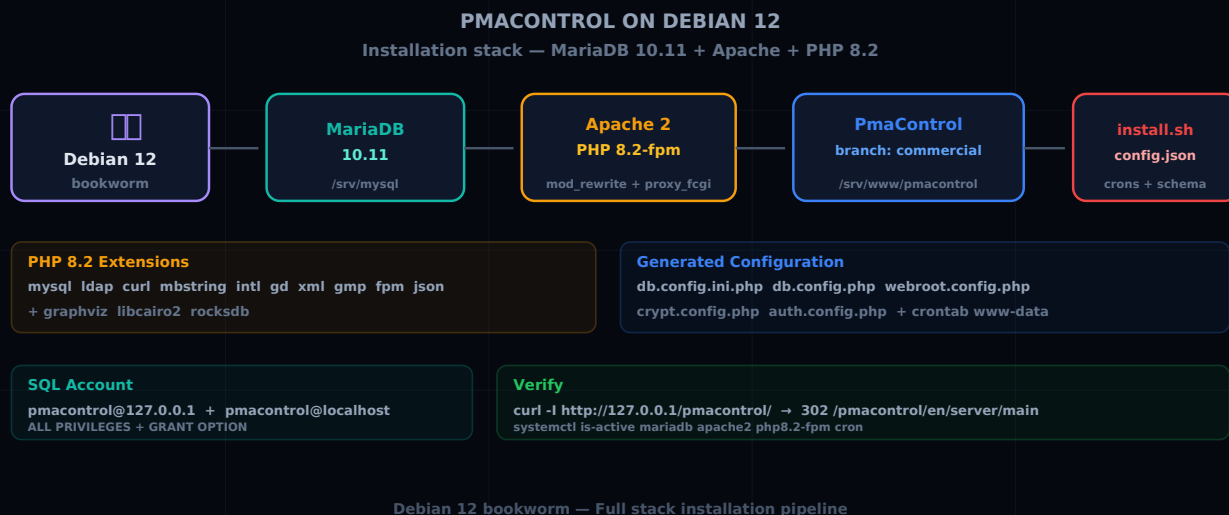


Instalacja PmaControl na Debian 12

Aurélien LEQUOY · April 12, 2026

PMACONTROL DEBIAN INSTALLATION MARIADB APACHE PHP



Cel

Ten artykuł wyjaśnia jak zainstalować PmaControl na Debian 12, korzystając z oficjalnych źródeł projektu, uwzględniając ważny aspekt praktyczny:

- wiki Debian 12 wskazuje na `install/debian12.sh`
- ten skrypt klonuje repozytorium `PmaControl` domyślnie
- jeśli chcesz jawnie gałąź `commercial`, musisz dostosować etap klonowania lub wykonać `checkout` zaraz po

Chodzi więc o zachowanie oficjalnej logiki projektu, ale uczynienie jej poprawnie wykorzystywalną dla aktualnej instalacji Debian 12.

Wykorzystane źródła:

- wiki GitHub `PmaControl/PmaControl`
- repozytorium `PmaControl/PmaControl`, gałąź `commercial`
- skrypt `install/debian12.sh`

- skrypt główny `install.sh`
- dokumentacja gałęzi `commercial`

Co robi oficjalna instalacja

Według wiki Debian 12, standardowa instalacja to:

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

Skrypt `install/debian12.sh` z gałęzi `commercial` wykonuje następnie głównie:

1. aktualizację systemu
2. instalację narzędzi podstawowych
3. instalację MariaDB `10.11`
4. instalację Apache, PHP `8.2`, Composer, Graphviz i kilku zależności
5. klonowanie repozytorium PmaControl do `/srv/www/pmacontrol`
6. wykonanie `composer install` jako `www-data`
7. utworzenie lokalnego konta MySQL `pmacontrol@127.0.0.1`
8. wygenerowanie tymczasowego `config.json`
9. uruchomienie `./install.sh -c /tmp/config.json`
10. wyświetlenie wygenerowanych danych uwierzytelniających

Wersje i komponenty obserwowane w źródłach

Skrypt Debian 12 z gałęzi `commercial` celuje w następującą bazę:

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

- plugin RocksDB dla MariaDB

Jawnie instalowane pakiety PHP:

- `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`

Zalecane wymagania wstępne

Przed uruchomieniem instalacji przewidź:

- czysty Debian 12
- dostęp root
- działający DNS i połączenie wychodzące
- co najmniej 4 GB RAM
- wystarczającą ilość miejsca na dysku dla MariaDB, `/srv/www/pmacontrol` i zależności Composer

Zalecam również przygotowanie:

- poprawnej nazwy hosta
- stałego adresu IP
- spójnej strefy czasowej
- dedykowanego magazynu dla MariaDB, jeśli maszyna ma nadzorować wiele serwerów

Ważna uwaga dotycząca gałęzi `commercial`

Wiki Debian 12 jest minimalistyczne. Odsyła do skryptu `install/debian12.sh`, ale ten skrypt po prostu klonuje główne repozytorium bez jawnego wykonania:

```
git checkout commercial
```

Jeśli Twoim celem jest gałąź `commercial`, wykonaj raczej:

```
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
```

Następnie, przed fazą `composer install`, zastąp część klonowania:

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

Ta adaptacja jest najczystsza, jeśli faktycznie chcesz zainstalować kod `commercial`.

Zalecana procedura

1. Przygotowanie 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
```

Dlaczego jawnie `cron`: `install.sh` używa `crontab`. Na minimalnej maszynie Debian brak `cron` może zepsuć zakończenie instalacji. Ta kwestia nie jest jawnie uwzględniona w `install/debian12.sh`, ale wynika z logiki `install.sh`.

2. Instalacja MariaDB 10.11

Oficjalny skrypt Debian 12 konfiguruje repozytorium MariaDB, a następnie wywołuje

`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 '<hasło_root_sql>' -d /srv/mysql -r
```

Uwagi:

- datadir jest ustawiony na `/srv/mysql`
- skrypt używa MariaDB, nie MySQL
- usługa jest następnie restartowana

3. Instalacja Apache, PHP 8.2 i zależności

Zawsze według `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
```

Następnie:

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

Ustawienie strefy czasowej 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. Adaptacja Apache do `/srv/www`

Skrypt Debian 12 zastępuje standardowy webroot Apache katalogiem `/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")}' /etc/apache2/apache2.conf > /tmp/xfgh  
mv /tmp/xfgh /etc/apache2/apache2.conf
```

Następnie:

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

Kluczowy punkt: PmaControl oczekuje, że będzie serwowany z `/srv/www/pmacontrol`, a webroot generowany przez projekt będzie `/pmacontrol/`.

5. Klonowanie PmaControl z gałęzi `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
```

Dlaczego preferuję tę sekwencję:

- odpowiada požądanej gałęzi
- unika instalacji innego stanu repozytorium domyślnie
- pozostaje kompatybilna z przepływem przewidzianym przez `install.sh`

6. Utworzenie lokalnego konta MySQL PmaControl

Oficjalny skrypt tworzy:

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

Zalecam dodanie również `localhost`, aby uniknąć niespodzianek w zależności od trybu połączenia:

```
CREATE OR REPLACE USER 'pmacontrol'@'127.0.0.1' IDENTIFIED BY '<hasło>';
CREATE OR REPLACE USER 'pmacontrol'@'localhost' IDENTIFIED BY '<hasło>';
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. Przygotowanie pliku `config.json`

Skrypt `install/debian12.sh` generuje tymczasowy `config.json`. Jego struktura jest ważna, ponieważ `install.sh` używa go do zdefiniowania bazy danych, utworzenia organizacji, konta administratora, konta webservice i konfiguracji webroota.

Przykład do wykorzystania:

```
{
  "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"
  }]
}
```

Radzę nie kopiować bloku `ssh` wstrzykiwanego w niektórych historycznych skryptach, chyba że jest to wyraźnie potrzebne i klucz jest kontrolowany.

8. Uruchomienie instalacji aplikacyjnej

Z sklonowanego repozytorium:

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

Zgodnie z `install.sh`, ta komenda wykonuje następujące:

1. generację `configuration/webroot.config.php`
2. inicjalizację bazy danych
3. utworzenie organizacji
4. utworzenie super administratora
5. integrację LDAP, jeśli aktywowana
6. utworzenie konta webservice
7. utworzenie tabel TS
8. aktualizację listy serwerów
9. generację modelu i cache'y
10. ustawienie cronów systemowych

9. Zrozumienie tego, co `install.sh` modyfikuje

Główny skrypt jest ważny, ponieważ nie ogranicza się do "instalowania". On:

- kopiuje pliki z `config_sample/` do `configuration/`
- generuje `configuration/db.config.ini.php`
- generuje `configuration/db.config.php`
- generuje `configuration/webroot.config.php`
- ustawia prawa na `tmp/` i `data/`
- instaluje crontab dla `www-data`
- instaluje crontab dla `root`
- uruchamia `composer install`, jeśli to konieczne

Przykłady dodawanych zadań cron:

```
* * * * * 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. Weryfikacja poprawności instalacji

Usługi:

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

Wersje:

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

HTTP:

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

Powinieneś otrzymać odpowiadający front controller, często z przekierowaniem do:

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

Oczekiwane pliki konfiguracyjne:

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

11. Weryfikacja konfiguracji bazy danych

Kontroler instalacji zapisuje plik tego typu w `configuration/db.config.ini.php` :

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

To jest ważne: nazwa połączenia używana przez framework to `pmacontrol`. Jeśli ten plik jest pusty lub uszkodzony, aplikacja nie uruchamia się poprawnie.

12. Punkty czujności

1. Wiki Debian 12 jest celowo bardzo krótkie. Podaje punkt wejścia, nie szczegóły wszystkich komponentów. Trzeba więc czytać skrypt `install/debian12.sh` i `install.sh`, aby zrozumieć, co jest faktycznie instalowane.

2. Gałąź `commercial` nie jest automatycznie gwarantowana przez skrót wiki. Jeśli chcesz gałąź `commercial`, klonuj ją jawnie.

3. `cron` musi być obecny. To praktyczne wymaganie `install.sh`. Uważam go za rzeczywisty warunek wstępny, nawet jeśli skrypt Debian 12 nie instaluje go jawnie.

4. Apache jest przekonfigurowany na `/srv/www`. Jeśli maszyna ma już istniejący hosting, ta zmiana jest potencjalnie inwazyjna.

5. Konto SQL `pmacontrol` otrzymuje `ALL PRIVILEGES`. Takie jest zachowanie skryptu. Jeśli chcesz później zahartować, zrób to po instalacji, gdy znasz dokładny zakres potrzeb aplikacji.

Procedura zwarta

Krótką i czystą wersją do szybkiej instalacji:

```
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
```

Podsumowanie

Dla Debian 12 oficjalna baza dokumentacyjna PmaControl jest wystarczająca, jeśli czyta się razem wiki Debian 12, `install/debian12.sh`, `install.sh` i dokumentację gałęzi `commercial`.

Prawdziwa logika instalacji jest prosta:

1. MariaDB `10.11`
2. Apache + PHP `8.2`
3. repozytorium PmaControl pod `/srv/www/pmacontrol`
4. `composer install`

5. lokalne konto SQL `pmacontrol`

6. `config.json`

7. `./install.sh -c ...`

Jeśli celem jest gałąź `commercial`, najważniejszym punktem jest jawne wymuszenie klonowania tej gałęzi zamiast ślepego podążania za skrótem wiki.

Odniesienia

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