Configure health probes for pods

Health probes in Kubernetes are used to check the health of containers running in pods. There are three types:

1. Liveness Probe: Checks if the container is running. If it fails, the container is restarted.

2. Readiness Probe: Checks if the container is ready to serve traffic. If it fails, the pod is removed from service endpoints.

3. Startup Probe: Checks if the application within the container has started. Useful for slow-starting apps.

Configuring Health Probes

pod-with-probes.yaml

```
apiVersion: v1
kind: Pod
metadata:
  name: health-probe-demo
spec:
  containers:
    - name: myapp
      image: nginx
      ports:
        - containerPort: 80
      livenessProbe:
        httpGet:
          path: /
          port: 80
        initialDelaySeconds: 10
        periodSeconds: 5
      readinessProbe:
        httpGet:
          path: /
          port: 80
        initialDelaySeconds: 5
        periodSeconds: 5
      startupProbe:
```

```
      httpGet:
        path: /
        port: 80
      failureThreshold: 30
      periodSeconds: 10
```

Probe Types and Options

We can use three types of actions for probes:

- httpGet: Performs an HTTP GET request.
- exec: Runs a command inside the container.
- tcpSocket: Checks if a TCP socket is open.

httpGet Example
yaml

```
livenessProbe:
  httpGet:
    path: /healthz
    port: 8080
  initialDelaySeconds: 15
  periodSeconds: 20
```

exec Example
yaml

```
livenessProbe:
  exec:
    command:
      - cat
      - /tmp/healthy
  initialDelaySeconds: 5
  periodSeconds: 5
```

tcpSocket Example
yaml

```
livenessProbe:
  tcpSocket:
    port: 3306
  initialDelaySeconds: 15
  periodSeconds: 20
```

Apply

```
$   kubectl apply -f pod-with-probes.yaml
```