

# Automatyczne dodawanie serwerów MariaDB / MySQL do PmaControl za pomocą webserwisu

Aurélien LEQUOY · April 13, 2026

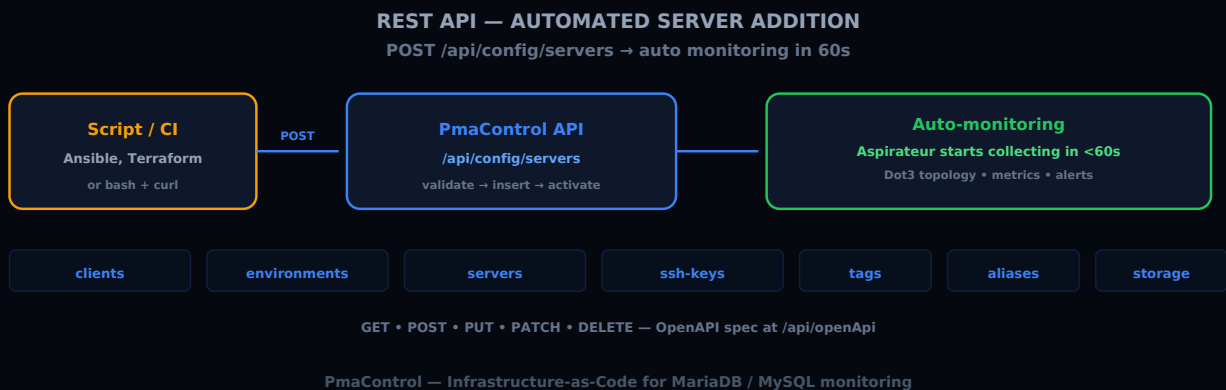
PMACONTROL

REST-API

AUTOMATION

DEVOPS

WEBSERVICE



## Dlaczego automatyzować dodawanie serwerów

Gdy nadzorujesz 5 serwerów MariaDB / MySQL, ręczne dodawanie ich do PmaControl jest akceptowalne. Przy 50 lub 200 serwerach już nie.

REST API PmaControl pozwala na industrializację tego procesu: skrypt, pipeline CI/CD lub narzędzie orkiestracji (Ansible, Terraform) może utworzyć nadzorowany serwer w jednym żądaniu HTTP.

## Wymagania wstępne

- Aktywne konto webserwisu w PmaControl (utworzone podczas instalacji przez `config.json`)
- URL twojej instancji PmaControl
- Docelowy klient i środowisko muszą istnieć (utwórz je najpierw przez API, jeśli to konieczne)

## Główny endpoint

```
POST /fr/api/config/servers
Content-Type: application/json
```

Uwierzytelnianie za pomocą konta webservice skonfigurowanego w

```
configuration/webservice.config.php .
```

## Payload do dodania serwera

```
{
  "id_client": 1,
  "id_environment": 1,
  "name": "prod-db-01",
  "display_name": "Production DB 01",
  "ip": "10.68.68.100",
  "hostname": "prod-db-01.internal",
  "login": "pmacontrol",
  "passwd": "SecurePassword123",
  "database": "information_schema",
  "port": 3306,
  "is_ssl": 0,
  "ssh_port": 22,
  "ssh_login": "pmacontrol",
  "is_sudo": 1,
  "is_root": 0,
  "is_monitored": 1,
  "is_proxy": 0,
  "is_vip": 0
}
```

## Ważne pola

Pole	Opis
<code>id_client</code>	Identyfikator klienta (organizacji)
<code>id_environment</code>	Identyfikator środowiska (Produkcja, Staging itp.)
<code>ip</code>	Adres IP dostępu do serwera MySQL
<code>port</code>	Port MySQL (domyślnie 3306)
<code>login / passwd</code>	Dane uwierzytelniające MySQL do zbierania metryk
<code>is_monitored</code>	1 = aktywny monitoring, 0 = wyłączony

Pole	Opis
is_proxy	1 jeśli to ProxySQL/MaxScale (testuje połączenie inaczej)
is_vip	1 jeśli to VIP/DNS (bez bezpośredniego połączenia, śledzenie przekierowań)

## Najpierw utwórz zależności

Przed dodaniem serwera klient i środowisko muszą istnieć.

### Tworzenie klienta

```
curl -s -X POST http://pmacontrol.local/fr/api/config/clients \  
-H "Content-Type: application/json" \  
-d '{"name": "68Koncept", "description": "Production infrastructure"}'
```

### Tworzenie środowiska

```
curl -s -X POST http://pmacontrol.local/fr/api/config/environments \  
-H "Content-Type: application/json" \  
-d '{"name": "Production", "description": "Live servers"}'
```

### Dodawanie klucza SSH

```
curl -s -X POST http://pmacontrol.local/fr/api/config/ssh-keys \  
-H "Content-Type: application/json" \  
-d '{"name": "deploy-key", "private_key": "-----BEGIN RSA PRIVATE KEY-----\n..."}'
```

## Kompletny skrypt masowego dodawania

```
#!/bin/bash  
PMAC="http://pmacontrol.local"  
CLIENT_ID=1  
ENV_ID=1  
  
SERVERS=(  
  "prod-db-01:10.68.68.100:3306"
```

```

"prod-db-02:10.68.68.101:3306"
"prod-db-03:10.68.68.102:3306"
"prod-proxy-01:10.68.68.200:6033"
)

for entry in "${SERVERS[@]}; do
IFS=: read -r name ip port <<< "$entry"
is_proxy=0
[[ "$name" == *proxy* ]] && is_proxy=1

curl -s -X POST "$PMAC/fr/api/config/servers" \
-H "Content-Type: application/json" \
-d "{
  \"id_client\": $CLIENT_ID,
  \"id_environment\": $ENV_ID,
  \"name\": \"$name\",
  \"display_name\": \"$name\",
  \"ip\": \"$ip\",
  \"port\": $port,
  \"login\": \"pmacontrol\",
  \"passwd\": \"PmacMonitor2026\",
  \"database\": \"information_schema\",
  \"is_monitored\": 1,
  \"is_proxy\": $is_proxy,
  \"ssh_port\": 22,
  \"ssh_login\": \"pmacontrol\"
}"
echo " → $name added"
done

```

## Wyświetlanie istniejących serwerów

```

curl -s http://pmacontrol.local/fr/api/config/servers | jq '.[] | {id, name, ip, is_monitored}'

```

## Aktualizacja serwera

```

PUT /fr/api/config/servers/{id}

```

```
curl -s -X PUT http://pmacontrol.local/fr/api/config/servers/5 \  
-H "Content-Type: application/json" \  
-d '{"is_monitored": 0}'
```

## Usuwanie serwera

Usunięcie jest **miękkim usunięciem** ( `is_deleted = 1` ). Serwer znika z interfejsu, ale pozostaje w bazie danych:

```
curl -s -X DELETE http://pmacontrol.local/fr/api/config/servers/5
```

## Tagi do organizacji serwerów

Tagi pozwalają kategoryzować serwery (datacenter, rola, wersja):

```
# Tworzenie tagu  
curl -s -X POST http://pmacontrol.local/fr/api/config/tags \  
-H "Content-Type: application/json" \  
-d '{"name": "dc-paris", "description": "Paris datacenter"}'  
  
# Przypisanie tagu do serwera (przez interfejs lub API)
```

## Weryfikacja po dodaniu

Po dodaniu PmaControl automatycznie rozpoczyna zbieranie metryk, jeśli `is_monitored = 1`.

Sprawdź w interfejsie:

1. Serwer pojawia się na liście `Server > Main`
2. Topologia Dot3 aktualizuje się w następnym cyklu
3. Pierwsze metryki pojawiają się w ciągu 60 sekund

## Specyfikacja OpenAPI

PmaControl udostępnia pełną specyfikację OpenAPI:

```
GET /fr/api/openApi
```

Można jej używać ze Swagger UI lub do automatycznego generowania klientów SDK.

## Podsumowanie

---

REST API PmaControl przekształca dodawanie serwerów z zadania ręcznego w operację skryptowalną i powtarzalną. W połączeniu z Ansible lub Terraform umożliwia podejście Infrastructure-as-Code do nadzoru MariaDB / MySQL.