

Apache

- [Host Multiple Websites On One Apache Server On Ubuntu](#)
- [phpMyAdmin](#)
- [Downgrade PHP8.1 to PHP8.0 or PHP7.4 on Ubuntu 22.04](#)

Host Multiple Websites On One Apache Server On Ubuntu

1. Install Apache Web Server

If you don't have an Apache server installed refer to post: [How To Install Apache On Ubuntu 20.04](#).

2. Create the Directory Structure

First, create a document root directory for both websites by commands:

```
1 sudo mkdir /var/www/html/example.com
```

```
1 sudo mkdir /var/www/html/example1.com
```

Next, create an **index.html** page to be track the results.

Create an **index.html** page for site `example.com`.

```
1 sudo nano /var/www/html/example.com/index.html
```

Copy and paste below content to file.

```
1 <html>
2 <head>
3   <title>example.com</title>
4 </head>
5 <body>
6   <h1>Welcome to example.com website</h1>
7 </body>
8 </html>
```

Next, create an **index.html** page for site `example1.com`.

```
1 sudo nano /var/www/html/example1.com/index.html
```

Copy and paste below content to file.

```
1 <html>
2 <head>
3   <title>example1.com</title>
4 </head>
5 <body>
6   <h1>Welcome to example1.com website</h1>
7 </body>
8 </html>
```

3. Grant Permissions

Grant permissions of `example.com` and `example1.com` directory by commands:

```
1 chown -R www-data:www-data /var/www/html/example.com
```

```
1 chown -R www-data:www-data /var/www/html/example1.com
```

4. Create a Virtual Host Configuration File

Create an Apache virtual host configuration file for `example.com`:

```
1 sudo nano /etc/apache2/sites-available/example.com.conf
```

Add the following lines:

```
1 <VirtualHost *:80>
2   ServerAdmin webmaster@localhost
3   ServerName example.com
4   ServerAlias www.example.com
5   DocumentRoot /var/www/html/example.com
6   ErrorLog ${APACHE_LOG_DIR}/error.log
7   CustomLog ${APACHE_LOG_DIR}/access.log combined
8 </VirtualHost>
```

Create an Apache virtual host configuration file for `example1.com`:

```
1 sudo nano /etc/apache2/sites-available/example1.com.conf
```

And add the following lines:

1	
2	
3	
4	
5	
6	
7	
8	

```
<VirtualHost *:80>
  ServerAdmin webmaster@localhost
  ServerName example1.com
  ServerAlias www.example1.com
  DocumentRoot /var/www/html/example1.com
  ErrorLog ${APACHE_LOG_DIR}/error.log
  CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Then, enable the virtual host configuration file with the following commands:

1	<code>sudo a2ensite example.com.conf</code>
---	---

1	<code>sudo a2ensite example1.com.conf</code>
---	--

Restart Apache for the changes to take effect.

1	<code>sudo systemctl restart apache2</code>
---	---

5. Test your Results

Visit the website with the path `http://example.com` you will see results like this:

Result of example
Result to focus on type unknown

And visit the website with the path `http://example1.com` you will see results like this:

Result of example1
Result to focus on type unknown

Thank you for reading!

phpMyAdmin

plugin_interface.lib.php error

```
sudo vim /usr/share/phpmyadmin/libraries/plugin_interface.lib.php
```

```
if ($options != null && count($options) > 0) {
```

```
if ($options != null && count((array)$options) > 0) {
```

Secure PhpMyAdmin Login Interface

1. Change Default PhpMyAdmin Login URL
2. Enable HTTPS on PhpMyAdmin
3. Password Protect on PhpMyAdmin
4. Disable root Login to PhpMyAdmin

```
# vim /opt/lampp/etc/extra/httpd-xampp.conf
```

```
Alias /db "/opt/lampp/phpmyadmin"
```

```
/opt/lampp/lampp restart
```

<http://localhost/db/index.php>

```
vim /etc/php/7.2/cli/php.ini
```

-> for database upload limit

vim /opt/lampp/phpmyadmin/config.inc.php -> for set root password

vim /opt/lampp/etc/extra/httpd-ssl.conf -> for ssl entry

vim /opt/lampp/etc/original/httpd.conf -> for apache configure

error phpmyadmin

```
sudo vim /usr/share/phpmyadmin/libraries/sql.lib.php
```

```
|| (count($analyzed_sql_results['select_expr'] == 1)
```

```
|| (count($analyzed_sql_results['select_expr']) == 1
```

root access denied

phpMyAdmin

```
sudo nano /opt/lampp/etc/my.cnf
```

Skip-grant-tables

How to increase

phpMyAdmin upload file size

Step - 1: open php.ini of the appropriate PHP version(s).

For PHP v5.0

```
sudo vim /etc/php5/apache2/php.ini
```

For PHP v7.2

```
sudo vim /etc/php/7.2/apache2/php.ini
```

Step - 2: Now need to increase below parameters in the php.ini file.

```
memory_limit = 1500M
```

```
post_max_size = 1500M
```

```
upload_max_filesize = 1500M
```

Step - 3: Now you need to restart your apache server to effect changes.

```
sudo service apache2 restart
```

increase import database size

nano

/etc/php/7.2/apache2/php.ini

```
memorylimit 128M
```

```
postmaxsize 12M
```

```
uploadmaxfilesize 10M
```

Downgrade PHP8.1 to PHP8.0 or PHP7.4 on Ubuntu 22.04

Ubuntu 22.04 comes with `PHP 8.1`. This article describes how to change this and install and set as default the version `8.0` or `7.4`.

Install PHP Version

Install the required dependencies.

```
sudo apt install software-properties-common ca-certificates lsb-release apt-transport-https
```

Add the required PPA

```
LC_ALL=C.UTF-8 sudo add-apt-repository ppa:ondrej/php
```

Update the Apt package manager

```
sudo apt update
```

Install PHP 8.0 and all the PHP modules for Drupal 9.

```
sudo apt install php8.0  
sudo apt install php8.0-mysql php8.0-mbstring php8.0-xml php8.0-curl php8.0-gd
```

Switch PHP versions and enable PHP 8.0

Set 8.09 as the default PHP version for CLI and Apache.

Command Line PHP

```
sudo update-alternatives --config php
```

Select php8.0 from the list with the available options:

```
There are 2 choices for the alternative php (providing /usr/bin/php). Selection    Path
-----
  0                /usr/bin/php8.1  81          auto mode
* 1                /usr/bin/php8.0  80          manual mode
  2                /usr/bin/php8.1  81          manual mode
Press <enter> to keep the current choice[*]
```

Apache PHP

Disable PHP 8.1

```
sudo a2dismod php8.1
sudo systemctl restart apache2
```

Enable PHP 8.0

```
sudo a2enmod php8.0
sudo systemctl restart apache2
```

Verify PHP Version

```
php -vOutput
PHP 8.0.18 (cli) (built: May 1 2022 04:42:09) ( NTS )
Copyright (c) The PHP Group
Zend Engine v4.0.18, Copyright (c) Zend Technologies
with Zend OPcache v8.0.18, Copyright (c), by Zend Technologies
```

For verifying the PHP version Apache uses, specify the configuration `.ini` file.

```
php -c /etc/php/apache2/php.ini -v
```