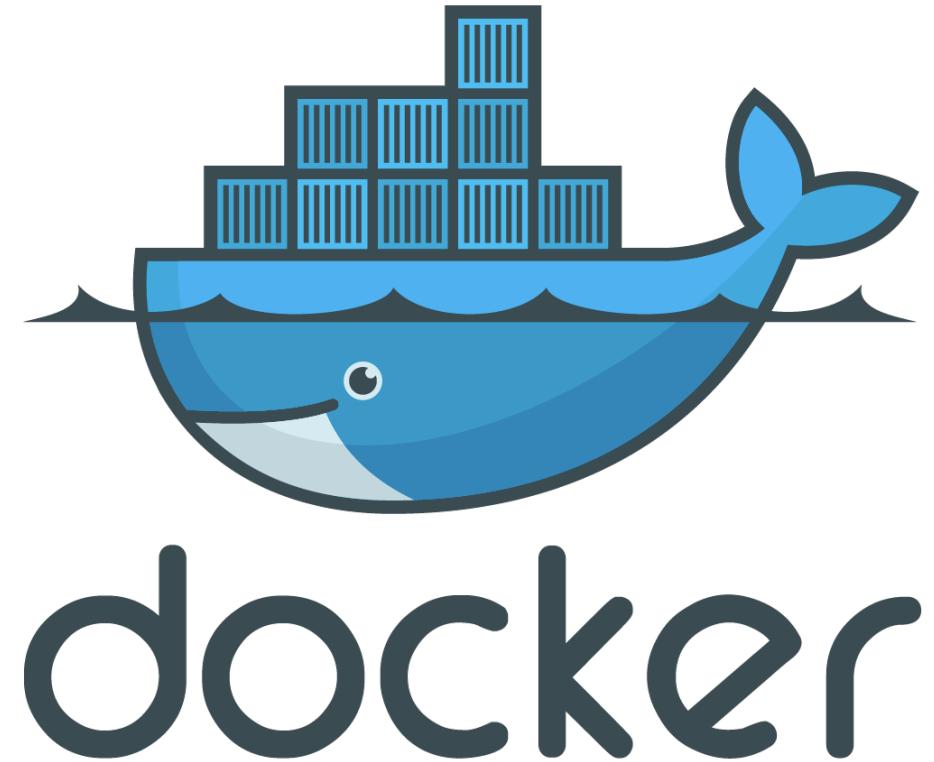

Docker

Nguyễn Hàn Duy

duy@techmaster.vn

Nội dung

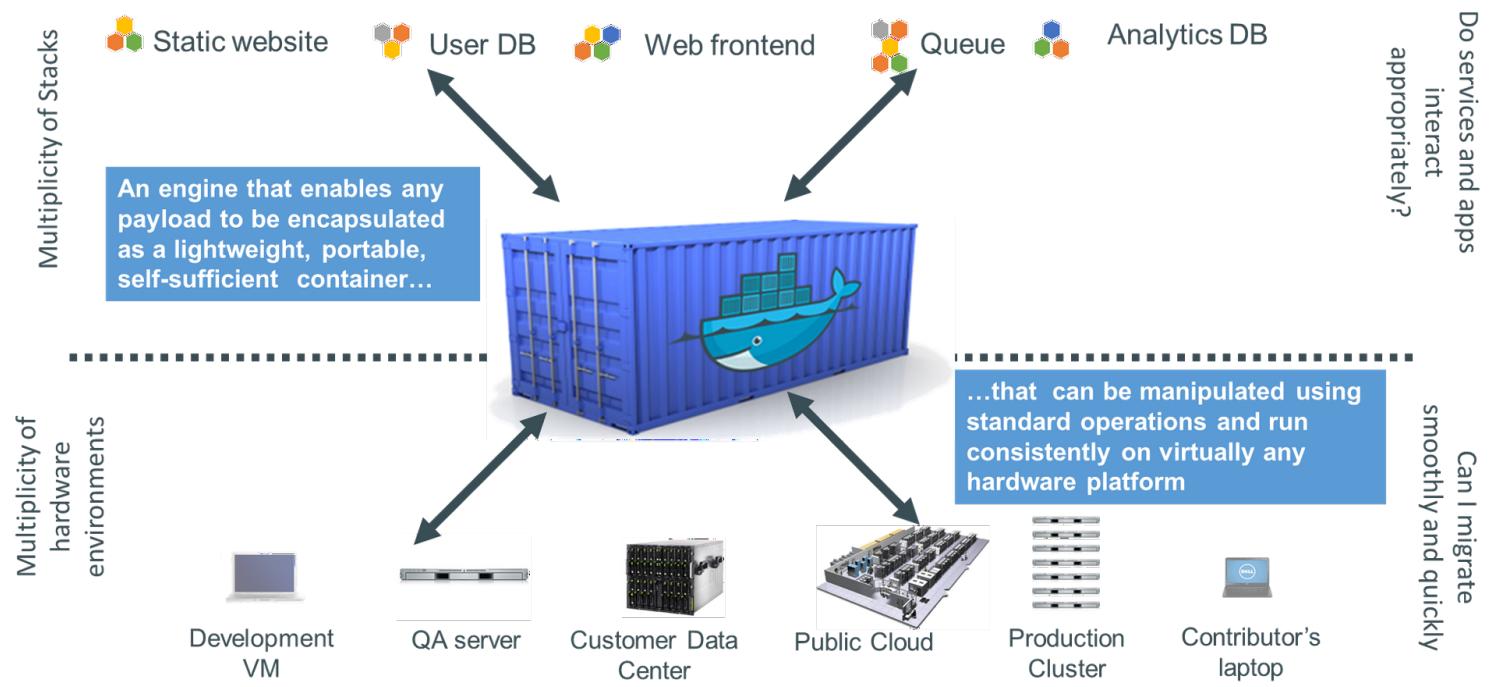
- Giới thiệu
- Cài đặt
- Image & Container
- Bind mount & volume
- Kết nối các container



Giới thiệu

Tác dụng của Docker

- Đóng gói source code vào Docker Image, dễ dàng triển khai và chạy ổn định ở mọi môi trường có Docker
- Cài đặt các ứng dụng một cách dễ dàng, nhanh chóng



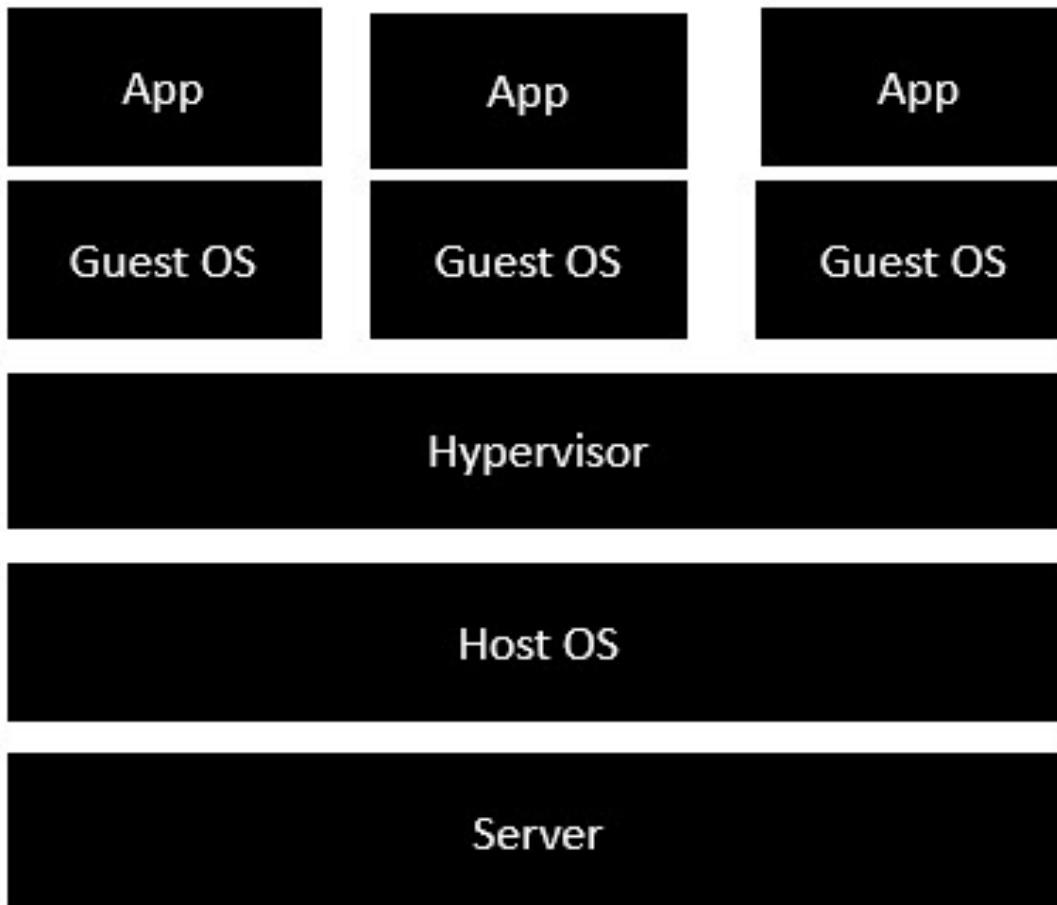
Before Docker

| | Static website | ? | ? | ? | ? | ? | ? | ? |
|--|--------------------|-----------|--------------------|----------------|--------------|----------------------|------------------|---|
| | Web frontend | ? | ? | ? | ? | ? | ? | ? |
| | Background workers | ? | ? | ? | ? | ? | ? | ? |
| | User DB | ? | ? | ? | ? | ? | ? | ? |
| | Analytics DB | ? | ? | ? | ? | ? | ? | ? |
| | Queue | ? | ? | ? | ? | ? | ? | ? |
| | Development VM | QA Server | Single Prod Server | Onsite Cluster | Public Cloud | Contributor's laptop | Customer Servers | |
| | | | | | | | | |

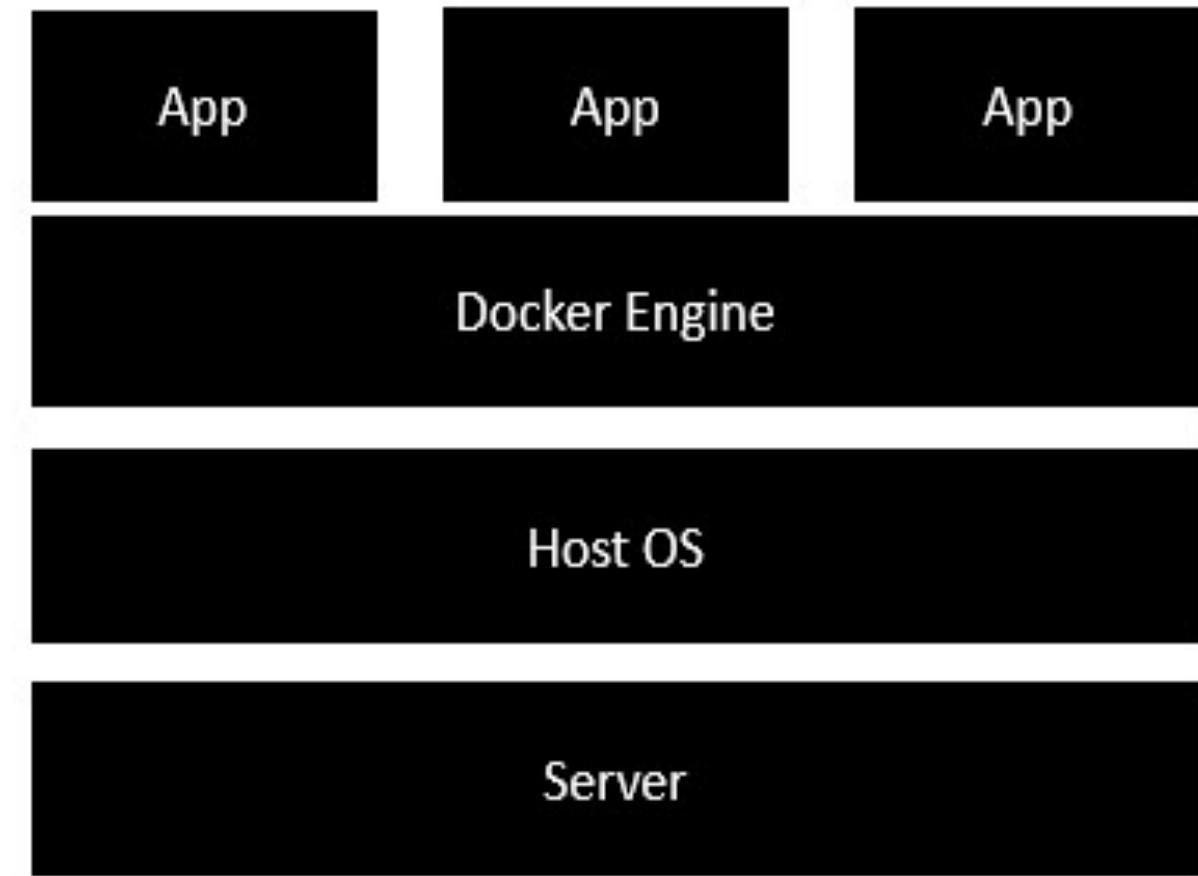
After Docker

| | Static website | | | | | | | |
|--|--------------------|-----------|--------------------|----------------|--------------|----------------------|------------------|--|
| | Web frontend | | | | | | | |
| | Background workers | | | | | | | |
| | User DB | | | | | | | |
| | Analytics DB | | | | | | | |
| | Queue | | | | | | | |
| | Development VM | QA Server | Single Prod Server | Onsite Cluster | Public Cloud | Contributor's laptop | Customer Servers | |
| | | | | | | | | |

VM



Docker



Cài đặt

Cài đặt Docker

- Trên Windows 10 Pro:

https://www.youtube.com/watch?v=d_sjMmxVbz4&feature=emb_lagog&ab_channel=TechMasterVietnam

- Trên MacOS: <https://docs.docker.com/docker-for-mac/install/>

- Trên Ubuntu/Linux: <https://docs.docker.com/engine/install/ubuntu/>

Chạy Docker trên trình duyệt web

Tham khảo: https://www.youtube.com/watch?v=-HINm8FKCY&feature=emb_logo&ab_channel=TechMasterVietnam

1. Đăng ký tài khoản trên Docker Hub: <https://hub.docker.com/>
2. Truy cập trang: <https://labs.play-with-docker.com/>
3. Đăng nhập bằng tài khoản Docker Hub
4. Bấm Start
5. Chọn Add New Instance
6. Kiểm tra bằng lệnh docker --version

Tài liệu học Docker

- Katacoda: <https://www.katacoda.com/courses/docker>
- Slide chi tiết: <https://container.training/intro-selfpaced.yml.html#1>

Image & Container

Docker Image

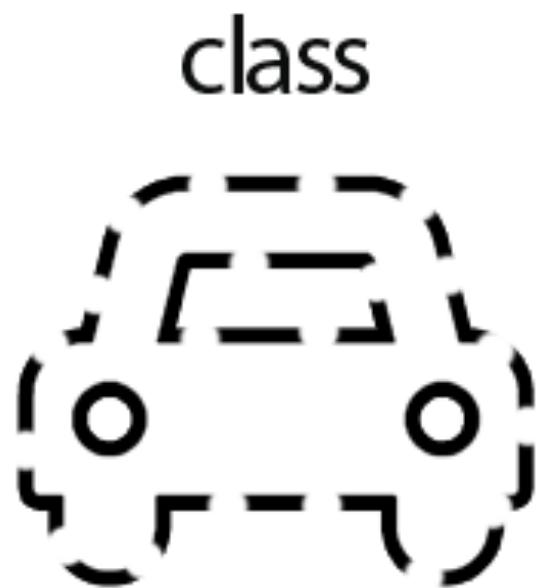
- Read-only
- Chứa source code, libraries, dependencies, tools và các files khác cần thiết để cho một ứng dụng chạy.

```
1 FROM node:alpine
2
3 # Create app directory
4 RUN mkdir -p /usr/src/app
5 WORKDIR /usr/src/app
6
7 # Install NPM
8 COPY package.json /usr/src/app/
9 RUN npm install
10
11 # Bundle app source
12 COPY . /usr/src/app
13 RUN npm run build
14
15 EXPOSE 3000
16
17 CMD [ "npm", "run", "start" ]
```

Docker Container

- Running Docker Image instances

Docker Image



Docker Containers

objects



Task #1: Cài đặt Nginx bằng Docker

docker run nginx

docker run -d --name nginx-1 -p 8081:80 nginx:latest

| Tham số chạy docker run | Ý nghĩa |
|-------------------------|--|
| -d | Chạy container ở chế độ ngầm |
| --name nginx-1 | Đặt tên cho container là nginx-1 |
| -p 8081:80 | Expose cổng 80 của container ra cổng 8081 của máy host |
| nginx:latest | Container được khởi chạy từ image là nginx:latest |

docker run nginx

Truy cập `http://localhost:8081`

localhost:8081

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Docker cheatsheet

- https://design.jboss.org/redhatdeveloper/marketing/docker_cheatsheet/cheatsheet/images/docker_cheatsheet_r3v2.pdf
- <https://dockerlabs.collabnix.com/docker/cheatsheet/>
- <https://phoenixnap.com/kb/list-of-docker-commands-cheat-sheet>
- <https://dev.to/hasone/docker-cheat-sheet-27po>
- <http://dockercheatsheet.painlessdocker.com/>

Task #2: Triển khai nhiều container Nginx

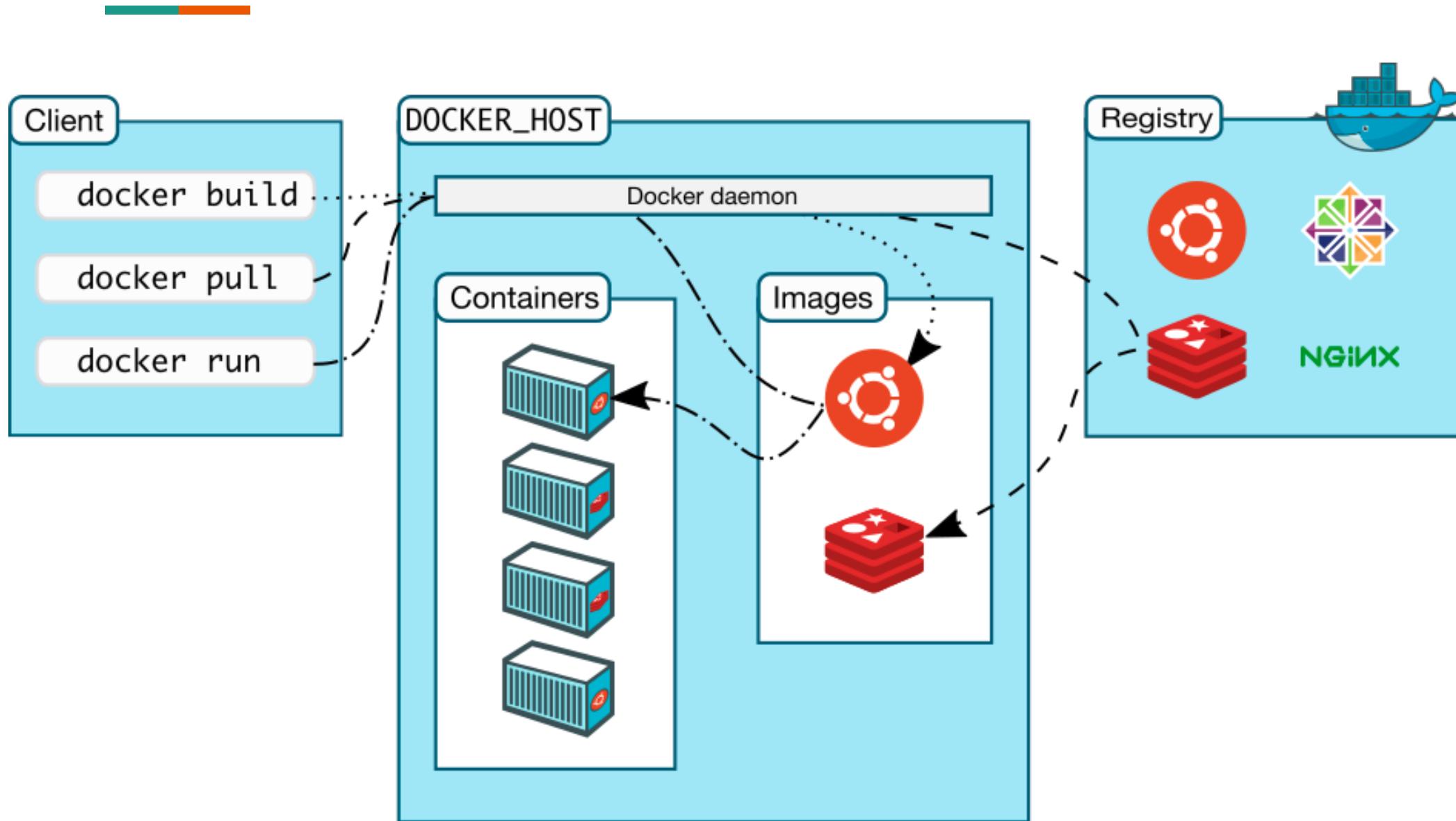
Yêu cầu

Triển khai thêm 2 container:

- Sử dụng image nginx:latest
- Chạy ở chế độ ngầm
- Expose ra 2 cổng khác nhau trên host

Sau khi triển khai, truy cập vào cổng tương ứng trên localhost để kiểm tra

Kiến trúc Docker



Task #3: Triển khai 1 ứng dụng React bằng Docker

Yêu cầu

1. Sử dụng image là **handuy1992/react-app:latest**
2. Chạy container ở chế độ ngầm
3. Tên của container là **react-app**
4. Expose cổng **3000** của container ra cổng **3001** của máy host
5. Sau khi triển khai, truy cập vào localhost:3001 để kiểm tra

Kết quả cần đạt được

A digital calculator interface with a numeric keypad, arithmetic operators, and a display showing '0'. The calculator has a light gray background with orange buttons for operators and a dark gray header bar.

The display shows '0'.

The numeric keypad is arranged in four rows:

- Row 1: AC, +/-, %, ÷
- Row 2: 7, 8, 9, ×
- Row 3: 4, 5, 6, -
- Row 4: 1, 2, 3, +

Below the numeric keypad is a row of buttons: 0, ., and =.

The top header bar includes a 'Fork me on GitHub' button, a search icon, a star icon, and various other icons for sharing and managing the application.

Task #4: Chui vào bên trong 1 container

docker exec vào trong 1 container

- Dùng lệnh **docker exec** để chui vào bên trong 1 container
- Tương tự SSH vào 1 máy ảo VPS
- Thuận tiện cho việc debug trong quá trình triển khai container

Yêu cầu

1. Chạy 1 container nginx ở chế độ ngầm, tên container là nginx-1 expose ra cổng 8081 (nếu đã có rồi thì thôi). Có thể kiểm tra bằng lệnh **docker ps**
2. Chui vào container nginx-1 bằng lệnh:
docker exec -it nginx-1 /bin/bash
3. Cài đặt trình soạn thảo nano (hoặc vi)
4. cd vào thư mục */usr/share/nginx/html* bên trong container nginx-1
5. Mở file index.html bằng nano (hoặc vi) và tiến hành chỉnh sửa
6. Reload lại trang localhost:8081 để kiểm tra

Task #5: Quản lý môi trường Docker bằng Portainer

Portainer

Là một công cụ quản lý Docker bằng giao diện web

Yêu cầu:

1. Cài đặt Portainer bằng Docker theo hướng dẫn tại:

<https://www.portainer.io/installation/>

2. Truy cập localhost:9000 để kiểm tra

Bind mount & volume

Dữ liệu của container

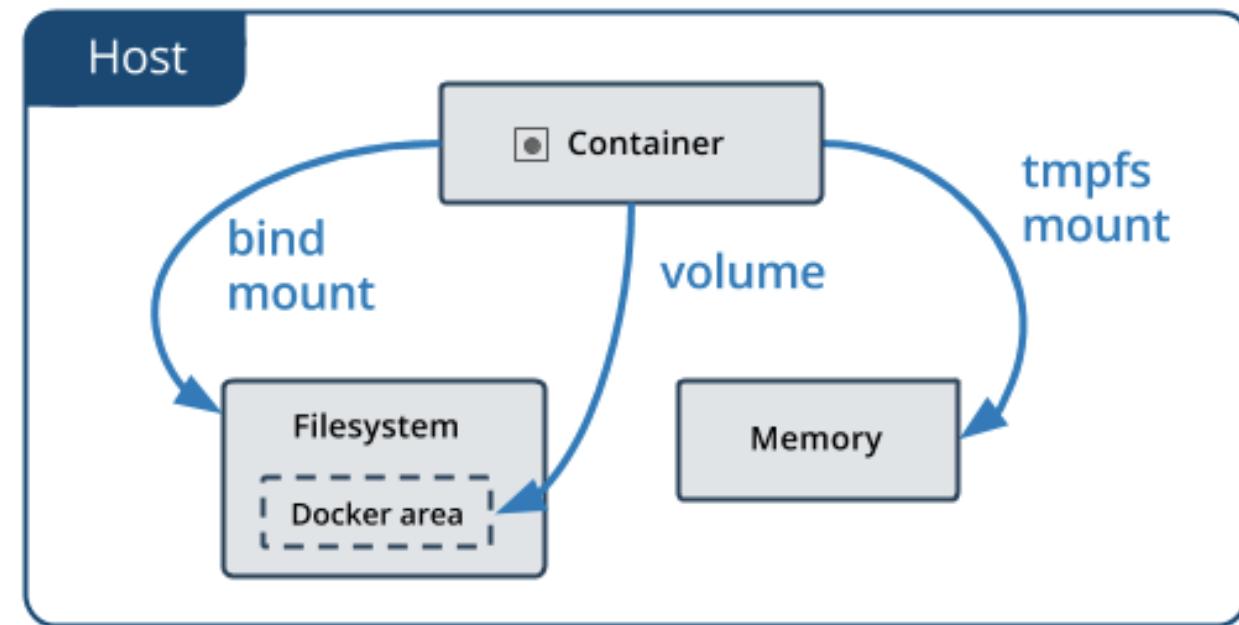
Dữ liệu trong quá trình chạy container được viết vào writeable layer và sẽ bị mất đi khi container bị xóa

Làm sao để giữ data của container ?

Mount một folder từ Docker Container
vào Docker Host

2 kiểu mount:

- **Volume**: Thư mục mount nằm ở `/var/lib/docker/volumes/` của Docker Host
- **Bind mount**: Thư mục mount có thể nằm ở bất kỳ đâu trong Docker Host và không được quản lý bởi Docker.



Kết nối giữa các container

- Mặc định container khi khởi tạo sẽ gắn với 1 default network là bridge
- Chạy lệnh sau để xem cấu hình của 1 container:

docker inspect container-ID

```
...  
"Networks": {  
    "bridge": {  
        "IPAMConfig": null,  
        "Links": null,  
        "Aliases": null,  
        "NetworkID": "e3cbd7ed790ed9c41ecceec287eec0a7b6a7a5896cb46c3763baec23db261cf7"  
        "EndpointID": "109dbbdee7b79d14f20c4e5e52b8dc0059fb4ee11dc3bb3101b2b8c985ee0dd2  
        "Gateway": "172.17.0.1",  
        "IPAddress": "172.17.0.2",  
        "IPPrefixLen": 16,  
        "IPv6Gateway": "",  
        "GlobalIPv6Address": "",  
        "GlobalIPv6PrefixLen": 0,  
        "MacAddress": "02:42:ac:11:00:02",  
        "DriverOpts": null  
    }  
}
```

Các network trong Docker

- Liệt kê các network: **docker network ls**
- Xem thông tin cấu hình 1 network: **docker network inspect tên-network**

```
configonly: false,
"Containers": {
    "a548979539961540ffdd14ca447404b54d53fb370e7604ee52c3e286211cb6b": {
        "Name": "react-app",
        "EndpointID": "109dbbdee7b79d14f20c4e5e52b8dc0059fb4ee11dc3bb3101b2b8c985ee0dd2",
        "MacAddress": "02:42:ac:11:00:02",
        "IPv4Address": "172.17.0.2/16",
        "IPv6Address": ""
    }
},
"Options": {
    "com.docker.network.bridge.default_bridge": "true",
    "com.docker.network.bridge.enable_icc": "true",
    "com.docker.network.bridge.enable_ip_masquerade": "true",
    "com.docker.network.bridge.host_binding_ipv4": "0.0.0.0",
    "com.docker.network.bridge.name": "docker0",
    "com.docker.network.driver.mtu": "1500"
},
"Labels": {}
```

**Task #6: Gọi từ container
này sang container kia trong
bridge network**

Yêu cầu

1. Có ít nhất 2 container đang chạy
2. Xem cấu hình network bridge, lấy ra danh sách các container và IP tương ứng
3. docker exec vào 1 container bất kì và cài gói phần mềm ping
4. ping container khác theo IP của nó
5. ping container khác theo tên của nó

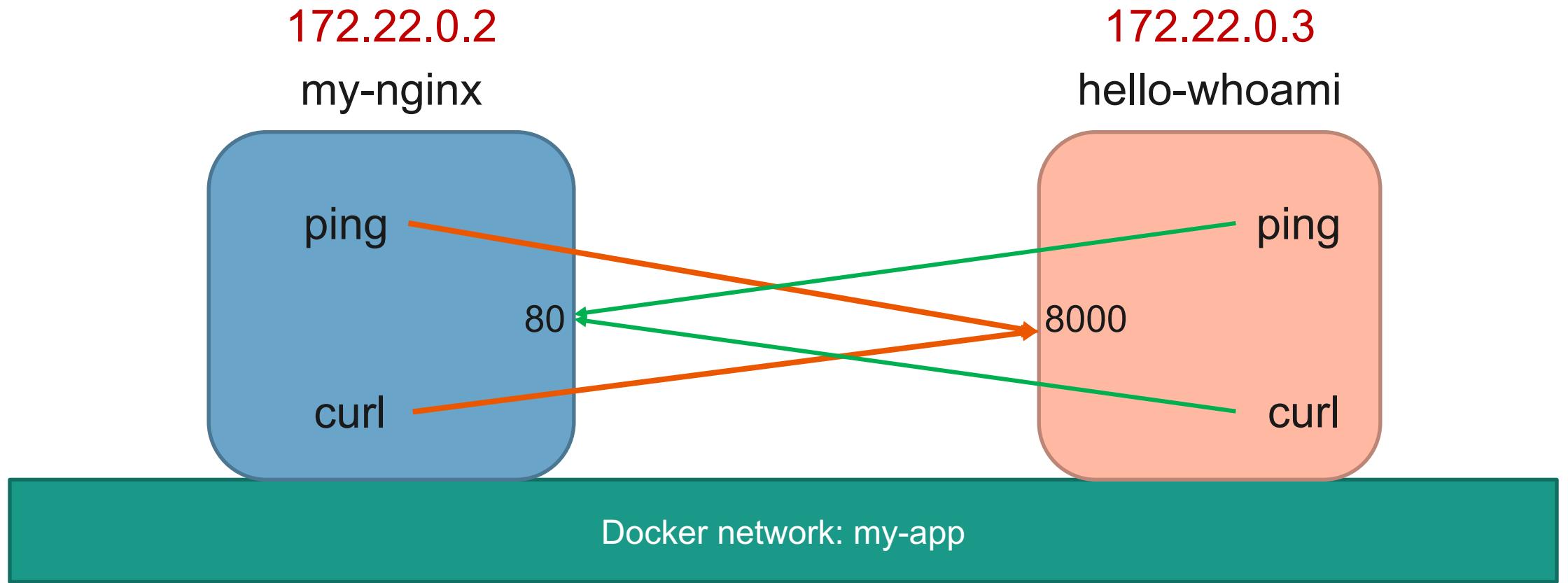
Kết nối các container trong network

- Trong network bridge, các container chỉ có thể liên lạc với nhau qua IP
- Để thuận tiện trong quá trình triển khai, các container cần liên lạc được với nhau qua tên container
- Tạo 1 network riêng: **docker network create** tên-network
- Đặt container vào network:
docker network connect tên-network container-ID
- Hoặc khi khởi chạy container, thêm option: **--network** tên-network

Task #7: Triển khai 2 container trên cùng 1 network

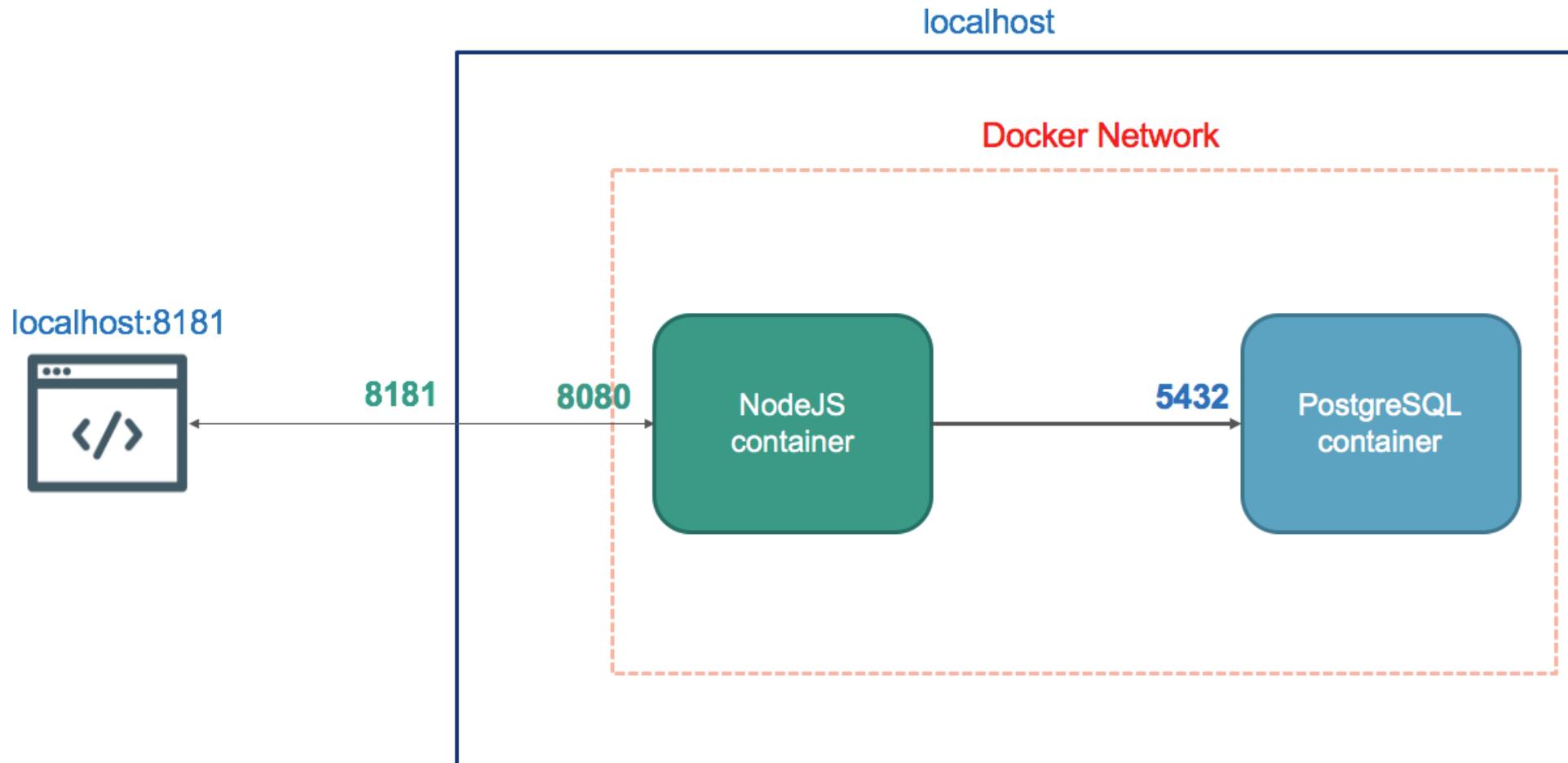
Yêu cầu

1. Tạo 1 docker network có tên là **my-app**
2. Chạy 1 **nginx** container có tên là **my-nginx**, chạy ngầm và đặt trong network my-app
3. Chạy 1 container sử dụng image **jwilder/whoami**
(<https://github.com/jwilder/whoami>), tên container là **hello-whoami**, chạy ngầm và cũng đặt trong network my-app
4. docker exec vào 2 container **my-nginx** và **hello-whoami**, sau đó tiến hành cài 2 gói phần mềm ping và curl (nếu chưa có)
5. Từ bên trong container **my-nginx**, lần lượt chạy 2 lệnh: ping hello-whoami và curl hello-whoami:8000. Kiểm tra kết quả thu được
6. Từ bên trong container **hello-whoami**, lần lượt chạy 2 lệnh: ping my-nginx và curl my-nginx:80. Kiểm tra kết quả



Task #8: Triển khai ứng dụng NodeJS kết nối PostgreSQL

Kiến trúc ứng dụng



Yêu cầu

1. Tạo một docker network có tên là **node-todo**
2. Chạy container PostgreSQL
3. Chạy container cho ứng dụng NodeJS
4. Kiểm tra kết quả

Yêu cầu cụ thể với bước 2, 3, 4 xem ở các slide sau

Chạy container PostgreSQL

Yêu cầu:

1. Chạy ngầm
2. Tên của container là **db**
3. Đặt trong network **node-todo**
4. Mount thư mục **/var/lib/postgresql/data** của container ra ngoài máy host
5. Đặt giá trị cho biến môi trường **POSTGRES_PASSWORD** là **postgres**
6. Sử dụng image là **handuy1992/postgres-12**

Chạy container cho ứng dụng NodeJS

Yêu cầu:

1. Chạy ngầm
2. Đặt trong network **node-todo**
3. Expose cổng **8080** của container ra cổng **8181** trên máy host
4. Sử dụng image là **handuy1992/node-todo:latest**

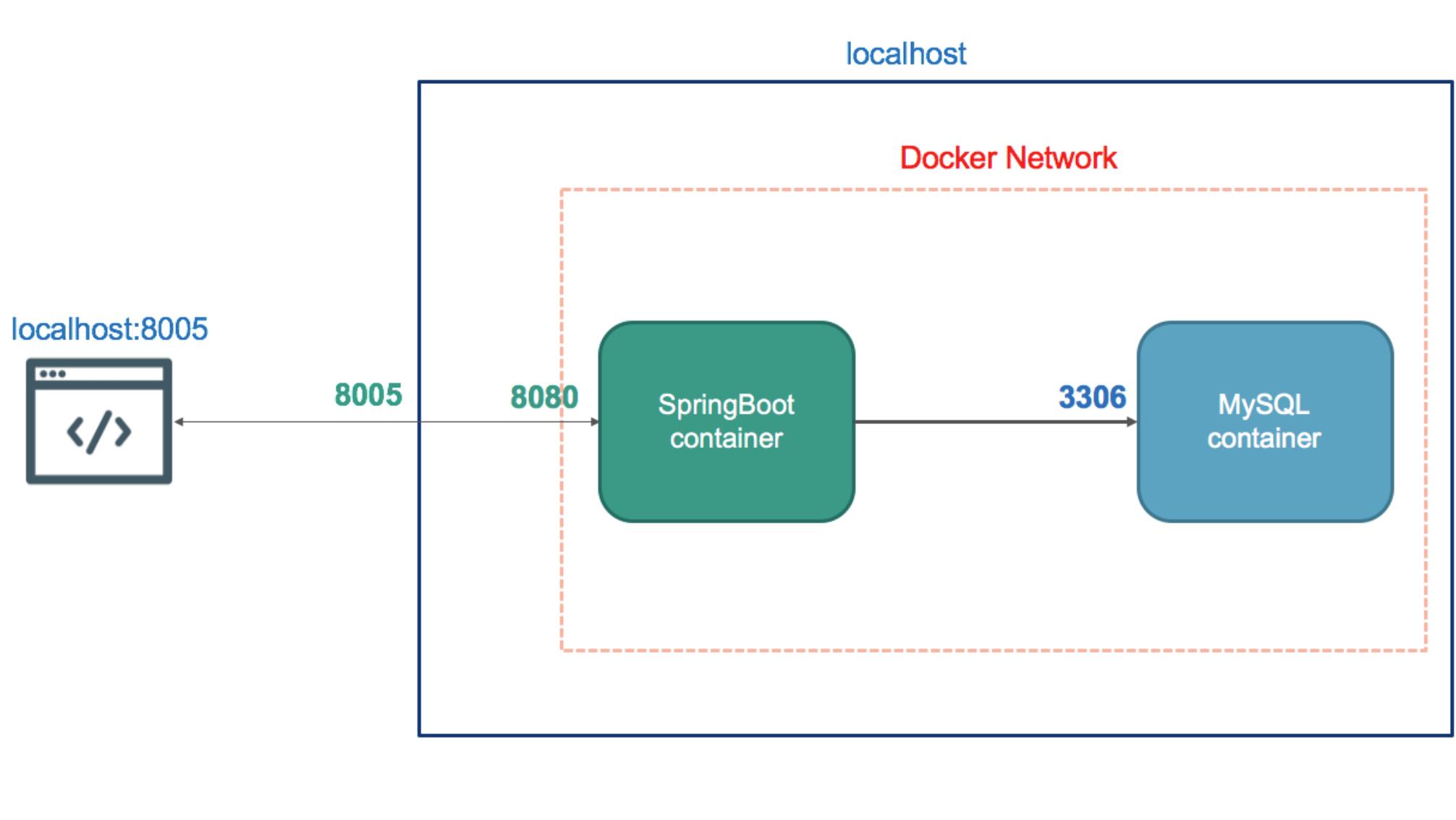
Kiểm tra kết quả

Truy cập <http://localhost:8181>

The screenshot shows a web-based todo application. At the top, there is a header bar with a back arrow, a search icon, a star icon, a notifications icon showing 2 notifications, a refresh icon, and other standard browser controls. The URL 'localhost:8181' is visible in the address bar. Below the header, there is a search bar labeled 'create new task' and a button labeled 'ADD'. The main content area is divided into two sections: 'Todo Items' on the left and 'Done Items' on the right. Both sections currently have an empty list box.

Task #9: Triển khai ứng dụng SpringBoot kết nối MySQL

Kiến trúc ứng dụng



Yêu cầu

1. Tạo một docker network có tên là **spring-app**
2. Chạy container MySQL
3. Chạy container cho ứng dụng SpringBoot
4. Kiểm tra kết quả

Yêu cầu cụ thể với bước 2, 3, 4 xem ở các slide sau

Chạy container MySQL

Yêu cầu:

1. Chạy ngầm
2. Tên của container là **mysql**
3. Đặt trong network **spring-app**
4. Mount thư mục **/var/lib/mysql** của container ra một thư mục bất kỳ trên máy host
5. Đặt giá trị cho biến môi trường **MYSQL_ROOT_PASSWORD** là một chuỗi bất kì
6. Đặt giá trị cho biến môi trường **MYSQL_DATABASE** là **obo**
7. Đặt giá trị cho biến môi trường **MYSQL_USER** là **admin**
8. Đặt giá trị cho biến môi trường **MYSQL_PASSWORD** là **123456**
9. Sử dụng image là **handuy1992/mysql**

Chạy container cho ứng dụng SpringBoot

Yêu cầu:

1. Chạy ngầm
2. Đặt trong network **spring-app**
3. Expose cổng **8080** của container ra cổng **8005** trên máy host
4. Sử dụng image là **handuy1992/obo:latest**

Kiểm tra kết quả

Truy cập <http://localhost:8005>



ĐẬP HỘP HÀNG MỚI

[XEM THÊM](#)

Kyrie 5 Spongebob Squarepants

1,350,000 đ

Đã bán 12 đôi



Vans Sk8-Hi Checkerboard Cap

1,500,000 đ

Đã bán 25 đôi



Converse Chuck Taylor All-Star 70s Hi Undercover

3,600,000 đ

Đã bán 157 đôi



Adidas Ultra Boost 4.0 Game of Thrones White

2,000,000 đ

Đã bán 247 đôi



Vans Sk8-Hi Deconstructed Black

1,500,000 đ

Đã bán 61 đôi