



Creazione della base dati PostgreSQL

PREREQUISITO:

Scaricare il file contenuto nella repository di riferimento di nome `shop_db.sql`

Introduzione

In questa lezione, esploreremo come creare e gestire una base dati utilizzando PostgreSQL all'interno di un ambiente Docker. Utilizzeremo Docker per creare due contenitori: uno per il DBMS PostgreSQL e l'altro per PG Admin, un'interfaccia grafica che facilita la gestione del database.

Obiettivo

L'obiettivo è configurare un ambiente di sviluppo dove un contenitore ospita PostgreSQL e un altro PG Admin. Questo setup ci permetterà di gestire facilmente il database senza dover utilizzare la riga di comando.

Prerequisiti

Assicurarsi di aver installato Docker e Docker Compose sul proprio sistema operativo (Linux, macOS o Windows).

Passaggi di Configurazione

1. Preparazione del File di Configurazione Docker Compose

Per gestire la configurazione dei contenitori, utilizzeremo Docker Compose. Creiamo un file chiamato `docker-compose.yml` nella directory di lavoro.

```
version: '3.3'

networks:
  ntpgsql:
    driver: bridge
    ipam:
      driver: default
      config:
        - subnet: 172.21.0.0/24

services:
  pgsql:
    image: postgres:latest
    restart: unless-stopped
    container_name: postgresql
    volumes:
      - psdb-volume:/var/lib/postgresql/data
    networks:
      ntpgsql:
        ipv4_address: 172.21.0.2
    ports:
      - target: 5432
        published: 5433
        protocol: tcp
        mode: host
    environment:
      - POSTGRES_PASSWORD=123_Adiener
  pgadmin:
    image: dpage/pgadmin4
    restart: unless-stopped
    container_name: pgadmin4
```

```
networks:
  ntpgsql:
    ipv4_address: 172.21.0.3
ports:
  - target: 80
    published: 80
    protocol: tcp
    mode: host
environment:
  - PGADMIN_CONFIG_SERVER_MODE=True
  - PGADMIN_DEFAULT_EMAIL=adienerlopez@gmail.com
  - PGADMIN_DEFAULT_PASSWORD=123_Adiener

volumes:
  psdb-volume:
```

2. Creazione e Avvio dei Contenitori

Apriamo un terminale nella directory dove si trova il file `docker-compose.yml` e eseguiamo il comando per avviare i contenitori.

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

Questo comando scaricherà le immagini necessarie, creerà i contenitori e li avvierà in modalità detached.

3. Verifica dell'Installazione

Per verificare che i contenitori siano attivi, utilizziamo il comando:

```
docker ps
```

Dovremmo vedere i due contenitori `postgresql` e `pgadmin4` in esecuzione.