**Kubernetes Hands-on Exercise:**

**NGINX Deployment with Job & Health Checks (Liveness, Readiness, Startup)**

**Objective:**

You will implement

Create a **Kubernetes Job** for initialization.

1. Deploy **NGINX** using a **Deployment** with multiple replicas.
2. Configure **Startup**, **Liveness**, and **Readiness** probes.
3. Expose the application using a **Service**.
4. Test the behavior of all probes and understand their purpose.

**Step-by-Step Exercise Instructions:**

**Step 1: Create a Kubernetes Job**

**Task:**

* Create a Job resource named init-job.
* Use the busybox or alpine image to simulate a task (e.g., echo message + sleep for 5s).
* Set restartPolicy to Never.
* Set backoffLimit to control retries.

**Goal:**  
Verify one-time task execution and completion.

**Step 2: Create an NGINX Deployment**

**Task:**

* Create a Deployment named nginx-deployment.
* Use the official nginx image.
* Set replicas to 2.
* Expose container port 80.
* Add label app: nginx for selectors.

**Goal:**  
Run multiple NGINX instances.

**Step 3: Add a Startup Probe**

**Task:**

* Add a **startupProbe** to the NGINX container:
  + Use an httpGet on / with port 80
  + Set failureThreshold high enough (e.g., 30)
  + Set periodSeconds appropriately (e.g., 5)

**Why:**  
startupProbe is useful if the application takes time to start. It delays the activation of livenessProbe and readinessProbe until it succeeds.

**Goal:**  
Ensure container gets enough time to fully boot before other probes kick in.

**Step 4: Add a Liveness Probe**

**Task:**

* Configure livenessProbe:
  + Use httpGet on / and port 80
  + Use appropriate initialDelaySeconds, periodSeconds, and failureThreshold

**Goal:**  
Allow Kubernetes to detect and restart containers that hang or crash.

**Step 5: Add a Readiness Probe**

**Task:**

* Configure readinessProbe:
  + Use httpGet on / and port 80
  + Set delay and period for proper readiness checking

**Goal:**  
Traffic should only be sent to pods that are ready to serve it.

**Step 6: Expose the Deployment via Service**

**Task:**

* Create a Service named nginx-service
* Type: NodePort (for local access) or LoadBalancer (cloud)
* Route port 80 to containerPort 80 using selector app: nginx

**Goal:**  
Allow external access to NGINX pods.

**Step 7: Test Probes**

**🔹 Test Startup Probe:**

* Optionally configure a slow-starting container or add delay to nginx using an initContainer or override command.
* Observe that livenessProbe and readinessProbe are not triggered until startupProbe succeeds.

**🔹 Test Liveness Probe:**

* Break the container (e.g., delete /usr/share/nginx/html/index.html)
* Wait for restart due to failed liveness probe.

**🔹 Test Readiness Probe:**

* Block port or simulate delay in response.
* Observe the pod being marked NotReady but not restarted.