

Чтобы установить WordPress (в качестве примера используется домен wp-site.test) на Linux Debian с самоподписанным сертификатом необходимо:

1. Обновить систему

```
apt update; apt upgrade
```

2. Устанавливаем необходимые пакеты

```
apt install nginx apache2 htop curl wget mariadb-server php php-curl  
php-gd php-mbstring php-xml php-xmlrpc php-soap php-intl php-zip unzip  
php-mysql
```

3. Временно выключим и добавим в автозапуск сервисы

```
systemctl enable nginx apache2 mariadb; systemctl stop nginx apache2
```

4. Поправим порт в ports.conf на 8080, т.к. делаем связку nginx+apache2+php+mariadb и создадим виртуальный хост на apache2

```
nano /etc/apache2/ports.conf; nano /etc/apache2/sites-available/wp-  
site.test.conf
```

```
<VirtualHost 127.0.0.1:8080>  
  ServerName wp-site.test  
  ServerAlias www.wp-site.test  
  DocumentRoot "/var/www/wp-site.test"
```

```
<Directory "/var/www/wp-site.test/">  
  AllowOverride All  
  allow from all  
  Options -Indexes  
  #Options FollowSymLinks  
</Directory>
```

```
<FilesMatch "\.ph(p[3-8]?|tml)$">  
  SetHandler application/x-httpd-php  
</FilesMatch>
```

```
ErrorLog /var/www/log/apache2/wp-site.test-error.log  
CustomLog /var/www/log/apache2/wp-site.test-access.log combined  
<ifModule mod_headers.c>  
Header set X-XSS-Protection "1; mode=block"
```

```
</IfModule>

<IfModule php8_module>
    php_flag session.cookie_httponly 1
    php_flag session.cookie_secure 1
</IfModule>

</VirtualHost>
```

5. Создаем папки для логов

```
mkdir /var/www/log/; mkdir /var/www/log/apache2/; mkdir
/var/www/log/nginx/
```

6. Создаем папку для сайта и тестовый php-файл

```
mkdir /var/www/wp-site.test/; echo '<?php phpinfo(); ?>' >
/var/www/wp-site.test/index.php
```

7. Включаем редиректы на apache2 и проверяем конфигурацию

```
a2enmod rewrite; apache2ctl configtest
```

8. Включаем сайт

```
a2ensite wp-site.test.conf
```

9. Создаем сертификат и ключ для https-соединения

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout
/etc/ssl/private/wp-site.test-selfsigned.key -out /etc/ssl/certs/wp-
site.test-selfsigned.crt
```

10. Создаем виртуальный хост для nginx и прописываем конфиг

```
nano /etc/nginx/sites-available/wp-site.test
```

```
server {
    listen 80;
    server_name wp-site.test www.wp-site.test;
    add_header Content-Security-Policy upgrade-insecure-requests;

```

```

        return 302 https://wp-site.test$request_uri;
    }

server {
    listen 443 ssl;
    server_name wp-site.test www.wp-site.test;
    access_log /var/www/log/nginx/wp-site.test-access.log;
    error_log /var/www/log/nginx/wp-site.test-error.log;

    ssl_certificate /etc/ssl/certs/wp-site.test-selfsigned.crt;
    ssl_certificate_key /etc/ssl/private/wp-site.test-selfsigned.key;

    ssl_protocols TLSv1 TLSv1.1 TLSv1.2 TLSv1.3;
    add_header Strict-Transport-Security 'max-age=15552000;
includeSubDomains';
    add_header Content-Security-Policy upgrade-insecure-requests;
    ssl_ciphers
EECDH+ECDSA+AESGCM:EECDH+aRSA+AESGCM:EECDH+ECDSA+SHA384:"убрать
перенос"
        EECDH+ECDSA+SHA256:EECDH+aRSA+SHA384: "убрать перенос"
EECDH+aRSA+SHA256:EECDH:EDH+aRSA:HIGH:!aNULL:!eNULL:!LOW:!RC4:!3DES:"y
братъ перенос"
        !MD5:!EXP:!PSK:!SRP:!SEED:!DSS:!CAMELLIA;
    ssl_prefer_server_ciphers on;

    client_max_body_size 200M;
    keepalive_requests 100;
    keepalive_timeout 30;

    reset_timedout_connection on;
    client_body_timeout 10;
    send_timeout 2;
    fastcgi_ignore_client_abort on;

location / {
    fastcgi_ignore_client_abort on;
    proxy_ignore_client_abort on;

    proxy_pass http://127.0.0.1:8080/;
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;

    #proxy_set_header Connection "";
    proxy_connect_timeout 120;
    proxy_send_timeout 120;
    proxy_read_timeout 180;

```

```
}  
  
location ~*  
\.(woff2|svg|png|jpg|jpeg|gif|css|js|pdf|doc|zip|rar|docx|avi|tiff|"уб  
рать перенос"  
rtf|gz|gzip|mpg|mpeg|"убрать перенос"  
mp3|mp4|tgz|wav|wmv|wma|xls|xlsx|xml|bmp|ico|psd)$ {  
root /var/www/wp-site.test/  
}  
}
```

11. Включаем хост на nginx

```
ln -s "/etc/nginx/sites-available/wp-site.test" /etc/nginx/sites-  
enabled/wp-site.test
```

12. Проверяем конфиг nginx -t[/code]

13. Задаем пароль root для mariadb

```
mysqladmin -u root password newpasstest
```

14. Заходим в СУБД

```
mysql -uroot -p
```

15. Создаем пользователя с паролем и базу данных для WordPress

```
CREATE DATABASE `wptestdb`;  
CREATE USER 'wpuzer'@'localhost' IDENTIFIED BY 'password';  
GRANT ALL PRIVILEGES ON `wptestdb`.* TO 'wpuzer'@'localhost';  
FLUSH PRIVILEGES;  
quit;
```

16. Перезапускаем сервисы и проверяем работу сайта (надо в браузере зайти на URL)

```
systemctl restart apache2 nginx mariadb
```

17. Переходим в папку сайта и удаляем ненужное.

```
cd /var/www/wp-site.test/; rm index.php
```

18. Качаем сам WordPress

```
wget https://ru.wordpress.org/latest-ru_RU.zip;
```

19. Распаковываем скаченное и перемещаем в корень сайта

```
unzip latest-ru_RU.zip;mv wordpress/* ./
```

20. Удаляем ненужное

```
rm -r wordpress/; rm latest-ru_RU.zip
```

21. Ставим права на папку с файлами сайта

```
cd ../; chown www-data:www-data -R wp-site.test/; chmod g+s wp-site.test/
```

22. Переходим в браузере по URL домена и настраиваем WordPress

```
Имя - Тестовый сайт  
Логин с паролем wp-user:4dveRElmfKjNrltN  
Почта для восстановления test@example.loc
```

Видео на Youtube:

Видео на Rutube: