

# Installer PmaControl sur Debian 12

Aurélien LEQUOY · 12 avril 2026

PMACONTROL DEBIAN INSTALLATION MARIADB APACHE PHP



## Objectif

Cet article explique comment installer PmaControl sur Debian 12 en s'appuyant sur les sources officielles du projet, tout en tenant compte d'un point pratique important :

- le wiki Debian 12 pointe vers `install/debian12.sh`
- ce script clone le dépôt `PmaControl` par défaut
- si vous voulez explicitement la branche `commercial`, il faut adapter l'étape de clonage ou faire le `checkout` juste après

L'idée est donc de conserver la logique officielle du projet, mais de la rendre exploitable proprement pour une installation Debian 12 actuelle.

### Sources utilisées :

- wiki GitHub `PmaControl/PmaControl`
- dépôt `PmaControl/PmaControl`, branche `commercial`
- script `install/debian12.sh`

- script racine `install.sh`
- documentation de la branche `commercial`

## Ce que fait l'installation officielle

---

D'après le wiki Debian 12, l'installation standard est :

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

Le script `install/debian12.sh` de la branche `commercial` fait ensuite principalement ceci :

1. met à jour le système
2. installe les outils de base
3. installe MariaDB `10.11`
4. installe Apache, PHP `8.2`, Composer, Graphviz et quelques dépendances
5. clone le dépôt PmaControl dans `/srv/www/pmacontrol`
6. exécute `composer install` en tant que `www-data`
7. crée un compte MySQL local `pmacontrol@127.0.0.1`
8. génère un `config.json` temporaire
9. lance `./install.sh -c /tmp/config.json`
10. affiche les identifiants générés

## Versions et composants observés dans les sources

---

Le script Debian 12 de la branche `commercial` vise ce socle :

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

- RocksDB plugin pour MariaDB

Paquets PHP explicitement installés :

- `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`

## Pré-requis recommandés

---

Avant de lancer l'installation, prévoyez :

- une Debian 12 propre
- accès root
- DNS et réseau sortant fonctionnels
- au moins 4 Go de RAM
- suffisamment d'espace disque pour MariaDB, `/srv/www/pmacontrol` et les dépendances Composer

Je recommande aussi de préparer :

- un hostname correct
- une IP fixe
- une zone horaire cohérente
- un stockage dédié pour MariaDB si la machine doit superviser beaucoup de serveurs

## Point important sur la branche `commercial`

---

Le wiki Debian 12 est minimaliste. Il renvoie vers le script `install/debian12.sh` , mais ce script clone simplement le dépôt principal sans faire explicitement :

```
git checkout commercial
```

Si votre cible est bien la branche `commercial` , faites plutôt :

```
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
```

Puis, avant la phase `composer install`, remplacez la partie clonage par :

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

Cette adaptation est la plus propre si vous voulez réellement installer le code `commercial`.

## Procédure recommandée

---

### 1. Préparer 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
```

Pourquoi `cron` explicitement : `install.sh` utilise `crontab`. Sur une machine Debian minimale, l'absence de `cron` peut casser la fin d'installation. Ce point n'est pas explicitement traité dans `install/debian12.sh`, mais il ressort de la logique de `install.sh`.

### 2. Installer MariaDB 10.11

Le script officiel Debian 12 configure le dépôt MariaDB puis appelle `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 '<mot_de_passe_root_sql>' -d /srv/mysql -r
```

Points à noter :

- le datadir est prévu sur `/srv/mysql`
- le script utilise MariaDB, pas MySQL
- le service est ensuite redémarré

### 3. Installer Apache, PHP 8.2 et dépendances

Toujours selon `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
```

Puis :

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

Réglage 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. Adapter Apache à `/srv/www`

Le script Debian 12 remplace le webroot Apache standard par `/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
```

Puis :

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

Le point clé ici : PmaControl attend d'être servi sous `/srv/www/pmacontrol` et le webroot généré par le projet sera `/pmacontrol/`.

## 5. Cloner PmaControl en branche `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
```

Pourquoi je préfère cette séquence :

- elle colle à la branche voulue
- elle évite d'installer un autre état du dépôt par défaut
- elle reste compatible avec le flux prévu par `install.sh`

## 6. Créer le compte MySQL local de PmaControl

Le script officiel crée :

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

Je recommande d'ajouter aussi `localhost` pour éviter les surprises selon le mode de connexion :

```
CREATE OR REPLACE USER 'pmacontrol'@'127.0.0.1' IDENTIFIED BY '<mot_de_passe>';
CREATE OR REPLACE USER 'pmacontrol'@'localhost' IDENTIFIED BY '<mot_de_passe>';
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. Préparer le fichier `config.json`

Le script `install/debian12.sh` génère un `config.json` temporaire. Sa structure est importante car `install.sh` l'utilise pour définir la base, créer l'organisation, le compte administrateur, le compte webservice et configurer le webroot.

Exemple exploitable :

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

Je vous conseille de ne pas reprendre tel quel le bloc `ssh` injecté dans certains scripts historiques, sauf besoin explicite et clé maîtrisée.

## 8. Lancer l'installation applicative

Depuis le dépôt cloné :

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

D'après `install.sh`, cette commande fait ensuite :

1. génération de `configuration/webroot.config.php`
2. initialisation de la base
3. création de l'organisation
4. création du super administrateur
5. intégration LDAP si activée
6. création du compte webservice
7. création des tables TS
8. mise à jour de la liste des serveurs
9. génération du modèle et des caches
10. pose de crons système

## 9. Comprendre ce que `install.sh` modifie

Le script racine est important car il ne se limite pas à "installer". Il :

- copie les fichiers de `config_sample/` vers `configuration/`
- génère `configuration/db.config.ini.php`
- génère `configuration/db.config.php`
- génère `configuration/webroot.config.php`
- positionne les droits sur `tmp/` et `data/`
- installe une crontab pour `www-data`
- installe une crontab pour `root`
- lance `composer install` si nécessaire

Exemples de tâches cron ajoutées :

```
* * * * * 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. Vérifier que l'installation est saine

Services :

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

Versions :

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

HTTP :

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

Vous devez obtenir un front controller répondant, souvent avec une redirection vers :

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

Fichiers de config attendus :

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

## 11. Vérifier la configuration base de données

Le contrôleur d'installation écrit un fichier de ce type dans `configuration/db.config.ini.php` :

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

Ce point est important : le nom de connexion utilisé par le framework est `pmacontrol` . Si ce fichier est vide ou cassé, l'application ne démarre pas correctement.

## 12. Points de vigilance

**1. Le wiki Debian 12 est volontairement très court.** Il donne la porte d'entrée, pas le détail de tous les composants. Il faut donc lire le script `install/debian12.sh` et `install.sh` pour comprendre ce qui est réellement installé.

**2. La branche `commercial` n'est pas automatiquement garantie par le raccourci wiki.** Si vous voulez la branche `commercial` , clonez-la explicitement.

**3. `cron` doit être présent.** C'est une exigence pratique de `install.sh` . Je le considère comme un prérequis réel, même si le script Debian 12 ne l'installe pas explicitement.

**4. Apache est réécrit vers `/srv/www` .** Si la machine a déjà un hébergement existant, ce changement est potentiellement intrusif.

**5. Le compte SQL `pmacontrol` reçoit ALL PRIVILEGES .** C'est le comportement du script. Si vous voulez durcir ensuite, faites-le après l'installation, quand vous connaissez le périmètre exact des besoins de l'application.

## Procédure compacte

Version courte et propre pour une installation rapide :

```
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
```

## Conclusion

Pour Debian 12, la base documentaire officielle PmaControl est suffisante si on lit ensemble le wiki Debian 12, [install/debian12.sh](#), [install.sh](#) et la documentation de la branche [commercial](#).

La vraie logique d'installation est simple :

1. MariaDB [10.11](#)
2. Apache + PHP [8.2](#)

3. dépôt PmaControl sous `/srv/www/pmacontrol`
4. `composer install`
5. compte SQL local `pmacontrol`
6. `config.json`
7. `./install.sh -c ...`

Si la cible est la branche `commercial`, le point le plus important est de forcer explicitement le clonage de cette branche au lieu de suivre aveuglément le raccourci du wiki.

## Références

---

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