

GABRIELE ROMANATO

Menu

Come usare PostgreSQL in Docker con Docker Compose e Go

In questo articolo, esploreremo come configurare PostgreSQL utilizzando Docker e Docker Compose e come integrarlo in un'applicazione Go. Questo approccio è utile per semplificare lo sviluppo locale e la distribuzione delle applicazioni.

Configurazione di Docker Compose

Creiamo un file `docker-compose.yml` per configurare PostgreSQL:

```
version: '3.8'
services:
  db:
    image: postgres:15
    container_name: postgres_container
    environment:
      POSTGRES_USER: user
      POSTGRES_PASSWORD: password
      POSTGRES_DB: mydb
    ports:
      - "5432:5432"
    volumes:
      - db_data:/var/lib/postgresql/data
volumes:
  db_data:
```

Avvio di PostgreSQL con Docker Compose

Esegui il comando seguente per avviare il container:

```
docker compose up -d
```

Verifica che il container sia in esecuzione:

```
docker ps
```

Connessione a PostgreSQL in Go

Aggiungi la libreria `pq` al progetto Go per connetterti a PostgreSQL:

```
go get github.com/lib/pq
```

Crea un file Go, ad esempio `main.go`, con il seguente codice:

```
package main

import (
    "database/sql"
    "fmt"
)
```

```
    _ "github.com/lib/pq"
)

func main() {
    connStr := "postgres://user:password@localhost:5432/mydb?sslmode=disable"
    db, err := sql.Open("postgres", connStr)
    if err != nil {
        panic(err)
    }
    defer db.Close()

    err = db.Ping()
    if err != nil {
        panic(err)
    }

    fmt.Println("Connessione a PostgreSQL riuscita!")
}
```

Test dell'Applicazione

Esegui l'applicazione Go per testare la connessione al database:

```
go run main.go
```

Se tutto è configurato correttamente, dovresti vedere il messaggio:

```
Connessione a PostgreSQL riuscita!
```

Conclusione

In questo articolo, abbiamo configurato PostgreSQL utilizzando Docker e Docker Compose, e abbiamo creato un'applicazione Go per connetterci al database. Questo flusso di lavoro semplifica notevolmente la gestione dei database nello sviluppo locale.

Applicazioni Correlate

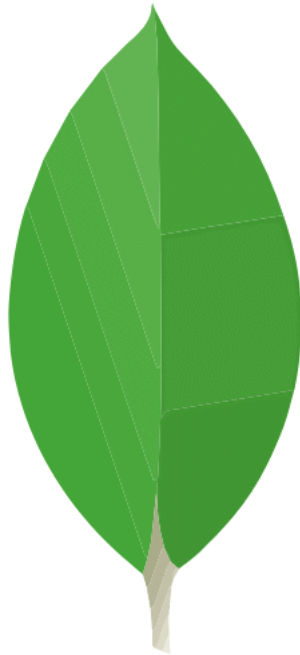


-

Go Placeholder Image

Applicazione in Go per la creazione di immagini segnaposto.

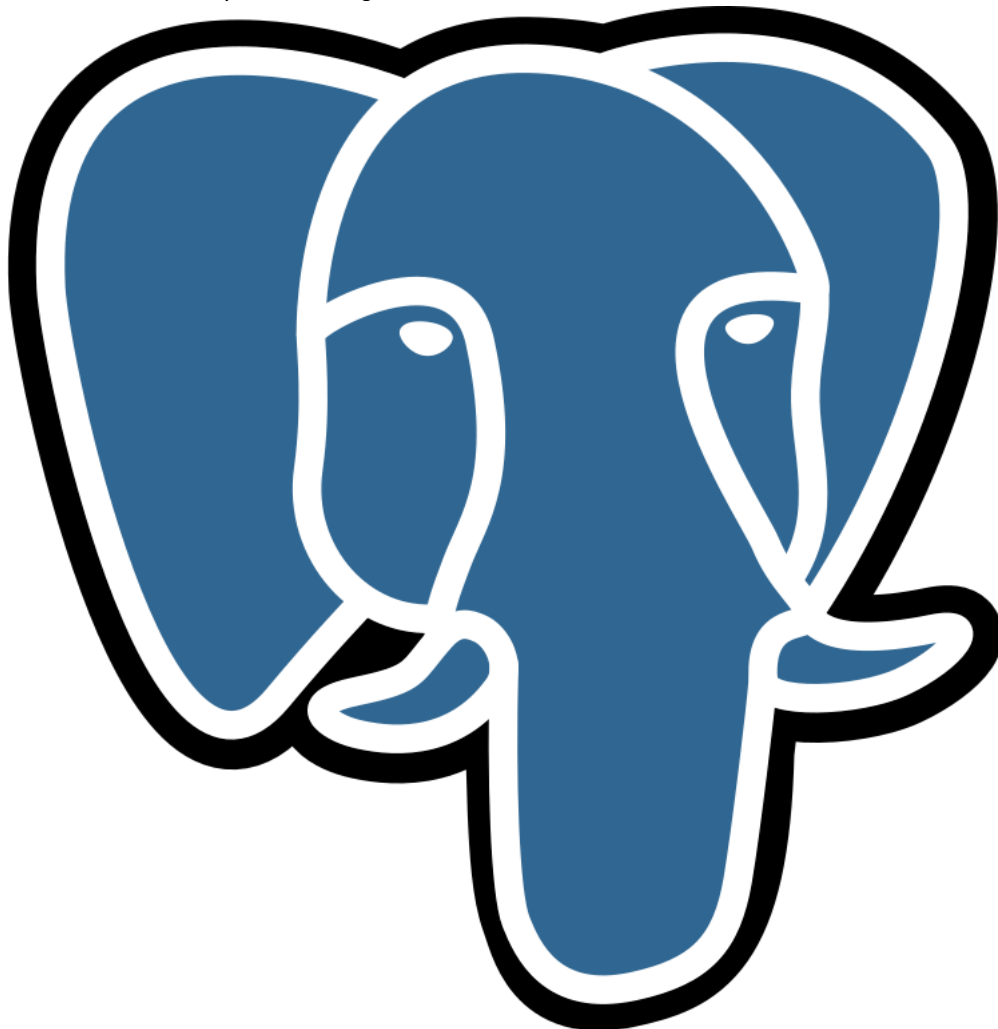
[Docker](#)[Docker Compose](#)[Go](#)[JavaScript](#)



•

Go MongoDB App

Applicazione basata su MongoDB ed implementata in Go con il driver ufficiale.
DockerDocker ComposeGoMongoDB



•

Go PostgreSQL App

Applicazione basata su PostgreSQL e sviluppata in Go.

Docker Docker Compose Go PostgreSQL