Module: ITS 4243 Microservices and Cloud Computing
Assignment: 2
Index No: ICT/18/813
Name: Dissanayake D.A.N.P.

Part 1 - A

1.

Microservices, also known as the microservices architecture, is an architectural style that structures an application as a collection of small autonomous services, modelled around a business domain. Each microservice is self-contained and should implement a single business capability. They communicate with each other via APIs and can be independently deployed and scaled.

2

Container Runtime Platform

A container runtime platform is a system that manages and runs containers. Examples include Docker, containerd, and CRI-O. It provides an environment where you can execute and manage container instances.

Container Image

A container image is a lightweight, standalone, executable software package that includes everything needed to run a piece of software, including the code, a runtime, libraries, environment variables, and config files. It's essentially a snapshot of a container that can be executed on any platform that supports containerization.

3.

Container Orchestration Tools

These are systems that automate and control the lifecycle of containers in a microservices architecture. They assist with deployment, scaling, networking, and availability of containers. Kubernetes is a popular example.

Container Registry

This is a repository for storing container images. Docker Hub and Google's Container Registry are examples.

Infrastructure Monitoring Tools

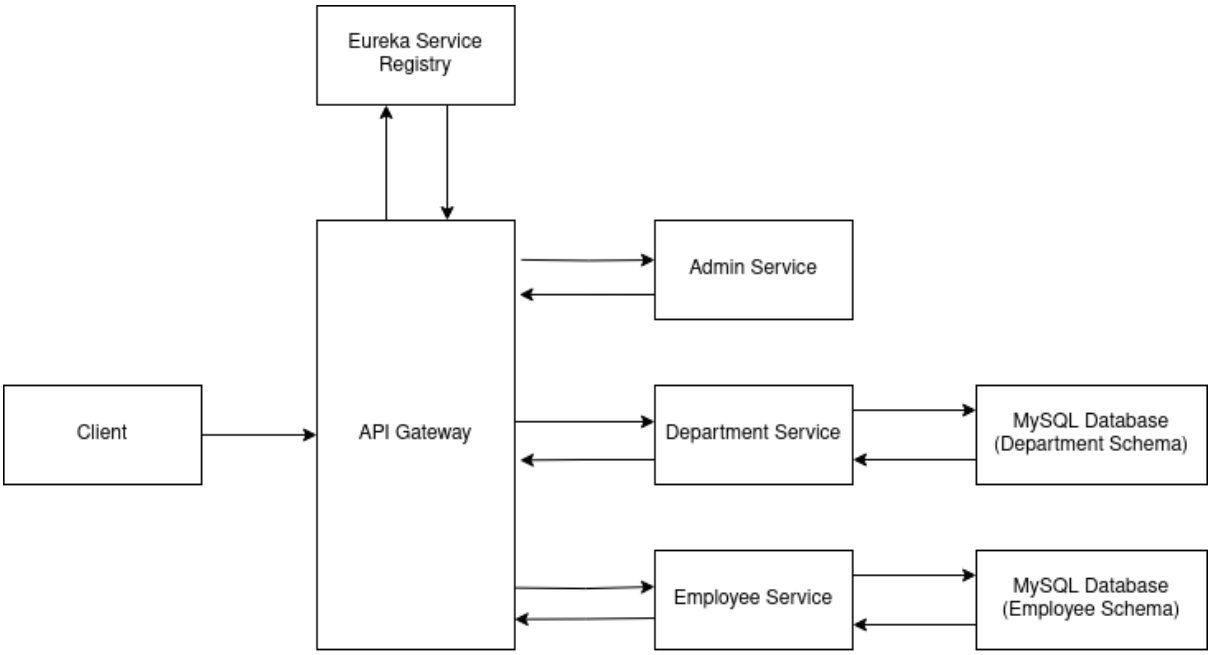
These tools provide insights into the performance and health of your microservices running inside containers.

Network Infrastructure

For containers to communicate, a network infrastructure capable of service discovery, load balancing, and secure inter-service communication is needed.

Part 1 - B

The following is a high level design diagram for the given scenario.



Implementation and source code

- https://github.com/ipmanlk/usjp-microservices-cw-3

Necessary steps required if the client further requires to containerize the above solution.

1. **Install Docker**: The user needs to ensure Docker is installed on their machine. Docker will be used to create and manage the containers for the services.
2. **Create Dockerfiles**: For each of the microservices (Admin, Employee, Department), a Dockerfile is created in the root directory. The Dockerfile contains the instructions to build the Docker image for the service. It specifies the base image, copies the service's jar file into the image, and defines how to start the service.
3. **Build Docker Images**: The user should then navigate to each service's directory and use the Docker CLI with “docker build” command to build a Docker image based on the Dockerfile. This creates a Docker image for each service.

4. **Create Docker Compose File**: In the root directory of the project, a docker-compose.yml file is created. This file defines each of the services (including any databases or other dependencies) and how they should interact. The user defines which Docker image to use for each service, any environment variables needed, and any ports that should be exposed or mapped to the host system.

5. **Start the Services**: Using the Docker CLI, the “docker compose up” command is used to start all services. Docker Compose reads the docker-compose.yml file, starts each of the services in separate containers, and sets up networking between them as defined.

6. **Test the Services**: With the services running, they can be tested to ensure they are working as expected. The services can be accessed through the defined ports on the localhost.

7. **Push Docker Images to Repository**: If the plan is to deploy the services to a server or a cloud provider, the Docker images need to be pushed to a Docker repository. This could be Docker Hub, or a repository provided by a cloud provider.

8. **Deploy the Services**: If deploying to a server, the Docker images can be pulled from the repository and Docker Compose can be used to start the services. If deploying to a cloud provider, they will likely have their own instructions for deploying Docker containers.

Part 2

1.

Cloud Service Model	Description
IaaS (Infrastructure as a Service)	Provides virtualized computing resources, such as VMs, storage, and networking.
PaaS (Platform as a Service)	Offers a platform for developing, testing, and deploying applications without managing the underlying infrastructure.
SaaS (Software as a Service)	Delivers fully functional software applications to users over the internet.

Cloud Deployment Model	Description
Public Cloud	Cloud services are provided over the internet and shared among multiple users.
Private Cloud	Cloud infrastructure is dedicated to a single organization, offering more control and security.
Hybrid Cloud	A combination of public and private clouds, enabling data and applications to be shared between them.

2.

Total Cost of Ownership is an important idea to think about when looking into cloud computing choices. It means figuring out all the costs that come with buying, setting up, and using a cloud-based system or service. TCO includes the direct costs like hardware, software, and licenses, as well as indirect costs like training employees, managing the system, fixing problems, and using energy. When organizations know the TCO, they can make better choices about which cloud solutions are the best deal, considering both the starting costs and the costs they will have to pay over time.

3.

Cost Efficiency

One of the most compelling reasons businesses should move to the cloud is the potential for significant cost savings. With cloud computing, companies can reduce the high costs of hardware, software, and system setups. The cloud follows a pay-as-you-go model, so you only pay for the resources you use.

Scalability & Flexibility

Cloud environments allow businesses to easily scale up or down their IT requirements as needs change and grow. This agility can give businesses using cloud computing a real advantage over competitors.

Disaster Recovery & Data Loss Prevention

With the data stored in cloud, it's backed up to a secure location offsite, reducing the potential for business downtime caused by local disasters such as fires, floods, or power outages. Also, the data in the cloud can be accessed from anywhere with an internet connection, which means business can continue as usual, even in the event of a disaster.

Automatic Software Updates

In the cloud, servers are off-premises and out of sight. The cloud service providers take care of them for you and roll out regular software updates, including security updates, so you don't have to worry about wasting time on maintaining the system yourself.

Work from Anywhere

With cloud computing, if you've got an internet connection, you can be at work. Businesses can offer more flexible working perks to employees so they can enjoy the work-life balance that suits them.

Improved Collaboration

Cloud-based workflow and file sharing apps allow dispersed groups of people to work together more easily and efficiently. Teams can edit files simultaneously and receive real-time updates.

Environmentally Friendly

Businesses using cloud services only use the server space they need, which decreases their carbon footprint. Using the cloud results in less energy consumption and carbon emissions than on-site servers.

4.

Planning Phase

This phase involves understanding the current IT infrastructure, identifying the needs of the business, and defining the goals of the migration. I would also select a suitable cloud service provider in this phase.

Assessment Phase

I would evaluate the readiness of applications and data for migration. This includes identifying dependencies, assessing the complexity of the move, estimating costs, and identifying potential risks.

Design Phase

Here, I would design the target state for the cloud environment. This includes the architectural design of the cloud environment, decisions about network topology, security considerations, and compliance controls.

Migration Phase

In this phase, applications, data, and other business elements are moved to the cloud. This can be done gradually, moving one or a few systems at a time (often referred to as a "strangler" approach), or all at once ("big bang" approach). The choice between these strategies depends on the business requirements and the nature of the systems being migrated.

Validation Phase

After the migration, I would validate that the systems are working as expected in the new environment. This includes functionality testing, performance testing, and security validation.

Optimization Phase

After the systems are operational in the cloud, the environment would be continuously monitored and optimized for cost, performance, security, and reliability. This might involve tuning performance settings, automating tasks, improving security measures, or other activities.

5.

Positives	Negatives
Cost-Effective: Reduces the cost of IT infrastructure and maintenance.	**Dependency on Service Provider**: Your operations can be affected by the downtime of the service provider.
Scalability: Easy to scale resources up or down to match business needs.	**Security and Compliance Issues**: While cloud providers offer security measures, sensitive data is still vulnerable to cyber threats.
Accessibility: Data and applications can be accessed from anywhere, facilitating remote work.	**Limited Control**: As the hardware is owned by the provider, you have limited control over it.
Automated Updates: Cloud providers take care of routine software updates, including security updates.	**Data Migration**: Moving large amounts of data to the cloud can be time-consuming and potentially costly.
Disaster Recovery: Cloud-based backup and recovery solutions are often more robust and less costly than on-premise solutions.	**Internet Dependency**: Cloud services require a reliable internet connection. Without it, you might not be able to access your data and applications.