Section 4 – Docker Containers

Giảng viên: Phạm Quang Anh Kiệt

Email: kietpham.dev@gmail.com

Docker Containers

- Start & Stop containers
- Publishing ports
- Xem logs
- Thực thi commands trong containers
- Xoá containers
- Sử dụng volumes để lưu dữ liệu containers
- Chia sẽ mã nguồn

Starting Containers

- docker ps: xem danh sách các containers đang chạy
- docker run {image}: khởi chạy 1 container sử dung {image}
- docker run -d {image name}: }: khởi chạy 1 container trong chế độ -d (Detach mode)
- docker run -d --name {new name} {existed name}: }: khởi chạy 1 container trong chế độ detach mode, với tên image mới {new name} từ image cũ {existed image}

View Logs

- docker logs {container id}: Xem logs của 1 container id
- docker logs –help:

```
Fetch the logs of a container

Options:
      --details        Show extra details provided to logs
  -f, --follow         Follow log output
      --since string   Show logs since timestamp (e.g. 2013-01-02T13:23:37Z) or relative (e.g. 42m for 42 minutes)
  -n, --tail string    Number of lines to show from the end of the logs (default "all")
  -t, --timestamps     Show timestamps
      --until string   Show logs before a timestamp (e.g. 2013-01-02T13:23:37Z) or relative (e.g. 42m for 42 minutes)
```

Publishing Ports

- docker run -d -p {host port}:{container port} --name {new name} {existed image}: **run (start)** a **new** container in {-d} detach mode and {mapped port} to localhost from {container port}, --name named it as {new name} from {existed image}

Executing Commands in Running Containers

- docker exec {container name} {command name}: Executing a {command name} in a {container name}: Ex: docker ex angular-app ls
- *docker run: **start a new container** and run a command
- *docker exec: **executing a command** in a **running container**

Stopping and Starting Containers

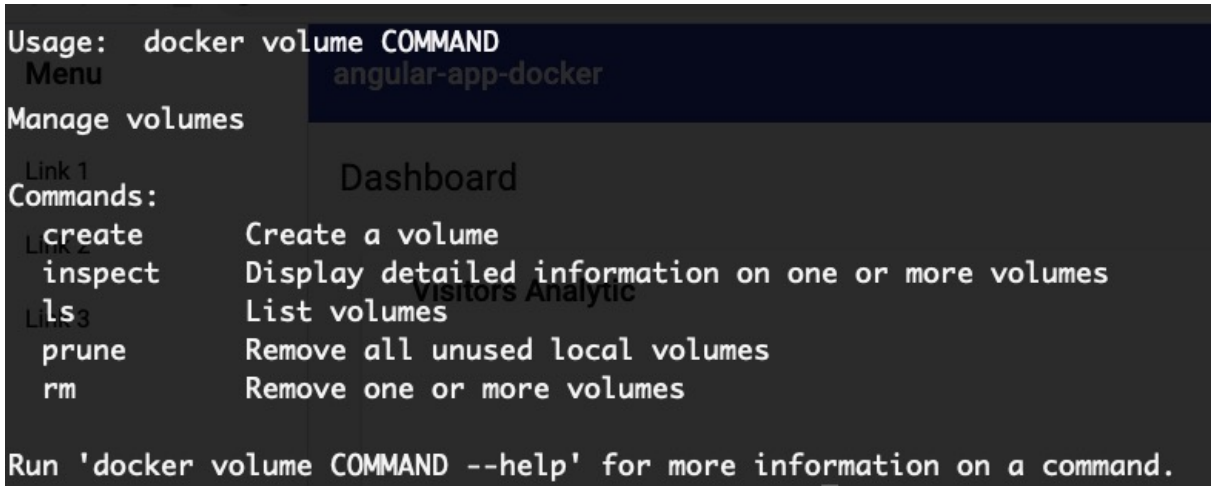
- docker stop {container name}: Stop a container with name {container name} (using: docker ps -a to see stopped container)
- docker start {container name}: Restart an existed/stopped container with name {container name}

Removing Containers

- docker rm {container name/id}: Remove a stopped container
- docker rm -f {container name/id}: Remove a stopped/running container
- docker ps -a | grep {container name}: Show a stopped container and filter by {container name}
- docker container prune: Remove all stopped containers

Containers File System

- docker volume



Containers File System


```
[
    {
        "CreatedAt": "2021-05-19T11:18:06Z",
        "Driver": "local",
        "Labels": {},
        "Mountpoint": "/var/lib/docker/volumes/app-data/_data",
        "Name": "app-data",
        "Options": {},
        "Scope": "local"
    }
]
```


- **Mountpoint** is the path **of virtual machine** (docker-machine) not a physical path

Containers File System

- docker run -d -p 4201:4200 -v {volume name}:/app/data {image name}
- docker exec -it {docker id} sh

Copying Files between the Host and Containers

- docker cp {container id}:/{workdir}/{file.x} . : Copy file.x to the current directory (workdir)
- docker cp {file.x} {container id}:/{workdir} : copy file.x to a {container id}/workdir

Sharing the Source Code with a Container

- PUBLISHING FLOWS:
- - Production: build a new image
- - Development:
  - Build a new image
  - Copy files
  - => Time consuming and bad practices :(

Sharing the Source Code with a Container

- docker run -d -p {host port}:{container port} -v $(pwd):/app {image name}