

Этот документ описывает пошаговую установку нового сайта на WordPress с помощью скрипта install_wp.sh. Всё работает через терминал, красиво и быстро.

Н Что нужно до начала

- Сервер с установленным **Debian Server** / **Ubuntu Server**
- Рабочий стек: Apache + MySQL (MariaDB) + PHP
- Готовый скрипт install_wp.sh (лежит в ~/scripts)

ຶ Шаг 1. Запуск скрипта

Открой терминал на сервере и выполни:

cd ~/scripts
sudo ./install_wp.sh

Скрипт спросит доменное имя сайта. Введи, например:

my_site.local

Он:

- создаст папку /var/www/my_site.local
- настроит Apache
- создаст БД wp_my_site_local
- задаст пароли
- установит WordPress и выведет логин/пароль
- добавит сайт в /etc/hosts

💢 Шаг 2. Настройка /etc/hosts на сервере

Если ты работаешь **локально на сервере**, то ничего делать не нужно — скрипт сам добавит: 127.0.0.1 my_site.local

Если ты заходишь с другого ПК в сети:

hostname -I # покажет IP, например 192.168.1.xx

■ Шаг 3. Файл hosts на клиенте

☆ Linux/macOS:

sudo nano /etc/hosts

В конце добавь:

192.168.1.xx my_site.local

🖈 Инструкция:

- 1. 🔍 Открой Блокнот от имени администратора:
 - Нажми Win → введи Блокнот
 - Правый клик Запуск от имени администратора
- 2. 🕅 В Блокноте выбери:

Копировать Редактировать Файл \rightarrow Открыть

И вставь в путь:

makefile КопироватьРедактировать C:\Windows\System32\drivers\etc\

Внизу переключи тип файла на «Все файлы (.)», чтобы hosts был виден.

- 3. 🗗 Открой файл hosts
- 4. 🕆 В конец файла добавь строку:

lua КопироватьРедактировать 192.168.1.xx my_site.local

Заменив XX на IP-адрес твоего сервера (например, 192.168.1.85)

- 5. 🖺 Сохрани файл (Ctrl+S)

powershell КопироватьРедактировать Add-Content -Path "C:\Windows\System32\drivers\etc\hosts" -Value "`n192.168.1.xx my_site.local"

Объяснение:

- Add-Content как echo >>
- `п это перевод строки

- Путь к hosts полный и без кавычек не сработает
- Выполняется только в админском PowerShell

Э Пример с реальным IP:

powershell КопироватьРедактировать Add-Content -Path "C:\Windows\System32\drivers\etc\hosts" -Value "`n192.168.1.85 wordpress.local"

Альтернатива через переменные:

powershell КопироватьРедактировать \$ip = "192.168.1.85" \$host = "wordpress.local" Add-Content -Path "\$env:SystemRoot\System32\drivers\etc\hosts" -Value "`n\$ip \$host"

6.

arduino КопироватьРедактировать http://my_site.local

☼ И он будет вести на твой WordPress, поднятый дома.

Хочешь, сделаю .ps1-скрипт, который сам добавляет такую строку в hosts под Windows?

Шаг 4. Открой сайт

В браузере:

http://my_site.local

🖺 Где смотреть доступы

cat ~/wp_sites.log

Пример:

http://my_site.local | WP: admin_my_site_local / zP2txA9kR0qN | MySQL: admin_my_site_local / sJf83lY9w8jLm

\delta Полезные команды

```
# Проверка Apache конфигов sudo apache2ctl -S

# Перезапуск Apache sudo systemctl restart apache2

# Удаление html и заглушка sudo rm -rf /var/www/html/*
echo "<h1>1 Здесь ничего нет</h1>" | sudo tee /var/www/html/index.html

# Удаление сайта sudo a2dissite my_site.local.conf sudo rm -rf /var/www/my_site.local sudo mysql -e "DROP DATABASE wp_my_site_local;"
```

② Рекомендации

- Используй строчные имена: myblog.local, site42.local
- Избегай пробелов, заглавных букв, особых символов
- Каждый WordPress-сайт и БД отдельные

₽ Вопросы?

Задавай в комментариях под видео или пиши сюда, если хочешь сделать WordPressсервер ещё мощнее!

Шаг 5. Автоматизация и локальная разработка WordPress на LAMP

5.1. Подготовка Dev-среды на базе Linux

Для локальной разработки рекомендуется использовать LAMP-стек (Linux, Apache, MySQL, PHP). Убедитесь, что установлены:

- Apache2
- MySQL/MariaDB
- PHP >= 7.4 с модулями: php-mysql, php-xml, php-mbstring, php-curl, php-zip, php-gd
- Дополнительно: curl, wget, unzip, tar, git

sudo apt update && sudo apt install apache2 mysql-server php php-mysql php-xml php-mbstring php-curl php-zip php-gd curl wget unzip git -y

5.2. Настройка виртуальных хостов (пример wordpress.local)

Создаем отдельные директории и конфигурации для каждого сайта:

```
sudo mkdir -p /var/www/site_name.local
sudo chown -R $USER:www-data /var/www/site name.local
```

Создаем файл конфигурации Apache:

sudo nano /etc/apache2/sites-available/wordpress.local.conf

Пример содержимого:

```
<VirtualHost *:80>
    ServerName site_name.local
    DocumentRoot /var/www/site_name.local
    <Directory /var/www/site_name.local>
        AllowOverride All
        Require all granted
    </Directory>
        ErrorLog ${APACHE_LOG_DIR}/site_name.local_error.log
        CustomLog ${APACHE_LOG_DIR}/site_name.local_access.log combined
</VirtualHost>
```

Активируем конфигурацию:

```
sudo a2ensite wordpress.local.conf
sudo systemctl reload apache2
```

И добавляем 127.0.0.1 site_name.localв/etc/hosts.

5.3. Скрипт install_wp.sh из репозитория WPS4LAMP

Скрипт позволяет развернуть WordPress-сайт в указанной директории с созданием БД и конфигурации виртуального хоста.

Использование:

```
chmod +x install_wp.sh
sudo ./install_wp.sh site_name.local
```

Скрипт выполнит:

- Загрузку последней версии WordPress
- Создание базы данных с уникальным именем и пользователем
- Настройку wp-config.php
- Копирование файлов в нужную директорию
- Настройку Арасһе и перезапуск сервера

5.4. VS Code Web и Git

Для редактирования кода на сервере можно использовать:

- <u>code-server</u>
- Git для отслеживания изменений

5.5. WordPress Playground (опционально)

Если вы хотите быстро протестировать внешний вид и базовые функции WordPress без установки на сервер — используйте https://playground.wordpress.net. Это полезно для обучения и предварительного дизайна.

5.6. Скрипт удаления WordPress-сайта (uninstall_wp.sh)

Для удаления установленного сайта предусмотрен скрипт uninstall_wp.sh из того же репозитория.

Что делает скрипт:

- отключает виртуальный хост Apache;
- удаляет конфиги и директорию сайта;
- удаляет базу данных и пользователя MySQL;
- очищает /etc/hosts и лог-файл установок;
- перезапускает Apache.

Пример использования:

chmod +x uninstall_wp.sh
sudo ./uninstall_wp.sh

Скрипт запросит имя сайта (например, my_site.local) и подтвердит удаление. Все действия выполняются безвозвратно, поэтому убедитесь, что данные больше не нужны.