1. Define a new Pod named web-server with the image nginx in a YAML manifest. Expose the container port 80. Do not create the Pod yet.

k run web-server --image=nginx:1.19.0 --restart=Never --dry-run=client

--port=80 -o yaml > web-server.yaml

1. For the container, declare a startup probe of type httpGet. Verify that the root context endpoint can be called. Use the default configuration for the probe.
2. For the container, declare a readiness probe of type httpGet. Verify that the root context endpoint can be called. Wait five seconds before checking for the first time.
3. For the container, declare a liveness probe of type httpGet. Verify that the root context endpoint can be called. Wait 10 seconds before checking for the first time. The probe should run the check every 30 seconds.

|  |
| --- |
| apiVersion: **v1**  kind: **Pod**  metadata:  creationTimestamp: **null**  labels:  run: **web-server**  name: **web-server**  spec:  containers:  - image: **nginx:1.19.0**  name: **web-server**  ports:  - containerPort: **80**  resources: {}  startupProbe:  httpGet:  path: **/**  port: **80**  **## below striked out config is not need, if asked for default config.**  ~~failureThreshold:~~ **~~3~~**  ~~initialDelaySeconds:~~  ~~periodSeconds:~~ **~~10~~**  ~~successThreshold:~~ **~~1~~**  ~~timeoutSeconds:~~ **~~1~~**  readinessProbe:  httpGet:  path: **/**  port: **80**  initialDelaySeconds: **5**  livenessProbe:  httpGet:  path: **/**  port: **80**  initialDelaySeconds: **10**  periodSeconds: **30**  dnsPolicy: **ClusterFirst**  restartPolicy: **Never**  status: {} |

1. Inspect the runtime details of the probes of the Pod.

k describe pod web-server

1. Retrieve the metrics of the Pod (e.g., CPU and memory) from the metrics server.

k top pod web-server

1. Create a Pod named custom-cmd with the image busybox. The container should run the command top-analyzer with the command-line flag –all

k run custom-cmd --image=busybox:latest

--restart=Never -- /bin/sh -c “top-analyzer –all”

1. Inspect the status. How would you further troubleshoot the Pod to identify the root cause of the failure?

K logs custom-cmd