

Exemple Setup Postgres Container

Nous allons mettre en place le service Postgres pour la première fois. Voici les démarches à suivre pour que le nouveau service fonctionne avec notre environnement :

Etape 1 : Création de la structure des volumes

```
sudo mkdir -p /mnt/volumes/postgres/{data,logs,conf,init}
sudo chown -R dockersvc:dockersvc /mnt/volumes/postgres
sudo setfacl -R -m u:dockersvc:rw,g:dockersvc:rw /mnt/volumes/postgres
sudo setfacl -R -d -m u:dockersvc:rw,g:dockersvc:rw /mnt/volumes/postgres
```

Cette arborescence permet :

- **data/** → stockage des bases
- **logs/** → logs PostgreSQL hors conteneur
- **conf/** → fichiers de configuration personnalisés (`postgresql.conf` , `pg_hba.conf`)
- **init/** → scripts SQL ou shell exécutés au premier lancement

Etape 2 : Création du `docker-compose.yml` PostgreSQL

```
version: "3.9"
services:
  postgres:
    image: postgres:17-alpine
    container_name: postgres
    restart: unless-stopped
    user: ${PUID}:${PGID}
    networks:
      - bddnetwork
    ports:
      - 5432:5432
    volumes:
      - ${VOLR00T}/postgres/data:/var/lib/postgresql/data
      - ${VOLR00T}/postgres/logs:/var/log/postgresql
      - ${VOLR00T}/postgres/conf:/etc/postgresql/custom
      - ${VOLR00T}/postgres/init:/docker-entrypoint-initdb.d
```

```
environment:
  - POSTGRES_DB=postgres
  - POSTGRES_USER=postgres
  - POSTGRES_PASSWORD=0YsA4X8N0xC0AN
  - TZ=${TZ}
healthcheck:
  test:
    - CMD-SHELL
    - pg_isready -U ${POSTGRES_USER}
  interval: 10s
  timeout: 5s
  retries: 5
volumes:
  postgres: null
networks:
  bddnetwork:
    external: true
```

Créer le network

Notre stack docker compose utilise un network "bddnetwork" externe. Pour que l'exécution docker compose fonctionne il est nécessaire de créer le network avant de la démarrer :

```
sudo docker network create bddnetwork
```

Etape 3 : Démarrer et Vérifier les UID/GID

Pour vérifier les UID/GID après le premier démarrage :

```
docker exec -it postgres id
```

Résultat attendu :

```
uid=10050(dockersvc) gid=10050(dockersvc) groups=10050(dockersvc)
```

Si l'image officielle PostgreSQL refuse `user: "${PUID}:${PGID}"`, alors **on garde dockersvc propriétaire des volumes** mais **on lance le conteneur sous avec l'utilisateur par défaut mais on va ajouter des ACL spécifique.**

Pour ajouter une ACL spécifique :

```
sudo setfacl -R -m u:999:rwX /mnt/volumes/postgres
```

Tester la connexion à la base de donnée

Se connecter dans le conteneur :

```
sudo docker exec -it postgres bash
```

Puis se connecter à la BDD avec l'user et le nom de la base de donnée définit dans la stack docker compose :

```
psql -U <username> -d <dbname>
```

Revision #4

Created 24 August 2025 15:04:37 by gpatruno

Updated 25 August 2025 15:31:31 by gpatruno