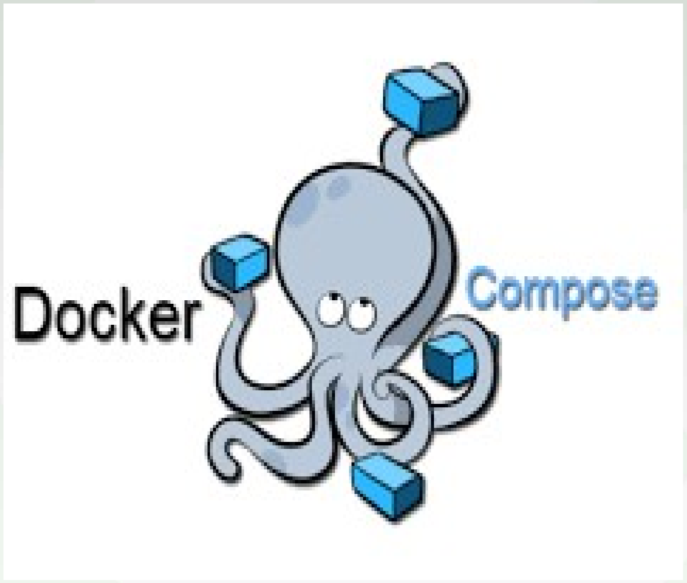
BATCH	BATCH 85
LESSON	Docker
DATE	01.10.2022
SUBJECT	Docker Compose

Table of Contents

What is Docker Compose?

What is Docker Compose?

- Compose is tool for defining and running muti-container Docker applications. With Compose, you use a YAM file to configure your application's services. hen, with a single command, you create and start all the services fom your configuration.
- Compose works in all environments: production, staging, development, testing, as well as I orkfows.



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

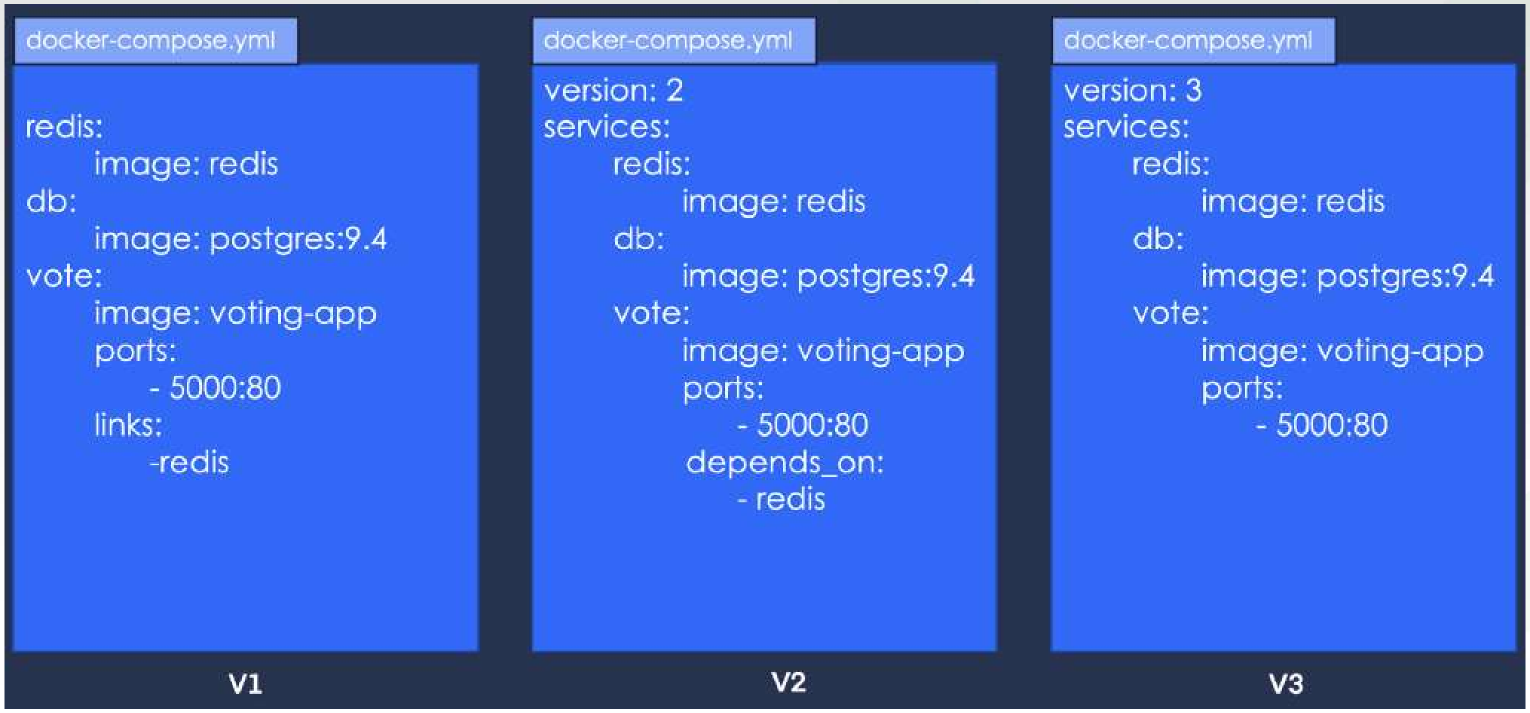
```
version: '2' #(1)

#(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

```
redis:
      image: redis
db:
      image: postgres:9.4
vote:
      image: voting-app
      ports:
            - 5000:80
      links:
            -redis
```

V1

docker-compose.yml

```
version: 2
services:
      redis:
            image: redis
      db:
            image: postgres:9.4
      vote:
            image: voting-app
            ports:
                  - 5000:80
            depends_on:
                  - redis
```

V2

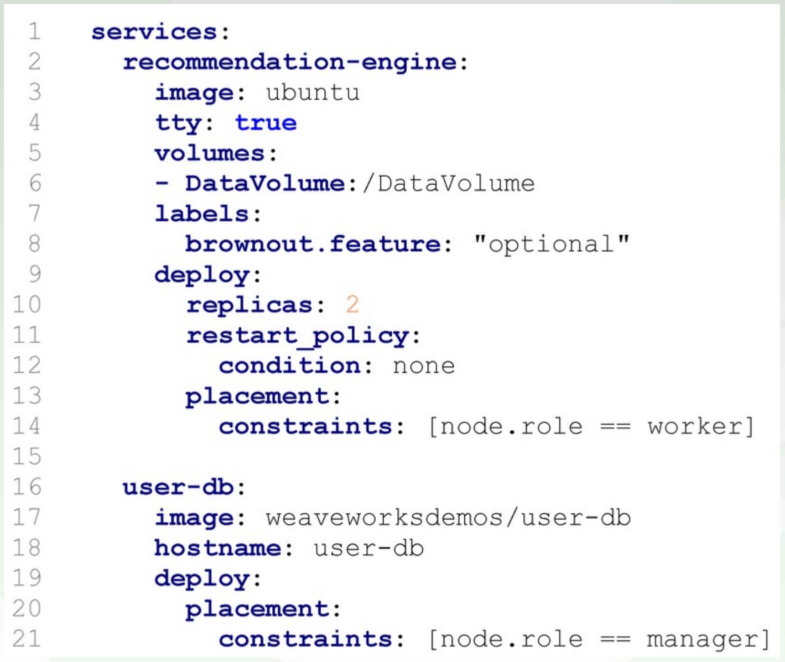
docker-compose.yml

```
version: 3
services:
      redis:
            image: redis
      db:
            image: postgres:9.4
      vote:
            image: voting-app
            ports:
                  - 5000:80
```

V3

Docker Compose File

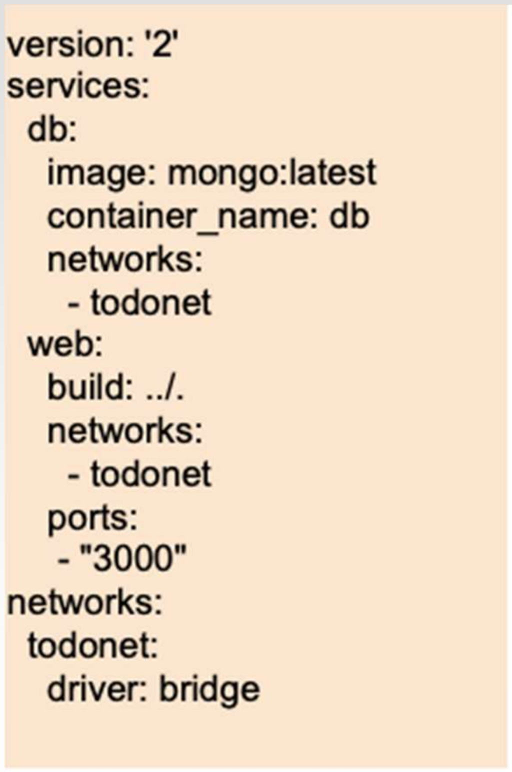
- A service definition contains configuration that is applied to each container started for that service, much like passing command-line parameters to ***docker run***.
- Likewise, network and volume definitions are analogous to docker network create and docker volume create



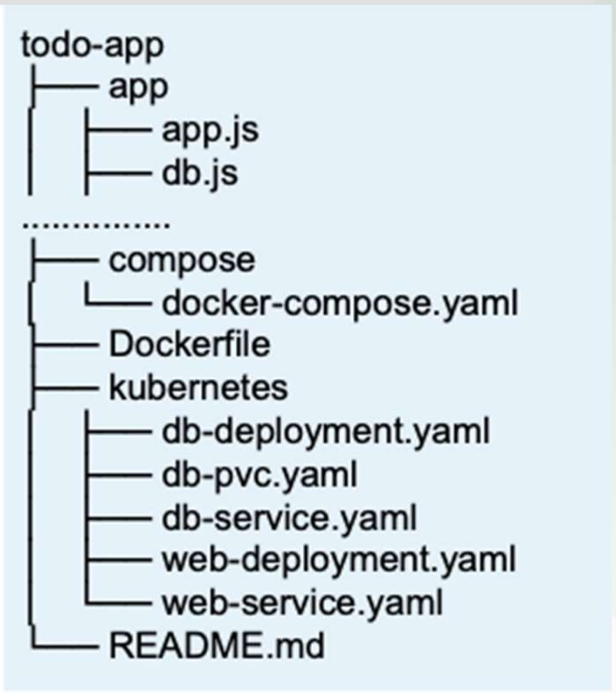
```yaml
services:
  recommendation-engine:
    image: ubuntu
    tty: true
    volumes:
    - DataVolume:/DataVolume
    labels:
      brownout.feature: "optional"
    deploy:
      replicas: 2
      restart_policy:
        condition: none
      placement:
        constraints: [node.role == worker]

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

Docker Compose File



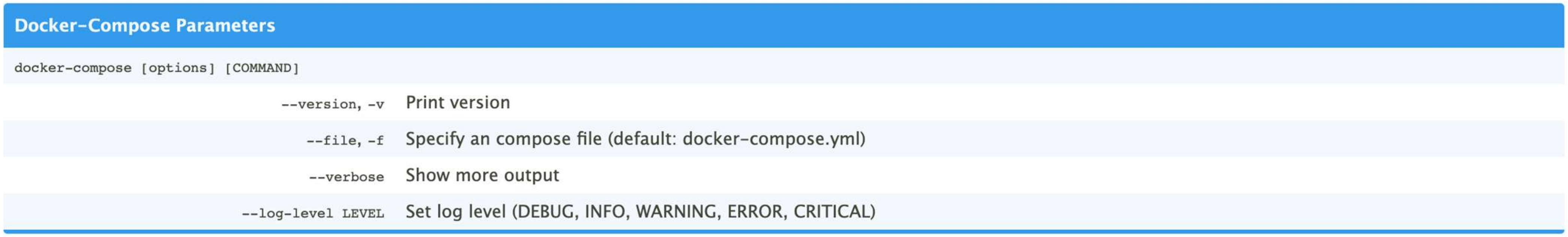
```
version: '2'
services:
  db:
    image: mongo:latest
    container_name: db
    networks:
      - todonet
  web:
    build: ../.
    networks:
      - todonet
    ports:
     - "3000"
networks:
  todonet:
    driver: bridge
```



```
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

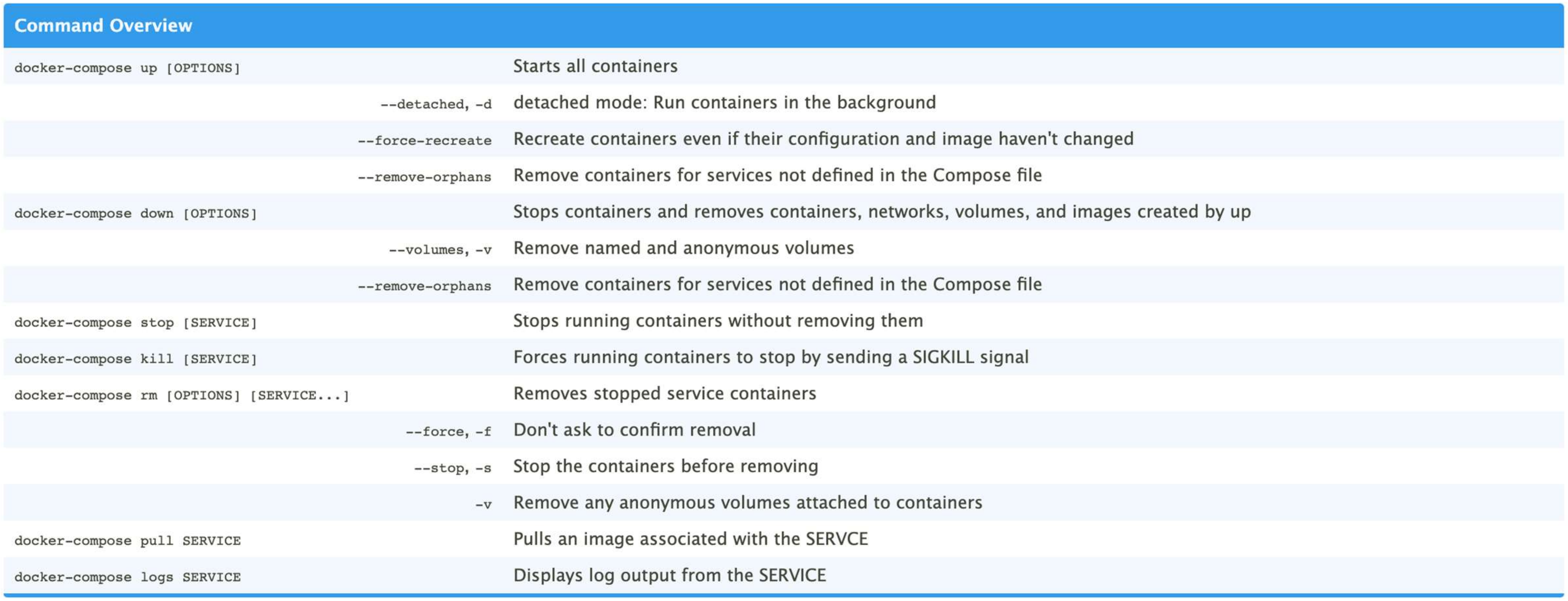


Docker-Compose Parameters

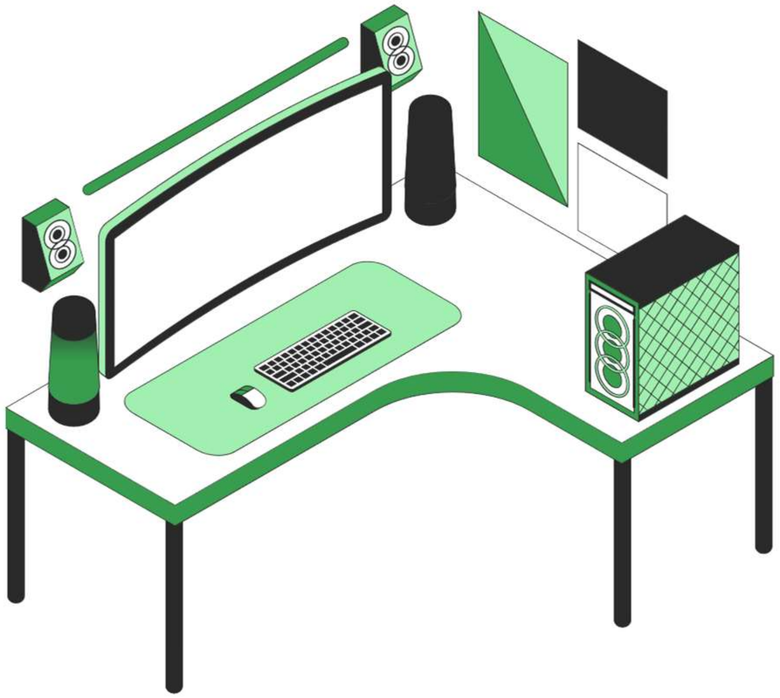
```
docker-compose [options] [COMMAND]
```

`--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	
`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



Do you have any questions?

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

TECHPROED