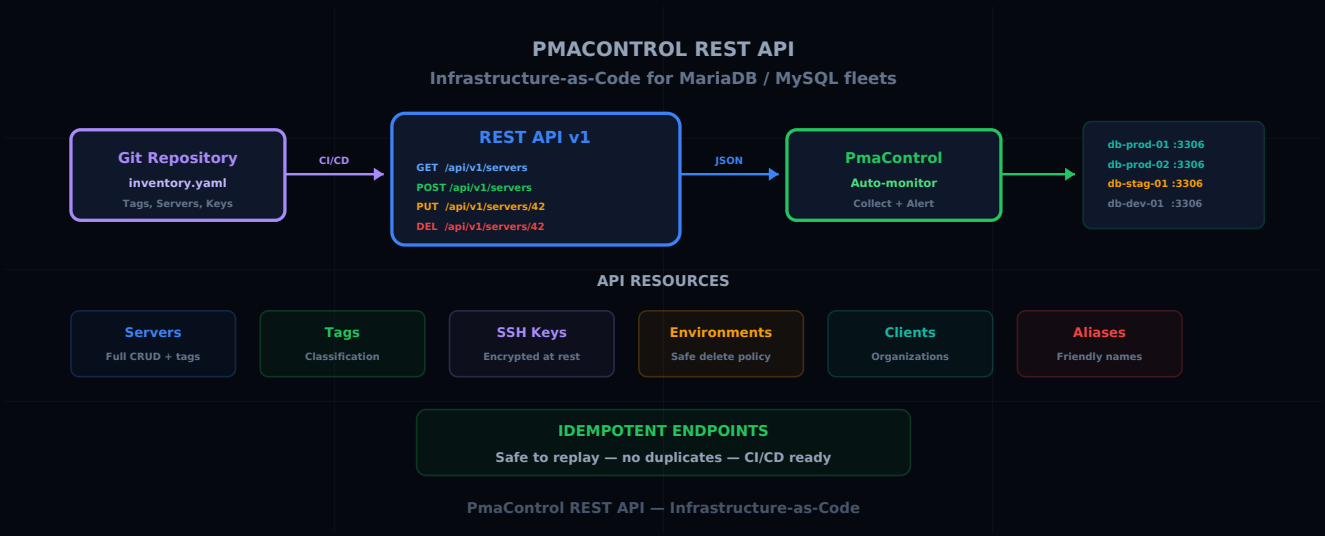
PmaControl REST API: Infrastructure-as-Code для MariaDB / MySQL

Aurélien LEQUOY · March 15, 2026

PMACONTROL REST-API INFRASTRUCTURE-AS-CODE DEVOPS AUTOMATION



Зачем REST API?

PmaControl начинался как инструмент с веб-интерфейсом. Сервер MariaDB / MySQL добавляли кликами по кнопкам, теги настраивали мышью, окружения управлялись через формы.

Это работает для 10 серверов. При 50 — уже болезненно. При 200 — невозможно.

REST API PmaControl превращает инструмент в **программируемую платформу**. Каждое действие, доступное в веб-интерфейсе, доступно через JSON-эндпоинт. Это открывает двери к Infrastructure-as-Code: описать свой инвентарь баз данных в файлах YAML или JSON и применять его автоматически.

Аутентификация

API использует аутентификацию через **пользователя веб-сервиса**. Это выделенная учётная запись PmaControl без доступа к веб-интерфейсу, чей токен служит ключом API.

```
curl -H "Authorization: Bearer <token>" \
     -H "Content-Type: application/json" \
     https://pmacontrol.example.com/api/v1/servers
```

Пользователь веб-сервиса наследует ACL своего профиля. Профиль «read-only» сможет только просматривать ресурсы. Профиль «admin» сможет создавать, изменять и удалять.

Ресурсы API

Теги

Теги — система классификации PmaControl. Каждый сервер может нести один или несколько тегов (production, staging, client-acme, dc-paris и т.д.).

```
# Список всех тегов
GET /api/v1/tags

# Создать тег
POST /api/v1/tags
{"name": "production", "color": "#22c55e"}

# Изменить тег
PUT /api/v1/tags/42
{"name": "prod", "color": "#22c55e"}

# Удалить тег
DELETE /api/v1/tags/42
```

Теги — базовый строительный блок организации. Они позволяют фильтровать дашборды, ограничивать область оповещений и ограничивать доступ по профилю.

Клиенты

Клиенты представляют организации или проекты, которым принадлежат серверы.

```
# Список клиентов
GET /api/v1/clients

# Создать клиента
```

```
POST /api/v1/clients
{"name": "Acme Corp", "contact_email": "dba@acme.com"}

# Изменить клиента
PUT /api/v1/clients/7
{"name": "Acme Corporation"}

# Удалить клиента
DELETE /api/v1/clients/7
```

Окружения

Окружения (production, staging, development и т.д.) имеют специальную политику удаления: **при удалении окружения его серверы не уничтожаются, а переназначаются** в окружение по умолчанию.

```
# Список окружений
GET /api/v1/environments

# Создать окружение
POST /api/v1/environments
{"name": "staging", "description": "Pre-production environment"}

# Удалить — серверы переназначаются
DELETE /api/v1/environments/3
# Ответ: {"reassigned_servers": 12, "target_environment": "default"}
```

Эта политика предотвращает случайное удаление серверов при реорганизации окружений.

Алиасы

Алиасы позволяют обращаться к серверу по читаемому имени вместо IP или внутреннего hostname.

```
# Список алиасов
GET /api/v1/aliases

# Создать алиас
POST /api/v1/aliases
```

```
{"alias": "db-writer-prod", "server_id": 42}
```

Зоны хранения

Зоны хранения представляют площадки хранения (дата-центры, облачные регионы и т.д.).

```
# Список зон хранения
GET /api/v1/storage-areas

# Создать зону хранения
POST /api/v1/storage-areas
{"name": "dc-paris-1", "provider": "OVH", "location": "Paris, France"}
```

SSH-ключи

PmaControl использует SSH-ключи для подключения к серверам и сбора метрик. API позволяет управлять жизненным циклом ключей.

```
# Список SSH-ключей
GET /api/v1/ssh-keys

# Создать SSH-ключ
POST /api/v1/ssh-keys
{
  "name": "pmacontrol-collector-2026",
  "public_key": "ssh-ed25519 AAAA...",
  "private_key": "-----BEGIN OPENSSH PRIVATE KEY-----\n..."
}

# Удалить ключ
DELETE /api/v1/ssh-keys/5
```

Замечание по безопасности: приватные ключи шифруются при хранении в базе PmaControl. API никогда не возвращает приватный ключ при GET-запросе.

Серверы

Основной ресурс. Полный CRUD для контролируемых экземпляров MariaDB / MySQL.

```
# Список всех серверов
GET /api/v1/servers

# Список с фильтрами
GET /api/v1/servers?tag=production&environment=staging

# Детали сервера
GET /api/v1/servers/42

# Создать сервер
POST /api/v1/servers
{
  "hostname": "db-prod-01.acme.com",
  "ip": "10.0.1.10",
  "port": 3306,
  "ssh_key_id": 5,
  "client_id": 7,
  "environment_id": 1,
  "tags": ["production", "galera", "dc-paris"]
}

# Изменить сервер
PUT /api/v1/servers/42
{"port": 3307}

# Назначить теги
POST /api/v1/servers/42/tags
{"tags": ["production", "critical"]}

# Назначить SSH-ключ
PUT /api/v1/servers/42/ssh-key
{"ssh_key_id": 5}

# Удалить сервер
DELETE /api/v1/servers/42
```

Рабочие процессы Infrastructure-as-Code

Основное преимущество API — возможность описать полный инвентарь в версионированном файле и применить его автоматически.

Пример: инвентарный файл YAML

```yaml
# pmacontrol-inventory.yaml
tags:
  - name: production
    color: "#22c55e"
  - name: staging
    color: "#f59e0b"
  - name: galera
    color: "#14b8a6"

environments:
  - name: production
  - name: staging
  - name: development

ssh_keys:
  - name: collector-2026
    public_key_file: ./keys/collector-2026.pub

servers:
  - hostname: db-prod-01.acme.com
    ip: 10.0.1.10
    port: 3306
    ssh_key: collector-2026
    environment: production
    tags: [production, galera]

  - hostname: db-prod-02.acme.com
    ip: 10.0.1.11
    port: 3306
    ssh_key: collector-2026
    environment: production
    tags: [production, galera]

  - hostname: db-staging-01.acme.com
    ip: 10.0.2.10
    port: 3306
    ssh_key: collector-2026
    environment: staging
    tags: [staging]
```

Скрипт применения

Скрипт на Python или Bash читает этот файл и вызывает API PmaControl для синхронизации состояния:

```
#!/bin/bash
API="https://pmacontrol.example.com/api/v1"
TOKEN="your-webservice-token"

# Создать теги
for tag in production staging galera; do
  curl -s -X POST "$API/tags" \
    -H "Authorization: Bearer $TOKEN" \
    -H "Content-Type: application/json" \
    -d "{\"name\": \"$tag\"}"
done

# Создать серверы
curl -s -X POST "$API/servers" \
  -H "Authorization: Bearer $TOKEN" \
  -H "Content-Type: application/json" \
  -d @server-prod-01.json
```

Интеграция CI/CD

Естественный паттерн — интеграция в CI/CD-конвейер:

1. DBA изменяет файл `pmacontrol-inventory.yaml` в Git
2. Merge request проходит ревью команды
3. CI-конвейер проверяет синтаксис и ограничения (нет дублей IP, SSH-ключи валидны)
4. CD-конвейер применяет изменения через API PmaControl
5. PmaControl автоматически начинает мониторинг новых серверов

Идемпотентность

Все эндпоинты создания **идемпотентны**: если ресурс уже существует (тот же hostname, тот же IP), API возвращает существующий ресурс вместо создания дубликата. Это позволяет повторно запускать инвентарный скрипт без побочных эффектов.

```
# Первый вызов: создаёт сервер, возвращает 201
POST /api/v1/servers  → 201 Created

# Второй идентичный вызов: возвращает существующий сервер, 200
POST /api/v1/servers  → 200 OK (existing)
```

Коды возврата

Код	Значение
200	Успех (чтение или обновление)
201	Ресурс создан
204	Удаление выполнено
400	Некорректный payload
401	Отсутствующий или невалидный токен
403	Недостаточно прав
404	Ресурс не найден
409	Конфликт (ограничение уникальности)

Текущие ограничения

API v1 охватывает CRUD-операции над инвентарём. Пока не охвачены:

- **Метрики** (чтение временных рядов) — запланировано для v2
- **Оповещения** (настройка порогов) — запланировано для v2
- **Бэкапы** (запуск и статус) — запланировано для v2
- **Экспорт конфигурации** (my.cnf, правила ProxySQL) — обсуждается

Заключение

REST API PmaControl превращает управление парком MariaDB / MySQL из ручного процесса в программируемый и версионированный рабочий процесс.

Опишите свой инвентарь в Git. Применяйте его через API. Предоставьте PmaControl автоматический мониторинг. Это Infrastructure-as-Code, применённый к мониторингу баз данных.