

Docker vs Docker Compose

Docker dan Docker Compose adalah dua alat yang berbeda tetapi saling melengkapi dalam ekosistem Docker. Meskipun keduanya digunakan untuk mengelola aplikasi berbasis kontainer, peran dan fungsinya memiliki perbedaan yang signifikan:

Docker

Docker adalah platform open-source yang memungkinkan pengembang untuk mengembangkan, menguji, dan menjalankan aplikasi di dalam kontainer. Berikut adalah poin utama terkait Docker:

1. **Kontainerisasi:** Docker memungkinkan pengguna untuk membuat, mengelola, dan menjalankan kontainer berdasarkan image yang telah didefinisikan sebelumnya. Setiap kontainer berjalan terisolasi satu sama lain, tetapi menggunakan kernel yang sama dari host.
2. **Image:** Docker menggunakan image sebagai blueprint untuk membuat kontainer. Image berisi segala yang diperlukan untuk menjalankan aplikasi, termasuk kode, runtime, library, variabel lingkungan, dan pengaturan lainnya.
3. **CLI Commands:** Docker dilengkapi dengan berbagai perintah CLI yang memungkinkan pengguna untuk mengelola kontainer, image, jaringan, dan volume. Beberapa perintah yang umum digunakan telah dijelaskan sebelumnya, seperti `docker run`, `docker build`, `docker stop`, `docker start`, dan lain-lain.
4. **Single Container Management:** Docker secara alami fokus pada manajemen kontainer tunggal. Ini berarti pengguna dapat membuat, menjalankan, dan menghentikan kontainer satu per satu menggunakan perintah-perintah Docker CLI.

Docker Compose

Docker Compose, di sisi lain, adalah alat yang dibangun di atas Docker yang memungkinkan pengguna untuk mendefinisikan dan menjalankan aplikasi multi-container dalam lingkungan yang terisolasi. Berikut adalah poin utama terkait Docker Compose:

1. **Multi-container Applications:** Docker Compose memungkinkan pengguna untuk mendefinisikan aplikasi yang terdiri dari beberapa layanan atau kontainer dalam satu file konfigurasi (`docker-compose.yml`). File ini menyediakan cara deklaratif untuk mendefinisikan layanan-layanan yang saling tergantung dan mengelola ketergantungan serta konfigurasi antar layanan tersebut.
2. **YAML Configuration:** Konfigurasi Docker Compose ditulis dalam format YAML yang mudah dibaca dan dimengerti oleh manusia. Ini memungkinkan pengguna untuk

mendefinisikan image, port, volume, jaringan, variabel lingkungan, dan pengaturan lainnya dalam satu file yang terstruktur.

3. **Orchestration:** Docker Compose tidak hanya untuk menjalankan kontainer, tetapi juga untuk mengelola siklus hidup aplikasi secara keseluruhan, termasuk membangun image, memulai, menghentikan, dan menghapus kontainer secara bersamaan sesuai dengan definisi dalam file `docker-compose.yml`.
4. **Command Line Interface:** Docker Compose memiliki perintah CLI tersendiri (`docker-compose`) yang berbeda dari CLI Docker. Ini memungkinkan pengguna untuk melakukan operasi terhadap aplikasi yang didefinisikan dalam file `docker-compose.yml`, seperti `docker-compose up`, `docker-compose down`, `docker-compose build`, `docker-compose start`, dan sebagainya.

Perbandingan Singkat

- **Docker:** Fokus pada manajemen kontainer individual dan image Docker, baik untuk pengembangan, pengujian, dan produksi.
- **Docker Compose:** Fokus pada definisi dan pengelolaan aplikasi multi-container, menyediakan cara deklaratif untuk mendefinisikan layanan-layanan yang tergantung satu sama lain dalam satu lingkungan.

Kapan Menggunakan Mana?

- Gunakan **Docker** untuk pengembangan, uji coba, dan penyebaran kontainer individual.
- Gunakan **Docker Compose** ketika Anda perlu mengelola aplikasi yang terdiri dari beberapa layanan atau kontainer, dan membutuhkan cara yang terstruktur untuk mendefinisikan dan menjalankan aplikasi tersebut.

Dengan memahami perbedaan dan kelebihan masing-masing alat, pengguna dapat memilih dan mengintegrasikan Docker dan Docker Compose sesuai dengan kebutuhan pengembangan dan operasional aplikasi mereka.

Revision #1

Created 17 December 2024 14:39:56 by Admin

Updated 17 December 2024 15:18:29 by Admin