

Docker Compose



Table of Contents

- ▶ What is Docker Compose?
- ▶ Using Compose
- ▶ Docker Compose File
- ▶ docker-compose Commands



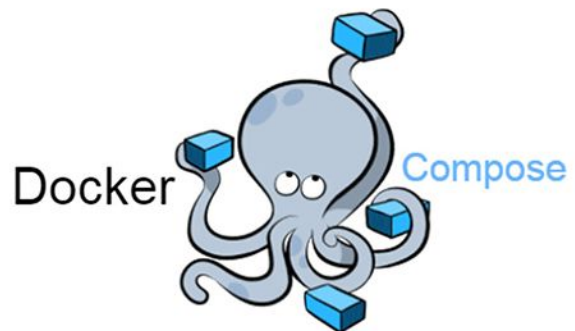
1

What is Docker Compose?



What is Docker Compose?

- ❑ **Compose** is a tool for defining and running **multi-container Docker applications**. With Compose, you use a YAML file to configure your application's services. Then, with a single command, you create and start all the services from your configuration.
- ❑ Compose works in all environments: production, staging, development, testing, as well as CI workflows.





Benefits of Docker Compose

- **Simplified control:** Allows you to define and manage multi-container applications in a single YAML file
- **Efficient collaboration:** Configuration files are easy to share
- **Rapid application development:** Compose caches the configuration used to create a container.
- **Portability across environments:** Compose supports variables in the Compose file



2 Using Compose



▶ Using Compose

Using Compose is basically a three-step process:

- Define your app's environment with a Dockerfile so it can be reproduced anywhere.
- Define the services that make up your app in **docker-compose.yml** so they can be run together in an isolated environment.
- Run **docker-compose** up and Compose starts and runs your entire app.



3

Docker Compose File



Docker Compose File

- The Compose file is a YAML file defining:
 - **name**
 - **services**
 - **networks**
 - **volumes**
 - **configs**
 - **secrets**
- The default path for a Compose file is **./docker-compose.yml**



Docker Compose File

- A **service** definition contains configuration that is applied to each container started for that service, much like passing command-line parameters to **docker run**.
- Likewise, **network** and **volume** definitions are analogous to **docker network create** and **docker volume create**.

```
services:
  web:
    build: .
    ports:
      - "5000:5000"
    volumes:
      - logvolume01:/var/log
    depends_on:
      - redis
    networks:
      - todonet
  redis:
    image: redis
    networks:
      - todonet
volumes:
  logvolume01: {}
networks:
  todonet:
    driver: bridge
```



docker-compose Commands

Command	Description
build	Build or rebuild services
config	Validate and view the Compose file
down	Stop and remove containers, networks, images, and volumes
events	Receive real time events from containers
exec	Execute a command in a running container
images	List images
kill	Kill containers
logs	View output from containers
pause	Pause services



docker-compose Commands

Command	Description
port	Print the public port for a port binding
ps	List containers
pull	Pull service images
push	Push service images
restart	Restart services
rm	Remove stopped containers
run	Run a one-off command
start	Start services
stop	Stop services



docker-compose Commands

Command	Description
top	Display the running processes
unpause	Unpause services
up	Create and start containers
version	Show the Docker-Compose version information



THANKS!

Any questions?





Docker Compose File

```
services:
  db:
    image: mongo:latest
    container_name: db
    networks:
      - todonet
  web:
    build: ../
    networks:
      - todonet
    ports:
      - "3000"
    networks:
      todonet:
        driver: bridge
```

```
todo-app
├── app
│   ├── app.js
│   └── db.js
├── .....
├── compose
│   └── docker-compose.yaml
├── Dockerfile
├── kubernetes
│   ├── db-deployment.yaml
│   ├── db-pvc.yaml
│   ├── db-service.yaml
│   ├── web-deployment.yaml
│   └── web-service.yaml
└── README.md
```



Docker Compose File

There are several **versions** of the Compose file format – 1, 2, 2.x, and 3.x.

Compose file format	Docker Engine release
3.8	19.03.0+
3.7	18.06.0+
3.6	18.02.0+
3.5	17.12.0+
3.4	17.09.0+
3.3	17.06.0+
3.2	17.04.0+
3.1	1.13.1+

Compose file format	Docker Engine release
3.0	1.13.0+
2.4	17.12.0+
2.3	17.06.0+
2.2	1.13.0+
2.1	1.12.0+
2.0	1.10.0+
1.0	1.9.1.+