

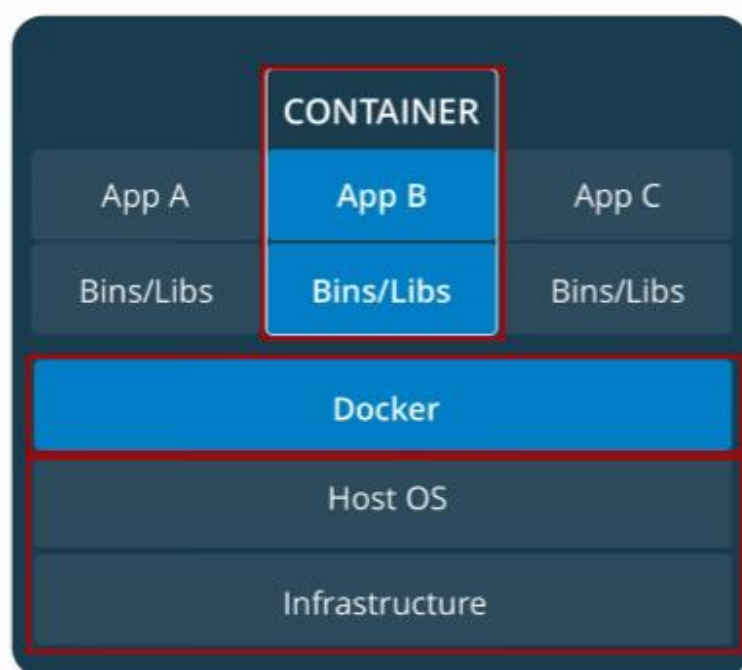
Báo cáo thực tập tuần 6 (từ 29/7 đến 4/8) : Tìm hiểu về docker triển khai ứng dụng trên nền docker

MỤC LỤC

1. Lí thuyết.....	2
2. Cài đặt Docker	4
3. Một số khái niệm trong Docker	4
3.1. Docker engine.....	5
3.1.1. Images.....	5
3.1.2. Dockerfile.....	6
3.1.3. Container	6
3.1.4. Network	7
3.1.5. Volume.....	7
3.2. Docker Hub	8
4. Ưu và nhược điểm khi sử dụng.....	8
4.1. Ưu điểm	8
4.2. Nhược điểm.....	8
5. Tạo và sử dụng mysql trong docker	9

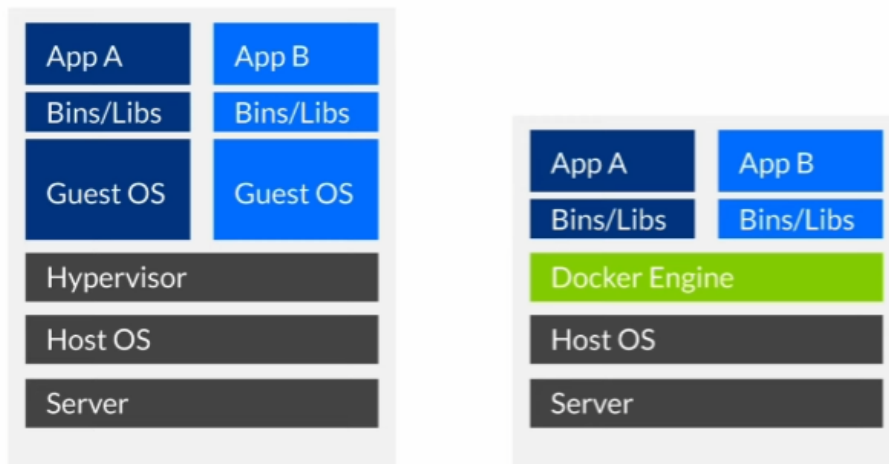
1. Lí thuyết

Docker là một nền tảng để cung cấp cách để building, deploying và running ứng dụng dễ dàng hơn bằng cách sử dụng các containers (trên nền tảng ảo hóa). Ban đầu viết bằng Python, hiện tại đã chuyển sang Golang.



Hình trên là tổng quan của docker. Có thể hiểu đơn giản docker là một ứng dụng giúp tạo ra các môi trường độc lập (container) để chạy các ứng dụng đơn lẻ. Các ứng dụng này chạy độc lập với nhau và không có tương tác qua lại với nhau (trừ khi ta kết nối chúng lại với nhau). Về mặt cấu trúc, ta có thể thấy đây là một nền tảng ảo hóa. Từ một nền tảng cơ sở hạ tầng (Infrastructure) đã được cài hệ điều hành (Host OS) và cài đặt Docker, ta có thể tạo ra nhiều container. Mỗi container này riêng biệt và chạy một ứng dụng cụ thể nào đó. Container này đóng gói lại tất cả các công cụ cần thiết hay môi trường cụ thể nào đó cho việc chạy ứng dụng. Giờ ứng dụng có thể khởi động và hoạt động một cách dễ dàng mà không gặp bất cứ lỗi nào liên quan đến môi trường không tương thích hay thiếu công cụ nào đó. Việc này làm cho quá trình cài đặt ứng dụng trở nên dễ dàng và tiết kiệm thời gian đi rất nhiều.

VMs vs Docker

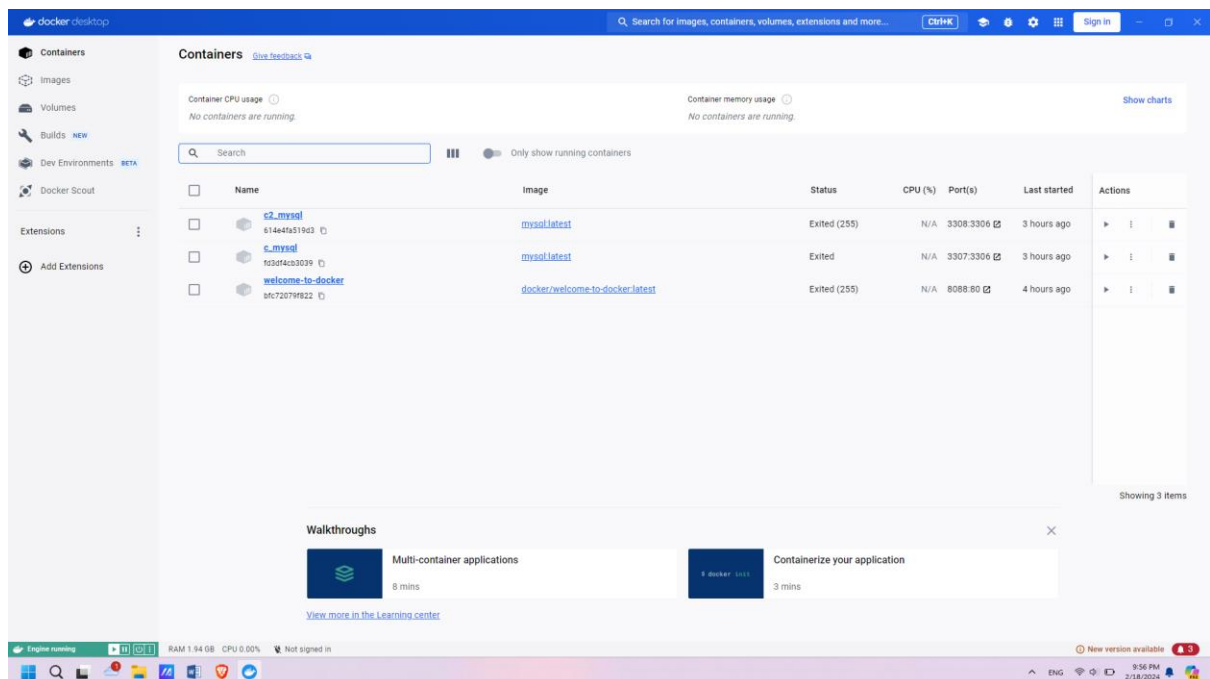


Đây là mô hình ảo hóa sử dụng Docker và việc sử dụng phần mềm ảo hóa virtuelle machine (VM). Mặc dù cũng là ảo hóa nhưng Docker lại có những phần khác hơn so với VM. Khi sử dụng các phần mềm ảo hóa, từ một máy chủ server ban đầu có một hệ điều hành riêng, thông qua hypervisor ta có thể chia sẻ tài nguyên và tạo ra được các máy ảo khác nhau. Mỗi máy ảo này có đặc điểm tương đương với máy chủ ban đầu, cũng có hệ điều hành riêng (có thể khác hệ điều hành so với máy chủ vật lý). Điều này giúp chạy và sử dụng các máy ảo như là một máy chủ vật lý, chạy các ứng dụng mong muốn. Tuy nhiên có một nhược điểm là các máy ảo có hệ điều hành riêng biệt nên thành ra tốc độ xử lý vì đó mà trở nên chậm lại (có thể nhận thấy rằng máy ảo xử lý chậm hơn máy vật lý nhiều). Docker thì giải quyết được nhược điểm trên. Docker tạo ra các vùng độc lập để chạy các ứng dụng riêng biệt. Các ứng dụng này được chạy trên hệ điều hành giống máy chủ vật lý. Điều này làm cho việc sử dụng docker sẽ tăng tốc độ xử lý lên tuy nhiên lại không linh hoạt được như sử dụng máy ảo. Và một điều nữa làm cho việc sử dụng docker ngày càng phổ biến hơn là sự linh động. Các không gian chạy các ứng dụng riêng biệt này có thể đóng gói và gửi đi một cách dễ dàng giống như việc ta đóng tất cả hàng hóa lại vào container sau đó gửi đi. Khách hàng chỉ cần mở container ra và sau đó sử dụng các ứng dụng trong đó. Nhanh chóng và tiện lợi. Hiện nay hầu hết các ứng dụng đã sử dụng Docker.

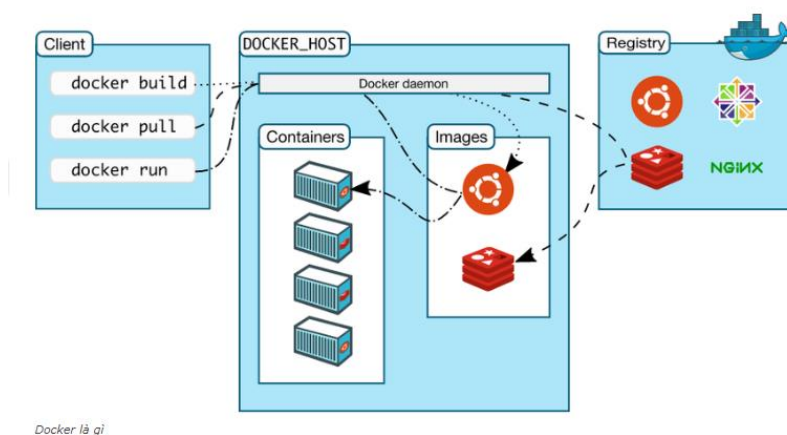
2. Cài đặt Docker

Việc cài đặt docker desktop khá đơn giản. Ta lên trang chủ <https://www.docker.com/products/docker-desktop/>

Sau đó tải về bản phù hợp với máy tính của mình. Sau khi tải về ta chạy file Docker_Desktop_Install.exe và thay đổi một vài tùy chỉnh là ta đã cài đặt thành công. Dưới đây là giao diện sau khi tải về và cài đặt thành công.



3. Một số khái niệm trong Docker



Đây là mô hình hoạt động của Docker, giờ sẽ làm rõ công dụng của từng phần

3.1. Docker engine

Docker engine là phần quan trọng nhất của Docker có tác dụng như một công cụ để đóng gói ứng dụng được xây dựng theo cấu trúc client-server và được cài trên máy host

Docker engine được chia làm 3 phần:

- Server : còn có tên gọi khác là docker daemon, có tác dụng quản lí các images, container, volumes và network
- Rest API: controller của docker daemon, sẽ chỉ ra những công việc mà docker daemon sẽ làm.
- Client: Là một công cụ giúp người dùng giao tiếp với Docker host. Người dùng tương tác với docker thông qua command trong terminal (CLI). Docker Client sẽ sử dụng API gửi lệnh tới Docker Daemon.

Ngoài ra còn 5 đối tượng lớn trong docker engine:

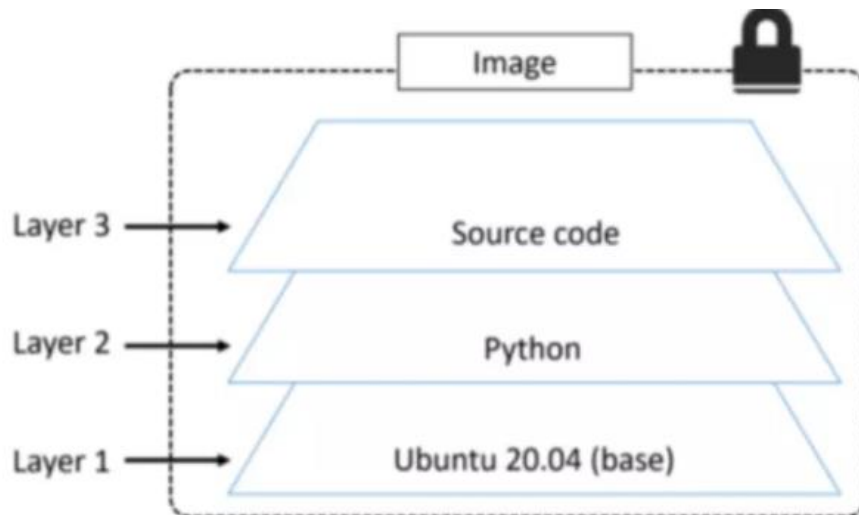
- Images
- Dockerfile
- Container
- Volumes
- Network

Ta sẽ tìm hiểu từng đối tượng lớn này

3.1.1. Images

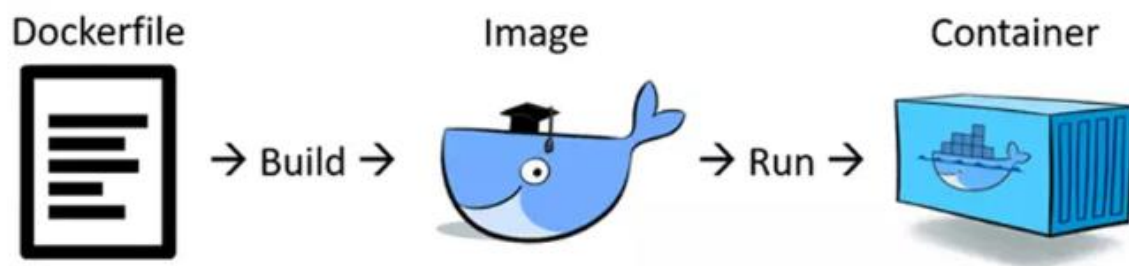
Images là một khuôn mẫu để tạo một container. Thường thì image sẽ dựa trên 1 image có sẵn với những tùy chỉnh thêm. Ví dụ bạn build 1 image dựa trên image Centos mẫu có sẵn để chạy Nginx và những tùy chỉnh, cấu hình để ứng dụng web của bạn có thể chạy được.

Ngoài ra, có thể tự build một image riêng cho mình hoặc sử dụng những image được chia sẻ từ cộng đồng Docker Hub. Một image sẽ được build dựa trên những chỉ dẫn của Dockerfile. Từ những câu lệnh đó, Docker sẽ build ra Docker image (thường có dung lượng nhỏ từ vài MB đến lớn vài GB).



3.1.2. Dockerfile

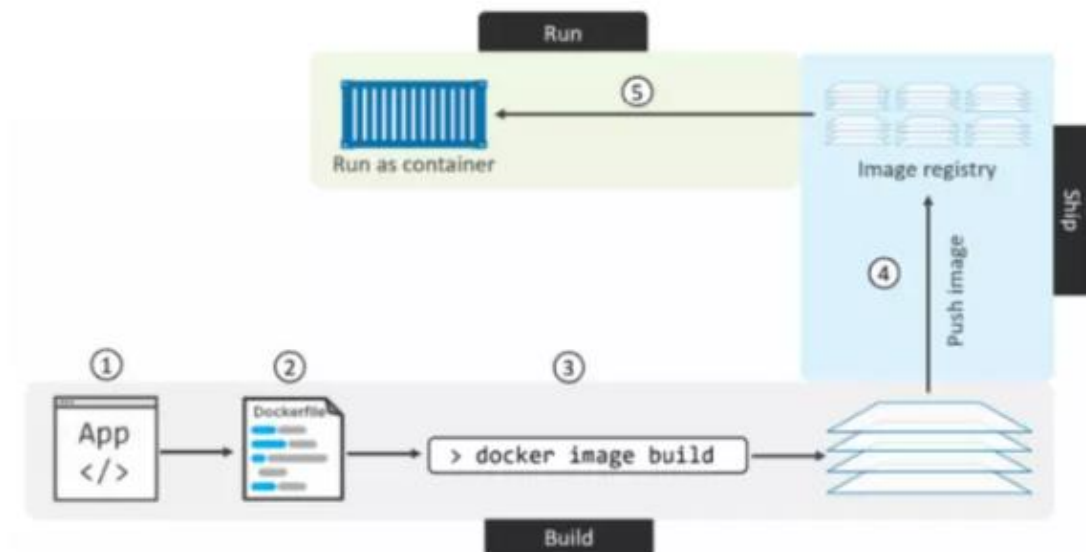
Dockerfile chứa các dòng lệnh để build nên một Images.



3.1.3. Container

Một image có thể **run** ra một hoặc nhiều container. Ta có thể create, start, stop, delete,... các container dựa trên Docker API hoặc Docker CLI. Mỗi container bao gồm tất cả mọi thứ để nó có thể hoạt động một cách độc lập như: code, runtime, system tools, system libraries, setting,... Bên trong nó sẽ có một ứng dụng cụ thể nào đó.

Quá trình đưa một ứng dụng chạy trong container như sau:



- Bước 1: ta sẽ viết ứng dụng và tất cả các phụ thuộc cần thiết của ứng dụng
- Bước 2: Sau khi ta viết xong ứng dụng, ta cần viết Dockerfile để chuẩn bị cho quá trình build ứng dụng thành Images
- Bước 3: build ứng dụng thành Image. Ta sử dụng docker build để làm
- Bước 4: lưu image tại docker hub (có thể có hoặc không)
- Bước 5 : sử dụng lệnh docker run để biến image thành container và ta đã thành công đưa một ứng dụng chạy trong container

3.1.4. Network

Docker network có nhiệm vụ cung cấp private network (VLAN) để các container trên một host có thể liên lạc được với nhau, hoặc các container trên nhiều hosts có thể liên lạc được với nhau (multi-host networking).

3.1.5. Volume

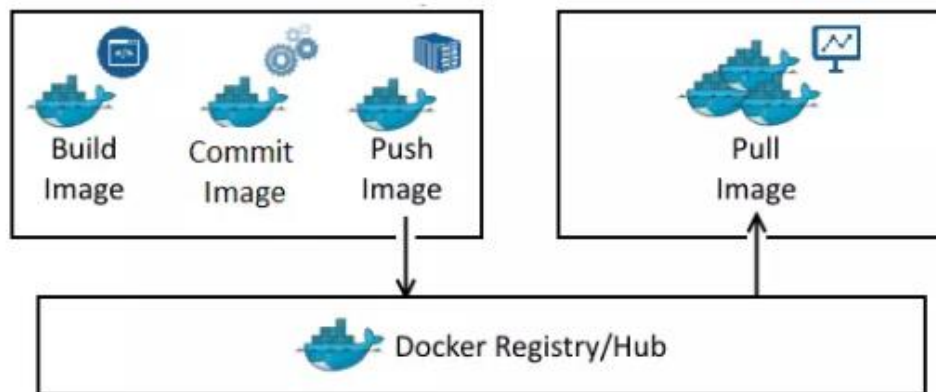
Docker volume là cơ chế tạo và sử dụng dữ liệu của docker, có nhiệm vụ lưu trữ dữ liệu độc lập với vòng đời của container.

Có 3 trường hợp sử dụng Docker Volume:

- Giữ lại dữ liệu khi một Container bị xóa.
- Để chia sẻ dữ liệu giữa máy chủ vật lý và Docker Container.
- Chia sẻ dữ liệu giữa các Docker Container.

3.2. Docker Hub

Docker Hub là một dịch vụ máy chủ cho phép lưu trữ các docker image của cá nhân, công ty, team,... Dịch vụ Docker Hub có thể được cung cấp bởi tổ chức thứ 3 hoặc là dịch vụ nội bộ được xây dựng riêng nếu bạn muốn.



Ta có thể build image và push nó lên docker hub để lưu trữ. Sau đó, khi muốn sử dụng tại nơi khác, ta có thể pull image về và run để tạo container.

4. Ưu và nhược điểm khi sử dụng

4.1. Ưu điểm

- Linh động: Dễ dàng triển khai ở nhiều môi trường khác nhau do loại bỏ được sự phụ thuộc của ứng dụng vào tầng OS cũng như cơ sở hạ tầng.
- Tiết kiệm không gian: container được xây dựng dựa trên nhiều image có sẵn, từ đó tiết kiệm được nhiều không gian lưu trữ hơn.
- Đồng nhất: không có sự sai khác về mặt môi trường khi triển khai ở bất kỳ nơi đâu, tạo ra sự nhất quán khi làm việc theo nhóm.
- Nhanh: do chia sẻ host OS nên các container có thể gần như được tạo một cách tức thì, việc khởi động cũng diễn ra nhanh hơn rất nhiều.

4.2. Nhược điểm

- Tất cả các container trên một máy chủ cụ thể phải được thiết kế để chạy trên cùng một loại hệ điều hành. Các container dựa trên một hệ điều hành khác sẽ yêu cầu một máy chủ lưu trữ khác.
- Bởi vì hệ điều hành được chia sẻ, một lỗ hổng bảo mật trong nhân hệ điều hành là mối đe dọa cho tất cả các container trên máy chủ

- Nếu OS của máy chủ bị lỗi, thì các Container chạy trên nó sẽ không hoạt động được

5. Tạo và sử dụng mysql trong docker

Các bước thực thi:

- Tải image Mysql
- Dùng các lệnh để tạo ra một cơ sở dữ liệu
- Kết nối cơ sở dữ liệu đó với mysql workbench

Bước 1: Tải image mysql

Ta sử dụng lệnh **docker pull mysql:latest** để kéo về để sử dụng

```
PS D:\documents\STUDY\TT viettel\lab> docker pull mysql:latest
latest: Pulling from library/mysql
d9a40b27c30f: Pull complete
fe4b01031aab: Pull complete
aa72c34c4347: Pull complete
473ade985fa2: Pull complete
cc168a9482de: Pull complete
3ca3786815dd: Pull complete
3e3fac98ea83: Pull complete
10e5505c3ae4: Pull complete
a79ade39aab9: Pull complete
ae34d51c6da2: Pull complete
Digest: sha256:d8df069848906979fd7511db00dc22efeb0a33a990d87c3c6d3fcdafd6fc6123
Status: Downloaded newer image for mysql:latest
docker.io/library/mysql:latest

What's Next?
View a summary of image vulnerabilities and recommendations → docker scout quickview mysql:latest
```

Bước 2: Dùng lệnh để tạo một cơ sở dữ liệu

Sau khi pull thành công mysql, ta thực hiện câu lệnh

docker run -it -p 3307:3306 -e MYSQL_ROOT_PASSWORD=123456 -d mysql

Trong đó:

- p 3307:3306** là ánh xạ địa chỉ cổng từ ứng dụng sang máy host vật lí
- e MYSQL_ROOT_PASSWORD=123456** là đặt mật khẩu cho cơ sở dữ liệu này là 123456
- d mysql** là tên image ta chạy

```
PS D:\documents\STUDY\TT viettel\lab> docker run -it -p 3307:3306 -e MYSQL_ROOT_PASSWORD=123456 -d mysql
24b89ce4bf39175b8e163fc14486d3b68e550e54df2e489fe21b0df8660383a9
```

Ta dùng câu lệnh docker ps để kiểm tra

```
PS D:\documents\STUDY\TT viettel\lab> docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	N
24b89ce4bf39	mysql	"docker-entrypoint.s..."	34 seconds ago	Up 32 seconds	33060/tcp, 0.0.0.0:3307->3306/tcp	a

Hiện lên như này có nghĩa là đang có một container mysql đang chạy

B3: sau khi đã xong 2 bước trên ta quay lại terminal và viết thêm 2 câu lệnh để kết nối thành công mysql

Docker exec -it 24b bash

Mysql -u root -p

Điền password và ta đã kết nối mysql thành công. Bây giờ ta chỉ cần sử dụng câu lệnh sql là được.

```
PS D:\documents\STUDY\TT viettel\lab> docker exec -it 24b bash
```

```
bash-5.1# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 9.0.1 MySQL Community Server - GPL

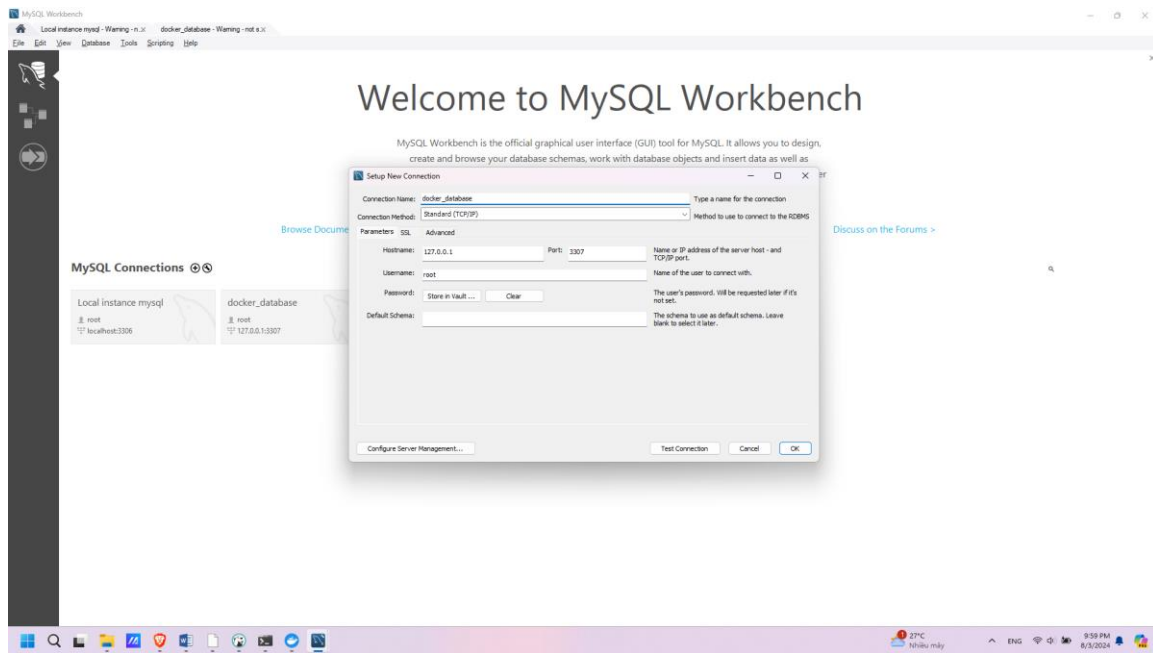
Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

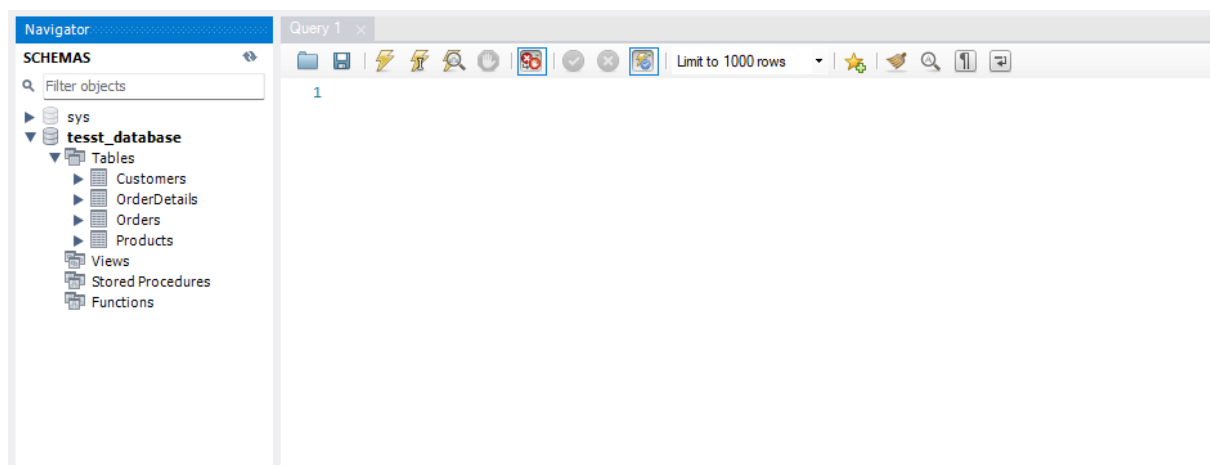
mysql> show table;
```

Ta đã kết nối thành công đến mysql trong docker



ta mở mysql workbench, tạo một connection đến cổng 3307, điền mật khẩu và đã kết nối thành công mysql workbench đến với mysql trong container mysql trong docker

ta tạo một schema và chuyển sang schema mới đó.



Tạo và chạy đoạn script như sau:

-- Tạo bảng Customers

```
CREATE TABLE Customers (
  CustomerID INT PRIMARY KEY,
  FirstName VARCHAR(50),
  LastName VARCHAR(50),
```

```
Email VARCHAR(100)
);

-- Tạo bảng Orders
CREATE TABLE Orders (
    OrderID INT PRIMARY KEY,
    OrderDate DATE,
    CustomerID INT,
    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)
);

-- Tạo bảng Products
CREATE TABLE Products (
    ProductID INT PRIMARY KEY,
    ProductName VARCHAR(100),
    Price DECIMAL(10, 2)
);

-- Tạo bảng OrderDetails
CREATE TABLE OrderDetails (
    OrderDetailID INT PRIMARY KEY,
    OrderID INT,
    ProductID INT,
    Quantity INT,
    FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),
    FOREIGN KEY (ProductID) REFERENCES Products(ProductID)
);
```

-- Chèn dữ liệu vào bảng Customers

INSERT INTO Customers (CustomerID, FirstName, LastName, Email)
VALUES

(1, 'John', 'Doe', 'john.doe@example.com'),
(2, 'Jane', 'Smith', 'jane.smith@example.com'),
(3, 'Michael', 'Johnson', 'michael.johnson@example.com');

-- Chèn dữ liệu vào bảng Orders

INSERT INTO Orders (OrderID, OrderDate, CustomerID)
VALUES

(1, '2024-08-01', 1),
(2, '2024-08-02', 2),
(3, '2024-08-03', 1);

-- Chèn dữ liệu vào bảng Products

INSERT INTO Products (ProductID, ProductName, Price)
VALUES

(1, 'Laptop', 999.99),
(2, 'Smartphone', 499.99),
(3, 'Tablet', 299.99);

-- Chèn dữ liệu vào bảng OrderDetails

INSERT INTO OrderDetails (OrderDetailID, OrderID, ProductID,
Quantity)
VALUES

(1, 1, 1, 1),
(2, 1, 2, 2),
(3, 2, 3, 1),

(4, 3, 1, 1),

(5, 3, 3, 3);

Tiếp theo chuyển sang terminal và thử một số câu lệnh liên quan đến dữ liệu ta vừa import

```
mysql> use tesst_database;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----+
| Tables_in_tesst_database |
+-----+
| Customers                |
| OrderDetails             |
| Orders                   |
| Products                 |
+-----+
4 rows in set (0.01 sec)
```

```
mysql> select * from Customers;
+-----+-----+-----+-----+
| CustomerID | FirstName | LastName | Email |
+-----+-----+-----+-----+
| 1 | John | Doe | john.doe@example.com |
| 2 | Jane | Smith | jane.smith@example.com |
| 3 | Michael | Johnson | michael.johnson@example.com |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
| tesst_database |
+-----+
5 rows in set (0.00 sec)

mysql>
```