

Web Server deployment

```
sudo apt update -y
```

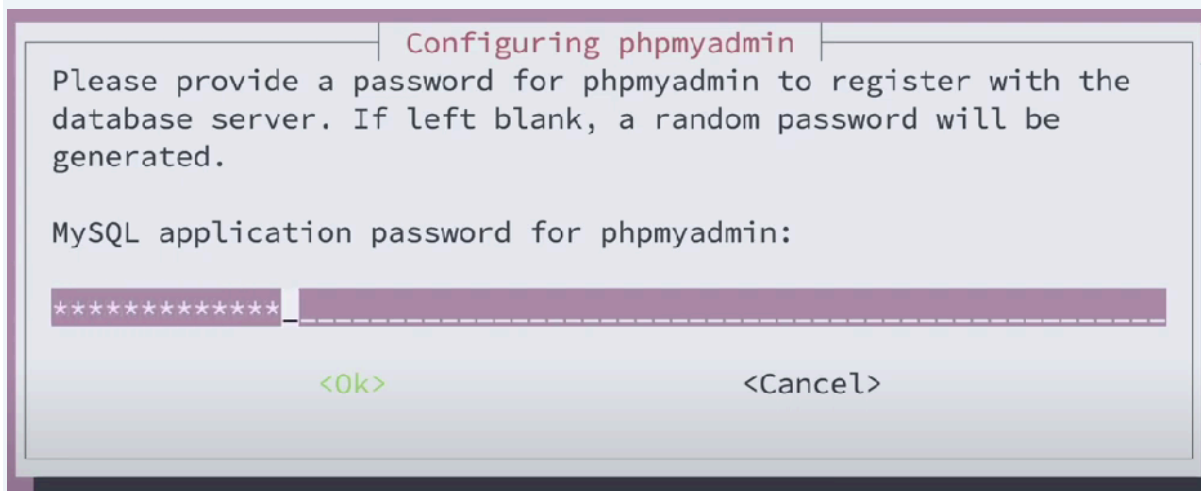
```
sudo apt install nginx php-fpm
```

```
sudo apt install mysql-server -y
```

```
sudo mysql -u root
```

```
mysql> ALTER USER 'root@localhost' identified by 'mypassword321';  
quit;\
```

```
sudo apt install phpmyadmin -y (tick for nginx else skip)
```



Type the mysql root password.

Apache/Nginx, MySQL, PHP deployment and configuration on Ubuntu

Prerequisites

- An Ubuntu server (20.04 or later recommended).
- A user with sudo privileges.

Step-by-Step Guide for Installation

1. Update Your Ubuntu Server

```
sudo apt update  
sudo apt upgrade
```

Step 2: Install Apache Server

```
sudo apt install apache2
```

Step 3: Install MySQL

```
sudo apt install mysql-server  
sudo mysql_secure_installation
```

Step 4: Install PHP and other library

```
sudo apt install php php-mysql libapache2-mod-php
```

[Installing PHP 8.x on Ubuntu 22 | Reintech media](#)

Step 5: Configure Apache

Create a Virtual Host file:

```
sudo vi /etc/apache2/sites-available/techaxis.conf
```

Add the following configuration:

```
<VirtualHost *:80>  
    ServerAdmin webmaster@yourdomain.com  
    ServerName yourdomain.com  
    ServerAlias www.yourdomain.com
```

```
DocumentRoot /var/www/yourdomain
<Directory /var/www/yourdomain>
    Options Indexes FollowSymLinks
    AllowOverride All
    Require all granted
</Directory>
ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

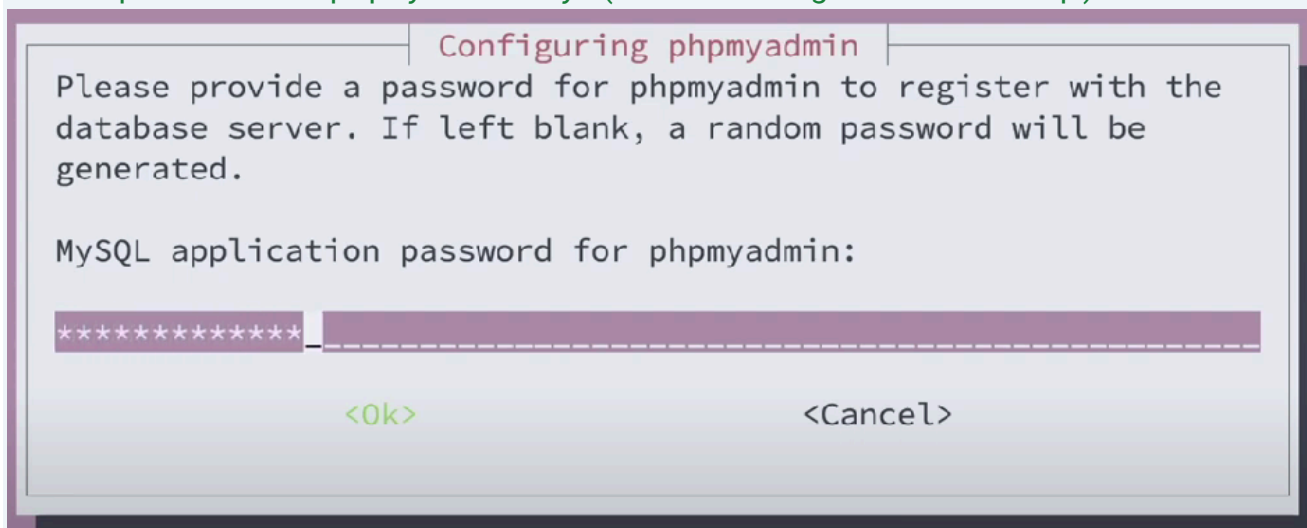
Enable the new site and the rewrite module:

```
sudo a2ensite yourdomain.conf
sudo a2enmod rewrite

sudo systemctl restart apache2
```

Install phpmyadmin:

```
sudo apt install phpmyadmin
sudo apt install phpmyadmin -y (tick for nginx else skip)
```



Leave no and proceed in installation

Configuring phpmyadmin

The phpmyadmin package must have a database installed and configured before it can be used. This can be optionally handled with dbconfig-common.

If you are an advanced database administrator and know that you want to perform this configuration manually, or if your database has already been installed and configured, you should refuse this option. Details on what needs to be done should most likely be provided in /usr/share/doc/phpmyadmin.

otherwise, you should probably choose this option.

Configure database for phpmyadmin with dbconfig-common?

<Yes>

<No>

Step 6: Configure MySQL Database for Your Application

Log in to MySQL:

```
sudo mysql -u root -p
```

Create a new database and user:

```
CREATE DATABASE mydatabase;  
CREATE USER techuser@'localhost' IDENTIFIED BY 'techpassword';  
GRANT ALL PRIVILEGES ON mydatabase.* TO 'techuser'@'localhost';  
FLUSH PRIVILEGES;  
EXIT;
```

```
JUST VERIFY:  SELECT user FROM mysql.user;
```

localhost/phpmyadmin =====> will not work by default

TO ENABLE PHPMYADMIN:

In /etc/apache2/apache2.conf (add the below line)

```
Include /etc/phpmyadmin/apache.conf
```

```
sudo phpenmod mbstring
```

```
sudo systemctl restart apache2
```

```
=====
```

```
GRANT ALL PRIVILEGES ON *.* TO 'phpmyadmin'@'localhost';
```

```
FLUSH PRIVILEGES;
```

```
=====
```

```
# sudo apachectl -S
```

```
# sudo a2dissite 000-default.conf
```

```
# sudo systemctl restart apache2
```

```
# sudo a2ensite mydemo.conf
```

```
# sudo systemctl restart apache2
```

LEMP Stack (Nginx)

Step 1: Install Nginx

```
sudo apt install nginx
```

Step 2: Install MySQL

```
sudo apt install mysql-server mysql-client
```

```
sudo mysql_secure_installation
```

Step 3: Install PHP

```
sudo apt install php-fpm php-mysql unzip curl -y
```

Step 4: Configure Nginx

Create a server block file as mentioned below:

```
sudo vi /etc/nginx/sites-available/wordpress
```

Add the following configuration:

```
server {  
  
    listen 80;  
    server_name yourdomain.com www.yourdomain.com;  
    root /var/www/yourdomain;  
  
    index index.php index.html index.htm;  
  
    location / {  
        try_files $uri $uri/ /index.php?$args;  
    }  
  
    location ~ /\.php$ {  
        include snippets/fastcgi-php.conf;  
        fastcgi_pass unix:/var/run/php/php7.4-fpm.sock;  
    }  
  
    location ~ /\.ht {  
        deny all;  
    }  
}
```

Enable the new site and test the configuration:

```
sudo ln -s /etc/nginx/sites-available/yourdomain
/etc/nginx/sites-enabled/
sudo nginx -t
sudo systemctl restart nginx
```

Step 6: Configure MySQL Database for Your Application

Log in to MySQL:

```
sudo mysql -u root -p
```

Create a new database and user:

```
CREATE DATABASE mydatabase;
CREATE USER techuser@'localhost' IDENTIFIED BY 'techpassword';
GRANT ALL PRIVILEGES ON mydatabase.* TO 'techuser'@'localhost';
FLUSH PRIVILEGES;
EXIT;
```

Step 7: Configure PHP

```
sudo vi /etc/php/7.4/apache2/php.ini # For Apache
sudo vi /etc/php/7.4/fpm/php.ini      # For Nginx
```

Ensure the following settings are configured (values can be adjusted to your needs):
ini

```
file_uploads = On
allow_url_fopen = On
memory_limit = 256M
upload_max_filesize = 100M
max_execution_time = 300
```

Restart the web server to apply the changes:

```
sudo systemctl restart apache2 # For Apache
sudo systemctl restart php7.4-fpm # For Nginx
```

Step 8: Secure Your Server allowing it to ufw

Configure the firewall to allow traffic on necessary ports:

```
sudo ufw allow OpenSSH
sudo ufw allow 'Apache Full' # For Apache
sudo ufw allow 'Nginx Full' # For Nginx
sudo ufw enable
Sudo ufw reload
```

Install and configure SSL/TLS using Let's Encrypt:

```
sudo apt install certbot python3-certbot-apache # For Apache
sudo apt install certbot python3-certbot-nginx # For Nginx
sudo certbot --apache # For Apache
sudo certbot --nginx # For Nginx
```

Follow the prompts to obtain and install the SSL certificate.

Lets create a phpinfo.php within the /var/www/html/phpinfo.php

With the content as phpinfo.php

<?php phpinfo() ?>

WORDPRESS

Install Apache2

Install mysql-server

Install php

#####

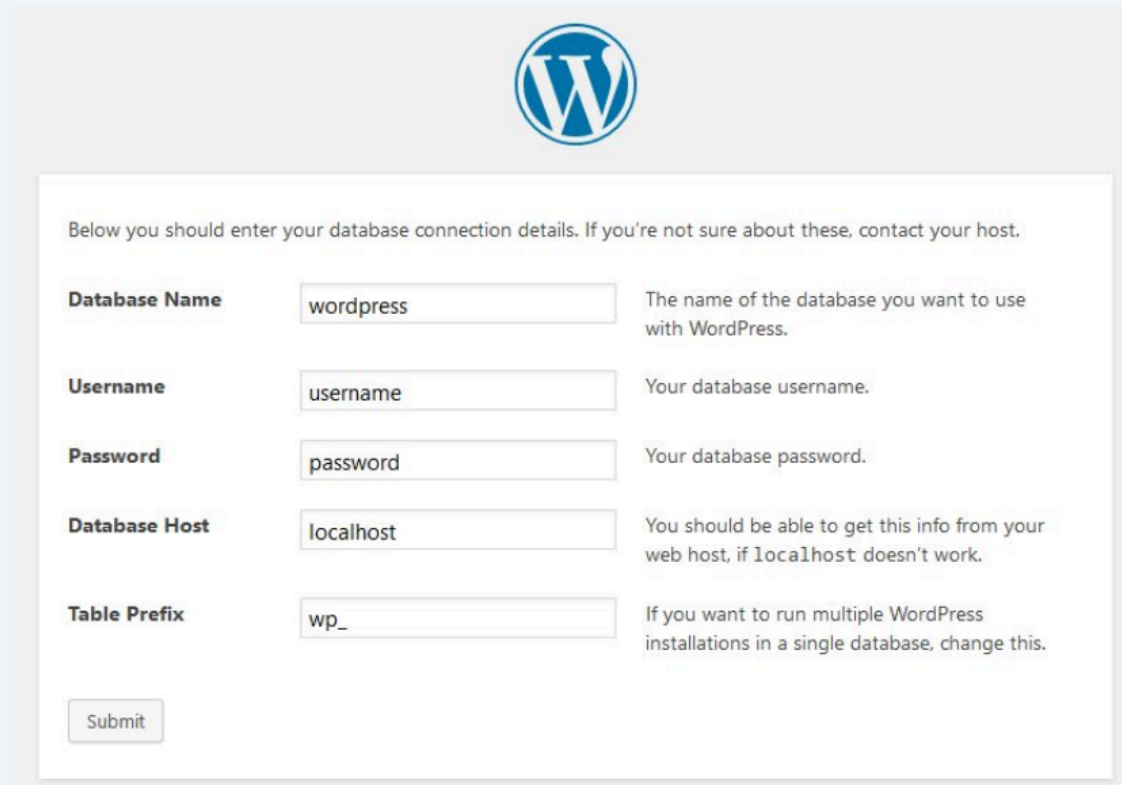
```
ALTER USER 'root'@localhost IDENTIFIED WITH mysql_native_password BY 'myrootpw@123';
CREATE USER 'wp_user'@'localhost' IDENTIFIED BY 'Mywppassword@123';
CREATE DATABASE wp;
GRANT ALL PRIVILEGES ON wp.* TO 'wp_user'@'localhost';
FLUSH PRIVILEGES;
```

DOWNLOAD WORDPRESS LATEST VERSION:


```
wget https://wordpress.org/latest.tar.gz
tar -xvf latest.tar.gz
sudo mv wordpress/ /var/www/html/
```

IN BROWSER:

Go To: localhost/wordpress

The image shows the WordPress installation database configuration screen. At the top is the WordPress logo. Below it is a text box with the instruction: "Below you should enter your database connection details. If you're not sure about these, contact your host." There are five input fields with labels and descriptions: "Database Name" (wordpress), "Username" (username), "Password" (password), "Database Host" (localhost), and "Table Prefix" (wp_). A "Submit" button is at the bottom left.

Database Name	<input type="text" value="wordpress"/>	The name of the database you want to use with WordPress.
Username	<input type="text" value="username"/>	Your database username.
Password	<input type="text" value="password"/>	Your database password.
Database Host	<input type="text" value="localhost"/>	You should be able to get this info from your web host, if localhost doesn't work.
Table Prefix	<input type="text" value="wp_"/>	If you want to run multiple WordPress installations in a single database, change this.

IF ERROR THEN:

CREATE A wp-config.php to write those content displayed.

since we have ip/wordpress in the browser.

If we need to change wordpress to be opened via. direct IP then,

Modify the 000-default.conf as Document Root to /var/www/html/wordpress

IF YOU HAVE A DOMAIN then, point the A record of DNS to the public IP of wordpress.

Finally we need to add below lines to 000-default.conf.

ServerName yourdomain.com

ServerAlias www.yourdomain.com

IN GENERAL SETTINGS OF WORDPRESS

URL : http://yourdomain.com

Site URL : http://yourdomain.com

FOR HTTPS

```
sudo apt-get update  
sudo apt install certbot python3-certbot-apache
```

```
Request and install ssl on your site with certbot  
sudo certbot --apache
```