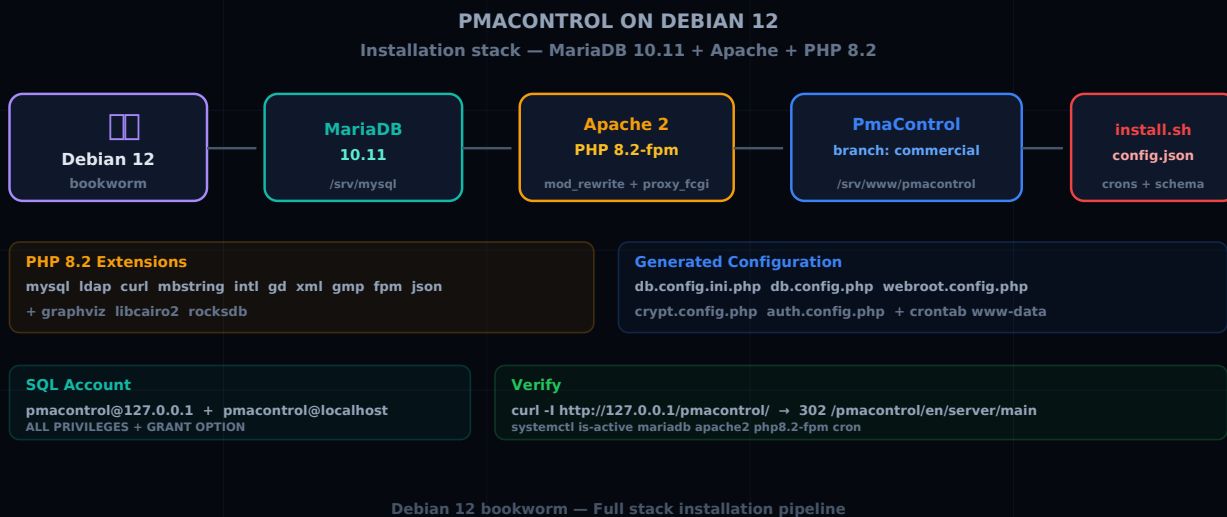


Установка PmaControl на Debian 12

Aurélien LEQUOY · April 12, 2026

PMACONTROL DEBIAN INSTALLATION MARIADB APACHE PHP



Цель

Эта статья объясняет, как установить PmaControl на Debian 12, опираясь на официальные исходники проекта, и учитывая важный практический момент:

- wiki Debian 12 ссылается на `install/debian12.sh`
- этот скрипт клонирует репозиторий `PmaControl` по умолчанию
- если вы хотите именно ветку `commercial`, необходимо адаптировать этап клонирования или выполнить `checkout` сразу после

Идея в том, чтобы сохранить официальную логику проекта, но сделать её пригодной для чистой установки на актуальной Debian 12.

Использованные источники:

- wiki GitHub `PmaControl/PmaControl`
- репозиторий `PmaControl/PmaControl`, ветка `commercial`
- скрипт `install/debian12.sh`

- корневой скрипт `install.sh`
- документация ветки `commercial`

Что делает официальная установка

Согласно wiki Debian 12, стандартная установка:

```
apt-get install curl
curl -o install-pmacontrol
https://raw.githubusercontent.com/PmaControl/PmaControl/master/install/debian12.sh
chmod +x install-pmacontrol
./install-pmacontrol
```

Скрипт `install/debian12.sh` ветки `commercial` затем выполняет следующие основные шаги:

1. обновляет систему
2. устанавливает базовые инструменты
3. устанавливает MariaDB `10.11`
4. устанавливает Apache, PHP `8.2`, Composer, Graphviz и некоторые зависимости
5. клонирует репозиторий PmaControl в `/srv/www/pmacontrol`
6. выполняет `composer install` от имени `www-data`
7. создаёт локальную учётную запись MySQL `pmacontrol@127.0.0.1`
8. генерирует временный `config.json`
9. запускает `./install.sh -c /tmp/config.json`
10. выводит сгенерированные учётные данные

Версии и компоненты, обнаруженные в исходниках

Скрипт Debian 12 ветки `commercial` рассчитан на следующий стек:

- MariaDB `10.11`
- Apache 2
- PHP `8.2`
- Composer
- Graphviz

- Плагин RocksDB для MariaDB

Явно установленные PHP-пакеты:

- `php8.2`, `php8.2-mysql`, `php8.2-ldap`, `php-json`, `php8.2-curl`
- `php8.2-cli`, `php8.2-mbstring`, `php8.2-intl`, `php8.2-fpm`
- `libapache2-mod-php8.2`, `php8.2-gd`, `php8.2-xml`, `php8.2-gmp`

Рекомендуемые предварительные требования

Перед запуском установки подготовьте:

- чистую Debian 12
- root-доступ
- работающий DNS и исходящую сеть
- минимум 4 ГБ оперативной памяти
- достаточно места на диске для MariaDB, `/srv/www/pmacontrol` и зависимостей Composer

Также рекомендую подготовить:

- корректный hostname
- статический IP
- согласованную timezone
- выделенное хранилище для MariaDB, если машина будет мониторить множество серверов

Важная информация о ветке `commercial`

Wiki Debian 12 минималистичен. Он ссылается на скрипт `install/debian12.sh`, но этот скрипт просто клонирует основной репозиторий, не выполняя явно:

```
git checkout commercial
```

Если ваша цель именно ветка `commercial`, лучше сделать так:

```
apt-get update
apt-get install -y curl git
```

```
cd /tmp
curl -o install-pmacontrol
https://raw.githubusercontent.com/PmaControl/PmaControl/commercial/install/debian12.sh
chmod +x install-pmacontrol
```

Затем перед этапом `composer install` замените часть клонирования на:

```
mkdir -p /srv/www
cd /srv/www
git clone --branch commercial --single-branch https://github.com/PmaControl/PmaControl.git
pmacontrol
cd /srv/www/pmacontrol
```

Эта адаптация — самый чистый вариант, если вы хотите установить именно код `commercial`.

Рекомендуемая процедура

1. Подготовка Debian 12

```
apt-get update
apt-get -y upgrade
apt-get install -y curl git sudo lsb-release unzip zip wget gnupg gnupg2 net-tools dnsutils
jq bc composer cron
timedatectl set-timezone Europe/Paris
```

Почему `cron` указан явно: `install.sh` использует `crontab`. На минимальной Debian-машине отсутствие `cron` может сломать завершение установки. Этот момент не рассматривается явно в `install/debian12.sh`, но следует из логики `install.sh`.

2. Установка MariaDB 10.11

Официальный скрипт Debian 12 настраивает репозиторий MariaDB, затем вызывает

```
Toolkit/install-mariadb.sh :
```

```
cd /tmp
git clone https://github.com/PmaControl/Toolkit.git
cd Toolkit
chmod +x install-mariadb.sh
```

```
curl -Ls https://r.mariadb.com/downloads/mariadb_repo_setup | bash -s -- --mariadb-server-version="mariadb-10.11"
./install-mariadb.sh -v 10.11 -p '<пароль_root_sql>' -d /srv/mysql -r
```

Важные моменты:

- datadir расположен в `/srv/mysql`
- скрипт использует MariaDB, а не MySQL
- затем сервис перезапускается

3. Установка Apache, PHP 8.2 и зависимостей

Также согласно `install/debian12.sh` :

```
apt-get install -y \
  php8.2 apache2 php8.2-mysql php8.2-ldap php-json php8.2-curl \
  php8.2-cli php8.2-mbstring php8.2-intl php8.2-fpm \
  libapache2-mod-php8.2 php8.2-gd php8.2-xml php8.2-gmp

apt-get install -y graphviz libcairo2 mariadb-plugin-rocksdb
```

Затем:

```
mysql -e "INSTALL SONAME 'ha_rocksdb'"
a2enmod proxy_fcgi setenvif
a2enconf php8.2-fpm
a2enmod rewrite
```

Настройка timezone PHP:

```
sed -i 's#;date.timezone =#date.timezone = Europe/Paris#g' /etc/php/8.2/fpm/php.ini
sed -i 's#;date.timezone =#date.timezone = Europe/Paris#g' /etc/php/8.2/apache2/php.ini
sed -i 's#;date.timezone =#date.timezone = Europe/Paris#g' /etc/php/8.2/cli/php.ini
```

4. Адаптация Apache к `/srv/www`

Скрипт Debian 12 заменяет стандартный webroot Apache на `/srv/www` :

```
sed -i 's#/var/www#/srv/www#g' /etc/apache2/apache2.conf
sed -i 's#/var/www/html#/srv/www#g' /etc/apache2/sites-enabled/000-default.conf
awk '/AllowOverride/ && ++i==3 {sub(/None/, "All")}1' /etc/apache2/apache2.conf > /tmp/xfgh
```

```
mv /tmp/xfgh /etc/apache2/apache2.conf
```

Затем:

```
mkdir -p /srv/www  
systemctl restart apache2
```

Ключевой момент: PmaControl ожидает размещения в `/srv/www/pmacontrol`, и webroot, генерируемый проектом, будет `/pmacontrol/`.

5. Клонирование PmaControl, ветка `commercial`

```
mkdir -p /srv/www  
cd /srv/www  
git clone --branch commercial --single-branch https://github.com/PmaControl/PmaControl.git  
pmacontrol  
cd /srv/www/pmacontrol  
chown -R www-data:www-data /srv/www/pmacontrol  
chown -R www-data:www-data /var/www  
sudo -u www-data composer install
```

Почему я предпочитаю именно эту последовательность:

- она соответствует нужной ветке
- она избегает установки другого состояния репозитория по умолчанию
- она остаётся совместимой с потоком, предусмотренным `install.sh`

6. Создание локальной учётной записи MySQL для PmaControl

Официальный скрипт создаёт:

```
GRANT ALL ON *.* TO pmacontrol@'127.0.0.1' IDENTIFIED BY '<пароль>' WITH GRANT OPTION;
```

Рекомендую также добавить `localhost`, чтобы избежать сюрпризов в зависимости от способа подключения:

```
CREATE OR REPLACE USER 'pmacontrol'@'127.0.0.1' IDENTIFIED BY '<пароль>';  
CREATE OR REPLACE USER 'pmacontrol'@'localhost' IDENTIFIED BY '<пароль>';  
GRANT ALL PRIVILEGES ON *.* TO 'pmacontrol'@'127.0.0.1' WITH GRANT OPTION;  
GRANT ALL PRIVILEGES ON *.* TO 'pmacontrol'@'localhost' WITH GRANT OPTION;
```

```
FLUSH PRIVILEGES;
```

7. Подготовка файла `config.json`

Скрипт `install/debian12.sh` генерирует временный `config.json`. Его структура важна, поскольку `install.sh` использует его для определения базы данных, создания организации, учётной записи администратора, учётной записи веб-сервиса и настройки `webroot`.

Рабочий пример:

```
{
  "mysql": {
    "ip": "127.0.0.1",
    "port": 3306,
    "user": "pmacontrol",
    "password": "CHANGE_ME_DB_PASSWORD",
    "database": "pmacontrol"
  },
  "organization": ["68Koncept"],
  "webroot": "/pmacontrol/",
  "ldap": { "enabled": false },
  "user": {
    "Member": null,
    "Administrator": null,
    "Super administrator": [{
      "email": "admin@example.net",
      "firstname": "Admin",
      "lastname": "PmaControl",
      "country": "France",
      "city": "Paris",
      "login": "admin",
      "password": "CHANGE_ME_ADMIN_PASSWORD"
    }]
  },
  "webservice": [{
    "user": "webservice",
    "host": "%",
    "password": "CHANGE_ME_WEBSERVICE_PASSWORD",
    "organization": "68Koncept"
  }]
}
```

```
}
```

Рекомендую не использовать блок `ssh` как есть из некоторых исторических скриптов, за исключением случаев, когда это явно необходимо и ключ контролируется.

8. Запуск установки приложения

Из клонированного репозитория:

```
cd /srv/www/pmacontrol  
./install.sh -c /tmp/config.json
```

Согласно `install.sh`, эта команда затем выполняет:

1. генерацию `configuration/webroot.config.php`
2. инициализацию базы данных
3. создание организации
4. создание суперадминистратора
5. интеграцию LDAP, если активирована
6. создание учётной записи веб-сервиса
7. создание таблиц TS
8. обновление списка серверов
9. генерацию модели и кэшей
10. установку системных cron-задач

9. Что изменяет `install.sh`

Корневой скрипт важен, поскольку он не ограничивается «установкой». Он:

- копирует файлы из `config_sample/` в `configuration/`
- генерирует `configuration/db.config.ini.php`
- генерирует `configuration/db.config.php`
- генерирует `configuration/webroot.config.php`
- устанавливает права на `tmp/` и `data/`
- устанавливает crontab для `www-data`
- устанавливает crontab для `root`

- запускает `composer install` при необходимости

Примеры добавляемых cron-задач:

```
* * * * * cd /srv/www/pmacontrol && ./glial agent check_daemon
05 */4 * * * cd /srv/www/pmacontrol && ./glial control service
* * * * * cd /srv/www/pmacontrol/script && ./monitor_mysql.sh
```

10. Проверка корректности установки

Сервисы:

```
systemctl is-active mariadb
systemctl is-active apache2
systemctl is-active php8.2-fpm
systemctl is-active cron
```

Версии:

```
php -v
mysql -Nse "SELECT VERSION()"
apache2 -v
```

HTTP:

```
curl -I http://127.0.0.1/pmacontrol/
```

Вы должны получить ответ от фронт-контроллера, часто с редиректом на:

```
/pmacontrol/en/server/main
```

Ожидаемые файлы конфигурации:

- `configuration/db.config.ini.php`
- `configuration/db.config.php`
- `configuration/webroot.config.php`
- `configuration/crypt.config.php`
- `configuration/auth.config.php`

11. Проверка конфигурации базы данных

Контроллер установки записывает файл такого типа в `configuration/db.config.ini.php` :

```
[pmacontrol]
driver=mysql
hostname=127.0.0.1
user=pmacontrol
password='...'
crypted='1'
database=pmacontrol
ssl=0
```

Этот момент важен: имя подключения, используемое фреймворком — `pmacontrol` . Если этот файл пуст или повреждён, приложение не запустится корректно.

12. Важные замечания

1. Wiki Debian 12 намеренно очень краток. Он даёт точку входа, а не описание всех компонентов. Поэтому нужно читать скрипт `install/debian12.sh` и `install.sh` , чтобы понять, что реально устанавливается.

2. Ветка `commercial` не гарантирована автоматически через сокращённую ссылку `wiki`. Если вы хотите ветку `commercial` , клонируйте её явно.

3. `cron` должен присутствовать. Это практическое требование `install.sh` . Я считаю его реальным предварительным требованием, даже если скрипт Debian 12 не устанавливает его явно.

4. Apache переписывается на `/srv/www` . Если на машине уже есть существующий хостинг, это изменение потенциально конфликтно.

5. Учётная запись SQL `pmacontrol` получает `ALL PRIVILEGES` . Это поведение скрипта. Если вы хотите ужесточить позже, сделайте это после установки, когда будете знать точный периметр потребностей приложения.

Компактная процедура

Короткая и чистая версия для быстрой установки:

```
apt-get update && apt-get -y upgrade
apt-get install -y curl git sudo composer cron

cd /tmp
git clone https://github.com/PmaControl/Toolkit.git
cd Toolkit
curl -Ls https://r.mariadb.com/downloads/mariadb_repo_setup | bash -s -- --mariadb-server-
version="mariadb-10.11"
./install-mariadb.sh -v 10.11 -p 'RootDbStrongPassword' -d /srv/mysql -r

apt-get install -y \
  apache2 php8.2 php8.2-fpm libapache2-mod-php8.2 \
  php8.2-mysql php8.2-ldap php8.2-curl php8.2-cli \
  php8.2-mbstring php8.2-intl php8.2-gd php8.2-xml php8.2-gmp \
  graphviz libcairo2 mariadb-plugin-rocksdb

mysql -e "INSTALL SONAME 'ha_rocksdb'"
a2enmod proxy_fcgi setenvif rewrite
a2enconf php8.2-fpm

sed -i 's#/var/www#/srv/www#g' /etc/apache2/apache2.conf
sed -i 's#/var/www/html#/srv/www#g' /etc/apache2/sites-enabled/000-default.conf
mkdir -p /srv/www
systemctl restart apache2

cd /srv/www
git clone --branch commercial --single-branch https://github.com/PmaControl/PmaControl.git
pmacontrol
cd /srv/www/pmacontrol
chown -R www-data:www-data /srv/www/pmacontrol
sudo -u www-data composer install

mysql <<'SQL'
CREATE OR REPLACE USER 'pmacontrol'@'127.0.0.1' IDENTIFIED BY 'ChangeMeDbPassword';
CREATE OR REPLACE USER 'pmacontrol'@'localhost' IDENTIFIED BY 'ChangeMeDbPassword';
GRANT ALL PRIVILEGES ON *.* TO 'pmacontrol'@'127.0.0.1' WITH GRANT OPTION;
GRANT ALL PRIVILEGES ON *.* TO 'pmacontrol'@'localhost' WITH GRANT OPTION;
FLUSH PRIVILEGES;
SQL

cat >/tmp/config.json <<'JSON'
```

```
{
  "mysql": {
    "ip": "127.0.0.1",
    "port": 3306,
    "user": "pmacontrol",
    "password": "ChangeMeDbPassword",
    "database": "pmacontrol"
  },
  "organization": ["68Koncept"],
  "webroot": "/pmacontrol/",
  "ldap": { "enabled": false },
  "user": {
    "Member": null,
    "Administrator": null,
    "Super administrator": [{
      "email": "admin@example.net",
      "firstname": "Admin",
      "lastname": "PmaControl",
      "country": "France",
      "city": "Paris",
      "login": "admin",
      "password": "ChangeMeAdminPassword"
    }]
  },
  "webservice": [{
    "user": "webservice",
    "host": "%",
    "password": "ChangeMeWebservicePassword",
    "organization": "68Koncept"
  }]
}
JSON
```

```
./install.sh -c /tmp/config.json
```

Заключение

Для Debian 12 официальная документация PmaControl достаточно, если читать вместе wiki Debian 12, `install/debian12.sh`, `install.sh` и документацию ветки `commercial`.

Реальная логика установки проста:

1. MariaDB `10.11`
2. Apache + PHP `8.2`
3. репозиторий PmaControl в `/srv/www/pmacontrol`
4. `composer install`
5. локальная учётная запись SQL `pmacontrol`
6. `config.json`
7. `./install.sh -c ...`

Если целью является ветка `commercial`, самый важный момент — явно принудительно клонировать эту ветку вместо слепого следования сокращённой ссылке `wiki`.

Ссылки

- [Wiki — New install on Debian 12](#)
- [Wiki — Install](#)
- [Dépôt commercial — install/debian12.sh](#)
- [Dépôt commercial — install.sh](#)