Experiment 14 - Puppet LAMP

Roll No.	24
Name	Iyer Sreekesh Subramanian
Class	D15-A
Subject	DevOps Lab
LO Mapped	LO1: To understand the fundamentals of DevOps engineering and be fully proficient with DevOps terminologies, concepts, benefits, and deployment options to meet your business requirements LO6: To Synthesize software configuration and provisioning using Ansible/Puppet.

Aim: To provision a LAMP/MEAN Stack using Puppet Manifest.

Introduction:
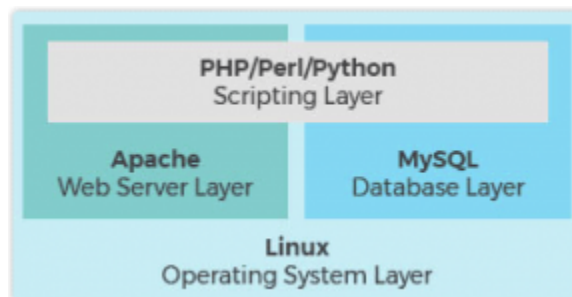
What is the LAMP Stack?

The widely popular LAMP stack is a set of open-source software used for web application development.

For a web application to work smoothly, it has to include an operating system, a web server, a database, and a programming language. The name LAMP is an acronym of the following programs:

➔ Linux Operating System
➔ Apache HTTP Server
➔ MySQL database management system
➔ PHP programming language

Each represents an essential layer of the stack, and together they can create a database-driven and dynamic website.

The illustration below can help visualize how the layers stack together:

Linux
Linux is the operating system layer and the backbone of the LAMP stack.

All the other components of the stack run on top of this foundation. You can efficiently manage the rest of the stack components on different operating systems such as Windows, macOS, and others. However, Linux has become the front-runner for web development not just because it is open-source, but also due to its flexibility, customization and easy to use technology.

Also, the programming language and database management used in developing a website may dictate the platform you choose to build it on. PHP and MySQL are better suited for Linux. On the other hand, SQL, ASP.NET, and ASP work more efficiently on Windows.

Apache

Apache HTTP Server is a web server software that runs on top of the Linux operating system.

It is the most widely used server, powering more than half of the websites on the internet. The role of the webserver is to process requests and transmit information through the internet, using HTTP.

An alternative to Apache is NGINX, a web server whose popularity has been continually increasing since 2008. Whether you go for one or the other depends on what kind of material you want to serve on a webpage, as well as the hosting.

NGINX is a better choice for static content. When it comes to dynamic content, there is a minor difference in performance between the two. Also, Apache is commonly used by shared hosting users, whereas NGINX is mainly used for virtual private servers, dedicated hosting or cluster containers.

MySQL

Note: SQL (Structured Query Language) is the most prevalent query language out there. A query is what we call a request for information or data stored in your database table.

MySQL earned its reputation as an acclaimed database system as it supports SQL and relational tables. By doing so, it makes it much easier to establish dynamic enterprise-level databases.

PHP

PHP (Hypertext Preprocessor) is a programming language that has the role of combining all the elements of the LAMP stack and allows the website or web application to run efficiently. In short, when a visitor opens the webpage, the server processes the PHP commands and sends the results to the visitor's browser.

PHP is the fourth layer of the original stack because it interacts exceptionally well with MySQL. It is commonly used for web development because it is a dynamically typed language, making it fast and easy to work with. This feature may be especially appealing if you are a beginner. The reason why PHP is so convenient to use is that it can be embedded into HTML enabling to jump in and out of it as you wish.

In the LAMP stack, the P can also refer to two other programming languages – Perl or Python. All three are simple, yet useful, dynamic tools for creating environments in which you can successfully develop applications. Nowadays, there is a wide variety of scripting languages to choose from, including JavaScript, Ruby, and many more.
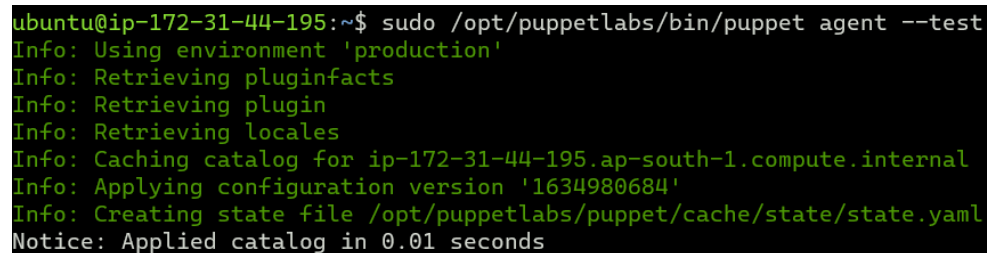
Demo

You'd need your Puppet cluster running to perform these commands.

Steps:

1. To test your cluster setup, run this command -

    ```
    sudo /opt/puppetlabs/bin/puppet agent --test
    ```
    ```
    ubuntu@ip-172-31-44-195:~$ sudo /opt/puppetlabs/bin/puppet agent --test
    Info: Using environment 'production'
    Info: Retrieving pluginfacts
    Info: Retrieving plugin
    Info: Retrieving locales
    Info: Caching catalog for ip-172-31-44-195.ap-south-1.compute.internal
    Info: Applying configuration version '1634980684'
    Info: Creating state file /opt/puppetlabs/puppet/cache/state/state.yaml
    Notice: Applied catalog in 0.01 seconds
    ```
    If the output is normal, you can proceed.

2. Change directories to the production folder
    ```
    cd /etc/puppetlabs/code/environments/production/manifests
    ```

3. Use nano to create a new lamp.pp file.
    ```
    sudo nano lamp.pp
    ```

4. Add this code to the file which will install all necessary dependencies and provision the stack. You can obtain this code from here.

```
  GNU nano 4.8
# execute 'apt-get update'
exec { 'apt-update':                          # exec resource named 'apt-update'
  command => '/usr/bin/apt-get update'  # command this resource will run
}

# install apache2 package
package { 'apache2':
  require => Exec['apt-update'],        # require 'apt-update' before installing
  ensure => installed,
}

# ensure apache2 service is running
service { 'apache2':
  ensure => running,
}

# install mysql-server package
package { 'mysql-server':
  require => Exec['apt-update'],        # require 'apt-update' before installing
  ensure => installed,
}

# ensure mysql service is running
service { 'mysql':
  ensure => running,
}

# install php package
package { 'php':
  require => Exec['apt-update'],        # require 'apt-update' before installing
  ensure => installed,
}

# ensure info.php file exists
file { '/var/www/html/info.php':
  ensure => file,
  content => '<?php  phpinfo(); ?>',   # phpinfo code
  require => Package['apache2'],       # require 'apache2' package before creating
}
```

5. Change directory to the bin folder of puppetlabs where the puppet executable is present.
    ```
    cd /opt/puppetlabs/bin
    ```
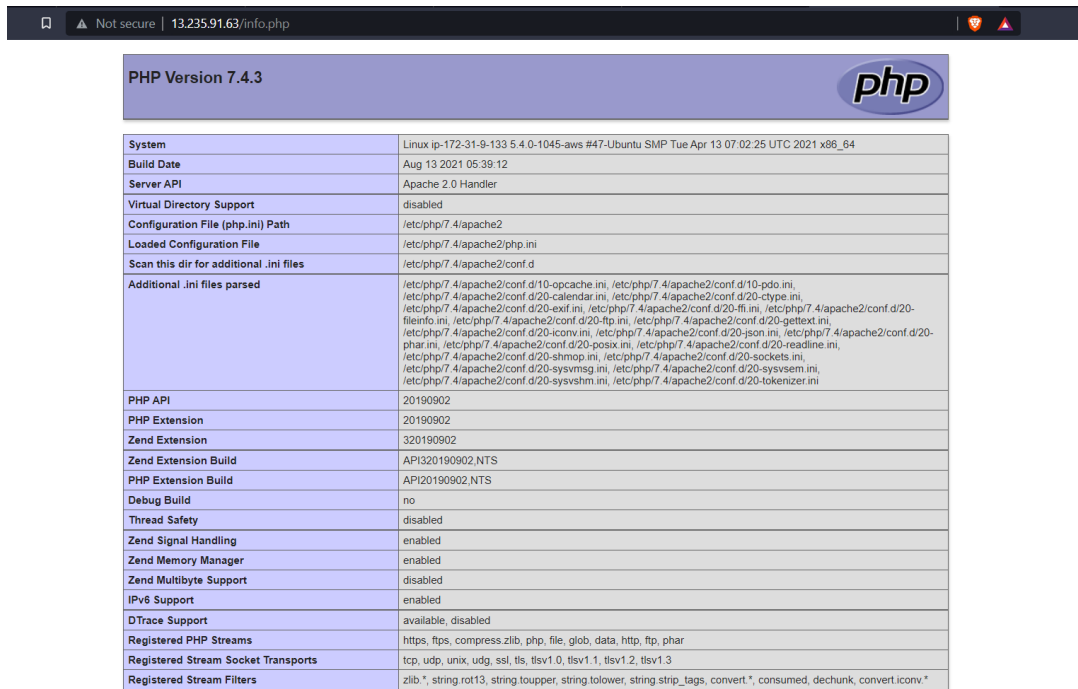
6. Use puppet apply to apply the scripts.
    ```
    ./puppet apply
    /etc/puppetlabs/code/environments/production/manifests/lamp.pp
    ```

```
root@ip-172-31-9-133:/opt/puppetlabs/bin# ./puppet apply /etc/puppetlabs/code/environments/production/manifests/lamp.pp
Notice: Compiled catalog for ip-172-31-9-133.ap-south-1.compute.internal in environment production in 0.75 seconds
Notice: /Stage[main]/Main/Exec[apt-update]/returns: executed successfully
```

7. Once done, go back to the EC2 Console, copy the public IP address of the client machine and put it in the browser. The URL is - http://ip_address_of_your_client/info.php

This verifies the provision of a LAMP stack using puppet.

Conclusion

Thus, we learned what a LAMP stack is and learned how to provision it using puppet scripts.