Kubernetes is often able to detect problems with containers and respond appropriately without the need for specialized configuration. But sometimes we need additional control over how Kubernetes determines container status. Kubernetes probes provide the ability to customize how Kubernetes detects the status of containers, allowing us to build more sophisticated mechanisms for managing container health. In this lesson, we discuss liveness and readiness probes in Kubernetes, and demonstrate how to create and configure them.

Relevant Documentation

- https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle/#container-probes
- https://kubernetes.io/docs/tasks/configure-pod-container/configure-liveness-readiness-probes/

Lesson Reference

Here is a pod with a liveness probe that uses a command:

`my-liveness-pod.yml`:

```
apiVersion: v1
kind: Pod
metadata:
  name: my-liveness-pod
spec:
  containers:
  - name: myapp-container
    image: busybox
    command: ['sh', '-c', "echo Hello, Kubernetes! && sleep 3600"]
    livenessProbe:
      exec:
        command:
        - echo
        - testing
      initialDelaySeconds: 5
      periodSeconds: 5
```

Here is a pod with a readiness probe that uses an http request:

`my-readiness-pod.yml`:

```
apiVersion: v1
kind: Pod
metadata:
  name: my-readiness-pod
spec:
  containers:
  - name: myapp-container
    image: nginx
    readinessProbe:
      httpGet:
        path: /
        port: 80
      initialDelaySeconds: 5
      periodSeconds: 5
```