

GABRIELE ROMANATO

Configurare e usare MongoDB in Docker con Docker Compose

MongoDB è un database NoSQL molto usato in applicazioni moderne. Grazie a Docker e Docker Compose, è possibile configurare rapidamente un'istanza MongoDB per sviluppo o test, senza installare nulla localmente. In questo articolo vedremo come fare.

Prerequisiti

- Docker installato
- Docker Compose (integrato in Docker da versioni recenti)
- Un editor di testo o IDE per modificare file YAML

1. Creare il file `docker-compose.yml`

In una nuova cartella di progetto, crea un file chiamato `docker-compose.yml` con il contenuto seguente:

```
services:
  mongo:
    image: mongo:latest
    container_name: mongodb
    ports:
      - "27017:27017"
    volumes:
      - ./data:/data/db
    environment:
      MONGO_INITDB_ROOT_USERNAME: root
      MONGO_INITDB_ROOT_PASSWORD: example
```

Questo file:

- Scarica l'immagine ufficiale di MongoDB
- Espone la porta 27017
- Monta una cartella locale per la persistenza dei dati
- Imposta utente e password di root

2. Avviare MongoDB

Dalla stessa cartella dove si trova il file `docker-compose.yml`, esegui:

```
docker compose up -d
```

Questo comando scarica l'immagine (se non presente), crea il container e lo avvia in background.

3. Connettersi a MongoDB

Puoi collegarti a MongoDB in diversi modi:

Usando la Mongo Shell via Docker

```
docker exec -it mongodb mongosh -u root -p example
```

Usando un client come MongoDB Compass

Stringa di connessione:

```
mongodb://root:example@localhost:27017
```

4. Aggiungere un database di esempio (opzionale)

Puoi pre-caricare un database scrivendo uno script JS da montare nella cartella `/docker-entrypoint-initdb.d`. Esempio:

```
services:
  mongo:
    image: mongo:latest
    container_name: mongodb
    ports:
      - "27017:27017"
    volumes:
      - ./data:/data/db
      - ./init:/docker-entrypoint-initdb.d
    environment:
      MONGO_INITDB_ROOT_USERNAME: root
      MONGO_INITDB_ROOT_PASSWORD: example
```

Nel file `./init/init.js`:

```
db = db.getSiblingDB('mydb');
db.createCollection('users');
db.utenti.insertMany([
  { nome: "Luca", eta: 30 },
  { nome: "Anna", eta: 25 }
]);
```

5. Fermare e rimuovere i container

```
docker compose down
```

Questo comando ferma e rimuove il container. I dati rimangono salvati nella cartella `./data`.

Conclusione

Grazie a Docker Compose è possibile configurare un ambiente MongoDB in pochi minuti, replicabile su qualsiasi sistema. È uno strumento ideale per sviluppatori che vogliono un database isolato, veloce da gestire e facilmente condivisibile.