PROBING

About Probe:-

Probe will use for health checks of container pod, two types of probes are there, liveness and readiness probe.

Liveness probe will check whether container is up or down.

Readiness Probe will check whether container is up or down, however in additional readiness will check the container, whether it's ready to accept the request or not.

In Aws, in backend of load balancing, once system bring into it and will not register immediately, first load balance will register then will do health checks like whether up or down, liveness and readiness both has same parameters but bit different name only.

Health checks can be done in three ways of method, by using three types of handlers to check condition. (ExecAction, TCPSocketAction, HTTPGetAction)

Actual Readme:-

A Probe is a diagnostic performed periodically by the kubelet on a Container. To perform a diagnostic, the kubelet calls a Handler implemented by the Container. There are three types of handlers

Handlers Type:

******

Exec Action: Executes a specified command inside the Container. The diagnostic is considered successful if the command exits with a status code of 0.

TCPSocketAction: Performs a TCP check against the Container's IP address on a specified port. The diagnostic is considered successful if the port is open.

HTTPGetAction: Performs an HTTP Get request against the Container's IP address on a specified port and path. The diagnostic is considered successful if the response has a status code greater than or equal to 200 and less than 400.

Each probe has one of three results:

*************Success: The Container passed the diagnostic.

Failure: The Container failed the diagnostic.

Unknown: The diagnostic failed, so no action should be taken.

The kubelet can optionally perform and react to two kinds of probes on running Containers:

livenessProbe: Indicates whether the Container is running. If the liveness probe fails, the kubelet kills the Container, and the Container is subjected to its restart policy. If a Container does not provide a liveness probe, the default state is Success.

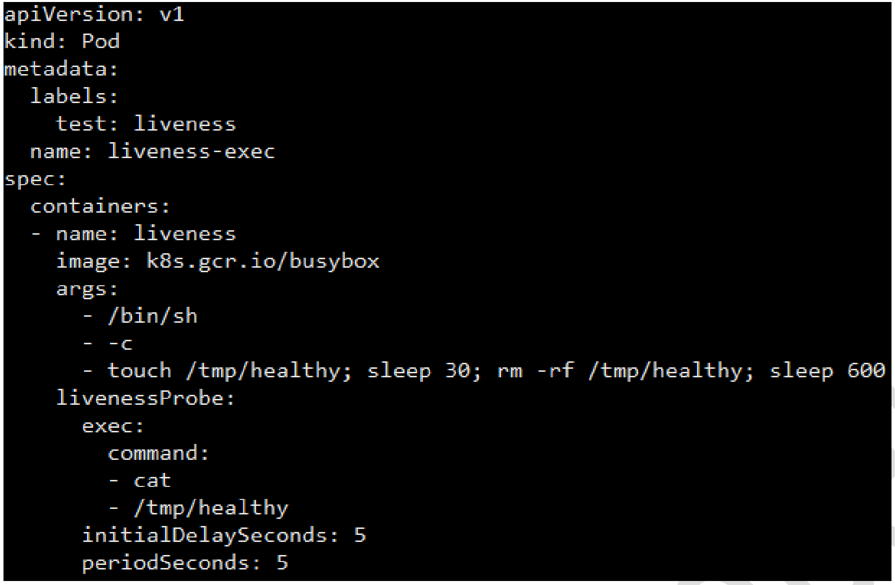
readinessProbe: Indicates whether the Container is ready to service requests. If the readiness probe fails, the endpoints controller removes the Pod's IP address from the endpoints of all Services that match the Pod. The default state of readiness before the initial delay is Failure. If a Container does not provide a readiness probe, the default state is Success

A PodSpec has a restartPolicy field with possible values Always, On Failure, and Never. The default value is Always.

##Create a pod with ExecAction##

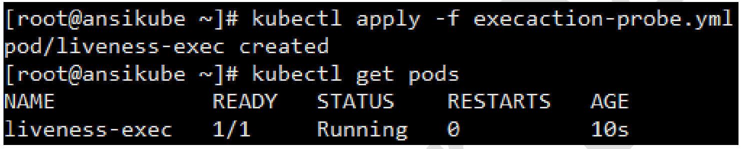
Executes a specified command inside the Container. The diagnostic is considered successful if the command exits with a status code of 0.

#cat execaction-probe.yml



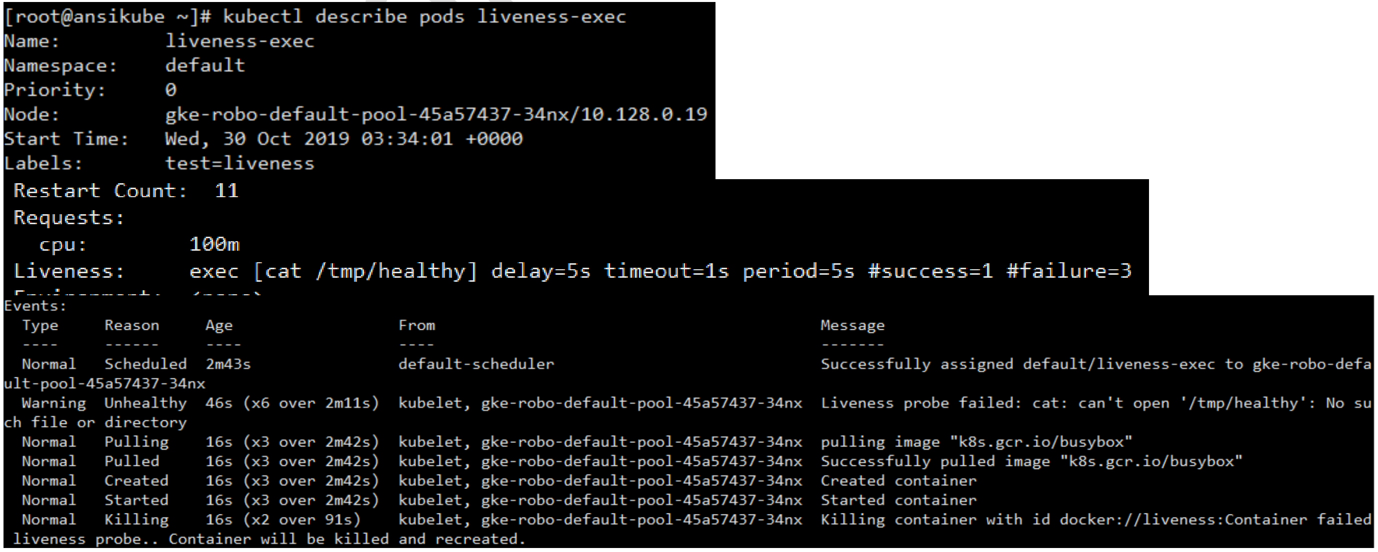
```
apiVersion: v1
kind: Pod
metadata:
  labels:
    test: liveness
  name: liveness-exec
spec:
  containers:
  - name: liveness
    image: k8s.gcr.io/busybox
    args:
      - /bin/sh
      - -c
      - touch /tmp/healthy; sleep 30; rm -rf /tmp/healthy; sleep 600
    livenessProbe:
      exec:
    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

#kubectl apply -f execaction-probe.yml
#kubectl get pods



```
[root@ansikube ~]# kubectl apply -f execaction-probe.yml
pod/liveness-exec created
[root@ansikube ~]# kubectl get pods
NAME            READY   STATUS    RESTARTS   AGE
liveness-exec   1/1     Running   0          10s
```

#kubectl describe pods liveness-exec



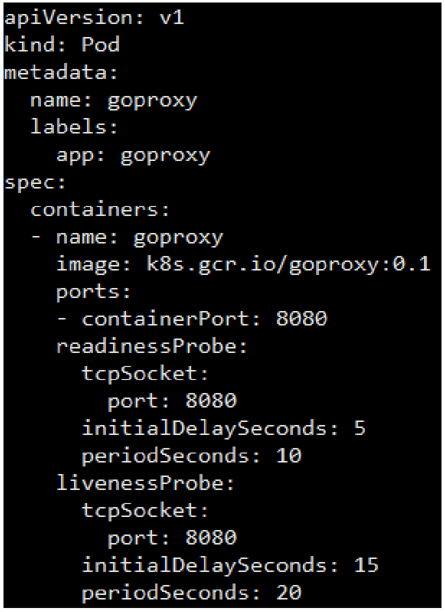
```
[root@ansikube ~]# kubectl describe pods liveness-exec
Name:         liveness-exec
Namespace:    default
Priority:     0
Node:         gke-robo-default-pool-45a57437-34nx/10.128.0.19
Start Time:   Wed, 30 Oct 2019 03:34:01 +0000
Labels:       test=liveness
 Restart Count:  11
 Requests:
   cpu:        100m
 Liveness:     exec [cat /tmp/healthy] delay=5s timeout=1s period=5s #success=1 #failure=3
Events:
  Type     Reason     Age                 From                                         Message
  ----     ------     ----                ----                                         -------
  Normal   Scheduled  2m43s               default-scheduler                            Successfully assigned default/liveness-exec to gke-robo-defa
ult-pool-45a57437-34nx
  Warning  Unhealthy  46s (x6 over 2m11s)  kubelet, gke-robo-default-pool-45a57437-34nx  Liveness probe failed: cat: can't open '/tmp/healthy': No su
ch file or directory
  Normal   Pulling    16s (x3 over 2m42s)  kubelet, gke-robo-default-pool-45a57437-34nx  pulling image "k8s.gcr.io/busybox"
  Normal   Pulled     16s (x3 over 2m42s)  kubelet, gke-robo-default-pool-45a57437-34nx  Successfully pulled image "k8s.gcr.io/busybox"
  Normal   Created    16s (x3 over 2m42s)  kubelet, gke-robo-default-pool-45a57437-34nx  Created container
  Normal   Started    16s (x3 over 2m42s)  kubelet, gke-robo-default-pool-45a57437-34nx  Started container
  Normal   Killing    16s (x2 over 91s)    kubelet, gke-robo-default-pool-45a57437-34nx  Killing container with id docker://liveness:Container failed
 liveness probe.. Container will be killed and recreated.
```

Note:- “kubectl describe” will show the details of pod, however i have pasted few outputs on above fYI… and in output, exec is failing because there is no /tmp/healthy file present.

##Create a pod with TCPSocketAction##

Performs a TCP check against the Container's IP address on a specified port. The diagnostic is considered successful if the port is open.

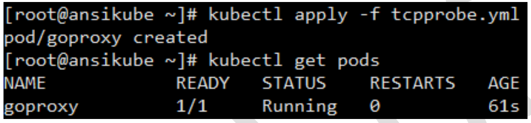
#cat tcpprobe.yml



```
apiVersion: v1
kind: Pod
metadata:
  name: goproxy
  labels:
    app: goproxy
spec:
  containers:
  - name: goproxy
    image: k8s.gcr.io/goproxy:0.1
    ports:
    - containerPort: 8080
    readinessProbe:
      tcpSocket:
        port: 8080
      initialDelaySeconds: 5
      periodSeconds: 10
    livenessProbe:
      tcpSocket:
        port: 8080
      initialDelaySeconds: 15
      periodSeconds: 20
```

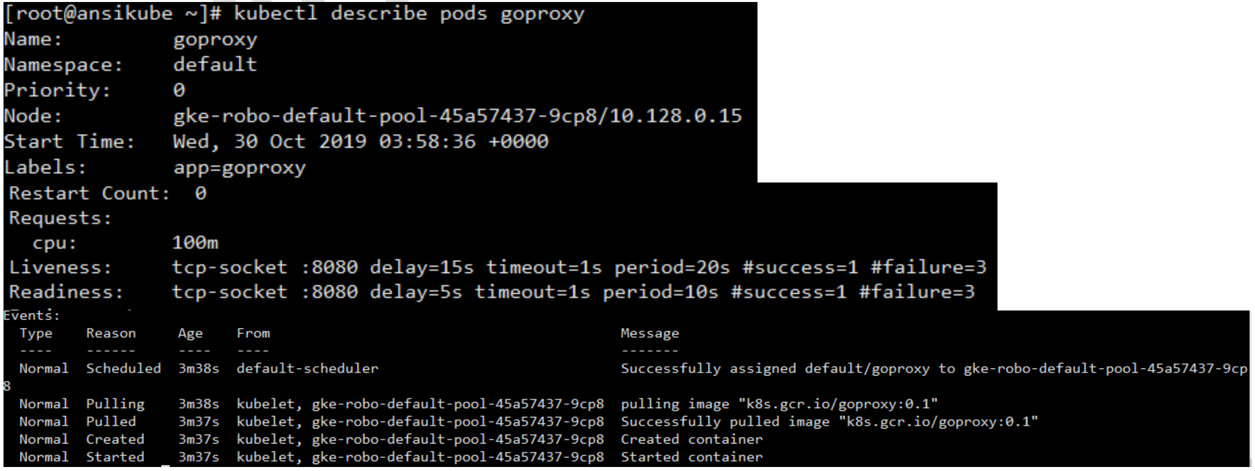
#kubectl apply -f tcpprobe.yml

#kubectl get pods



```
[root@ansikube ~]# kubectl apply -f tcpprobe.yml
pod/goproxy created
[root@ansikube ~]# kubectl get pods
NAME             READY   STATUS    RESTARTS   AGE
goproxy          1/1     Running   0          61s
```

#kubectl describe pods goproxy



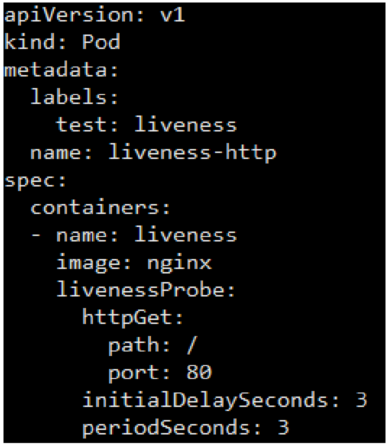
```
[root@ansikube ~]# kubectl describe pods goproxy
Name:         goproxy
Namespace:    default
Priority:     0
Node:         gke-robo-default-pool-45a57437-9cp8/10.128.0.15
Start Time:   Wed, 30 Oct 2019 03:58:36 +0000
Labels:       app=goproxy
 Restart Count:  0
 Requests:
   cpu:        100m
 Liveness:     tcp-socket :8080 delay=15s timeout=1s period=20s #success=1 #failure=3
 Readiness:    tcp-socket :8080 delay=5s timeout=1s period=10s #success=1 #failure=3
Events:
  Type    Reason     Age    From                                         Message
  ----    ------     ----   ----                                         -------
  Normal  Scheduled  3m38s  default-scheduler                            Successfully assigned default/goproxy to gke-robo-default-pool-45a57437-9cp
8
  Normal  Pulling    3m38s  kubelet, gke-robo-default-pool-45a57437-9cp8  pulling image "k8s.gcr.io/goproxy:0.1"
  Normal  Pulled     3m37s  kubelet, gke-robo-default-pool-45a57437-9cp8  Successfully pulled image "k8s.gcr.io/goproxy:0.1"
  Normal  Created    3m37s  kubelet, gke-robo-default-pool-45a57437-9cp8  Created container
  Normal  Started    3m37s  kubelet, gke-robo-default-pool-45a57437-9cp8  Started container
```

Note:- Above will show the output of Liveness and Readiness status.

##Create a pod with HTTPGetAction##

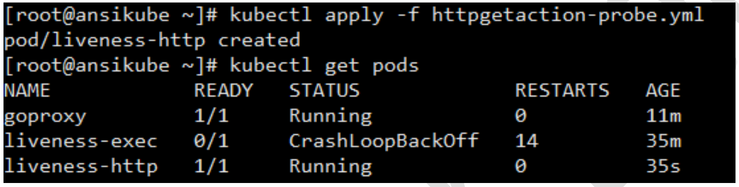
Performs an HTTP Get request against the Container's IP address on a specified port and path. The diagnostic is considered successful if the response has a status code greater than or equal to 200 and less than 400.

#cat httpgetaction-probe.yml



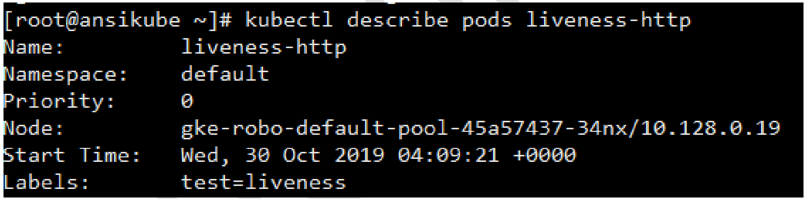
```
apiVersion: v1
kind: Pod
metadata:
  labels:
    test: liveness
  name: liveness-http
spec:
  containers:
  - name: liveness
    image: nginx
    livenessProbe:
      httpGet:
        path: /
        port: 80
      initialDelaySeconds: 3
      periodSeconds: 3
```

#kubectl apply -f httpgetaction-probe.yml
#kubectl get pods

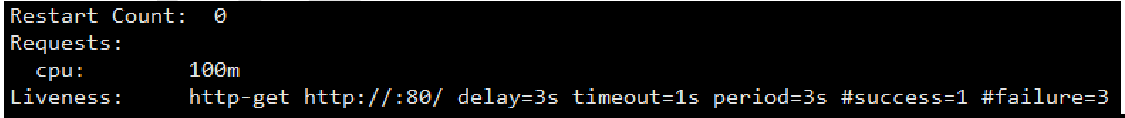


```
[root@ansikube ~]# kubectl apply -f httpgetaction-probe.yml
pod/liveness-http created
[root@ansikube ~]# kubectl get pods
NAME            READY   STATUS             RESTARTS   AGE
goproxy         1/1     Running            0          11m
liveness-exec   0/1     CrashLoopBackOff   14         35m
liveness-http   1/1     Running            0          35s
```

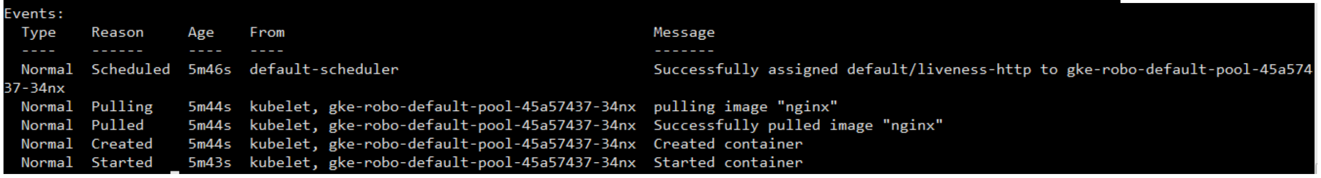
kubectl describe pods liveness-http



```
[root@ansikube ~]# kubectl describe pods liveness-http
Name:         liveness-http
Namespace:    default
Priority:     0
Node:         gke-robo-default-pool-45a57437-34nx/10.128.0.19
Start Time:   Wed, 30 Oct 2019 04:09:21 +0000
Labels:       test=liveness
```



```
 Restart Count:  0
 Requests:
   cpu:        100m
 Liveness:     http-get http://:80/ delay=3s timeout=1s period=3s #success=1 #failure=3
```



```
Events:
  Type    Reason     Age    From                                          Message
  ----    ------     ----   ----                                          -------
  Normal  Scheduled  5m46s  default-scheduler                             Successfully assigned default/liveness-http to gke-robo-default-pool-45a574
37-34nx
  Normal  Pulling    5m44s  kubelet, gke-robo-default-pool-45a57437-34nx  pulling image "nginx"
  Normal  Pulled     5m44s  kubelet, gke-robo-default-pool-45a57437-34nx  Successfully pulled image "nginx"
  Normal  Created    5m44s  kubelet, gke-robo-default-pool-45a57437-34nx  Created container
  Normal  Started    5m43s  kubelet, gke-robo-default-pool-45a57437-34nx  Started container
```

Note:- Above output will show the liveness probe status.