HEALTH PROBES IN KUBERNETES

Kubernetes health probes are used to monitor the health and readiness of applications running inside containers. These probes allow Kubernetes to determine whether an application is functioning correctly and ready to handle traffic.

There are three types of Kubernetes health probes:

1. Liveness Probe

- **Purpose**: Checks whether the application is alive and running. If the liveness probe fails, Kubernetes will kill the container and, based on the restart policy, restart it.
- **Common Use Case**: Detect when an application has become deadlocked or is in an unrecoverable state.

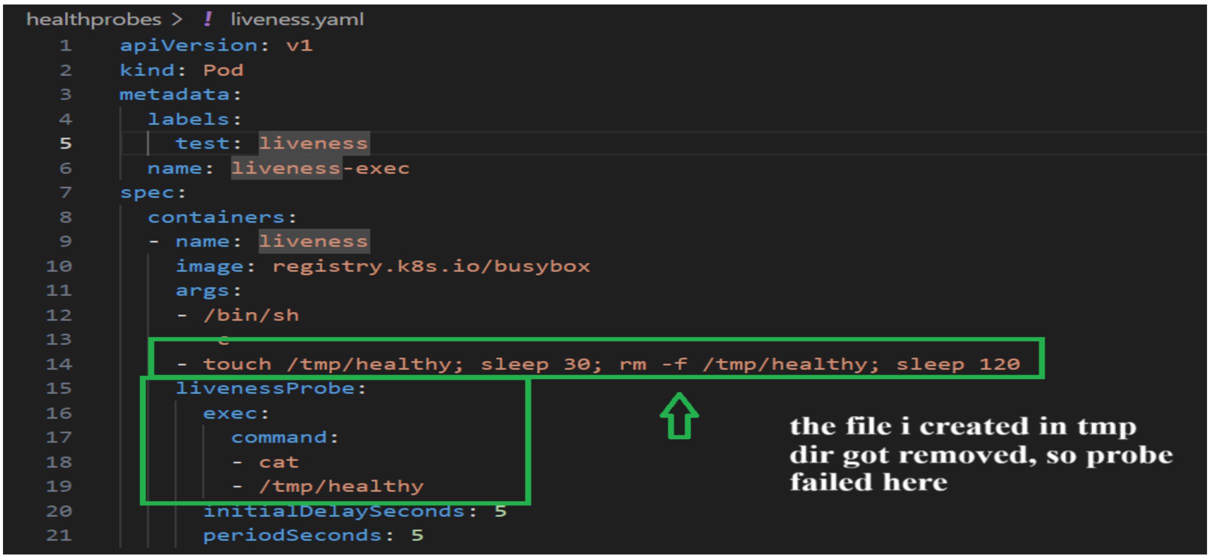
2. Readiness Probe

- **Purpose**: Determines if the application is ready to serve traffic. Now let's say there is an application which take 30sec to bootup and start completely till that user will get "ERROR" can't access the application. To avoid this readiness makes sure that your application only expose to user only when it fully added to loadbalancer once the readiness probe are passed. If the readiness probe fails, Kubernetes will remove the Pod from the list of endpoints for the Service until it passes.
- **Common Use Case**: Delay traffic to the application until it's fully initialized and ready to accept requests.

3. Startup Probe

- **Purpose**: Ensures that the application has started successfully. If the startup probe fails, Kubernetes will restart the container. Once the startup probe passes, Kubernetes stops performing liveness or readiness checks.
- **Common Use Case**: Useful for applications with slow startup times, to prevent them from being prematurely killed or marked as unready by the liveness or readiness probes.

LIVNESS PROBE case: The container starts and immediately creates the file /tmp/healthy. After 5 seconds (as per initialDelaySeconds), Kubernetes runs the cat /tmp/healthy command to check if the file exists. The probe will succeed while the file exists.After 30 seconds, the file /tmp/healthy is deletedOnce the file is deleted, the next liveness probe (which runs every 5 seconds) will fail because the cat /tmp/healthy command will return an error. Once the probe fails, Kubernetes will consider the container unhealthy and will restart it.



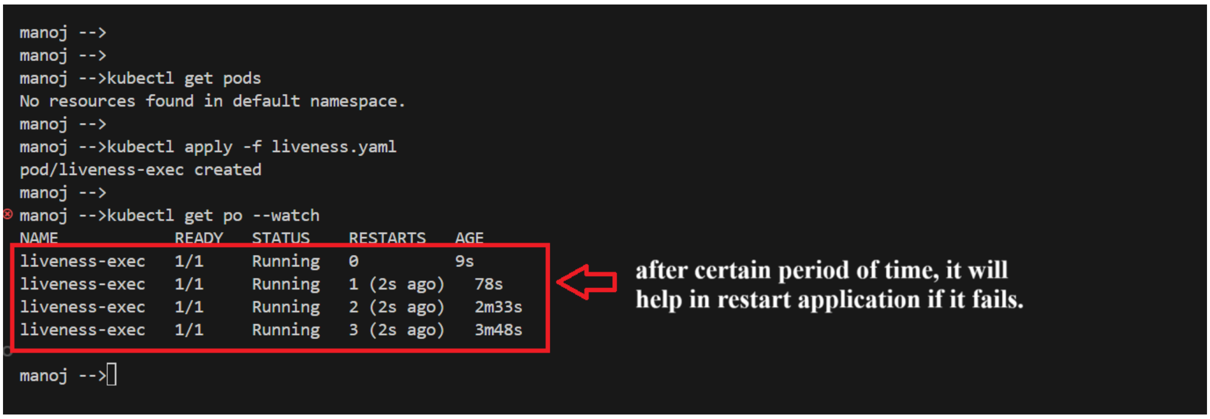
healthprobes > ! liveness.yaml

```yaml
apiVersion: v1
kind: Pod
metadata:
  labels:
    test: liveness
  name: liveness-exec
spec:
  containers:
  - name: liveness
    image: registry.k8s.io/busybox
    args:
    - /bin/sh
    - -c
    - touch /tmp/healthy; sleep 30; rm -f /tmp/healthy; sleep 120
    livenessProbe:
      exec:
        command:
        - cat
        - /tmp/healthy
      initialDelaySeconds: 5
      periodSeconds: 5
```

the file i created in tmp dir got removed, so probe failed here

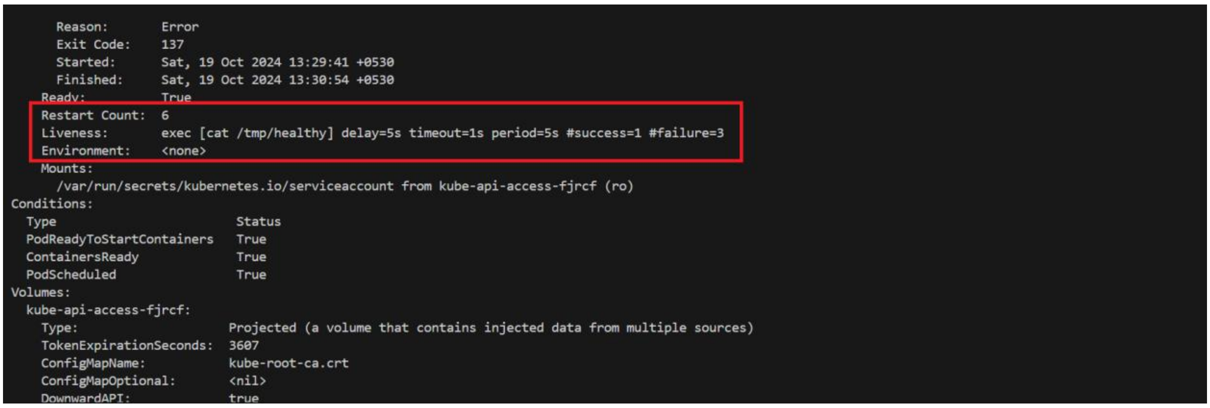
HEALTH PROBES IN KUBERNETES

Pod is getting restarted again and again



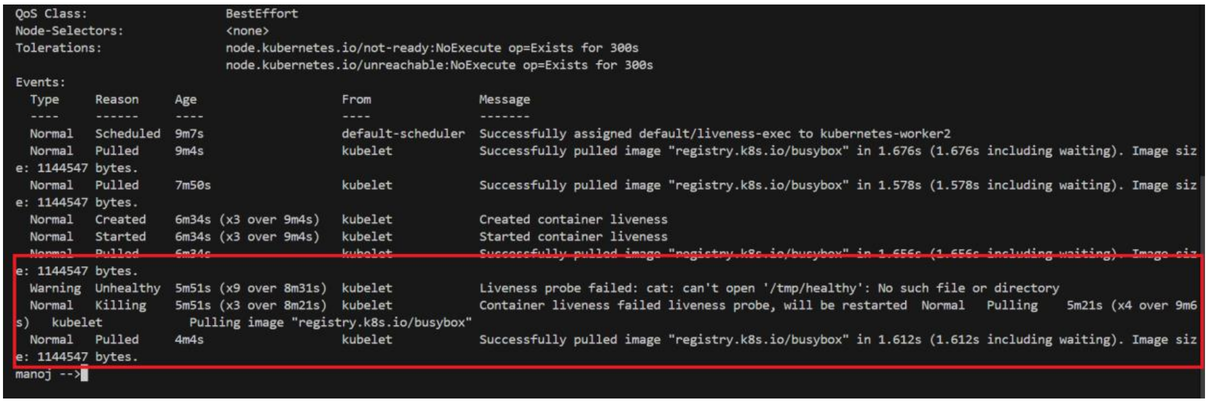
```
manoj -->
manoj -->
manoj -->kubectl get pods
No resources found in default namespace.
manoj -->
manoj -->kubectl apply -f liveness.yaml
pod/liveness-exec created
manoj -->
manoj -->kubectl get po --watch
NAME            READY   STATUS    RESTARTS      AGE
liveness-exec   1/1     Running   0             9s
liveness-exec   1/1     Running   1 (2s ago)    78s
liveness-exec   1/1     Running   2 (2s ago)    2m33s
liveness-exec   1/1     Running   3 (2s ago)    3m48s

manoj -->
```

after certain period of time, it will help in restart application if it fails.



```
      Reason:       Error
      Exit Code:    137
      Started:      Sat, 19 Oct 2024 13:29:41 +0530
      Finished:     Sat, 19 Oct 2024 13:30:54 +0530
    Ready:          True
    Restart Count:  6
    Liveness:       exec [cat /tmp/healthy] delay=5s timeout=1s period=5s #success=1 #failure=3
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-fjrcf (ro)
Conditions:
  Type                        Status
  PodReadyToStartContainers   True
  ContainersReady             True
  PodScheduled                True
Volumes:
  kube-api-access-fjrcf:
    Type:                    Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds:  3607
    ConfigMapName:           kube-root-ca.crt
    ConfigMapOptional:       <nil>
    DownwardAPI:             true
```

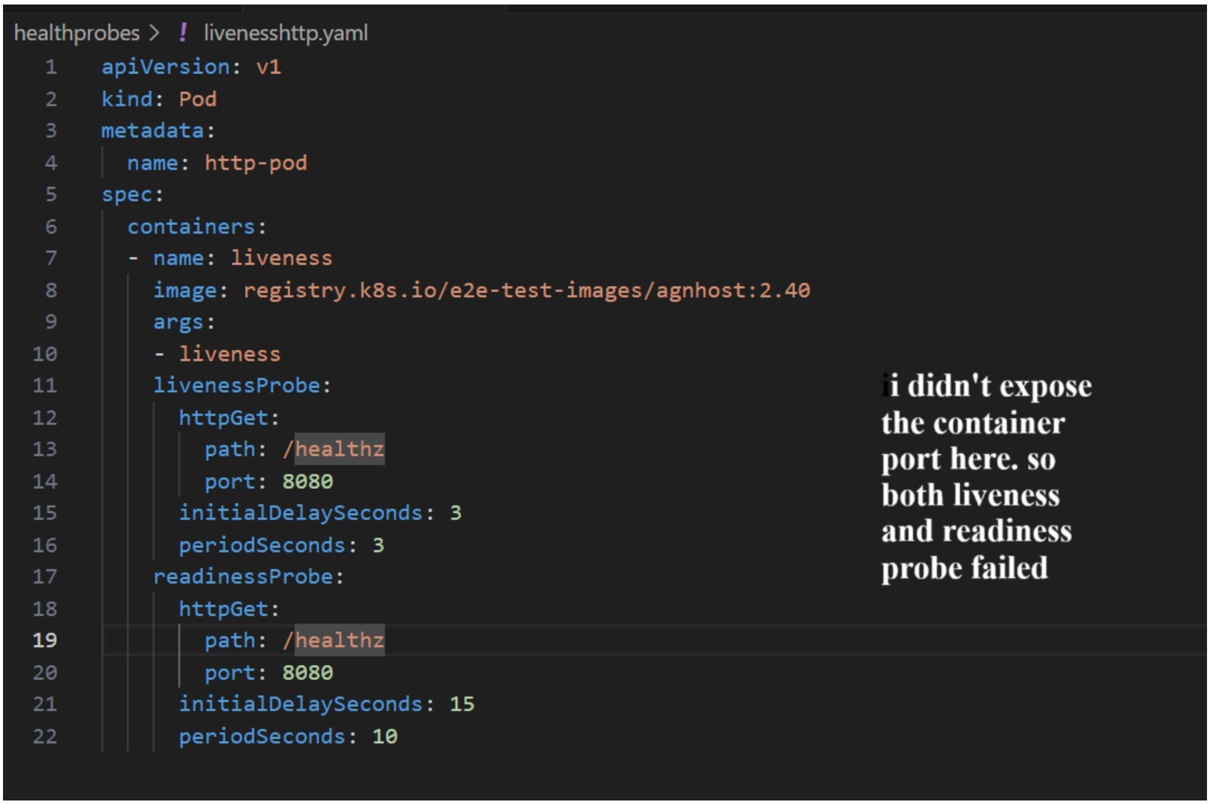
We can see that liveness probe failed because “tmp/healthy” directory can’t be found.



```
QoS Class:                   BestEffort
Node-Selectors:              <none>
Tolerations:                 node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                             node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type     Reason     Age                  From               Message
  ----     ------     ----                 ----               -------
  Normal   Scheduled  9m7s                 default-scheduler  Successfully assigned default/liveness-exec to kubernetes-worker2
  Normal   Pulled     9m4s                 kubelet            Successfully pulled image "registry.k8s.io/busybox" in 1.676s (1.676s including waiting). Image siz
e: 1144547 bytes.
  Normal   Pulled     7m50s                kubelet            Successfully pulled image "registry.k8s.io/busybox" in 1.578s (1.578s including waiting). Image siz
e: 1144547 bytes.
  Normal   Created    6m34s (x3 over 9m4s)   kubelet            Created container liveness
  Normal   Started    6m34s (x3 over 9m4s)   kubelet            Started container liveness
  Normal   Pulled     6m34s                kubelet            Successfully pulled image "registry.k8s.io/busybox" in 1.656s (1.656s including waiting). Image siz
e: 1144547 bytes.
  Warning  Unhealthy  5m51s (x9 over 8m31s)  kubelet            Liveness probe failed: cat: can't open '/tmp/healthy': No such file or directory
  Normal   Killing    5m51s (x3 over 8m21s)  kubelet            Container liveness failed liveness probe, will be restarted  Normal   Pulling    5m21s (x4 over 9m6
s)   kubelet            Pulling image "registry.k8s.io/busybox"
  Normal   Pulled     4m4s                 kubelet            Successfully pulled image "registry.k8s.io/busybox" in 1.612s (1.612s including waiting). Image siz
e: 1144547 bytes.
manoj -->
```

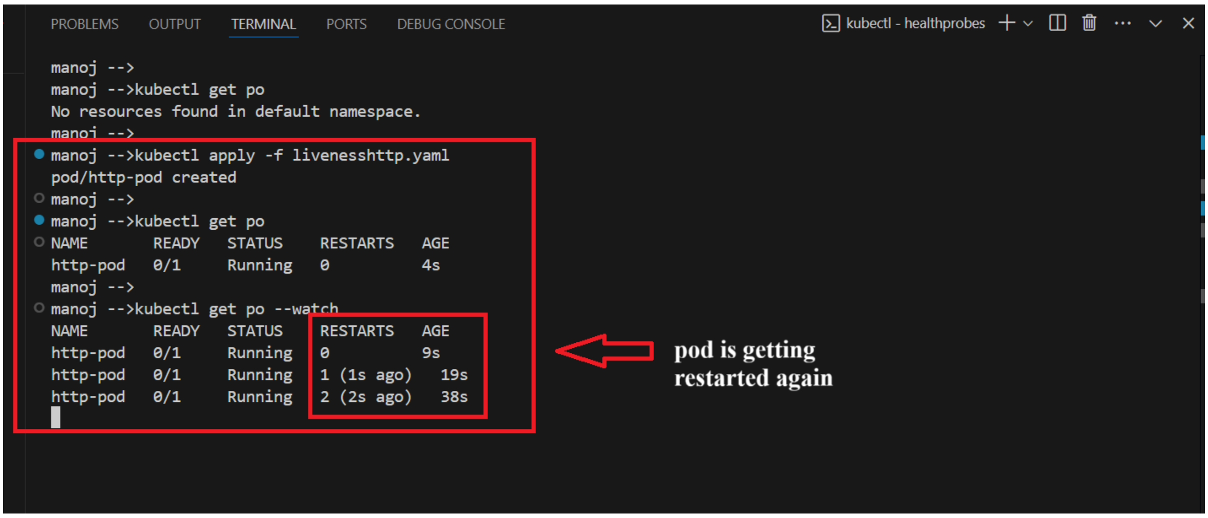
HEALTH PROBES IN KUBERNETES

LIVENESS PROBE AND READINESS PROBE case: In this I didn’t exposed the container port



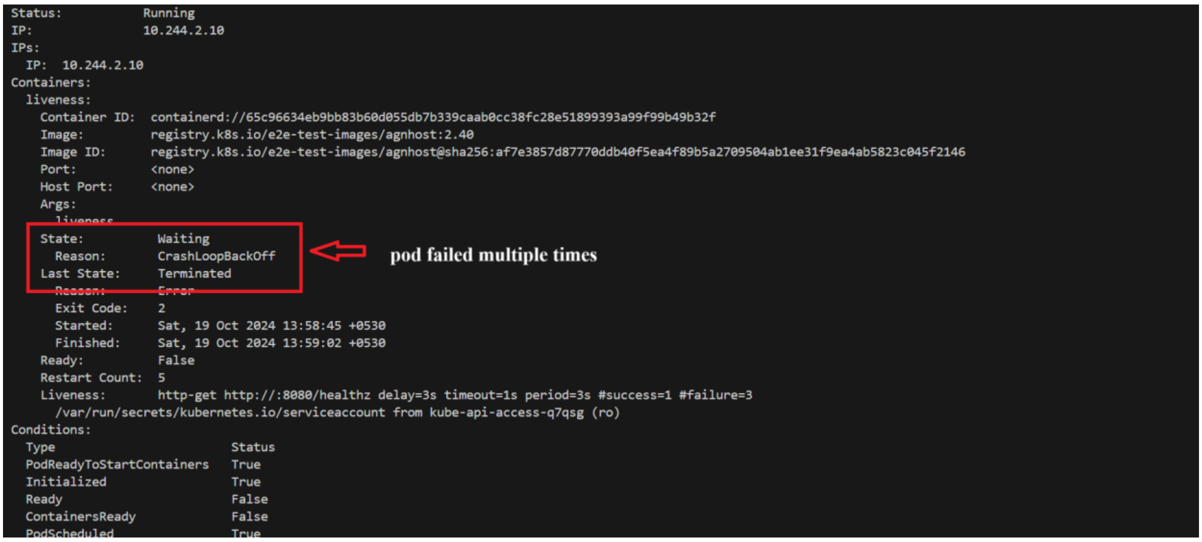
```yaml
apiVersion: v1
kind: Pod
metadata:
  name: http-pod
spec:
  containers:
  - name: liveness
    image: registry.k8s.io/e2e-test-images/agnhost:2.40
    args:
    - liveness
    livenessProbe:
      httpGet:
        path: /healthz
        port: 8080
      initialDelaySeconds: 3
      periodSeconds: 3
    readinessProbe:
      httpGet:
        path: /healthz
        port: 8080
      initialDelaySeconds: 15
      periodSeconds: 10
```

i didn't expose the container port here. so both liveness and readiness probe failed



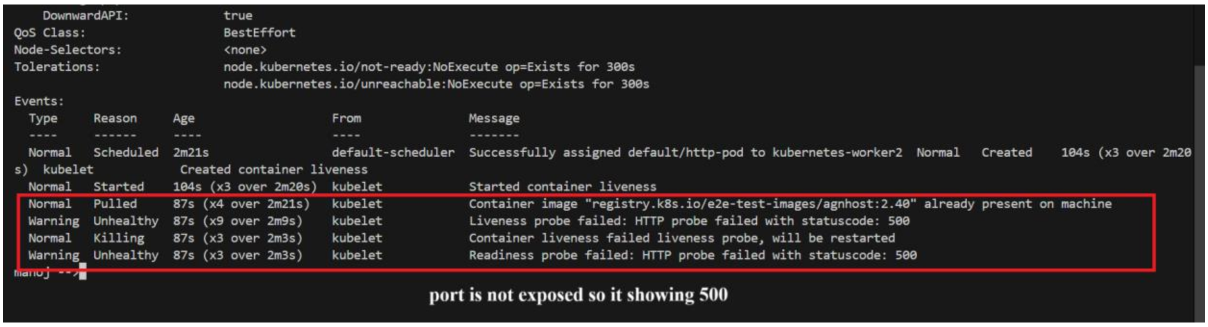
```
manoj -->
manoj -->kubectl get po
No resources found in default namespace.
manoj -->
manoj -->kubectl apply -f livenesshttp.yaml
pod/http-pod created
manoj -->
manoj -->kubectl get po
NAME       READY   STATUS    RESTARTS   AGE
http-pod   0/1     Running   0          4s
manoj -->
manoj -->kubectl get po --watch
NAME       READY   STATUS    RESTARTS     AGE
http-pod   0/1     Running   0            9s
http-pod   0/1     Running   1 (1s ago)   19s
http-pod   0/1     Running   2 (2s ago)   38s
```

pod is getting restarted again



```
Status:           Running
IP:               10.244.2.10
IPs:
  IP:  10.244.2.10
Containers:
  liveness:
    Container ID:  containerd://65c96634eb9bb83b60d055db7b339caab0cc38fc28e51899393a99f99b49b32f
    Image:         registry.k8s.io/e2e-test-images/agnhost:2.40
    Image ID:      registry.k8s.io/e2e-test-images/agnhost@sha256:af7e3857d87770ddb40f5ea4f89b5a2709504ab1ee31f9ea4ab5823c045f2146
    Port:          <none>
    Host Port:     <none>
    Args:
      liveness
    State:          Waiting
      Reason:       CrashLoopBackOff
    Last State:     Terminated
      Reason:       Error
      Exit Code:    2
      Started:      Sat, 19 Oct 2024 13:58:45 +0530
      Finished:     Sat, 19 Oct 2024 13:59:02 +0530
    Ready:          False
    Restart Count:  5
    Liveness:       http-get http://:8080/healthz delay=3s timeout=1s period=3s #success=1 #failure=3
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-q7qsg (ro)
Conditions:
  Type                        Status
  PodReadyToStartContainers   True
  Initialized                 True
  Ready                       False
  ContainersReady             False
  PodScheduled                True
```

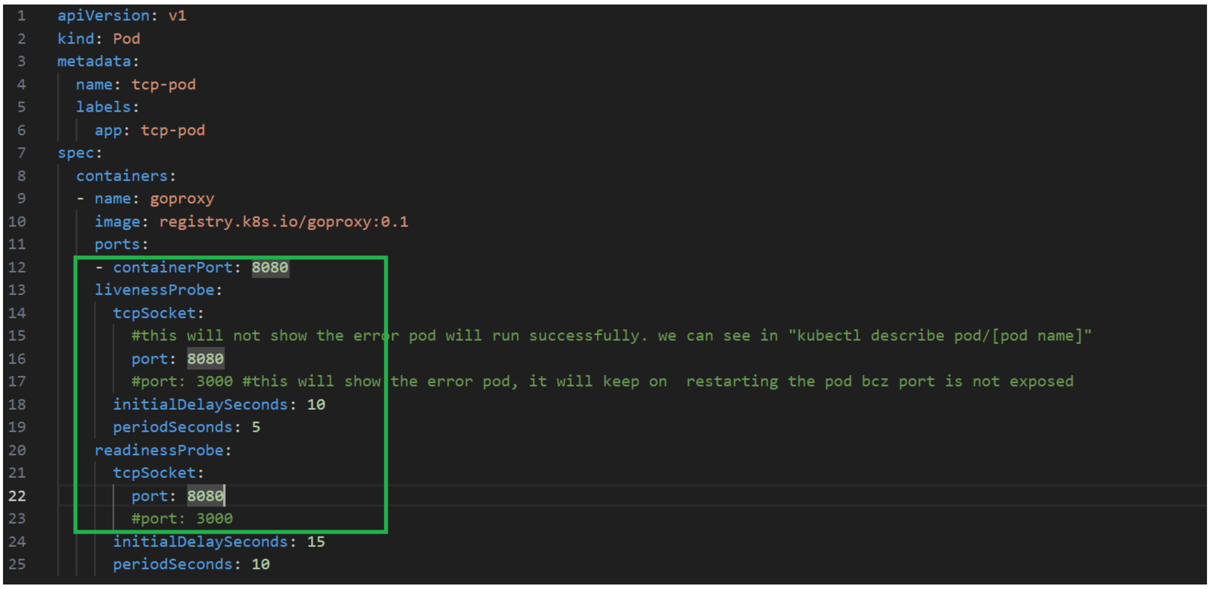
pod failed multiple times



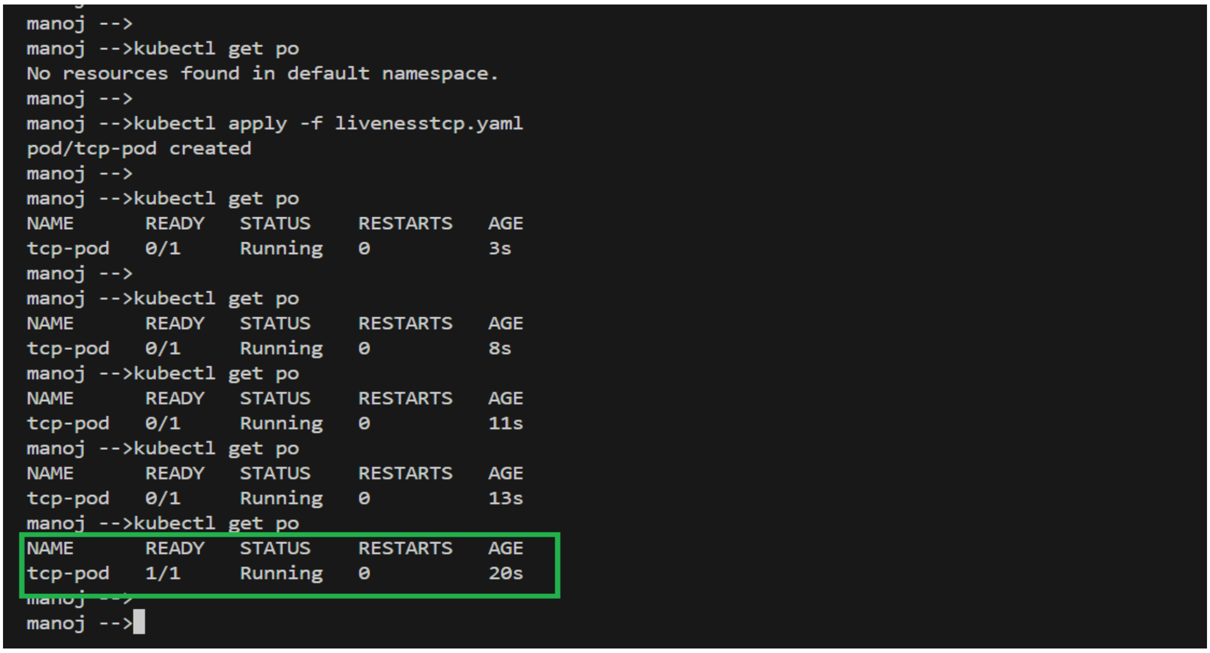
```
    DownwardAPI:             true
QoS Class:                   BestEffort
Node-Selectors:              <none>
Tolerations:                 node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                             node.kubernetes.io/unreachable:NoExecute op=Exists for 300s

Events:
  Type     Reason     Age                   From               Message
  ----     ------     ----                  ----               -------
  Normal   Scheduled  2m21s                 default-scheduler  Successfully assigned default/http-pod to kubernetes-worker2  Normal   Created    104s (x3 over 2m20
s)  kubelet            Created container liveness
  Normal   Started    104s (x3 over 2m20s)  kubelet            Started container liveness
  Normal   Pulled     87s (x4 over 2m21s)   kubelet            Container image "registry.k8s.io/e2e-test-images/agnhost:2.40" already present on machine
  Warning  Unhealthy  87s (x9 over 2m9s)    kubelet            Liveness probe failed: HTTP probe failed with statuscode: 500
  Normal   Killing    87s (x3 over 2m3s)    kubelet            Container liveness failed liveness probe, will be restarted
  Warning  Unhealthy  87s (x3 over 2m3s)    kubelet            Readiness probe failed: HTTP probe failed with statuscode: 500
manoj -->
```

port is not exposed so it showing 500

Now I exposed the port:

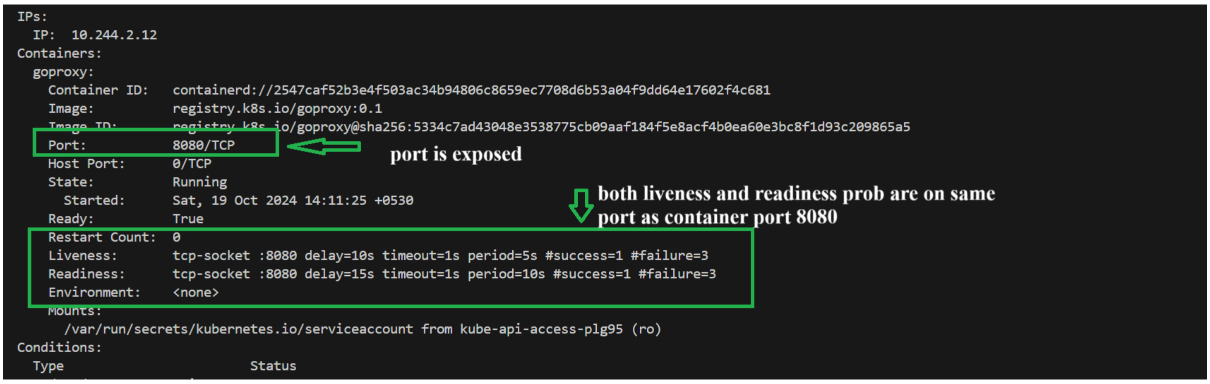


```yaml
apiVersion: v1
kind: Pod
metadata:
  name: tcp-pod
  labels:
    app: tcp-pod
spec:
  containers:
  - name: goproxy
    image: registry.k8s.io/goproxy:0.1
    ports:
    - containerPort: 8080
    livenessProbe:
      tcpSocket:
        #this will not show the error pod will run successfully. we can see in "kubectl describe pod/[pod name]"
        port: 8080
        #port: 3000 #this will show the error pod, it will keep on  restarting the pod bcz port is not exposed
      initialDelaySeconds: 10
      periodSeconds: 5
    readinessProbe:
      tcpSocket:
        port: 8080
        #port: 3000
      initialDelaySeconds: 15
      periodSeconds: 10
```



```
manoj -->
manoj -->kubectl get po
No resources found in default namespace.
manoj -->
manoj -->kubectl apply -f livenesstcp.yaml
pod/tcp-pod created
manoj -->
manoj -->kubectl get po
NAME      READY   STATUS    RESTARTS   AGE
tcp-pod   0/1     Running   0          3s
manoj -->
manoj -->kubectl get po
NAME      READY   STATUS    RESTARTS   AGE
tcp-pod   0/1     Running   0          8s
manoj -->kubectl get po
NAME      READY   STATUS    RESTARTS   AGE
tcp-pod   0/1     Running   0          11s
manoj -->kubectl get po
NAME      READY   STATUS    RESTARTS   AGE
tcp-pod   0/1     Running   0          13s
manoj -->kubectl get po
NAME      READY   STATUS    RESTARTS   AGE
tcp-pod   1/1     Running   0          20s
manoj -->
manoj -->
```

We can see both probes are passed successfully

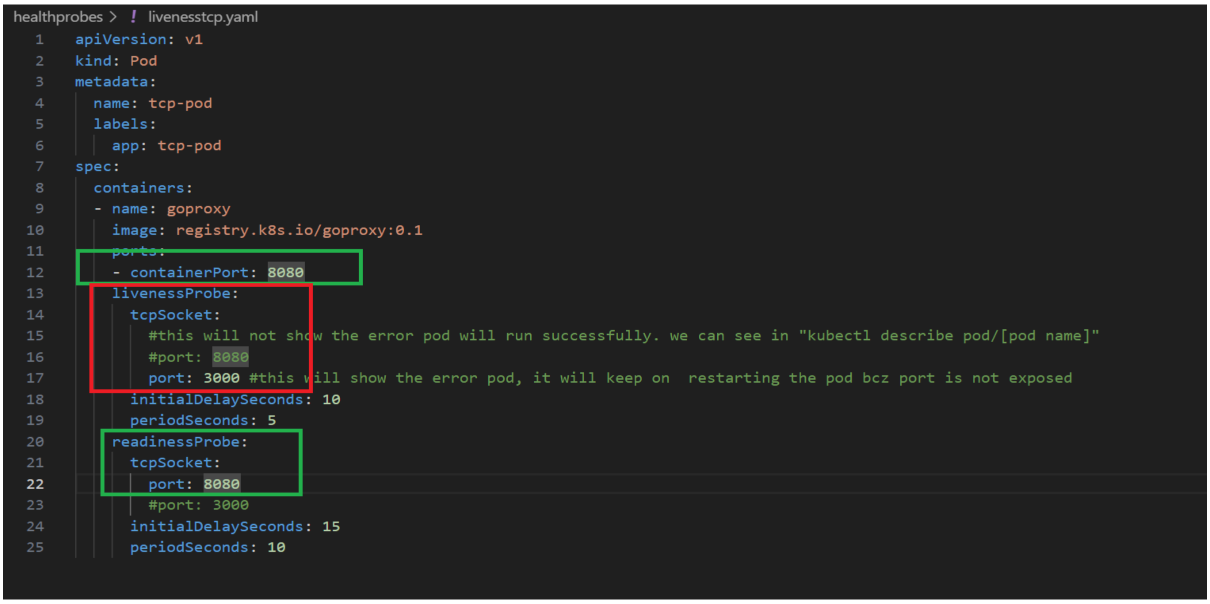


```
IPs:
  IP:  10.244.2.12
Containers:
  goproxy:
    Container ID:   containerd://2547caf52b3e4f503ac34b94806c8659ec7708d6b53a04f9dd64e17602f4c681
    Image:          registry.k8s.io/goproxy:0.1
    Image ID:       registry.k8s.io/goproxy@sha256:5334c7ad43048e3538775cb09aaf184f5e8acf4b0ea60e3bc8f1d93c209865a5
    Port:           8080/TCP
    Host Port:      0/TCP
    State:          Running
      Started:      Sat, 19 Oct 2024 14:11:25 +0530
    Ready:          True
    Restart Count:  0
    Liveness:       tcp-socket :8080 delay=10s timeout=1s period=5s #success=1 #failure=3
    Readiness:      tcp-socket :8080 delay=15s timeout=1s period=10s #success=1 #failure=3
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-plg95 (ro)
Conditions:
  Type              Status
```

port is exposed

both liveness and readiness prob are on same port as container port 8080

Now I’m exposing only readiness probe port but not the liveness probe port

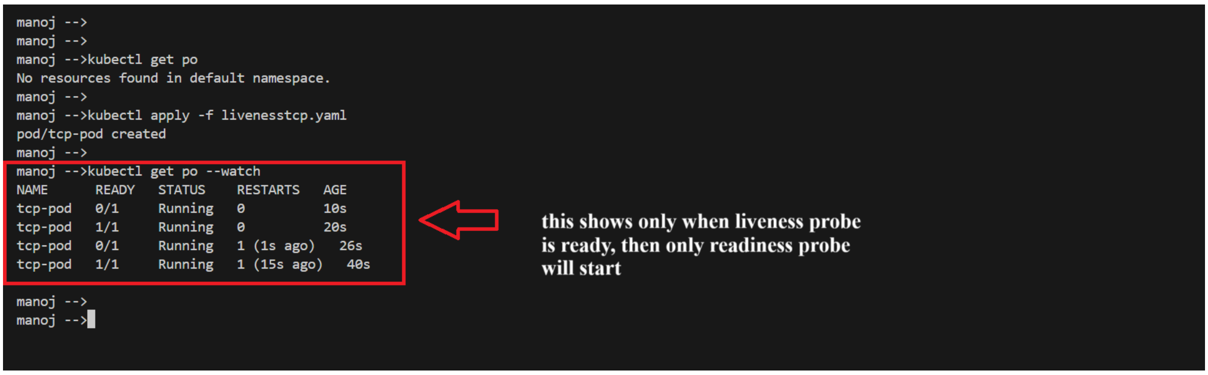


healthprobes > ! livenesstcp.yaml

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: tcp-pod
  labels:
    app: tcp-pod
spec:
  containers:
  - name: goproxy
    image: registry.k8s.io/goproxy:0.1
    ports:
    - containerPort: 8080
    livenessProbe:
      tcpSocket:
        #this will not show the error pod will run successfully. we can see in "kubectl describe pod/[pod name]"
        #port: 8080
        port: 3000 #this will show the error pod, it will keep on  restarting the pod bcz port is not exposed
      initialDelaySeconds: 10
      periodSeconds: 5
    readinessProbe:
      tcpSocket:
        port: 8080
        #port: 3000
      initialDelaySeconds: 15
      periodSeconds: 10
```

HEALTH PROBES IN KUBERNETES

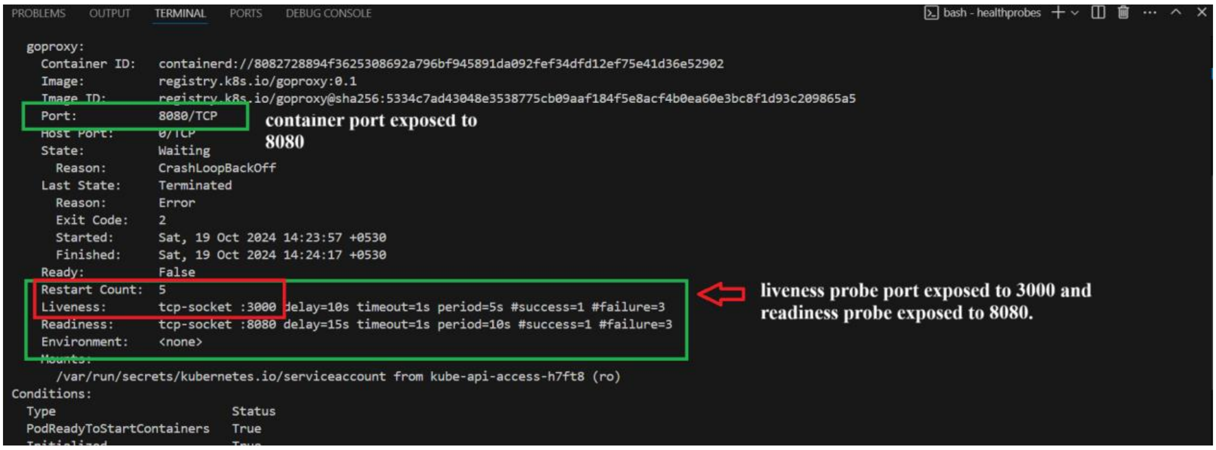
Because liveness probe failed application is restarting again and again.



```
manoj -->
manoj -->
manoj -->kubectl get po
No resources found in default namespace.
manoj -->
manoj -->kubectl apply -f livenesstcp.yaml
pod/tcp-pod created
manoj -->
manoj -->kubectl get po --watch
NAME      READY   STATUS    RESTARTS      AGE
tcp-pod   0/1     Running   0             10s
tcp-pod   1/1     Running   0             20s
tcp-pod   0/1     Running   1 (1s ago)    26s
tcp-pod   1/1     Running   1 (15s ago)   40s

manoj -->
manoj -->
```

this shows only when liveness probe is ready, then only readiness probe will start

We can see container port is exposed to 8080 but liveness probe is exposed on 3000

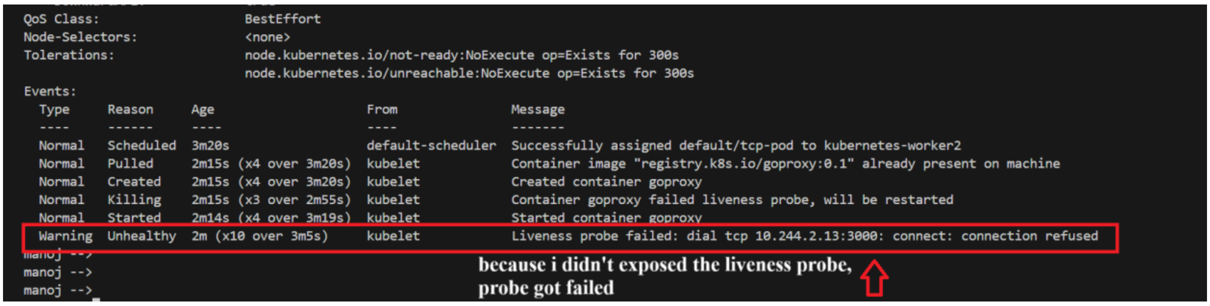


```
  goproxy:
    Container ID:   containerd://8082728894f3625308692a796bf945891da092fef34dfd12ef75e41d36e52902
    Image:          registry.k8s.io/goproxy:0.1
    Image ID:       registry.k8s.io/goproxy@sha256:5334c7ad43048e3538775cb09aaf184f5e8acf4b0ea60e3bc8f1d93c209865a5
    Port:           8080/TCP
    Host Port:      0/TCP
    State:          Waiting
      Reason:       CrashLoopBackOff
    Last State:     Terminated
      Reason:       Error
      Exit Code:    2
      Started:      Sat, 19 Oct 2024 14:23:57 +0530
      Finished:     Sat, 19 Oct 2024 14:24:17 +0530
    Ready:          False
    Restart Count:  5
    Liveness:       tcp-socket :3000 delay=10s timeout=1s period=5s #success=1 #failure=3
    Readiness:      tcp-socket :8080 delay=15s timeout=1s period=10s #success=1 #failure=3
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-h7ft8 (ro)
Conditions:
  Type                        Status
  PodReadyToStartContainers   True
```

container port exposed to 8080

liveness probe port exposed to 3000 and readiness probe exposed to 8080.

We can see the ERROR

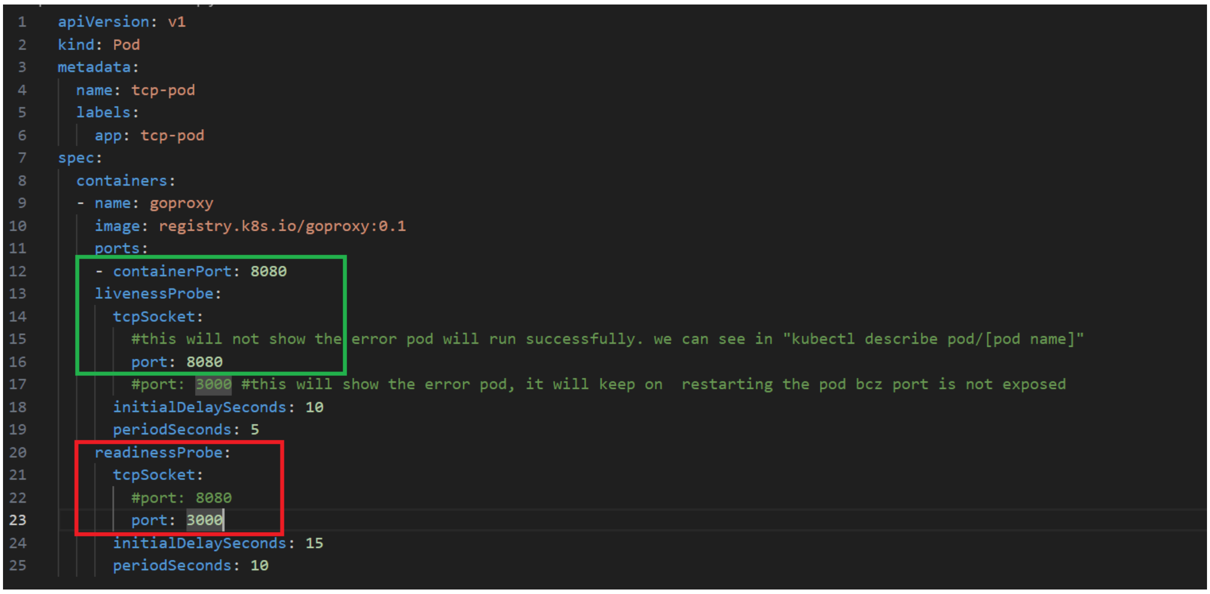


```
  QoS Class:                   BestEffort
  Node-Selectors:              <none>
  Tolerations:                 node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                               node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
  Events:
    Type     Reason     Age                   From               Message
    ----     ------     ----                  ----               -------
    Normal   Scheduled  3m20s                 default-scheduler  Successfully assigned default/tcp-pod to kubernetes-worker2
    Normal   Pulled     2m15s (x4 over 3m20s) kubelet            Container image "registry.k8s.io/goproxy:0.1" already present on machine
    Normal   Created    2m15s (x4 over 3m20s) kubelet            Created container goproxy
    Normal   Killing    2m15s (x3 over 2m55s) kubelet            Container goproxy failed liveness probe, will be restarted
    Normal   Started    2m14s (x4 over 3m19s) kubelet            Started container goproxy
    Warning  Unhealthy  2m (x10 over 3m5s)    kubelet            Liveness probe failed: dial tcp 10.244.2.13:3000: connect: connection refused
  manoj -->
  manoj -->
  manoj -->
```

because i didn't exposed the liveness probe, probe got failed

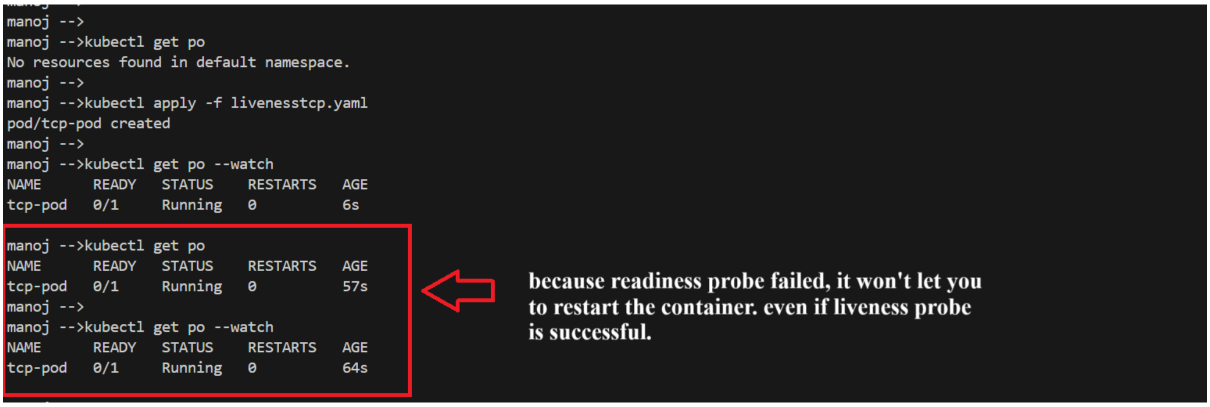
HEALTH PROBES IN KUBERNETES

Now I exposed liveness probe but not the readiness probe



```yaml
apiVersion: v1
kind: Pod
metadata:
  name: tcp-pod
  labels:
    app: tcp-pod
spec:
  containers:
  - name: goproxy
    image: registry.k8s.io/goproxy:0.1
    ports:
    - containerPort: 8080
    livenessProbe:
      tcpSocket:
        #this will not show the error pod will run successfully. we can see in "kubectl describe pod/[pod name]"
        port: 8080
        #port: 3000 #this will show the error pod, it will keep on  restarting the pod bcz port is not exposed
      initialDelaySeconds: 10
      periodSeconds: 5
    readinessProbe:
      tcpSocket:
        #port: 8080
        port: 3000
      initialDelaySeconds: 15
      periodSeconds: 10
```

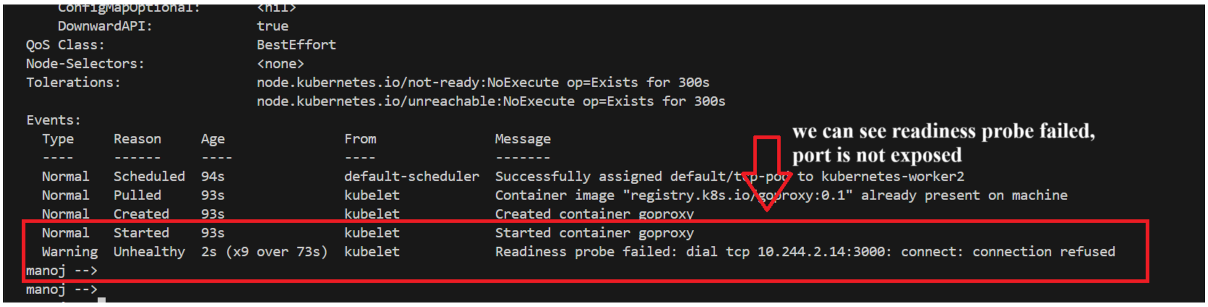
We can see container running successfully but “not READY”



```
manoj -->
manoj -->kubectl get po
No resources found in default namespace.
manoj -->
manoj -->kubectl apply -f livenesstcp.yaml
pod/tcp-pod created
manoj -->
manoj -->kubectl get po --watch
NAME      READY   STATUS    RESTARTS   AGE
tcp-pod   0/1     Running   0          6s

manoj -->kubectl get po
NAME      READY   STATUS    RESTARTS   AGE
tcp-pod   0/1     Running   0          57s
manoj -->
manoj -->kubectl get po --watch
NAME      READY   STATUS    RESTARTS   AGE
tcp-pod   0/1     Running   0          64s
```

because readiness probe failed, it won't let you to restart the container. even if liveness probe is successful.

Because readiness probe failed, container was still in “not ready” state.



```
    ConfigMapOptional:       <nil>
    DownwardAPI:             true
QoS Class:                   BestEffort
Node-Selectors:              <none>
Tolerations:                 node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                             node.kubernetes.io/unreachable:NoExecute op=Exists for 300s

Events:
  Type     Reason     Age                From               Message
  ----     ------     ----               ----               -------
  Normal   Scheduled  94s                default-scheduler  Successfully assigned default/tcp-pod to kubernetes-worker2
  Normal   Pulled     93s                kubelet            Container image "registry.k8s.io/goproxy:0.1" already present on machine
  Normal   Created    93s                kubelet            Created container goproxy
  Normal   Started    93s                kubelet            Started container goproxy
  Warning  Unhealthy  2s (x9 over 73s)   kubelet            Readiness probe failed: dial tcp 10.244.2.14:3000: connect: connection refused
manoj -->
manoj -->
```

we can see readiness probe failed, port is not exposed

In-short:

- **Liveness Probes**: Check if the application is alive.
- **Readiness Probes**: Check if the application is ready to serve traffic.
- **Startup Probes**: Ensure that the application has successfully started before other probes run.