



BATCH : BATCH 48
LESSON : Docker
DATE : 12.04.2022
SUBJECT : Docker Compose



techproeducation



techproeducation



techproeducation



techproeducation



techproedu



Table of Contents

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



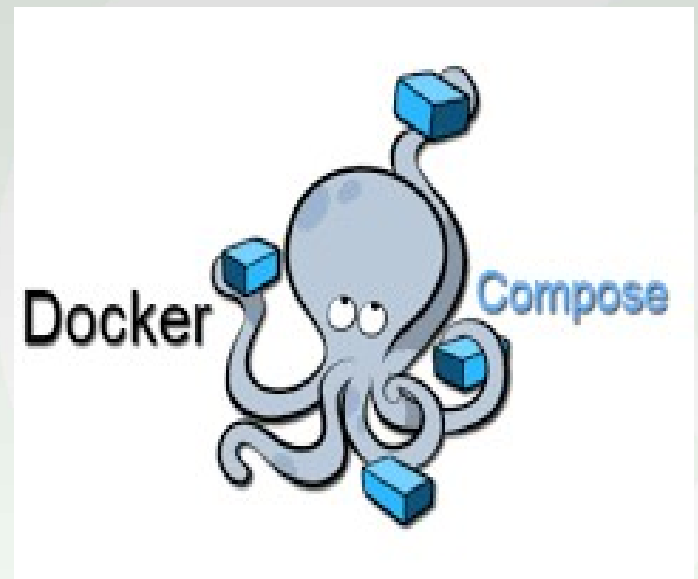
What is Docker Compose?





What is Docker Compose?

- ✓ Compose is tool for defining and running multi-container Docker applications. With Compose, you use a YAM 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 workflows.





Using Compose





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





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: '2' #(1)

#(2)
services:
  server_a:
    image: nginx:latest
    volumes:
      - DataVolume: /DataVolume
    ports:
      - "8001:80"

  server_b:
    image: nginx:latest
    ports:
      - "8002:80"

  server_c:
    image: nginx:latest
    ports:
      - "8003:80"
```




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

```
1  services:
2    recommendation-engine:
3      image: ubuntu
4      tty: true
5      volumes:
6      - DataVolume:/DataVolume
7      labels:
8        brownout.feature: "optional"
9      deploy:
10       replicas: 2
11       restart_policy:
12         condition: none
13       placement:
14         constraints: [node.role == worker]
15
16  user-db:
17    image: weaveworksdemos/user-db
18    hostname: user-db
19    deploy:
20      placement:
21        constraints: [node.role == manager]
```



Docker Compose File

```
version: '2'
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 Commands





Docker Compose Commands

Docker-Compose Parameters

`docker-compose [options] [COMMAND]`

`--version, -v` Print version

`--file, -f` Specify an compose file (default: docker-compose.yml)

`--verbose` Show more output

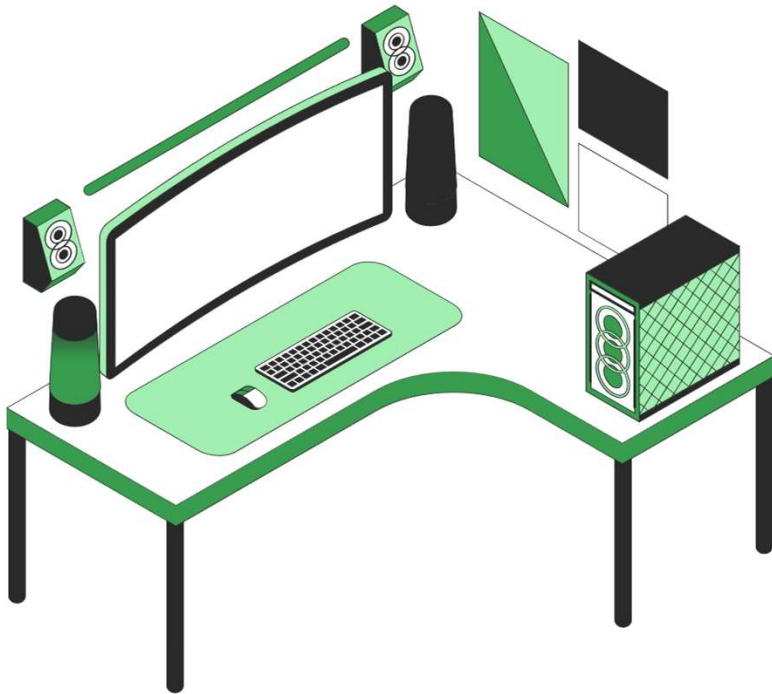
`--log-level LEVEL` Set log level (DEBUG, INFO, WARNING, ERROR, CRITICAL)



Docker Compose Commands

Command Overview

<code>docker-compose up [OPTIONS]</code>	Starts all containers
<code>--detached, -d</code>	detached mode: Run containers in the background
<code>--force-recreate</code>	Recreate containers even if their configuration and image haven't changed
<code>--remove-orphans</code>	Remove containers for services not defined in the Compose file
<code>docker-compose down [OPTIONS]</code>	Stops containers and removes containers, networks, volumes, and images created by up
<code>--volumes, -v</code>	Remove named and anonymous volumes
<code>--remove-orphans</code>	Remove containers for services not defined in the Compose file
<code>docker-compose stop [SERVICE]</code>	Stops running containers without removing them
<code>docker-compose kill [SERVICE]</code>	Forces running containers to stop by sending a SIGKILL signal
<code>docker-compose rm [OPTIONS] [SERVICE...]</code>	Removes stopped service containers
<code>--force, -f</code>	Don't ask to confirm removal
<code>--stop, -s</code>	Stop the containers before removing
<code>-v</code>	Remove any anonymous volumes attached to containers
<code>docker-compose pull SERVICE</code>	Pulls an image associated with the SERVICE
<code>docker-compose logs SERVICE</code>	Displays log output from the SERVICE



Do you have any questions?

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



TECHPROED