

Guida alla Build e Deploy - Homework 2

Questo documento descrive i passaggi necessari per eseguire la build, il deploy e l'utilizzo del sistema distribuito.

1. Clonare la repository

Per iniziare, clonare la repository del progetto utilizzando il seguente comando:

```
git clone https://github.com/TanoBont/Homework2-DSBD_2425.git
```

Posizionarsi nella cartella del progetto con il comando:

```
cd Homework2-DSBD_2425
```

2. Build e avvio dei container

Prima di avviare i container, è necessario costruire le immagini Docker dei servizi. Utilizzare il comando:

```
docker-compose up --build
```

Questo comando esegue contemporaneamente:

- La build delle immagini.
- L'avvio dei container definiti nel file *docker-compose.yml*.

Nota importante:

Il file *docker-compose.yml* include un health check per verificare che il database sia completamente operativo prima di avviare gli altri container. Questo garantisce che il server gRPC e il data collector non tentino di connettersi al database prima che sia funzionante.

3. Verificare che il sistema sia pronto

Attendere qualche secondo dopo che i log si sono fermati per essere certi che il sistema sia completamente operativo.

4. Avvio del client

Per utilizzare il sistema, aprire un nuovo terminale e avviare il client con il seguente comando:

```
python client.py
```

Il client si conatterà al server gRPC avviato all'interno dei container. Sarà quindi possibile interagire con le funzionalità del sistema.

5. Arrestare il sistema

Per interrompere il sistema e rimuovere i container senza eliminare i dati persistenti (volumi), utilizzare i comandi:

Ctrl + c

(nel terminale dove è stato fatto il docker compose e, nel terminale dove è in run il client, per interrompere l'esecuzione)

e successivamente:

```
docker-compose down
```

Se si desidera rimuovere anche i volumi associati (e quindi eliminare i dati persistenti):

```
docker-compose down -v
```

Comandi Utili per MySQL

Entrare nel container del database:

```
docker exec -it db bash
```

Accedere a MySQL con il seguente comando:

```
mysql -u myuser -p
```

Inserire la password:

```
mypassword
```

Selezionare il database del sistema:

```
USE dsbd_db;
```

I nomi delle tabelle sono i seguenti:

```
users;
```

```
stock_data;
```

```
registration_messages
```

```
update_messages
```

Comandi Utili per Kafka

Per accedere alla shell del container Kafka, eseguire il seguente comando:

```
docker exec -it kafka bash
```

Per vedere la lista di tutti i topic Kafka presenti nel sistema, usare il comando:

```
kafka-topics --bootstrap-server kafka:9092 --list
```

Per **consumare i messaggi dal topic to-alert-system** e vedere i messaggi in tempo reale a partire dal primo messaggio, usare:

```
kafka-console-consumer --bootstrap-server kafka:9092 --topic to-alert-system  
--from-beginning
```

Per **consumare i messaggi dal topic to-notifier** e vedere i messaggi in tempo reale a partire dal primo messaggio, usare:

```
kafka-console-consumer --bootstrap-server kafka:9092 --topic to-notifier --from-beginning
```

Per **inviare messaggi al topic to-alert-system**, usare il comando kafka-console-producer. Questo comando permetterà di scrivere messaggi direttamente dalla shell che saranno inviati al topic specificato:

```
kafka-console-producer --bootstrap-server kafka:9092 --topic to-alert-system
```

Dopo aver eseguito il comando per il producer Kafka, inserire i messaggi da inviare in modalità interattiva.

```
> inserire messaggio da inviare
```

Ogni volta che viene premuto **Enter**, il messaggio viene inviato immediatamente al topic to-alert-system.