

# Write and apply a basic Puppet manifest to manage a service on a node

---

## Table of Contents

1. **Introduction**
  2. **Problem Statement**
  3. **Prerequisites**
    - **Software Requirements**
    - **Hardware Requirements**
  4. **Implementation Steps**
    - **Step 1: Verify Communication Between Puppet Master and Agent**
    - **Step 2: Create the Puppet Manifest**
    - **Step 3: Apply the Manifest**
    - **Step 4: Configure Puppet Agent to Apply the Manifest**
    - **Step 5: Verify the Configuration**
  5. **References**
- 

## Introduction

In this guide, we will walk through the steps of writing and applying a basic Puppet manifest to manage a service (e.g., `nginx`) on a node. Puppet automates the process of managing software packages and services across a fleet of machines, and this tutorial will show you how to use Puppet to ensure that a service is installed, running, and enabled to start at boot.

---

## Problem Statement

Many systems administrators face the challenge of manually installing and managing services across multiple nodes. This guide provides a solution by using Puppet to automate the process of managing services on a node, ensuring consistency and reducing the potential for human error.

---

## Prerequisites

Completion of all previous lab guides (up to Lab Guide-01) is required before proceeding with Lab Guide-02.

---

## Software Requirements

- Puppet Master installed and configured.
  - Puppet Agent installed on the target node.
  - A machine with Ubuntu (or any Linux distribution) for both Puppet Master and Agent.
  - `nginx` service package available for installation.
-

## Hardware Requirements

- A minimum of 2 machines or virtual machines (VMs) for the Puppet Master and Agent.
- Network connection between Puppet Master and Puppet Agent.

---

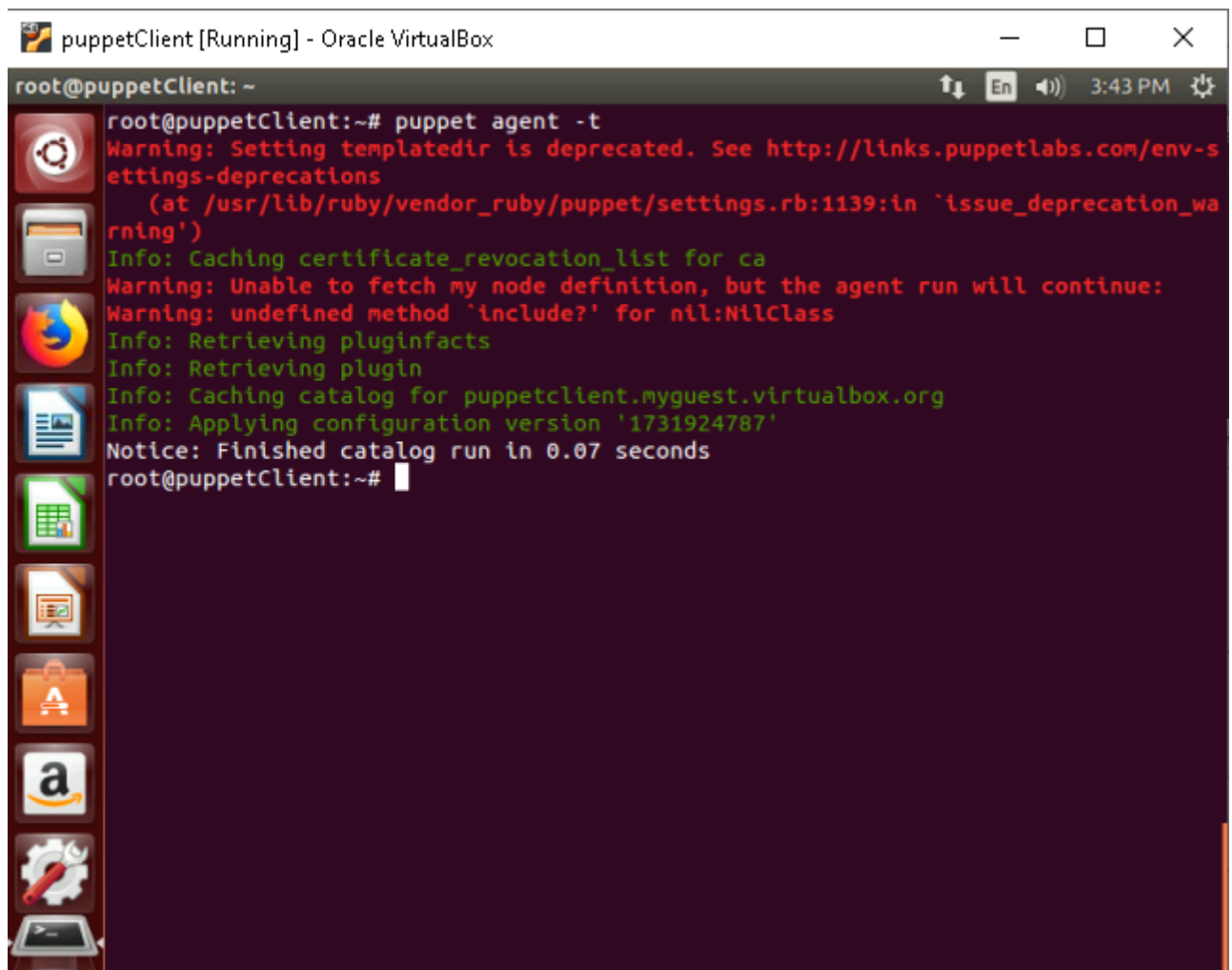
## Implementation Steps

### Step 1: Verify Communication Between Puppet Master and Agent

Before applying any manifest, ensure that the Puppet Master and Agent can communicate without issues.

- **On the Puppet Agent node**, run the following command:

```
puppet agent -t
```



```
root@puppetClient: ~  
root@puppetClient:~# puppet agent -t  
Warning: Setting templatedir is deprecated. See http://links.puppetlabs.com/env-  
settings-deprecations  
      (at /usr/lib/ruby/vendor_ruby/puppet/settings.rb:1139:in `issue_deprecation_wa  
rning')  
Info: Caching certificate_revocation_list for ca  
Warning: Unable to fetch my node definition, but the agent run will continue:  
Warning: undefined method `include?' for nil:NilClass  
Info: Retrieving pluginfacts  
Info: Retrieving plugin  
Info: Caching catalog for puppetclient.myguest.virtualbox.org  
Info: Applying configuration version '1731924787'  
Notice: Finished catalog run in 0.07 seconds  
root@puppetClient:~#
```

*Explanation:* This command tells the Puppet Agent to connect to the Puppet Master and check for any configuration changes. It should output success if the communication is working correctly.

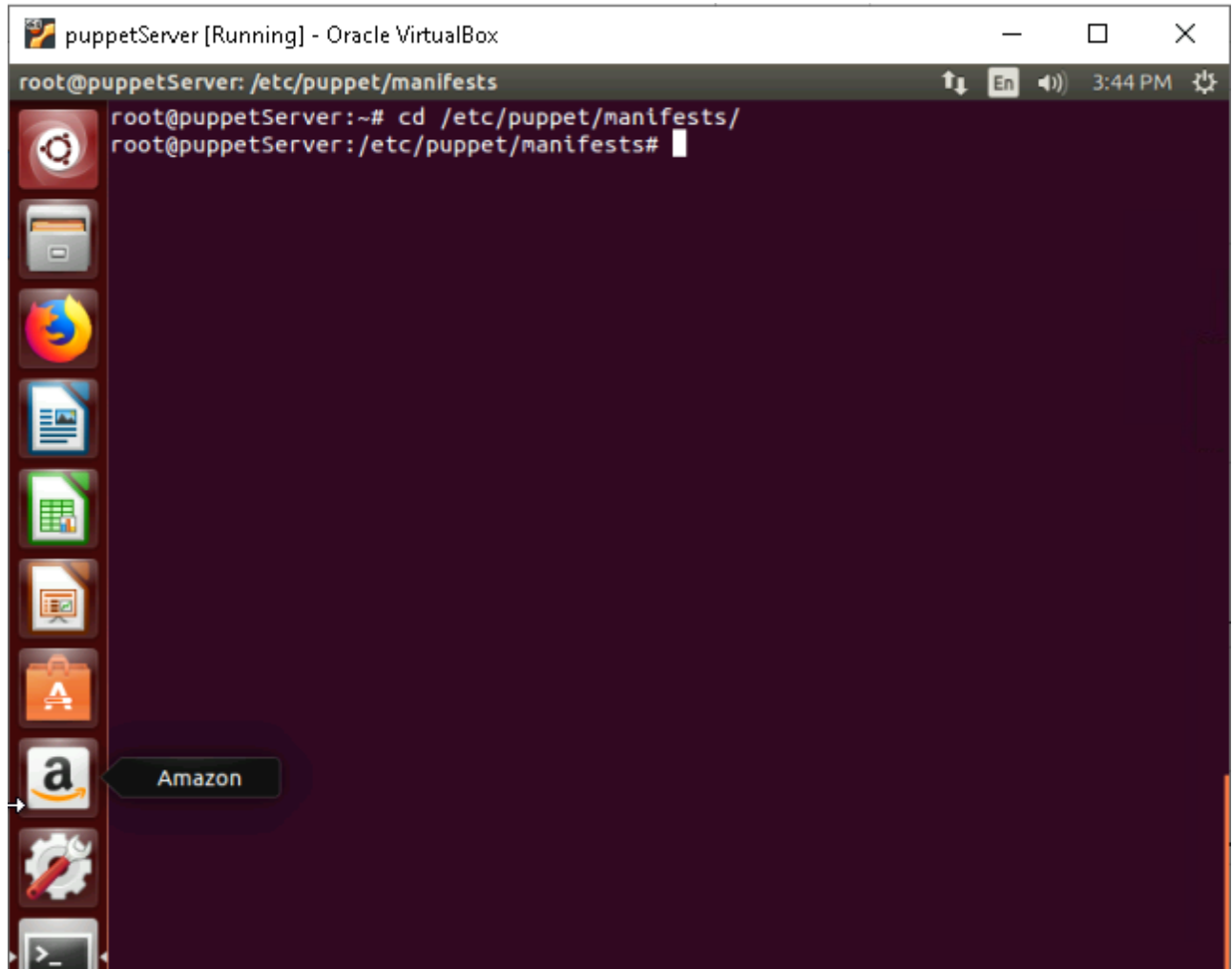
---

### Step 2: Create the Puppet Manifest

Now, we'll write a Puppet manifest to install and manage the `nginx` service.

1. **On the Puppet Master**, navigate to the Puppet manifest directory:

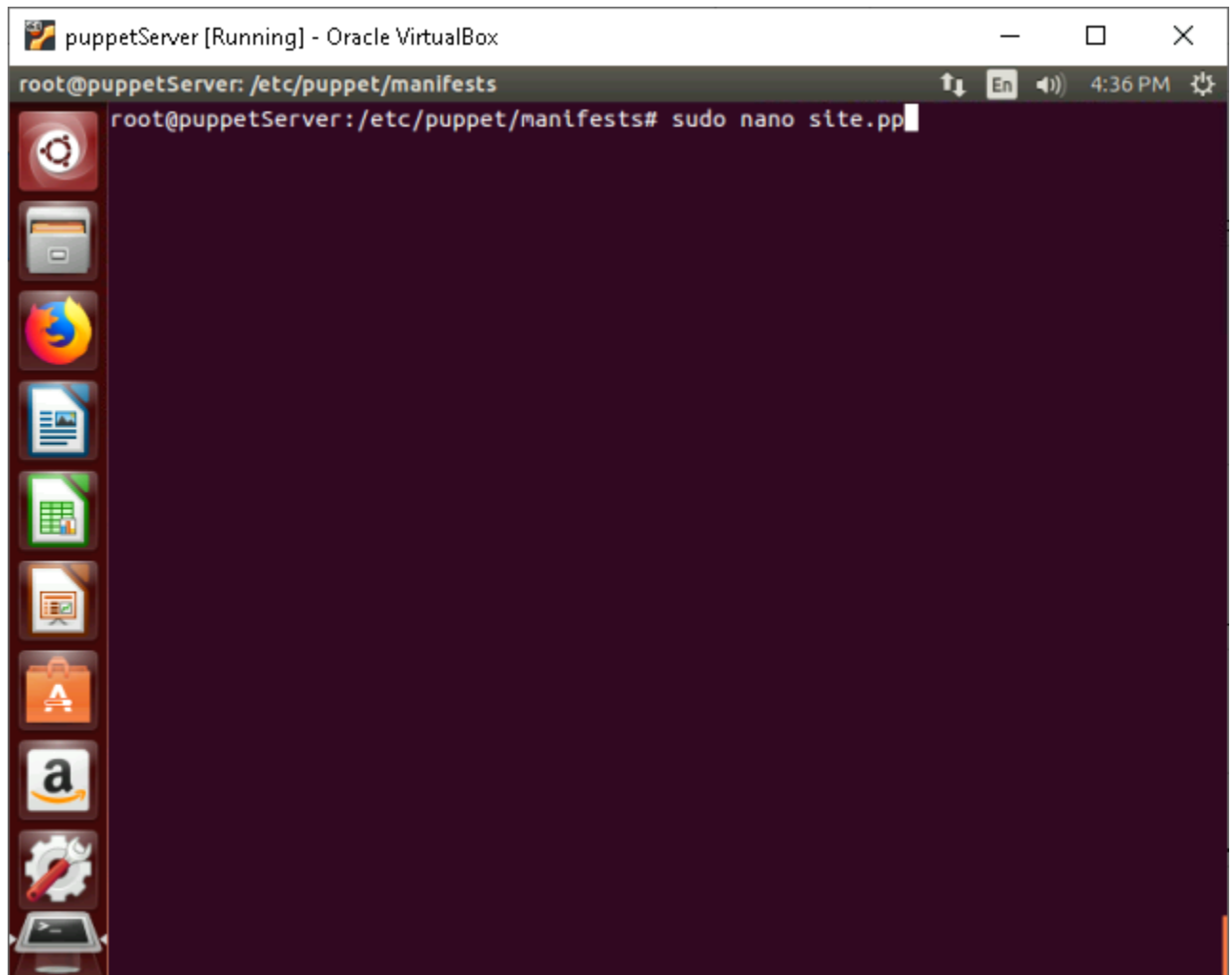
```
cd /etc/puppet/manifests
```



- *Explanation:* This is where all the Puppet manifests are stored, and we will create our new manifest in this directory.

2. **Create a new file** called `site.pp`:

```
sudo nano site.pp
```



### 3. Write the manifest to install and manage **nginx**:

```
# service_management.pp

# Ensure nginx is installed
package { 'nginx':
  ensure => installed,
}

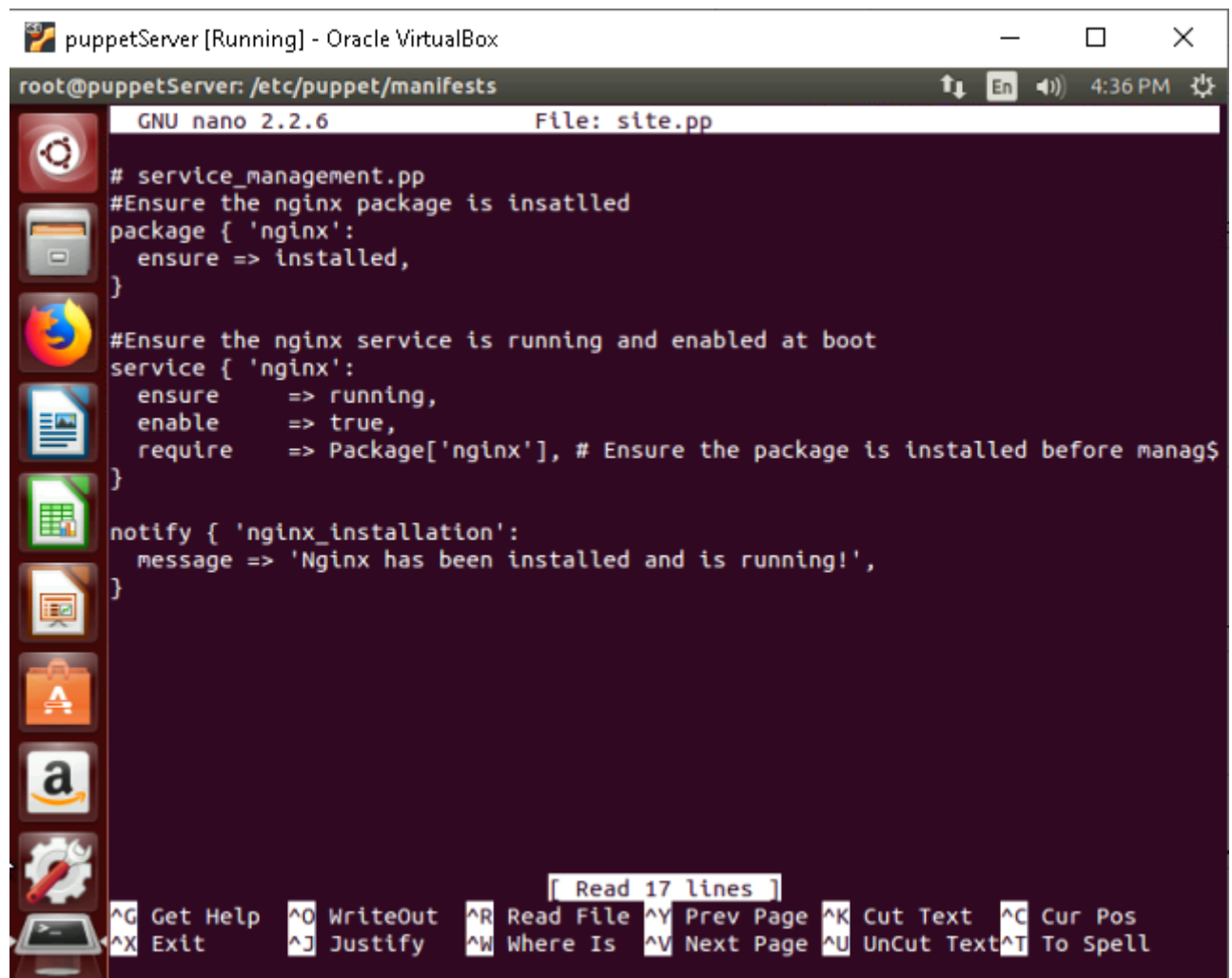
# Ensure nginx service is running and enabled at boot
service { 'nginx':
  ensure    => running,
  enable    => true,
  require   => Package['nginx'], # Make sure the package is installed first
}

notify { 'nginx_installation':
  message => 'Nginx has been installed and is running!',
}
```

#### ◦ Explanation:

- The **package** resource ensures that the **nginx** package is installed on the target node.

- The `service` resource ensures that the `nginx` service is running and enabled at boot. The `require` statement ensures that the package is installed before the service is started.



```
GNU nano 2.2.6 File: site.pp

# service_management.pp
#Ensure the nginx package is insatlled
package { 'nginx':
  ensure => installed,
}

#Ensure the nginx service is running and enabled at boot
service { 'nginx':
  ensure      => running,
  enable      => true,
  require     => Package['nginx'], # Ensure the package is installed before manag$
}

notify { 'nginx_installation':
  message => 'Nginx has been installed and is running!',
}

[ Read 17 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
```

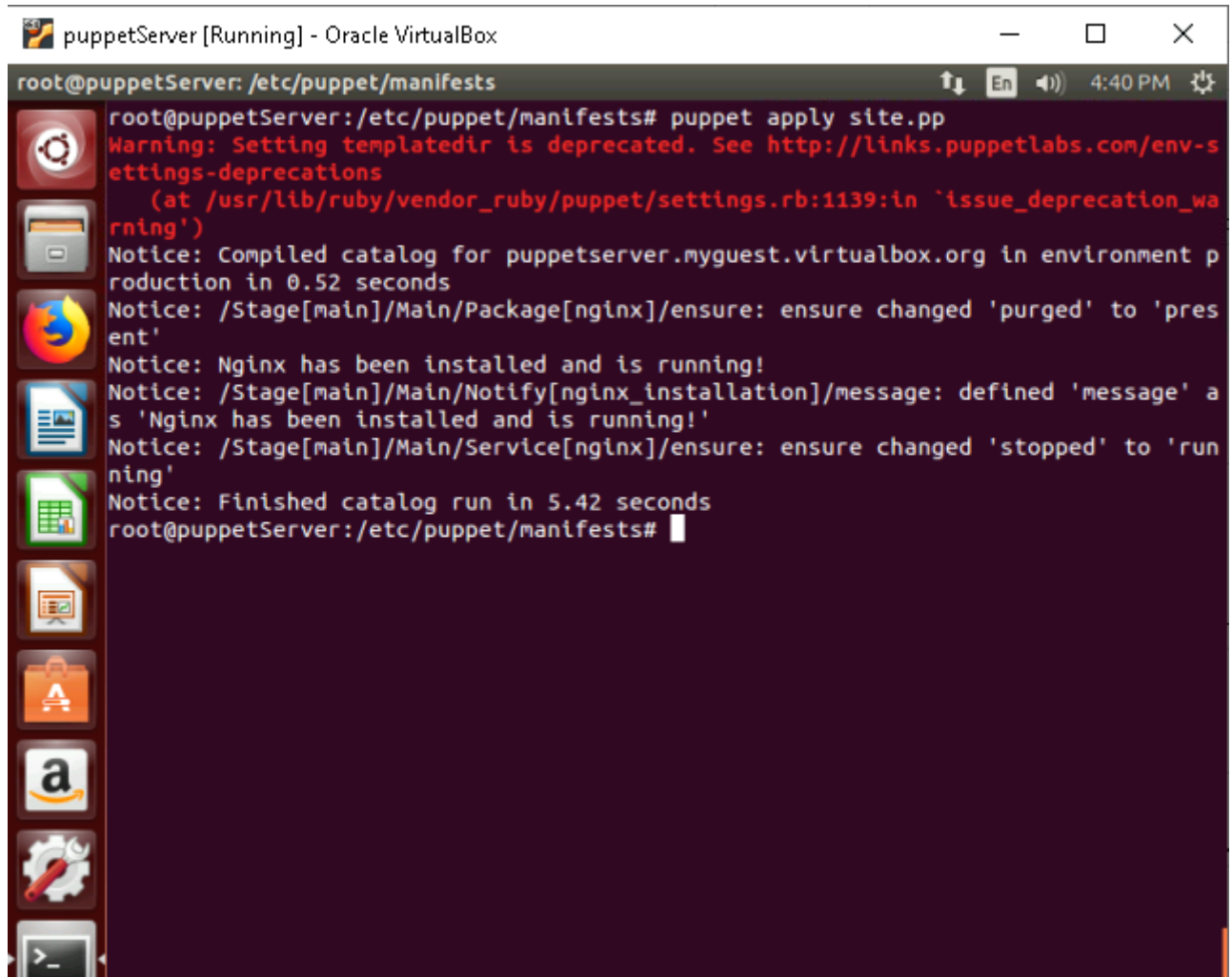
4. Save and exit the editor.

---

## Step 3: Apply the Manifest

1. Test the manifest on the Puppet Master:

```
puppet apply site.pp
```



```
root@puppetServer: /etc/puppet/manifests
root@puppetServer:/etc/puppet/manifests# puppet apply site.pp
Warning: Setting templatedir is deprecated. See http://links.puppetlabs.com/env-s
ettings-deprecations
(at /usr/lib/ruby/vendor_ruby/puppet/settings.rb:1139:in `issue_deprecation_wa
rning')
Notice: Compiled catalog for puppetserver.myguest.virtualbox.org in environment p
roduction in 0.52 seconds
Notice: /Stage[main]/Main/Package[nginx]/ensure: ensure changed 'purged' to 'pres
ent'
Notice: Nginx has been installed and is running!
Notice: /Stage[main]/Main/Notify[nginx_installation]/message: defined 'message' a
s 'Nginx has been installed and is running!'
Notice: /Stage[main]/Main/Service[nginx]/ensure: ensure changed 'stopped' to 'run
ning'
Notice: Finished catalog run in 5.42 seconds
root@puppetServer:/etc/puppet/manifests#
```

- *Explanation:* This command applies the manifest locally on the Puppet Master to check if there are any errors before deploying it to the Puppet Agent.

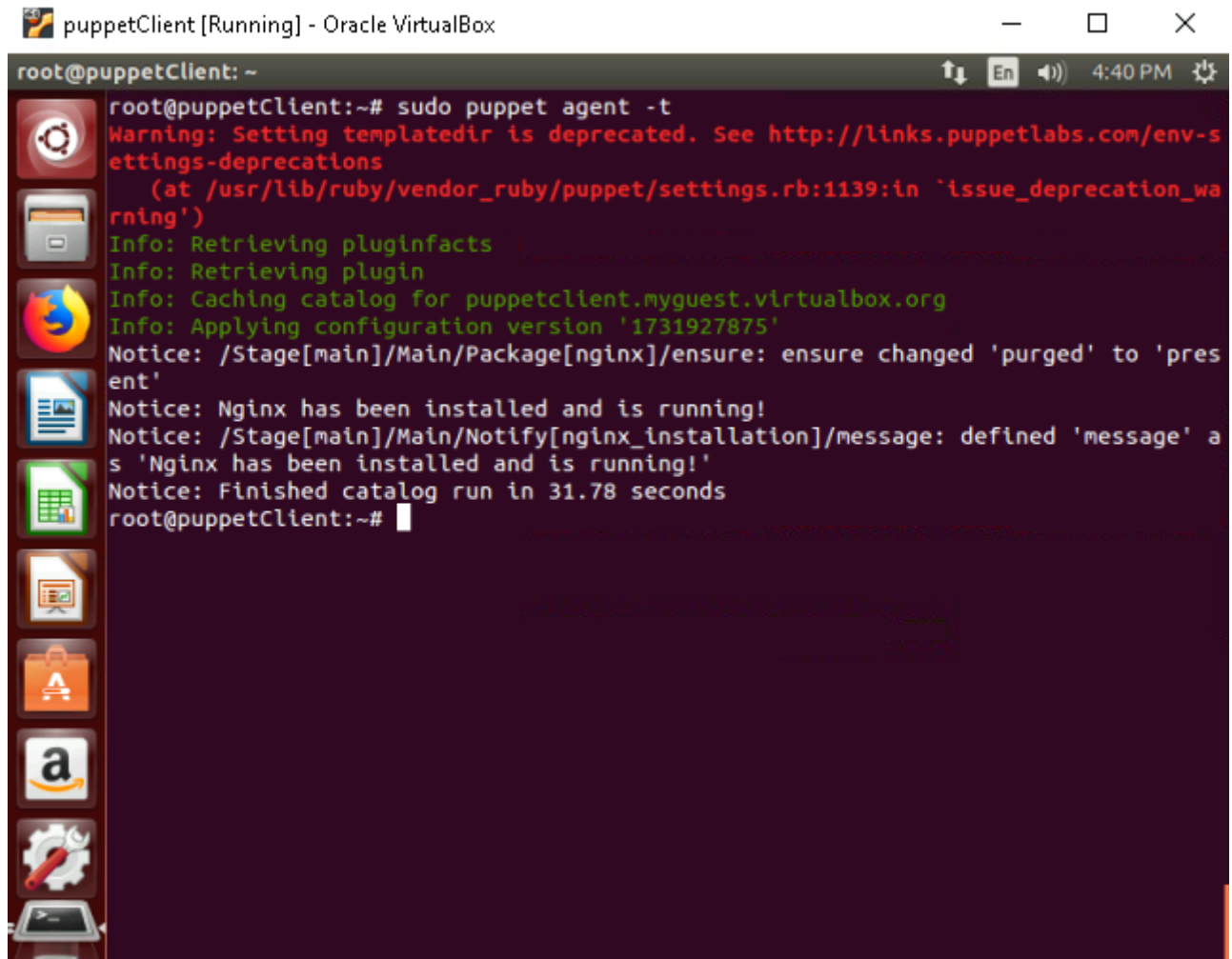
---

## Step 4: Configure Puppet Agent to Apply the Manifest

After testing the manifest locally, now we need to make sure that the Puppet Agent receives and applies the manifest.

1. **Trigger the Puppet Agent** to apply the configuration:

```
puppet agent -t
```



The screenshot shows a terminal window titled "puppetClient [Running] - Oracle VirtualBox". The terminal output is as follows:

```
root@puppetClient: ~  
root@puppetClient:~# sudo puppet agent -t  
Warning: Setting templatedir is deprecated. See http://links.puppetlabs.com/env-  
settings-deprecations  
(at /usr/lib/ruby/vendor_ruby/puppet/settings.rb:1139:in `issue_deprecation_wa  
rning')  
Info: Retrieving pluginfacts  
Info: Retrieving plugin  
Info: Caching catalog for puppetclient.myguest.virtualbox.org  
Info: Applying configuration version '1731927875'  
Notice: /Stage[main]/Main/Package[nginx]/ensure: ensure changed 'purged' to 'pres  
ent'  
Notice: Nginx has been installed and is running!  
Notice: /Stage[main]/Main/Notify[nginx_installation]/message: defined 'message' a  
s 'Nginx has been installed and is running!'  
Notice: Finished catalog run in 31.78 seconds  
root@puppetClient:~#
```

- *Explanation:* This command initiates the Puppet Agent, which will connect to the Puppet Master and apply the manifest defined by the Master.

---

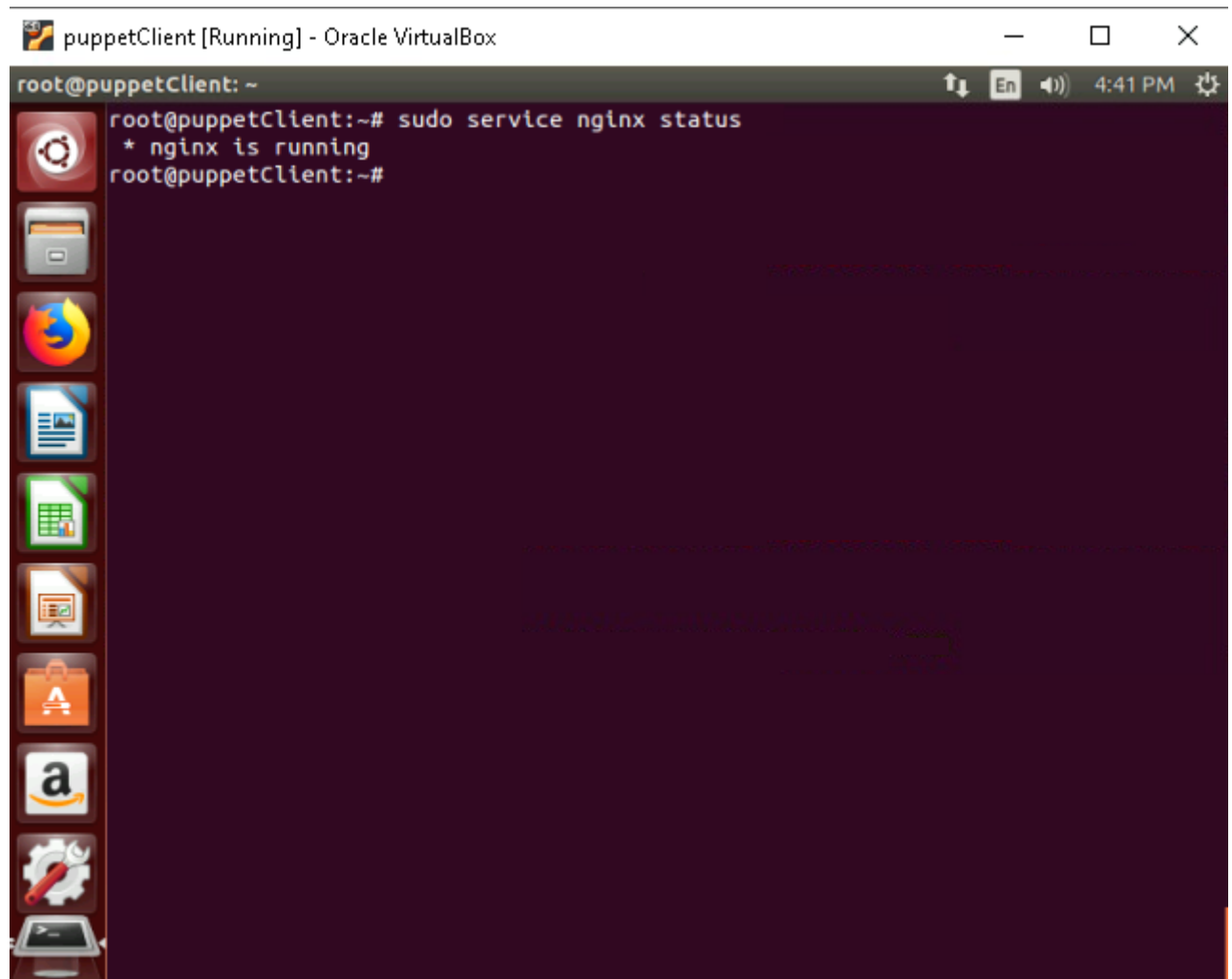
## Step 5: Verify the Configuration

After applying the manifest, it's important to verify that the `nginx` service is installed and running correctly on the Puppet Agent.

### 1. Check the status of the `nginx` service:

```
sudo service nginx status
```

- *Explanation:* This command checks whether the `nginx` service is running on the Agent node. The service should be active and running.



```
puppetClient [Running] - Oracle VirtualBox
root@puppetClient: ~
root@puppetClient:~# sudo service nginx status
* nginx is running
root@puppetClient:~#
```

---

## References

- [Manifests Documentation](#)
-