BATCH : B107 AWS-DevOps

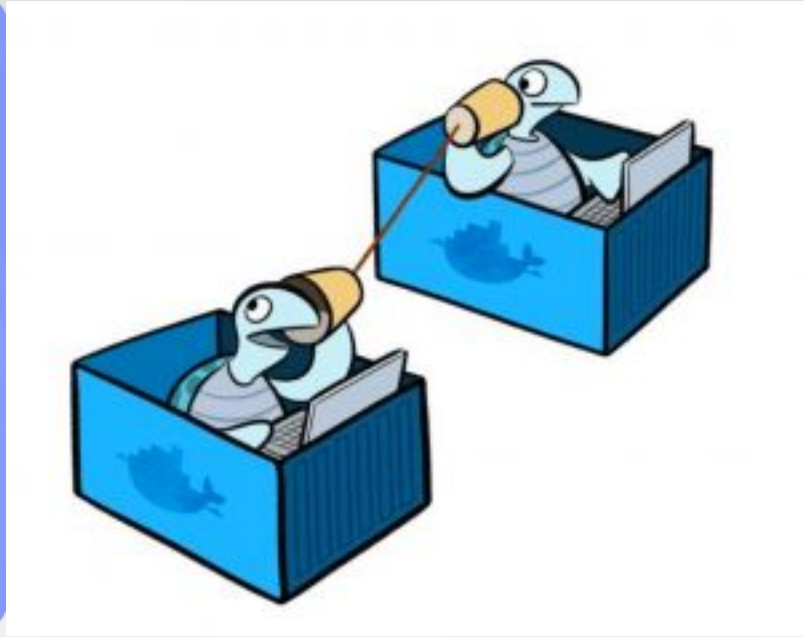
LESSON : **Docker**

DATE : 14.04.2023

SUBJECT : **Docker Compose**

ZOOM GİRİŞLERİNİZİ LÜTFEN **LMS** SİSTEMİ ÜZERİNDEN YAPINIZ

Review

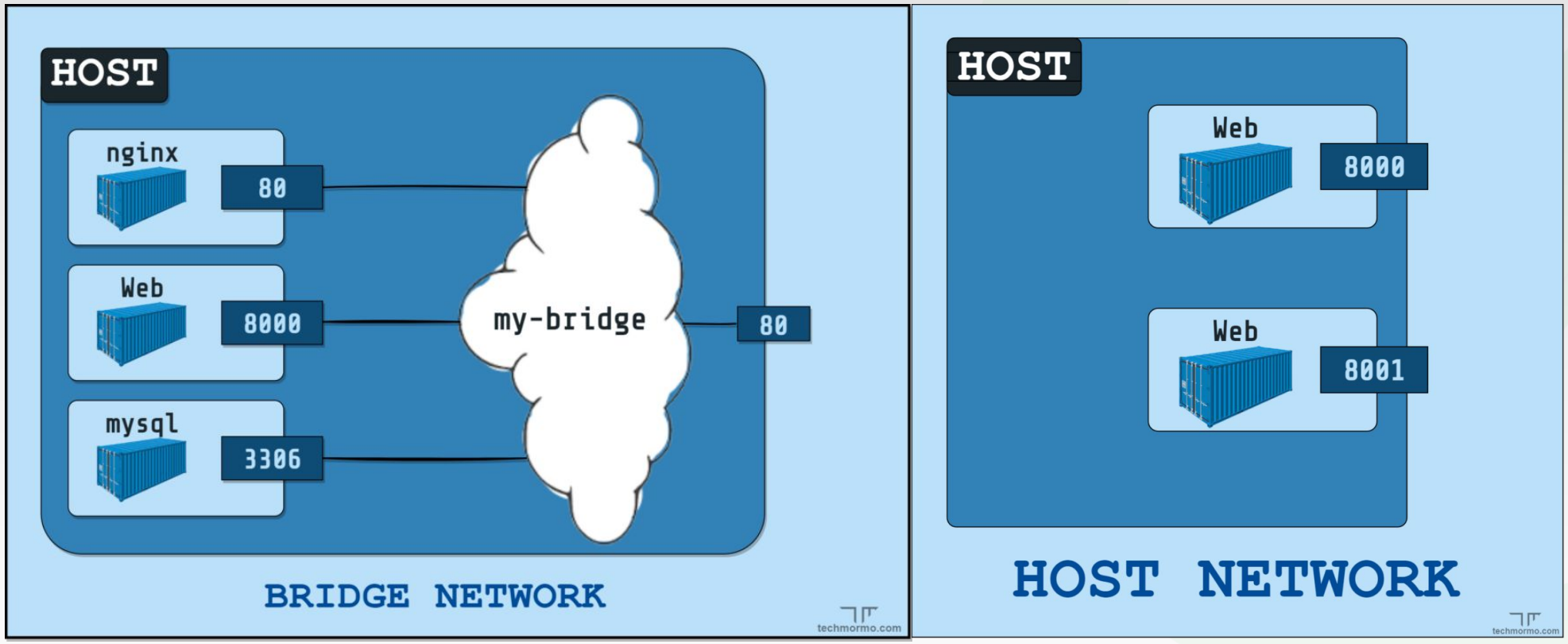


Network Drivers

As default, docker has three network drivers.

- Bridge
- Host
- None

Network Drivers

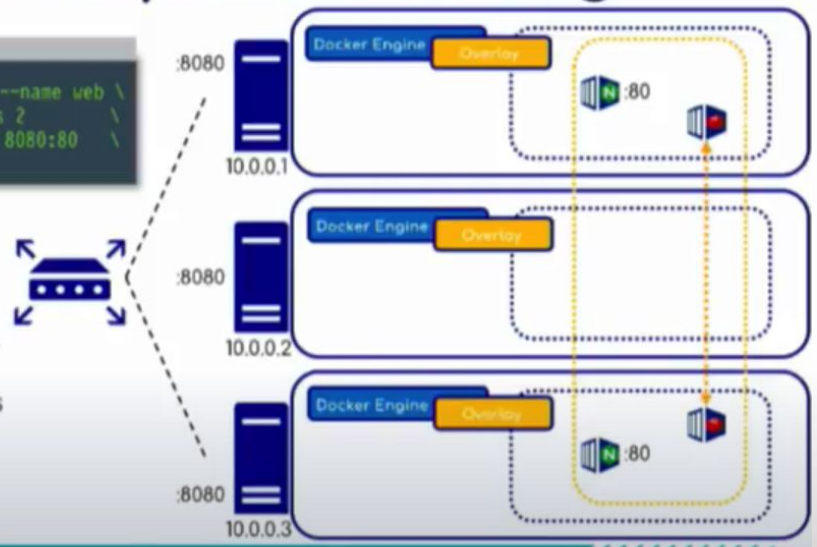


Network Drivers

Swarm Overlay networking

```
[dan@dockercon ~]$ docker service create --name web \
                       --replicas 2          \
                       --publish 8080:80     \
                       nginx
```

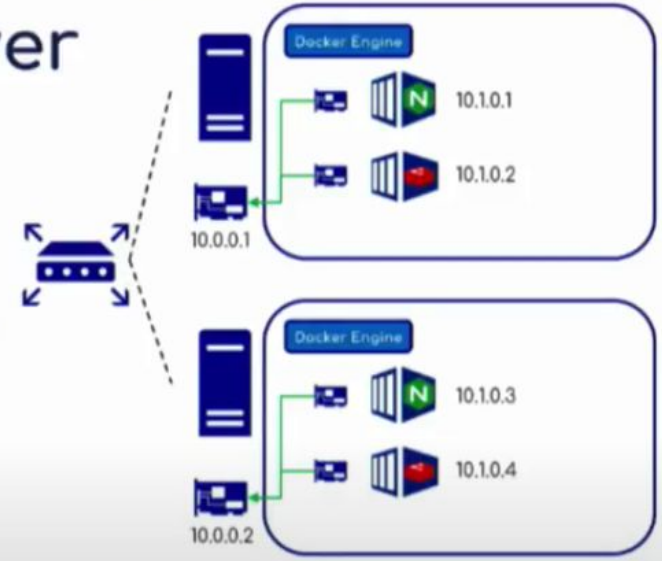
- The Overlay network makes use of VXLAN in order to create an overlay network over the underlying network.
- The tunnel allows containers across hosts to communicate.



dockercon17

Macvlan driver

- The Macvlan driver provides a hardware (MAC) address for each container, allowing them to have a full TCP/IP stack.
- Allows containers to become part of the traditional network, and use things like external IPAM or VLAN trunking when numerous networks are needed.
- No overhead from technologies such as VXLAN or NAT.

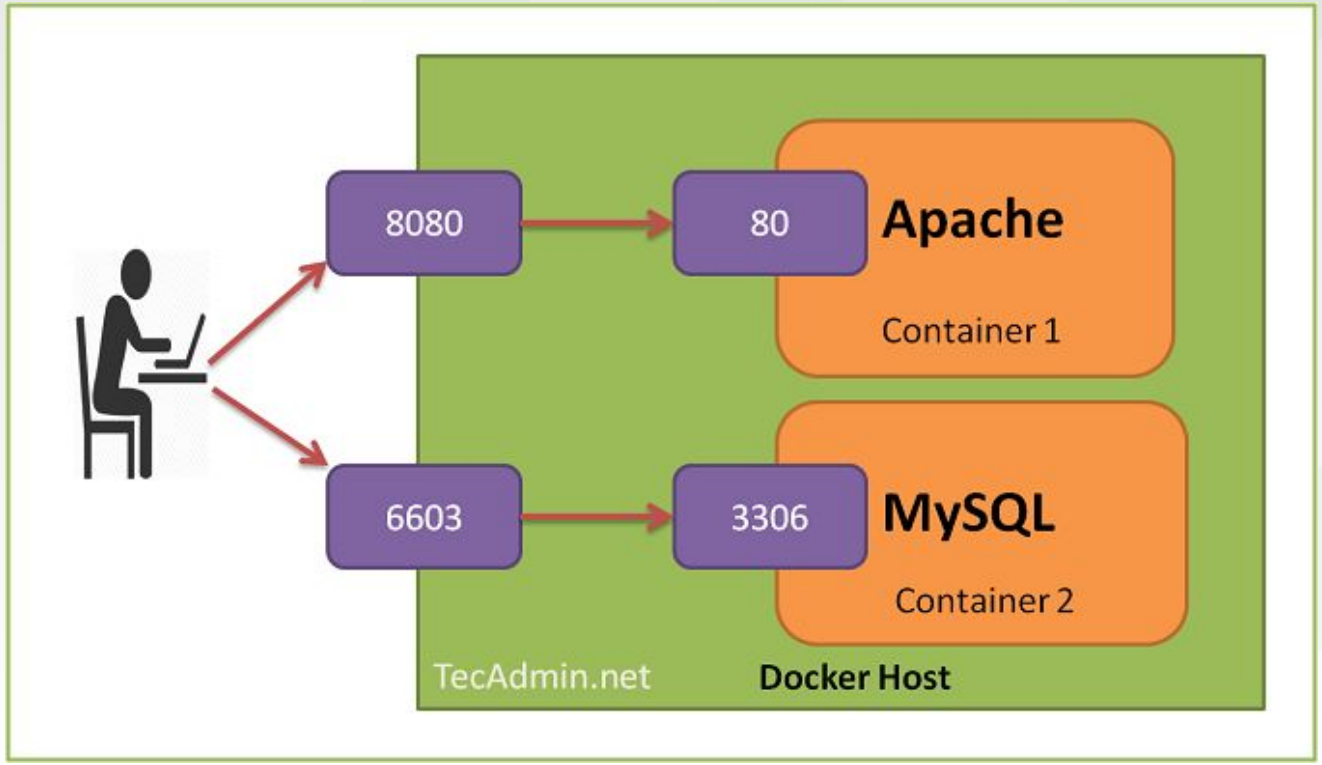


dockercon17

Run – Port Mappings

$ docker run-d-p 8080:80 apache_image

$ docker run d -p 6603:3306 mysql_image

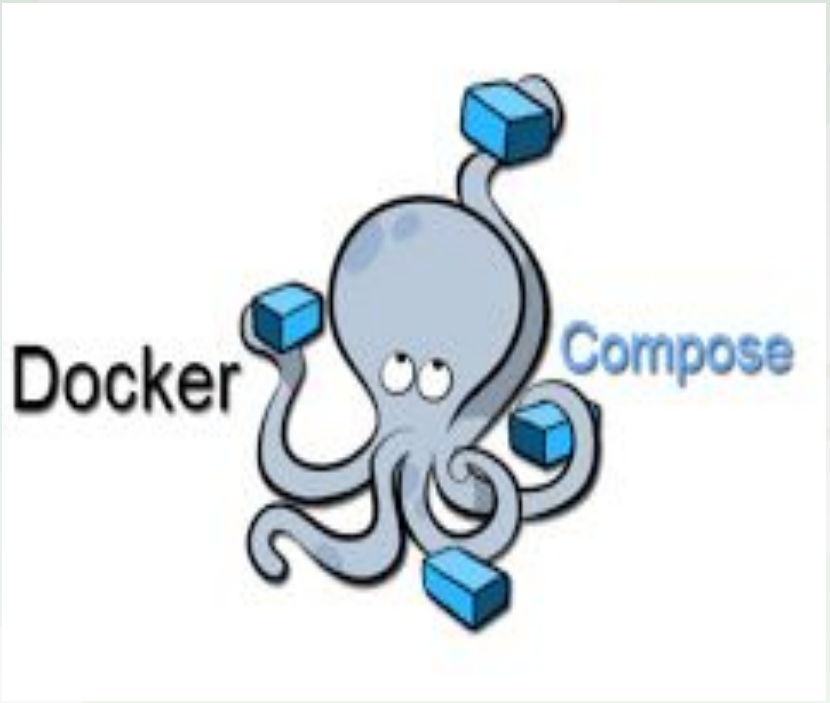


Docker Compose

Table of Contents

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

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

imperative

```
docker run –name=web -p 8080:80 nginx
```

declarative

```
version: “3”
services:
  web:
    image: nginx
    ports:
      - 8080:80
```

Docker Compose File

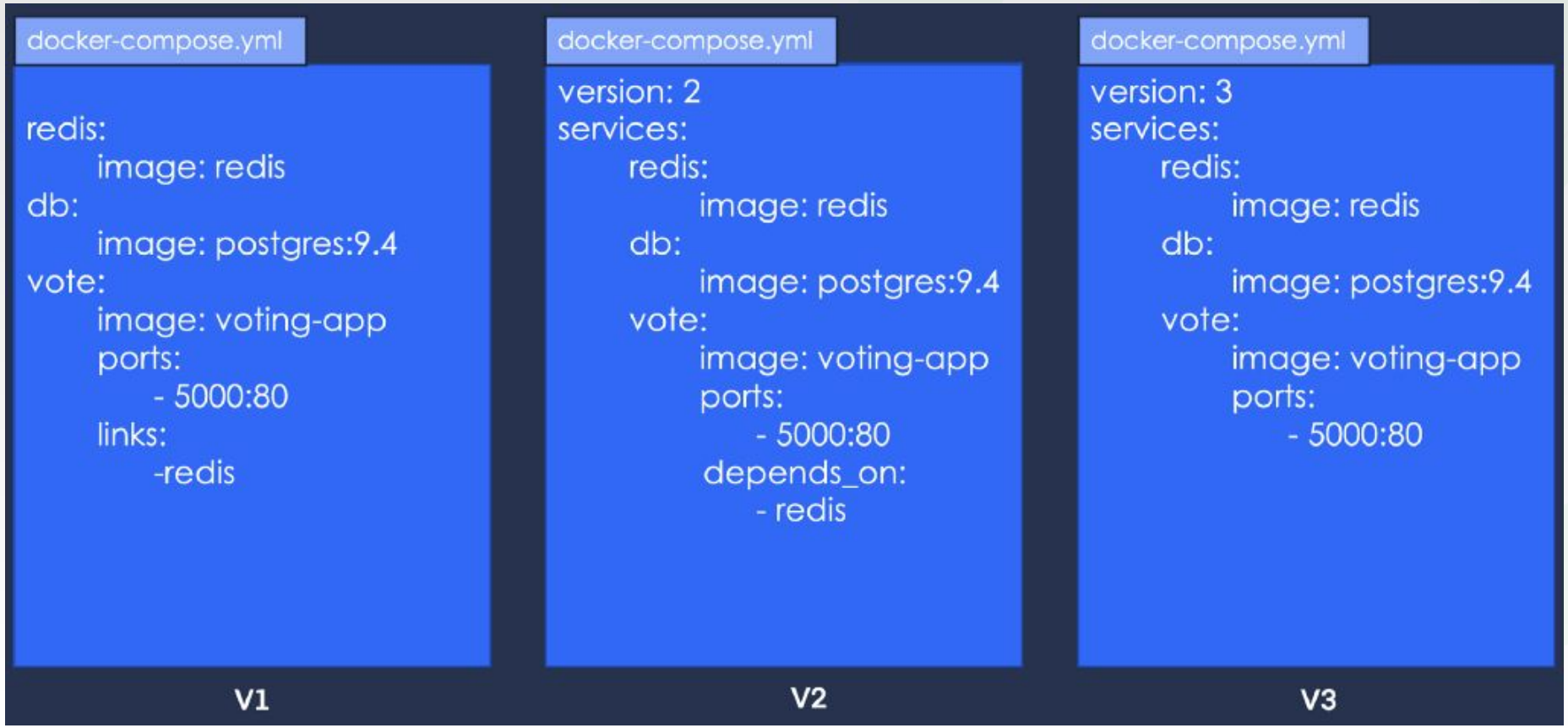
- The Compose file is a YAML file defining:
  - **services**
  - **networks**
  - **volumes**
- The default path for a Compose file is ./docker-compose.yml

```
version: '2'
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 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:
  recommendation-engine:
    image: ubuntu
    tty: true
    volumes:
    - DataVolume:/DataVolume
    labels:
      brownout.feature: "optional"
    deploy:
      replicas: 2
      restart_policy:
        condition: none
      placement:
        constraints: [node.role == worker]

  user-db:
    image: weaveworksdemos/user-db
    hostname: user-db
    deploy:
      placement:
        constraints: [node.role == manager]
```

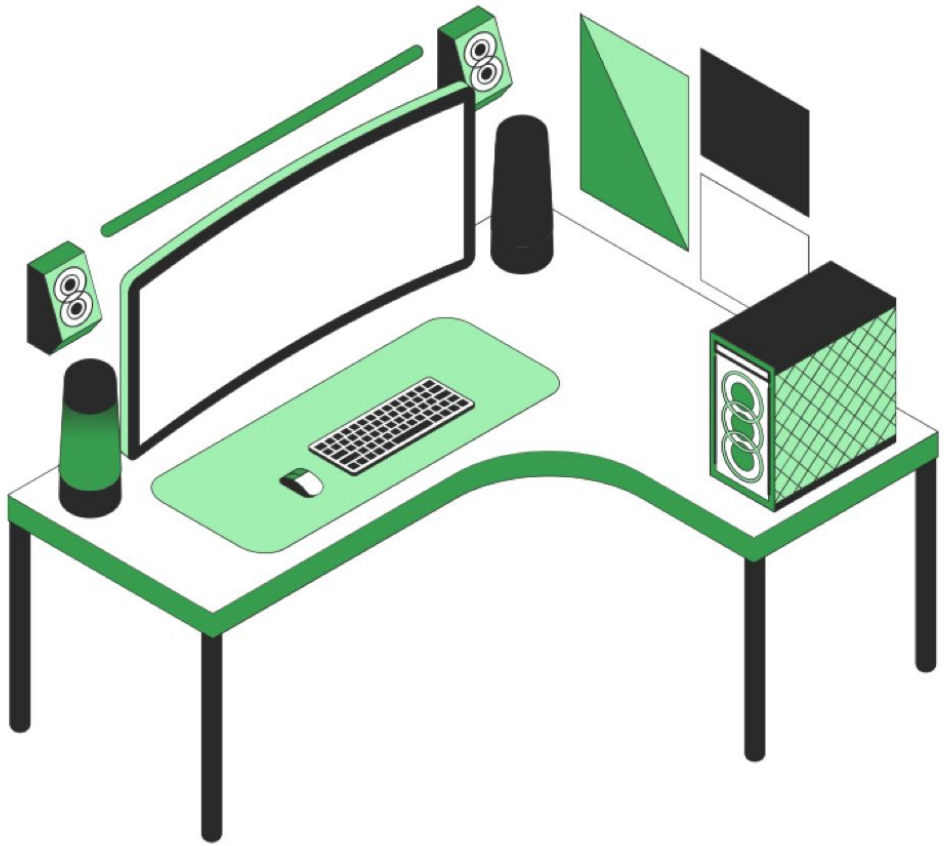
Docker Compose Commands

Docker Compose Commands

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

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)



Do you have any questions?

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