

THỰC HÀNH

DEMO MỘT KẾT NỐI THỰC TẾ VỚI MYSQL/POSTGRESQL

Phương pháp

- Sử dụng Docker nhằm thiết lập nhanh một môi trường MySQL hoặc PostgreSQL để có thể thực hiện demo kết nối trong bài học

Quy trình demo kết nối với CSDL Client-Server sử dụng Docker, bao gồm cả các bước tiên quyết về cài đặt và giải thích khái niệm Docker.

Quy trình này được chia thành **3 Giai đoạn chính** và tập trung vào các bước then chốt (notes) để đảm bảo buổi demo diễn ra hiệu quả và chuyên nghiệp.

I. Giai đoạn 1: Thiết lập & Giải thích khái niệm (Các bước tiên quyết)

1.1. Giải thích Vai trò của Docker (Containerization)

Mục đích: Giúp sinh viên hiểu tại sao chúng ta không cài đặt MySQL trực tiếp lên máy.

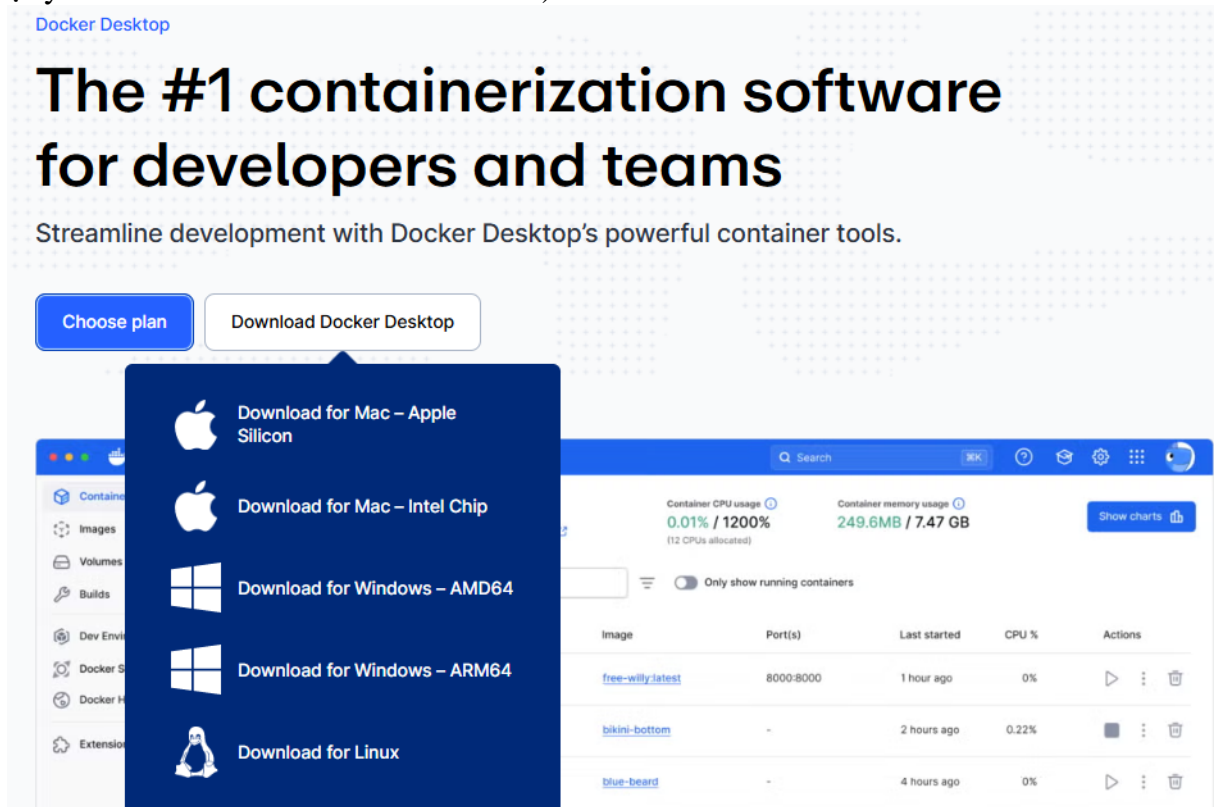
Lưu ý quan trọng	Giải thích
Tính nhất quán môi trường	Docker tạo ra một môi trường biệt lập (container) chứa MySQL Server, giống hệt môi trường Production, không bị ảnh hưởng bởi hệ điều hành máy Host.
Sử dụng tài nguyên tối ưu	Dễ dàng khởi động và tắt máy chủ CSDL chỉ bằng lệnh đơn giản, giải phóng tài nguyên ngay sau khi demo.
Minh họa Client-Server	Container đóng vai trò là Server từ xa, buộc code Python (Client) phải kết nối thông qua Host, Port, User, Password – minh họa hoàn hảo cho kiến trúc Client-Server.
Kiến trúc:	Docker chạy các ứng dụng trong các container, nhẹ hơn nhiều so với Máy ảo (VM).

1.2. Cài đặt các Công cụ (Thực hiện ở nhà trước buổi học)

Công cụ	Lệnh/Thao tác	Lưu ý quan trọng
---------	---------------	------------------

Docker Desktop	Tải và cài đặt từ trang chủ Docker. https://www.docker.com/products/docker-desktop/	Đảm bảo Docker đã chạy trước khi bắt đầu demo.
-----------------------	--	---

Tùy hệ điều hành và cấu hình máy: Lựa chọn bản Docker Desktop để tải về cài đặt.
(Gợi ý: Download for Windows-AMD64)



II. Giai đoạn 2: Chuẩn bị & Thực thi Demo

2.1. Khởi động Server MySQL bằng Docker

Đây là bước thiết lập Server để kết nối.

Bước 0: khởi động:

Sử dụng trình soạn thảo notepad hoặc trong VSC, tạo một file Bash lưu lại trên máy với tên: **connect_MySQL_Docker.bat**

Lưu ý quan trọng: toàn bộ nội dung file Bash dưới đây (**connect_MySQL_Docker.bat**) gõ trên một dòng duy nhất (không được enter để ngắt dòng) như sau:

```
docker run --name mysql-demo-server -e
MYSQL_ROOT_PASSWORD=uneti -e MYSQL_DATABASE=data_science_db
-p 3307:3306 -d mysql:latest
```

- **Lưu ý 1 (Lỗi Cổng):** Luôn giải thích tại sao phải dùng -p 3307:3306 (để tránh xung đột với cổng 3306 mặc định nếu có chương trình khác đang dùng).
- **Lưu ý 2 (Tham số):** Nhắc lại các tham số tạo nên chuỗi xác thực:
 - **Host** (127.0.0.1),
 - **User** (root),
 - **Password** (uneti),
 - **Port** (3307),
 - **DB** (data_science_db).

Bước 0:

1. Đảm bảo đã chạy docker

2. Mở PowerShell ở chế độ Administrator, chuyển đến thư mục chứa file connect_MySQL_Docker.bat, từ dấu nhắc gõ lệnh:

```
.\connect_MySQL_Docker.bat (gõ Enter)
```

3. Kiểm tra: Trong PowerShell tiếp tục chạy docker ps để xác nhận container đang ở trạng thái **Up** trên cổng 0.0.0.0:3307->3306/tcp.

Gõ lệnh:

```
docker ps (gõ Enter)
```

Nếu thành công trong PowerShell hiển thị màn hình dạng

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
c88f44efdeea	mysql:latest	"docker-entrypoint.s..."	51 seconds ago	Up 49 seconds	33060/tcp, 0.0.0.0:3307->3306/tcp	mysql-demo-server

Bước 1. Viết Code Python Demo

Code Python: Đảm bảo code sử dụng các tham số đã thiết lập.

Sử dụng VSC viết file code : connect_mySQL.ipynb

Một số lưu ý khi viết code

```
# SỬ DỤNG PORT 3307
PORT = 3307
USER = "root"
PASSWORD = "uneti"
```

```
DATABASE = "data_science_db"
HOST = "127.0.0.1"
```

```
conn = mysql.connector.connect(
    host=HOST,
    user=USER,
    password=PASSWORD,
    database=DATABASE,
    port=PORT
)
# ...
```

- **Lưu ý 3 (Sự khác biệt):** Chỉ ra rõ ràng sự khác biệt với `sqlite3.connect('database.db')` (chỉ cần 1 tham số).

Bước 3. Thực thi Demo & Giải thích

1. **Chạy Code Python:** Chạy code trong môi trường lập trình (Jupyter/IDE).
2. **Kết quả mong đợi:**
✓ Kết nối THÀNH CÔNG tới CSDL 'data_science_db' qua cổng 3307.
Phiên bản MySQL: 9.5.0
...
3. **Kết luận (Chứng minh):** Kết quả thành công này chứng minh rằng Python đã gửi yêu cầu xác thực qua mạng (local network) tới máy chủ (Container Docker) và nhận lại phản hồi, hoàn tất chu trình kết nối Client-Server.

III. Giai đoạn 3: Dọn dẹp tài nguyên

Đây là bước không thể thiếu để duy trì tính chuyên nghiệp và sự ổn định của hệ thống.

3.1. Dừng và Xóa Container

Sau khi demo hoàn tất, thực hiện hai lệnh sau trong Terminal (Môi trường PoweShell ở chế độ Administrator):

Gõ các lệnh	Mục đích
<code>docker stop mysql-demo-server</code>	Dừng hoạt động của Server MySQL, giải phóng bộ nhớ.

<code>docker rm mysql-demo-server</code>	Xóa container đã tạo ra khỏi danh sách Docker.
--	--

- **Lưu ý (Quan trọng):** Nếu không xóa (docker rm), lần sau khi chạy lại lệnh docker run với cùng tên (mysql-demo-server), sẽ gặp lại lỗi **Conflict**.