Docker Compose

Overview

Docker Compose is a tool for defining and running multi-container applications. It is the key to unlocking a streamlined and efficient development and deployment experience.

Compose simplifies the control of your entire application stack, making it easy to manage services, networks, and volumes in a single, comprehensible YAML configuration file. Then, with a single command, you create and start all the services from your configuration file.

Compose works in all environments; production, staging, development, testing, as well as CI workflows. It also has commands for managing the whole lifecycle of your application:

- Start, stop, and rebuild services
- View the status of running services
- Stream the log output of running services
- Run a one-off command on a service

Why use Compose?

Using Docker Compose offers several benefits that streamline the development, deployment, and management of containerized applications:

- Simplified control: Docker Compose allows you to define and manage multi-container applications in a single YAML file. This simplifies the complex task of orchestrating and coordinating various services, making it easier to manage and replicate your application environment.
- Efficient collaboration: Docker Compose configuration files are easy to share, facilitating collaboration among developers, operations teams,

and other stakeholders. This collaborative approach leads to smoother workflows, faster issue resolution, and increased overall efficiency.

- Rapid application development: Compose caches the configuration used to create a container. When you restart a service that has not changed, Compose re-uses the existing containers. Re-using containers means that you can make changes to your environment very quickly.
- Portability across environments: Compose supports variables in the Compose file. You can use these variables to customize your composition for different environments, or different users.
- Extensive community and support: Docker Compose benefits from a vibrant and active community, which means abundant resources, tutorials, and support. This community-driven ecosystem contributes to the continuous improvement of Docker Compose and helps users troubleshoot issues effectively.

Common use cases of Docker Compose

Compose can be used in many different ways. Some common use cases are outlined below.

Development environments

When you're developing software, the ability to run an application in an isolated environment and interact with it is crucial. The Compose command line tool can be used to create the environment and interact with it.

The Compose file provides a way to document and configure all of the application's service dependencies (databases, queues, caches, web service APIs, etc). Using the Compose command line tool you can create and start one or more containers for each dependency with a single command (docker compose up).

Together, these features provide a convenient way for you to get started on a project. Compose can reduce a multi-page "developer getting started guide" to a single machine-readable Compose file and a few commands.

Automated testing environments

An important part of any Continuous Deployment or Continuous Integration process is the automated test suite. Automated end-to-end testing requires an environment in which to run tests. Compose provides a convenient way to create and destroy isolated testing environments for your test suite. By defining the full environment in a Compose file, you can create and destroy these environments in just a few commands:

```
$ docker compose up -d
$ ./run_tests
$ docker compose down
```