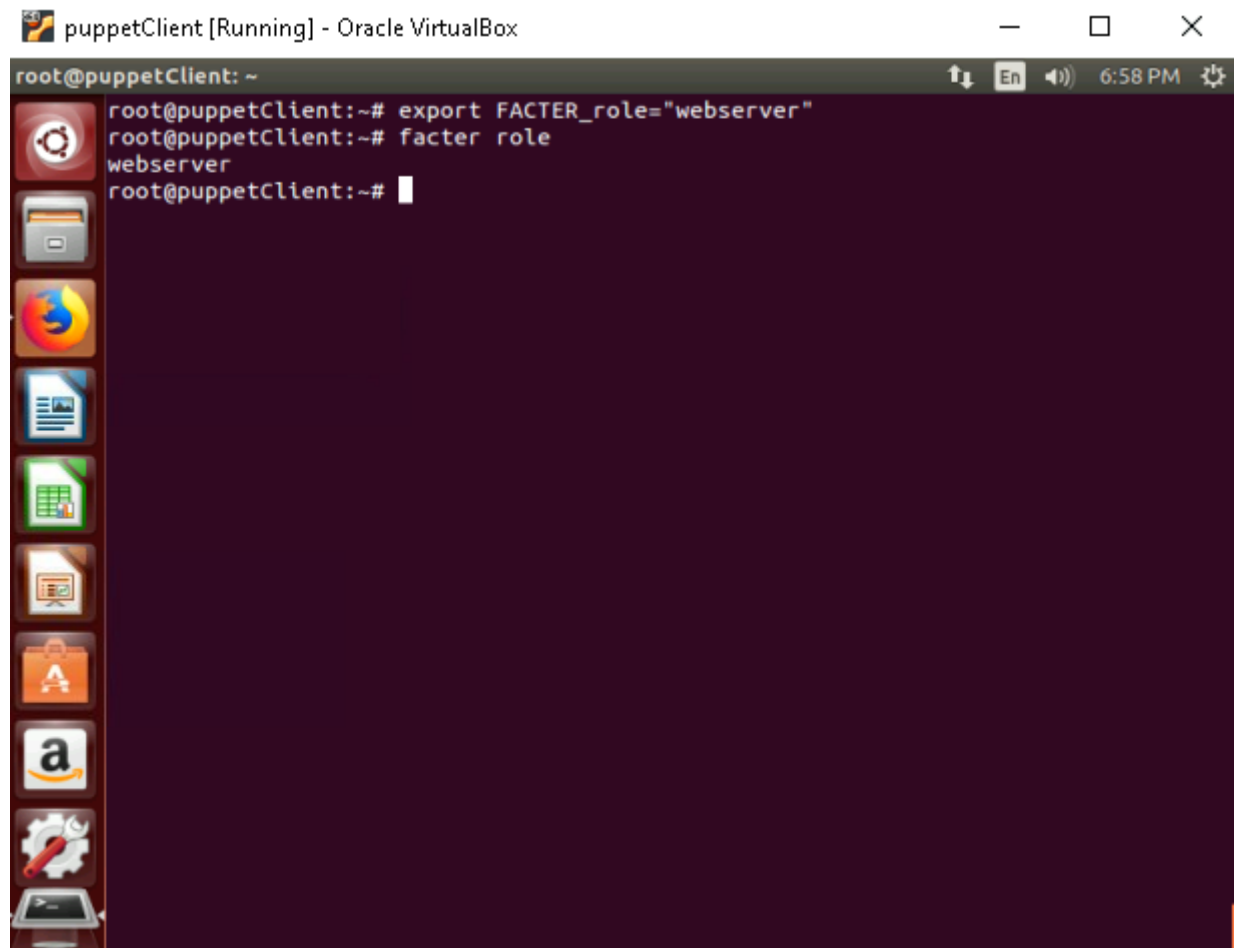
- The `sqlite3` package should be installed and the `/tmp/sqlite-demo.txt` file should be created if the role is `database`.
- The `nginx` package should be installed and the `nginx` service should be running if the role is `webserver`.
- If the role is not recognized, a notification should be displayed.

Step 4: Test the Custom Fact

1. Change the Node Role:

"export FACTER" Syntax is used to set the custom fact value for the current session.

```
export FACTER_role="webserver"
```

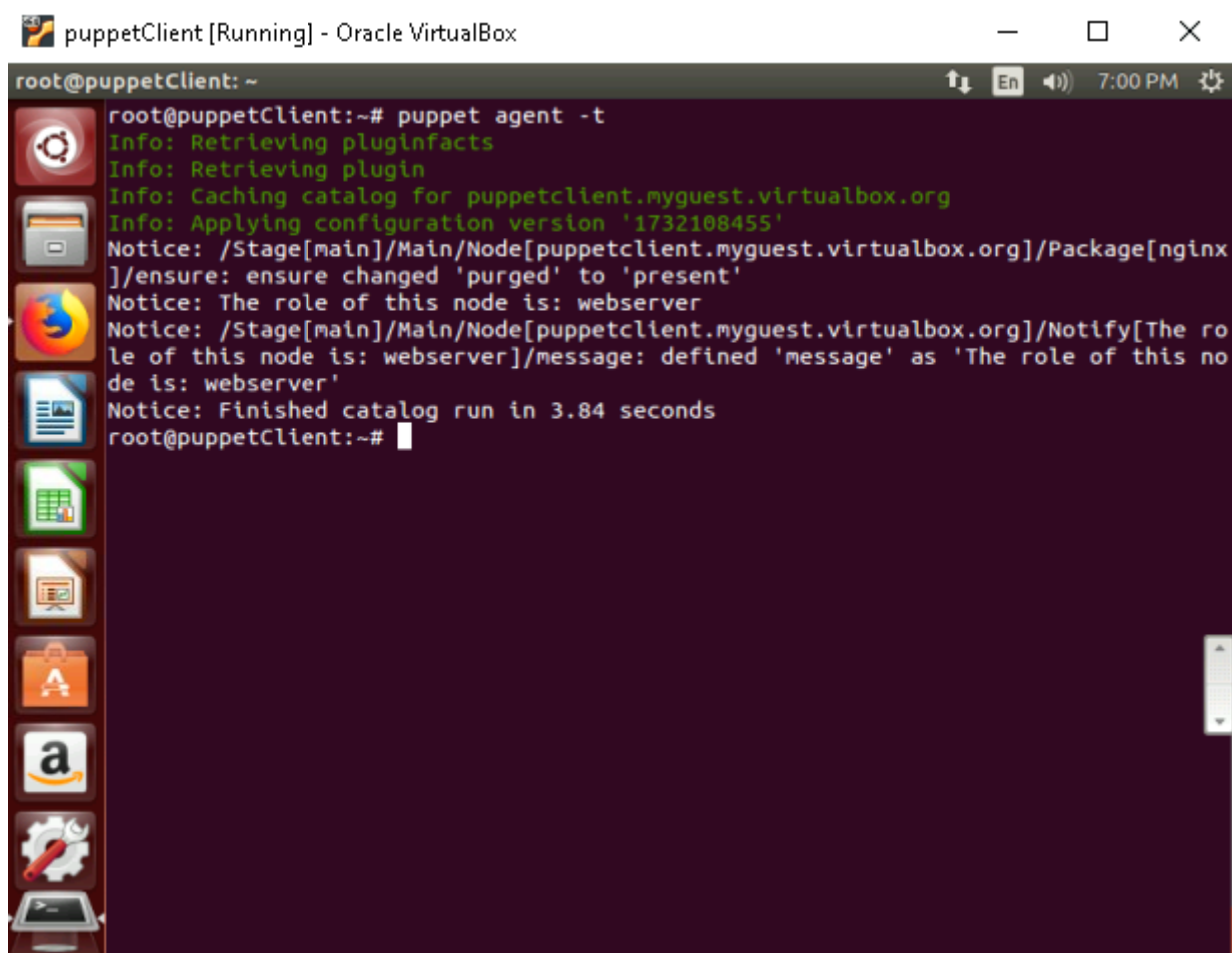


```
puppetClient [Running] - Oracle VirtualBox
root@puppetClient: ~
root@puppetClient:~# export FACTER_role="webserver"
root@puppetClient:~# factor role
webserver
root@puppetClient:~#
```

2. Run Puppet Agent:

Trigger the Puppet Agent to apply the updated role:

```
puppet agent -t
```


A screenshot of a terminal window titled 'puppetClient [Running] - Oracle VirtualBox'. The terminal shows the output of the command 'puppet agent -t'. The output includes several informational and notice messages: 'Info: Retrieving pluginfacts', 'Info: Retrieving plugin', 'Info: Caching catalog for puppetclient.myguest.virtualbox.org', 'Info: Applying configuration version '1732108455'', 'Notice: /Stage[main]/Main/Node[puppetclient.myguest.virtualbox.org]/Package[nginx]/ensure: ensure changed 'purged' to 'present'', 'Notice: The role of this node is: webserver', 'Notice: /Stage[main]/Main/Node[puppetclient.myguest.virtualbox.org]/Notify[The role of this node is: webserver]/message: defined 'message' as 'The role of this node is: webserver'', and 'Notice: Finished catalog run in 3.84 seconds'. The prompt 'root@puppetClient:~#' is visible at the bottom of the terminal output. The terminal window has a dark background and a sidebar on the left with various application icons.

```
root@puppetClient: ~
root@puppetClient:~# puppet agent -t
Info: Retrieving pluginfacts
Info: Retrieving plugin
Info: Caching catalog for puppetclient.myguest.virtualbox.org
Info: Applying configuration version '1732108455'
Notice: /Stage[main]/Main/Node[puppetclient.myguest.virtualbox.org]/Package[nginx]/ensure: ensure changed 'purged' to 'present'
Notice: The role of this node is: webserver
Notice: /Stage[main]/Main/Node[puppetclient.myguest.virtualbox.org]/Notify[The role of this node is: webserver]/message: defined 'message' as 'The role of this node is: webserver'
Notice: Finished catalog run in 3.84 seconds
root@puppetClient:~#
```

3. Verify Changes:

- Check if the package corresponding to the new role (`nginx` for `webserver`) is installed.

References

- [Puppet Facter Documentation](#)
 - [Writing Custom Facts](#)
 - [Puppet Facter Facts - Tutorialspoint](#)
-