

**DATE : 02.04.2025**  
**DT/NT : NT**  
**LESSON : DOCKER**  
**SUBJECT: DOCKER-COMPOSE**

**BATCH : B 303**

**AWS-DEVOPS**



**TECHPRO**  
EDUCATION



techproeducation.com



+1 (585) 304 29 59



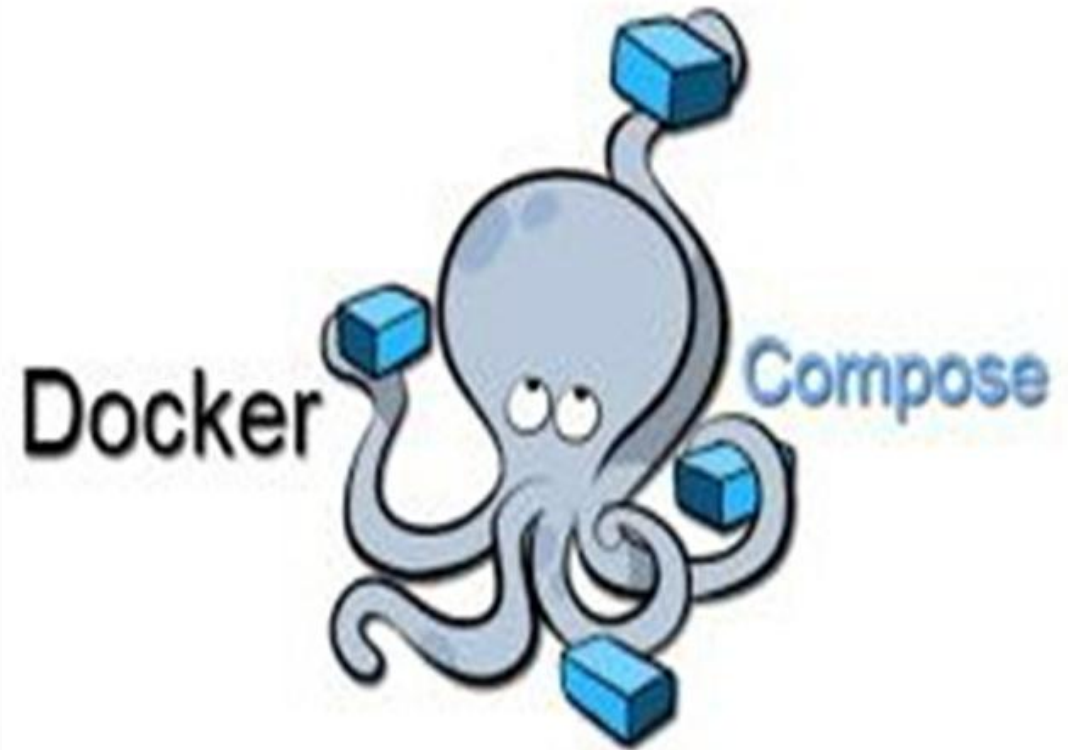


# Table of Contents

- What is Docker Compose?
- Using Compose
- Docker Compose File
- Docker Compose Commands

# 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** based on your configuration.
- Compose works in all environments: production, staging, development, testing, as well as workflows.



# Using Compose

Using Compose is basically 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.

# Docker Compose File

- The Compose file is a YAML file defining:

- services**

- networks**

- volumes**

- The default path for a Compose file is  
./docker-compose.yml

```
version: "3.8"
services:
  web:
    build: .
    depends_on:
      - db
      - redis
  redis:
    image: redis
  db:
    image: postgres
```

```
version: '3.0'
services:
  web:
    build: .
    ports:
      - "5000:5000"
    volumes:
      - ./code
      - logvolume01:/var/log
    depends_on:
      - redis
  redis:
    image: redis
volumes:
  logvolume01: {}
networks:
```

# Docker Compose File

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

docker-compose.yml	docker-compose.yml	docker-compose.yml
<pre>redis:   image: redis db:   image: postgres:9.4 vote:   image: voting-app   ports:     - 5000:80   links:     - redis</pre>	<pre>version: 2 services:   redis:     image: redis   db:     image: postgres:9.4   vote:     image: voting-app     ports:       - 5000:80     depends_on:       - redis</pre>	<pre>version: 3 services:   redis:     image: redis   db:     image: postgres:9.4   vote:     image: voting-app     ports:       - 5000:80</pre>
V1	V2	V3



# 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.

```
version: '3.0'
services:
  web:
    build: .
    ports:
      - "5000:5000"
    volumes:
      - ./code
      - logvolume01:/var/log
    links:
      - redis
  redis:
    image: redis
volumes:
  logvolume01: {}
```

# 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

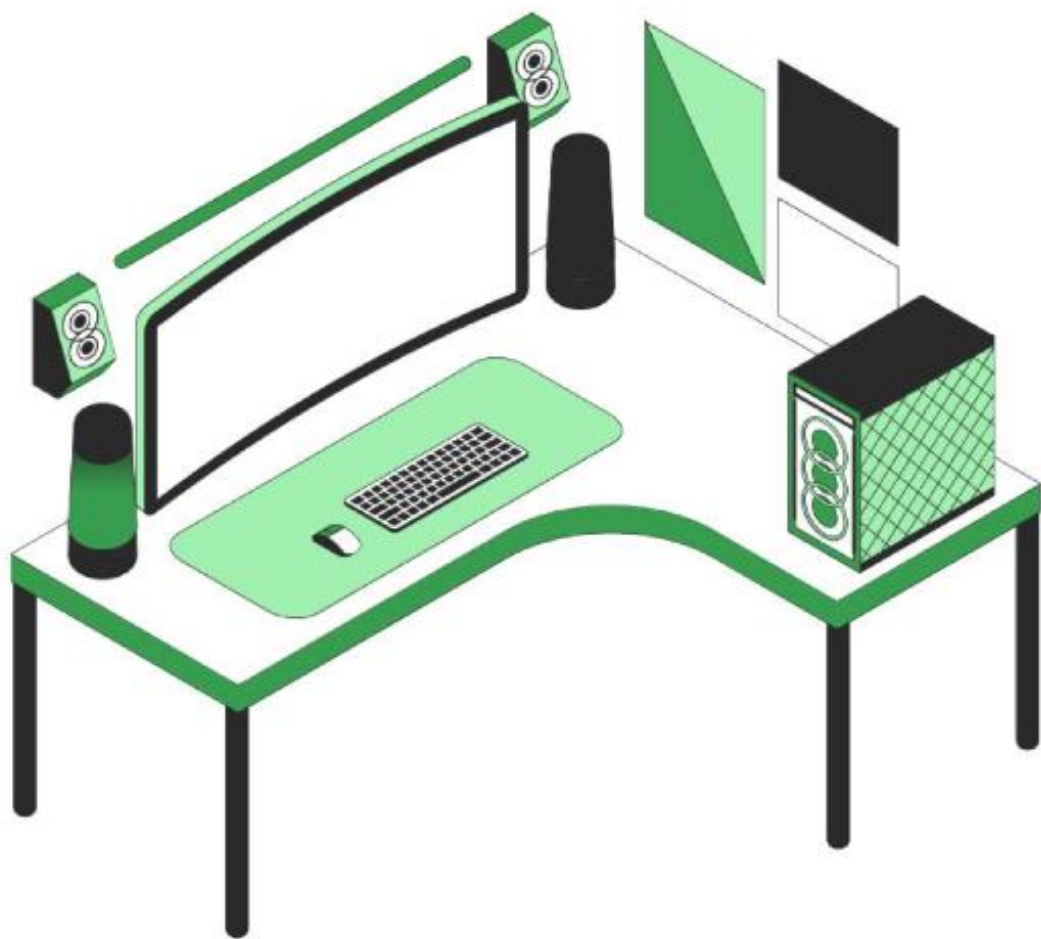


# Docker-compose Commands

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



# Do you have any questions?

Send it to us! We hope you learned something new.