

Konfigurasi Docker Compose

Berkaitan dengan file `docker-compose.yml`, berikut adalah penjelasan rinci tentang semua kata kunci atau bagian yang mungkin terdapat dalam konfigurasi tersebut. File ini digunakan untuk mendefinisikan dan mengelola aplikasi multi-container menggunakan Docker Compose, yang menyediakan cara deklaratif untuk menyusun dan menjalankan berbagai layanan dalam lingkungan kontainer Docker.

Versi Docker Compose

```
version: '3.8'
```

- **Version:** Menentukan versi Docker Compose yang digunakan. Versi ini menentukan fitur-fitur dan sintaks yang tersedia dalam file konfigurasi. Versi terbaru adalah 3.8 pada contoh ini.

Services

```
services:
  web:
    image: nginx:latest
    ports:
      - "8080:80"
    volumes:
      - ./nginx.conf:/etc/nginx/nginx.conf
    depends_on:
      - api
```

- **Services:** Bagian ini mendefinisikan layanan atau container yang akan dijalankan sebagai bagian dari aplikasi. Setiap layanan didefinisikan dengan nama unik (misalnya `web`, `api`) yang mengacu pada nama container. Ini juga merupakan level utama dalam struktur YAML.
 - **Image:** Menentukan image Docker yang digunakan untuk layanan ini. Contoh di atas menggunakan `nginx:latest`.
 - **Ports:** Mengaitkan port dari host dengan port dalam container, memungkinkan akses ke layanan di dalam container melalui host.

- **Volumes:** Mendefinisikan volume Docker yang akan di-mount ke dalam container, memungkinkan untuk menyimpan konfigurasi atau data yang persisten.
- **Depends_on:** Menentukan ketergantungan antara layanan. Misalnya, `web` bergantung pada `api`, yang berarti `api` akan dimulai sebelum `web`.

Networks

```
networks:  
  app-network:  
    driver: bridge
```

- **Networks:** Bagian ini mendefinisikan jaringan yang digunakan oleh aplikasi atau layanan. Ini memungkinkan container-container dalam aplikasi untuk berkomunikasi satu sama lain.
 - **Driver:** Menentukan driver jaringan yang digunakan. Contoh di atas menggunakan driver `bridge`, yang merupakan driver jaringan default untuk Docker.

Volumes

```
volumes:  
  data-volume:  
    driver: local
```

- **Volumes:** Mendefinisikan volume Docker yang digunakan oleh aplikasi. Ini memungkinkan untuk menyimpan data persisten atau berbagi data antara container-container.
 - **Driver:** Menentukan driver volume yang digunakan. Contoh di atas menggunakan driver `local`, yang berarti volume akan dikelola secara lokal oleh Docker.

Environment Variables

```
environment:  
  MYSQL_ROOT_PASSWORD: example  
  MYSQL_DATABASE: myapp
```

- **Environment:** Menentukan variabel lingkungan yang diset dalam container. Ini digunakan untuk mengatur konfigurasi aplikasi atau koneksi ke basis data, misalnya.

Build Configuration

```
build:
  context: .
  dockerfile: Dockerfile
```

- **Build:** Konfigurasi ini mendefinisikan bagaimana Docker Compose harus membangun image dari Dockerfile yang diberikan.
 - **Context:** Menentukan direktori atau path di mana Dockerfile dan file lain yang diperlukan berada.
 - **Dockerfile:** Nama file Dockerfile yang digunakan untuk membangun image. Jika tidak disediakan, Docker Compose akan mencari file `Dockerfile` secara default.

Restart Policy

```
restart: always
```

- **Restart:** Menentukan kebijakan restart untuk layanan. Contoh di atas menggunakan `always`, yang berarti Docker Compose akan selalu mencoba untuk memulai ulang container jika terjadi kegagalan.

Links (Deprecated)

```
links:
  - db:database
```

- **Links:** Awalnya digunakan untuk mengaitkan container dengan container lain, tetapi sekarang dianggap usang dan tidak disarankan. Penggantinya adalah penggunaan `network`.

Example Docker Compose File

Berikut adalah contoh lengkap dari berbagai kata kunci yang telah dijelaskan di atas dalam satu file `docker-compose.yml`:

```
version: '3.8'

services:
  web:
    image: nginx:latest
    ports:
```

- "8080:80"

volumes:

- ./nginx.conf:/etc/nginx/nginx.conf

depends_on:

- api

api:

image: node:14

environment:

- NODE_ENV=production

volumes:

- ./app:/app

ports:

- "3000:3000"

depends_on:

- db

db:

image: mysql:5.7

environment:

- MYSQL_ROOT_PASSWORD=secret

- MYSQL_DATABASE=myapp

- MYSQL_USER=myuser

- MYSQL_PASSWORD=mypassword

ports:

- "3306:3306"

volumes:

- db-data:/var/lib/mysql

networks:

app-network:

driver: bridge

volumes:

db-data:

driver: local

environment:

MYSQL_ROOT_PASSWORD: example

MYSQL_DATABASE: myapp

```
restart: always
```

File `docker-compose.yml` ini mengilustrasikan penggunaan berbagai kata kunci yang digunakan untuk mendefinisikan aplikasi multi-container menggunakan Docker Compose. Setiap bagian dan kata kunci memiliki peran penting dalam menentukan bagaimana aplikasi diatur, dibangun, dan dijalankan dalam lingkungan kontainer Docker.

Revision #1

Created 17 December 2024 14:34:24 by Admin

Updated 17 December 2024 15:18:29 by Admin