**CHƯƠNG 2: Bài thực hành số 2**

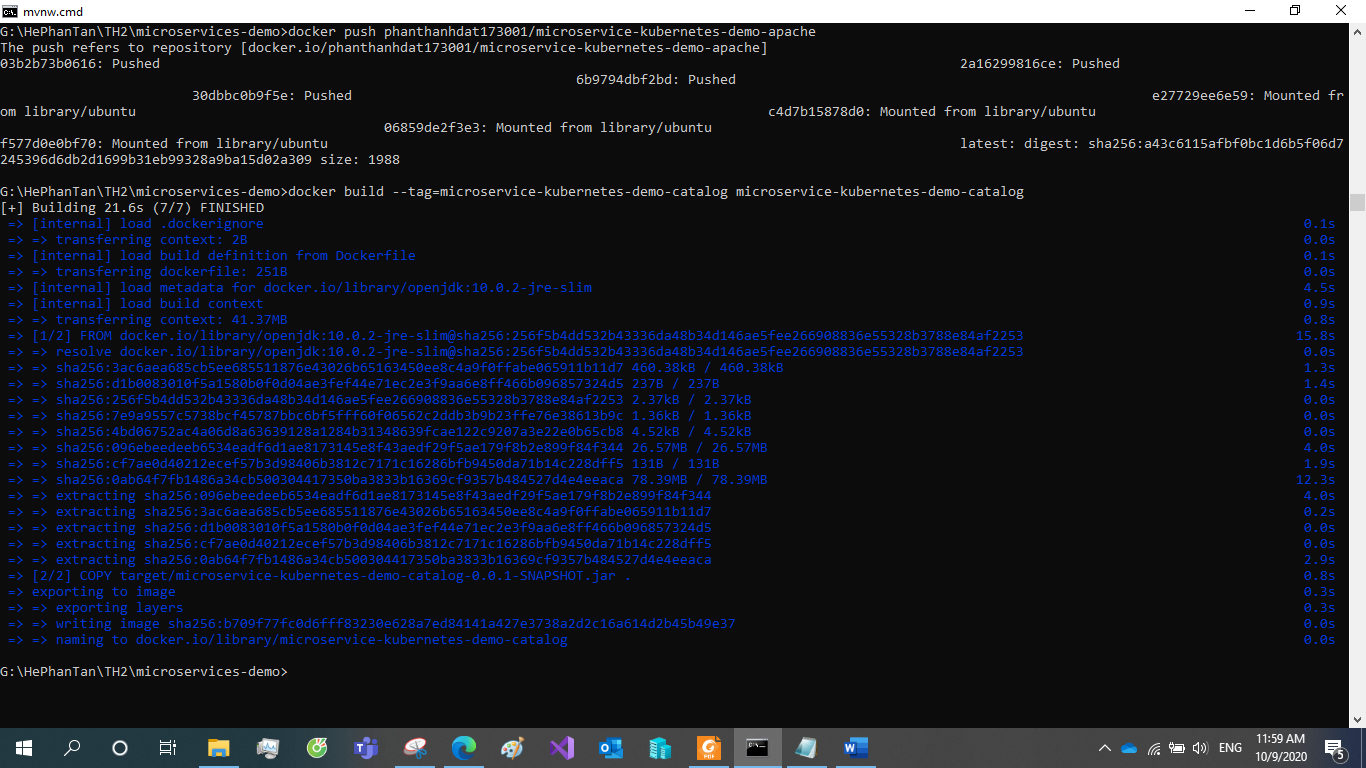
**Họ tên sinh viên: Phan Thành Đạt MSSV:20173001**

**Mã Lớp:** **118636 Mã học phần: IT4611**

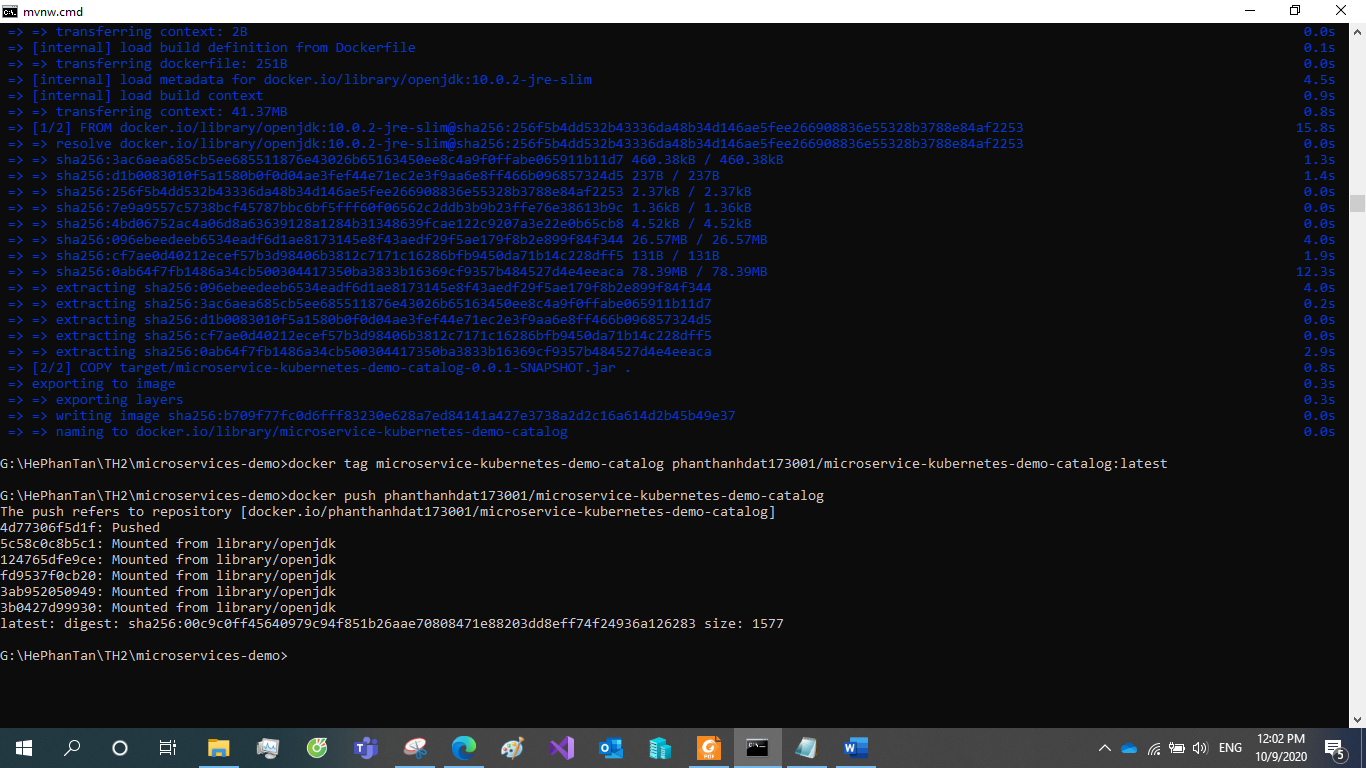
**Câu 1:**

1. **Dịch vụ Catalog**

**Build**

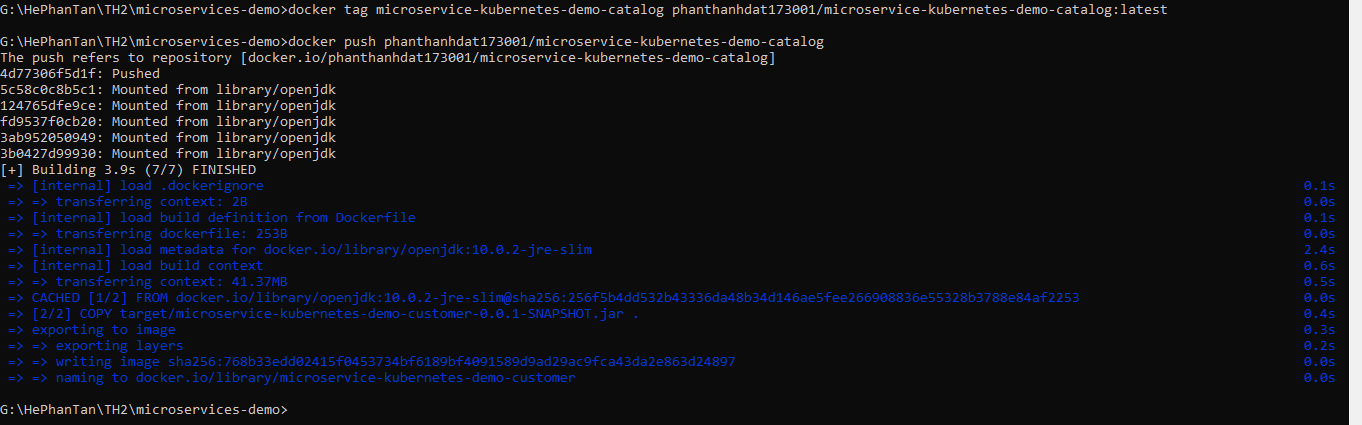


**Docker Image and push**

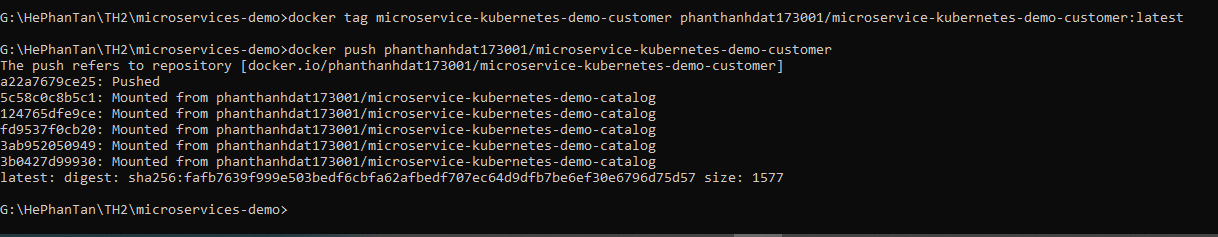


1. **Dịch vụ Customer**

**Build**

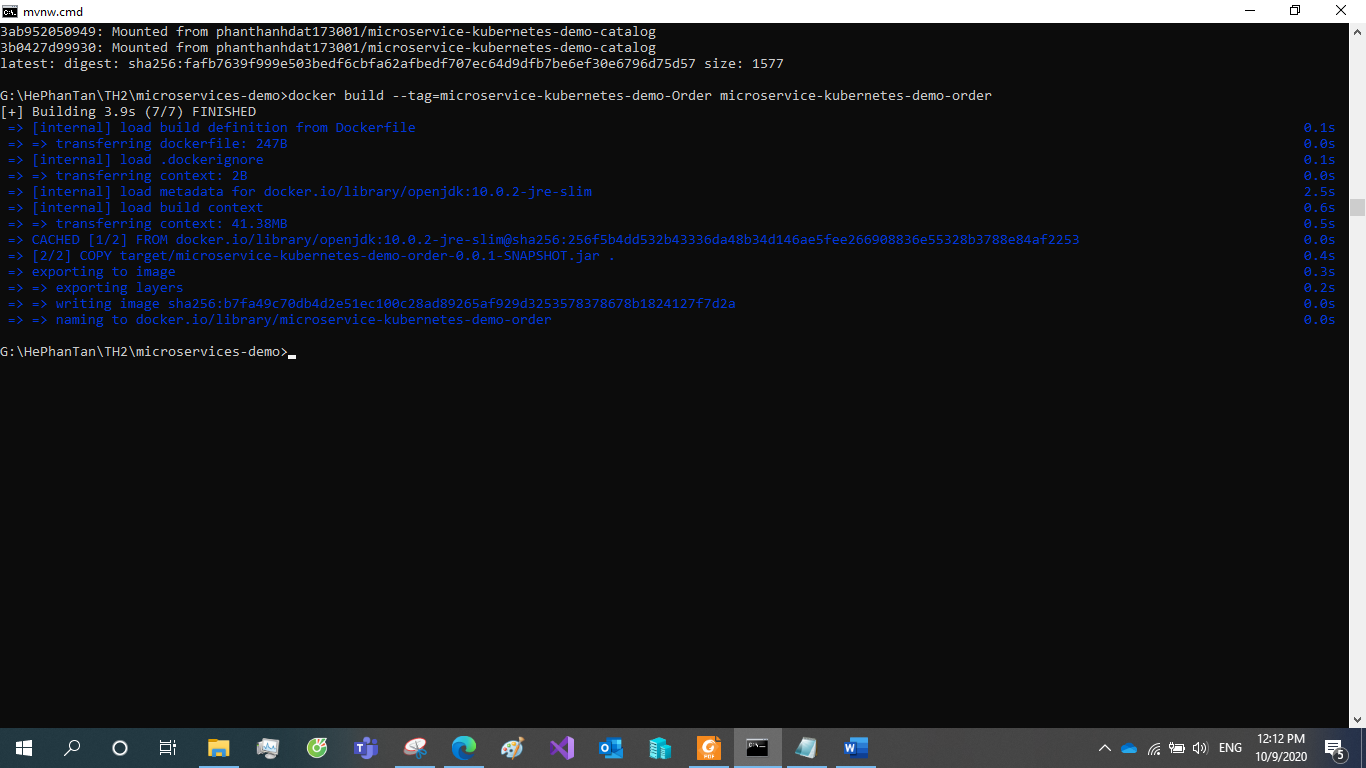


**Docker Image and Push**

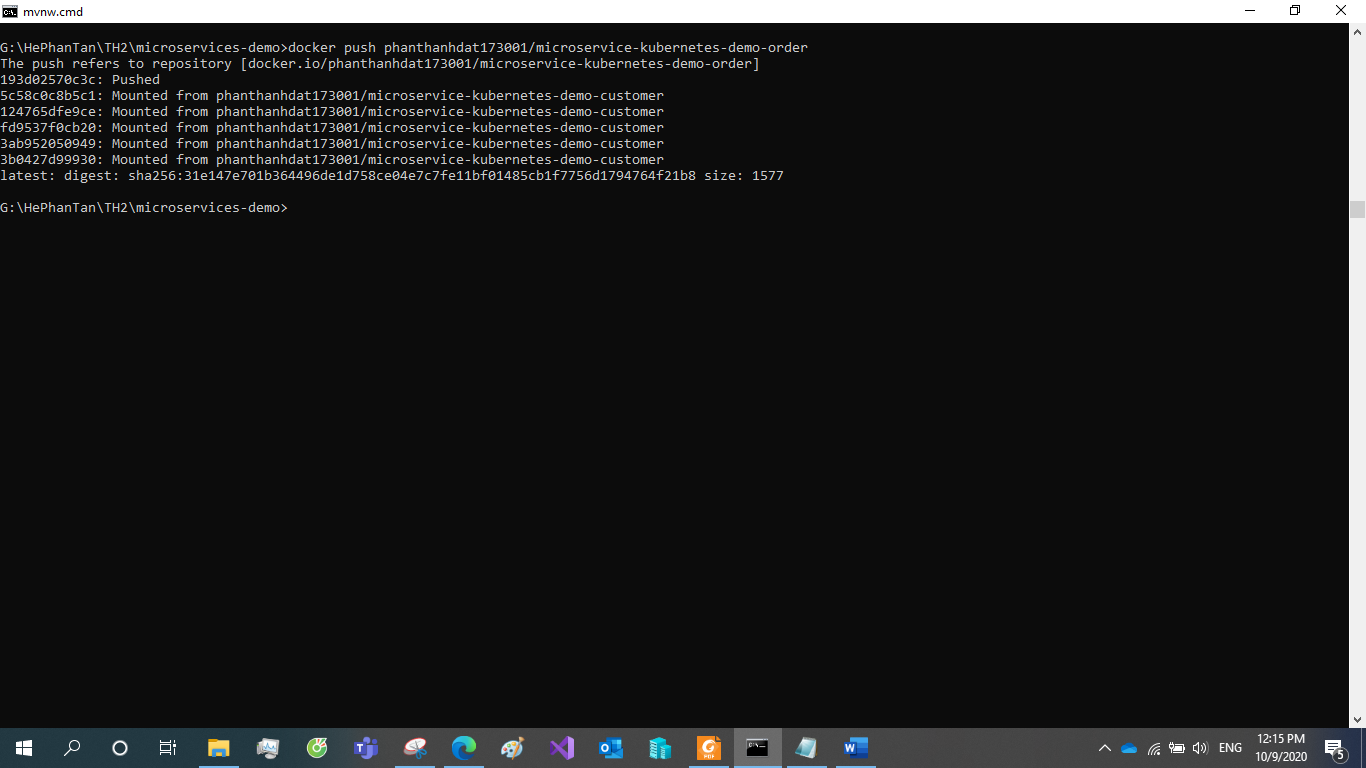


1. **Dịch vụ Order**

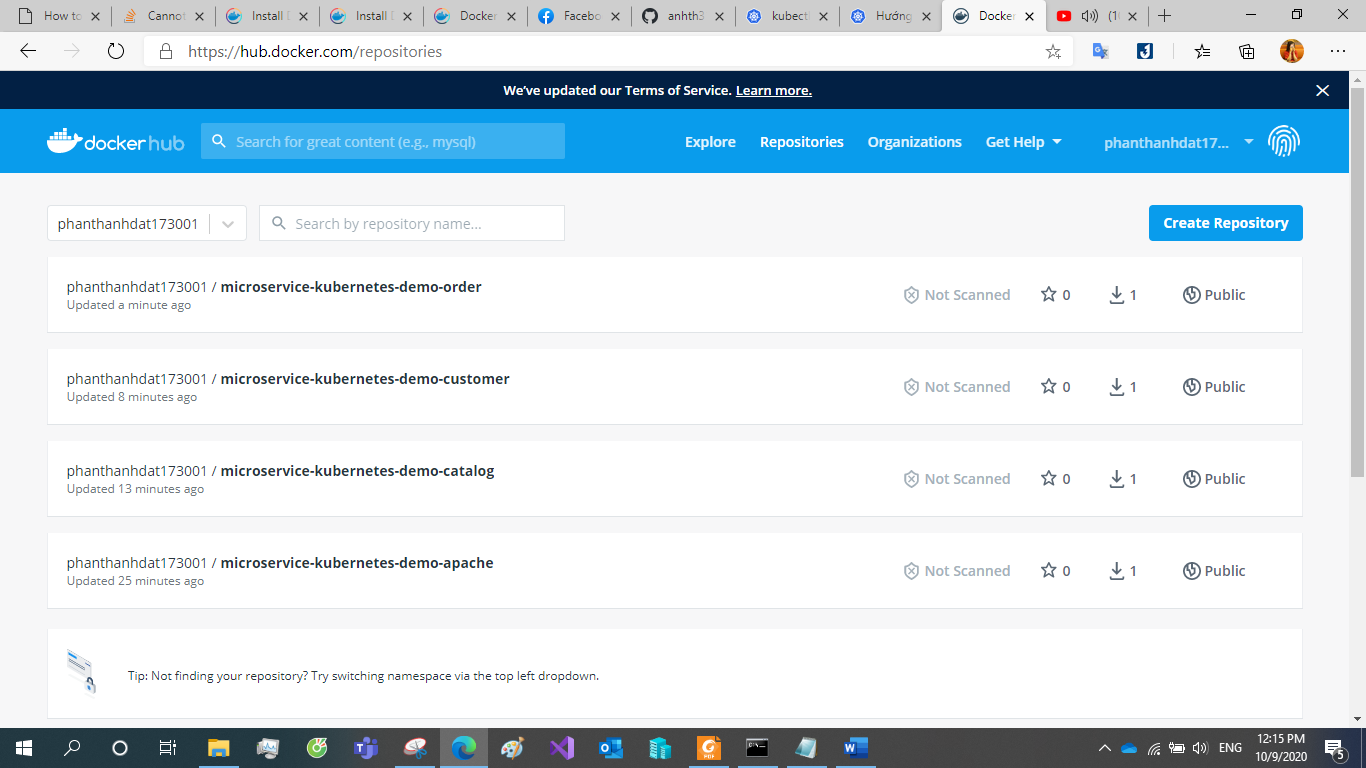
**Build**



**Docker Image and Push**



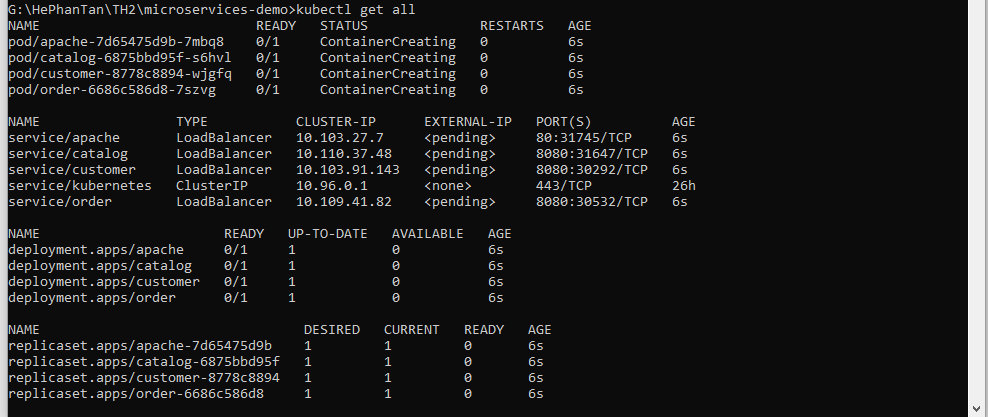
**Câu 2:**



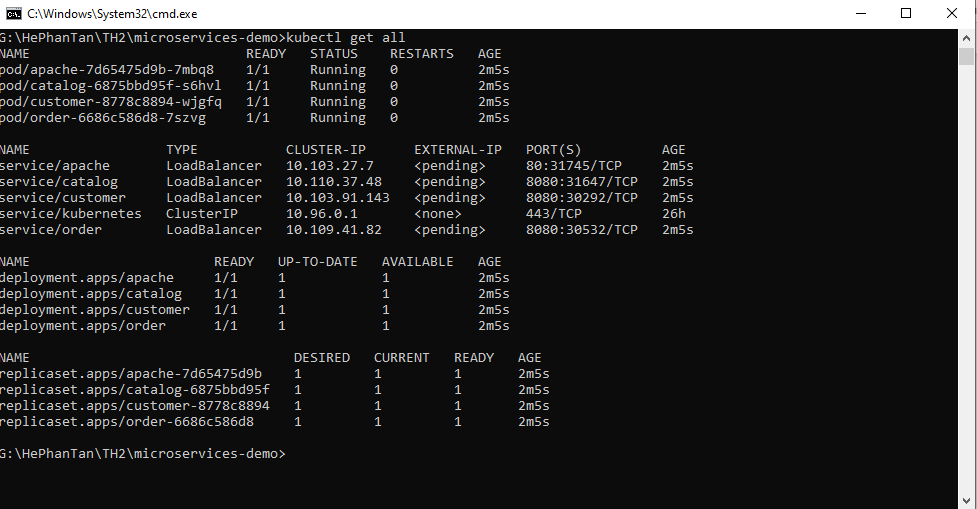
**Sau khi thực hiện, trên docker hub repository có 4 repository được push.**

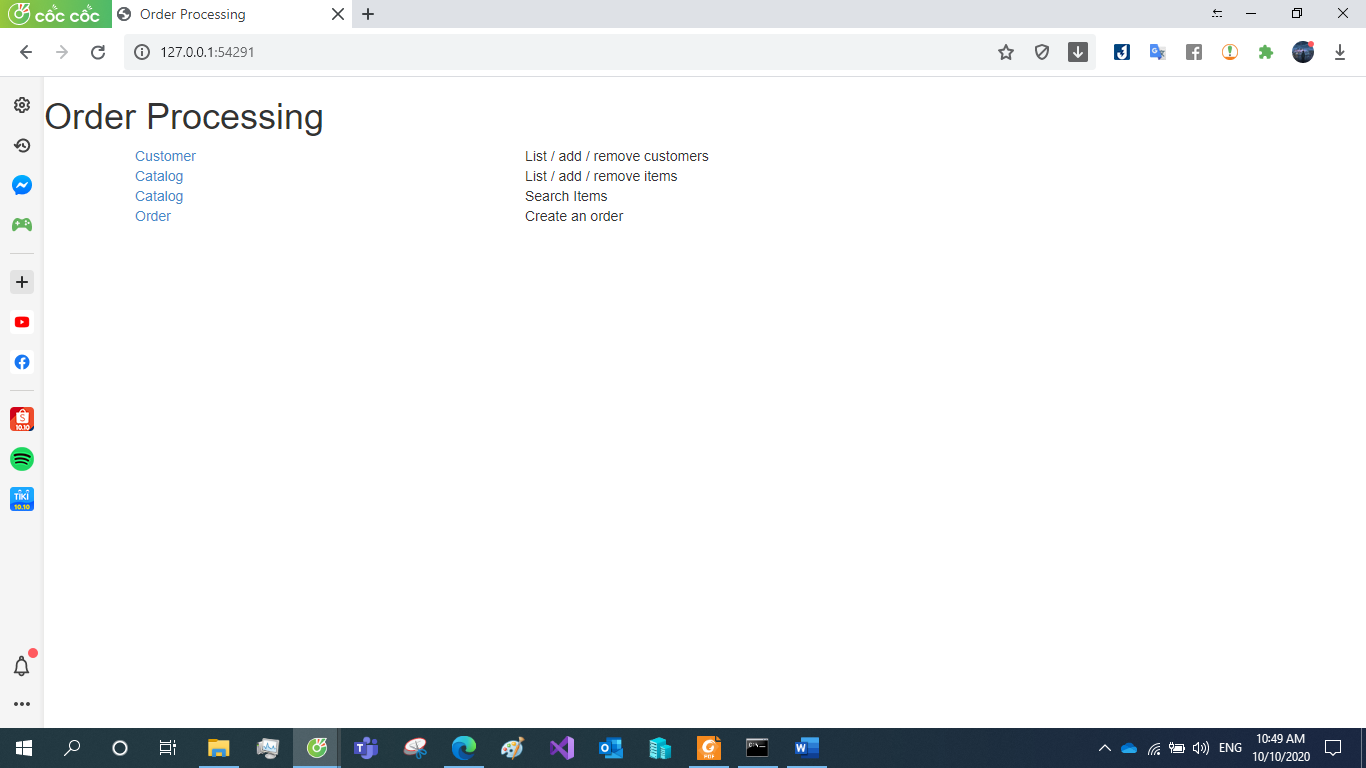
**Câu 3:**

Ban đầu sẽ thấy trạng thái ContainerCreating



Sau đó sẽ chuyển sang trạng thái running





**Câu 4**

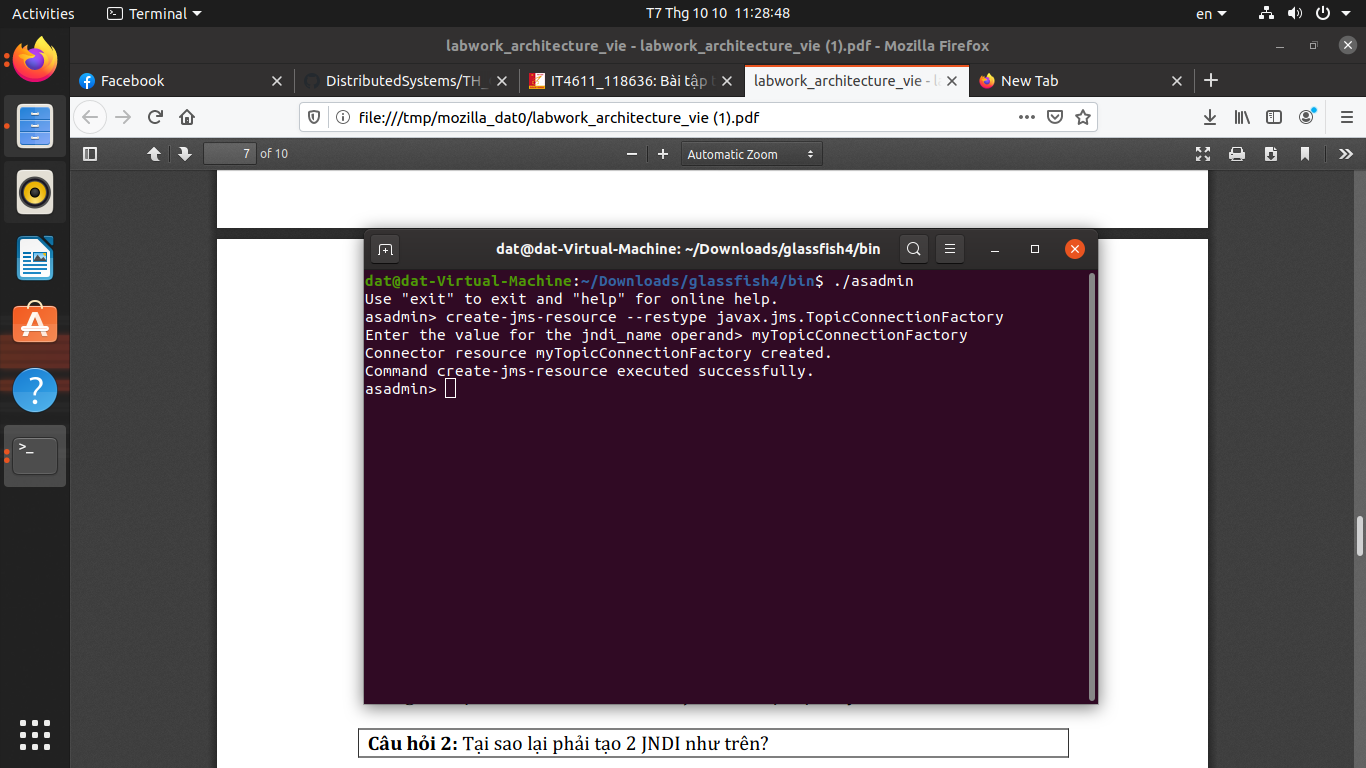
Vai trò của application server glassfish:

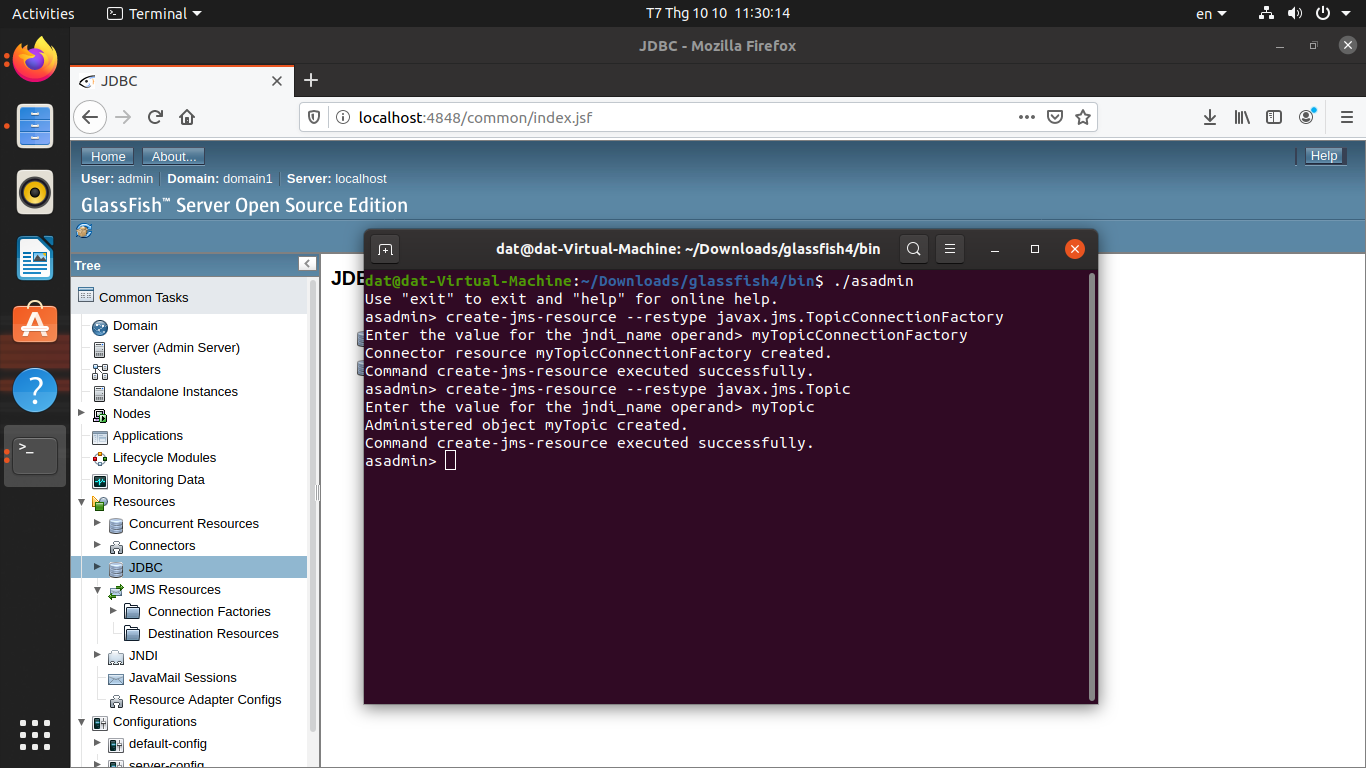
* Glassfish server là một Webserver, cho phép triển khai các ứng dụng web viết bằng Java lên đó sử dụng giao thức http.
* Glassfish server cung cấp môi trường nơi các ứng dụng có thể chạy.
* Glassfish server cho phép quản lý, cấu hình nhanh và linh hoạt.

**Câu 5:**

Chúng ta phải tạo 2 JNDI vì:

* myTopicConnectionFactory: Chính là Broker – Nhà môi giới. Nơi đây sẽ quản lý các kết nối, thiết lập các session cho Publisher và Subcrieber.
* Topic: Chính là hàng đợi tin nhắn, các Puslisher sẽ gửi tin nhắn đến 1 kho (tên kho là “myTopic”) do Broker quản lý, các Subcrieber sẽ đến kho nhận tin nhắn và xử lý. Vì Broker quản lý rất nhiều loại thông điệp, tức rất nhiều kho nên phải chỉ rõ là kho nào.





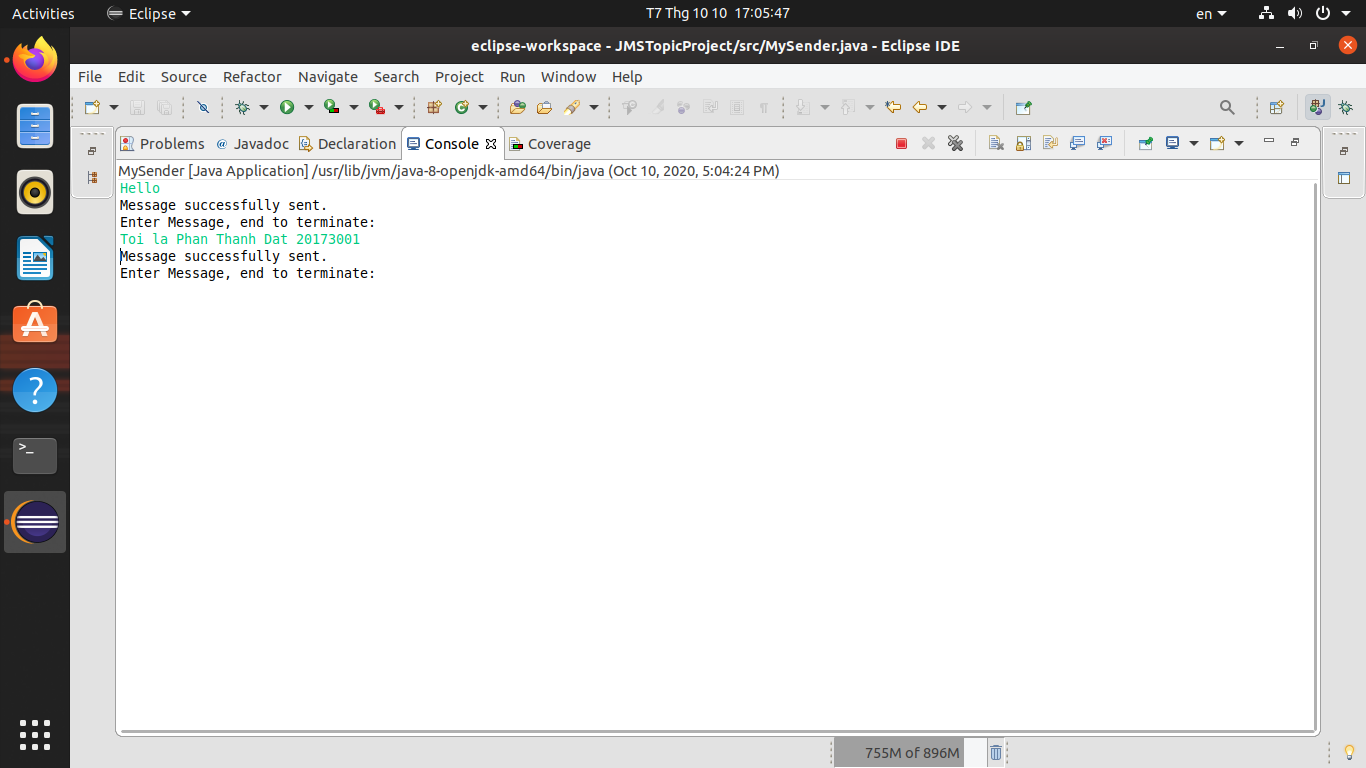
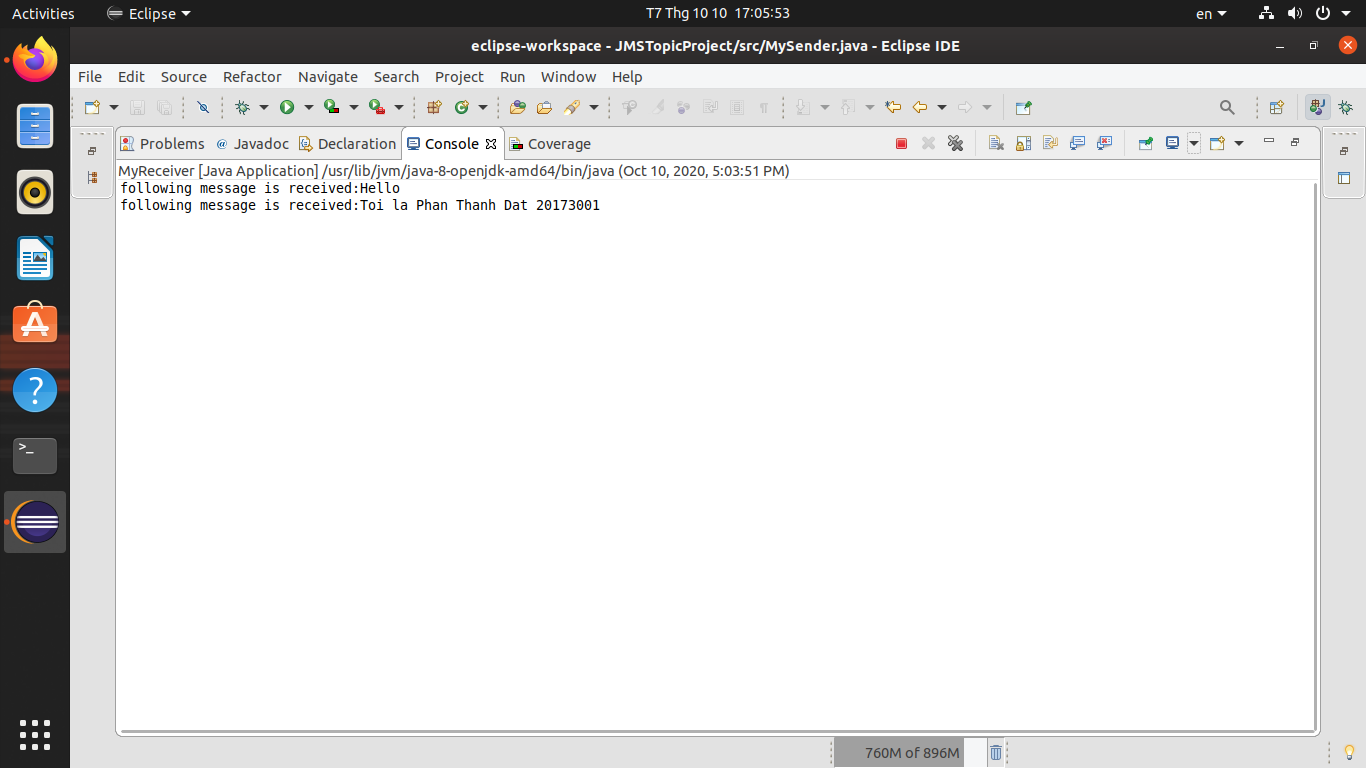
**Câu 6:**

Sender:

* Tìm kiếm và bắt đầu kết nối với một Connection Factory(Broker).
* Tạo hàng đợi Session
* Tìm kiếm và bắt đầu kết nối với Topic của Connection Factory.
* Tạo một đối tượng Publisher
* Tạo đối tượng thông điệp
* Ghi thông điệp
* Đẩy đối tượng thông điệp đó và Publisher
* Kết thúc Connection.

Receiver

* Tìm kiếm và bắt đầu kết nối với một Connection Factory(Broker).
* Tạo hàng đợi Session
* Tìm kiếm và bắt đầu kết nối với Topic để nhận Messages.
* Tạo đối tượng Subcrieber
* Tạo đối tượng Listen khi Subcrieber nhận được Messages.
* Đăng ký Subcriber với Connection Factory (Broker).

**Câu 7:**

|  |  |
| --- | --- |
| JMS | DDS |
| * Java Messaging Service * Tập trung * Middleware API bởi Java Messager * Sử dụng cho việc gửi thông điệp giữa 2 hoặc nhiều client. * Cho phép giao tiếp giữa các thành phần khác nhau của mộ ứng dụng phân tán * Xây dựng bằng ngôn ngữ java | * Data distributed service * Phi tập trung * DDS là phần mềm mạng trung gian giúp đơn giản hóa sự phức tạp của lập trình mạng * Là một thể hiện của mô hình pub/sub cho việc gửi, nhận dữ liệu, sự kiện giữa các node * Có thể được xây dựng bởi nhiều ngôn ngữ khác nhau, nhiều nền tảng khác nhau. |