

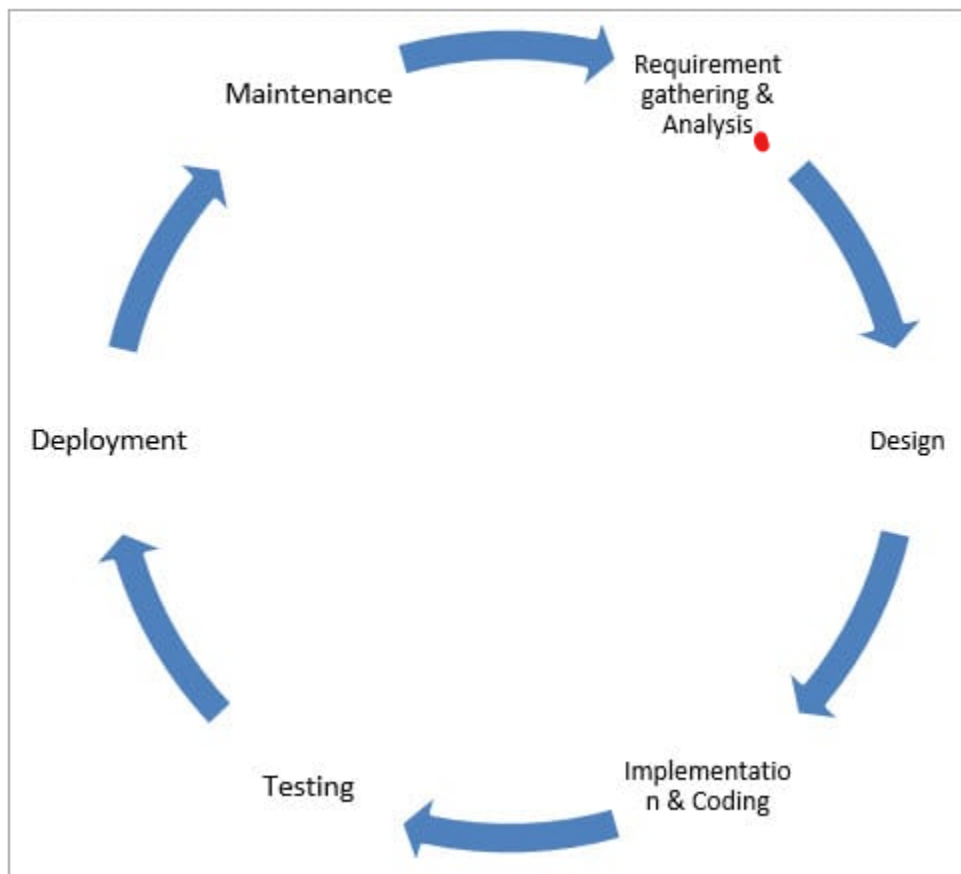
## Software Development Life Cycle (SDLC)

Software Development Life Cycle (SDLC) is a framework that defines the steps involved in the development of software at each phase. It covers the detailed plan for building, deploying and maintaining the software.

SDLC defines the complete cycle of development i.e. all the tasks involved in planning, creating, testing, and deploying a Software Product.

SDLC Cycle represents the process of developing software.

**Below is the diagrammatic representation of the SDLC cycle:**



### #1) Requirement Gathering and Analysis

During this phase, all the relevant information is collected from the customer to develop a product as per their expectation. Any ambiguities must be resolved in this phase only.

Business analyst and Project Manager set up a meeting with the customer to gather all the information like what the customer wants to build, who will be the end-user, what is the purpose of the product. Before building a product a core understanding or knowledge of the product is very important.

**For Example**, A customer wants to have an application which involves money transactions. In this case, the requirement has to be clear like what kind of transactions will be done, how it will be done, in which currency it will be done, etc.

Once the requirement gathering is done, an analysis is done to check the feasibility of the development of a product. In case of any ambiguity, a call is set up for further discussion.

Once the requirement is clearly understood, the SRS (Software Requirement Specification) document is created. This document should be thoroughly understood by the developers and also should be reviewed by the customer for future reference.

## #2) Design

In this phase, the requirement gathered in the SRS document is used as an input and software architecture that is used for implementing system development is derived.

## #3) Implementation or Coding

Implementation/Coding starts once the developer gets the Design document. The Software design is translated into source code. All the components of the software are implemented in this phase.

## #4) Testing

Testing starts once the coding is complete and the modules are released for testing. In this phase, the developed software is tested thoroughly and any defects found are assigned to developers to get them fixed.

Retesting, regression testing is done until the point at which the software is as per the customer's expectation. Testers refer SRS document to make sure that the software is as per the customer's standard.

## #5) Deployment

Once the product is tested, it is deployed in the production environment or first **UAT (User Acceptance testing)** is done depending on the customer expectation.

In the case of UAT, a replica of the production environment is created and the customer along with the developers does the testing. If the customer finds the application as expected, then sign off is provided by the customer to go live.

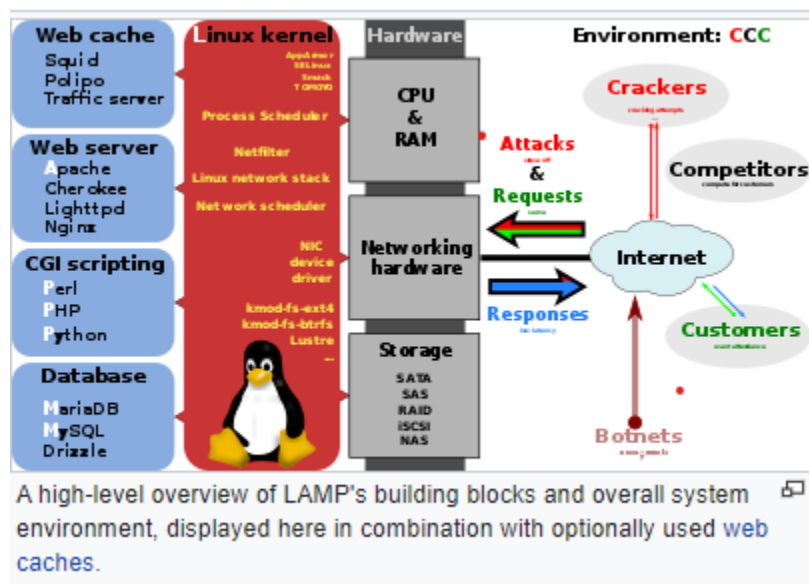
## #6) Maintenance

After the deployment of a product on the production environment, maintenance of the product i.e. if any issue comes up and needs to be fixed or any enhancement is to be done is taken care by the developers.

LAMP (software bundle)

**LAMP** (Linux, Apache, MySQL, PHP/Perl/Python) is a very common example of a [web service stack](#), named as an [acronym](#) of the names of its original four [open-source](#) components: the Linux operating system, the Apache HTTP Server, the MySQL relational database management system (RDBMS), and the PHP programming language. The LAMP components are largely interchangeable and not limited to the original selection. As a solution stack, LAMP is suitable for building [dynamic web sites](#) and [web applications](#).

Since its creation, the LAMP model has been adapted to other componentry, though typically consisting of [free and open-source software](#). For example, an equivalent installation on the [Microsoft Windows](#) family of operating systems is known as [WAMP](#) and an equivalent installation on [macOS](#) is known as [MAMP](#).



## Project Summary

1. Download, install and configure Virtual Box.  
Go to <https://www.virtualbox.org/wiki/Downloads> and select platform package (windows host in my case)  
Double click to install the downloaded file.  
Open the Virtual box application
2. Download, install and configure Ubuntu 20.04  
Search for Ubuntu download and select version 20.04  
On the virtual Box, go to tools and click on new to create a new virtual machine.  
Follow system prompts and set up memory (RAM) @2gb and virtual hard disk @ 10gb  
Load the ISO Image of Ubuntu 20.04 and complete the installation.

## Step 1 — Installing Apache and Updating the Firewall

After installing the Apache2, I did not get SSH but instead CUPS. The output for `$ sudo ufw app list` yielded the result below..

```
emumenwa@emserver:~$ sudo ufw app list
Available applications:
  Apache
  Apache Full
  Apache Secure
  CUPS
```

---

I ran the commands below to install and enable SSH on my Apache

```
sudo apt install openssh-server
sudo systemctl status ssh
sudo ufw allow ssh
```

```

enumenwa@emserver:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: ve
   Active: active (running) since Thu 2020-12-17 04:56:29 EST; 24h ago
     Docs: man:sshd(8)
           man:sshd_config(5)
    Main PID: 684 (sshd)
      Tasks: 1 (limit: 2319)
     Memory: 2.0M
    CGroup: /system.slice/ssh.service
            └─684 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups

Dec 17 04:56:28 emserver systemd[1]: Starting OpenBSD Secure Shell server...
Dec 17 04:56:29 emserver sshd[684]: Server listening on 0.0.0.0 port 22.
Dec 17 04:56:29 emserver sshd[684]: Server listening on :: port 22.
Dec 17 04:56:29 emserver systemd[1]: Started OpenBSD Secure Shell server.
lines 1-15/15 (END)

```

```

enumenwa@emserver:~$ sudo ufw allow ssh
Skipping adding existing rule
Skipping adding existing rule (v6)
enumenwa@emserver:~$ sudo ufw status
Status: active

To Action From
--
Apache ALLOW Anywhere
22/tcp ALLOW Anywhere
Apache (v6) ALLOW Anywhere (v6)
22/tcp (v6) ALLOW Anywhere (v6)

enumenwa@emserver:~$

```

```

$ hostname -I
10.0.2.15


```

I entered the IP address <http://10.0.2.15/> on my browser to get the result below

Activities Firefox Web Browser Dec 21 14:36

Apache2 Ubuntu Default Page

10.0.2.15



# 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 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.

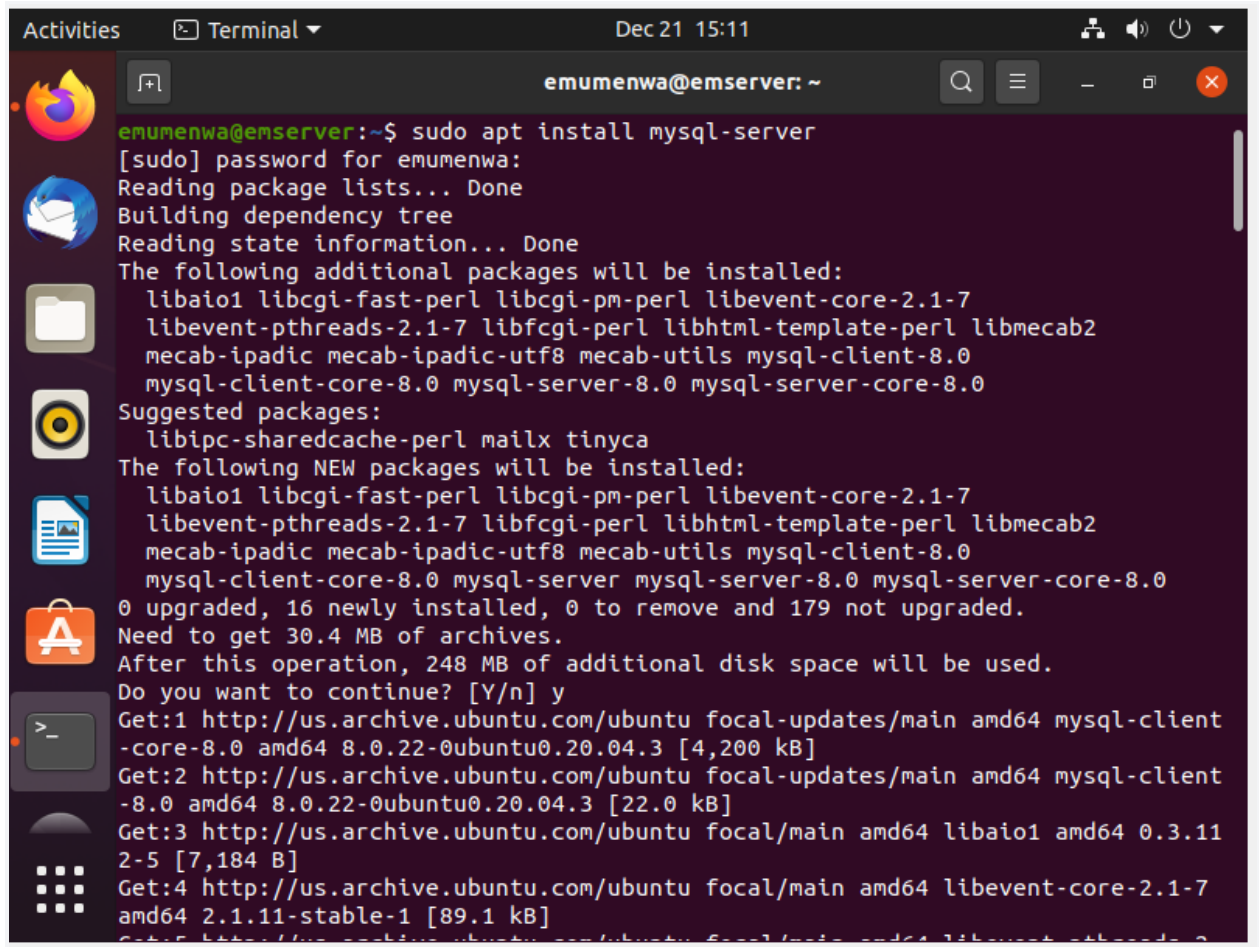
### Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and is 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:

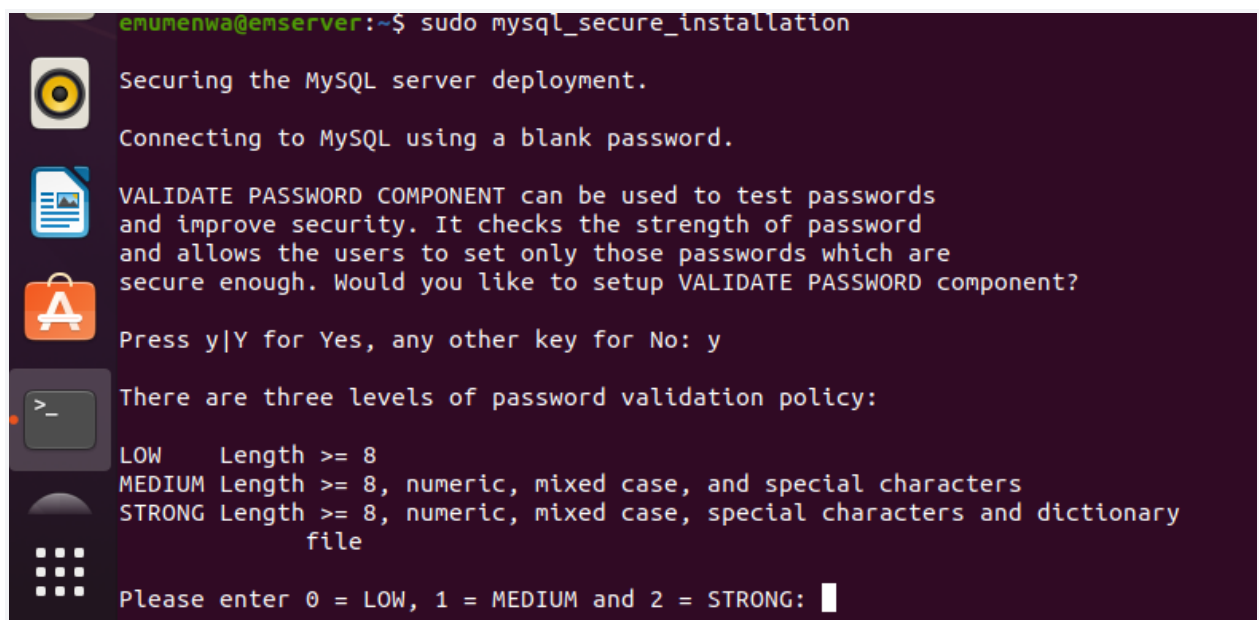
```
/etc/apache2/  
├── apache2.conf  
├── mods-enabled/  
├── mods-available/  
├── sites-enabled/  
└── sites-available/
```

## Step 2 — Installing MySQL



A terminal window titled "emumenwa@emserver: ~" showing the command to install MySQL server. The output lists additional packages to be installed, suggested packages, and new packages to be installed. It also shows the disk space requirements and the progress of downloading packages from the Ubuntu archive.

```
emumenwa@emserver:~$ sudo apt install mysql-server
[sudo] password for emumenwa:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libaio1 libcgi-fast-perl libcgi-pm-perl libevent-core-2.1-7
  libevent-pthreads-2.1-7 libfcgi-perl libhtml-template-perl libmecab2
  mecab-ipadic mecab-ipadic-utf8 mecab-utils mysql-client-8.0
  mysql-client-core-8.0 mysql-server-8.0 mysql-server-core-8.0
Suggested packages:
  libipc-sharedcache-perl mailx tinyca
The following NEW packages will be installed:
  libaio1 libcgi-fast-perl libcgi-pm-perl libevent-core-2.1-7
  libevent-pthreads-2.1-7 libfcgi-perl libhtml-template-perl libmecab2
  mecab-ipadic mecab-ipadic-utf8 mecab-utils mysql-client-8.0
  mysql-client-core-8.0 mysql-server mysql-server-8.0 mysql-server-core-8.0
0 upgraded, 16 newly installed, 0 to remove and 179 not upgraded.
Need to get 30.4 MB of archives.
After this operation, 248 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 mysql-client
-core-8.0 amd64 8.0.22-0ubuntu0.20.04.3 [4,200 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 mysql-client
-8.0 amd64 8.0.22-0ubuntu0.20.04.3 [22.0 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu focal/main amd64 libaio1 amd64 0.3.11
2-5 [7,184 B]
Get:4 http://us.archive.ubuntu.com/ubuntu focal/main amd64 libevent-core-2.1-7
amd64 2.1.11-stable-1 [89.1 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu focal/main amd64 libevent-pthreads-2.1-7
amd64 2.1.11-stable-1 [12.1 kB]
```



A terminal window showing the output of the 'mysql\_secure\_installation' command. It prompts the user to validate the password component and offers three levels of password validation policy: LOW, MEDIUM, and STRONG. The user is prompted to enter a number (0, 1, or 2) to select the policy.

```
emumenwa@emserver:~$ sudo mysql_secure_installation

Securing the MySQL server deployment.

Connecting to MySQL using a blank password.

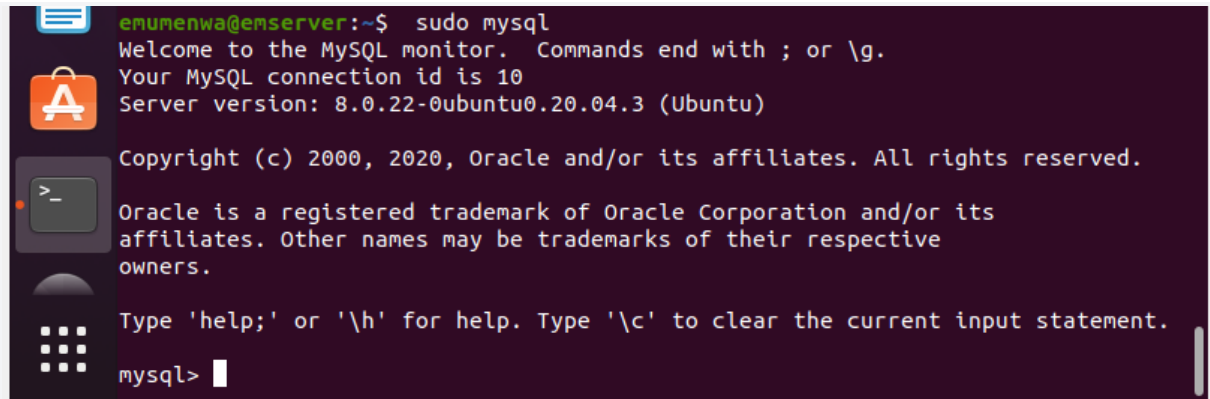
VALIDATE PASSWORD COMPONENT can be used to test passwords
and improve security. It checks the strength of password
and allows the users to set only those passwords which are
secure enough. Would you like to setup VALIDATE PASSWORD component?

Press y|Y for Yes, any other key for No: y

There are three levels of password validation policy:

LOW      Length >= 8
MEDIUM  Length >= 8, numeric, mixed case, and special characters
STRONG  Length >= 8, numeric, mixed case, special characters and dictionary
         file

Please enter 0 = LOW, 1 = MEDIUM and 2 = STRONG: 1
```

A terminal window with a dark purple background. On the left side, there is a vertical dock with several application icons: a blue document icon, an orange shopping bag icon, a terminal icon, and a grid of dots icon. The terminal text shows the command 'sudo mysql' being executed, followed by a welcome message for the MySQL monitor, connection details, and copyright information. The prompt 'mysql>' is visible at the bottom.

```
emumenwa@emserver:~$ sudo mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.22-0ubuntu0.20.04.3 (Ubuntu)

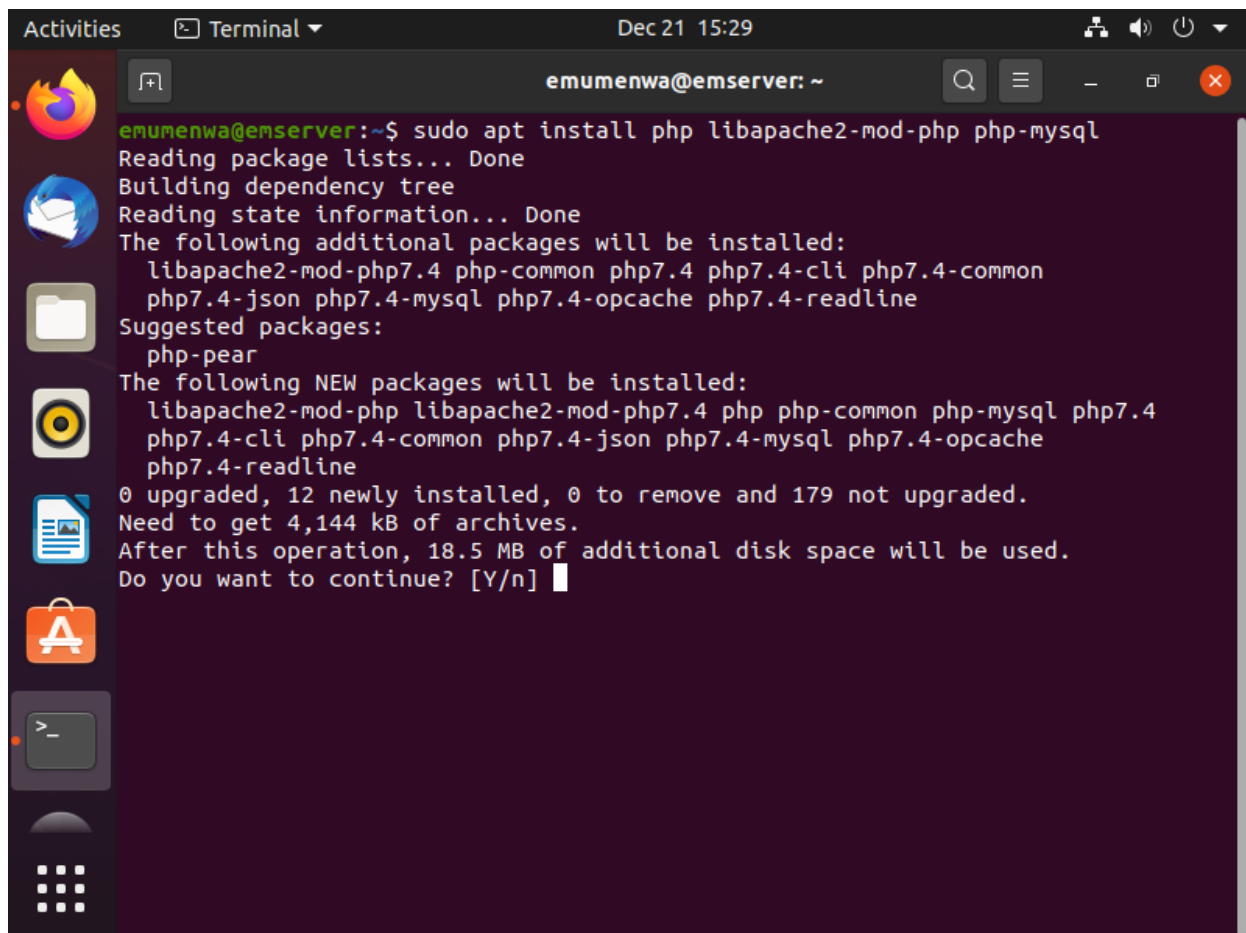
Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

### Step 3 — Installing PHP

A terminal window titled 'Terminal' with a dark purple background. The top bar shows 'Activities', 'Terminal', and the date 'Dec 21 15:29'. On the left, there is a dock with icons for Firefox, a mail client, a file manager, a media player, a document viewer, an application store, a terminal, and a grid of dots. The terminal text shows the command 'sudo apt install php libapache2-mod-php php-mysql' being executed. It displays the progress of package installation, including reading lists, building dependencies, and listing additional and new packages to be installed. It also shows disk space requirements and asks for confirmation to continue.

```
emumenwa@emserver: ~
emumenwa@emserver:~$ sudo apt install php libapache2-mod-php php-mysql
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libapache2-mod-php7.4 php-common php7.4 php7.4-cli php7.4-common
  php7.4-json php7.4-mysql php7.4-opcache php7.4-readline
Suggested packages:
  php-pear
The following NEW packages will be installed:
  libapache2-mod-php libapache2-mod-php7.4 php php-common php-mysql php7.4
  php7.4-cli php7.4-common php7.4-json php7.4-mysql php7.4-opcache
  php7.4-readline
0 upgraded, 12 newly installed, 0 to remove and 179 not upgraded.
Need to get 4,144 kB of archives.
After this operation, 18.5 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```



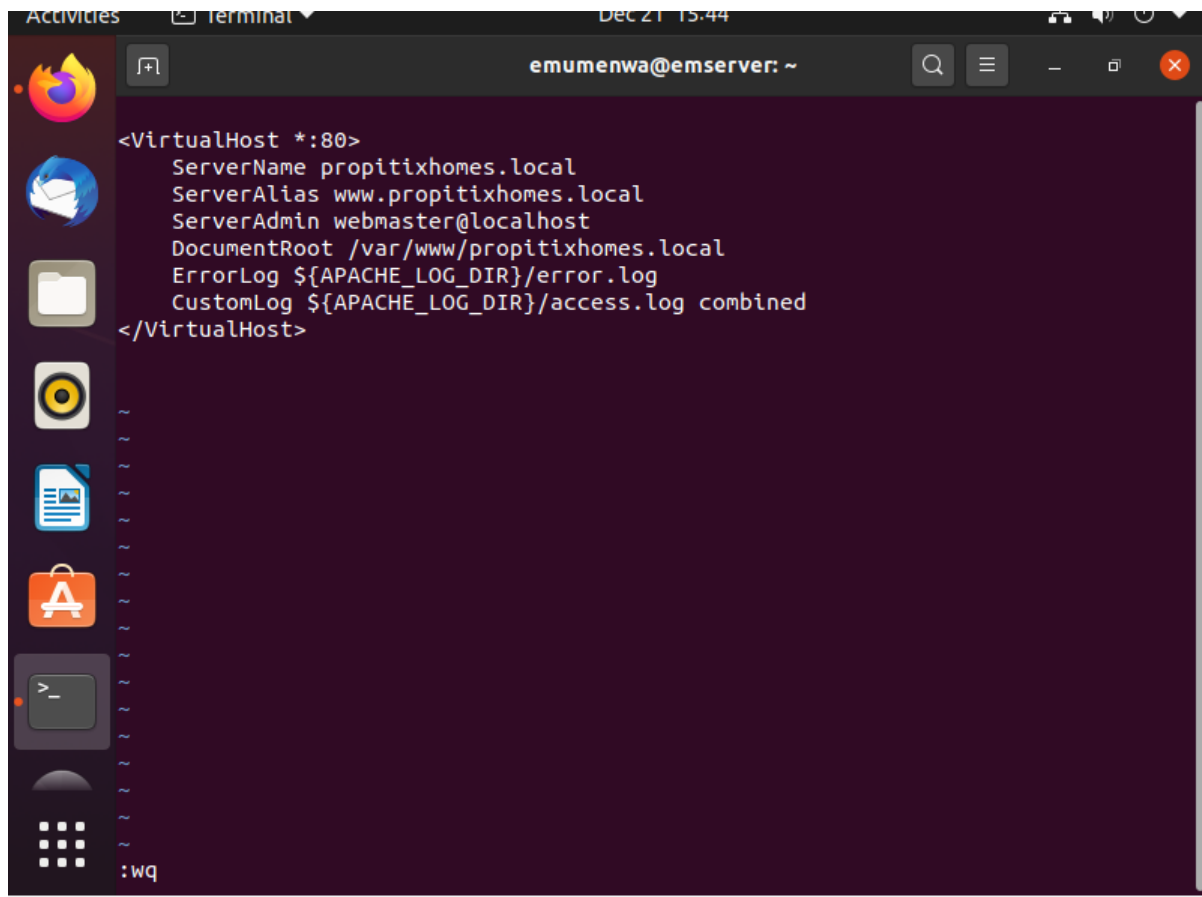
A terminal window titled "Terminal" with a date and time of "Dec 21 15:30". The window shows the command "php -v" being executed. The output displays the PHP version as 7.4.3 (cli), built on October 6, 2020, at 15:47:56, using the NTS (Non-Thread Safe) configuration. It also shows the Zend Engine version as v3.4.0, Copyright (c) Zend Technologies, and the Zend OPcache version as v7.4.3, Copyright (c), by Zend Technologies. The prompt "emumenwa@emserver: ~" is visible at the top of the terminal.

```
emumenwa@emserver:~$ php -v
PHP 7.4.3 (cli) (built: Oct  6 2020 15:47:56) ( NTS )
Copyright (c) The PHP Group
Zend Engine v3.4.0, Copyright (c) Zend Technologies
    with Zend OPcache v7.4.3, Copyright (c), by Zend Technologies
emumenwa@emserver:~$
```

#### Step 4 — Creating a Virtual Host for your Website using Apache

```
emumenwa@emserver:~$ sudo mkdir /var/www/propitixhomes.local
emumenwa@emserver:~$
```

```
emumenwa@emserver:~$ sudo chown -R $USER:$USER /var/www/propitixhomes.local
emumenwa@emserver:~$
```



A terminal window titled "emumenwa@emserver: ~" displays the configuration for a VirtualHost. The configuration is enclosed in <VirtualHost \*:80> and </VirtualHost> tags. Inside, the following settings are listed: ServerName propitixhomes.local, ServerAlias www.propitixhomes.local, ServerAdmin webmaster@localhost, DocumentRoot /var/www/propitixhomes.local, ErrorLog \${APACHE\_LOG\_DIR}/error.log, and CustomLog \${APACHE\_LOG\_DIR}/access.log combined. The terminal has a dark purple background and a sidebar on the left with various application icons.

```
<VirtualHost *:80>
    ServerName propitixhomes.local
    ServerAlias www.propitixhomes.local
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/propitixhomes.local
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

```
emumenwa@emserver:~$ sudo ls /etc/apache2/sites-available
000-default.conf  default-ssl.conf  propitixhomes.local.conf
emumenwa@emserver:~$
```

```
emumenwa@emserver:~$ sudo a2ensite propitixhomes.local
Enabling site propitixhomes.local.
To activate the new configuration, you need to run:
    systemctl reload apache2
emumenwa@emserver:~$
```

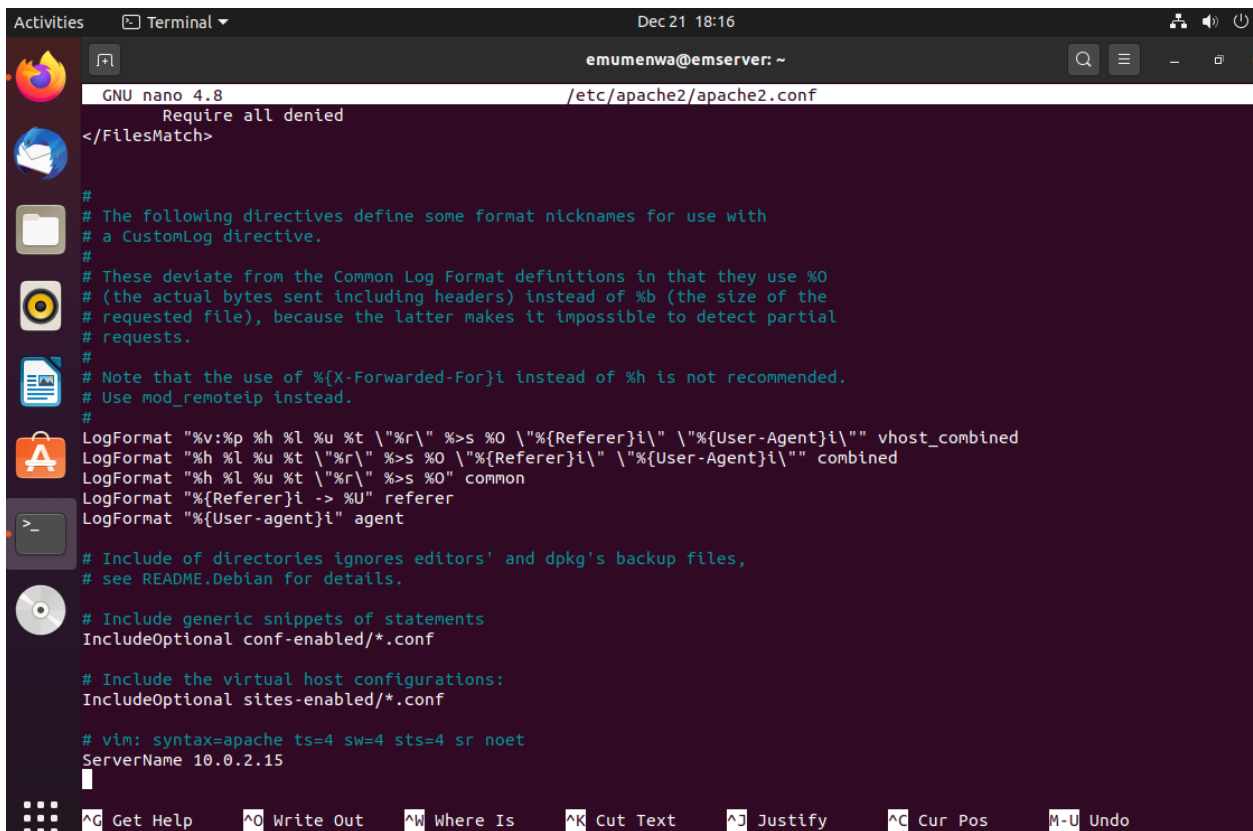
```
emumenwa@emserver:~$ sudo a2ensite propitixhomes.local
[sudo] password for emumenwa:
Site propitixhomes.local already enabled
emumenwa@emserver:~$
```

```
emumenwa@emserver:~$ sudo a2dissite 000-default
Site 000-default already disabled
emumenwa@emserver:~$
```

```
emumenwa@emserver:~$ sudo apache2ctl configtest
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set the 'ServerName' directive globally to suppress this message
Syntax OK
emumenwa@emserver:~$
```

To clear the message above, I ran the command shown next added a line containing the server name

```
emumenwa@emserver:~$ sudo nano /etc/apache2/apache2.conf
emumenwa@emserver:~$
```



```
emumenwa@emserver: ~
GNU nano 4.8 /etc/apache2/apache2.conf
Require all denied
</FilesMatch>

#
# The following directives define some format nicknames for use with
# a CustomLog directive.
#
# These deviate from the Common Log Format definitions in that they use %O
# (the actual bytes sent including headers) instead of %b (the size of the
# requested file), because the latter makes it impossible to detect partial
# requests.
#
# Note that the use of %{X-Forwarded-For}i instead of %h is not recommended.
# Use mod_remoteip instead.
#
LogFormat "%v:%p %h %l %u %t \"%r\" %>s %O \"%{Referer}i\" \"%{User-Agent}i\"" vhost_combined
LogFormat "%h %l %u %t \"%r\" %>s %O \"%{Referer}i\" \"%{User-Agent}i\"" combined
LogFormat "%h %l %u %t \"%r\" %>s %O" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent

# Include of directories ignores editors' and dpkg's backup files,
# see README.Debian for details.

# Include generic snippets of statements
IncludeOptional conf-enabled/*.conf

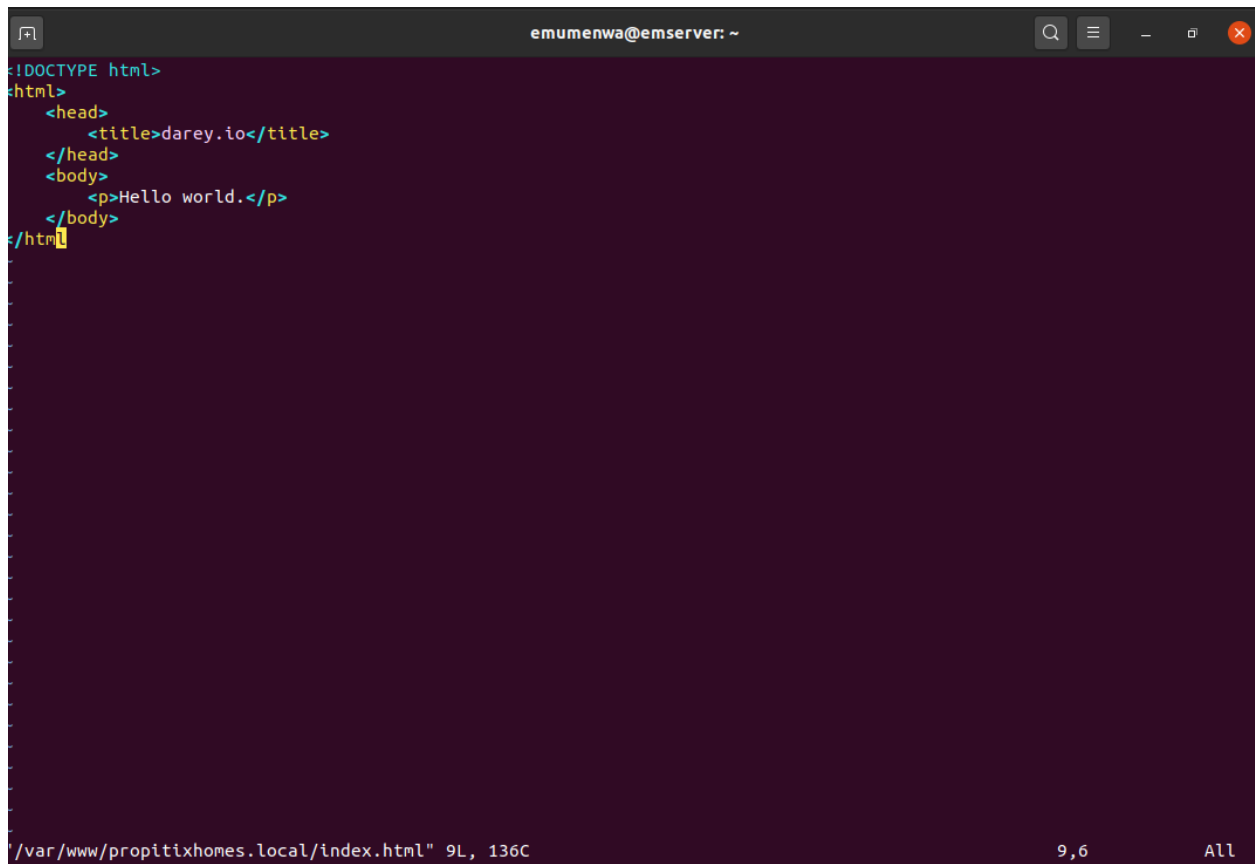
# Include the virtual host configurations:
IncludeOptional sites-enabled/*.conf

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
ServerName 10.0.2.15
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos M-U Undo
```

```
emumenwa@emserver:~$ sudo apache2ctl configtest
Syntax OK
emumenwa@emserver:~$
```

```
emumenwa@emserver:~$ sudo systemctl reload apache2
emumenwa@emserver:~$
```

emumenwa@emserver:~\$ vi /var/www/propitixhomes.local/index.html



```
emumenwa@emserver: ~
<!DOCTYPE html>
<html>
  <head>
    <title>darey.io</title>
  </head>
  <body>
    <p>Hello world.</p>
  </body>
</html>
```

"/var/www/propitixhomes.local/index.html" 9L, 136C 9,6 All



```
emumenwa@emserver:~$ sudo vim /etc/apache2/mods-enabled/dir.conf
[sudo] password for emumenwa:
emumenwa@emserver:~$
```

```
emumenwa@emserver: ~
<IfModule mod_dir.c>
    DirectoryIndex index.php index.html index.cgi index.pl index.xhtml index.htm
</IfModule>
# vim: syntax=apache ts=4 sw=4 sts=4 sr noet

"/etc/apache2/mods-enabled/dir.conf" 4L, 163C          3,11          All
```

```
emumenwa@emserver:~$ sudo systemctl reload apache2
emumenwa@emserver:~$
```

```
emumenwa@emserver:~$ vim /var/www/propitixhomes.local/info.php
```

[illegible]

<b>System</b>	Linux emserver 5.4.0-58-generic #64-Ubuntu SMP Wed Dec 9 08:16:25 UTC 2020 x86_64
<b>Build Date</b>	Oct 6 2020 15:47:56
<b>Server API</b>	Apache 2.0 Handler
<b>Virtual Directory Support</b>	disabled
<b>Configuration File (php.ini) Path</b>	/etc/php/7.4/apache2
<b>Loaded Configuration File</b>	/etc/php/7.4/apache2/php.ini
<b>Scan this dir for additional .ini files</b>	/etc/php/7.4/apache2/conf.d
<b>Additional .ini files parsed</b>	/etc/php/7.4/apache2/conf.d/10-mysqlnd.ini, /etc/php/7.4/apache2/conf.d/10-opcache.ini, /etc/php/7.4/apache2/conf.d/10-pdo.ini, /etc/php/7.4/apache2/conf.d/20-calendar.ini, /etc/php/7.4/apache2/conf.d/20-ctype.ini, /etc/php/7.4/apache2/conf.d/20-exif.ini, /etc/php/7.4/apache2/conf.d/20-ffi.ini, /etc/php/7.4/apache2/conf.d/20-fileinfo.ini, /etc/php/7.4/apache2/conf.d/20-ftp.ini, /etc/php/7.4/apache2/conf.d/20-gettext.ini, /etc/php/7.4/apache2/conf.d/20-iconv.ini, /etc/php/7.4/apache2/conf.d/20-json.ini, /etc/php/7.4/apache2/conf.d/20-mysqli.ini, /etc/php/7.4/apache2/conf.d/20-pdo-mysqli.ini, /etc/php/7.4/apache2/conf.d/20-phar.ini, /etc/php/7.4/apache2/conf.d/20-posix.ini, /etc/php/7.4/apache2/conf.d/20-readline.ini, /etc/php/7.4/apache2/conf.d/20-shmop.ini, /etc/php/7.4/apache2/conf.d/20-sockets.ini, /etc/php/7.4/apache2/conf.d/20-sysvmsg.ini, /etc/php/7.4/apache2/conf.d/20-sysvsem.ini, /etc/php/7.4/apache2/conf.d/20-sysvshm.ini, /etc/php/7.4/apache2/conf.d/20-tokenizer.ini
<b>PHP API</b>	20190902
<b>PHP Extension</b>	20190902
<b>Zend Extension</b>	320190902
<b>Zend Extension Build</b>	API320190902.NTS
<b>PHP Extension Build</b>	API20190902.NTS
<b>Debug Build</b>	no
<b>Thread Safety</b>	disabled
<b>Zend Signal Handling</b>	enabled

```
emumenwa@enserver:~$ sudo rm /var/www/propitixhomes.local/info.php  
[sudo] password for emumenwa:  
emumenwa@enserver:~$
```