

Experiment No. 12

Aim: To provision a LAMP/MEAN Stack using puppet manifest.

LO: LO 6: Synthesis software configuration and provisioning using Ansible.

Theory:

What is LAMP Stack?

- 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 for the following programs:
 - Linux Operating System
 - Apache HTTP Web Server
 - MySQL/MariaDB database management system
 - PHP programming language

Linux Operating System

- 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 is more popular for web development not just because it is open-source, but also due to its flexibility, customization, and easy-to-use technology.

Apache Web Server

- 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 web server is to process requests and transmit information through the internet, using HTTP.

PHP(Programming Language)

- PHP(Hypertext Preprocessor) is a programming language that has the role of combining all the elements of the LAMP stack and allowing the website or web application to run efficiently.
- 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 you to jump in and out of it as you wish.

MYSQL/MariaDB

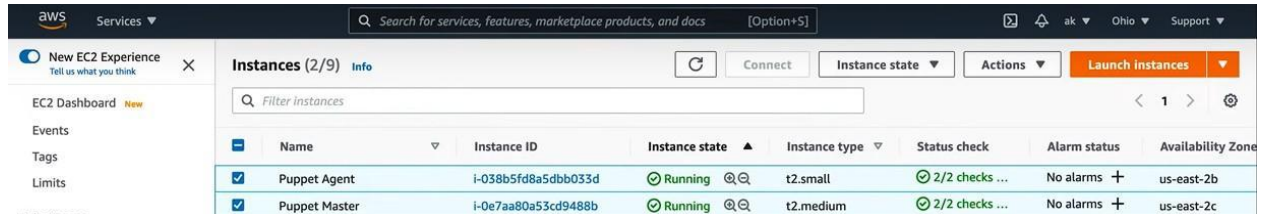
- MYSQL earned its reputation as an acclaimed database system as it supports SQL and relational tables. By doing so, it makes it much easier to establish dynamic enterprise-level databases.
- Another relational database management system that can be part of the LAMP platform is MariaDB. Both are quite similar, and MariaDB claims to be completely compatible with MySQL, allowing users to transfer their database without any complications or losses.

Why LAMP?

- The LAMP stack consists of four(4) components, all of which are examples of Free and Open-Source Software(Foss). As they are free and available for download, it attracts the attention of many users who wish to avoid paying large sums of money when developing their website.
- Because it is FOSS, the source code of the software is shared and available for people to make changes and improvements, enhancing its overall performance.
- The LAMP stack has proven to be a secure and stable platform thanks to its vast community that contributes when any problems arise.
- What makes it so attractive is that you can easily customize the stack and interchange the components with other open-source software to suit your needs.

Implementation:

Step 1: Connection between puppet master and agent.



	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
<input checked="" type="checkbox"/>	Puppet Agent	i-038b5fd8a5ddb033d	Running	t2.small	2/2 checks ...	No alarms +	us-east-2b
<input checked="" type="checkbox"/>	Puppet Master	i-0e7aa80a53cd9488b	Running	t2.medium	2/2 checks ...	No alarms +	us-east-2c

Step 2: Change the directory to the puppet modules.

```
ubuntu@ip-172-31-41-186:/opt/puppetlabs/puppet/modules/aws-examples$ sudo nano /etc/puppetlabs/code/environments/production/manifests/site.pp
ubuntu@ip-172-31-41-186:/opt/puppetlabs/puppet/modules/aws-examples$
ubuntu@ip-172-31-41-186:/opt/puppetlabs/puppet/modules/aws-examples$ cd /opt/puppetlabs/puppet/modules
ubuntu@ip-172-31-41-186:/opt/puppetlabs/puppet/modules$ clear
```

Step 3: Create the lamp directory to install the LAMP stack.

```
ubuntu@ip-172-31-41-186:/opt/puppetlabs/puppet/modules$
ubuntu@ip-172-31-41-186:/opt/puppetlabs/puppet/modules$ sudo mkdir lamp
mkdir: cannot create directory 'lamp': File exists
ubuntu@ip-172-31-41-186:/opt/puppetlabs/puppet/modules$ cd lamp
ubuntu@ip-172-31-41-186:/opt/puppetlabs/puppet/modules/lamp$ pwd
/opt/puppetlabs/puppet/modules/lamp
ubuntu@ip-172-31-41-186:/opt/puppetlabs/puppet/modules/lamp$ cd ..
ubuntu@ip-172-31-41-186:/opt/puppetlabs/puppet/modules$ rm -rf lamp
rm: cannot remove 'lamp/manifests/init.pp': Permission denied
ubuntu@ip-172-31-41-186:/opt/puppetlabs/puppet/modules$
```

Step 4: open init.pp and the following code to install the LAMP stack.

```
ubuntu@ip-172-31-41-186:/opt/puppetlabs/puppet/modules$ sudo mkdir lamp
ubuntu@ip-172-31-41-186:/opt/puppetlabs/puppet/modules$ cd lamp
ubuntu@ip-172-31-41-186:/opt/puppetlabs/puppet/modules/lamp$ sudo mkdir manifests
ubuntu@ip-172-31-41-186:/opt/puppetlabs/puppet/modules/lamp$ cd manifests/
ubuntu@ip-172-31-41-186:/opt/puppetlabs/puppet/modules/lamp/manifests$
ubuntu@ip-172-31-41-186:/opt/puppetlabs/puppet/modules/lamp/manifests$ sudo vi init.p
```

```
class lamp
# execute 'apt-get update'
exec { 'apt-update':
  command => '/usr/bin/apt-get update' # exec resource named 'apt-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 php7 package
package { 'php7.2-cli':
  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
}
```

Step 5: Make the configuration in the following file, get to know about the agent's IP.

```
ubuntu@ip-172-31-41-186: /opt/puppetlabs/puppet/modules/lamp/manifests$
ubuntu@ip-172-31-41-186: /opt/puppetlabs/puppet/modules/lamp/manifests$ sudo vi /etc/puppetlabs/code
/environments/production/manifests/site.pp
```

```
ubuntu@ip-172-31-41-186: /opt/puppetlabs/puppet/modules/lamp/manifests$
file {'/tmp/puppet_test.txt':
  ensure => present, # resource type file and filename
  mode => '0644', # make sure it exists
  content => "Howdy from Puppet master to agent on $(ipaddress_eth0)\n", # Print the eth0 IP fact
}

node 'ip-172-31-18-220.us-east-2.compute.internal' {
  include lamp
}
```

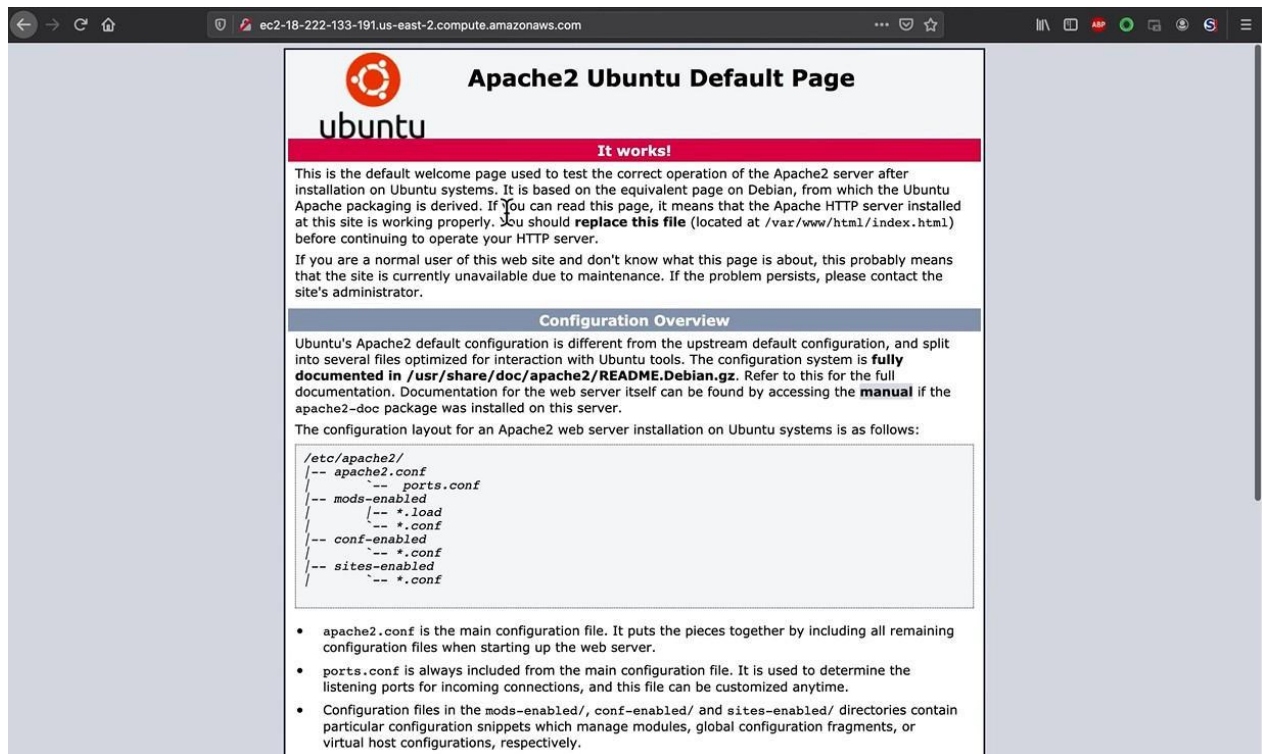
```
Hello from Puppet master to agent on 172.31.18.220!
ubuntu@ip-172-31-18-220:~$
ubuntu@ip-172-31-18-220:~$
ubuntu@ip-172-31-18-220:~$ sudo /opt/puppetlabs/bin/puppet agent --test
Info: Using configured environment 'production'
Info: Retrieving pluginfacts
Info: Retrieving plugin
Info: Retrieving locales
Info: Caching catalog for ip-172-31-18-220.us-east-2.compute.internal
Info: Applying configuration version '1613077602'
Notice: /Stage[main]/Main/File[/tmp/puppet_test.txt]/content:
--- /tmp/puppet_test.txt      2021-02-11 21:05:30.312821525 +0000
+++ /tmp/puppet-file20210211-2568-vqwalc 2021-02-11 21:06:42.573396524 +0000
@@ -1,1 @@
+Howdy from Puppet master to agent on 172.31.18.220!
+Howdy from Puppet master to agent on 172.31.18.220!

Info: Computing checksum on file /tmp/puppet_test.txt
Info: /Stage[main]/Main/File[/tmp/puppet_test.txt]: Filebucketed /tmp/puppet_test.txt to puppet wi
th sum ced49ea945662ec026fc3bb9f27fbb7a
Notice: /Stage[main]/Main/File[/tmp/puppet_test.txt]/content: content changed '[md5]ced49ea945662e
c026fc3bb9f27fbb7a' to '[md5]7cfbe05b9bd1707f6dbd4e4586422745'
Notice: Applied catalog in 0.07 seconds
ubuntu@ip-172-31-18-220:~$ sudo cat /tmp/puppet_test.txt
Howdy from Puppet master to agent on 172.31.18.220!
ubuntu@ip-172-31-18-220:~$ sudo /opt/puppetlabs/bin/puppet agent --test^C
ubuntu@ip-172-31-18-220:~$
ubuntu@ip-172-31-18-220:~$
ubuntu@ip-172-31-18-220:~$ clear
ubuntu@ip-172-31-18-220:~$
ubuntu@ip-172-31-18-220:~$
```

Step 6: Installing the necessary tools required

```
ubuntu@ip-172-31-18-220:~$ sudo /opt/puppetlabs/bin/puppet agent --test
Info: Using configured environment 'production'
Info: Retrieving pluginfacts
Info: Retrieving plugin
Info: Retrieving locales
Info: Caching catalog for ip-172-31-18-220.us-east-2.compute.internal
Info: Applying configuration version '1613078964'
Notice: /Stage[main]/Lamp/Exec[apt-update]/returns: executed successfully
█
```

```
ubuntu@ip-172-31-18-220:~$
ubuntu@ip-172-31-18-220:~$
ubuntu@ip-172-31-18-220:~$ mysql --version
mysql Ver 14.14 Distrib 5.7.33, for Linux (x86_64) using EditLine wrapper
ubuntu@ip-172-31-18-220:~$ php --version
PHP 7.2.24-0ubuntu0.18.04.7 (cli) (built: Oct 7 2020 15:24:25) ( NTS )
Copyright (c) 1997-2018 The PHP Group
Zend Engine v3.2.0, Copyright (c) 1998-2018 Zend Technologies
    with Zend OPcache v7.2.24-0ubuntu0.18.04.7, Copyright (c) 1999-2018, by Zend Technologies
ubuntu@ip-172-31-18-220:~$ █
```



The screenshot shows a web browser window displaying the 'Apache2 Ubuntu Default Page'. The page has a header with the Ubuntu logo and the title 'Apache2 Ubuntu Default Page'. Below the header, there is a red banner that says 'It works!'. The main content area contains a welcome message, a 'Configuration Overview' section, and a list of configuration files. The browser's address bar shows the URL 'ec2-18-222-133-191.us-east-2.compute.amazonaws.com'.

Apache2 Ubuntu Default Page

ubuntu

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|-- mods-enabled
|   |-- *.load
|   |-- *.conf
|-- conf-enabled
|   |-- *.conf
|-- sites-enabled
|   |-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.

Conclusion:

From the above experiment, I conclude that LAMP is nothing but Linux, Apache, MySQL, and PHP and the LAMP stack has proven to be a secure and stable platform. A manifest is a file containing Puppet configuration language that describes how resources should be configured. And hence, with this experiment we have achieved the Lab Outcome Six (LO6).

POs Achieved: PO1, PO5, PO12.