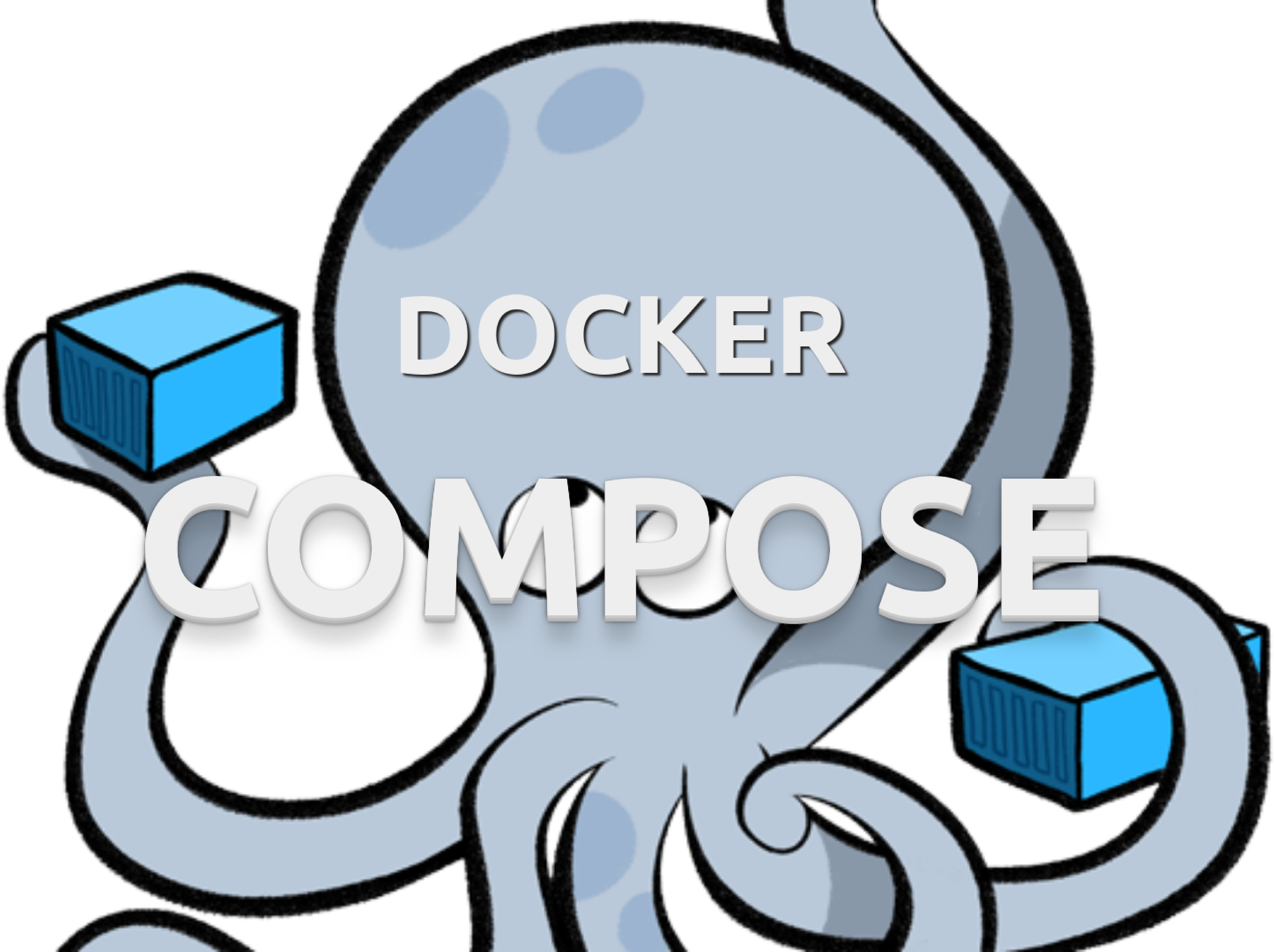
DOCKER COMPOSE



WHY DOCKER COMPOSE?

- Applications/services can contain several components
- Best practice to package only one component in a container
  - Reusability
- Launching and linking containers by hand is annoying
- First step to more complex deployments (Docker Swarm)
  - Scaling

TUTORIAL - MYSQL

- Install & cleanup

```
$ sudo apt install docker-compose
$ sudo systemctl stop mysql
```

- MySQL **5.7** (instead of latest)

```
$ sudo mkdir /home/db
$ docker run -d -p3306:3306 -e MYSQL_ROOT_PASSWORD=secret \
    -v /home/db:/var/lib/mysql \
    --name my-mysql mysql:5.7
$ docker ps
$ mysql -h 0.0.0.0 -u root -p
$ docker stop my-mysql
$ docker ps -a
$ docker start my-mysql
```

TUTORIAL - PHPMYADMIN

- PHPMyAdmin

```
$ docker run -d --link my-mysql:db -p 1080:80 \
    --name myadmin phpmyadmin/phpmyadmin
$ docker stop myadmin
$ docker ps -a
```

- Together with MySQL

```
$ docker start my-mysql
$ docker start myadmin
$ docker ps -a
```

- Cleanup

```
$ docker rm my-mysql
$ docker rm myadmin
$ docker ps -a
```

TUTORIAL - DOCKER COMPOSE

- Auto-name containers
- Similar commands than `docker` that applies to a group of containers
- docker-compose.yml
  - Several (incompatible) "versions"
    - v3 is compatible with Docker Swarm
    - Use v2 or v3
  - YAML format

```
men: [John Smith, Bill Jones]
women:
  - Mary Smith
  - Susan Williams
```

TUTORIAL - DOCKER COMPOSE

- docker-compose.yml

```yaml
version: '2'
services:
  db:
    image: mysql:5.7
    environment:
      MYSQL_ROOT_PASSWORD: secret
    volumes:
      - /home/db:/var/lib/mysql
  phpmyadmin:
    image: phpmyadmin/phpmyadmin
    depends_on:
      - db
    ports:
      - "1080:80"
```

TUTORIAL - DOCKER COMPOSE

- run

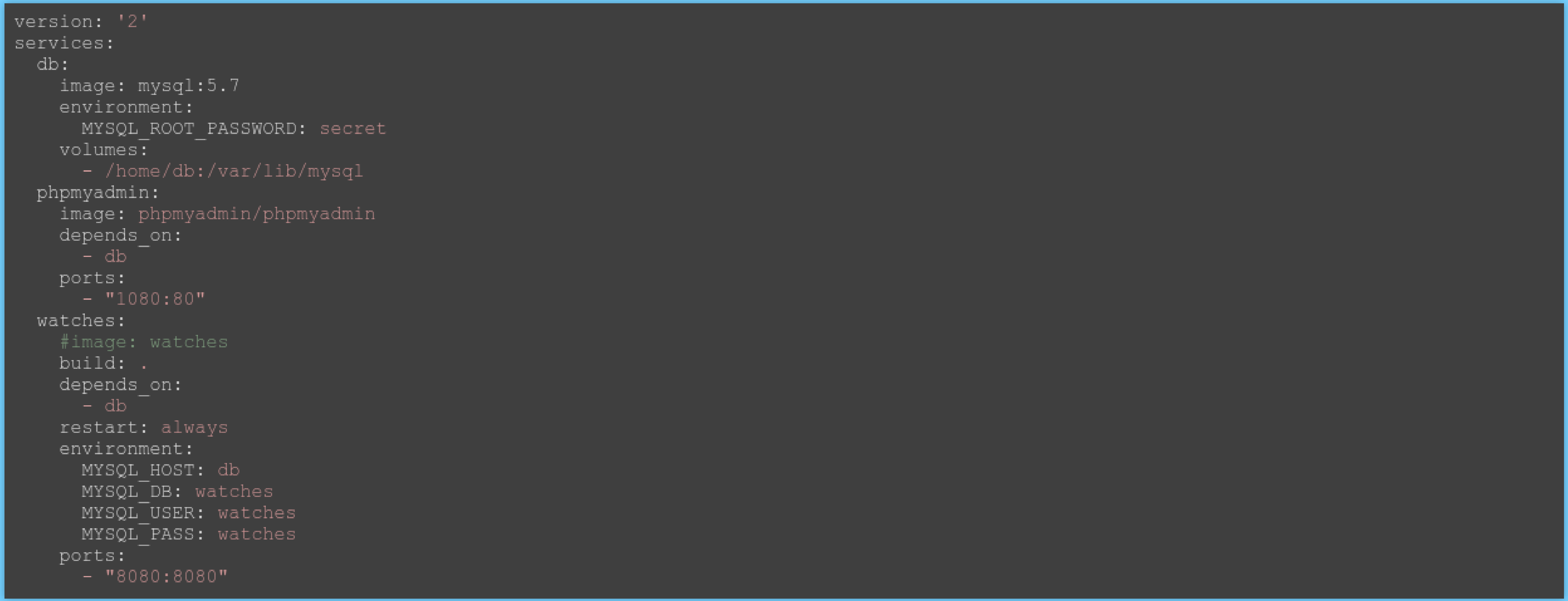
```
$ docker-compose up
$ docker-compose down
Stopping tmp_phpmyadmin_1 ... done
Stopping tmp_db_1         ... done
Removing tmp_phpmyadmin_1 ... done
Removing tmp_db_1         ... done
Removing network tmp_default
```

- run (keeping state)

```
$ docker-compose restart
$ docker-compose start
$ docker-compose stop
$ docker-compose logs -f
```

TUTORIAL - ADD A MICROSERVICE

- Extend compose file
  - Add service
    - image or build
    - Set environment
    - restart: always



```yaml
version: '2'
services:
  db:
    image: mysql:5.7
    environment:
      MYSQL_ROOT_PASSWORD: secret
    volumes:
      - /home/db:/var/lib/mysql
  phpmyadmin:
    image: phpmyadmin/phpmyadmin
    depends_on:
      - db
    ports:
      - "1080:80"
  watches:
    #image: watches
    build: .
    depends_on:
      - db
    restart: always
    environment:
      MYSQL_HOST: db
      MYSQL_DB: watches
      MYSQL_USER: watches
      MYSQL_PASS: watches
    ports:
      - "8080:8080"
```