



# Docker Compose

---

.NET

***docker compose*** is a tool for defining and running multi-container Dockerized applications. ***docker compose*** uses a YAML file to configure an application's services.

[HTTPS://DOCS.DOCKER.COM/COMPOSE/](https://docs.docker.com/compose/)

# docker compose – Overview

<https://docs.docker.com/compose/>

---

**docker-compose up** allows you to create and start all the services composing your application at once. **docker-compose** works in all environments: production, staging, development, testing, and CI/CD workflows.

Using **docker compose** is a three-step process:

1. Define your app's environment ('Dockerize' it) with a *Dockerfile*
2. Define the app's services including their docker run arguments in **docker-compose.yml** so they can be run together in an isolated environment.
3. Run **docker-compose up** and the **Compose** app builds and runs all your containers using the configuration you provide it the **docker-compose.yml**.

```
version: '2.0'
services:
  web:
    build: .
    ports:
      - "5000:5000"
    volumes:
      - ./code
      - logvolume01:/var/log
    links:
      - redis
  redis:
    image: redis
volumes:
  logvolume01: {}
```

# Docker Compose - Features

<https://docs.docker.com/compose/#features>

---

Have many isolated environments on one host	Preserve volume data when containers are created	Recreate containers that have changed	A composition is portable between environments
Because <b>Compose</b> uses unique names for each project, the projects run independently of each other. Use <b>-p</b> to give your project a unique name.	When <b>docker-compose up</b> runs, it copies the volumes from the old <b>container</b> to the new <b>container</b> . This ensures that any data you've created in volumes isn't lost.	<b>Compose</b> re-uses existing <b>containers</b> if they haven't been changed.	You can use variables in the <b>Compose</b> file to customize your composition for different environments or different users.

# Docker Compose – Use Cases

<https://docs.docker.com/compose/#common-use-cases>

---

**Compose** has traditionally been focused on development and testing workflows and is typically used in a development environment. A **docker-compose.yml** file is used to automate testing configurations and dependencies so that workflows are automated and end-to-end testing is faster and easier.

Compose provides a convenient way to create and destroy isolated testing environments for your test suite.

