



# Dockerization

*Toward an Agile Infrastructure*

---

Ahmed Hassanien

Lead Software Engineer

 Gabrianoo  Ahmed Hassanien  @aka\_gabrianoo

 Garage Education  @GarageEducation

 eng.ahmedgaber@gmail.com

April 29, 2020

1. Docker - More than one container app

# Docker - More than one container app

---

## Docker default bridge

- What if we have an application with more than one container.

## Docker default bridge

- What if we have an application with more than one container.
- Ex; WordPress rich content management system uses apache httpd and mysql servers.

## Docker default bridge

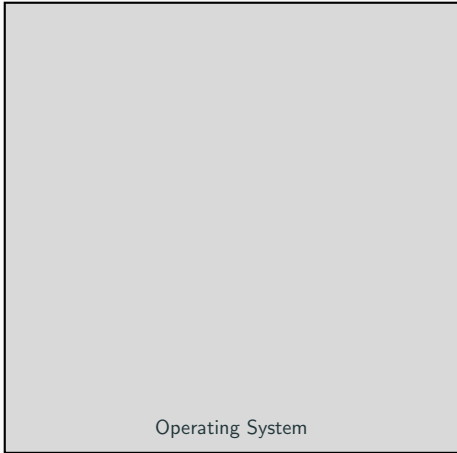
- What if we have an application with more than one container.
- Ex; WordPress rich content management system uses apache httpd and mysql servers.
- `-e, --env list` Set environment variables.

## Docker default bridge

- What if we have an application with more than one container.
- Ex; WordPress rich content management system uses apache httpd and mysql servers.
- `-e`, `--env list` Set environment variables.

```
docker run --name mysql -e MYSQL_ROOT_PASSWORD=P@ssw0rd -d mysql:5.7
docker run --name mysql -p 3306:3306 -e MYSQL_ROOT_PASSWORD=P@ssw0rd \
-d mysql:5.7
hostname
# hostname.lan
docker run -it --rm mysql:5.7 mysql -hhostname.lan -uroot -pP@ssw0rd
```

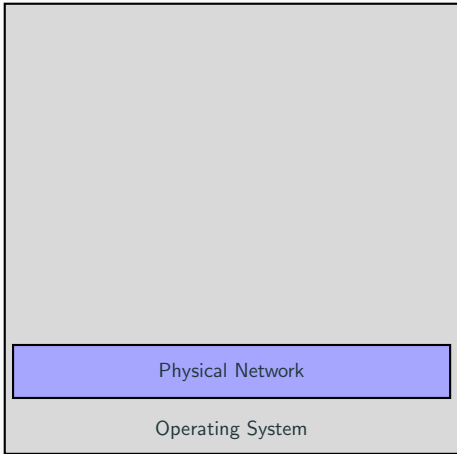
# User-defined bridge



**Figure 1:** User-defined Bridge Networking

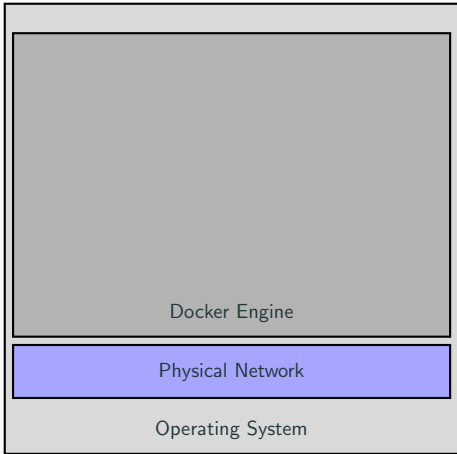


# User-defined bridge



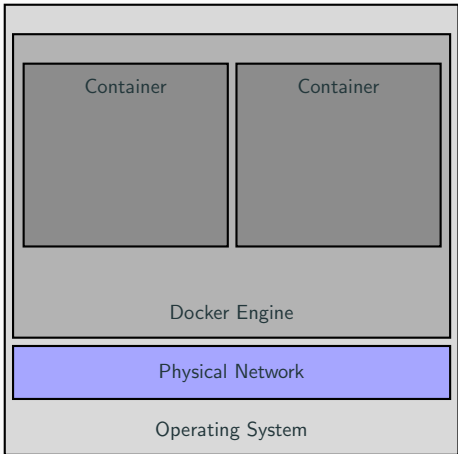
**Figure 1:** User-defined Bridge Networking

# User-defined bridge



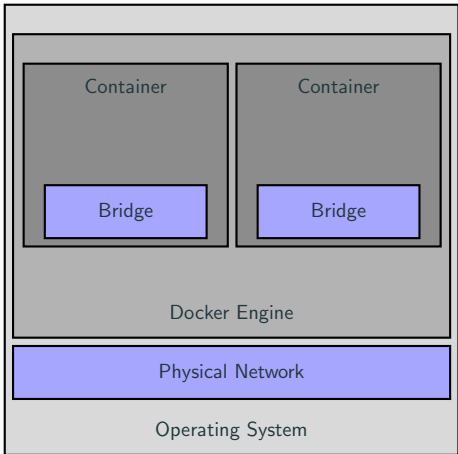
**Figure 1:** User-defined Bridge Networking

# User-defined bridge



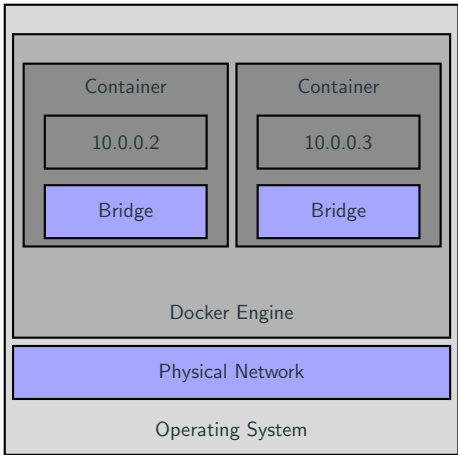
**Figure 1:** User-defined Bridge Networking

# User-defined bridge



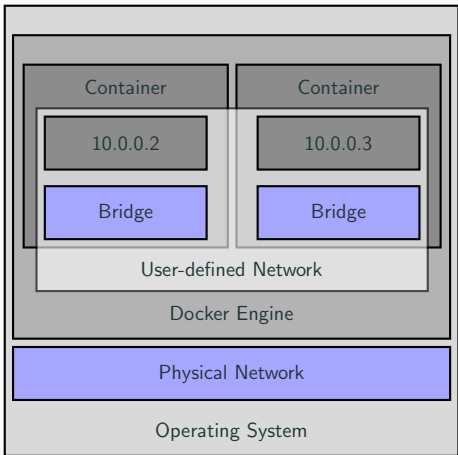
**Figure 1:** User-defined Bridge Networking

# User-defined bridge



**Figure 1:** User-defined Bridge Networking

# User-defined bridge



**Figure 1:** User-defined Bridge Networking

## User-defined bridge

- User-defined bridges provide automatic DNS resolution between containers.

## User-defined bridge

- User-defined bridges provide automatic DNS resolution between containers.
- User-defined bridges provide better isolation.



## User-defined bridge

- User-defined bridges provide automatic DNS resolution between containers.
- User-defined bridges provide better isolation.
- Containers can be attached and detached from user-defined networks on the fly.

## User-defined bridge

- User-defined bridges provide automatic DNS resolution between containers.
- User-defined bridges provide better isolation.
- Containers can be attached and detached from user-defined networks on the fly.
- Each user-defined network creates a configurable bridge.

## User-defined bridge

- User-defined bridges provide automatic DNS resolution between containers.
- User-defined bridges provide better isolation.
- Containers can be attached and detached from user-defined networks on the fly.
- Each user-defined network creates a configurable bridge.
- Linked containers on the default bridge network share environment variables.

## Docker Network Commands

- create Create a network.

## Docker Network Commands

- create Create a network.
- ls List networks.

## Docker Network Commands

- `create` Create a network.
- `ls` List networks.
- `connect` Connect a container to a network.

## Docker Network Commands

- `create` Create a network.
- `ls` List networks.
- `connect` Connect a container to a network.
- `inspect` Display detailed information on one or more networks.

## Docker Network Commands

- `create` Create a network.
- `ls` List networks.
- `connect` Connect a container to a network.
- `inspect` Display detailed information on one or more networks.
- `rm` Deletes one or more networks.



# Docker Network Commands

- `create` Create a network.
- `ls` List networks.
- `connect` Connect a container to a network.
- `inspect` Display detailed information on one or more networks.
- `rm` Deletes one or more networks.

```
docker network create wp-network
docker network connect wp-network mysql
docker network inspect wp-network
docker run --name wordpress --network wp-network -p 8080:80 \
  -e WORDPRESS_DB_HOST=mysql -e WORDPRESS_DB_NAME=wp_db \
  -e WORDPRESS_DB_USER=root -e WORDPRESS_DB_PASSWORD=P@ssw0rd \
  -d wordpress
docker network inspect wp-network
```