

Если вы видите что-то необычное, просто сообщите мне.

# Percona

Запустить **Percona Monitoring and Management (PMM)** и **PostgreSQL** в Docker, чтобы не устанавливать ничего на хост.

---

## 1. Запуск PMM Server + PostgreSQL в Docker Compose

Создайте файл `docker-compose.yml`:

```
version: '3'

services:
  # PMM Server
  pmm-server:
    image: percona/pmm-server:2
    container_name: pmm-server
    restart: unless-stopped
    ports:
      - "80:80"          # Веб-интерфейс
      - "443:443"        # HTTPS (опционально)
    volumes:
      - pmm-data:/srv     # Для хранения данных
    environment:
      - PMM_DEBUG=1       # Для отладки (опционально)

  # PostgreSQL для мониторинга
  postgresql:
    image: postgres:13
    container_name: postgresql
    restart: unless-stopped
```

```
environment:
  POSTGRES_USER: postgres
  POSTGRES_PASSWORD: postgres
  POSTGRES_DB: test_db

ports:
  - "5432:5432"      # Порт PostgreSQL

volumes:
  - postgres-data:/var/lib/postgresql/data

# PMM Client (для мониторинга PostgreSQL)
pmm-client:
  image: percona/pmm-client:2
  container_name: pmm-client
  restart: unless-stopped
  depends_on:
    - pmm-server
    - postgresql
  environment:
    - PMM_SERVER=http://pmm-server:80
    - PMM_USER=admin      # Login PMM (по умолчанию)
    - PMM_PASSWORD=admin  # Пароль PMM (по умолчанию)
  cap_add:
    - NET_ADMIN          # Нужно для сбора метрик
  volumes:
    - /var/run/docker.sock:/var/run/docker.sock # Для мониторинга Docker

volumes:
  pmm-data:
  postgres-data:
```

## 2. Запуск и настройка

### 1. Запустите контейнеры:

```
docker-compose up -d
```

### 2. Добавьте PostgreSQL в мониторинг:

```
docker exec -it pmm-client pmm-admin add postgresql \
  --username=postgres \
  --password=postgres \
  --server-url=http://pmm-server:80 \
  --service-name=postgresql-docker
```

- `--username` и `--password` — учётные данные PostgreSQL.
- `--service-name` — имя сервиса в PMM.

### 3. Проверьте статус:

```
docker exec -it pmm-client pmm-admin status
```

Должно появиться что-то вроде:

Service type	Service name	Address and Port	Status
PostgreSQL	postgresql-docker	postgresql:5432	RUNNING

---

## 3. Проверка в PMM UI

#### 1. Откройте веб-интерфейс PMM:

**`http://localhost:80`** (или IP вашего сервера).

#### 2. Перейдите в **PostgreSQL → Overview**.

#### 3. Если данные не отображаются:

- Проверьте логи PMM Client:

```
docker logs pmm-client
```

- Убедитесь, что PostgreSQL доступен из контейнера `pmm-client`:

```
docker exec -it pmm-client psql -h postgresql -U postgres -c "SELECT 1"
```

---

## 4. Дополнительные настройки

### Настройка `pg_stat_statements` (для мониторинга запросов)

1. Подключитесь к PostgreSQL:

```
docker exec -it postgresql psql -U postgres
```

2. Включите расширение:

```
CREATE EXTENSION pg_stat_statements;  
ALTER SYSTEM SET shared_preload_libraries = 'pg_stat_statements';
```

3. Перезапустите PostgreSQL:

```
docker restart postgresql
```

### Настройка Alertmanager (опционально)

Если нужны уведомления (Slack, Email), настройте **Alertmanager** в PMM через веб-интерфейс (**Alerting** → **Alertmanager**).

---

## Вывод

**Всё в Docker:**

- PMM Server (`pmm-server`) — сбор и визуализация метрик.
- PostgreSQL (`postgresql`) — СУБД для мониторинга.

- PMM Client (`pmm-client`) — агент, который отправляет данные в PMM.
- 

Revision #1

Created 11 June 2025 08:18:14 by gasick

Updated 11 June 2025 08:19:37 by gasick