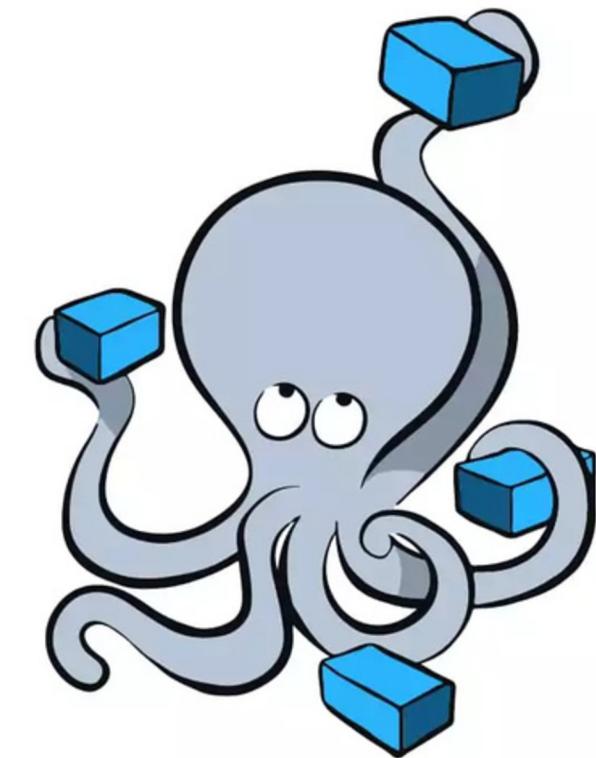


DOCKER COMPOSE

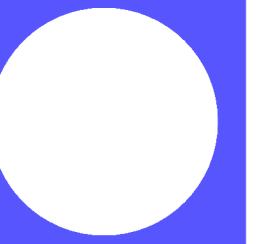


docker
Compose

MỤC TIÊU

- Hiểu được khái niệm và mục đích của Docker Compose trong việc quản lý và khởi động nhiều dịch vụ liên quan.
- Nắm vững cách viết file docker-compose.yml để định nghĩa và cấu hình các dịch vụ cần thiết.
- Thực hành sử dụng Docker Compose để khởi động, dừng và quản lý một nhóm dịch vụ.
- Biết cách kiểm tra trạng thái và log của các dịch vụ trong một hệ thống Docker Compose.
- Xử lý các lỗi cơ bản khi khởi động và vận hành một nhóm dịch vụ bằng Docker Compose.

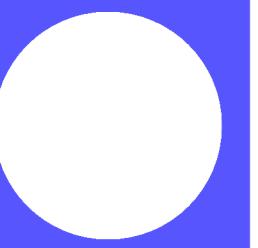
Giới thiệu Docker Compose



- **Docker Compose** là công cụ để định nghĩa và chạy các ứng dụng Docker đa container. Với Docker Compose, bạn có thể mô tả các container, mạng, volume trong một tệp YAML (docker-compose.yml) và khởi chạy toàn bộ ứng dụng với một lệnh duy nhất. Điều này rất hữu ích trong việc quản lý các ứng dụng phức tạp có nhiều thành phần.
- Lợi ích của Docker Compose:
 - Giúp quản lý và triển khai nhiều container một cách đồng bộ.
 - Dễ dàng cấu hình và tái sử dụng cấu hình cho các môi trường phát triển, kiểm thử, và sản xuất.
 - Tự động hóa các tác vụ thông qua file cấu hình duy nhất (docker-compose.yml).



So sánh sử dụng docker và docker compose



Khía cạnh	Docker	Docker Compose
Mục đích chính	Chạy, quản lý các container riêng lẻ.	Quản lý và điều phối nhiều container liên quan trong cùng một ứng dụng.
Cấu hình	Mỗi container phải được khởi động bằng câu lệnh riêng (dài và phức tạp nếu có nhiều container).	Cho phép định nghĩa toàn bộ ứng dụng trong một tệp YAML (<code>docker-compose.yaml</code>), giúp cấu hình dễ hiểu và quản lý dễ dàng.
Quy mô ứng dụng	Tốt cho các ứng dụng đơn giản hoặc chạy các container độc lập.	Lý tưởng cho các ứng dụng phức tạp với nhiều dịch vụ (frontend, backend, database, etc.).
Quản lý mạng	Phải cấu hình thủ công để kết nối các container với nhau.	Tự động tạo và quản lý mạng giữa các container trong cùng một dự án.
Khả năng tái sử dụng	Khó tái sử dụng cấu hình nếu không lưu các lệnh khởi chạy container.	File YAML có thể được chia sẻ, tái sử dụng, dễ dàng sửa đổi và áp dụng cho các môi trường khác nhau.
Độ phức tạp	Yêu cầu nhiều lệnh CLI riêng lẻ, dễ gây lỗi nếu không nhất quán.	Dễ dàng hơn nhờ một file YAML và một lệnh duy nhất (<code>docker-compose up</code>).
Khả năng mở rộng	Mở rộng container thủ công hoặc thông qua công cụ quản lý khác.	Hỗ trợ mở rộng số lượng container (scale services) dễ dàng với một lệnh (<code>docker-compose up --scale service=x</code>).
Quản lý vòng đời ứng dụng	Phải thực hiện nhiều lệnh thủ công cho start, stop, log, etc.	Có các lệnh tích hợp để quản lý toàn bộ vòng đời ứng dụng (<code>up</code> , <code>down</code> , <code>start</code> , <code>stop</code> , <code>logs</code>).
Tích hợp môi trường	Cấu hình môi trường phức tạp (phải dùng <code>--env</code> hoặc file riêng lẻ).	Hỗ trợ biến môi trường (<code>environment</code>) trực tiếp trong tệp YAML, dễ quản lý và tích hợp môi trường.
Hỗ trợ CI/CD	Khó cấu hình khi cần chạy nhiều container trong pipeline CI/CD.	Tích hợp tốt hơn trong CI/CD nhờ tệp YAML duy nhất mô tả toàn bộ ứng dụng.



Cấu trúc Manifest file

File docker-compose.yml là nơi bạn định nghĩa các dịch vụ (services), mạng (networks), và volumes cần thiết cho ứng dụng của bạn.

Tham khảo cách viết tại: <https://docs.docker.com/reference/compose-file/>

Giải thích từng phần:

- version: Chỉ định phiên bản của Docker Compose đang được sử dụng.
- services: Định nghĩa các dịch vụ sẽ được chạy. Mỗi dịch vụ tương ứng với một container.
- ports: Ánh xạ cổng từ máy host vào container.
- environment: Đặt các biến môi trường cho container.
- depends_on: chỉ ra sự phụ thuộc. Tức là services nào phải được cài đặt và chạy trước thì service được config tại đó mới được chạy. VD:
depends_on: - db khai báo container phụ thuộc vào container có tên là db.

```
version: "3.7"

services:
  nginx:
    build:
      context: docker/nginx
      dockerfile: Dockerfile
      args:
        - HOST=${HOST}
    depends_on:
      - app
    ports:
      - 80:80
      - 443:443
    env_file:
      - .env
    networks:
      - my_network

  app:
    depends_on:
      - db
    env_file:
      - .env
    ports:
      - 3000:3000
    build:
      context: .
      dockerfile: docker/app/Dockerfile
    volumes:
      - ./app
      - bundle_data:/bundle
```

Cấu trúc Manifest file

Giải thích từng phần:

- command: Giống như Dockerfile, khai báo command nào sẽ chạy khi container được chạy
- volumes: dùng để mount hai thư mục trên host và container với nhau.
- context: Đường dẫn đến context build mà Docker sử dụng. Build context là thư mục chứa tất cả tài nguyên cần thiết (như Dockerfile, mã nguồn, tệp cấu hình, v.v.) để xây dựng image.
- build: Xác định cách tạo ra image từ mã nguồn
- restart
 - restart: "no" → default nó sẽ không khởi động lại container trong bất cứ trường hợp nào
 - restart: always → Luôn khởi động lại khi xảy ra lỗi hoặc bị stop
 - restart: on-failure → Khởi động lại nếu xảy ra lỗi
 - restart: unless-stopped → Luôn khởi động lại container khi bị lỗi, ngoại trừ container bị stop

```
- 80:80
- 443:443
env_file:
- .env
networks:
- my_network

app:
depends_on:
- db
env_file:
- .env
ports:
- 3000:3000
build:
context: .
dockerfile: docker/app/Dockerfile
volumes:
- .:/app
- bundle_data:/bundle
command: sh /scripts/command.sh
stdin_open: true
tty: true

db:
image: mysql:5.7
restart: on-failure
env_file:
- .env
environment:
- MYSQL_ROOT_PASSWORD=${DATABASE_ROOT_PASSWORD}
- MYSQL_DATABASE=${DATABASE_NAME}
```

Sử dụng Docker Compose để khởi chạy một nhóm dịch vụ

To start all the services defined in your `compose.yaml` file:

```
$ docker compose up
```

To stop and remove the running services:

```
$ docker compose down
```

If you want to monitor the output of your running containers and debug issues, you can view the logs with:

```
$ docker compose logs
```

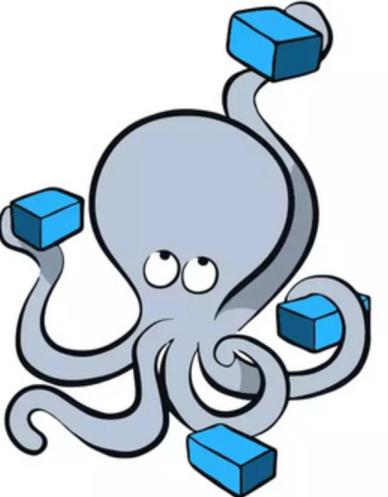
To lists all the services along with their current status:

```
$ docker compose ps
```



Các câu lệnh làm việc với docker compose sau khởi chạy container

- docker compose ps
- docker compose start
- docker compose stop
- docker compose config (-q)
- docker volume ls
- docker inspect volume container_name
- docker compose top: Xem các process chạy trong các container
- docker-compose logs
- docker container exec -it container_name /bin/bash
- docker-compose rm
- docker compose down = docker-compose stop + docker-compose rm
- docker compose scale service_name=x



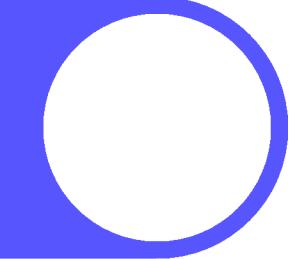
docker
Compose

Tổng hợp các câu lệnh docker compose

Command	Description
<code>docker compose alpha</code>	Experimental commands
<code>docker compose build</code>	Build or rebuild services
<code>docker compose config</code>	Parse, resolve and render compose file in canonical format
<code>docker compose cp</code>	Copy files/folders between a service container and the local filesystem
<code>docker compose create</code>	Creates containers for a service
<code>docker compose down</code>	Stop and remove containers, networks
<code>docker compose events</code>	Receive real time events from containers
<code>docker compose exec</code>	Execute a command in a running container
<code>docker compose images</code>	List images used by the created containers
<code>docker compose kill</code>	Force stop service containers
<code>docker compose logs</code>	View output from containers
<code>docker compose ls</code>	List running compose projects
<code>docker compose pause</code>	Pause services
<code>docker compose port</code>	Print the public port for a port binding
<code>docker compose ps</code>	List containers
<code>docker compose pull</code>	Pull service images
<code>docker compose push</code>	Push service images
<code>docker compose restart</code>	Restart service containers
<code>docker compose rm</code>	Removes stopped service containers
<code>docker compose run</code>	Run a one-off command on a service

<code>docker compose start</code>	Start services
<code>docker compose stop</code>	Stop services
<code>docker compose top</code>	Display the running processes
<code>docker compose unpause</code>	Unpause services
<code>docker compose up</code>	Create and start containers
<code>docker compose version</code>	Show the Docker Compose version information
<code>docker compose wait</code>	Block until containers of all (or specified) services stop.
<code>docker compose watch</code>	Watch build context for service and rebuild/refresh containers when files are updated

Ví dụ minh họa (n8n)



Hình bên phải là manifest file (.yaml) chạy n8n và database lưu trữ postgres sử dụng docker compose

- Run các service của docker compose file: docker compose up -d

```
cdev@ck-pc:~/docker-compose-demo$ docker compose up -d
[+] Running 3/3
  ● Network docker-compose-demo_default      Created
  ● Container docker-compose-demo-postgres-1  Healthy
  ● Container docker-compose-demo-n8n-1        Started
```

- Stop, start các service trong compose

```
cdev@ck-pc:~/docker-compose-demo$ docker compose ps
NAME          COMMAND           SERVICE      STATUS       PORTS
docker-compose-n8n-1    "tini -- /docker-ent..."  n8n        running
docker-compose-postgres-1 "docker-entrypoint.s..."  postgres   running (healthy)
5432/tcp
cdev@ck-pc:~/docker-compose-demo$ docker compose stop docker-compose-n8n-1
no such service: docker-compose-n8n-1
cdev@ck-pc:~/docker-compose-demo$ docker compose stop n8n
[+] Running 1/1
  ● Container docker-compose-n8n-1  Stopped
cdev@ck-pc:~/docker-compose-demo$ docker compose ps
NAME          COMMAND           SERVICE      STATUS       PORTS
docker-compose-n8n-1    "tini -- /docker-ent..."  n8n        exited (0)
docker-compose-postgres-1 "docker-entrypoint.s..."  postgres   running (healthy)
5432/tcp
cdev@ck-pc:~/docker-compose-demo$ docker compose start n8n
[+] Running 2/2
  ● Container docker-compose-postgres-1  Healthy
  ● Container docker-compose-n8n-1        Started
cdev@ck-pc:~/docker-compose-demo$ docker compose ps
NAME          COMMAND           SERVICE      STATUS       PORTS
docker-compose-n8n-1    "tini -- /docker-ent..."  n8n        running
docker-compose-postgres-1 "docker-entrypoint.s..."  postgres   running (healthy)
5432/tcp
```

- Kiểm tra container được chạy với cấu hình nào trên docker engine

```
cdev@ck-pc:~/docker-compose-demo$ docker compose config
name: docker-compose-demo
services:
  n8n:
    depends_on:
      - postgres
        condition: service_healthy
    environment:
      DB_POSTGRESDB_DATABASE: n8n
      DB_POSTGRESDB_HOST: postgres
```

```
  ● docker-compose.yaml > {} services > {} n8n > {} depends_on > {} postgres
    docker-compose.yml - The Compose specification establishes a standard for the definition of multi-container platform-agnostic applications (...)
    version: '3.8'

    volumes:
      db_storage:
      n8n_storage:

    services:
      postgres:
        image: postgres:16
        restart: always
        environment:
          POSTGRES_USER: admin
          POSTGRES_PASSWORD: password
          POSTGRES_DB: n8n
        volumes:
          - db_storage:/var/lib/postgresql/data
        healthcheck:
          test: ['CMD-SHELL', 'pg_isready -h postgres -U ${POSTGRES_USER} -d ${POSTGRES_DB}']
          interval: 10s
          timeout: 5s
          retries: 10

      n8n:
        image: docker.n8n.io/n8nio/n8n
        restart: always
        environment:
          DB_TYPE: postgresdb
          DB_POSTGRESDB_HOST: postgres
          DB_POSTGRESDB_PORT: 5432
          DB_POSTGRESDB_DATABASE: n8n
          DB_POSTGRESDB_USER: admin
          DB_POSTGRESDB_PASSWORD: password
        ports:
          - 7070:5678
        volumes:
          - n8n_storage:/home/node/.n8n
        depends_on:
          postgres:
            condition: service_healthy
```

Ví dụ minh họa (n8n)

- Kiểm tra các volume được khởi tạo bởi docker

```
cdev@ck-pc:~/docker-compose-demo$ docker volume ls
DRIVER      VOLUME NAME
local      5eb5a788d3b4938702b829b872a495f6e8deb1896e9419d67e4b26a3ee14f1e3
local      9fe059c673feeaa8f91580913bb4d9ba58d57f14aed1d597291140f79c29662ec
local      43d0e7ddeb0b5e94f7cbdf15f13e96bdbe1f55c25fdbeaf4e44503830bf509be
local      46dbbc6df29f6b8955628a521cf07cf8e9aac60907f940bfef6b87922144d083
local      73c560f54eeef56929c9be9cff04958a73d9ac1c4232a80a7e09a5572214041
```

- Kiểm tra cấu hình volume

```
cdev@ck-pc:~/docker-compose-demo$ docker inspect volume docker-compose-demo-n8n-1
[
    {
        "Id": "29e06cf6a54c93501866b6bf606933dbda3e0c12a4b922c02a1869f18c794b67",
        "Created": "2025-01-16T04:00:19.444631975Z".
    }
]
```

- Xem logs của container

```
cdev@ck-pc:~/docker-compose-demo$ docker compose logs n8n
docker-compose-demo-n8n-1 | User settings loaded from: /home/node/.n8n/config
docker-compose-demo-n8n-1 | Initializing n8n process
docker-compose-demo-n8n-1 | n8n ready on 0.0.0.0, port 5678
docker-compose-demo-n8n-1 | Version: 1.74.1
docker-compose-demo-n8n-1 | Editor is now accessible via:
docker-compose-demo-n8n-1 | http://localhost:5678/
docker-compose-demo-n8n-1 | Received SIGTERM. Shutting down...
docker-compose-demo-n8n-1 | Stopping n8n...
docker-compose-demo-n8n-1 | User settings loaded from: /home/node/.n8n/config
docker-compose-demo-n8n-1 | Initializing n8n process
docker-compose-demo-n8n-1 | n8n ready on 0.0.0.0, port 5678
docker-compose-demo-n8n-1 | Version: 1.74.1
docker-compose-demo-n8n-1 | Editor is now accessible via:
docker-compose-demo-n8n-1 | http://localhost:5678/
```

- Stop containers and remove containers, networks, volumes,images created by up

```
cdev@ck-pc:~/docker-compose-demo$ docker compose down
[+] Running 3/3
  ⚡ Container docker-compose-demo-n8n-1     Removed
  ⚡ Container docker-compose-demo-postgres-1 Removed
  ⚡ Network docker-compose-demo_default     Removed
cdev@ck-pc:~/docker-compose-demo$ docker compose ps
NAME          COMMAND   SERVICE   STATUS      PORTS
```

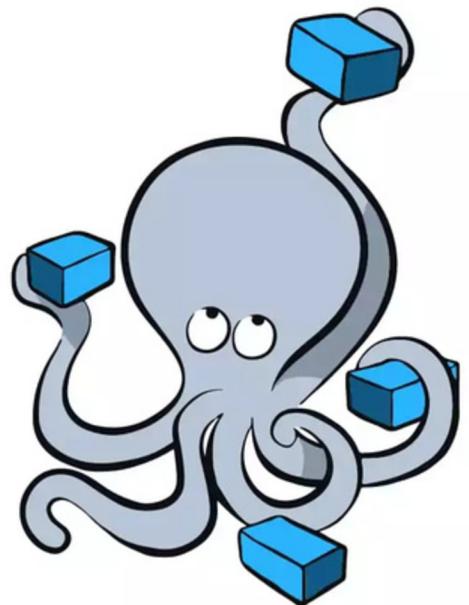
The screenshot shows a web browser window with the URL `localhost:7070/setup`. The page has a white header with the n8n logo. Below the header, there is a large white form area with a title "Set up owner account". The form contains several input fields:

- Email: A red-bordered field with the placeholder "Email *". Below it is a message: "This field is required".
- First Name: A red-bordered field with the placeholder "First Name *". Below it is a message: "This field is required".
- Last Name: A red-bordered field with the placeholder "Last Name *". Below it is a message: "This field is required".
- Password: A red-bordered field with the placeholder "Password *". Below it is a message: "This field is required".
- A checkbox labeled "I want to receive security and product updates".
- A red "Next" button at the bottom right.

Kết luận

Quy trình sử dụng docker compose:

- 1.Tạo Dockerfile: Định nghĩa image cho từng service.
- 2.Viết docker-compose.yaml: Khai báo các service, network, volume.
- 3.Chạy lệnh:
 - docker-compose up -d: Chạy ngầm các container.
 - docker-compose ps: Kiểm tra container đang chạy.
- 4.Theo dõi log: Sử dụng docker-compose logs.
- 5.Thực hiện các thao tác với các service trong compose
- 6.Tắt container: Sử dụng docker-compose down.



docker
Compose

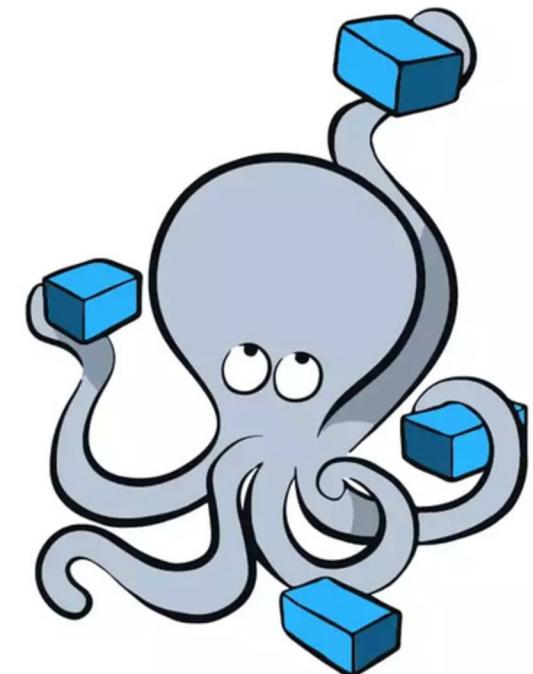
Lưu ý khi sử dụng Docker Compose

- Sử dụng biến môi trường (để bảo mật thông tin nhạy cảm).
- Tạo volume để tránh mất mát dữ liệu khi container bị xóa.
- Kiểm tra phiên bản docker-compose phù hợp với Docker Engine:
<https://docs.docker.com/compose/releases/release-notes/>

Docker Compose là một công cụ hữu ích giúp tối ưu hóa quản lý container và triển khai nhiều dịch vụ. Nó cung cấp cách dễ dàng để mô phỏng, tích hợp và quản lý hệ thống Docker một cách hiệu quả.

Bài tập

- Sử dụng docker compose để khởi tạo 1 nhóm các service sonarQube cùng lúc bao gồm:
 - database
 - sonaqube
- Đảm bảo:
 - mount volume dữ liệu cho service sonaqube
 - mount db data (mount point) ra thư mục cùng cấp với docker-compose.yaml để đảm bảo remove container và run lại không mất dữ liệu



docker
Compose

Chapter BE

THANK
YOU

