

Implement Hiera to manage configuration data hierarchically.

Table of Contents

1. **Introduction**
 2. **Problem Statement**
 3. **Pre-requisites**
 4. **Steps to Implement Hiera**
 - **1. Install Hiera (if not already installed)**
 - **2. Configure Hiera**
 - **3. Create Hierarchical Data Files**
 - **4. Create a Puppet Module**
 - **5. Apply the Module**
 5. **References**
-

Introduction

This guide provides a step-by-step approach to implementing Hiera for managing configuration data hierarchically in Puppet.

Problem Statement

Managing configuration data directly within Puppet manifests can lead to duplication and reduced maintainability. Hiera helps to separate configuration data from code, making it easier to manage and reuse.

Pre-requisites

Completion of all previous lab guides (up to Lab Guide-03) is required before proceeding with Lab Guide-04.

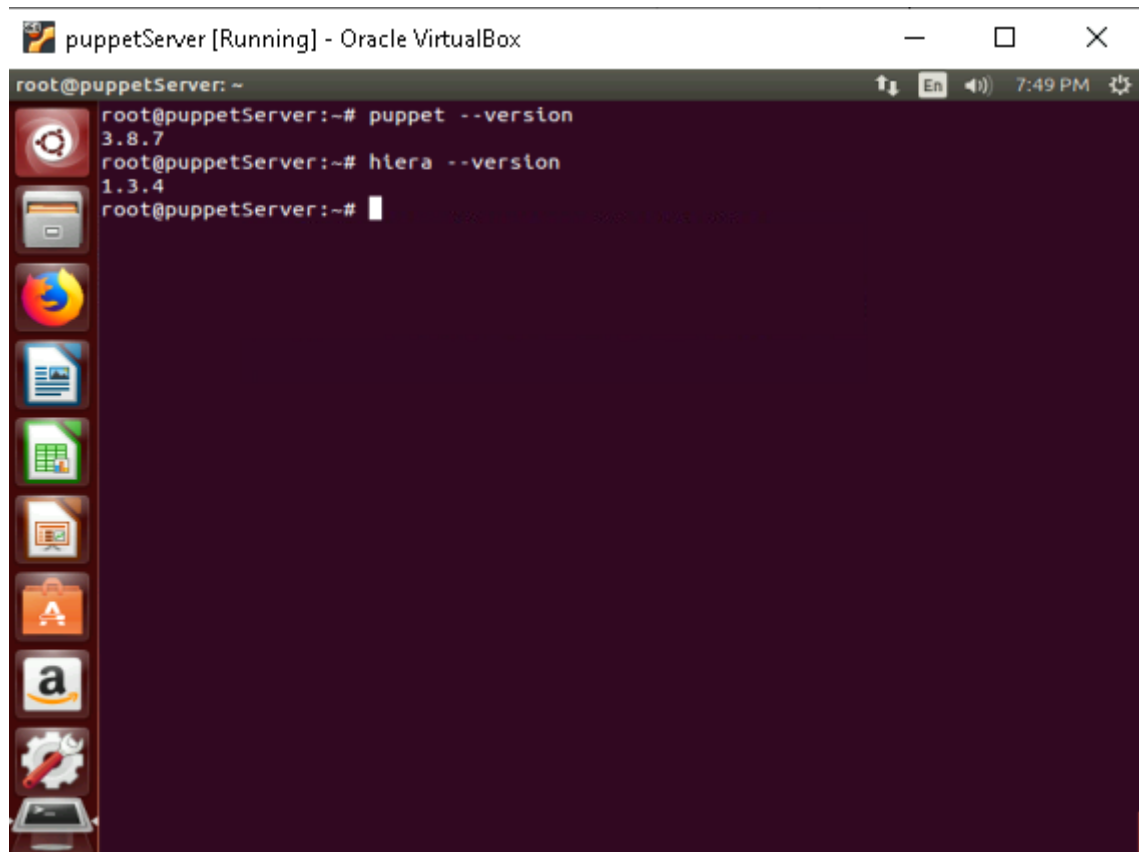
1. Puppet 3.8.7 is installed on your Puppet Master and Agent nodes.
 2. Ensure YAML support is available, as it's the default backend for Hiera.
 3. Administrative access to configure files on the Puppet server.
-

Steps to Implement Hiera

1. Install Hiera (if not already installed)

Hiera is bundled with Puppet 3.8.7. Verify installation using:

```
puppet --version
hiera --version
```



```
puppetServer [Running] - Oracle VirtualBox
root@puppetServer: ~
root@puppetServer:~# puppet --version
3.8.7
root@puppetServer:~# hiera --version
1.3.4
root@puppetServer:~#
```

If not installed, run:

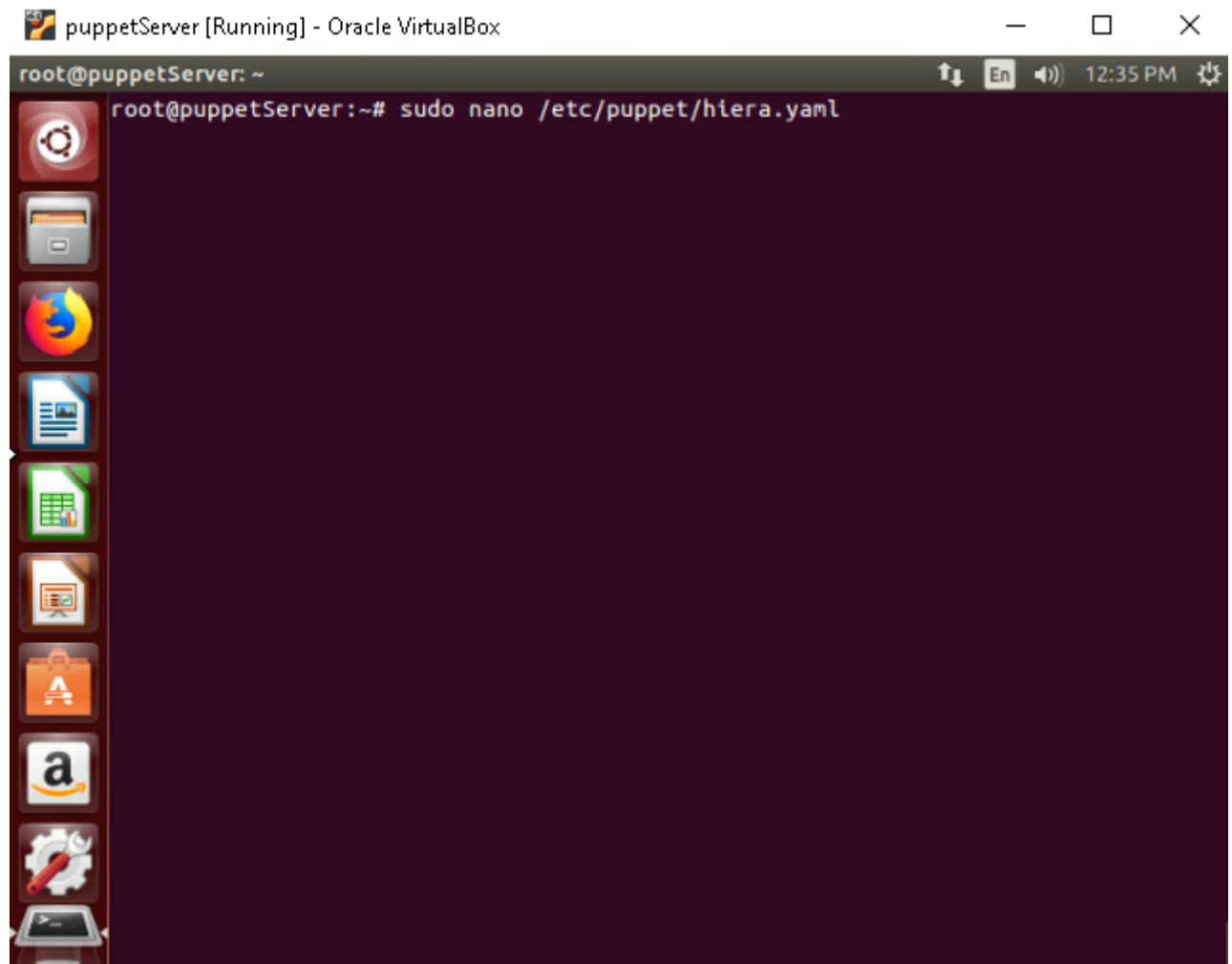
```
sudo apt-get install hiera
```

2. Configure Hiera

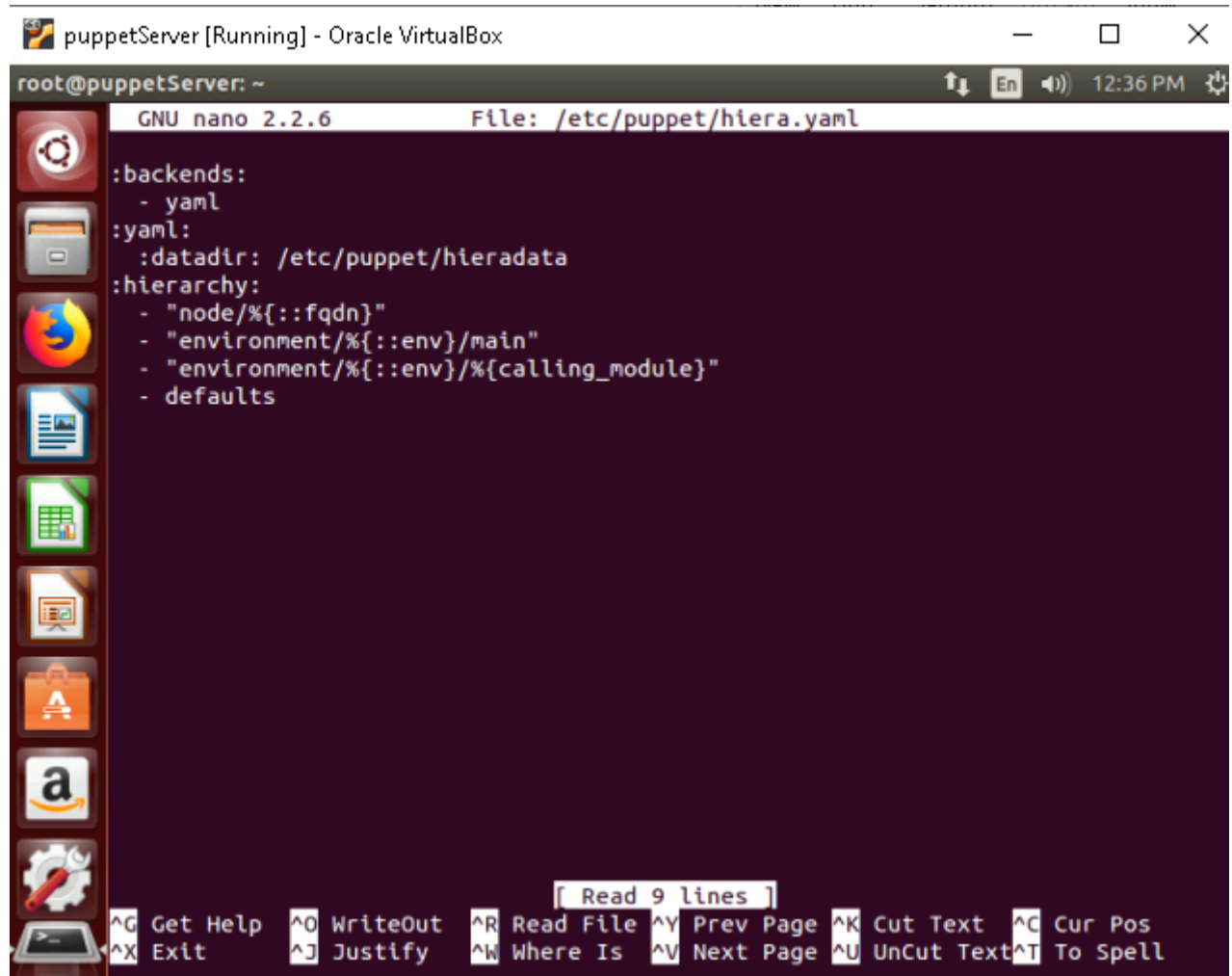
1. Create the Hiera Configuration File

The `hiera.yaml` file should be located at `/etc/puppet/hiera.yaml`. Example configuration:

```
sudo nano /etc/puppet/hiera.yaml
```



```
:backends:
  - yaml
:yaml:
  :datadir: /etc/puppet/hieradata
:hierarchy:
  - "node/%{::fqdn}"
  - "environment/%{::env}/main"
  - "environment/%{::env}/%{calling_module}"
  - defaults
```



```
root@puppetServer: ~
GNU nano 2.2.6 File: /etc/puppet/hiera.yaml

:backends:
- yaml
:yaml:
:datadir: /etc/puppet/hieradata
:hierarchy:
- "node/%{::fqdn}"
- "environment/%{::env}/main"
- "environment/%{::env}/%{calling_module}"
- defaults

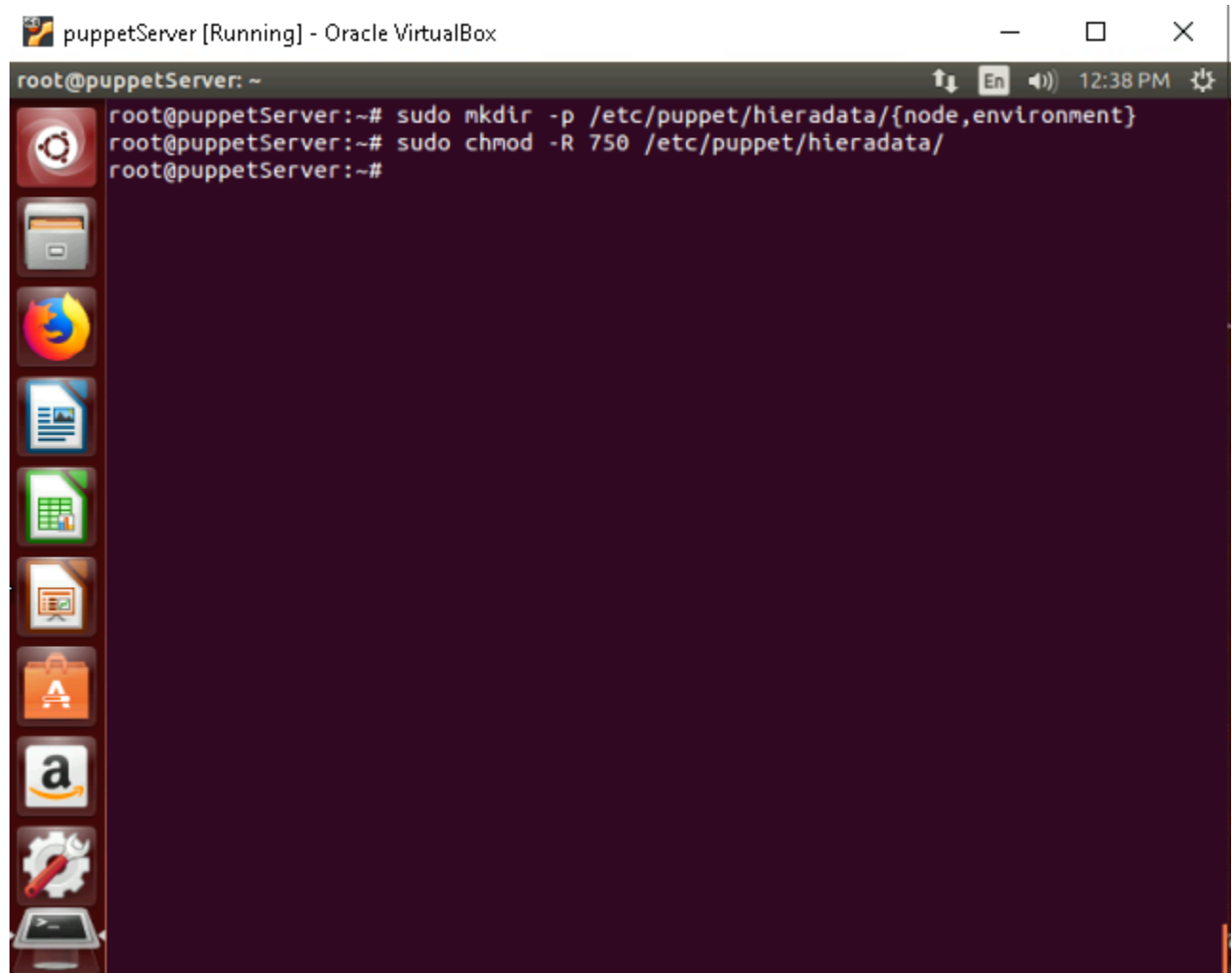
[ Read 9 lines ]
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

- **Backends:** Defines data storage formats (YAML is default).
- **Data Directory:** `/etc/puppet/hieradata` is where your YAML files will reside.
- **Hierarchy:** Defines the order of data lookups. It uses variables like `::fqdn` (node hostname) or `::env` (environment).

2. Create Data Directory

Ensure the directory `/etc/puppet/hieradata` exists:

```
sudo mkdir -p /etc/puppet/hieradata/{node,environment}
sudo chmod -R 750 /etc/puppet/hieradata
```



```
puppetServer [Running] - Oracle VirtualBox
root@puppetServer: ~
root@puppetServer:~# sudo mkdir -p /etc/puppet/hieradata/{node,environment}
root@puppetServer:~# sudo chmod -R 750 /etc/puppet/hieradata/
root@puppetServer:~#
```

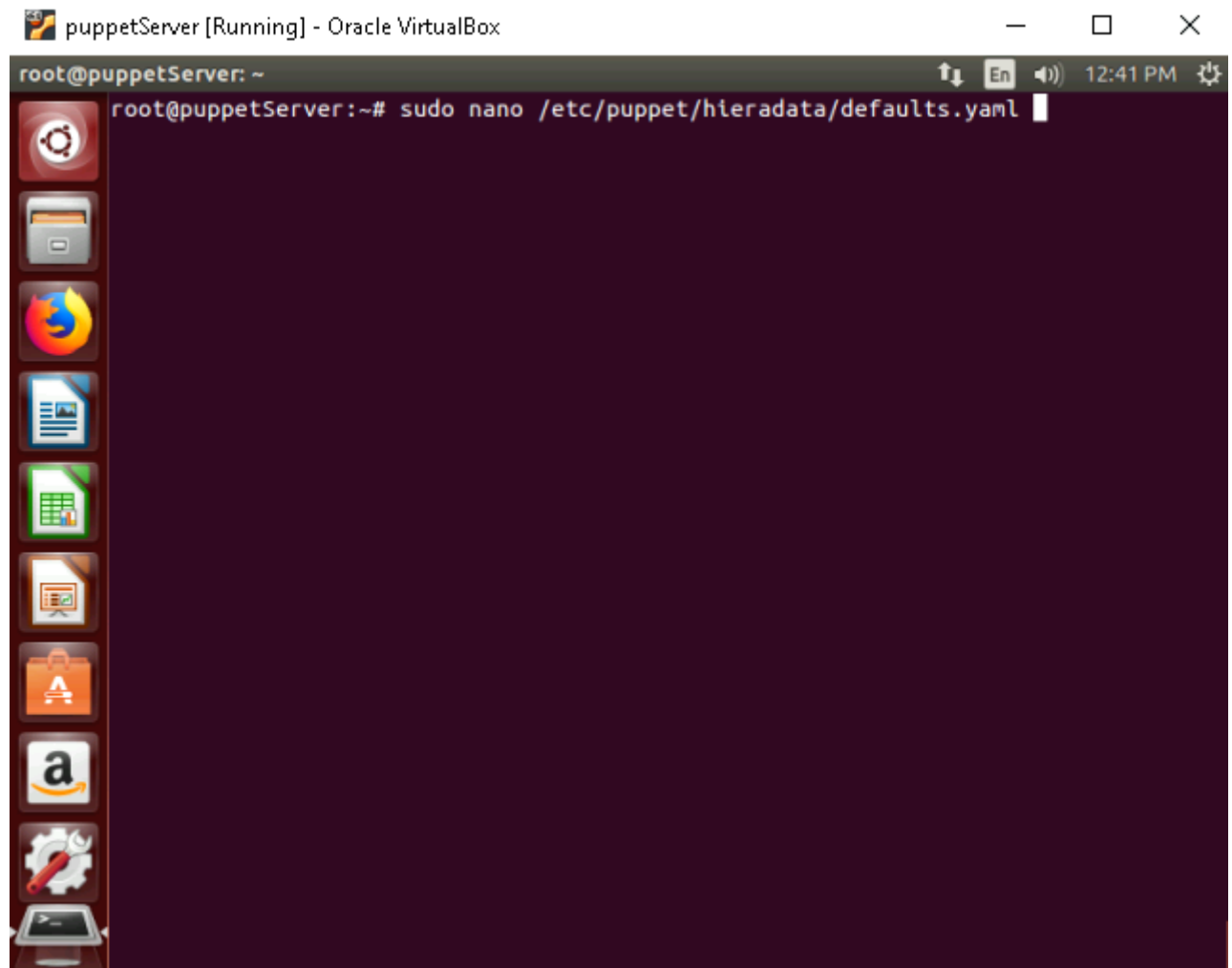
- **Node Data:** Contains node-specific data files.
- **Environment Data:** Contains environment-specific data files.

3. Create Hierarchical Data Files

1. **Common Data:** Default values shared across nodes.

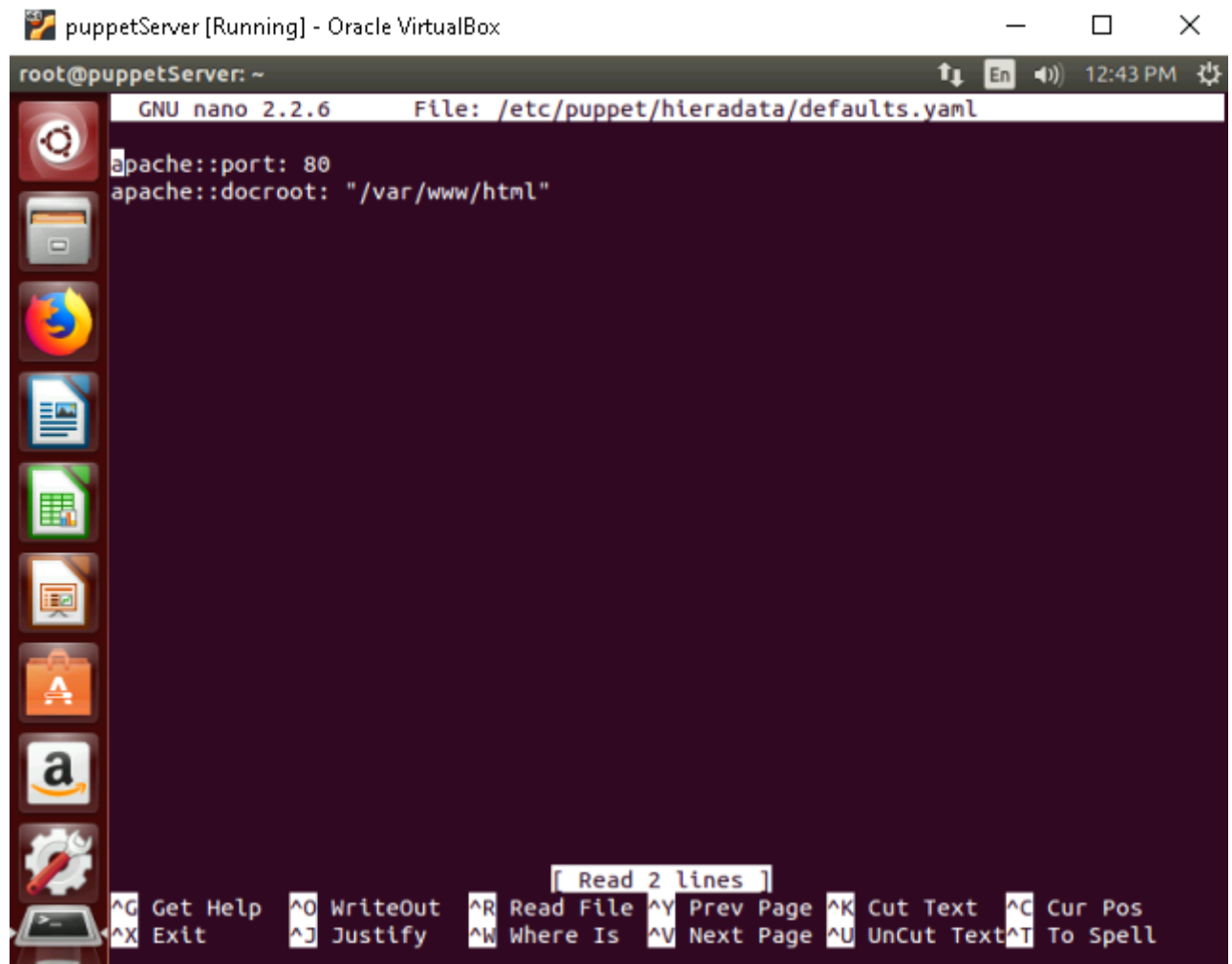
File: `/etc/puppet/hieradata/defaults.yaml`

```
sudo nano /etc/puppet/hieradata/defaults.yaml
```



Example:

```
apache::port: 80
apache::docroot: "/var/www/html"
```

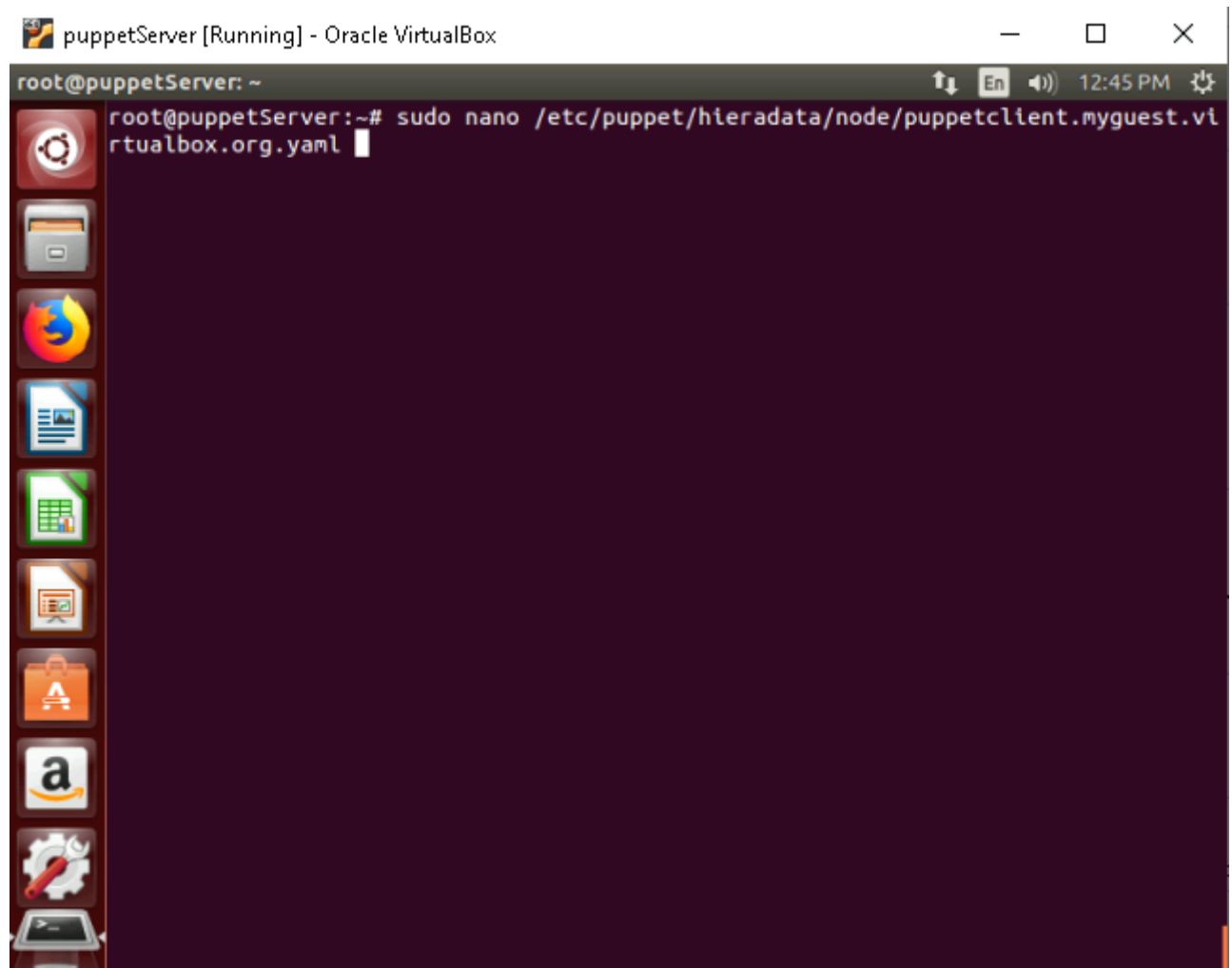


```
root@puppetServer: ~  
GNU nano 2.2.6 File: /etc/puppet/hieradata/defaults.yaml  
apache::port: 80  
apache::docroot: "/var/www/html"  
  
[ Read 2 lines ]  
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos  
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

2. **Node-Specific Data:** Values for individual nodes.

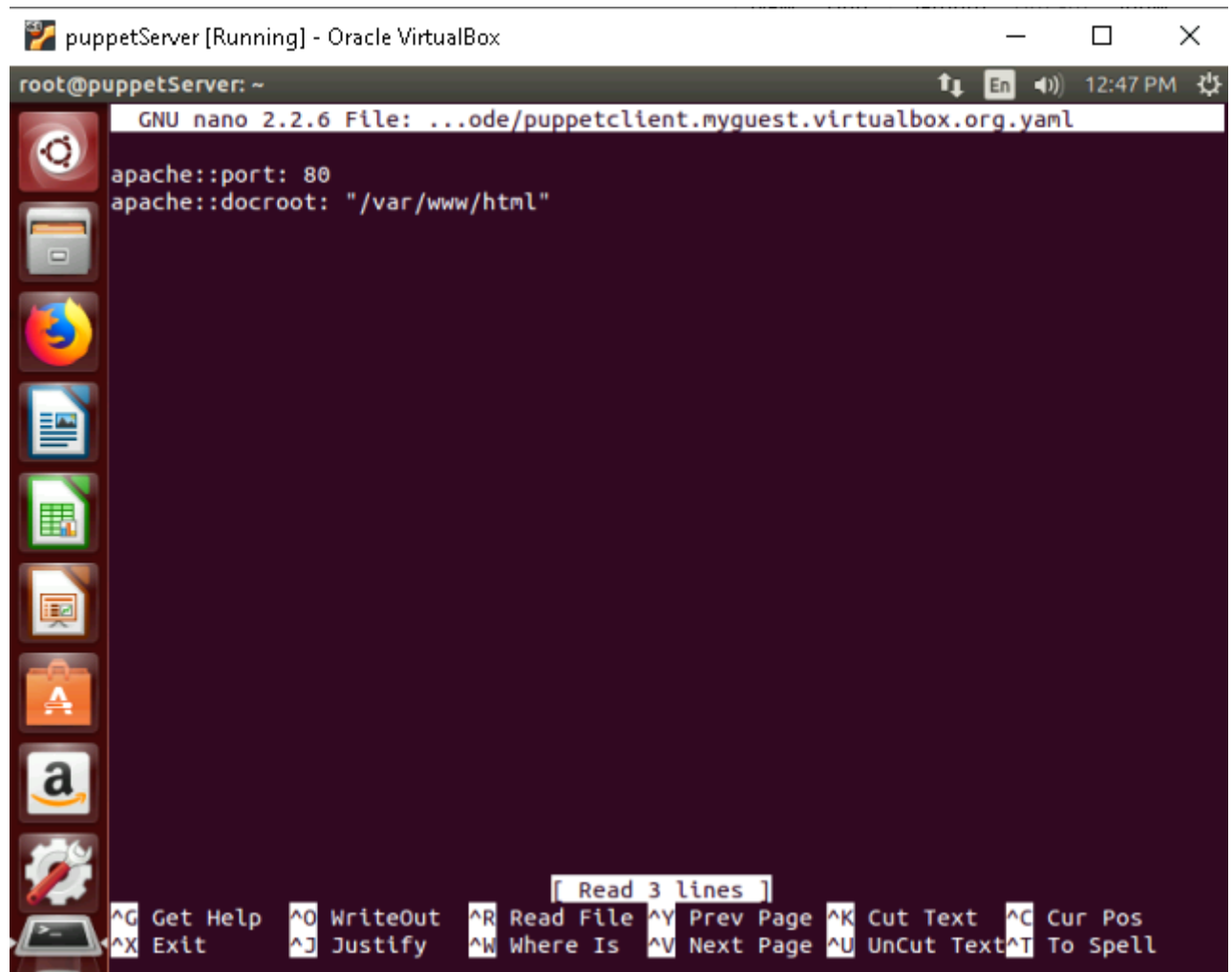
File: `/etc/puppet/hieradata/node/puppetclient.myguest.virtualbox.org.yaml`

```
sudo nano  
/etc/puppet/hieradata/node/puppetclient.myguest.virtualbox.org.yaml
```



Example:

```
apache::port: 80
apache::docroot: "/home/www/html"
```

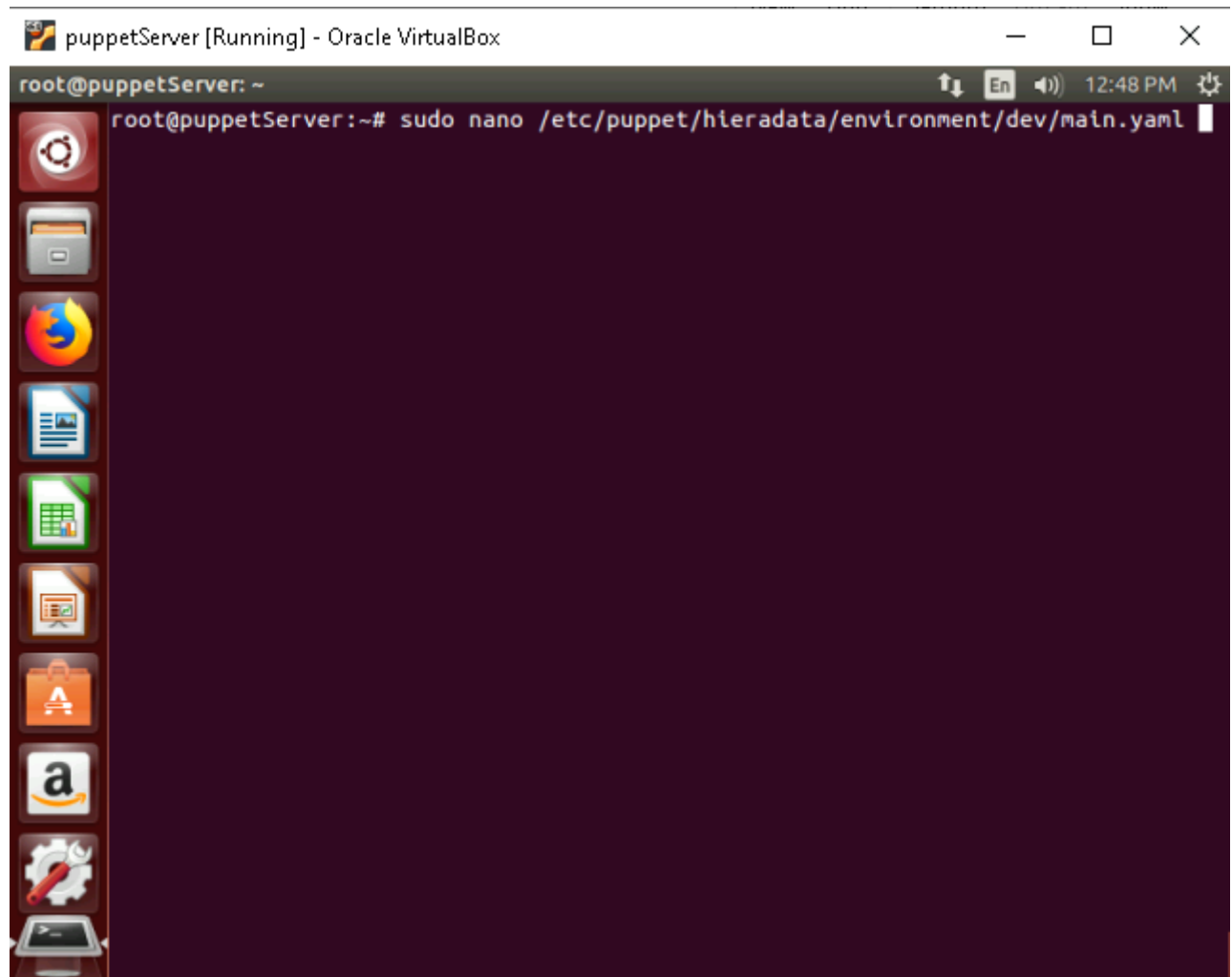



```
puppetServer [Running] - Oracle VirtualBox
root@puppetServer: ~
GNU nano 2.2.6 File: ...ode/puppetclient.myquest.virtualbox.org.yaml
apache::port: 80
apache::docroot: "/var/www/html"
[ Read 3 lines ]
^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text   ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is   ^V Next Page  ^U UnCut Text ^T To Spell
```

3. **Environment-Specific Data:** Values for environments like `dev` or `prod`, environments has to be created in `/etc/puppet/hieradata/environment/` directory manually.

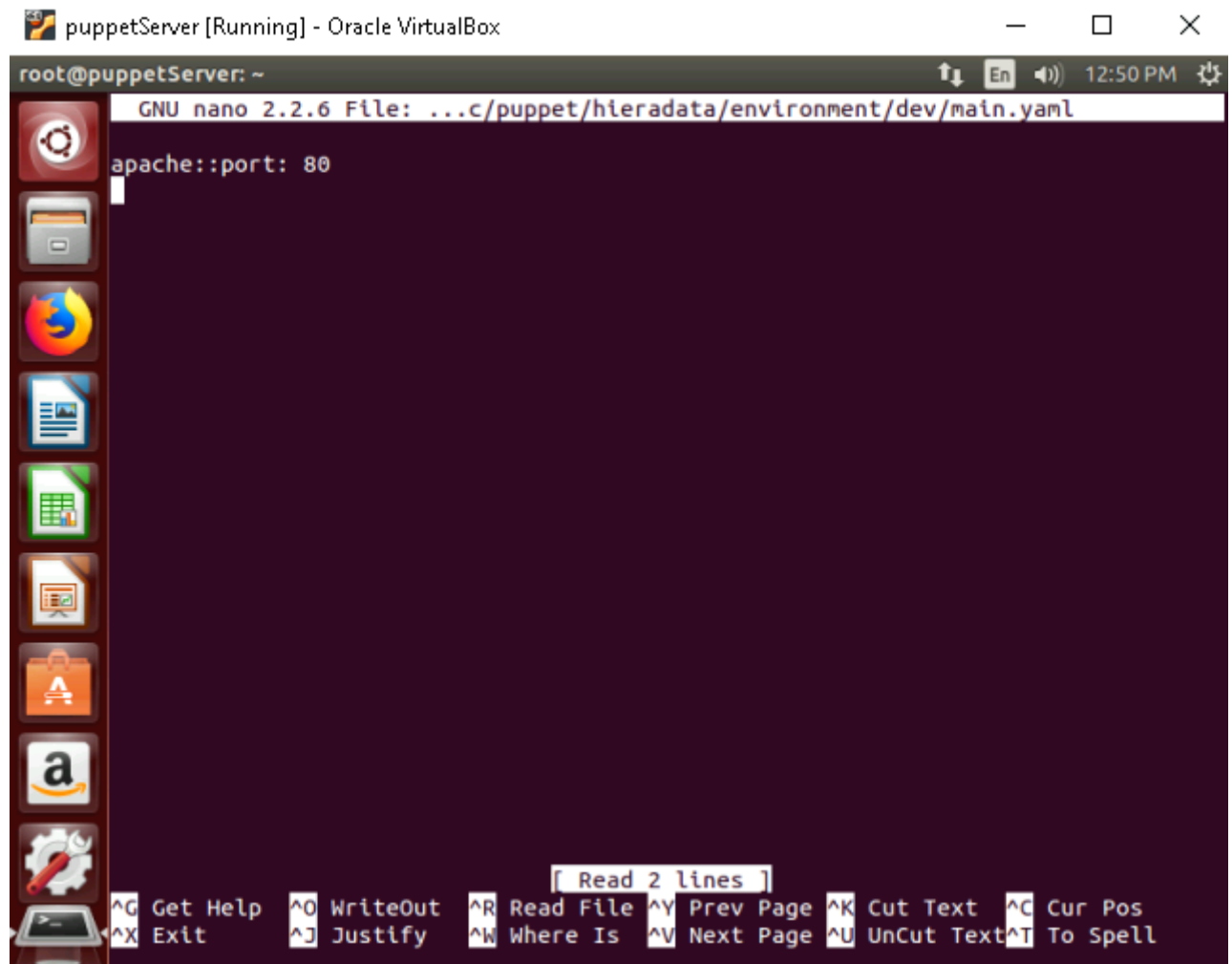
File: `/etc/puppet/hieradata/environment/prod/main.yaml`

```
sudo nano /etc/puppet/hieradata/environment/prod/main.yaml
```



Example:

```
apache::port: 80
```

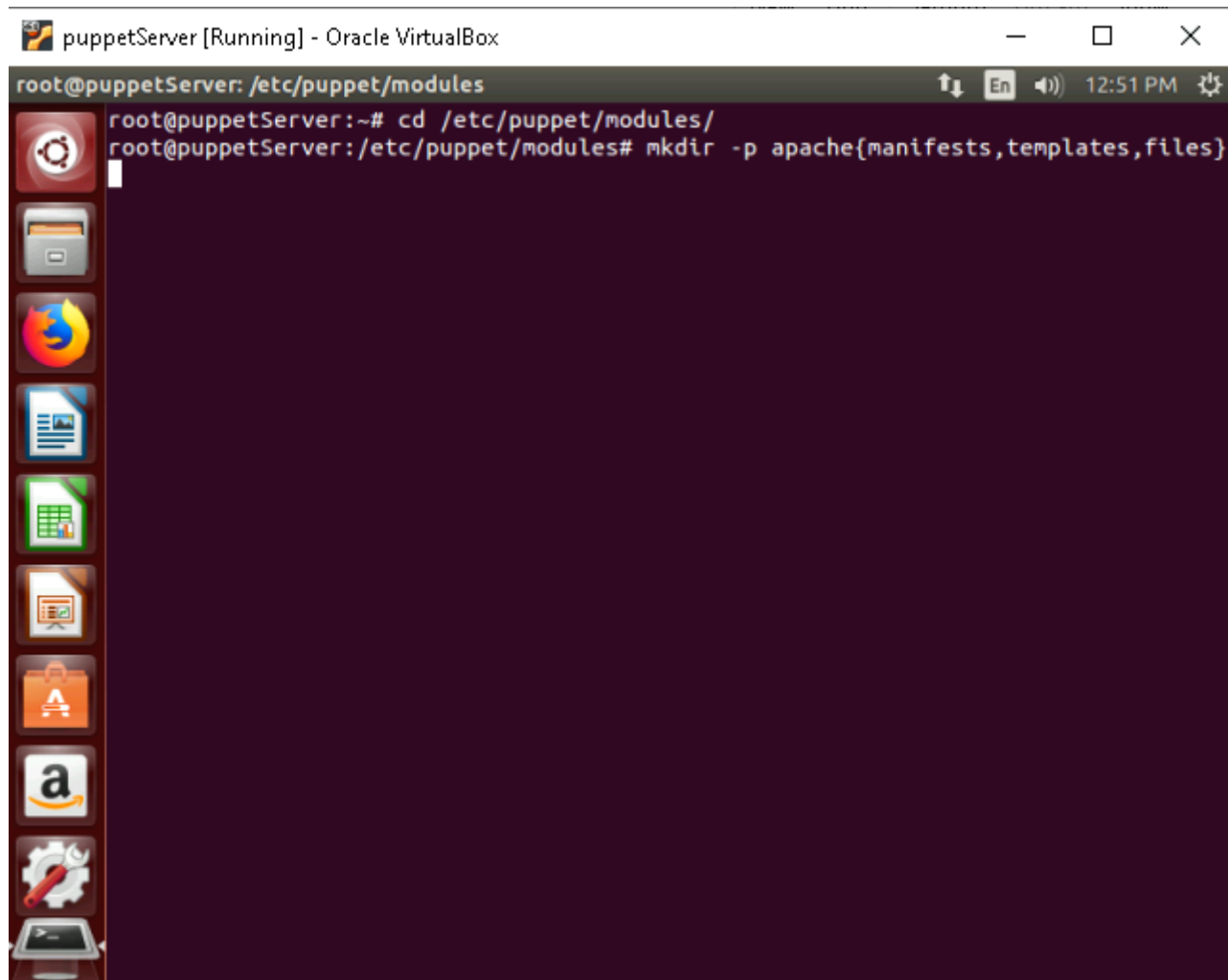


4. Create a Puppet Module

Modules are essential for organizing Puppet manifests and enabling Hieradata integration.

Step 1: Create the Module Directory

```
cd /etc/puppet/modules
mkdir -p apache{manifests,templates,files}
```

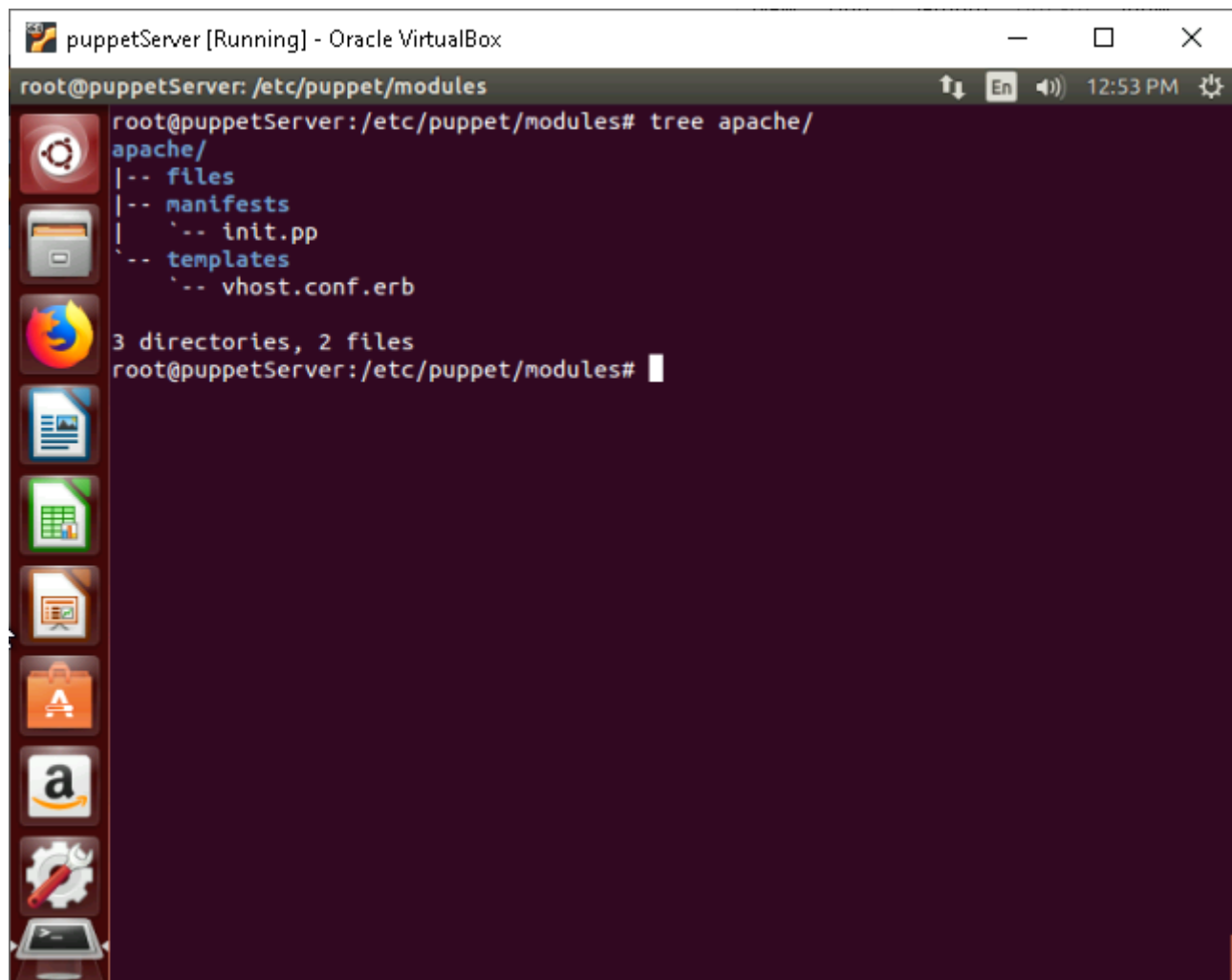


The screenshot shows a terminal window titled "puppetServer [Running] - Oracle VirtualBox". The terminal prompt is "root@puppetServer: /etc/puppet/modules". The user has entered the command "cd /etc/puppet/modules/" and then "mkdir -p apache{manifests,templates,files}". The terminal output shows the directory structure being created. On the left side of the terminal window, there is a vertical dock with various application icons including a terminal, a file manager, a web browser, and others.

```
root@puppetServer: /etc/puppet/modules
root@puppetServer:~# cd /etc/puppet/modules/
root@puppetServer:/etc/puppet/modules# mkdir -p apache{manifests,templates,files}
```

This creates a skeleton module structure:

```
modules/apache/
├── manifests/
├── templates/
└── files/
```



The screenshot shows a terminal window titled "puppetServer [Running] - Oracle VirtualBox". The prompt is "root@puppetServer: /etc/puppet/modules". The command "tree apache/" has been executed, resulting in the following output:

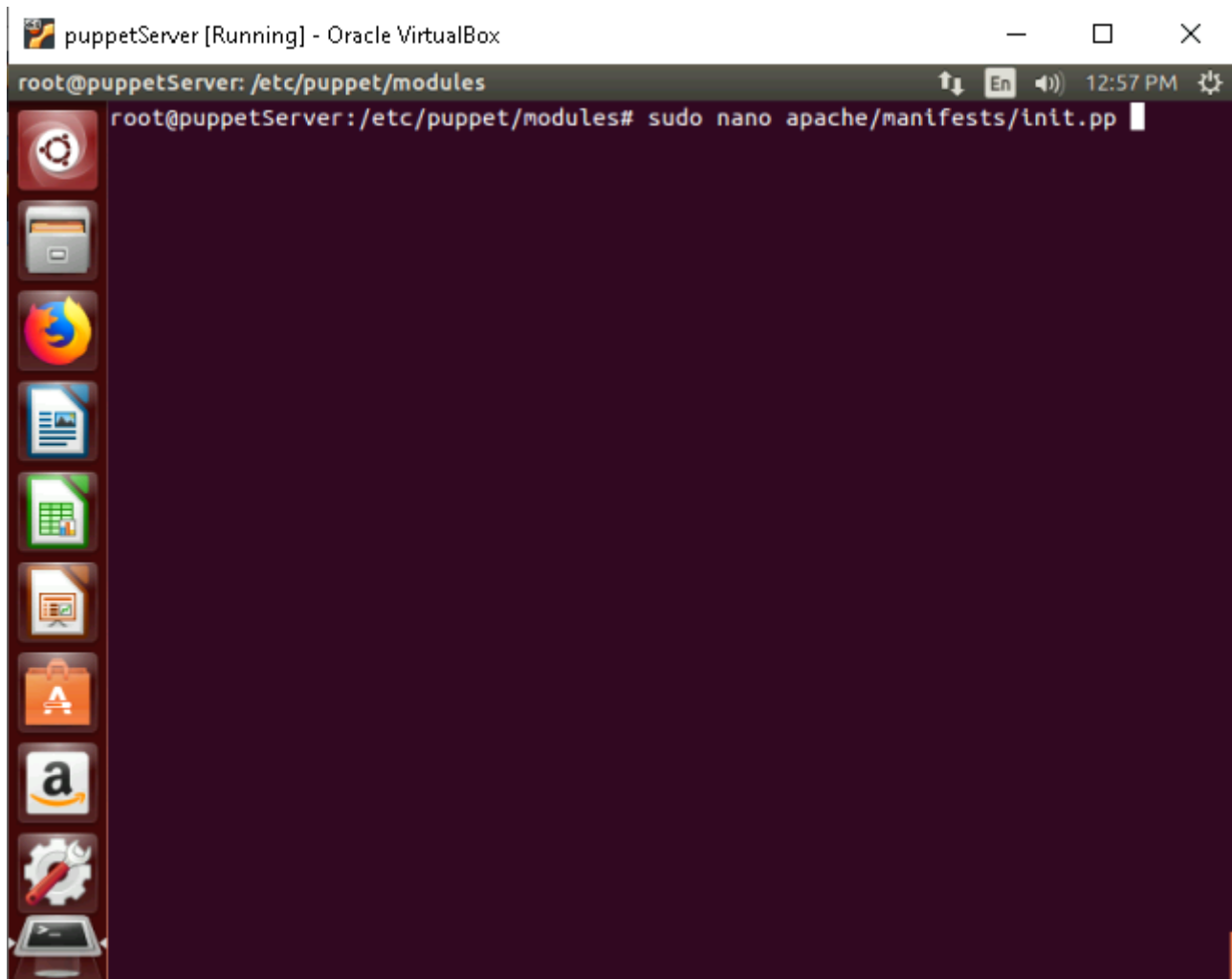
```
root@puppetServer:/etc/puppet/modules# tree apache/
apache/
|-- files
|-- manifests
|   |-- init.pp
|   |-- templates
|   |-- vhost.conf.erb
3 directories, 2 files
root@puppetServer:/etc/puppet/modules#
```

The terminal window has a dark purple background and a vertical sidebar on the left containing various application icons. The top of the window shows standard window controls and system status icons.

Step 2: Define a Manifest

Create a manifest to define your class in `manifests/init.pp`:

```
sudo nano apache/manifests/init.pp
```



```
class apache (
  $port      = hiera('apache::port', 80),
  $docroot   = hiera('apache::docroot', '/var/www/html'),
) {
  package { 'apache2':
    ensure => installed,
  }

  service { 'apache2':
    ensure    => running,
    enable    => true,
    require   => Package['apache2'],
  }

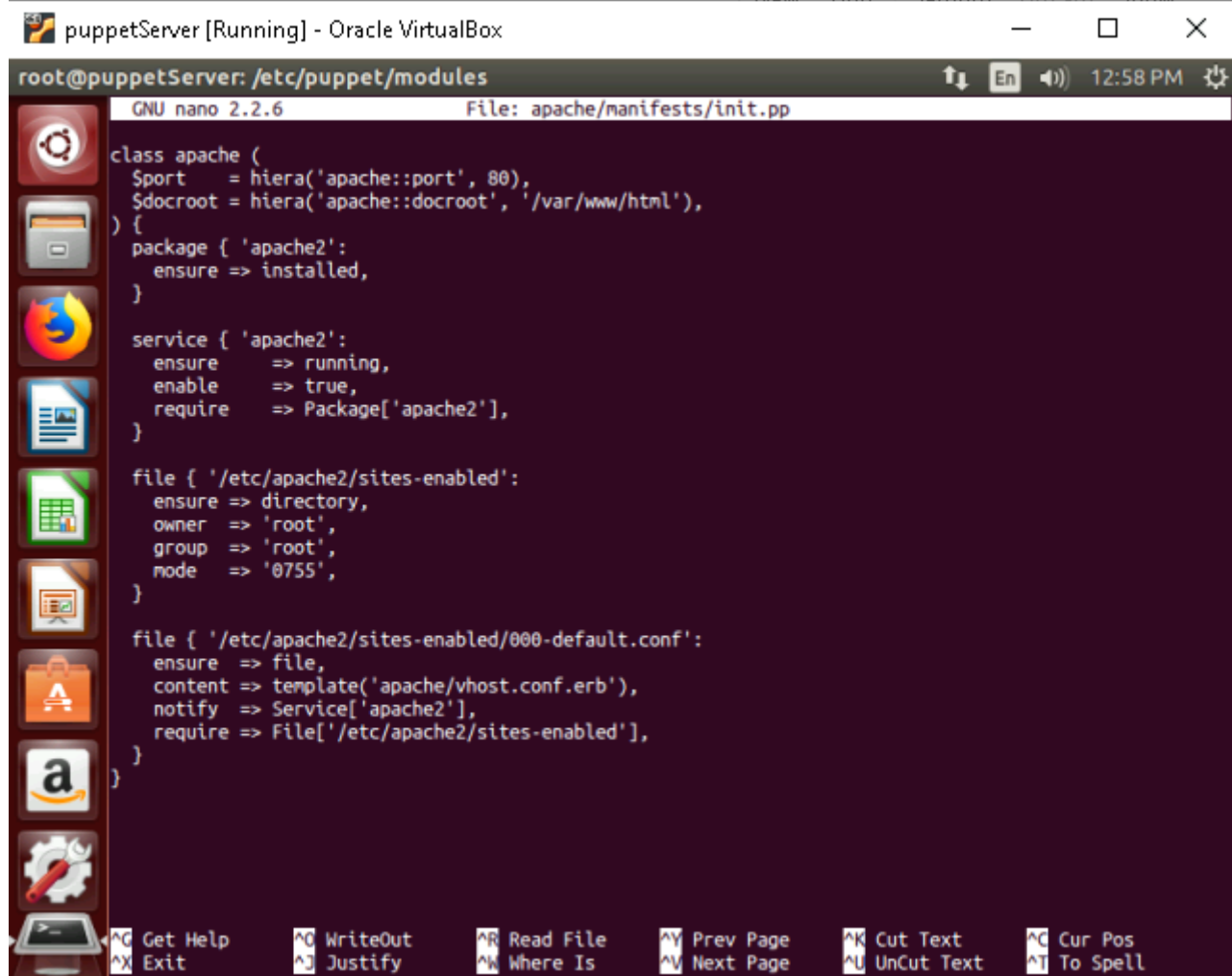
  file { ['/etc/apache2/sites-enabled':
    ensure => directory,
    owner  => 'root',
    group  => 'root',
    mode   => '0755',
  ]

  file { ['/etc/apache2/sites-enabled/000-default.conf':
    ensure => file,
    content => template('apache/vhost.conf.erb'),
  ]
}
```

```

    notify => Service['apache2'],
    require => File['/etc/apache2/sites-enabled'], # Ensure the directory exists
  first
}
}

```



The screenshot shows a terminal window titled "puppetServer [Running] - Oracle VirtualBox". The terminal is running the GNU nano 2.2.6 editor, editing the file "apache/manifests/init.pp". The code defines a class "apache" with the following attributes and dependencies:

```

class apache (
  $port    = hiera('apache::port', 80),
  $docroot = hiera('apache::docroot', '/var/www/html'),
) {
  package { 'apache2':
    ensure => installed,
  }

  service { 'apache2':
    ensure => running,
    enable => true,
    require => Package['apache2'],
  }

  file { ['/etc/apache2/sites-enabled':
    ensure => directory,
    owner  => 'root',
    group  => 'root',
    mode   => '0755',
  ]

  file { ['/etc/apache2/sites-enabled/000-default.conf':
    ensure => file,
    content => template('apache/vhost.conf.erb'),
    notify  => Service['apache2'],
    require => File['/etc/apache2/sites-enabled'],
  ]
}

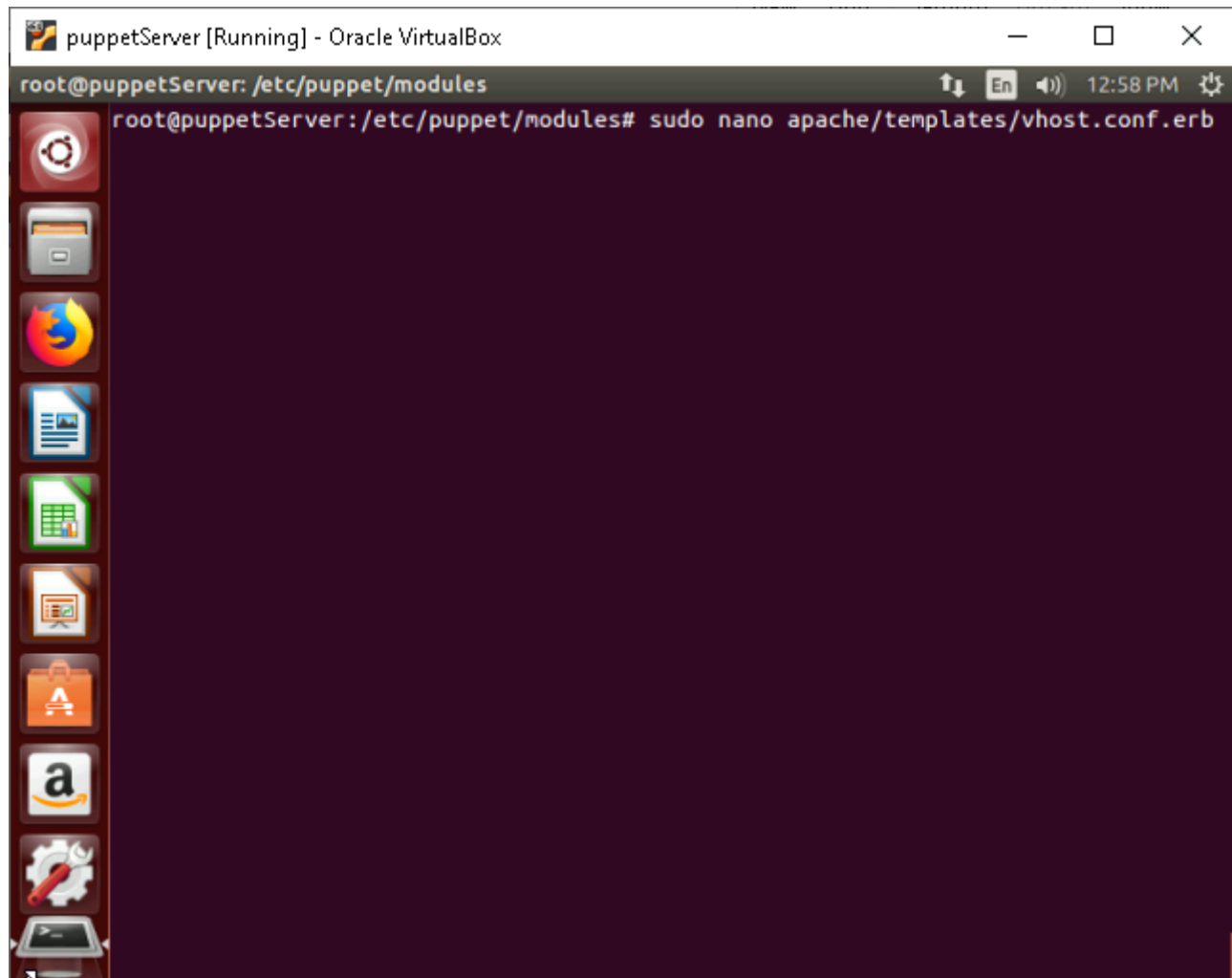
```

The terminal window also shows a sidebar with various application icons and a bottom status bar with navigation and editing shortcuts.

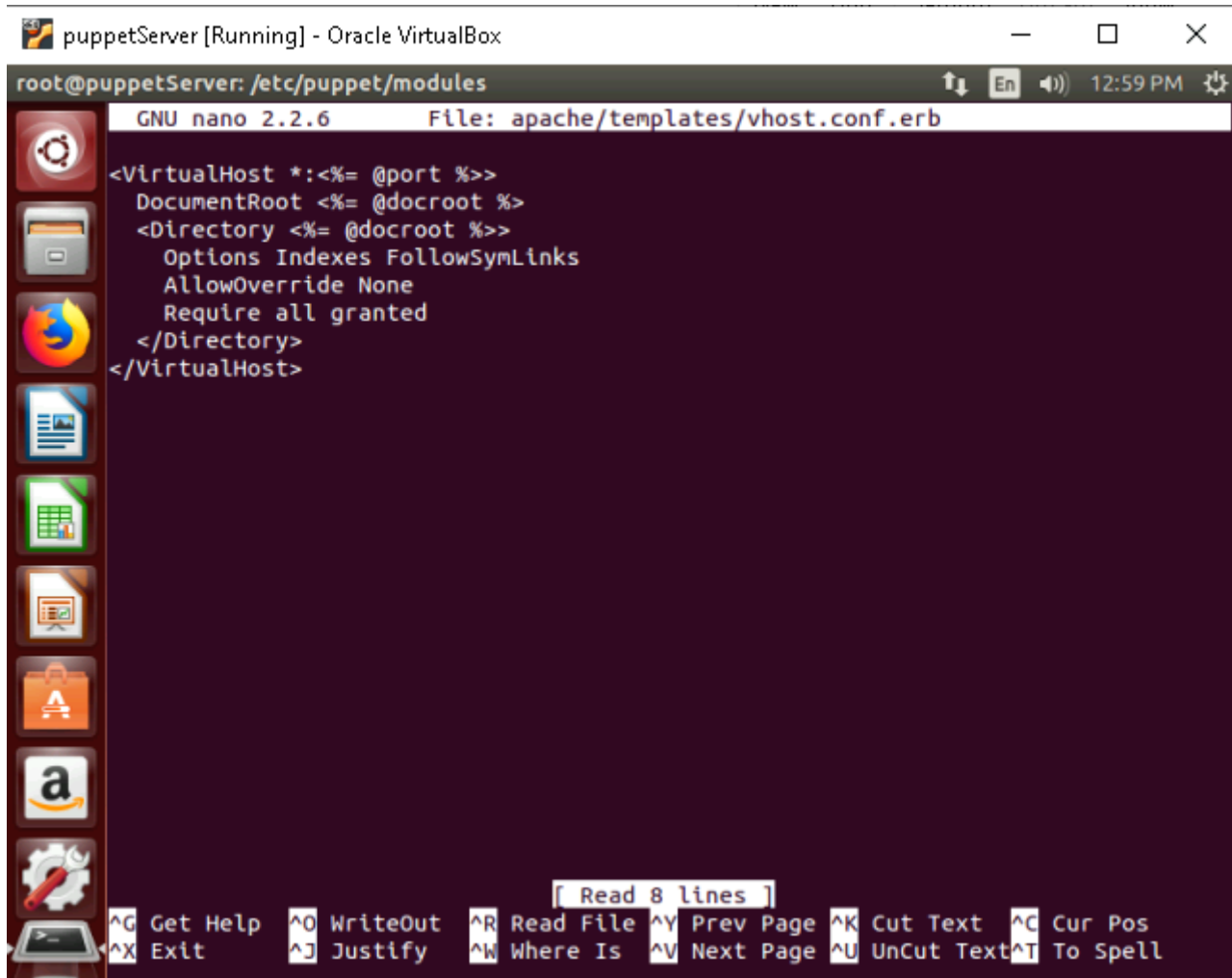
Step 3: Add a Template

Create a file `templates/vhost.conf.erb` for the virtual host configuration:

```
sudo nano apache/templates/vhost.conf.erb
```



```
<VirtualHost *:<%= @port %>>
  DocumentRoot <%= @docroot %>
  <Directory <%= @docroot %>>
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
  </Directory>
</VirtualHost>
```

```
puppetServer [Running] - Oracle VirtualBox
root@puppetServer: /etc/puppet/modules
GNU nano 2.2.6 File: apache/templates/vhost.conf.erb

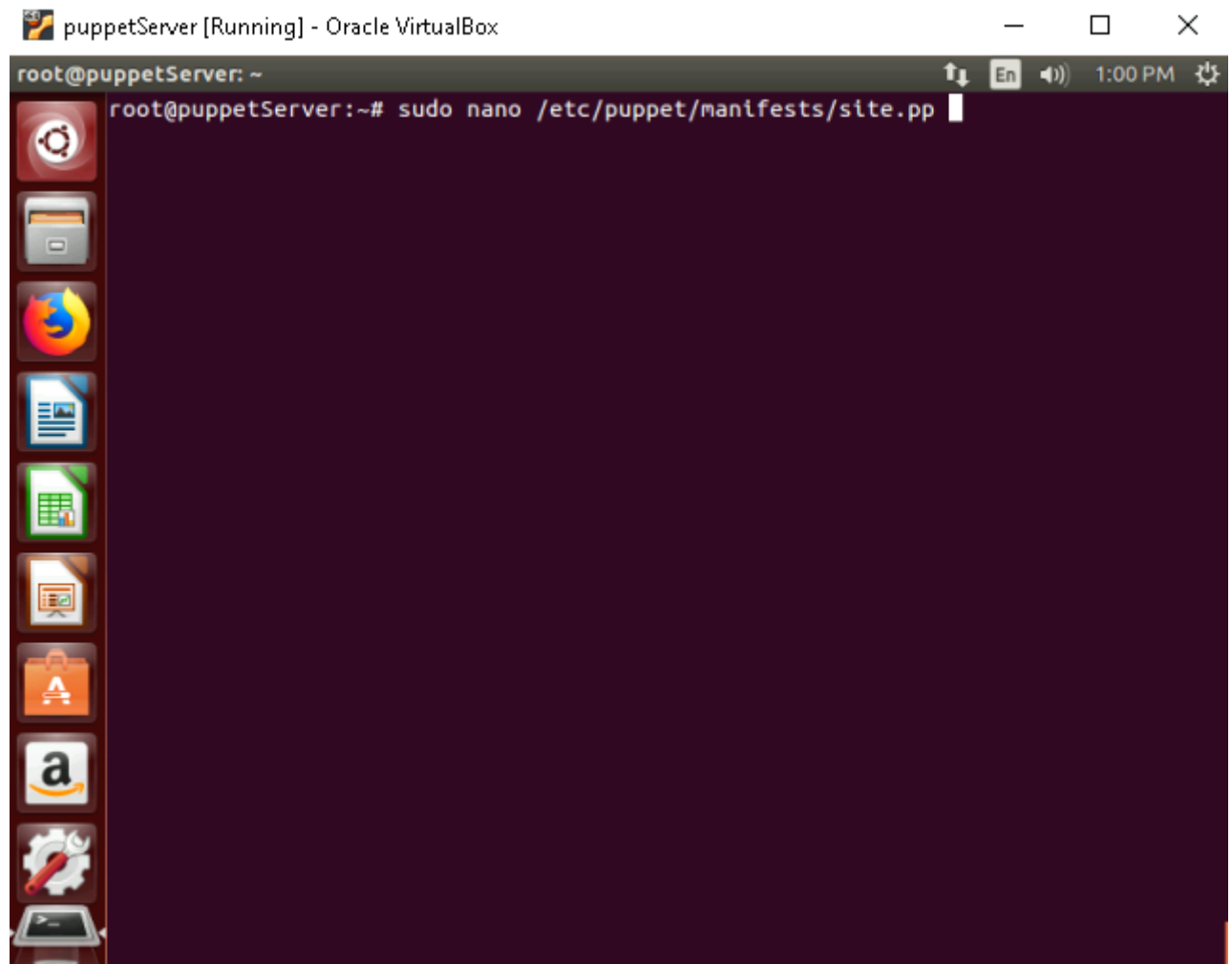
<VirtualHost *:<%= @port %>>
  DocumentRoot <%= @docroot %>
  <Directory <%= @docroot %>>
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
  </Directory>
</VirtualHost>

[ Read 8 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
```

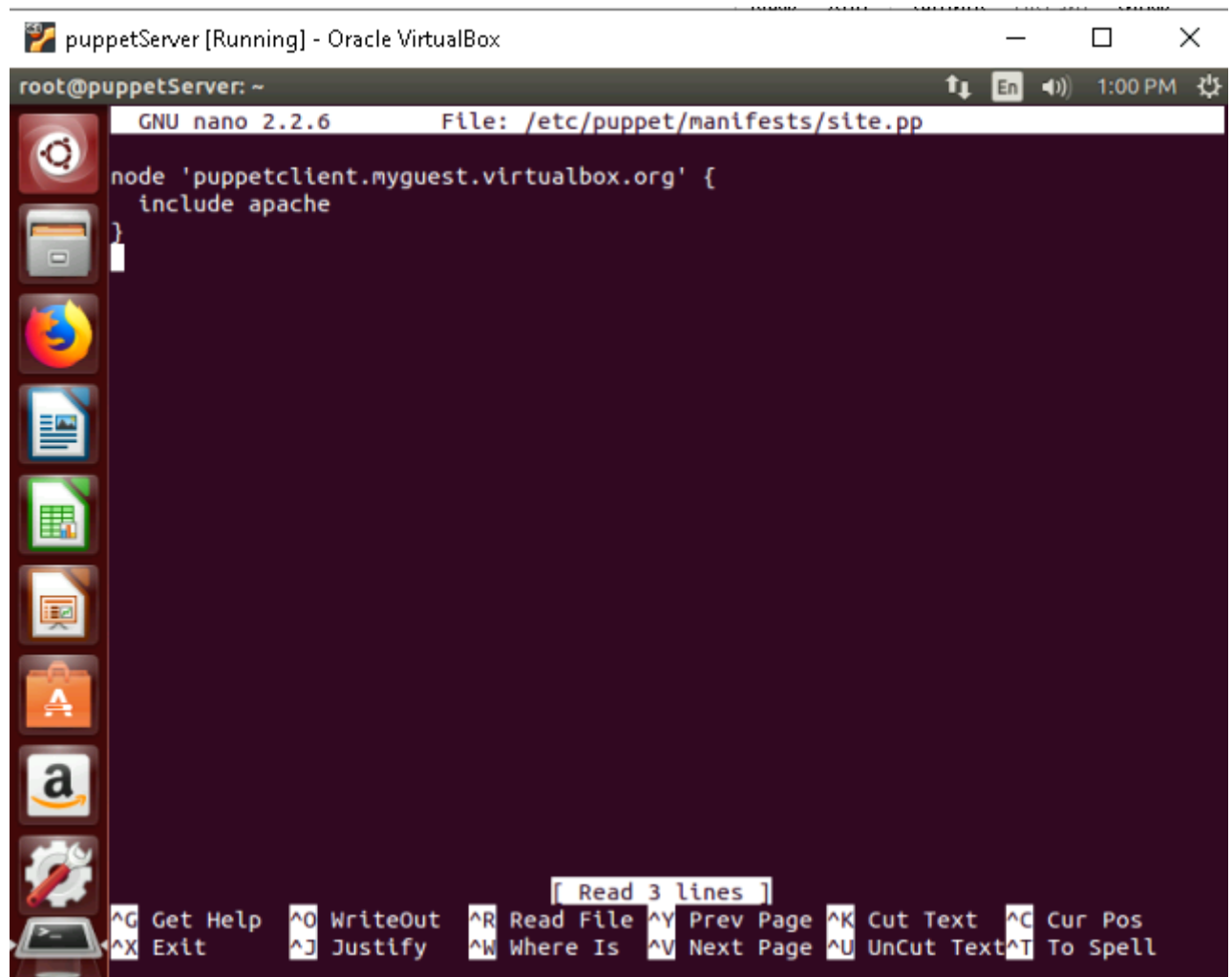
5. Apply the Module

1. Include the module in a node manifest:

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



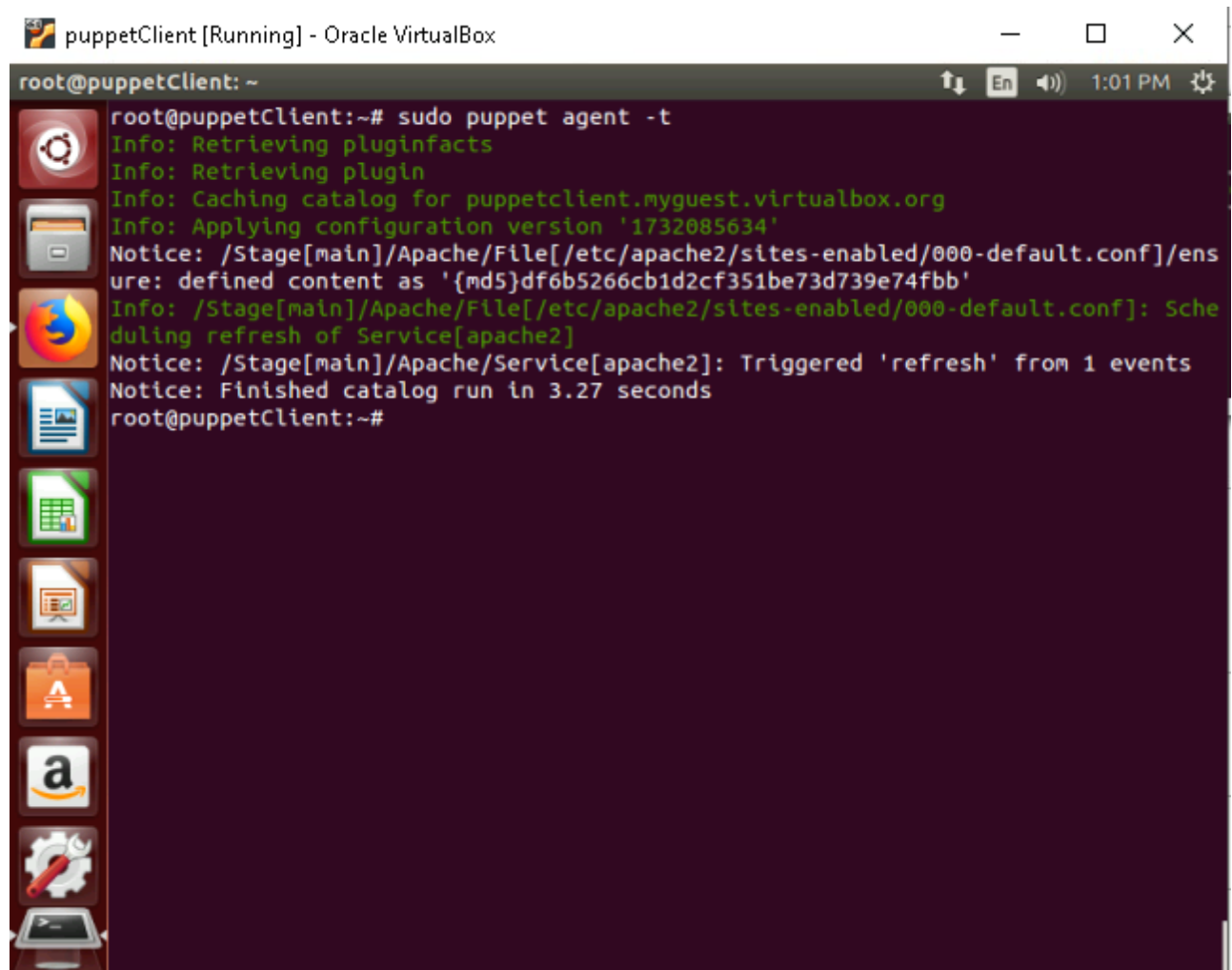
```
node 'puppetclient.myguest.virtualbox.org' {  
  include apache  
}
```



```
puppetServer [Running] - Oracle VirtualBox
root@puppetServer: ~
GNU nano 2.2.6 File: /etc/puppet/manifests/site.pp
node 'puppetclient.myguest.virtualbox.org' {
  include apache
}
[ Read 3 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. Run Puppet to apply the changes:

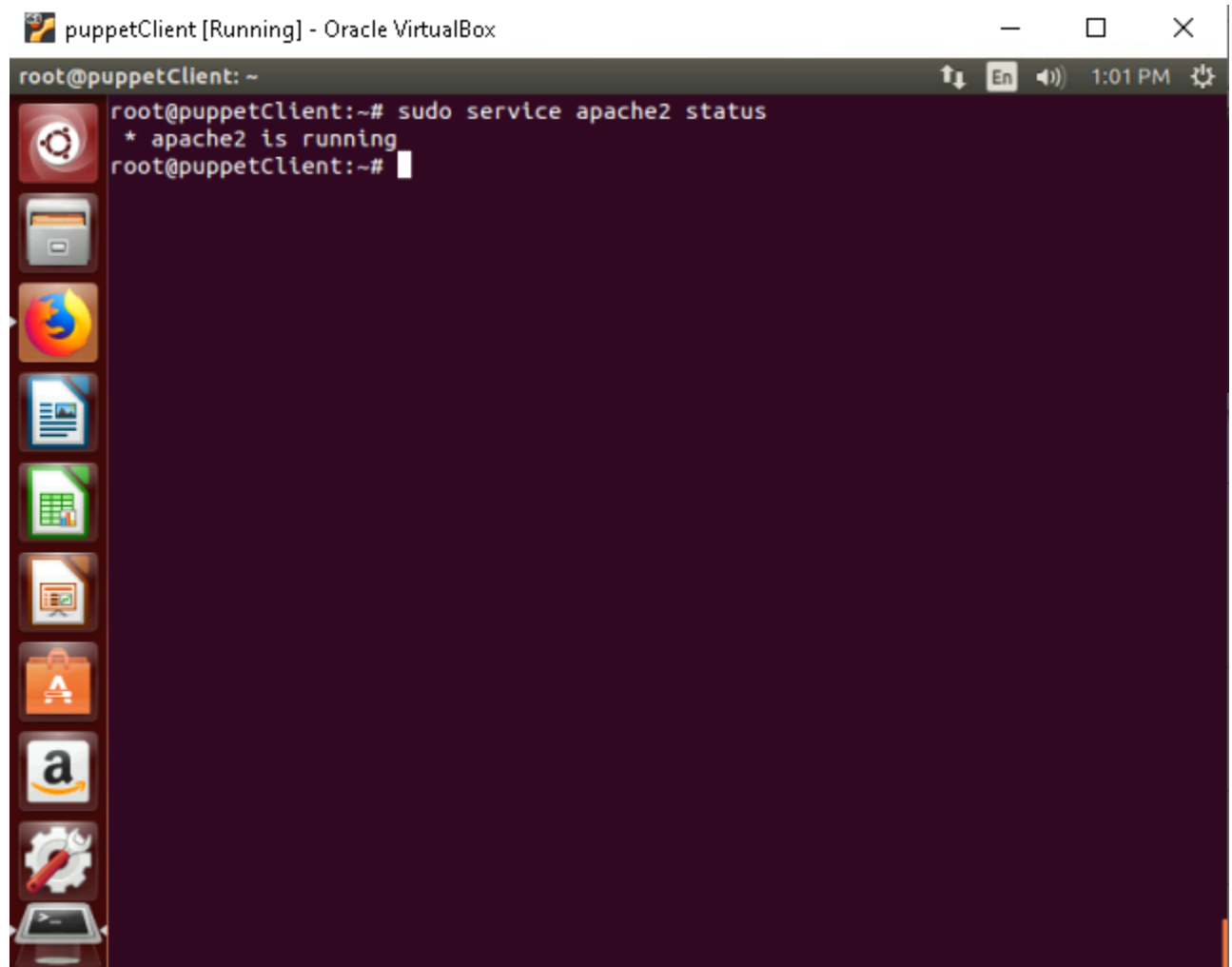
```
puppet agent --test
```



```
root@puppetClient: ~  
root@puppetClient:~# sudo puppet agent -t  
Info: Retrieving pluginfacts  
Info: Retrieving plugin  
Info: Caching catalog for puppetclient.myguest.virtualbox.org  
Info: Applying configuration version '1732085634'  
Notice: /Stage[main]/Apache/File[/etc/apache2/sites-enabled/000-default.conf]/ensure: defined content as '{md5}df6b5266cb1d2cf351be73d739e74fbb'  
Info: /Stage[main]/Apache/File[/etc/apache2/sites-enabled/000-default.conf]: Scheduling refresh of Service[apache2]  
Notice: /Stage[main]/Apache/Service[apache2]: Triggered 'refresh' from 1 events  
Notice: Finished catalog run in 3.27 seconds  
root@puppetClient:~#  
root@puppetClient:~# sudo service apache2 status
```

3. Verify the Apache service is running:

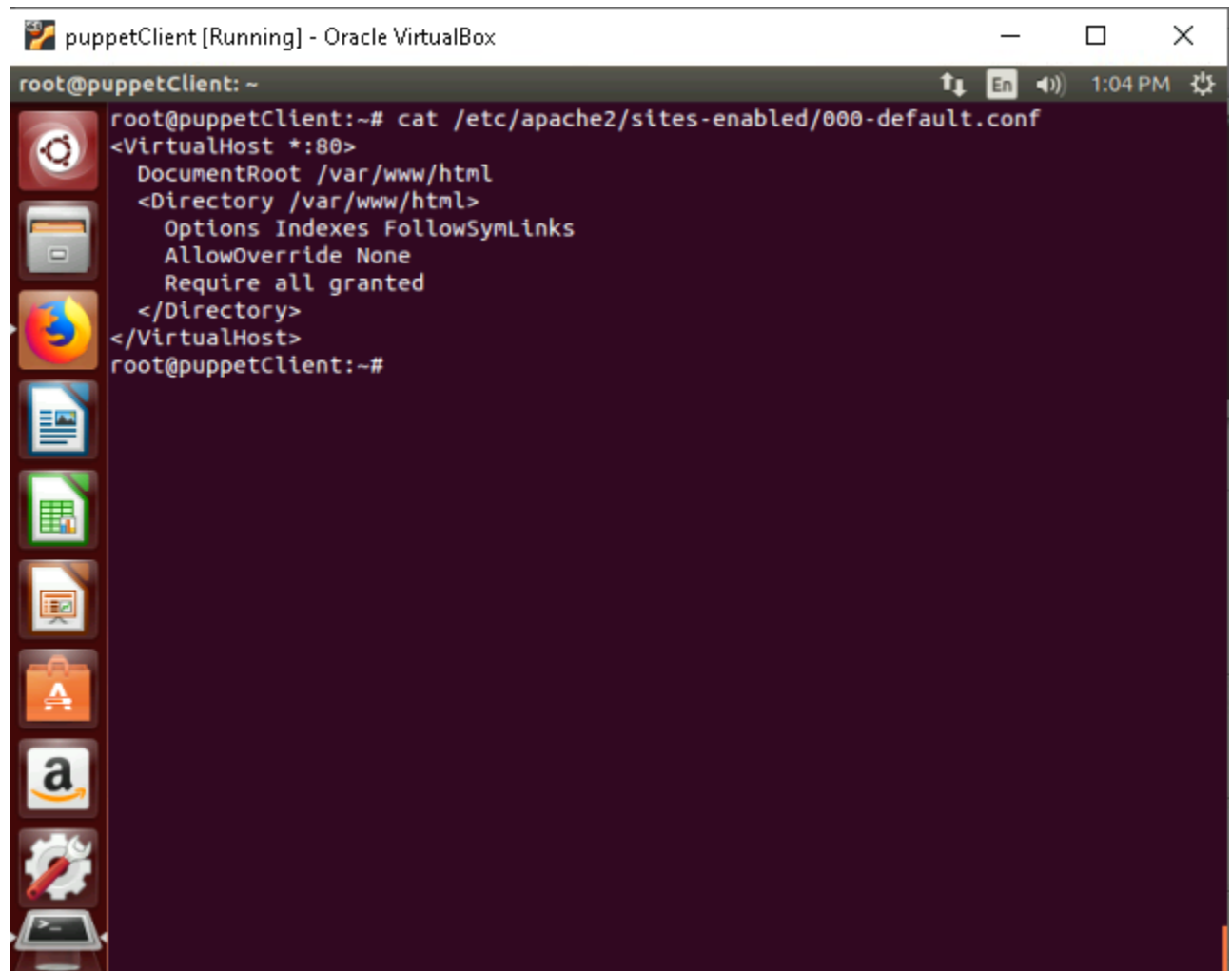
```
sudo service apache2 status
```



```
root@puppetClient: ~  
root@puppetClient:~# sudo service apache2 status  
* apache2 is running  
root@puppetClient:~#
```

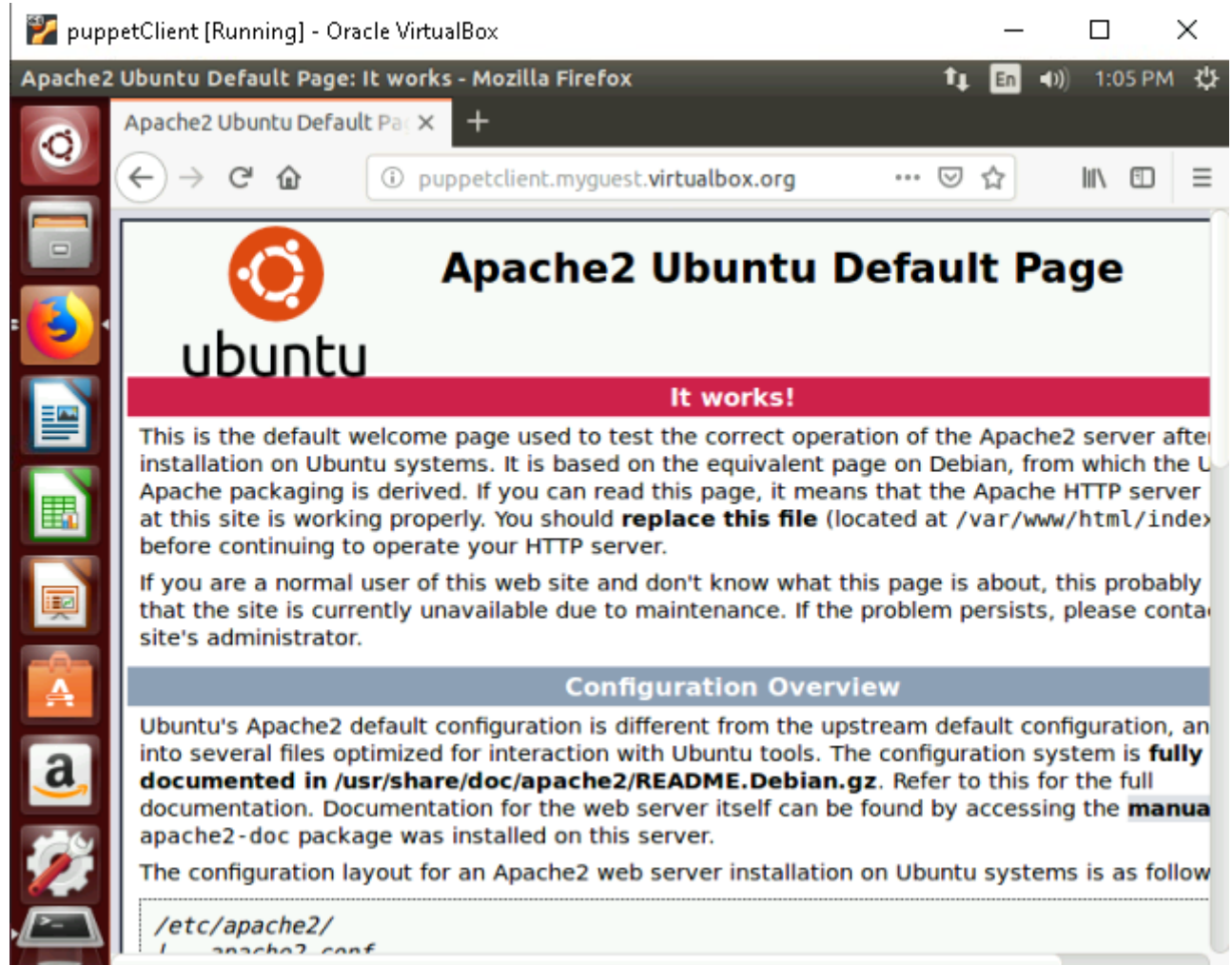
4. Check the virtual host configuration file:

```
cat /etc/apache2/sites-enabled/000-default.conf
```



```
puppetClient [Running] - Oracle VirtualBox
root@puppetClient: ~
root@puppetClient:~# cat /etc/apache2/sites-enabled/000-default.conf
<VirtualHost *:80>
  DocumentRoot /var/www/html
  <Directory /var/www/html>
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
  </Directory>
</VirtualHost>
root@puppetClient:~#
```

5. Test the Apache server by browsing to <http://puppetclient.myguest.virtualbox.org> in a web browser.



References

- [Puppet Hiera Documentation](#)
 - [Linux Journal: Using Hiera with Puppet](#)
-