TASK 5- DevOps

1. What is an init container and a sidecar container? Can you give a simple scenario where we use these containers?

- **Init Container:** Runs before the main container starts. Used for setup tasks like downloading files or waiting for a database to be ready.

Example: A pod needs a config file before the app starts. The init container downloads it first.

- **Sidecar Container:** Runs alongside the main container to help it (like logging, monitoring, or syncing data).

Example: A web server pod has a sidecar that collects and sends logs to a logging system.

2. Which one is the default deployment strategy? How does it work?

- **Default Strategy:** Rolling Update
- **How it works:**
  - Slowly replaces old pods with new ones, one by one.
  - Ensures no downtime (users don't notice the change).

3. Command to check the container logs in a pod?

kubectl logs <pod-name>-c <container-name>
(If the pod has only one container: kubectl logs <pod-name>)

4. What are the types of services in Kubernetes?

1. **ClusterIP** – Internal IP (only accessible inside the cluster).
2. **NodePort** – Opens a port on the node (accessible from outside).
3. **LoadBalancer** – Creates an external load balancer (cloud providers like AWS/GCP).
4. **ExternalName** – Maps a service to a DNS name (like a CNAME record).

5. What is the link between a pod and a service?

- A Service acts as a stable IP/DNS name for pods (which can come and go).
- It routes traffic to the right pods using labels.
- Example: A "web-service" sends requests to any pod with the label app: web.