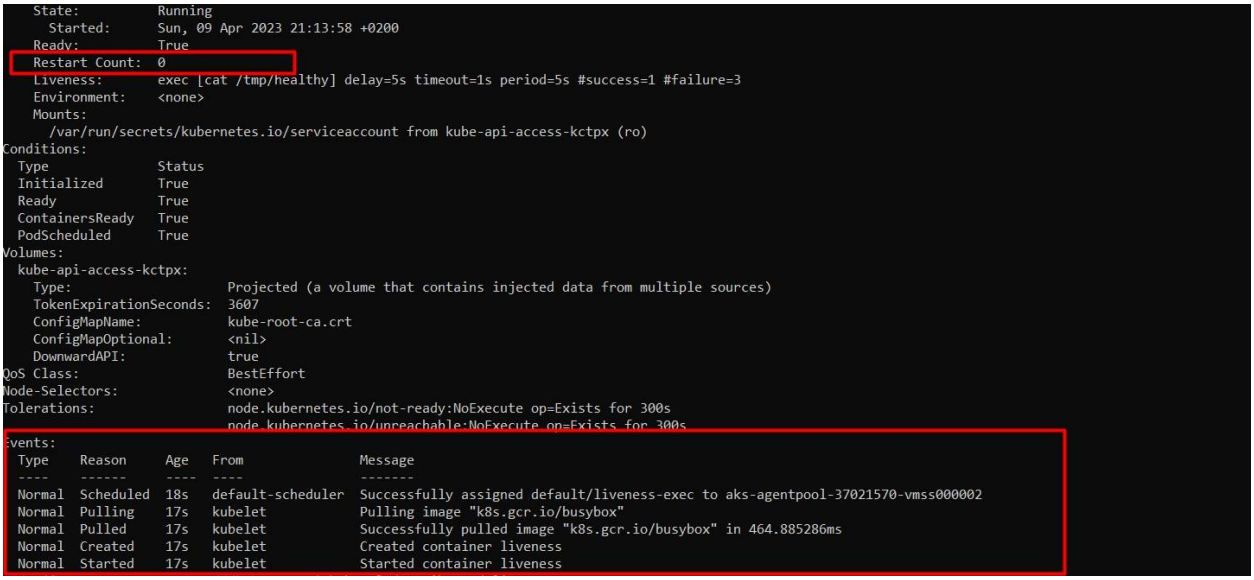
Practice4: Probes

1. First we will create and test liveness probe with exec test. Create a file named probes_exec.yaml with following content

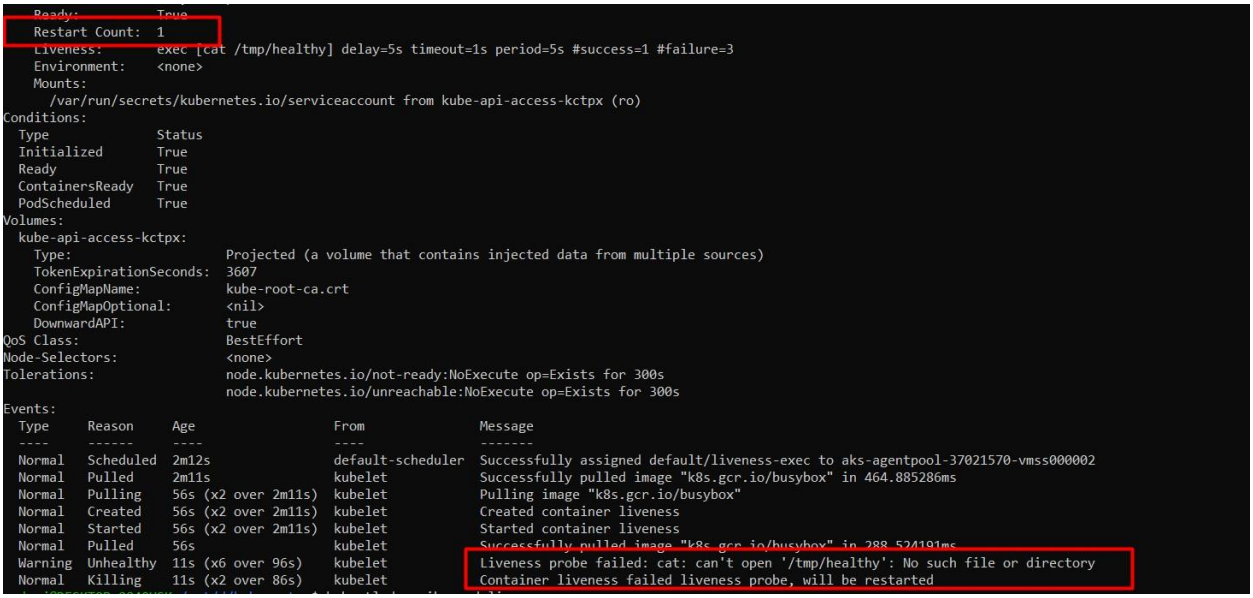
D: > kubernetes > ! probes_exec.yaml

```yaml
apiVersion: v1
kind: Pod
metadata:
 labels:
 test: liveness
 name: liveness-exec
spec:
 containers:
 - name: liveness
 image: k8s.gcr.io/busybox
 args:
 - /bin/sh
 - -c
 - touch /tmp/healthy; sleep 30; rm -rf /tmp/healthy; sleep 600
 livenessProbe:
 exec:
 command:
 - cat
 - /tmp/healthy
 initialDelaySeconds: 5
 periodSeconds: 5
```

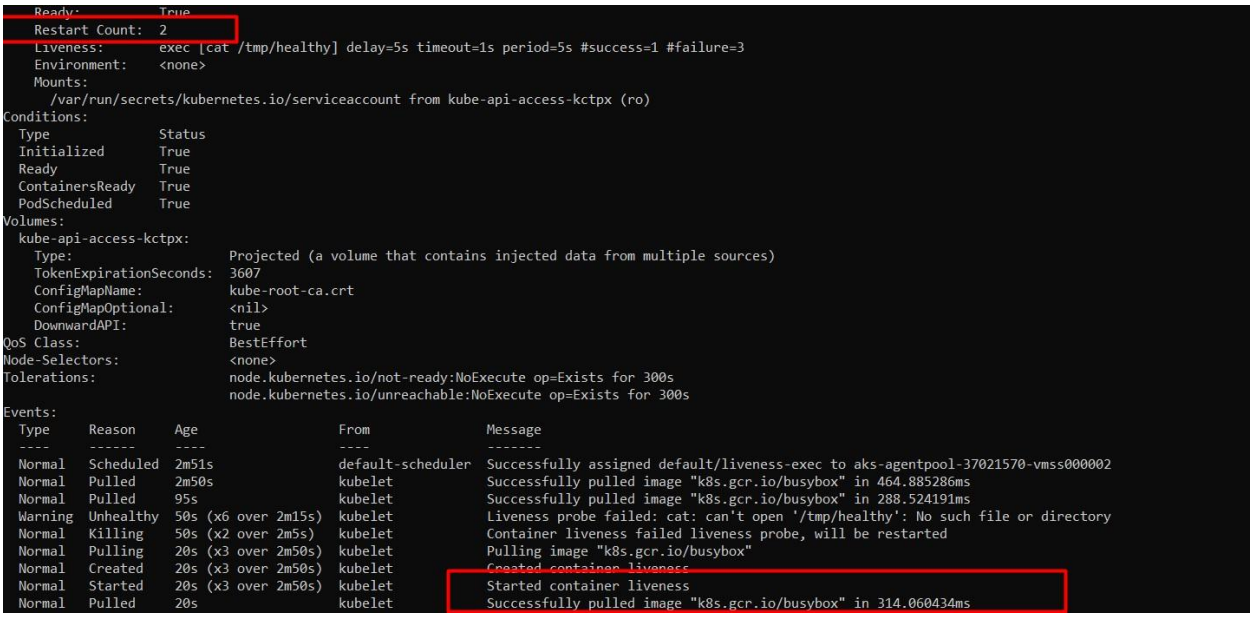
2. **Run kubectl create –f probes_exec.yaml**.
3. **Run kubectl describe pod liveness-exec** immediately after you deploy the pod. The output should indicate that no liveness probes have failed yet



4. After 35 seconds, view the Pod events again. Run kubectl describe pod liveness-exec



7. Wait another 30 seconds, and verify that the container has been restarted. Run kubectl get pod liveness-exec.



===The output should show **that RESTARTS has been incremented**===

9. Create file named probes_http.yaml with following content

probes_http.yaml

```yaml
apiVersion: v1
kind: Pod
metadata:
  labels:
  test: liveness
  name: liveness-http
spec:
  containers:
  - name: liveness
  image: k8s.gcr.io/liveness
  args:
  - /server
  livenessProbe:
  httpGet:
  path: /healthz
  port: 8080
  httpHeaders:
  - name: Custom-Header
  value: Awesome
  initialDelaySeconds: 3
  periodSeconds: 3
```

For the first 10 seconds that the container is alive, the /healthz handler returns a status of 200. After that, the handler returns a status of 500

-Run kubectl create –f probes_http.yaml.

-Immediately run (you only have 10 secs to run this command) **kubectl describe pod liveness-http**

```
    Ready:          False
    Restart Count:  0
    Liveness:       http-get http://:8080/healthz delay=3s timeout=1s period=3s #success=1 #failure=3
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-fn24t (ro)
Conditions:
  Type              Status
  Initialized       True
  Ready             False
  ContainersReady   False
  PodScheduled      True
Volumes:
  kube-api-access-fn24t:
    Type:                    Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds:  3607
    ConfigMapName:           kube-root-ca.crt
    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  4s    default-scheduler  Successfully assigned default/liveness-http to aks-agentpool-37021570-vmss000002
  Normal  Pulling    4s    kubelet            Pulling image "k8s.gcr.io/echoserver:1.4"
```

After 10 seconds, view Pod events to verify that liveness probes have failed and the container has been restarted. Run again **kubectl describe pod liveness-http**

```
  Ready:          False
  Restart Count:  2
  Liveness:       http-get http://:8080/healthz delay=3s timeout=1s period=3s #success=1 #failure=3
  Environment:    <none>
  Mounts:
    /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-fn24t (ro)
Conditions:
  Type              Status
  Initialized       True
  Ready             False
  ContainersReady   False
  PodScheduled      True
Volumes:
  kube-api-access-fn24t:
    Type:                    Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds:  3607
    ConfigMapName:           kube-root-ca.crt
    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  26s                default-scheduler  Successfully assigned default/liveness-http to aks-agentpool-37021570-vmss000002
  Normal   Pulling    26s                kubelet            Pulling image "k8s.gcr.io/echoserver:1.4"
  Normal   Pulled     22s                kubelet            Successfully pulled image "k8s.gcr.io/echoserver:1.4" in 3.410420255s
  Normal   Created    3s (x3 over 22s)   kubelet            Created container liveness
  Warning  Failed     3s (x3 over 21s)   kubelet            Error: failed to create containerd task: failed to create shim task: OCI runtime create failed: container l
```

Create file named **probes_tcp.yaml** with following content:

probes_tcp.yaml

D: > kubernetes > probes_tcp.yaml

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: liveness-tcp
  labels:
    app: goproxy
spec:
  containers:
  - name: goproxy
    image: k8s.gcr.io/goproxy:0.1
    ports:
    - containerPort: 8080
    livenessProbe:
      tcpSocket:
        port: 8080 # Updated port to match containerPort
      initialDelaySeconds: 15
      periodSeconds: 20

```

In the lab tcpSocket/port value was 9999

```
    Ready:          True
    Restart Count:  3
    Liveness:       tcp-socket :9999 delay=15s timeout=1s period=20s #success=1 #failure=3
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-n57b6 (ro)
Conditions:
  Type              Status
  Initialized       True
  Ready             True
  ContainersReady   True
  PodScheduled      True
Volumes:
  kube-api-access-n57b6:
    Type:                    Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds:  3607
    ConfigMapName:           kube-root-ca.crt
    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  3m9s               default-scheduler  Successfully assigned default/liveness-tcp to aks-agentpool-37021570-vmss000002
  Normal   Pulled     9s (x4 over 3m9s)  kubelet            Container image "k8s.gcr.io/goproxy:0.1" already present on machine
  Normal   Created    9s (x4 over 3m9s)  kubelet            Created container goproxy
  Normal   Started    9s (x4 over 3m8s)  kubelet            Started container goproxy
  Warning  Unhealthy  9s (x9 over 2m49s) kubelet            Liveness probe failed: dial tcp 10.244.0.26:9999: connect: connection refused
  Normal   Killing    9s (x3 over 2m9s)  kubelet            Container goproxy failed liveness probe, will be restarted
```

And we can see that it failed.

After the update of the port to 8080 like in the picture above we can see that liveness-tcp pod is running .

```
Events:
  Type    Reason     Age   From               Message
  ----    ------     ----  ----               -------
  Normal  Scheduled  27s   default-scheduler  Successfully assigned default/liveness-tcp to aks-agentpool-37021570-vmss000002
  Normal  Pulled     27s   kubelet            Container image "k8s.gcr.io/goproxy:0.1" already present on machine
  Normal  Created    27s   kubelet            Created container goproxy
  Normal  Started    27s   kubelet            Started container goproxy
andrej@DESKTOP-8840HSK:/mnt/d/kubernetes$ kubectl get pods
NAME            READY   STATUS             RESTARTS          AGE
liveness-exec   0/1     CrashLoopBackOff   16 (2m40s ago)    45m
liveness-http   0/1     CrashLoopBackOff   10 (104s ago)     27m
liveness-tcp    1/1     Running            0                 2m9s
andrej@DESKTOP-8840HSK:/mnt/d/kubernetes$
```

Create file named **readiness_http.yaml** with following content:

```
D: > kubernetes > ! readiness_http.yaml
apiVersion: v1
kind: Pod
metadata:
  name: readiness-http
  labels:
    app: test
spec:
  containers:
  - name: nginx
    image: nginx
    ports:
    - containerPort: 80
    readinessProbe:
      initialDelaySeconds: 1
      periodSeconds: 2
      timeoutSeconds: 1
      successThreshold: 1
      failureThreshold: 1
      httpGet:
        host:
        scheme: HTTP
        path: /
        httpHeaders:
        - name: Host
          value: myapplication1.com
        port: 80
```

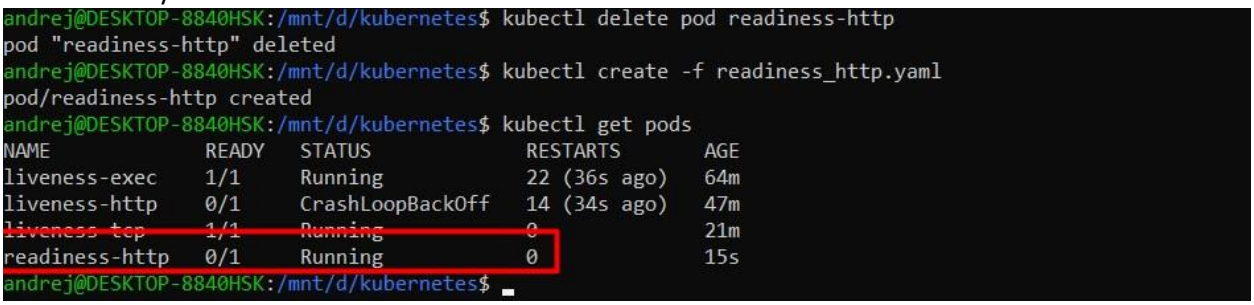
```
andrej@DESKTOP-8840HSK:/mnt/d/kubernetes$ kubectl create -f readiness_http.yaml
pod/readiness-http created
```

Run **kubectl get pods** to see the status of your pod.

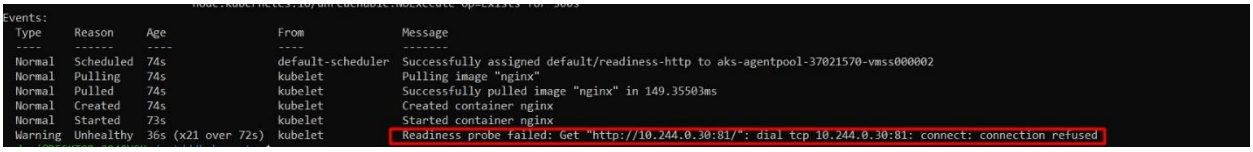
```
andrej@DESKTOP-8840HSK:/mnt/d/kubernetes$ kubectl get pods
NAME             READY   STATUS             RESTARTS         AGE
liveness-exec    1/1     Running            20 (8s ago)      56m
liveness-http    0/1     CrashLoopBackOff   12 (2m49s ago)   39m
liveness-tcp     1/1     Running            0                13m
readiness-http   1/1     Running            0                85s
andrej@DESKTOP-8840HSK:/mnt/d/kubernetes$
```

Run **kubectl describe pod readiness-http**. Examine the events for this pod. Everything should be OK

Now delete the pod and edit the readiness_http.yaml so that the port parameter **has 81 value**.
Run again **kubectl create –f readiness_http.yaml**.
Run **kubectl get pods** to see the status of your pod. You should see that the pod is running but it is not in ready state



Describe the pod. Run **kubectl describe pod readiness-http**.



===From the events we can see that readiness probe failed due to the connection being refused therefore pod will not receive any traffic====

Delete all pods under the default namespace.

```
andrej@DESKTOP-8840HSK:/mnt/d/kubernetes$ kubectl delete pods --all -n default
pod "liveness-exec" deleted
pod "liveness-http" deleted
pod "liveness-tcp" deleted
pod "readiness-http" deleted
andrej@DESKTOP-8840HSK:/mnt/d/kubernetes$
```