Wordpress?? - Debian 12

????

- **PVE**
- **LXC**
- **Debian 12**

??nginx mariadb

```
apt update
apt -y install nginx mariadb-server

systemctl enable nginx
systemctl start nginx

systemctl enable mariadb
systemctl start mariadb
```

??PHP

```
apt -y install php php-cli php-fpm php-mysql php-json php-opcache php-mbstring php-xml php-gd
php-curl libapache2-mod-php8.2-
```

???mariadb

```
NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
      SERVERS IN PRODUCTION USE!  PLEASE READ EACH STEP CAREFULLY!
In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
haven't set the root password yet, you should just press enter here.
#□□Enter□□□□□□□
Enter current password for root (enter for none):
OK, successfully used password, moving on...
#□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
#□□□root□□□□□□□□□□□□□□□□□□□n□□□□□□□
#□MariaDB□□□□□□□□□□□□□□□n□□□□□□□□□y□□□□MariaDB root□□□□□
Setting the root password or using the unix_socket ensures that nobody
can log into the MariaDB root user without the proper authorisation.
```

```
You already have your root account protected, so you can safely answer 'n'.
Switch to unix_socket authentication [Y/n] n
 ... skipping.
You already have your root account protected, so you can safely answer 'n'.
Change the root password? [Y/n] n
 ... skipping.
#□□□□□□□□□□□□□□□□□□□□□□□□□□□□y□□□□□□□□□□□□
By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them.  This is intended only for testing, and to make the installation
go a bit smoother.  You should remove them before moving into a
production environment.
Remove anonymous users? [Y/n] y
 ... Success!
#□□□□□□□□□□root□localhost□□DB□□□□□y□□□□□□□□□□
Normally, root should only be allowed to connect from 'localhost'.  This
ensures that someone cannot guess at the root password from the network.
Disallow root login remotely? [Y/n] y
 ... Success!
#□□□□□□□□□□□□test□□□□□□□□□□y□□□□□□
By default, MariaDB comes with a database named 'test' that anyone can
access.  This is also intended only for testing, and should be removed
before moving into a production environment.
Remove test database and access to it? [Y/n] y
 - Dropping test database...
 ... Success!
 - Removing privileges on test database...
 ... Success!
#□□□□□□□□□□□□□□□□□□□□□□y□□□□□□□□□□□□□□□□□□
Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.
Reload privilege tables now? [Y/n] y
 ... Success!
Cleaning up...
All done!  If you've completed all of the above steps, your MariaDB
installation should now be secure.
#□□□□□□□□MariaDB□□□□□□□□□□□□□□
Thanks for using MariaDB!
```

?????????

```
mariadb -u root -p
CREATE DATABASE wordpress;
GRANT ALL PRIVILEGES ON wordpress.* TO 'wpuser'@'localhost' IDENTIFIED BY 'wpuserpswd' WITH
GRANT OPTION;
FLUSH PRIVILEGES;
exit
```

??PHP??

□□ `/etc/php/8.2/fpm/php.ini`

```
cgi.fix_pathinfo=0
[Date]
date.timezone = Asia/Taipei
upload_max_filesize = 50M
post_max_size = 200M
memory_limit = 512M
max_execution_time = 600
max_input_time = 600
default_socket_timeout = 600
```

```
systemctl start php8.2-fpm && systemctl enable php8.2-fpm
```

?? Nginx

□□ `/etc/nginx/nginx.conf`

```
http□□□□

client_max_body_size 50M;
client_body_timeout 120s;
```

??nginx wordpress???

□□ `/etc/nginx/conf.d/wordpress.conf`

```
server {
            listen 80;
            root /var/www/html;
            index index.php index.html;
```

```
        server_name _;

        access_log /var/log/nginx/wordpress.access.log;
        error_log /var/log/nginx/wordpress.error.log;

        location / {
                try_files $uri $uri/ =404;
        }
                location /wordpress {
                                        if (-f $request_filename/index.html){
                                                rewrite (.*) $1/wordpress/index.html
break;
                                        }
                                        if (-f $request_filename/index.php){
                                                rewrite (.*) $1/wordpress/index.php;
                                        }
                                        if (!-f $request_filename){
                                                rewrite (.*) /wordpress/index.php;
                                        }
        }
                location /wordpress/wp-admin {
                                        #try_files $uri $uri/ /index.php?$args;
                                         rewrite /wp-admin$ $scheme://$host$uri/
permanent;
        }
        location ~ \.php$ {
                include snippets/fastcgi-php.conf;
                fastcgi_intercept_errors on;
                fastcgi_pass unix:/run/php/php8.2-fpm.sock;
        }

        location ~ /\.ht {
                deny all;
        }

        location = /favicon.ico {
                log_not_found off;
                access_log off;
        }
```

```
            location = /robots.txt {
                        allow all;
                        log_not_found off;
                        access_log off;
            }

            location ~* \.(js|css|png|jpg|jpeg|gif|ico)$ {
                        expires max;
                        log_not_found off;
            }
}
```

```
rm /etc/nginx/sites-enabled/default
rm /etc/nginx/sites-available/default

systemctl restart nginx
```

??Wordpress

```
mkdir -p /var/www/html

cd /var/www/html
wget https://tw.wordpress.org/wordpress-6.7-zh_TW.tar.gz
tar -zxvf wordpress-6.7-zh_TW.tar.gz
rm wordpress-6.7-zh_TW.tar.gz
chown -R www-data:www-data wordpress/
```

□□□ `http://ip`□□□□□□□□□□□□□□□□□□□□□□

??????

□□□□□□□□□□□□□ `wp-config.php`□□□□□□

```
define( 'AUTOMATIC_UPDATER_DISABLED', true );
define( 'WP_AUTO_UPDATE_CORE', false );
```

Revision #3
Created 6 June 2025 00:36:52 by Jason
Updated 6 June 2025 01:06:21 by Jason