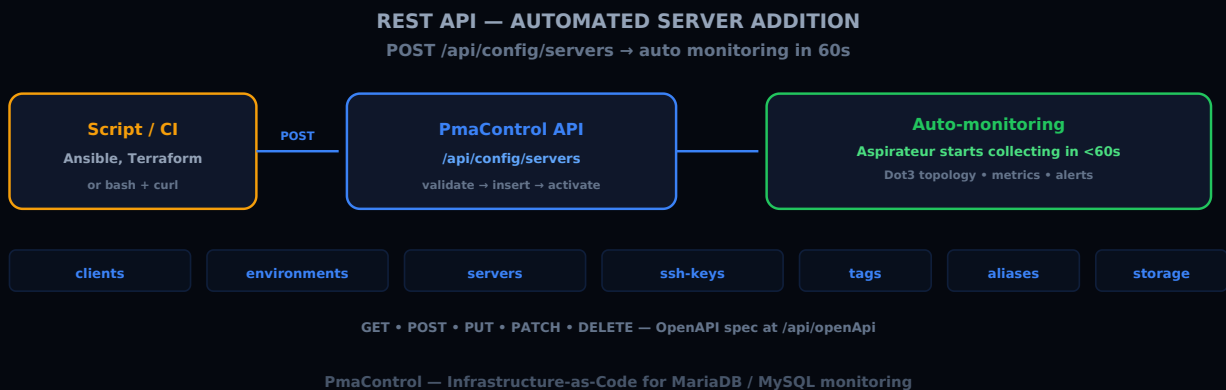


Ajouter automatiquement des serveurs MariaDB / MySQL dans PmaControl via le webservice

Aurélien LEQUOY · 13 avril 2026

PMACONTROL REST-API AUTOMATION DEVOPS WEBSERVICE



Pourquoi automatiser l'ajout de serveurs

Quand vous supervisez 5 serveurs MariaDB / MySQL, les ajouter à la main dans PmaControl est acceptable. À 50 ou 200 serveurs, ça ne l'est plus.

L'API REST de PmaControl permet d'industrialiser cet ajout : un script, un pipeline CI/CD, ou un outil d'orchestration (Ansible, Terraform) peut créer un serveur supervisé en une seule requête HTTP.

Pré-requis

- Un compte webservice actif dans PmaControl (créé lors de l'installation via `config.json`)
- L'URL de votre instance PmaControl
- Le client et l'environnement cibles doivent exister (créez-les d'abord via l'API si besoin)

Endpoint principal

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

Authentification via le compte webservice configuré dans `configuration/webservice.config.php`.

Payload pour ajouter un serveur

```
{
  "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
}
```

Champs importants

Champ	Description
<code>id_client</code>	Identifiant du client (organisation)
<code>id_environment</code>	Identifiant de l'environnement (Production, Staging, etc.)
<code>ip</code>	Adresse IP d'accès au serveur MySQL
<code>port</code>	Port MySQL (3306 par défaut)
<code>login</code> / <code>passwd</code>	Identifiants MySQL pour la collecte
<code>is_monitored</code>	1 = supervision active, 0 = désactivé
<code>is_proxy</code>	1 si c'est un ProxySQL/MaxScale (teste la connexion différemment)

Champ	Description
is_vip	1 si c'est une VIP/DNS (pas de connexion directe, suivi de redirection)

Créer les dépendances d'abord

Avant d'ajouter un serveur, le client et l'environnement doivent exister.

Créer un client

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

Créer un environnement

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

Ajouter une clé 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..."}'
```

Script complet d'ajout en masse

```
#!/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

```

Lister les serveurs existants

```

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

```

Mettre à jour un serveur

```

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}'
```

Supprimer un serveur

La suppression est un **soft delete** (`is_deleted = 1`). Le serveur disparaît de l'interface mais reste en base :

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

Tags pour organiser vos serveurs

Les tags permettent de catégoriser les serveurs (datacenter, rôle, version) :

```
# Créer un tag  
curl -s -X POST http://pmacontrol.local/fr/api/config/tags \  
-H "Content-Type: application/json" \  
-d '{"name": "dc-paris", "description": "Paris datacenter"}'  
  
# Associer un tag à un serveur (via l'interface ou l'API)
```

Vérification après ajout

Après l'ajout, PmaControl commence automatiquement la collecte si `is_monitored = 1`. Vérifiez dans l'interface :

1. Le serveur apparaît dans la liste `Server > Main`
2. La topologie Dot3 se met à jour au prochain cycle
3. Les premières métriques arrivent dans les 60 secondes

Spécification OpenAPI

PmaControl expose sa spécification OpenAPI complète :

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

Utilisable avec Swagger UI ou pour générer des clients SDK automatiquement.

Conclusion

L'API REST de PmaControl transforme l'ajout de serveurs d'une tâche manuelle en une opération scriptable et reproductible. Combinée avec Ansible ou Terraform, elle permet une approche Infrastructure-as-Code de la supervision MariaDB / MySQL.