Kubernetes_8_Probes

Liveness probe
Readiness probe
Startup probe

Readiness probe: If your application is ready to receive traffic or not. If it fails, Kubernetes stops sending new connections to that particular pod and does not restart the Pod. (When to allow and when to stop the traffic to the Pods)

Liveness probe: If your app is alive and healthy. If it fails, Kubernetes will restart the container. But Readiness will not restart the Pod (When to restart the container when issue is not resolved)

ubuntu@ip-172-31-9-165:~/Probes$ cat readiness-live-manifest.yml

```
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: javawebdeploy
  labels:
    app: javawebapp
spec:
  replicas: 3
  selector:
    matchLabels:
      app: javawebapp
  template:
    metadata:
      name: javawebpod
      labels:
        app: javawebapp
    spec:
      containers:
        - name: javawebapp
          image: hacker123shiva/springbt-in-docker:latest
          ports:
            - containerPort: 8080 # Your Spring Boot app listens on 8080
          readinessProbe:
            initialDelaySeconds: 30
            periodSeconds: 5
            timeoutSeconds: 10
            successThreshold: 1
            failureThreshold: 3
            httpGet:
              path: /
              port: 8080 # Corrected: Probe must check the port your app is listening on
          livenessProbe: # <--- ADD THIS SECTION
            httpGet:
              path: /
              port: 8080
            initialDelaySeconds: 45 # Give the app ample time to fully start before health checks begin
            periodSeconds: 10       # Check every 10 seconds
            timeoutSeconds: 5       # Consider a failure if no response within 5 seconds
            successThreshold: 1
            failureThreshold: 3
---
apiVersion: v1
```

```
kind: Service
metadata:
  name: javawebapp-service # Name for your LoadBalancer Service
  labels:
    app: javawebapp
spec:
  type: LoadBalancer # This exposes your service externally
  selector:
    app: javawebapp # This must match the 'app' label in your Deployment's template
  ports:
    - protocol: TCP
      port: 80 # The port the LoadBalancer will listen on (e.g., standard HTTP port)
      targetPort: 8080 # The port your Spring Boot application is running on inside the pod
      # You can optionally specify a nodePort if needed, e.g., nodePort: 30080
```

...

initialDelaySeconds: 30 --> Initial 30 seconds the Pods will not start
periodSeconds: 5 --> Every 5 seconds Kubernetes will perform the readiness check
Suppose continuously if the pod is not ready then Kubernetes will only come to a conclusion that the pod is not ready and I should not allow traffic to this particular pod. Liveness will recreate the container while also redirecting the traffic.

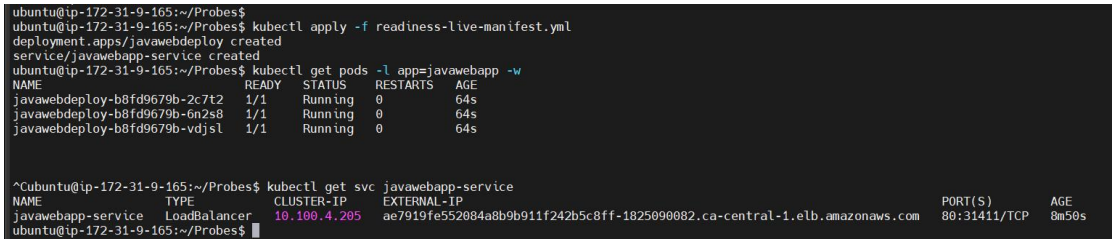
```
ubuntu@ip-172-31-9-165:~/Probes$ kubectl apply -f readiness-live-manifest.yml
deployment.apps/javawebdeploy created
service/javawebapp-service created
ubuntu@ip-172-31-9-165:~/Probes$
ubuntu@ip-172-31-9-165:~/Probes$
ubuntu@ip-172-31-9-165:~/Probes$ kubectl get pods
NAME                         READY   STATUS    RESTARTS   AGE
javawebdeploy-b8fd9679b-6wbdd   0/1     Running   0          40s
javawebdeploy-b8fd9679b-6xdbl   0/1     Running   0          40s
javawebdeploy-b8fd9679b-qht2b   0/1     Running   0          40s
```

```
ubuntu@ip-172-31-9-165:~/Probes$ kubectl delete all --all
pod "javawebdeploy-b8fd9679b-6wbdd" deleted
pod "javawebdeploy-b8fd9679b-6xdbl" deleted
pod "javawebdeploy-b8fd9679b-qht2b" deleted
service "javawebapp-service" deleted
service "kubernetes" deleted
deployment.apps "javawebdeploy" deleted
replicaset.apps "javawebdeploy-b8fd9679b" deleted
```

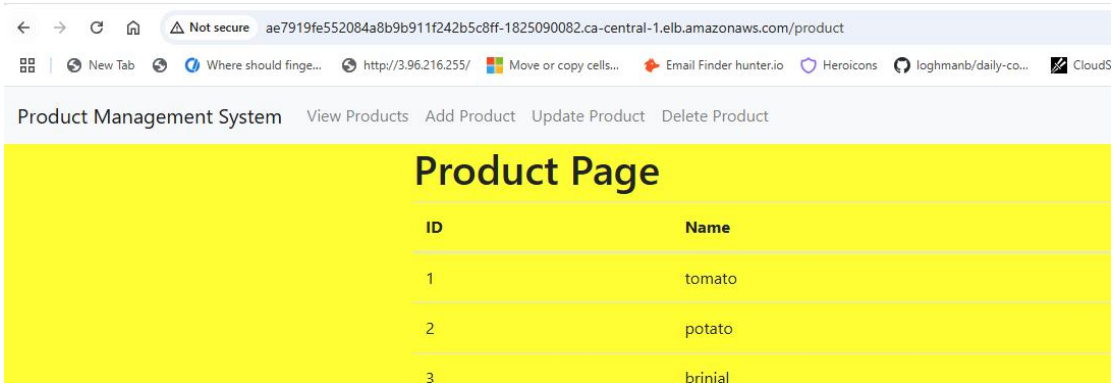
```
ubuntu@ip-172-31-9-165:~/Probes$ kubectl apply -f readiness-live-manifest.yml
deployment.apps/javawebdeploy created
service/javawebapp-service created
ubuntu@ip-172-31-9-165:~/Probes$ kubectl get pods -l app=javawebapp -w
NAME                         READY   STATUS    RESTARTS   AGE
javawebdeploy-b8fd9679b-2c7t2   1/1     Running   0          64s
javawebdeploy-b8fd9679b-6n2s8   1/1     Running   0          64s
javawebdeploy-b8fd9679b-vdjsl   1/1     Running   0          64s
```

```
^Cubuntu@ip-172-31-9-165:~/Probes$ kubectl get svc javawebapp-service
```

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
javawebapp-service LoadBalancer 10.100.4.205 ae7919fe552084a8b9b911f242b5c8ff-1825090082.ca-central-1.elb.amazonaws.com 80:31411/TCP 8m50s



ae7919fe552084a8b9b911f242b5c8ff-1825090082.ca-central-1.elb.amazonaws.com



Application is live without a problem

How to demonstrate Readiness probe?

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
javawebapp-service LoadBalancer 10.100.4.205 ae7919fe552084a8b9b911f242b5c8ff-1825090082.ca-central-1.elb.amazonaws.com 80:31411/TCP 8m50s
ubuntu@ip-172-31-9-165:~/Probes$
ubuntu@ip-172-31-9-165:~/Probes$
ubuntu@ip-172-31-9-165:~/Probes$
ubuntu@ip-172-31-9-165:~/Probes$ kubectl exec -it javawebdeploy-b8fd9679b-vdjsl -- /bin/sh
sh-4.4# kill 1
sh-4.4# command terminated with exit code 137

Pod is restarting
ubuntu@ip-172-31-9-165:~/Probes$ kubectl get pods -l app=javawebapp -w
NAME READY STATUS RESTARTS AGE
javawebdeploy-b8fd9679b-2c7t2 1/1 Running 0 14m
javawebdeploy-b8fd9679b-6n2s8 1/1 Running 0 14m
javawebdeploy-b8fd9679b-vdjsl 1/1 Running 1 (83s ago) 14m

```
^Cubuntu@ip-172-31-9-165:~/Probes$ kubectl get svc javawebapp-service
NAME                 TYPE           CLUSTER-IP      EXTERNAL-IP
javawebapp-service   LoadBalancer   10.100.4.205    ae7919fe552084a8b9b911f242b5c8ff-1825090082.ca-central-1.elb.amazonaws.com
ubuntu@ip-172-31-9-165:~/Probes$
ubuntu@ip-172-31-9-165:~/Probes$
ubuntu@ip-172-31-9-165:~/Probes$
ubuntu@ip-172-31-9-165:~/Probes$ kubectl exec -it javawebdeploy-b8fd9679b-vdjsl -- /bin/sh
sh-4.4# kill 1
sh-4.4# command terminated with exit code 137
ubuntu@ip-172-31-9-165:~/Probes$ kubectl get pods -l app=javawebapp -w
NAME                              READY   STATUS    RESTARTS      AGE
javawebdeploy-b8fd9679b-2c7t2   1/1     Running   0             14m
javawebdeploy-b8fd9679b-6n2s8   1/1     Running   0             14m
javawebdeploy-b8fd9679b-vdjsl   1/1     Running   1 (83s ago)   14m
```

ubuntu@ip-172-31-9-165:~/Probes$ kubectl describe pod javawebdeploy-b8fd9679b-vdjsl

```
ubuntu@ip-172-31-9-165:~/Probes$ kubectl describe pod javawebdeploy-b8fd9679b-vdjsl
Name:             javawebdeploy-b8fd9679b-vdjsl
Namespace:        default
Priority:         0
Service Account:  default
Node:             ip-192-168-27-126.ca-central-1.compute.internal/192.168.27.126
Start Time:       Sun, 15 Jun 2025 15:45:22 +0000
Labels:           app=javawebapp
                  pod-template-hash=b8fd9679b
Annotations:      <none>
Status:           Running
IP:               192.168.12.128
IPs:
  IP:           192.168.12.128
Controlled By:  ReplicaSet/javawebdeploy-b8fd9679b
Containers:
  javawebapp:
    Container ID:   containerd://f012eeba83fdd9ca8fe84863bccaf97a4c551cf77cd24a9a9169d1c7e4c4fa74
    Image:          hacker123shiva/springbt-in-docker:latest
    Image ID:       docker.io/hacker123shiva/springbt-in-docker@sha256:1535b83e22cc9dafe5a031570682e4a818f473d4571d617699d6777a6dfccf40
    Port:           8080/TCP
    Host Port:      0/TCP
    State:          Running
      Started:      Sun, 15 Jun 2025 15:58:18 +0000
    Last State:     Terminated
      Reason:       Error
      Exit Code:    143
      Started:      Sun, 15 Jun 2025 15:45:23 +0000
      Finished:     Sun, 15 Jun 2025 15:58:17 +0000
    Ready:          True
    Restart Count:  1
    Liveness:       http-get http://:8080/ delay=45s timeout=5s period=10s #success=1 #failure=3
    Readiness:      http-get http://:8080/ delay=30s timeout=10s period=5s #success=1 #failure=3
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-76qvq (ro)
Conditions:
  Type                        Status
  PodReadyToStartContainers   True
  Initialized                 True
  Ready                       True
  ContainersReady             True
  PodScheduled                True
Volumes:
  kube-api-access-76qvq:
    Type:                    Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds:  3607
    ConfigMapName:           kube-root-ca.crt
    Optional:                false
    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  21m                  default-scheduler  Successfully assigned default/javawebdeploy-b8fd9679b-vdjs
pute.internal
  Normal  Pulled     21m                  kubelet            Successfully pulled image "hacker123shiva/springbt-in-dock
iting). Image size: 262359412 bytes.
  Normal  Pulling    8m20s (x2 over 21m)  kubelet            Pulling image "hacker123shiva/springbt-in-docker:latest"
  Normal  Created    8m20s (x2 over 21m)  kubelet            Created container: javawebapp
  Normal  Pulled     8m20s                kubelet            Successfully pulled image "hacker123shiva/springbt-in-dock
iting). Image size: 262359412 bytes.
  Normal  Started    8m19s (x2 over 21m)  kubelet            Started container javawebapp
ubuntu@ip-172-31-9-165:~/Probes$
```

```
ubuntu@ip-172-31-9-165:~/Probes$ kubectl describe pod javawebdeploy-b8fd9679b-2c7t2
Name:             javawebdeploy-b8fd9679b-2c7t2
Namespace:        default
Priority:         0
Service Account:  default
Node:             ip-192-168-60-81.ca-central-1.compute.internal/192.168.60.81
Start Time:       Sun, 15 Jun 2025 15:45:22 +0000
Labels:           app=javawebapp
                  pod-template-hash=b8fd9679b
Annotations:      <none>
Status:           Running
IP:               192.168.63.105
IPs:
  IP:           192.168.63.105
Controlled By:  ReplicaSet/javawebdeploy-b8fd9679b
Containers:
  javawebapp:
    Container ID:   containerd://ef790d87dfa727125006293bd726ff34324c158811d7b7d7a1376d254ab9a2ba
    Image:          hacker123shiva/springbt-in-docker:latest
    Image ID:       docker.io/hacker123shiva/springbt-in-docker@sha256:1535b83e22cc9dafe5a031570682e4a81
    Port:           8080/TCP
    Host Port:      0/TCP
    State:          Running
      Started:      Sun, 15 Jun 2025 16:08:52 +0000
    Last State:     Terminated
      Reason:       Error
      Exit Code:    143
      Started:      Sun, 15 Jun 2025 15:45:23 +0000
      Finished:     Sun, 15 Jun 2025 16:08:51 +0000
    Ready:          False
```

ubuntu@ip-172-31-9-165:~/Probes$ kubectl exec -it javawebdeploy-b8fd9679b-2c7t2 -- /bin/sh
sh-4.4# kill 1
sh-4.4# command terminated with exit code 137

ubuntu@ip-172-31-9-165:~/Probes$ kubectl get pods -l app=javawebapp -w

NAME	READY	STATUS	RESTARTS	AGE
javawebdeploy-b8fd9679b-2c7t2	1/1	Running	1 (107s ago)	25m
javawebdeploy-b8fd9679b-6n2s8	1/1	Running	0	25m
javawebdeploy-b8fd9679b-vdjsl	1/1	Running	1 (12m ago)	25m

I killed the pod twice, now I see it is restarting because of the Liveness probe

```
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-rnmqx (ro)
Conditions:
  Type                        Status
  PodReadyToStartContainers   True
  Initialized                 True
  Ready                       False
  ContainersReady             False
  PodScheduled                True
Volumes:
  kube-api-access-rnmqx:
    Type:                     Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds:   3607
    ConfigMapName:            kube-root-ca.crt
    Optional:                 false
    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  28m                  default-scheduler  Successfully assigned default/javawebdeploy-b8fd96
pute.internal
  Normal   Pulled     28m                  kubelet            Successfully pulled image "hacker123shiva/springbt
aiting). Image size: 262359412 bytes.
  Normal   Pulling    5m24s (x2 over 28m)  kubelet            Pulling image "hacker123shiva/springbt-in-docker:l
  Normal   Created    5m23s (x2 over 28m)  kubelet            Created container: javawebapp
  Normal   Started    5m23s (x2 over 28m)  kubelet            Started container javawebapp
  Normal   Pulled     5m23s                kubelet            Successfully pulled image "hacker123shiva/springbt
aiting). Image size: 262359412 bytes.
  Warning  BackOff    3s (x3 over 7s)      kubelet            Back-off restarting failed container javawebapp in
b8b01-92d7-4e21-be43-82f29a759b7a)
ubuntu@ip-172-31-9-165:~/Probes$
```

Removed liveness

```
        app: javawebapp
    spec:
      containers:
        - name: javawebapp
          image: hacker123shiva/springbt-in-docker:latest
          ports:
            - containerPort: 8080 # Your Spring Boot app listens on 8080
          readinessProbe:
            initialDelaySeconds: 30
            periodSeconds: 5
            timeoutSeconds: 10
            successThreshold: 1
            failureThreshold: 3
            httpGet:
              path: /
              port: 8080 # Corrected: Probe must check the port your app is listening on
---
apiVersion: v1
kind: Service
metadata:
  name: javawebapp-service # Name for your LoadBalancer Service
  labels:
    app: javawebapp
spec:
```

ubuntu@ip-172-31-9-165:~/Probes$ kubectl apply -f readiness-live-manifest-v1.yml

ubuntu@ip-172-31-9-165:~/Probes$ kubectl get pods
NAME READY STATUS RESTARTS AGE
javawebdeploy-76f9f5b96c-gdnnh 0/1 Running 0 7s
javawebdeploy-76f9f5b96c-jhkj7 0/1 Running 0 7s
javawebdeploy-76f9f5b96c-jl7w7 0/1 Running 0 7s

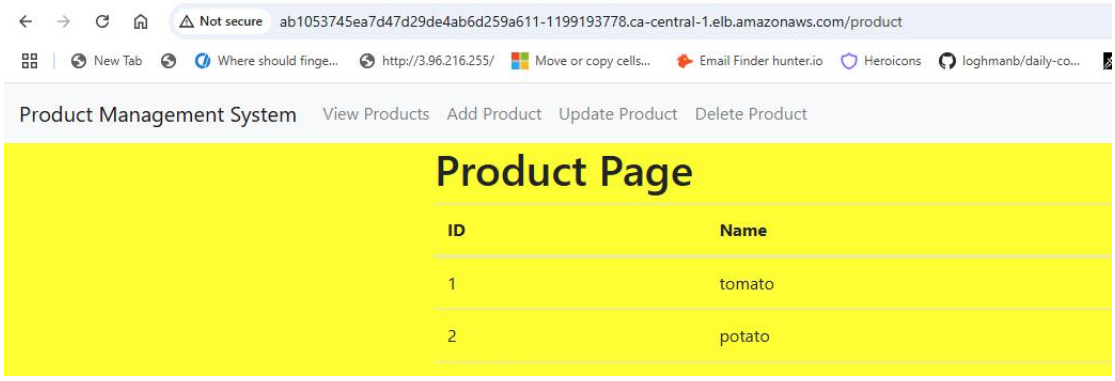
```
ubuntu@ip-172-31-9-165:~/Probes$ vi readiness-live-manifest-v1.yml
ubuntu@ip-172-31-9-165:~/Probes$ vi readiness-live-manifest-v1.yml
ubuntu@ip-172-31-9-165:~/Probes$ kubectl apply -f readiness-live-manifest-v1.yml
deployment.apps/javawebdeploy created
service/javawebapp-service created
ubuntu@ip-172-31-9-165:~/Probes$ kubectl get pods
NAME                             READY   STATUS    RESTARTS   AGE
javawebdeploy-76f9f5b96c-gdnnh   0/1     Running   0          7s
javawebdeploy-76f9f5b96c-jhkj7   0/1     Running   0          7s
javawebdeploy-76f9f5b96c-jl7w7   0/1     Running   0          7s
ubuntu@ip-172-31-9-165:~/Probes$
```

ubuntu@ip-172-31-9-165:~/Probes$ kubectl apply -f readiness-live-manifest-v1.yml
deployment.apps/javawebdeploy created
service/javawebapp-service created
ubuntu@ip-172-31-9-165:~/Probes$ kubectl get pods
NAME READY STATUS RESTARTS AGE
javawebdeploy-76f9f5b96c-gdnnh 0/1 Running 0 7s
javawebdeploy-76f9f5b96c-jhkj7 0/1 Running 0 7s
javawebdeploy-76f9f5b96c-jl7w7 0/1 Running 0 7s
ubuntu@ip-172-31-9-165:~/Probes$ kubectl get pods -l app=javawebapp -w
NAME READY STATUS RESTARTS AGE
javawebdeploy-76f9f5b96c-gdnnh 1/1 Running 0 55s
javawebdeploy-76f9f5b96c-jhkj7 1/1 Running 0 55s
javawebdeploy-76f9f5b96c-jl7w7 1/1 Running 0 55s

```
ubuntu@ip-172-31-9-165:~/Probes$ vi readiness-live-manifest-v1.yml
ubuntu@ip-172-31-9-165:~/Probes$ vi readiness-live-manifest-v1.yml
ubuntu@ip-172-31-9-165:~/Probes$ kubectl apply -f readiness-live-manifest-v1.yml
deployment.apps/javawebdeploy created
service/javawebapp-service created
ubuntu@ip-172-31-9-165:~/Probes$ kubectl get pods
NAME                              READY   STATUS    RESTARTS   AGE
javawebdeploy-76f9f5b96c-gdnnh   0/1     Running   0          7s
javawebdeploy-76f9f5b96c-jhkj7   0/1     Running   0          7s
javawebdeploy-76f9f5b96c-jl7w7   0/1     Running   0          7s
ubuntu@ip-172-31-9-165:~/Probes$ kubectl get pods -l app=javawebapp -w
NAME                              READY   STATUS    RESTARTS   AGE
javawebdeploy-76f9f5b96c-gdnnh   1/1     Running   0          55s
javawebdeploy-76f9f5b96c-jhkj7   1/1     Running   0          55s
javawebdeploy-76f9f5b96c-jl7w7   1/1     Running   0          55s
^Cubuntu@ip-172-31-9-165:~/Probes$
ubuntu@ip-172-31-9-165:~/Probes$
ubuntu@ip-172-31-9-165:~/Probes$
```

ubuntu@ip-172-31-9-165:~/Probes$ kubectl get svc
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
javawebapp-service LoadBalancer 10.100.127.99 ab1053745ea7d47d29de4ab6d259a611-1199193778.ca-central-1.elb.amazonaws.com 80:30926/TCP 116s
kubernetes ClusterIP 10.100.0.1 <none> 443/TCP 10m
ubuntu@ip-172-31-9-165:~/Probes$ kubectl get svc javawebapp-service
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
javawebapp-service LoadBalancer 10.100.127.99 ab1053745ea7d47d29de4ab6d259a611-1199193778.ca-central-1.elb.amazonaws.com 80:30926/TCP 2m10s

http://ab1053745ea7d47d29de4ab6d259a611-1199193778.ca-central-1.elb.amazonaws.com/product



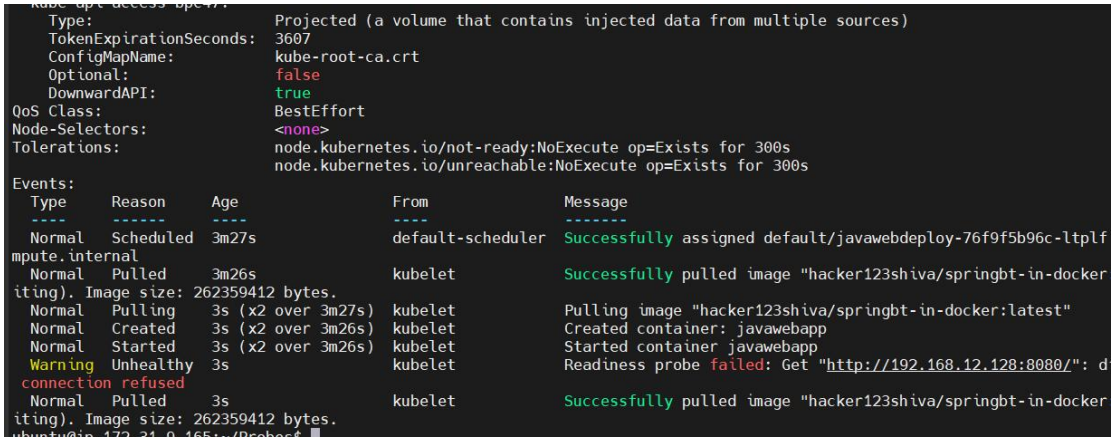
Demonstrating readiness probe

In the event section, you will notice about warning and backoff which demonstrate the probes

Traffic reroute -> Readiness probe and Restart container -> Liveness probe

Readiness probe -> Failure action -> Traffic routing control -> stops sending new connections to not ready pod (to tell the service to direct traffic somewhere else)
Liveness probe -> Failure action -> Container restart -> It terminates and restarts container

ubuntu@ip-172-31-9-165:~/Probes$ kubectl apply -f readiness-live-manifest-v1.yml
deployment.apps/javawebdeploy created
service/javawebapp-service created
ubuntu@ip-172-31-9-165:~/Probes$ kubectl get pods -l app=javawebapp -w
NAME READY STATUS RESTARTS AGE
javawebdeploy-76f9f5b96c-2fqt9 0/1 Running 0 14s
javawebdeploy-76f9f5b96c-b82bg 0/1 Running 0 14s
javawebdeploy-76f9f5b96c-ltplf 0/1 Running 0 14s
^Cubuntu@ip-172-31-9-165:~/Probes$



Normal Started 3s (x2 over 3m26s) kubelet Started container javawebapp
Warning Unhealthy 3s kubelet Readiness probe failed: Get "http://192.168.12.128:8080/": dial tcp 192.168.12.128:8080: connect: connection refused

Readiness probe, connection is refused

eksctl delete cluster --name my-eks-cluster --region ca-central-1