

List the necessary steps required if the client further requires containerizin

- Set up Docker: Install Docker on your development machine and ensure it is running properly.
- Dockerize Microservices: Microservices should be Dockerized by producing a Docker file for each microservice (Admin, Employee, Department). A Docker file outlines how to create a Docker image for the application.
- Define Base Image: For each microservice, choose a suitable base image and add it to the Docker file, for example, an OpenJDK image.
- Copy Application Files: Into the Docker image, copy the required application files (such as JAR files). This includes any dependencies as well as the produced code.
- Expose Ports: S: Inside the Docker image, specify the network ports on which the microservice will listen, and expose them if necessary.
- Push Images to Container Registry: Push the created Docker images, if necessary, to a private registry or a container registry like Docker Hub. This stage enables simple distribution and deployment of the images across various settings.
- Container Orchestration: Utilize a platform for container orchestration, such as Kubernetes or Docker Swarm, to manage and deploy the containerized microservices.
- Configure Container Environment: Configure the container environment by setting up any necessary environment variables or configuration files inside the Docker image, such as the configuration of the service discovery configuration.
- Define Deployment Manifests: For each microservice, create deployment manifests (like Kubernetes YAML files). The container image, replica count, network configuration, and any other necessary variables are all specified in these manifests as well as the desired state of the microservices.
- Configure Service Discovery: Establish service discovery within the container orchestration platform to allow for microservice communication. Utilizing platform features like overlay networks for Docker Swarm or Kubernetes Services is one way to achieve this.
- Test and Validate: Check to see if the containerized solution performs as anticipated in the containerized environment. Verify appropriate operation, access endpoints, and test the interactions between microservices.