# Building a LAMP Stack Web Server

## Introduction:

The LAMP Stack Web Server is also known as Linux Web Server. It is a open-source web development platform used for building dynamic websites and web applications. LAMP stands for Linux, Apache, MySQL, and PHP/Perl/Python. These are the four main components required to setup a Web Server.

## Components of the LAMP Stack:

### Linux (Operating System)

- Linux servers as the foundation of the LAMP stack. It Provides the Operating System on which the other components are installed and run.
- Linux distributions for LAMP Stack deployment include Ubuntu and Debian.

### Apache (Web Server Software)

- Apache is a widely used open-source web server software that serves web content to clients over the internet.
- Apache handles HTTP requests from web browsers and delivers web pages.
- It extends various modules for extending its functionality.

### MySQL (Database)

- It is a software which is used to maintain and manage the Database.
- It provides security and authorization are the important features of the DBMS.
- MySQL is commonly used in web applications to store user information and other data.

### PHP/Perl/Python (Server Side Scripting Language)

- PHP, Perl, Python are the server-side scripting languages used for dynamic web development.
- These languages enables developers to create interactive and dynamic web pages that respond to user input.
- PHP is mostly commonly used language in LAMP Stack.

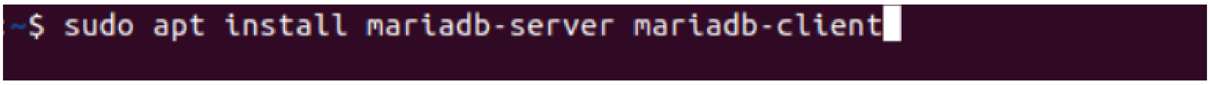
## Setting up a LAMP Stack Web Server

### Step1: Install Linux

- Before we are going to further you must need a Linux or One of Linux Distributions.
  Ubuntu by using this link https://ubuntu.com/download.

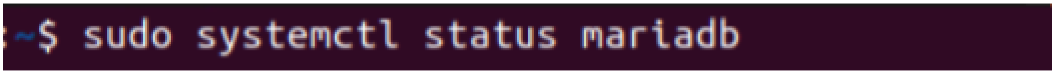
## Step 2: Installing MySQL/Mariadb

- Now, install MariaDB server and client packages from the official package repository of Debian 10 with the following command:



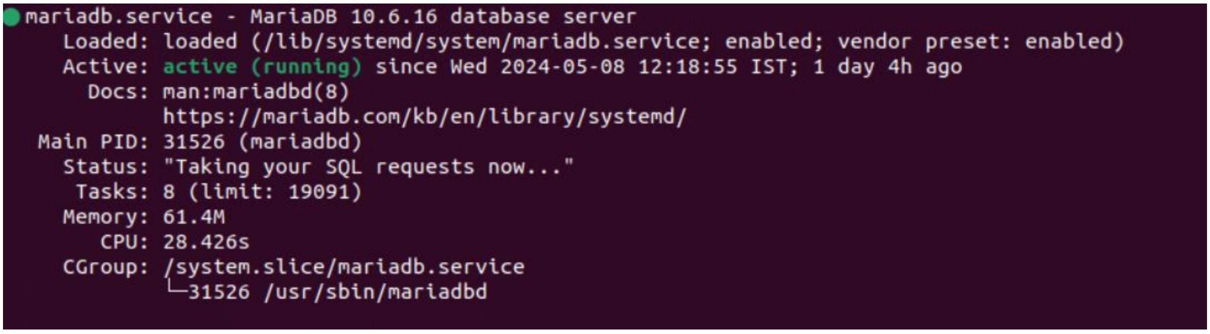
```
~$ sudo apt install mariadb-server mariadb-client
```

- Now conform your Installation, Press Y to confirm or n to stop the process.
- If you want to check the status of mariadb server is running or not Enter the following command



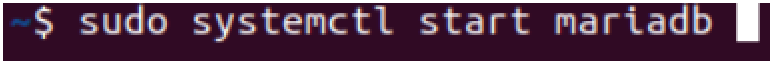
```
~$ sudo systemctl status mariadb
```

Make sure the Server is running like this



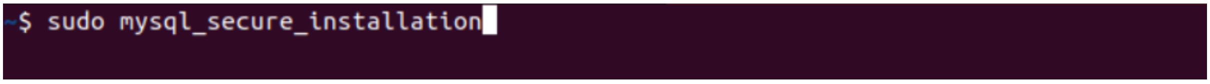
```
● mariadb.service - MariaDB 10.6.16 database server
     Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor preset: enabled)
     Active: active (running) since Wed 2024-05-08 12:18:55 IST; 1 day 4h ago
       Docs: man:mariadbd(8)
             https://mariadb.com/kb/en/library/systemd/
   Main PID: 31526 (mariadbd)
     Status: "Taking your SQL requests now..."
      Tasks: 8 (limit: 19091)
     Memory: 61.4M
        CPU: 28.426s
     CGroup: /system.slice/mariadb.service
             └─31526 /usr/sbin/mariadbd
```

- If server is not running then enter the following command



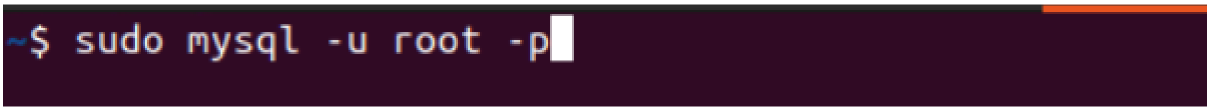
```
~$ sudo systemctl start mariadb
```

- Enter the following command to install mySQL and setup according to it



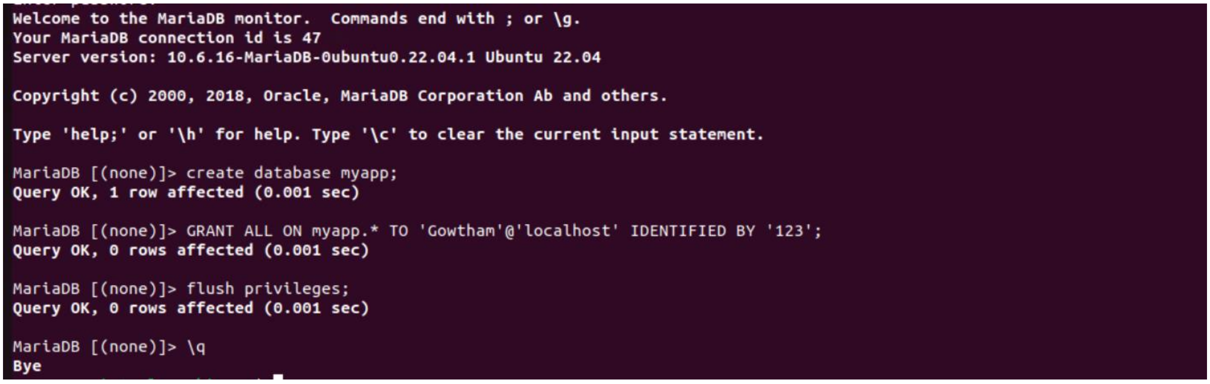
```
~$ sudo mysql_secure_installation
```

- After, Successfully installation of MYSQL, Enter the following command to enter Mariadb server



```
~$ sudo mysql -u root -p
```

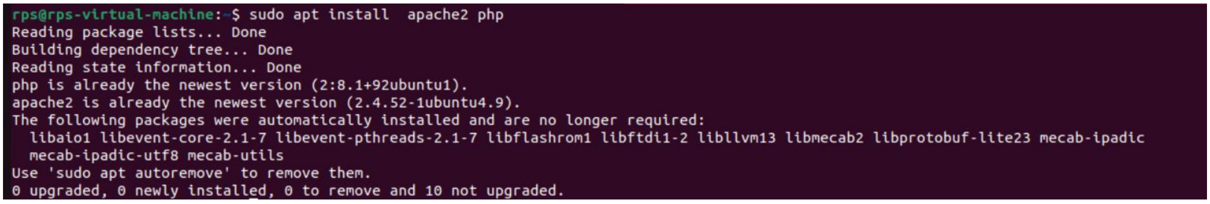
After Entering into the database we need to create our own database and give an access to a person by using Grant all Statement



```
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 47
Server version: 10.6.16-MariaDB-0ubuntu0.22.04.1 Ubuntu 22.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MariaDB [(none)]> create database myapp;
Query OK, 1 row affected (0.001 sec)

MariaDB [(none)]> GRANT ALL ON myapp.* TO 'Gowtham'@'localhost' IDENTIFIED BY '123';
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> flush privileges;
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> \q
Bye
```

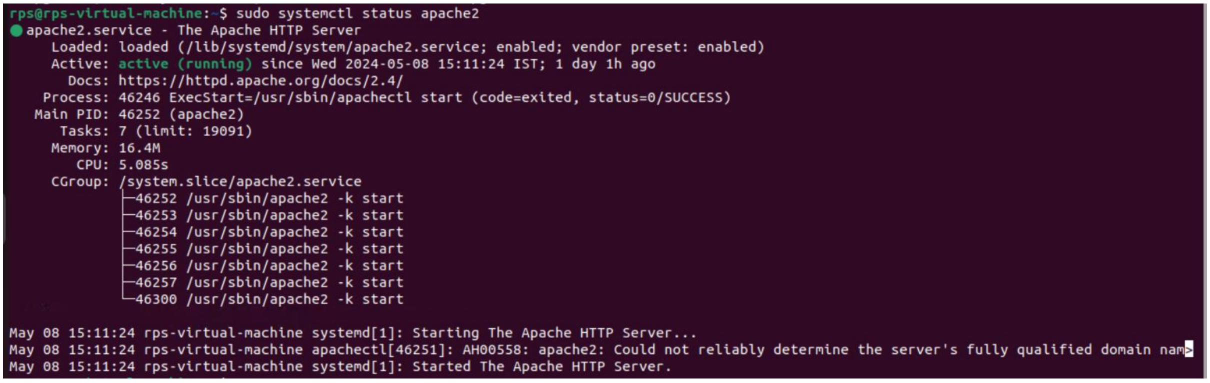
## Step-3:Install Apache2 and PHP

- To install Apache2 and PHP in your Operating System use the following command and setup according to it:



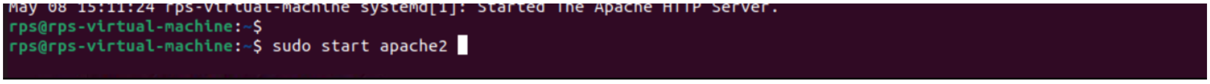
```
rps@rps-virtual-machine:~$ sudo apt install  apache2 php
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
php is already the newest version (2:8.1+92ubuntu1).
apache2 is already the newest version (2.4.52-1ubuntu4.9).
The following packages were automatically installed and are no longer required:
  libaio1 libevent-core-2.1-7 libevent-pthreads-2.1-7 libflashrom1 libftdi1-2 libllvm13 libmecab2 libprotobuf-lite23 mecab-ipadic
  mecab-ipadic-utf8 mecab-utils
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 10 not upgraded.
```

- After the installation done check the status of that server is running or not by using the following command



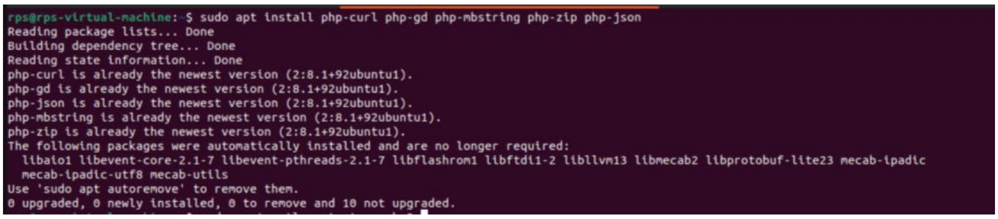
```
rps@rps-virtual-machine:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
     Active: active (running) since Wed 2024-05-08 15:11:24 IST; 1 day 1h ago
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 46246 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 46252 (apache2)
      Tasks: 7 (limit: 19091)
     Memory: 16.4M
        CPU: 5.085s
     CGroup: /system.slice/apache2.service
             ├─46252 /usr/sbin/apache2 -k start
             ├─46253 /usr/sbin/apache2 -k start
             ├─46254 /usr/sbin/apache2 -k start
             ├─46255 /usr/sbin/apache2 -k start
             ├─46256 /usr/sbin/apache2 -k start
             ├─46257 /usr/sbin/apache2 -k start
             └─46300 /usr/sbin/apache2 -k start

May 08 15:11:24 rps-virtual-machine systemd[1]: Starting The Apache HTTP Server...
May 08 15:11:24 rps-virtual-machine apachectl[46251]: AH00558: apache2: Could not reliably determine the server's fully qualified domain nam>
May 08 15:11:24 rps-virtual-machine systemd[1]: Started The Apache HTTP Server.
```

- If the server is not running run the following command



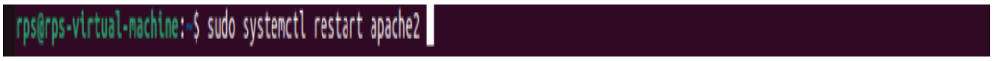
```
May 08 15:11:24 rps-virtual-machine systemd[1]: Started The Apache HTTP Server.
rps@rps-virtual-machine:~$
rps@rps-virtual-machine:~$ sudo start apache2
```

- To install the most common PHP extensions/libraries, run the following command:



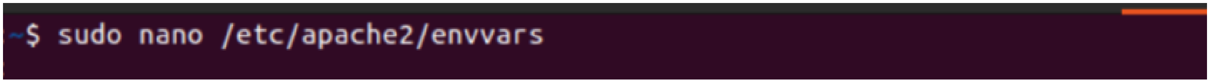
```
rps@rps-virtual-machine:~$ sudo apt install php-curl php-gd php-mbstring php-zip php-json
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
php-curl is already the newest version (2:8.1+92ubuntu1).
php-gd is already the newest version (2:8.1+92ubuntu1).
php-json is already the newest version (2:8.1+92ubuntu1).
php-mbstring is already the newest version (2:8.1+92ubuntu1).
php-zip is already the newest version (2:8.1+92ubuntu1).
The following packages were automatically installed and are no longer required:
  libaio1 libevent-core-2.1-7 libevent-pthreads-2.1-7 libflashrom1 libftdi1-2 libllvm13 libmecab2 libprotobuf-lite23 mecab-ipadic
  mecab-ipadic-utf8 mecab-utils
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 10 not upgraded.
```

- Then, Again Restart the Apache2 Server by using following command



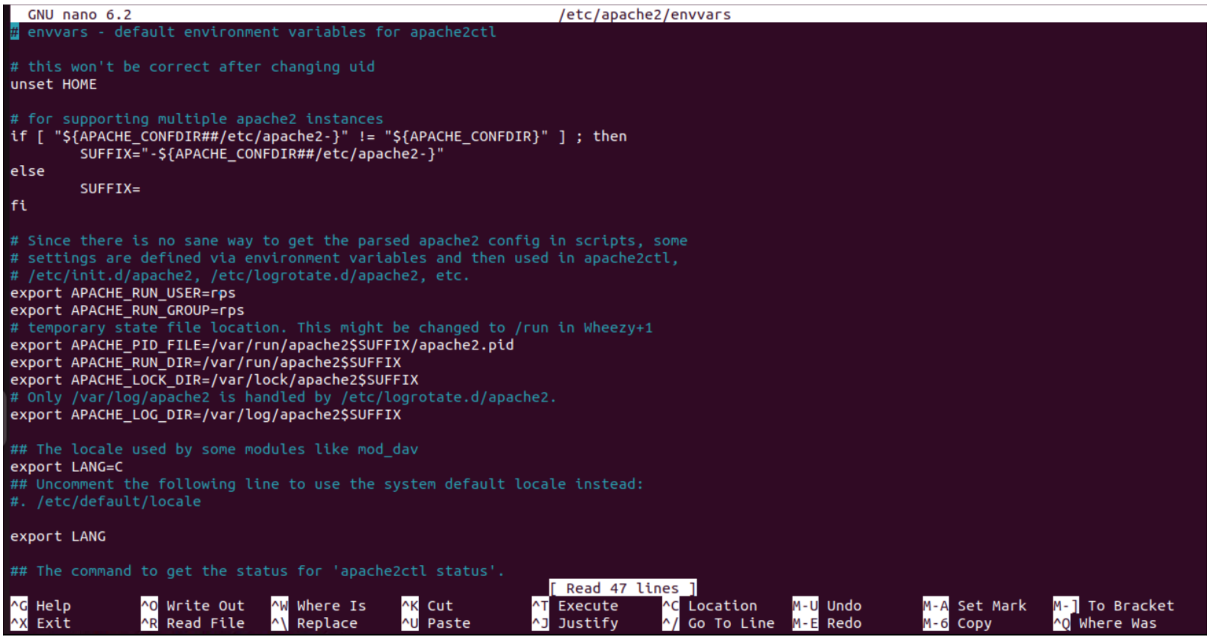
```
rps@rps-virtual-machine:~$ sudo systemctl restart apache2
```

- To change the Apache run user, edit **/etc/apache2/envvars** configuration file with the following command:



```
~$ sudo nano /etc/apache2/envvars
```

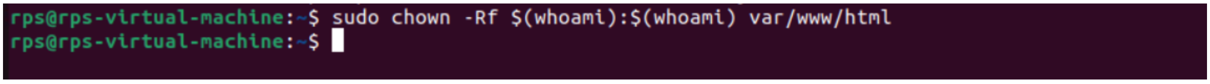
- Modify-
  the **APACHE_RUN_USER** and **APACHE_RUN_GROUP** environment variables.



```
# envvars - default environment variables for apache2ctl

# this won't be correct after changing uid
unset HOME

# for supporting multiple apache2 instances
if [ "${APACHE_CONFDIR##/etc/apache2-}" != "${APACHE_CONFDIR}" ] ; then
        SUFFIX="-${APACHE_CONFDIR##/etc/apache2-}"
else
        SUFFIX=
fi

# Since there is no sane way to get the parsed apache2 config in scripts, some
# settings are defined via environment variables and then used in apache2ctl,
# /etc/init.d/apache2, /etc/logrotate.d/apache2, etc.
export APACHE_RUN_USER=rps
export APACHE_RUN_GROUP=rps
# temporary state file location. This might be changed to /run in Wheezy+1
export APACHE_PID_FILE=/var/run/apache2$SUFFIX/apache2.pid
export APACHE_RUN_DIR=/var/run/apache2$SUFFIX
export APACHE_LOCK_DIR=/var/lock/apache2$SUFFIX
# Only /var/log/apache2 is handled by /etc/logrotate.d/apache2.
export APACHE_LOG_DIR=/var/log/apache2$SUFFIX

## The locale used by some modules like mod_dav
export LANG=C
## Uncomment the following line to use the system default locale instead:
#. /etc/default/locale

export LANG

## The command to get the status for 'apache2ctl status'.
                                  [ Read 47 lines ]
```

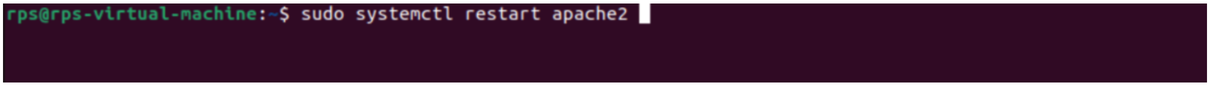
Now, Instead of rps you have to give your root user. In this case my user is rps.

- Now, change the owner and group of the **/var/www/html** directory to the username of your login user with the following command:



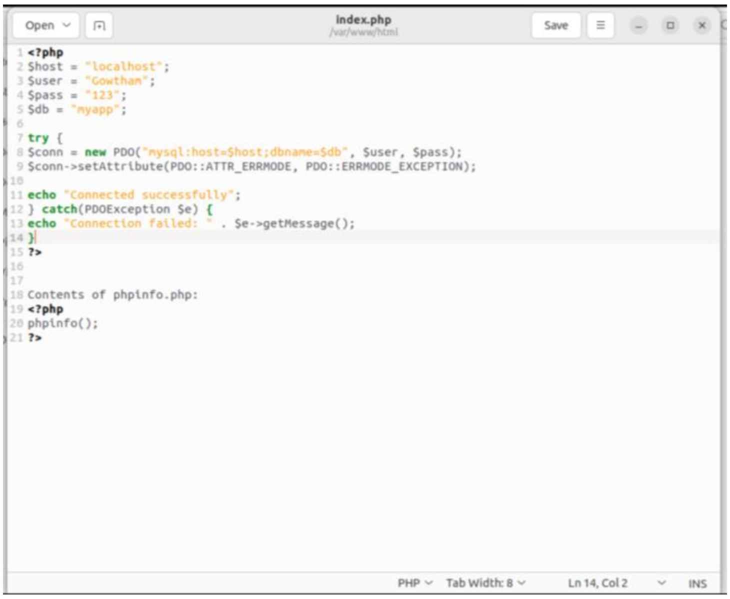
```
rps@rps-virtual-machine:~$ sudo chown -Rf $(whoami):$(whoami) var/www/html
rps@rps-virtual-machine:~$ 
```

- Then, Restart the Apache2 server again



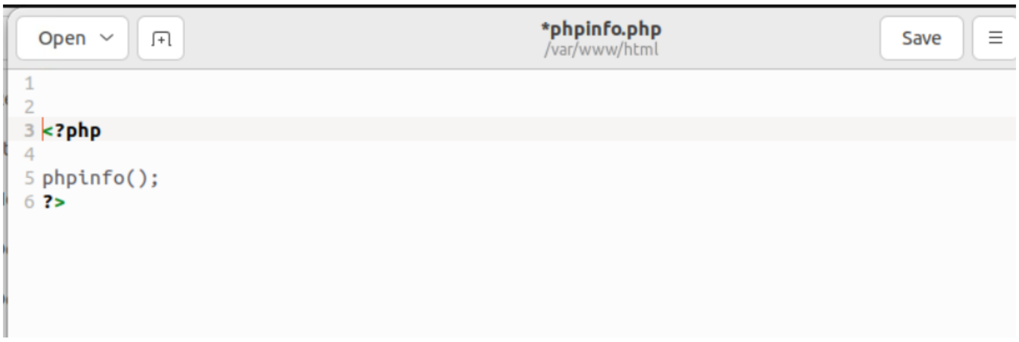
```
rps@rps-virtual-machine:~$ sudo systemctl restart apache2 
```

Now, create the 2 PHP pages index.php and phpinfo.php
index.php



```
<?php
$host = "localhost";
$user = "Gowtham";
$pass = "123";
$db = "myapp";

try {
$conn = new PDO("mysql:host=$host;dbname=$db", $user, $pass);
$conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

echo "Connected successfully";
} catch(PDOException $e) {
echo "Connection failed: " . $e->getMessage();
}
?>


Contents of phpinfo.php:
<?php
phpinfo();
?>
```

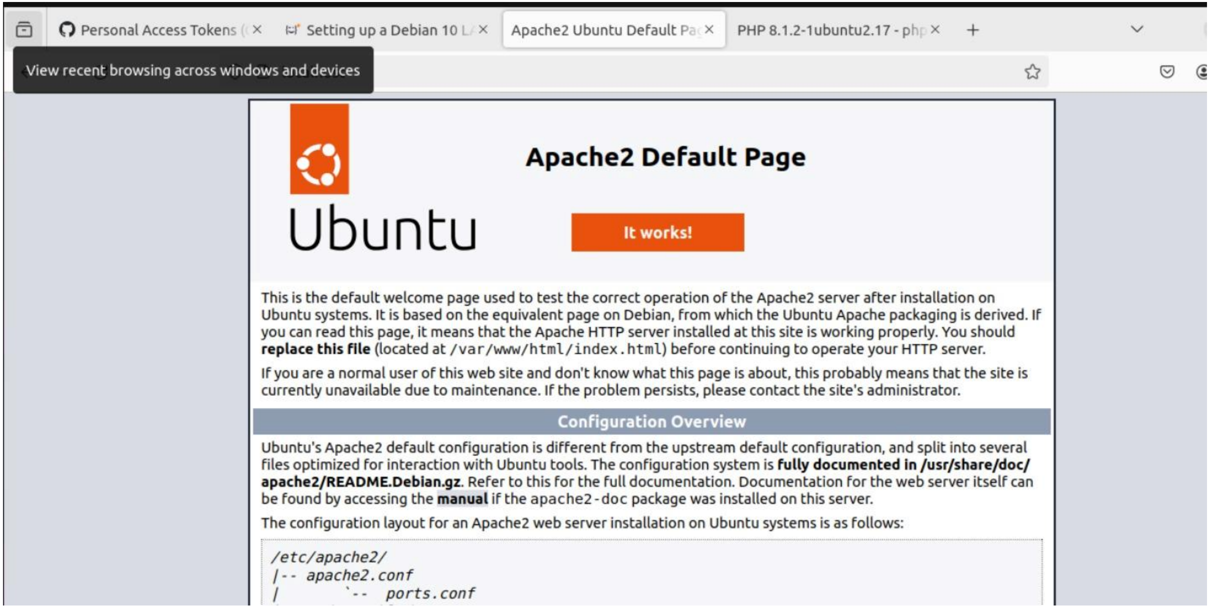
phpinfo.php



```


<?php

phpinfo();
?>
```

## OUTPUT:

- In Here this is the index.html it will gives Automatically when you run the [http://localhost](http://localhost)


Personal Access Tokens (
Setting up a Debian 10 L
Apache2 Ubuntu Default Pa
PHP 8.1.2-1ubuntu2.17 - php
View recent browsing across windows and devices
Ubuntu
It works!

**Apache2 Default Page**

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.

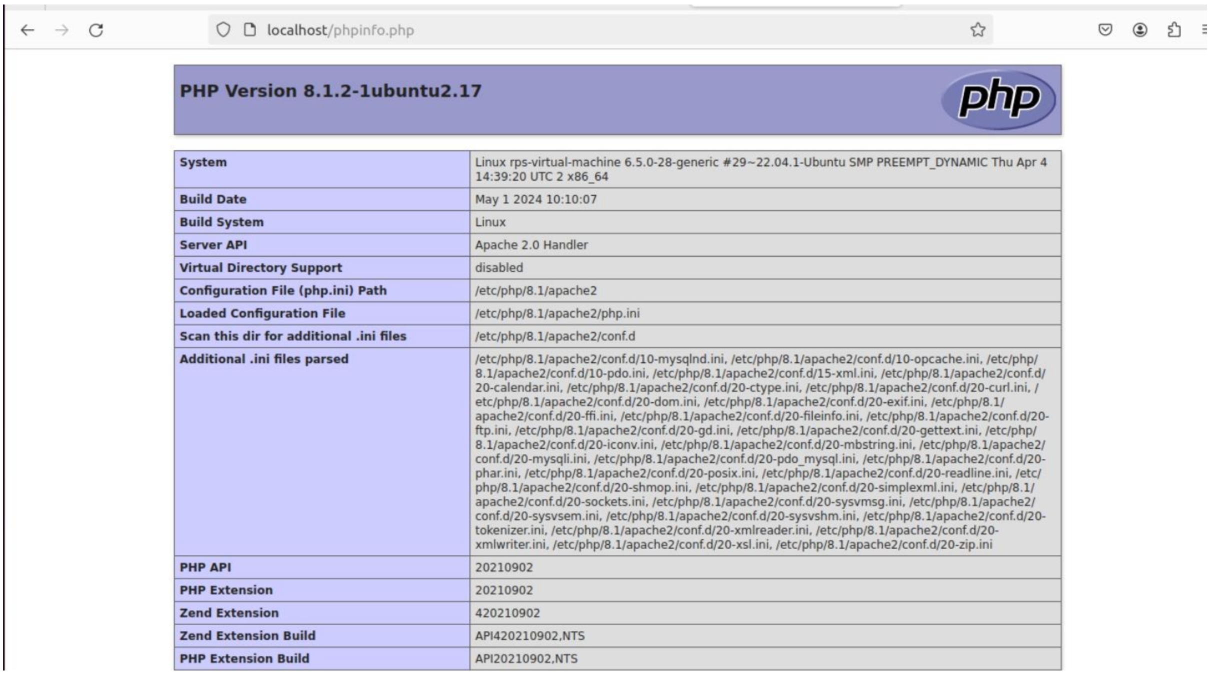
**Configuration Overview**

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.

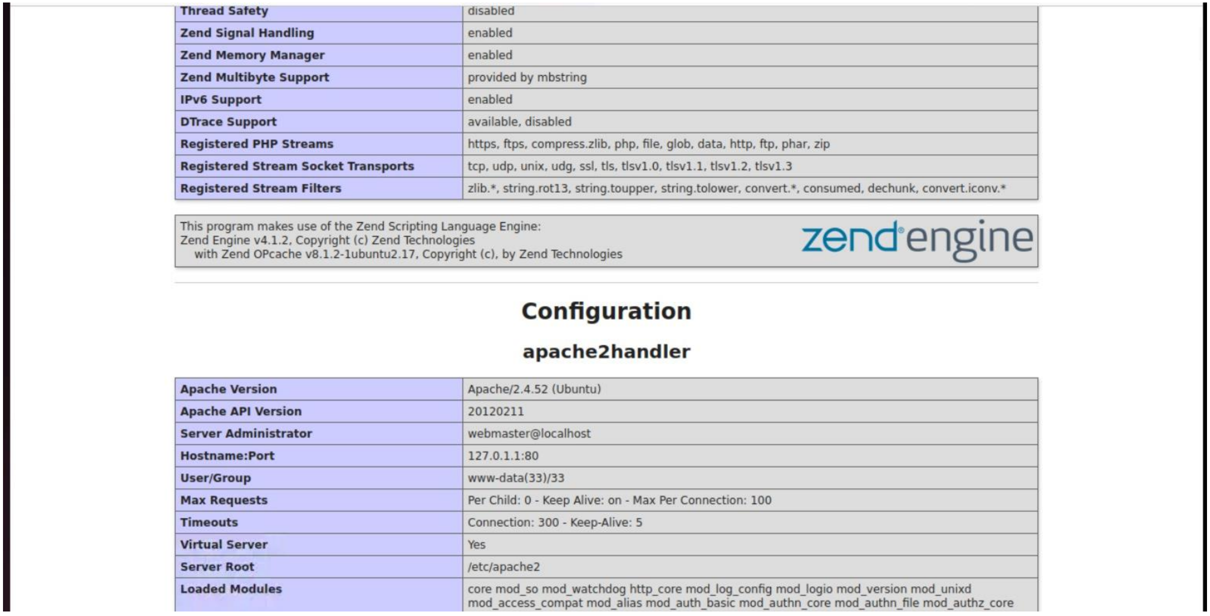
The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
```

- It is the PHP page you can see the below when you enter this Address **[http://localhost/phpinfo.php](http://localhost/phpinfo.php)**



## PHP Version 8.1.2-1ubuntu2.17



| | |
|---|---|
| **System** | Linux rps-virtual-machine 6.5.0-28-generic #29~22.04.1-Ubuntu SMP PREEMPT_DYNAMIC Thu Apr 4 14:39:20 UTC 2 x86_64 |
| **Build Date** | May 1 2024 10:10:07 |
| **Build System** | Linux |
| **Server API** | Apache 2.0 Handler |
| **Virtual Directory Support** | disabled |
| **Configuration File (php.ini) Path** | /etc/php/8.1/apache2 |
| **Loaded Configuration File** | /etc/php/8.1/apache2/php.ini |
| **Scan this dir for additional .ini files** | /etc/php/8.1/apache2/conf.d |
| **Additional .ini files parsed** | /etc/php/8.1/apache2/conf.d/10-mysqlnd.ini, /etc/php/8.1/apache2/conf.d/10-opcache.ini, /etc/php/8.1/apache2/conf.d/10-pdo.ini, /etc/php/8.1/apache2/conf.d/15-xml.ini, /etc/php/8.1/apache2/conf.d/20-calendar.ini, /etc/php/8.1/apache2/conf.d/20-ctype.ini, /etc/php/8.1/apache2/conf.d/20-curl.ini, /etc/php/8.1/apache2/conf.d/20-dom.ini, /etc/php/8.1/apache2/conf.d/20-exif.ini, /etc/php/8.1/apache2/conf.d/20-ffi.ini, /etc/php/8.1/apache2/conf.d/20-fileinfo.ini, /etc/php/8.1/apache2/conf.d/20-ftp.ini, /etc/php/8.1/apache2/conf.d/20-gd.ini, /etc/php/8.1/apache2/conf.d/20-gettext.ini, /etc/php/8.1/apache2/conf.d/20-iconv.ini, /etc/php/8.1/apache2/conf.d/20-mbstring.ini, /etc/php/8.1/apache2/conf.d/20-mysqli.ini, /etc/php/8.1/apache2/conf.d/20-pdo_mysql.ini, /etc/php/8.1/apache2/conf.d/20-phar.ini, /etc/php/8.1/apache2/conf.d/20-posix.ini, /etc/php/8.1/apache2/conf.d/20-readline.ini, /etc/php/8.1/apache2/conf.d/20-shmop.ini, /etc/php/8.1/apache2/conf.d/20-simplexml.ini, /etc/php/8.1/apache2/conf.d/20-sockets.ini, /etc/php/8.1/apache2/conf.d/20-sysvmsg.ini, /etc/php/8.1/apache2/conf.d/20-sysvsem.ini, /etc/php/8.1/apache2/conf.d/20-sysvshm.ini, /etc/php/8.1/apache2/conf.d/20-tokenizer.ini, /etc/php/8.1/apache2/conf.d/20-xmlreader.ini, /etc/php/8.1/apache2/conf.d/20-xmlwriter.ini, /etc/php/8.1/apache2/conf.d/20-xsl.ini, /etc/php/8.1/apache2/conf.d/20-zip.ini |
| **PHP API** | 20210902 |
| **PHP Extension** | 20210902 |
| **Zend Extension** | 420210902 |
| **Zend Extension Build** | API420210902,NTS |
| **PHP Extension Build** | API20210902,NTS |
| **Thread Safety** | disabled |
| **Zend Signal Handling** | enabled |
| **Zend Memory Manager** | enabled |
| **Zend Multibyte Support** | provided by mbstring |
| **IPv6 Support** | enabled |
| **DTrace Support** | available, disabled |
| **Registered PHP Streams** | https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar, zip |
| **Registered Stream Socket Transports** | tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2, tlsv1.3 |
| **Registered Stream Filters** | zlib.*, string.rot13, string.toupper, string.tolower, convert.*, consumed, dechunk, convert.iconv.* |

This program makes use of the Zend Scripting Language Engine:
Zend Engine v4.1.2, Copyright (c) Zend Technologies
with Zend OPcache v8.1.2-1ubuntu2.17, Copyright (c), by Zend Technologies

### Configuration

#### apache2handler

| | |
|---|---|
| **Apache Version** | Apache/2.4.52 (Ubuntu) |
| **Apache API Version** | 20120211 |
| **Server Administrator** | webmaster@localhost |
| **Hostname:Port** | 127.0.1.1:80 |
| **User/Group** | www-data(33)/33 |
| **Max Requests** | Per Child: 0 - Keep Alive: on - Max Per Connection: 100 |
| **Timeouts** | Connection: 300 - Keep-Alive: 5 |
| **Virtual Server** | Yes |
| **Server Root** | /etc/apache2 |
| **Loaded Modules** | core mod_so mod_watchdog http_core mod_log_config mod_logio mod_version mod_unixd mod_access_compat mod_alias mod_auth_basic mod_authn_core mod_authn_file mod_authz_core |

## Conclusion

Building a LAMP Stack web server provides a robust and flexible platform for hosting websites and web applications. By combining Linux, Apache, MySQL, and PHP/Perl/Python.