

BÀI TẬP TRÊN LỚP
MÔN HỌC: HỆ PHÂN TÁN
CHƯƠNG 2: KIẾN TRÚC TRONG HỆ PHÂN TÁN

HỌ TÊN SV: Mạc Quang Huy
MÃ LỚP: 118636

MSSV: 20173169
MÃ HỌC PHẦN: IT4611

I. Kiến trúc Microservices

Câu hỏi 1: Hãy thực hiện gõ lệnh trên với 3 dịch vụ còn lại.

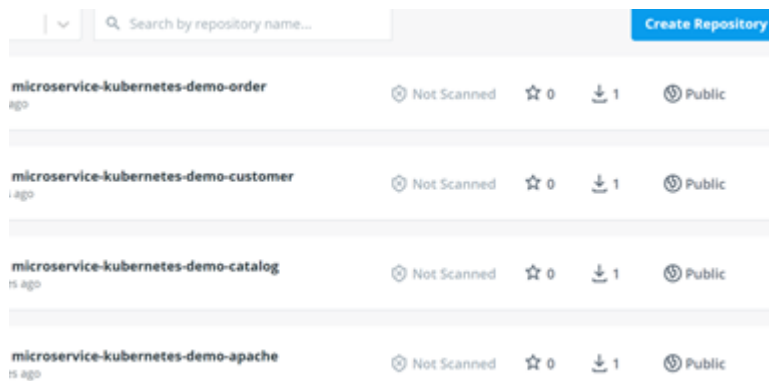
➤ Dịch vụ catalog

- docker build --tag=microservice-kubernetes-demo-catalog microservice-kubernetes-demo-catalog
- docker tag microservice-kubernetes-demo-catalog huymq/microservice-kubernetes-demo-catalog:latest
- docker push huymq/microservice-kubernetes-demo-catalog

Các dịch vụ order, customer thực hiện đưa ra kết quả tương tự.

Câu hỏi 2: Vào trang web DockerHub và đăng nhập vào tài khoản của bạn. Bạn thấy những gì mới xuất hiện trên docker hub repository của bạn?

➤ Đăng nhập vào trang hub.docker.com xuất hiện file các service vừa thực hiện ở phía trên



Câu hỏi 3: Trạng thái(status) của các pods vừa mới tạo được là gì? Bây giờ, hãy chờ vài phút và gõ lại lệnh đó, trạng thái mới của các pods giờ đã chuyển thành gì?

➤ Các status của các Pods vừa mới tạo ra

NAME	READY	STATUS	RESTARTS	AGE
pod/apache-585f4495b6-nlgqf	0/1	ContainerCreating	0	26s
pod/catalog-58775f4db5-hnlth	0/1	ContainerCreating	0	26s
pod/customer-86484f669b-pz4xp	0/1	ContainerCreating	0	26s
pod/order-56fbc4c9db-h8bwu	0/1	ContainerCreating	0	26s

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/apache	LoadBalancer	10.100.0.243	<pending>	80:30102/TCP	26s
service/catalog	LoadBalancer	10.106.156.135	<pending>	8080:32686/TCP	26s
service/customer	LoadBalancer	10.104.137.255	<pending>	8080:30256/TCP	26s
service/kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	4m5s
service/order	LoadBalancer	10.97.133.130	<pending>	8080:32631/TCP	26s

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/apache	0/1	1	0	26s
deployment.apps/catalog	0/1	1	0	26s
deployment.apps/customer	0/1	1	0	26s
deployment.apps/order	0/1	1	0	26s

NAME	DESIRED	CURRENT	READY	AGE
replicaset.apps/apache-585f4495b6	1	1	0	26s
replicaset.apps/catalog-58775f4db5	1	1	0	26s
replicaset.apps/customer-86484f669b	1	1	0	26s
replicaset.apps/order-56fbc4c9db	1	1	0	26s

➤ Trạng thái pods chuyển thành

NAME	READY	STATUS	RESTARTS	AGE
pod/apache-585f4495b6-nlgqf	1/1	Running	0	4m21s
pod/catalog-58775f4db5-hnlth	0/1	ImagePullBackOff	0	4m21s
pod/customer-86484f669b-pz4xp	1/1	Running	0	4m21s
pod/order-56fbc4c9db-h8bwu	1/1	Running	0	4m21s

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/apache	LoadBalancer	10.100.0.243	<pending>	80:30102/TCP	4m21s
service/catalog	LoadBalancer	10.106.156.135	<pending>	8080:32686/TCP	4m21s
service/customer	LoadBalancer	10.104.137.255	<pending>	8080:30256/TCP	4m21s
service/kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	8m1s
service/order	LoadBalancer	10.97.133.130	<pending>	8080:32631/TCP	4m21s

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/apache	1/1	1	1	4m21s
deployment.apps/catalog	0/1	1	0	4m21s
deployment.apps/customer	1/1	1	1	4m21s
deployment.apps/order	1/1	1	1	4m21s

NAME	DESIRED	CURRENT	READY	AGE
replicaset.apps/apache-585f4495b6	1	1	1	4m21s
replicaset.apps/catalog-58775f4db5	1	1	0	4m21s
replicaset.apps/customer-86484f669b	1	1	1	4m21s
replicaset.apps/order-56fbc4c9db	1	1	1	4m21s

II. Kiến trúc JMS và DDS

Câu hỏi 1: Vai trò của application server glassfish

Application server glassfish là một application server mã nguồn mở được phát triển dựa trên nền JavaEE hỗ trợ đầy đủ các tính năng cần thiết của JavaEE như web ứng dụng, Enterprise JavaBeans, JPA, JavaServer Faces, JMS, RMI, JavaServer Pages, Servlets, etc. Nó cho phép những nhà phát triển ứng dễ quản lý, cấu hình nhanh và linh hoạt. Glass server support nhiều nền tảng như windows, Linux, Mac Os...

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

Tạo 2 JDNI mà không cần quan tâm đến dịch vụ bên trong. Sử dụng để gọi 2 dịch vụ TopicConnection và Topic

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

- MySender vai trò publisher. Tin nhắn được gửi đi khi nhận có điều kiện: Nhận được thông điệp từ MySender
- Khi server nhận được thông tin, MyReceiver đóng vai trò là subscriber, nhận được thông tin, tự động kích hoạt hàm onMessage của MyListener và hiển thị thông điệp ra màn hình.

Câu hỏi 4: So sánh JMS và DDS

Trả lời:

	DDS	JMS
Nền tảng	Hỗ trợ nhiều nền tảng	Chỉ chuyên cho Java
Ưu điểm của DDS so với JMS	Độ trễ thấp ⇒ Phù hợp các hệ thống quân sự, báo cháy,..	Có độ trễ ⇒ Phù hợp với các jobs không cần thời gian thực(độ trễ nhỏ)