

**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI  
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**



**SOICT**

**BÁO CÁO**  
**Các hệ thống phân tán và ứng dụng**  
*LAB 01*

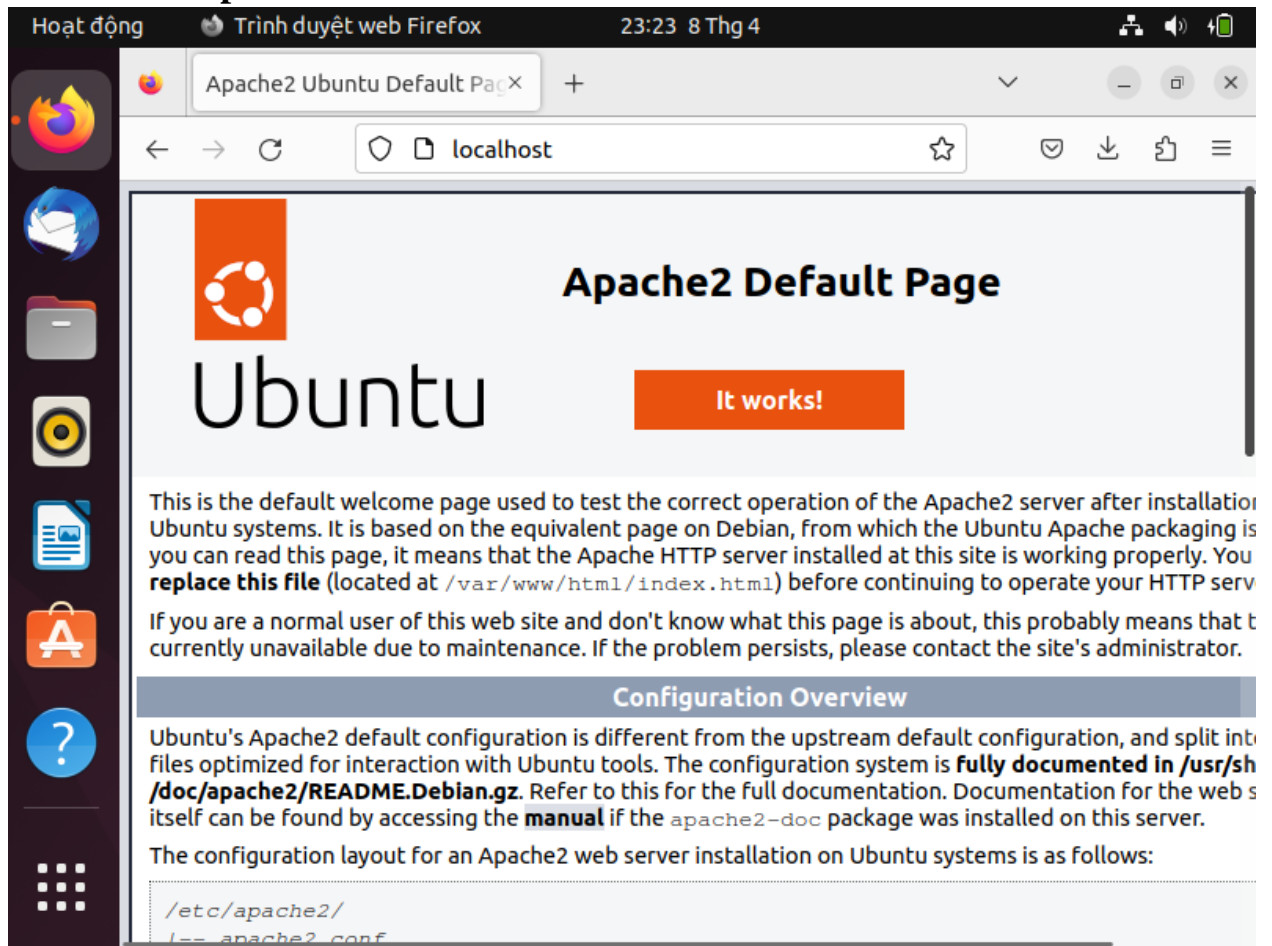
Sinh viên thực hiện: **Nguyễn Xuân Bách**

Mã số sinh viên: **20204714**

Giảng viên: **TS. Trần Hải Anh**

Hà Nội, 10 – 2021

## 1. Web server apache2



2.

- Câu hỏi 1: Đường dẫn đến file html chứa nội dung mặc định của trang web các bạn vừa xem là gì?

TL: /var/www/html/index.html

- Câu hỏi 2: Cổng mặc định của dịch vụ www là gì?

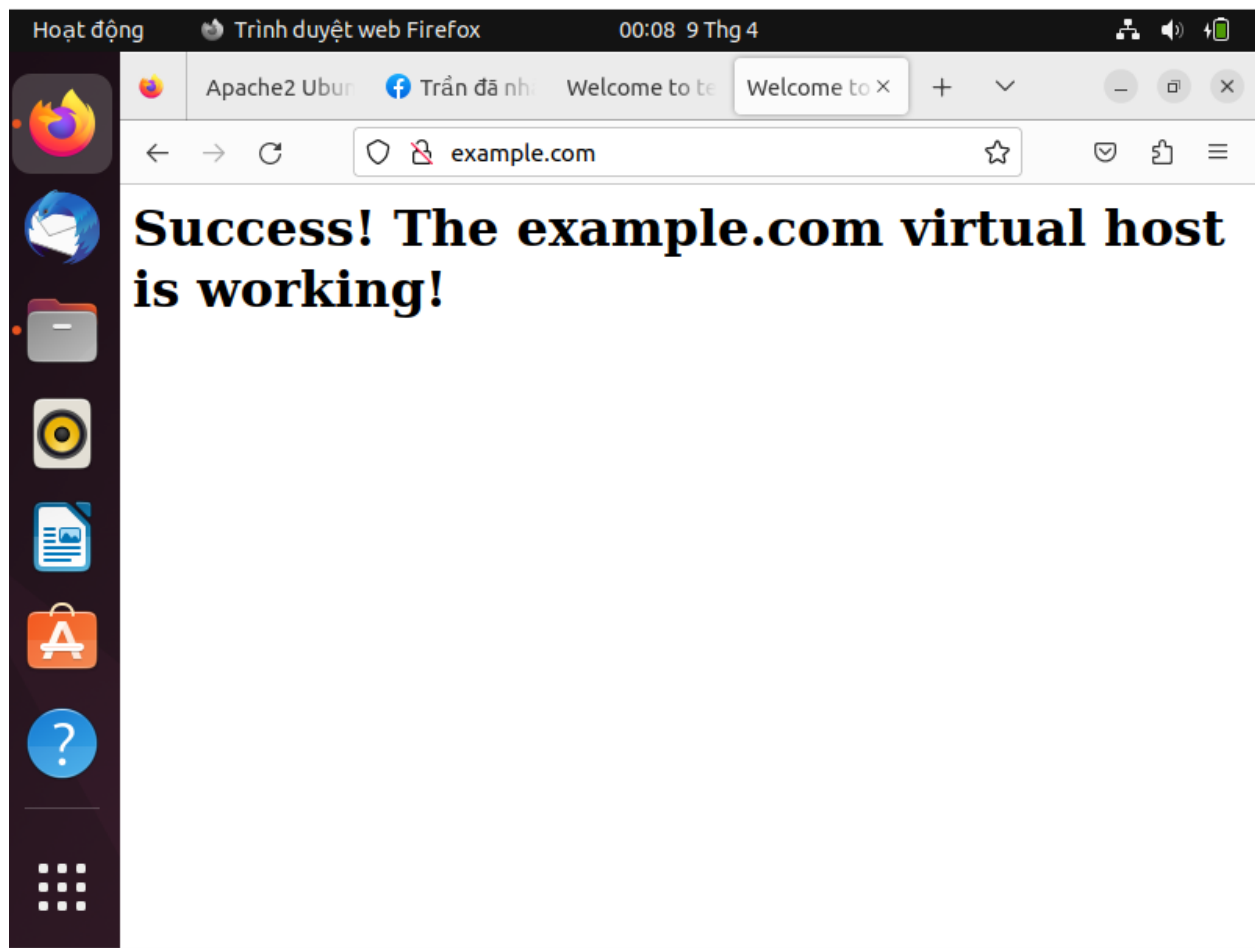
TL: Cổng 80

- Câu hỏi 3: Hãy giải thích quyền mang số 755 là gì?

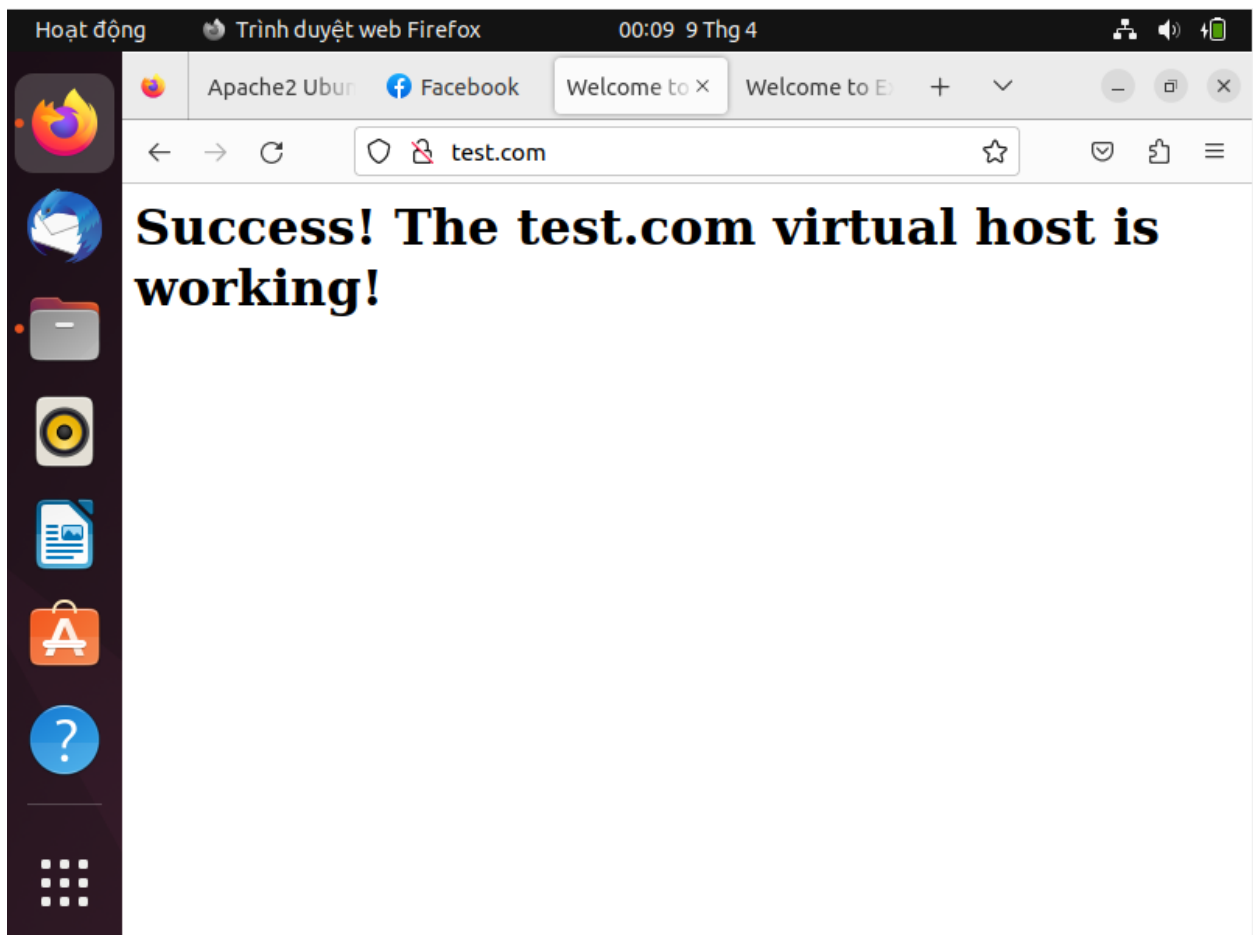
TL: Cho phép chủ sở hữu truy cập, sửa file theo cách họ muốn. Mọi người khác chỉ có thể truy cập mà không có quyền sửa nó. Tất cả mọi người đều có quyền thực thi.

Câu hỏi 4: Bạn quan sát thấy nội dung gì sau khi gõ 2 địa chỉ trên? Giải thích.

- “Welcome to Example.com!”, “Success! The example.com virtual host is working!” trong phần nội dung.



- “Welcome to Test.com!”, “Success! The test.com virtual host is working!” trong phần nội dung.

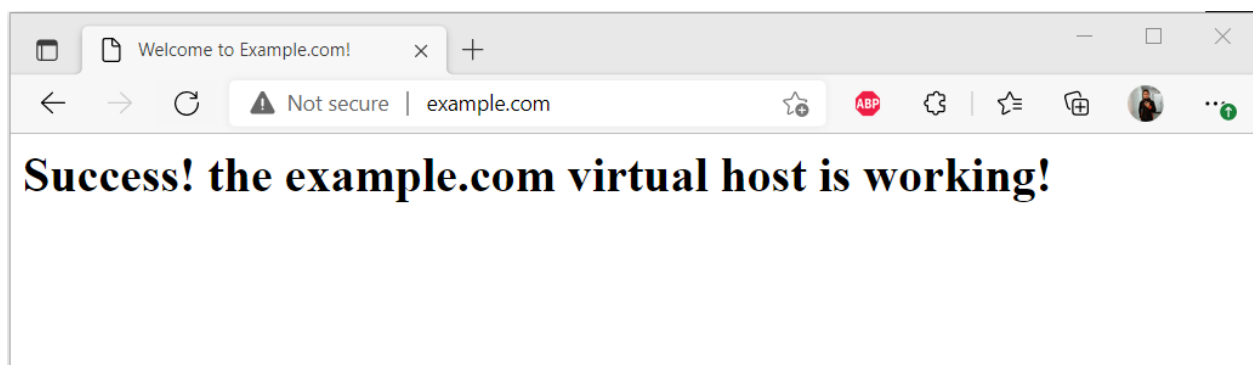


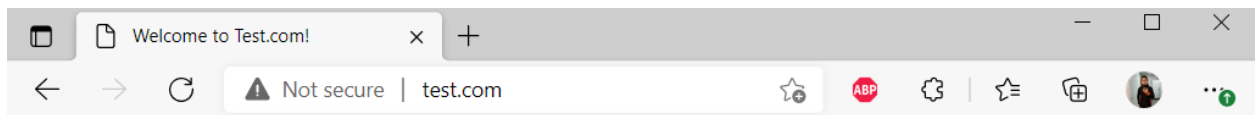
- Giải thích: Trang web đã được thực thi và chủ sở hữu có thể nhìn thấy giao diện web.

*Câu hỏi 5: Thử truy cập từ các máy tính khác trong cùng mạng LAN vào 2 trang web đó.*

- Phải cấu hình file hosts và proxy mới có thể truy cập được.

- Kết quả tại máy chính (Windows) bật kết nối Host-only với máy ảo (Ubuntu) (cùng 1 LAN).





**Success! The Test.com virtual host is working!**

## 2. Interface trong Java

- Câu hỏi 6: Hãy tự viết một đoạn code để thực hiện 1 vòng lặp while sao cho nó sẽ nhận các số mà người dùng gõ và gửi về server, cho đến khi nào người dùng gõ ký tự rỗng rồi ấn enter.

Gợi ý: hãy dùng lệnh sau để nhận xâu ký tự người dùng gõ vào: `String message = scanner.nextLine();`

-TL:

```
while (true) {  
    response = input.readLine();  
    System.out.println(response);  
    message = scanner.nextLine();  
    if (message.isEmpty()) {  
        break;  
    }  
    output.println(message);  
}
```

- Câu hỏi 7: Vai trò của phương thức run là gì? Khi nào thì nó được gọi?

TL: Phương thức “run” thực thi luồng tương ứng nếu phương thức “start” được gọi. Khi có một kết nối từ Client đến Server, một thread mới tại server được tạo ra, phương thức “start” được gọi để bắt đầu thực thi phương thức “run”.

## 3. Kiến trúc Microservices

Câu hỏi 1:

- Lệnh “`.mvnwclean package -Dmaven.test.skip=true`” giúp dọn dẹp các project maven và xây dựng các dịch vụ cho các tệp jar là các dịch vụ được xây dựng trong Java maven.
- Lệnh “`docker build --tag=microservice-kubernetes-demo-apache apache`” xây dựng ảnh docker bằng cách sử dụng thiết kế từ Docker hub là một microservice-kubernetes-demo-apache. Apache là thư mục chứa Dockerfile cần để xây dựng ảnh docker ‘`docker tag microservice-kubernetes-demo-apache`

your\_docker\_account/microservice-kubernetes-demo-apache:latest' giúp gắn liền kết tag ảnh mới với tài khoản. Bằng cách này, ta đang tạo một tham chiếu đến hình ảnh để có thể deploy chúng về sau.

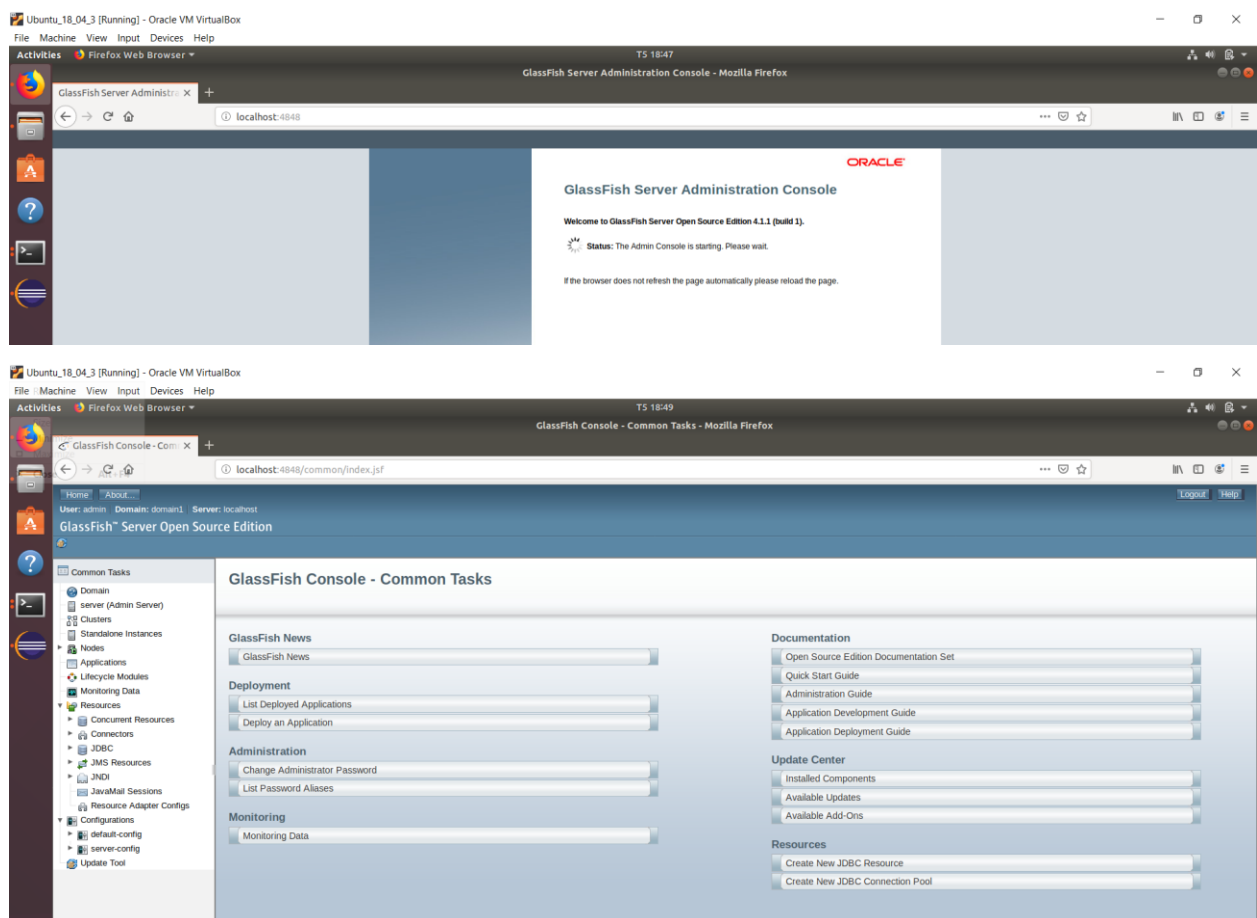
- Lệnh “docker push your\_docker\_account/microservice-kubernetes-demo-apache” được sử dụng để deploy các ảnh mới lên Docker hub.
- Các lệnh khác được dùng để clone một git repository hoặc đăng nhập vào một Docker account.
- 

Câu hỏi 2: Repository chứa ảnh mới, là ảnh microservice-kubernetes-demo-apache với tag mới nhất.

Câu hỏi 3: Trạng thái của các pods vừa mới được tạo này là Container Creating, nhưng sau đó chúng chuyển sang trạng thái Running.

#### 4. Kiến trúc JMS và DDS

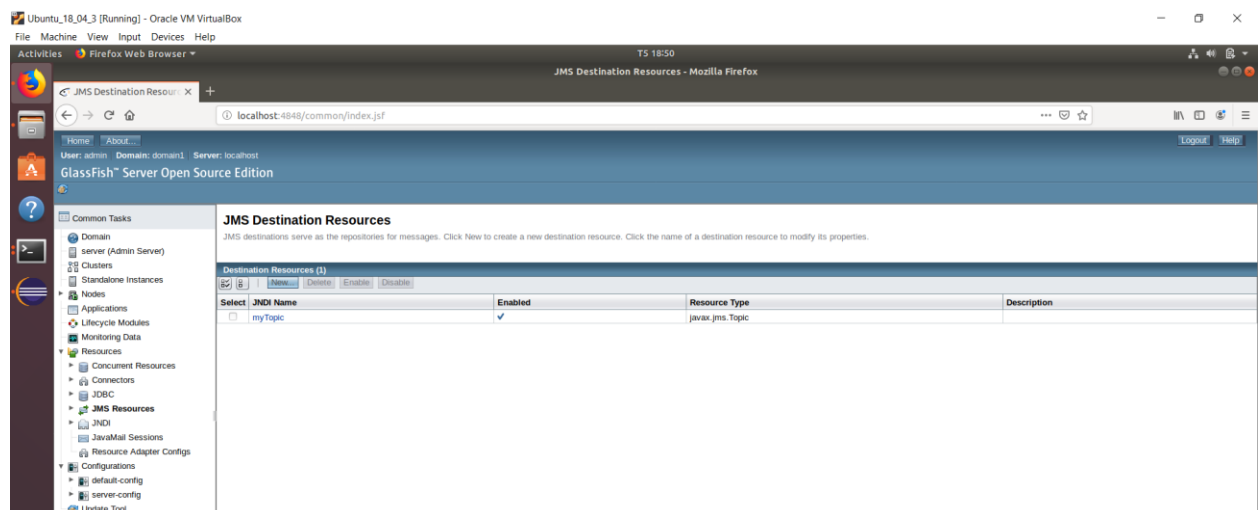
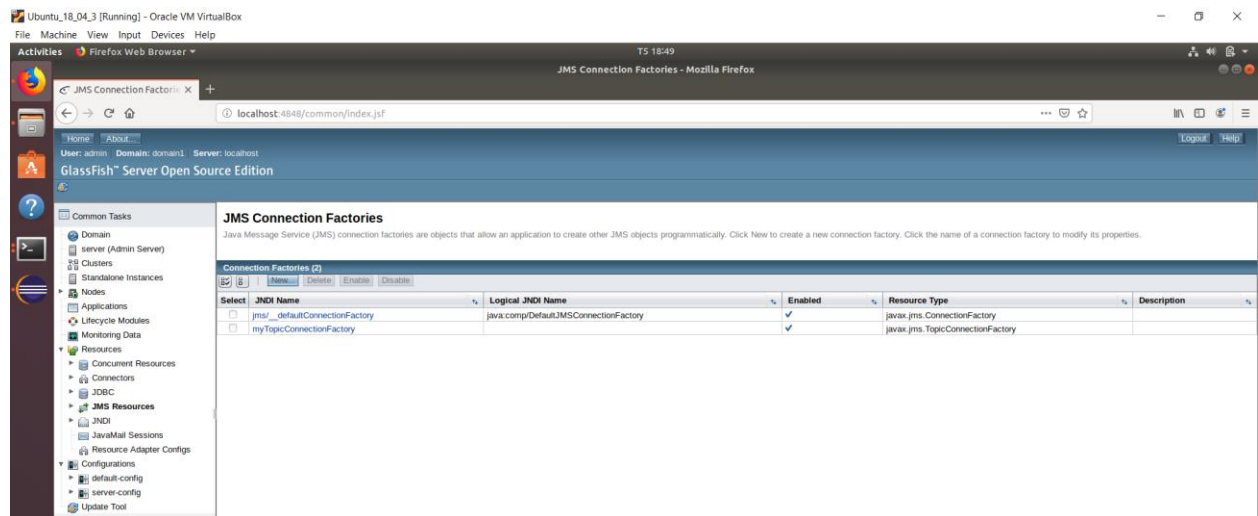
*Câu hỏi 1: Giải thích vai trò của application server glassfish.*



- Glassfish cung cấp một ứng dụng server cho phép triển khai các ứng dụng web viết bằng Java.

- Glassfish cho phép cho phép nhiều nhà phát triển có khả năng mở rộng, tích hợp các công nghệ trên web và cấu hình dễ dàng bằng giao diện đồ họa.

*Câu hỏi 2: Tại sao lại phải tạo 2 JNDI như trên?*



- Phải tạo 2 JNDI vì

myTopicConnectionFactory để GlassFish tạo một connector connection pool và connector connector resource.

myTopic để GlassFish tạo connector admin object resource.

*Câu hỏi 3: Sau khi chạy thử chương trình Sender và Receiver, vận dụng lý thuyết kiến trúc hướng sự kiện đã học trên lớp để giải thích cơ chế chuyển và nhận thông điệp của Sender và Receiver.*

Cơ chế chuyển và nhận thông điệp của Sender và Receiver:

- Cơ chế chuyển (Sender):

Khi có một nguyên nhân, kích thích làm thay đổi một sự kiện nào đó, bên Sender sẽ gửi một thông điệp (message) cho bên nhận. Thông điệp ấy được nằm trong bộ nhớ tạm, nếu có tín hiệu chuyển phát (commit) mới thì thông điệp mới được chuyển sang cho người nhận.

## - Cơ chế nhận (Receiver):

Khi nhận được một thông điệp được gửi đến. Nếu chưa kịp xử lý thì bên nhận đưa tạm vào bộ nhớ đệm, nếu bộ nhớ đầy thì báo lỗi và sử dụng cơ chế rollback gửi lại cho người gửi. Trong quá trình xử lý gặp lỗi thì phía bên nhận cũng phải báo lại cho bên gửi

*Câu hỏi 4: So sánh JMS và DDS.*

## Điểm giống:

Đặc điểm	JMS – Java Messaging	DDS – Data distributed service
Cấu trúc	Publish / Subscribe	Publish / Subscribe (đa hướng)
Nền tảng độc lập	Cùng một API được hiển thị cho tất cả HW, OS và các ngôn ngữ được hỗ trợ	Cùng một API được hiển thị cho tất cả HW, OS và các ngôn ngữ được hỗ trợ

## Điểm khác:

Đặc điểm	JMS – Java Messaging	DDS – Data distributed service
Khám phá các điểm cuối	Máy chủ JNDI và JMS phải được chỉ định và cấu hình	Khám phá động, không cần chỉ định nơi các điểm cuối cư trú
Loại an toàn	Đối tượng Tổng quát (Generic) và XML không phải là loại an toàn	An toàn kiểu cách mạnh mẽ, các cuộc gọi ứng dụng write () và read () với một kiểu dữ liệu cụ thể
Điều chỉnh hành vi giao tiếp	Khả năng điều chỉnh thông tin liên lạc hạn chế	Các chính sách QoS cho phép dễ dàng điều chỉnh các hành vi ứng xử giao tiếp
Khả năng tương tác	Không có	Tiêu chuẩn mở với khả năng tương tác đã được chứng minh