

# Автоматическое добавление серверов MariaDB / MySQL в PmaControl через веб-сервис

Aurélien LEQUOY · April 13, 2026

PMACONTROL

REST-API

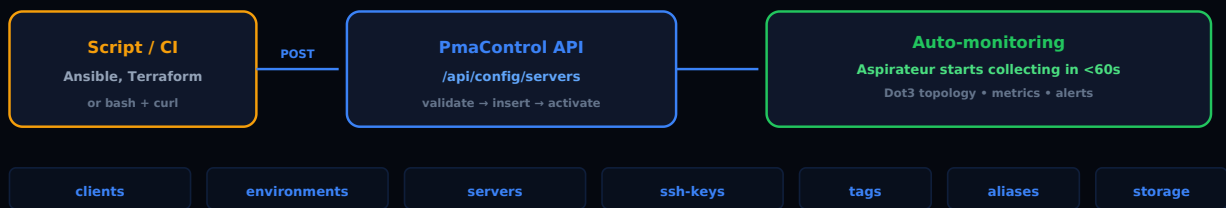
AUTOMATION

DEVOPS

WEBSERVICE

## REST API — AUTOMATED SERVER ADDITION

POST /api/config/servers → auto monitoring in 60s



GET • POST • PUT • PATCH • DELETE — OpenAPI spec at /api/openApi

PmaControl — Infrastructure-as-Code for MariaDB / MySQL monitoring

## Зачем автоматизировать добавление серверов

Когда вы мониторите 5 серверов MariaDB / MySQL, добавлять их вручную в PmaControl — вполне допустимо. При 50 или 200 серверах — уже нет.

REST API PmaControl позволяет индустриализировать этот процесс: скрипт, CI/CD-пайплайн или инструмент оркестрации (Ansible, Terraform) может создать наблюдаемый сервер одним HTTP-запросом.

## Предварительные требования

- Активная учётная запись веб-сервиса в PmaControl (создаётся при установке через `config.json`)
- URL вашего экземпляра PmaControl
- Целевой клиент и окружение должны существовать (при необходимости создайте их через API)

## Основной endpoint

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

Аутентификация через учётную запись веб-сервиса, настроенную в `configuration/webservice.config.php`.

## Тело запроса для добавления сервера

```
{
  "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
}
```

## Ключевые поля

Поле	Описание
<code>id_client</code>	Идентификатор клиента (организации)
<code>id_environment</code>	Идентификатор окружения (Production, Staging и т.д.)

Поле	Описание
ip	IP-адрес для подключения к серверу MySQL
port	Порт MySQL (по умолчанию 3306)
login / passwd	Учётные данные MySQL для сбора метрик
is_monitored	1 = мониторинг активен, 0 = отключен
is_proxy	1 если это ProxySQL/MaxScale (другой способ проверки соединения)
is_vip	1 если это VIP/DNS (без прямого подключения, отслеживание редиректов)

## Сначала создайте зависимости

---

Перед добавлением сервера клиент и окружение должны существовать.

### Создание клиента

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

### Создание окружения

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

### Добавление SSH-ключа

```
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..."}'
```

## Скрипт массового добавления

---

```

#!/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

```

## Просмотр существующих серверов

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

## Обновление сервера

```
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}'
```

## Удаление сервера

Удаление выполняется как **мягкое удаление** (`is_deleted = 1`). Сервер исчезает из интерфейса, но остаётся в базе:

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

## Теги для организации серверов

Теги позволяют категоризировать серверы (дата-центр, роль, версия):

```
curl -s -X POST http://pmacontrol.local/fr/api/config/tags \  
-H "Content-Type: application/json" \  
-d '{"name": "dc-paris", "description": "Paris datacenter"}'
```

## Проверка после добавления

После добавления PmaControl автоматически начинает сбор данных, если `is_monitored = 1`.

Проверьте в интерфейсе:

1. Сервер появился в списке `Server > Main`
2. Топология Dot3 обновится при следующем цикле
3. Первые метрики поступят в течение 60 секунд

## Спецификация OpenAPI

---

PmaControl предоставляет полную спецификацию OpenAPI:

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

Используйте с Swagger UI или для автоматической генерации SDK-клиентов.

## Заключение

---

REST API PmaControl превращает добавление серверов из ручной задачи в скриптуемую и воспроизводимую операцию. В сочетании с Ansible или Terraform это позволяет применять подход Infrastructure-as-Code к мониторингу MariaDB / MySQL.