Kubernetes course Practices

Practice 1

Scenario

registry.k8s.io/e2e-test-images/agnhost:2.40 is a lightweight container image that exposes HTTP endpoints used for health checks and testing (/healthz, /readyz, /echo).
Your task is to deploy this application following production-ready best practices.

Requirements

1. **Application Configuration**
   - The application image and container name (agnhost) should be **set**
     - **Startup probe**
     - **Readiness probe**
     - **Liveness probe**

   - Container port 8080 with name http should be set.

   - 2 **environment variables** for the container must be set to configure:
     - Listening **port with the key PORT and value 8080**
     - A custom **message with the key MESSAGE and value "Hello from agnhost"**
2. **Probes**
   - Configure a **startup probe** to handle an initialization period of about **30 seconds** before the app becomes ready based on the /healthz endpoint.
   - Add a **readiness probe** so the pod only receives traffic when the application is ready based on the /readyz endpoint.
   - Add a **liveness probe** to automatically restart the pod if it becomes unresponsive based on the /healthz endpoint.
   - Also, carefully define initialDealySeconds, timeoutSeconds, periodSeconds, failureThreshold, and successThreshold in all probe sections.
3. **Resources**

- Define **resource requests and limits** for both CPU and memory to ensure fair scheduling (Memory 256Mi and CPU 500m is enough as resource limits).

4. **Rolling Update Strategy**
   - Configure a **rolling update** strategy that guarantees **zero downtime** during version upgrades.
   - Demonstrate a rolling update by modifying a configuration or environment variable and re-deploying.
5. **Service**
   - Expose the deployment inside the cluster using a **ClusterIP/NodePort Service**.

Deliverables

Your submission must include:

1. A **Deployment YAML** manifest file that meets all requirements.
2. A **Service YAML** manifest exposing the application.
3. Deployment, pod, service statuses, logs and describes
4. Execute the below curl commands and investigate outputs and container restarts.
5. A short report (1–2 paragraphs) explaining:
   - What each probe does and why it's configured that way.
   - How the rolling update mechanism avoids downtime.
   - How you confirmed the probes are working correctly (commands, outputs, or screenshots).

Practice 2

You've been asked to deploy the below Example: PHP Guestbook application with Redis on a Kubernetes cluster and ensure that the components communicate correctly using **ClusterIP Services**. Please, read the below document carefully and do what it says to deploy PHP Guestbook application with Redis.

https://kubernetes.io/docs/tutorials/stateless-application/guestbook/

Deliverables

Submit the following:

1. The complete YAML manifests for redis and php frontend.
2. A short summary (one page or less) describing:

    - ✓ How the frontend and backend discover each other.
    - ✓ How scaling works for the frontend pods.
    - ✓ What happens during a rolling update?
    - ✓ Evidence that data persistence works as expected.

Bonus Challenges (Optional)

1. Add a **readiness probe** to the Redis Deployment that checks if the service is responding on port 6379.
2. Add **resource requests and limits** to both Deployments.