Student’s name: Nguyễn Đức Dũng

Class: ICT.02-K61

**Class Exercises**

**Module: Distributed Systems**

**Chapter 2: Architectures**

**Practical Exercises:**

Question 1: What are the commands did you use?

* Docker login: login to a docker hub account
* Docker build: build the current folder into an image
* Docker tag: assign an image with a tag in order to determine the image as a version of service
* Docker push: push an image into the docker hub
* docker build --tag=dungnd/microservice-kubernetes-demo-apache apache
* docker build --tag=dungnd/microservice-kubernetes-demo-catalog microservice-kubernetes-demo-catalog
* docker build --tag=dungnd/microservice-kubernetes-demo-customer microservice-kubernetes-demo-customer
* docker build --tag=dungnd/microservice-kubernetes-demo-order microservice-kubernetes-demo-order
* docker image push dungnd/microservice-kubernetes-demo-apache
* docker image push dungnd/microservice-kubernetes-demo-order
* docker image push dungnd/microservice-kubernetes-demo-customer
* docker image push dungnd/microservice-kubernetes-demo-catalog

Question 2: Open the website Docker Hub and login with your account. What’s new in your docker hub repository?

We have four new images in our docker hub.

* microservice-kubernetes-demo-apache
* microservice-kubernetes-demo-order
* microservice-kubernetes-demo-customer
* microservice-kubernetes-demo-catalog

Question 3: What is the status of these created pods? Now, wait few minutes and re-type this command, what is the new status of these pods?

At initial, all services have status ContainerCreating, and after retyping the command, all the services have new status Running.

Question 4: What is the role of application server glassfish?

GlassFish is the reference implementation of Java EE and as such supports Enterprise JavaBeans, JPA, JavaServer Faces, JMS, RMI, JavaServer Pages, servlets, etc. This allows developers to create enterprise applications that are portable and scalable, and that integrate with legacy technologies.

Question 5: Why do we need to create the 2 JNDI above?

The myTopicConnectionFactory JDNI is used to manage the connection creation so that the system can reduce overload. And, the myTopic JDNI is used for messaging service.

Question 6: Explain the message passing method of Sender and Receiver in basing on the theory of event-based architecture.

Sender is created as a Publisher for the topic “myTopic” and it will call the method publish() of TopicPublisher to send the message to the channel “myTopic”. Receiver is created as a Subscriber of the topic “myTopic” and it implements a message listener. This message listener will always listen the messages that is published by topic’s publishers and can get messages by calling method onMessage().

Question 7: Compare the JMS and DDS.

|  |  |
| --- | --- |
| JMS | DDS |
| Focuses only on application portability across implementation of the standard | Ensures portability and interoperability across implementation of the standard |
| Promotes fully distributed architectures | Promotes Hub and Spoke architectures |
| Provides mechanisms for transparently using Brokers/Routers | Fully distributed architectures are possible but complicated by some use cases |