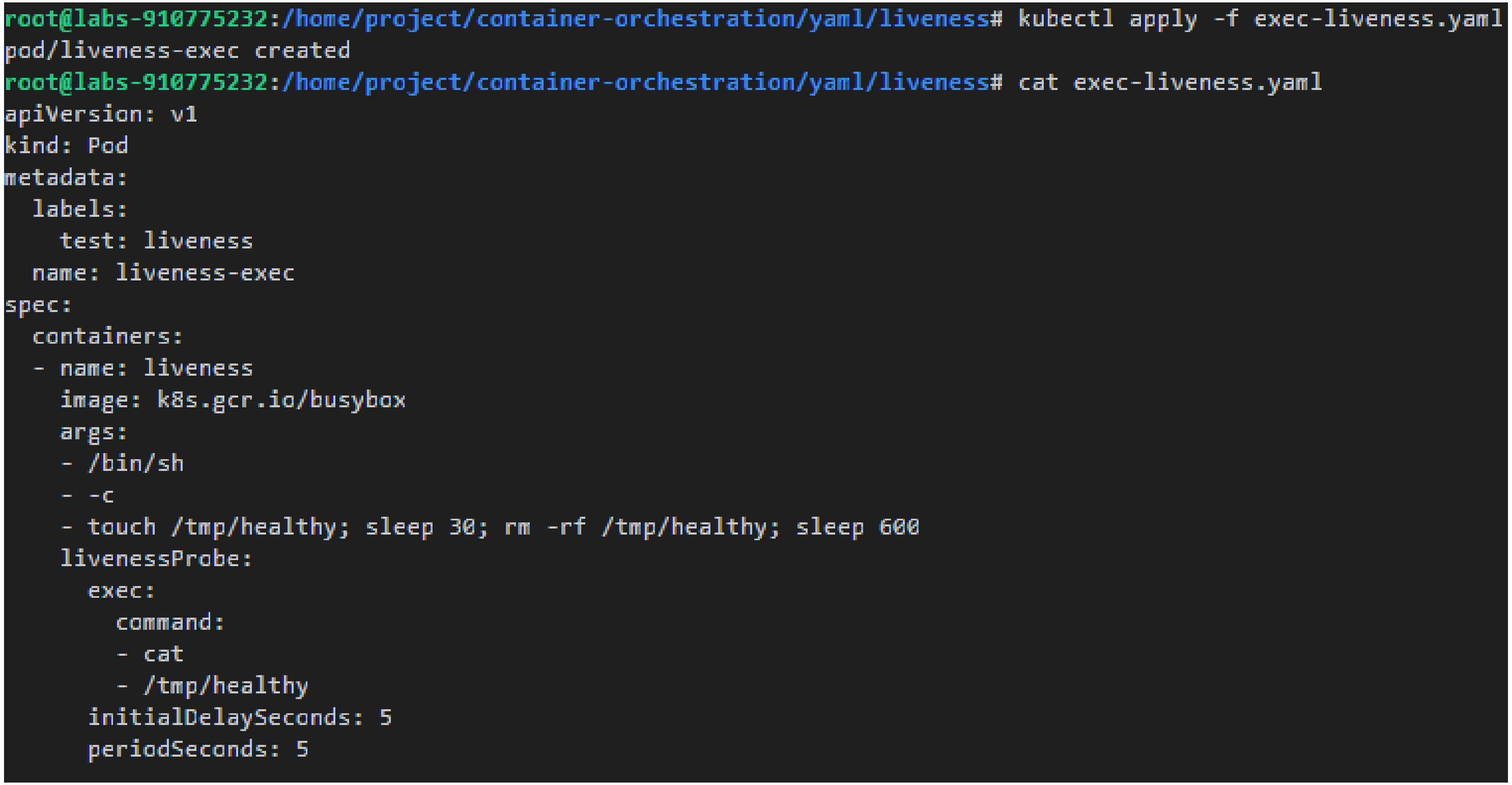
Probe types

쿠버네티스는 각 컨테이너의 **상태를 주기적으로 체크 (Health Check)**해서 문제가 있는 컨테이너는 자동으로 재시작 하거나, 서비스에서 제외하는 등 실행중인 상태를 파악하고 라이프 사이클을 제어한다.

Liveness Probe	Readiness Probe	Startup Probe
-. 문제가 발생한 **컨테이너를 종료**하고, Restart Policy*(default : Always)*에 따라 다시 만들어지거나, 종료된 상태로 남음.	-. 문제가 발생한 컨테이너를 **서비스 백엔드에서 일시적으로 제외**함. -. 해당 컨테이너를 임시적으로 disable 상태로 전환시킨 후, 지속적으로 핸들러를 호출하여 결과를 확인 함. **요청을 받을 준비가 되었는지 확인함**	-. 일부 컨테이너 시작시 많은 시간이 소요될 수 있는데, 이런 경우 **다른 Probe 보다 먼저 실행**됨. **컨테이너 라이프 사이클에서 시작시, 1번 수행되는 Probe 다른 Probe들의 버든을 줄여줌**

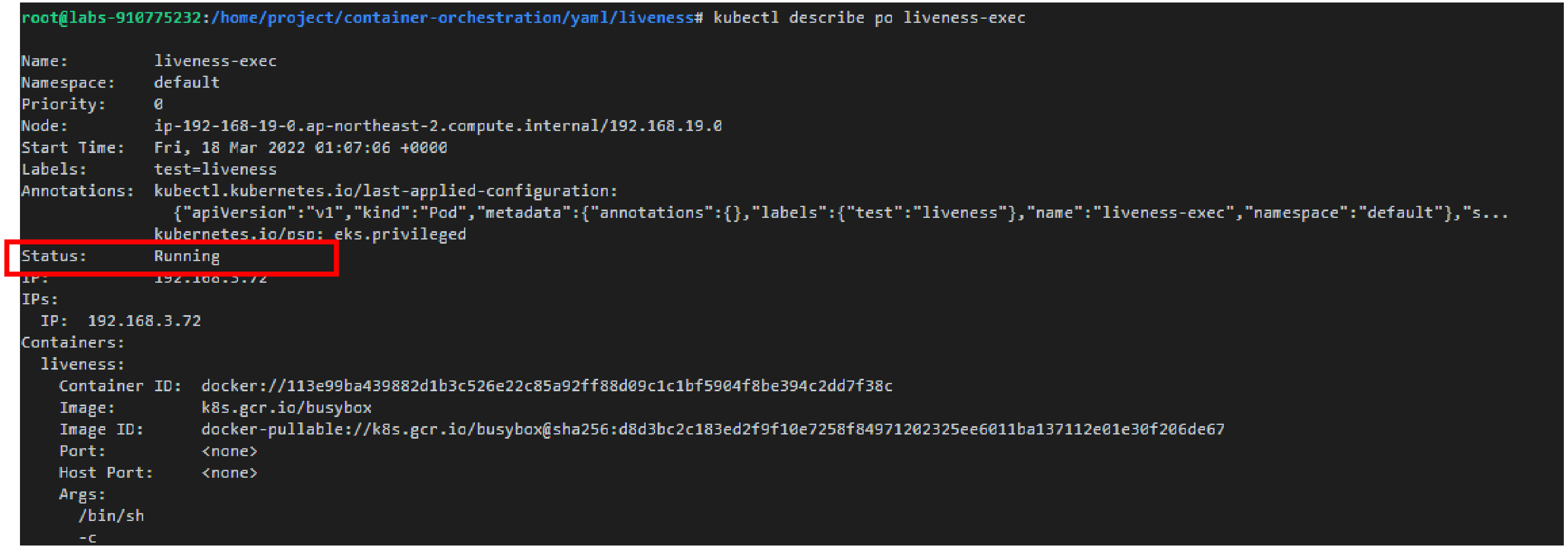
Liveness- Command Probe

kubectl apply -f exec-liveness.yaml

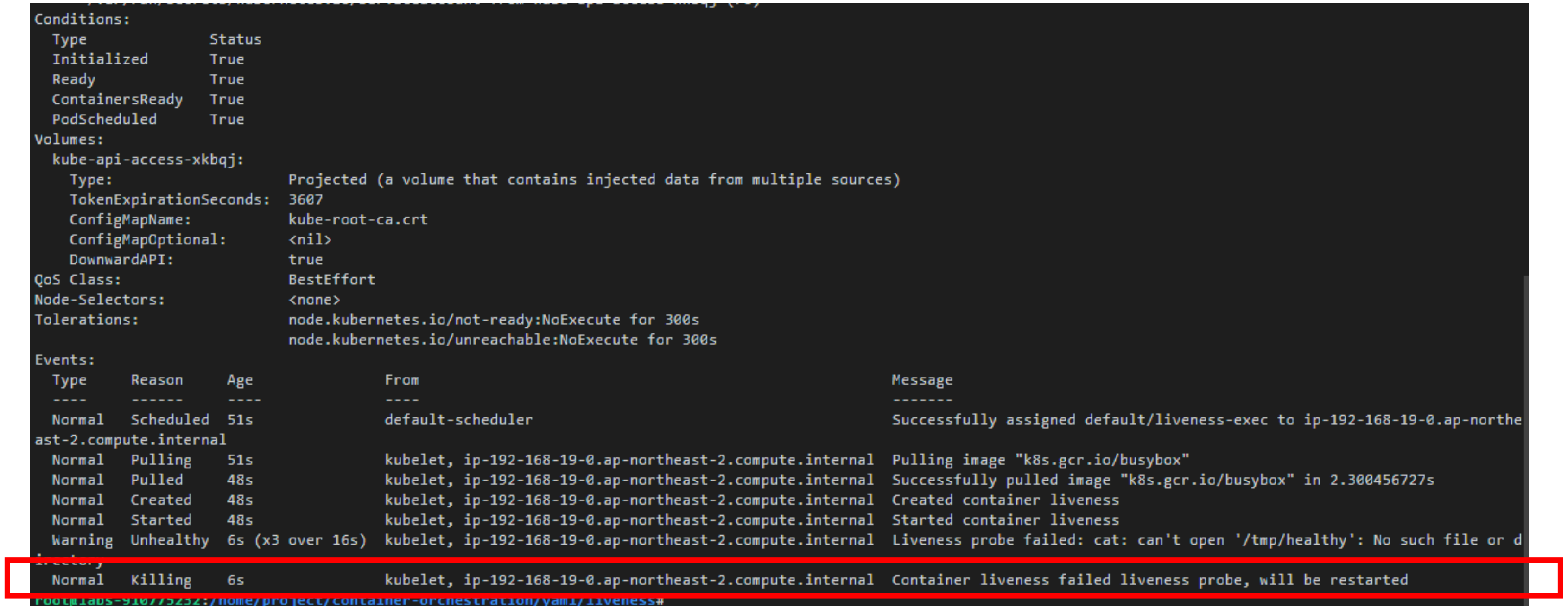


```
root@labs-910775232:/home/project/container-orchestration/yaml/liveness# kubectl apply -f exec-liveness.yaml
pod/liveness-exec created
root@labs-910775232:/home/project/container-orchestration/yaml/liveness# cat exec-liveness.yaml
apiVersion: v1
kind: Pod
metadata:
  labels:
    test: liveness
  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
```

kubectl describe po liveness-exec



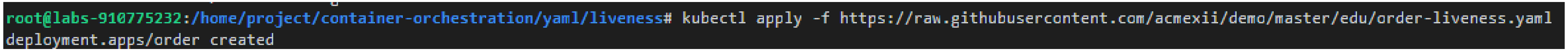
```
root@labs-910775232:/home/project/container-orchestration/yaml/liveness# kubectl describe po liveness-exec

Name:         liveness-exec
Namespace:    default
Priority:     0
Node:         ip-192-168-19-0.ap-northeast-2.compute.internal/192.168.19.0
Start Time:   Fri, 18 Mar 2022 01:07:06 +0000
Labels:       test=liveness
Annotations:  kubectl.kubernetes.io/last-applied-configuration:
                {"apiVersion":"v1","kind":"Pod","metadata":{"annotations":{},"labels":{"test":"liveness"},"name":"liveness-exec","namespace":"default"},"s...
              kubernetes.io/psp: eks.privileged
Status:       Running
IP:           192.168.3.72
IPs:
  IP:  192.168.3.72
Containers:
  liveness:
    Container ID:  docker://113e99ba439882d1b3c526e22c85a92ff88d09c1c1bf5904f8be394c2dd7f38c
    Image:         k8s.gcr.io/busybox
    Image ID:      docker-pullable://k8s.gcr.io/busybox@sha256:d8d3bc2c183ed2f9f10e7258f84971202325ee6011ba137112e01e30f206de67
    Port:          <none>
    Host Port:     <none>
    Args:
      /bin/sh
      -c
```



```
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-xkbqj (ro)
Conditions:
  Type              Status
  Initialized       True
  Ready             True
  ContainersReady   True
  PodScheduled      True
Volumes:
  kube-api-access-xkbqj:
    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 for 300s
                             node.kubernetes.io/unreachable:NoExecute for 300s

Events:
  Type     Reason     Age                From                                              Message
  ----     ------     ----               ----                                              -------
  Normal   Scheduled  51s                default-scheduler                                 Successfully assigned default/liveness-exec to ip-192-168-19-0.ap-northe
ast-2.compute.internal
  Normal   Pulling    51s                kubelet, ip-192-168-19-0.ap-northeast-2.compute.internal  Pulling image "k8s.gcr.io/busybox"
  Normal   Pulled     48s                kubelet, ip-192-168-19-0.ap-northeast-2.compute.internal  Successfully pulled image "k8s.gcr.io/busybox" in 2.300456727s
  Normal   Created    48s                kubelet, ip-192-168-19-0.ap-northeast-2.compute.internal  Created container liveness
  Normal   Started    48s                kubelet, ip-192-168-19-0.ap-northeast-2.compute.internal  Started container liveness
  Warning  Unhealthy  6s (x3 over 16s)   kubelet, ip-192-168-19-0.ap-northeast-2.compute.internal  Liveness probe failed: cat: can't open '/tmp/healthy': No such file or d
irectory
  Normal   Killing    6s                 kubelet, ip-192-168-19-0.ap-northeast-2.compute.internal  Container liveness failed liveness probe, will be restarted
root@labs-910775232:/home/project/container-orchestration/yaml/liveness#
```

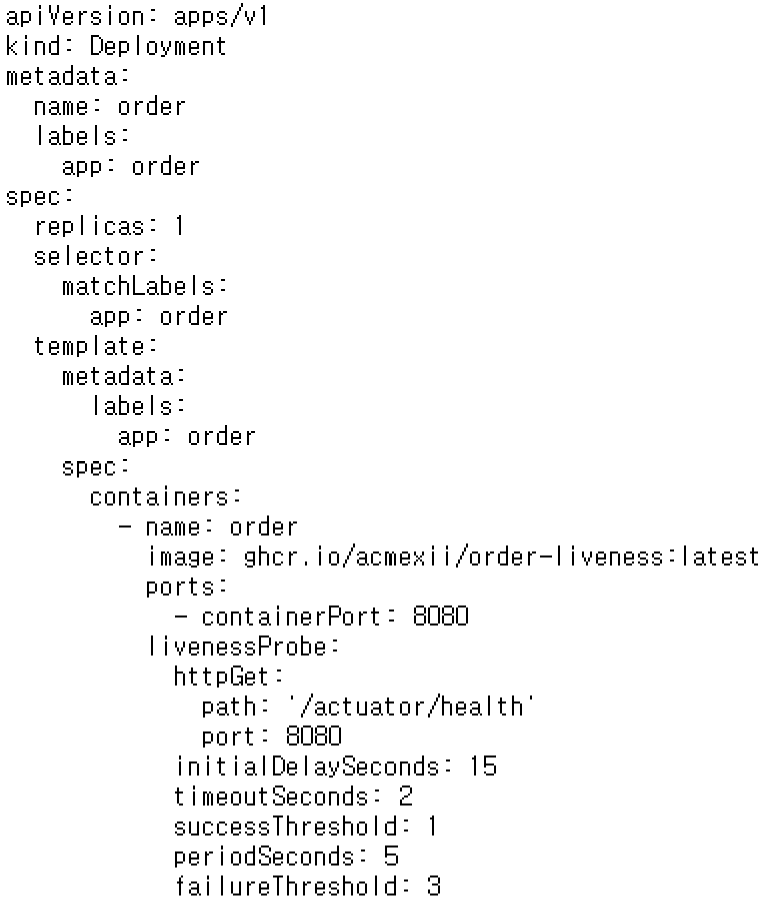
컨테이너 상태가 Running 상태로 보이나, liveness Probe로 인해 종료되고 재시작 됨을 확인

Liveness- HTTP Probe Liveness가 적용된 주문 마이크로 서비스 배포



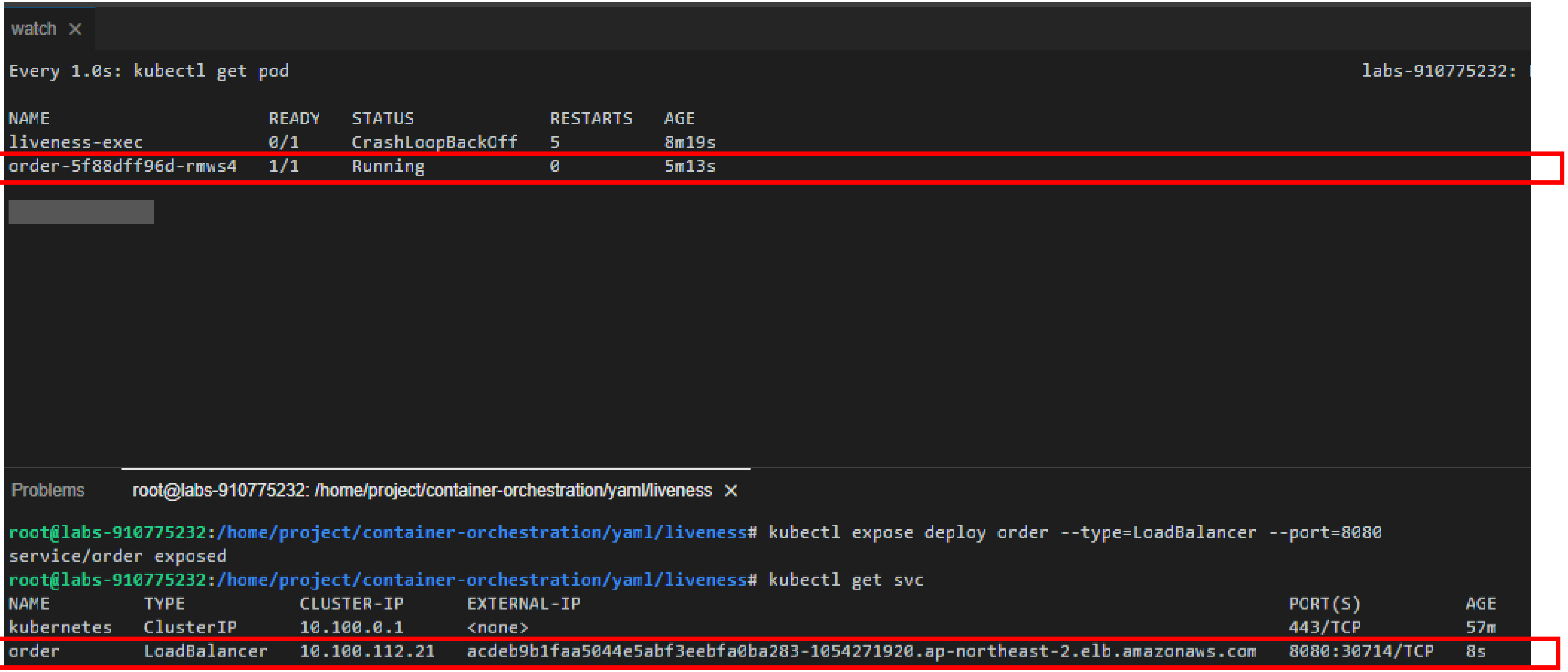
```
root@labs-910775232:/home/project/container-orchestration/yaml/liveness# kubectl apply -f https://raw.githubusercontent.com/acmexii/demo/master/edu/order-liveness.yaml
deployment.apps/order created
```

order-liveness.yaml



```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: order
  labels:
    app: order
spec:
  replicas: 1
  selector:
    matchLabels:
      app: order
  template:
    metadata:
      labels:
        app: order
    spec:
      containers:
        - name: order
          image: ghcr.io/acmexii/order-liveness:latest
          ports:
            - containerPort: 8080
          livenessProbe:
            httpGet:
              path: '/actuator/health'
              port: 8080
            initialDelaySeconds: 15
            timeoutSeconds: 2
            successThreshold: 1
            periodSeconds: 5
            failureThreshold: 3
```

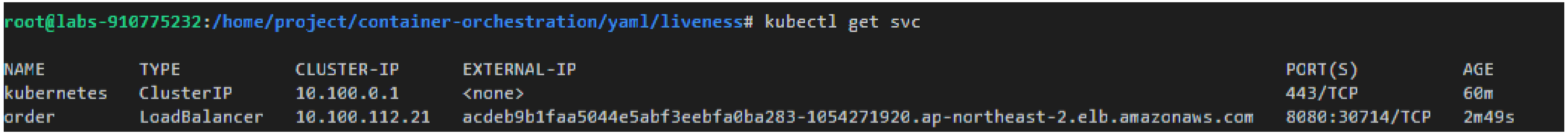
배포된 Order에 대해 서비스 생성



watch ×
Problems
root@labs-910775232: /home/project/container-orchestration/yaml/liveness ×

```
Every 1.0s: kubectl get pod                                                                   labs-910775232:

NAME                     READY   STATUS             RESTARTS   AGE
liveness-exec            0/1     CrashLoopBackOff   5          8m19s
order-5f88dff96d-rmws4   1/1     Running            0          5m13s
```

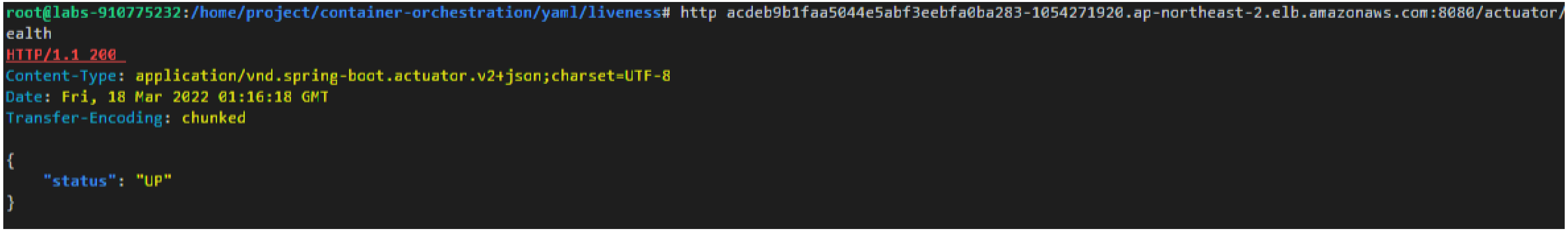
```
root@labs-910775232:/home/project/container-orchestration/yaml/liveness# kubectl expose deploy order --type=LoadBalancer --port=8080
service/order exposed
root@labs-910775232:/home/project/container-orchestration/yaml/liveness# kubectl get svc
NAME         TYPE           CLUSTER-IP      EXTERNAL-IP                                                                   PORT(S)          AGE
kubernetes   ClusterIP      10.100.0.1      <none>                                                                        443/TCP          57m
order        LoadBalancer   10.100.112.21   acdeb9b1faa5044e5abf3eebfa0ba283-1054271920.ap-northeast-2.elb.amazonaws.com   8080:30714/TCP   8s
```



```
root@labs-910775232:/home/project/container-orchestration/yaml/liveness# kubectl get svc

NAME         TYPE           CLUSTER-IP      EXTERNAL-IP                                                                    PORT(S)          AGE
kubernetes   ClusterIP      10.100.0.1      <none>                                                                         443/TCP          60m
order        LoadBalancer   10.100.112.21   acdeb9b1faa5044e5abf3eebfa0ba283-1054271920.ap-northeast-2.elb.amazonaws.com   8080:30714/TCP   2m49s
```

# Liveness Probe 확인

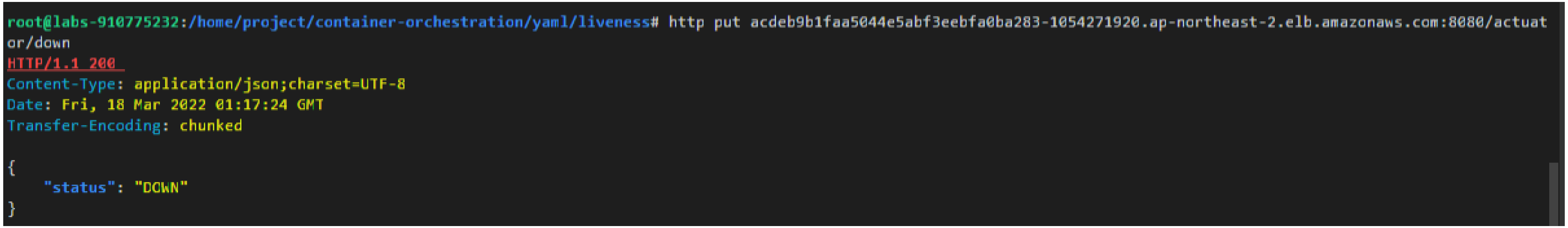
http EXTERNAL-IP:8080/actuator/health



```
root@labs-910775232:/home/project/container-orchestration/yaml/liveness# http acdeb9b1faa5044e5abf3eebfa0ba283-1054271920.ap-northeast-2.elb.amazonaws.com:8080/actuator/
ealth
HTTP/1.1 200
Content-Type: application/vnd.spring-boot.actuator.v2+json;charset=UTF-8
Date: Fri, 18 Mar 2022 01:16:18 GMT
Transfer-Encoding: chunked

{
    "status": "UP"
}
```

Order liveness Probe를 명시적으로 Fail 상태로 전환

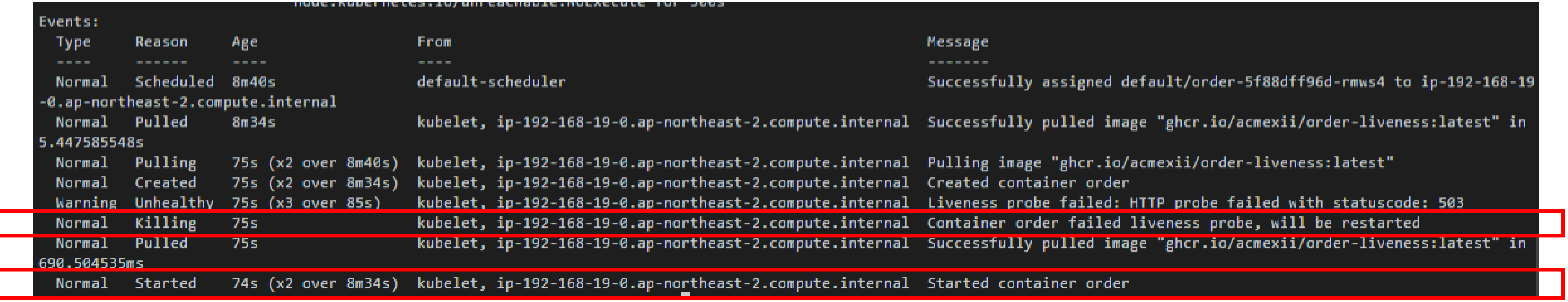
http put acdeb9b1faa5044e5abf3eebfa0ba283-1054271920.ap-northeast-2.elb.amazonaws.com:8080/actuator/down



```
root@labs-910775232:/home/project/container-orchestration/yaml/liveness# http put acdeb9b1faa5044e5abf3eebfa0ba283-1054271920.ap-northeast-2.elb.amazonaws.com:8080/actuat
or/down
HTTP/1.1 200
Content-Type: application/json;charset=UTF-8
Date: Fri, 18 Mar 2022 01:17:24 GMT
Transfer-Encoding: chunked

{
    "status": "DOWN"
}
```

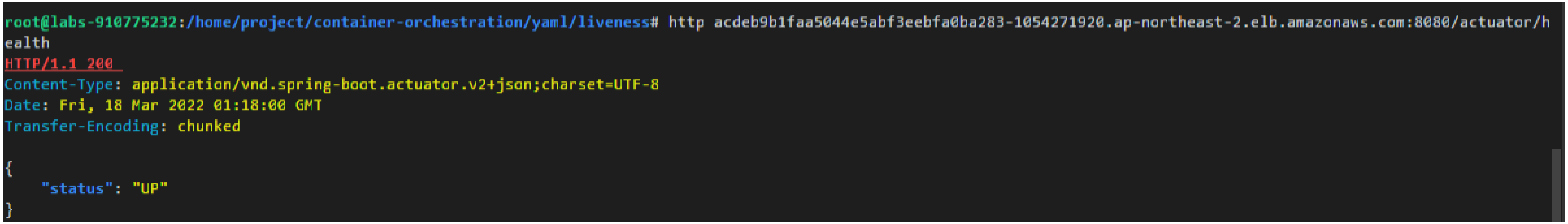
Probe Fail 상태에 따른 쿠버네티스 동작 확인

kubectl describe pod/order-5f88dff96d-rmws4



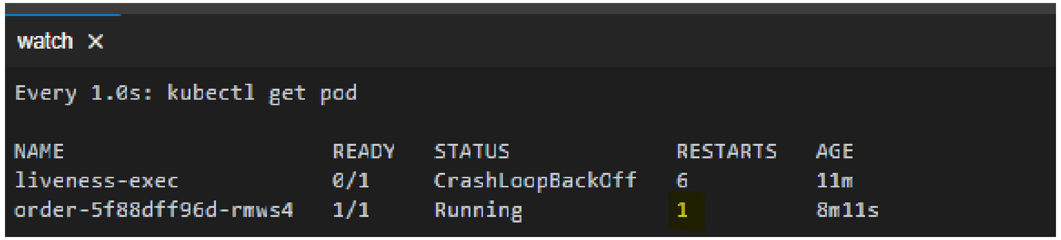
```
                                   node.kubernetes.io/unreachable:NoExecute for 300s
Events:
  Type     Reason     Age                   From                                                 Message
  ----     ------     ----                  ----                                                 -------
  Normal   Scheduled  8m40s                 default-scheduler                                    Successfully assigned default/order-5f88dff96d-rmws4 to ip-192-168-19
-0.ap-northeast-2.compute.internal
  Normal   Pulled     8m34s                 kubelet, ip-192-168-19-0.ap-northeast-2.compute.internal  Successfully pulled image "ghcr.io/acmexii/order-liveness:latest" in
5.447585548s
  Normal   Pulling    75s (x2 over 8m40s)   kubelet, ip-192-168-19-0.ap-northeast-2.compute.internal  Pulling image "ghcr.io/acmexii/order-liveness:latest"
  Normal   Created    75s (x2 over 8m34s)   kubelet, ip-192-168-19-0.ap-northeast-2.compute.internal  Created container order
  Warning  Unhealthy  75s (x3 over 85s)     kubelet, ip-192-168-19-0.ap-northeast-2.compute.internal  Liveness probe failed: HTTP probe failed with statuscode: 503
  Normal   Killing    75s                   kubelet, ip-192-168-19-0.ap-northeast-2.compute.internal  Container order failed liveness probe, will be restarted
  Normal   Pulled     75s                   kubelet, ip-192-168-19-0.ap-northeast-2.compute.internal  Successfully pulled image "ghcr.io/acmexii/order-liveness:latest" in
690.504535ms
  Normal   Started    74s (x2 over 8m34s)   kubelet, ip-192-168-19-0.ap-northeast-2.compute.internal  Started container order
```

http EXTERNAL-IP:8080/actuator/healt



```
root@labs-910775232:/home/project/container-orchestration/yaml/liveness# http acdeb9b1faa5044e5abf3eebfa0ba283-1054271920.ap-northeast-2.elb.amazonaws.com:8080/actuator/h
ealth
HTTP/1.1 200
Content-Type: application/vnd.spring-boot.actuator.v2+json;charset=UTF-8
Date: Fri, 18 Mar 2022 01:18:00 GMT
Transfer-Encoding: chunked

{
    "status": "UP"
}
```

Probe Fail 상태에 따른 쿠버네티스 동작 – 재시작하여 다시 Running 상태 유지

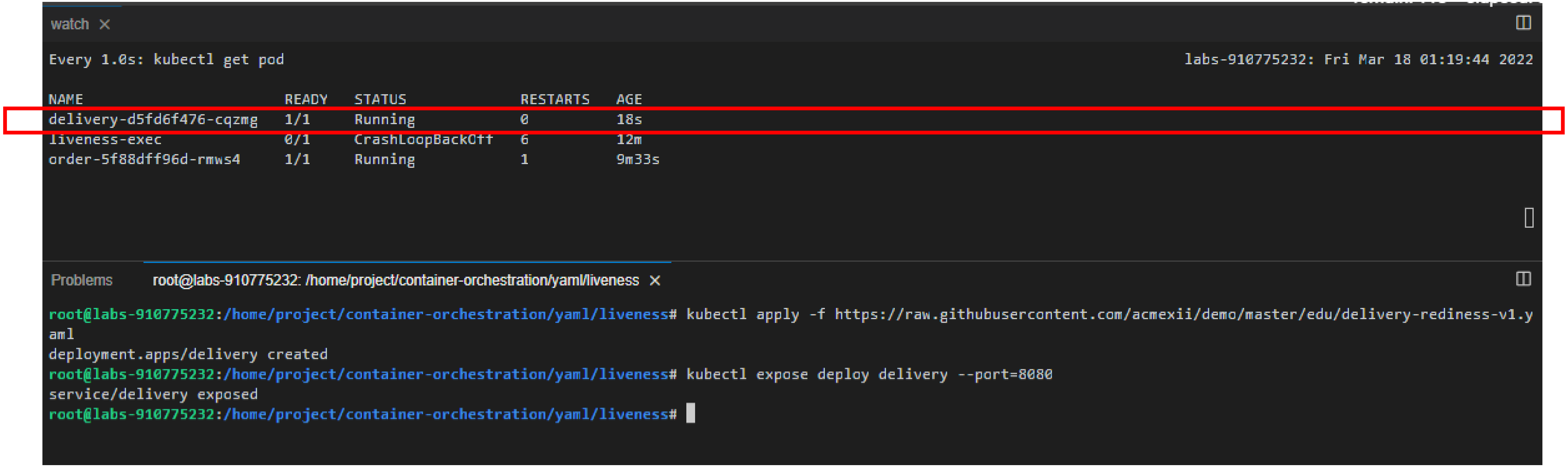


```
Every 1.0s: kubectl get pod

NAME                    READY   STATUS             RESTARTS   AGE
liveness-exec           0/1     CrashLoopBackOff   6          11m
order-5f88dff96d-rmws4  1/1     Running            1          8m11s
```

Readiness- HTTP Probe

배송 마이크로서비스를 배포한다.

```
kubectl apply -f https://raw.githubusercontent.com/acmexii/demo/master/edu/delivery-rediness-v1.yaml
kubectl expose deploy delivery --port=8080
```

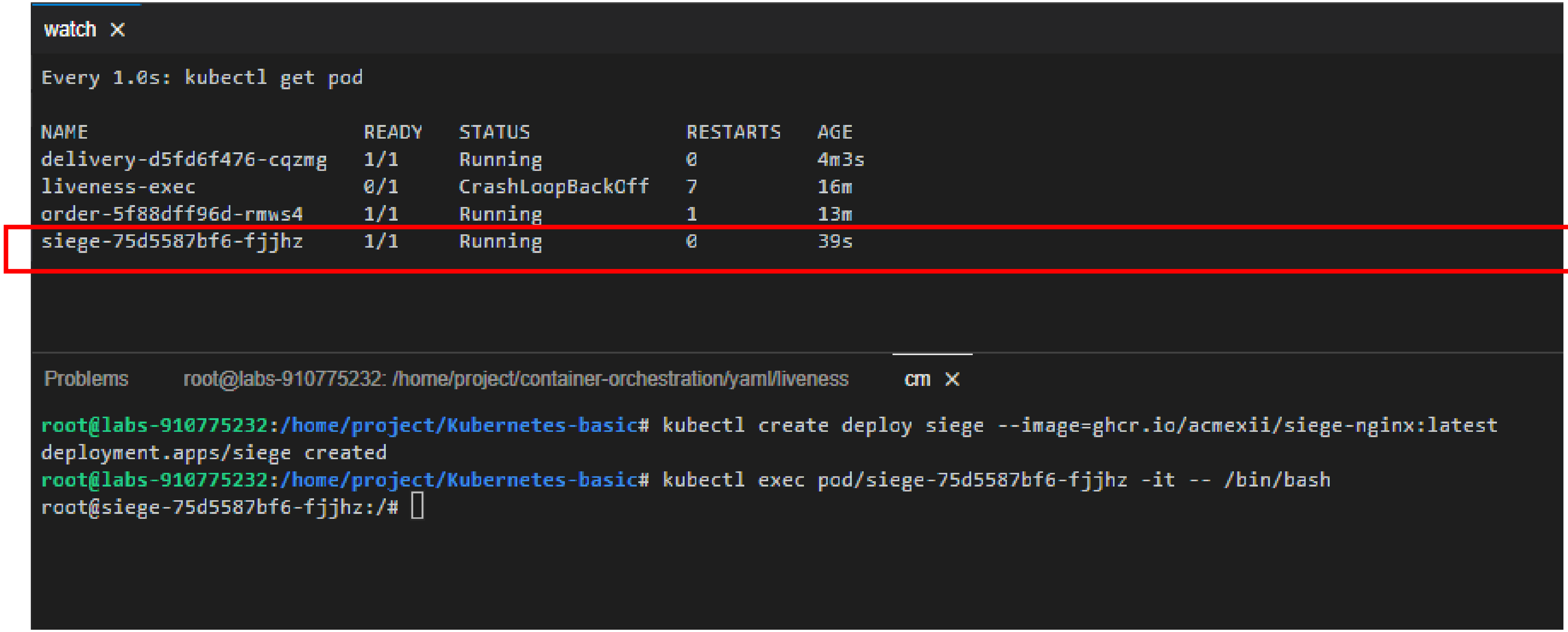


watch ×
Problems
root@labs-910775232: /home/project/container-orchestration/yaml/liveness ×

```
Every 1.0s: kubectl get pod                                          labs-910775232: Fri Mar 18 01:19:44 2022

NAME                         READY   STATUS             RESTARTS   AGE
delivery-d5fd6f476-cqzmg     1/1     Running            0          18s
liveness-exec                0/1     CrashLoopBackOff   6          12m
order-5f88dff96d-rmws4       1/1     Running            1          9m33s
```

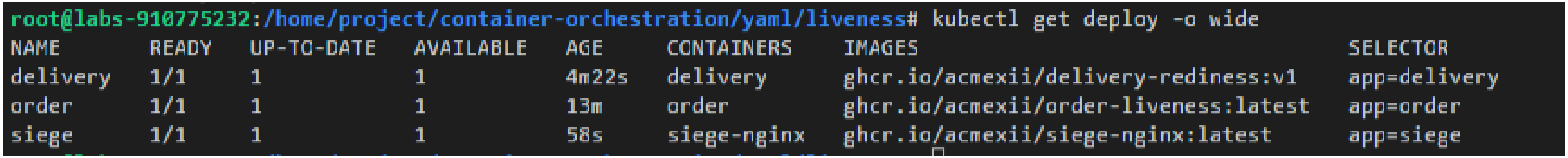
```
root@labs-910775232:/home/project/container-orchestration/yaml/liveness# kubectl apply -f https://raw.githubusercontent.com/acmexii/demo/master/edu/delivery-rediness-v1.y
aml
deployment.apps/delivery created
root@labs-910775232:/home/project/container-orchestration/yaml/liveness# kubectl expose deploy delivery --port=8080
service/delivery exposed
root@labs-910775232:/home/project/container-orchestration/yaml/liveness#
```

Siege (로더제너레이터)를 설치하고 해당 컨테이너로 접속 – 부하 테스트를 위함


watch ×
Problems
root@labs-910775232: /home/project/container-orchestration/yaml/liveness
cm ×

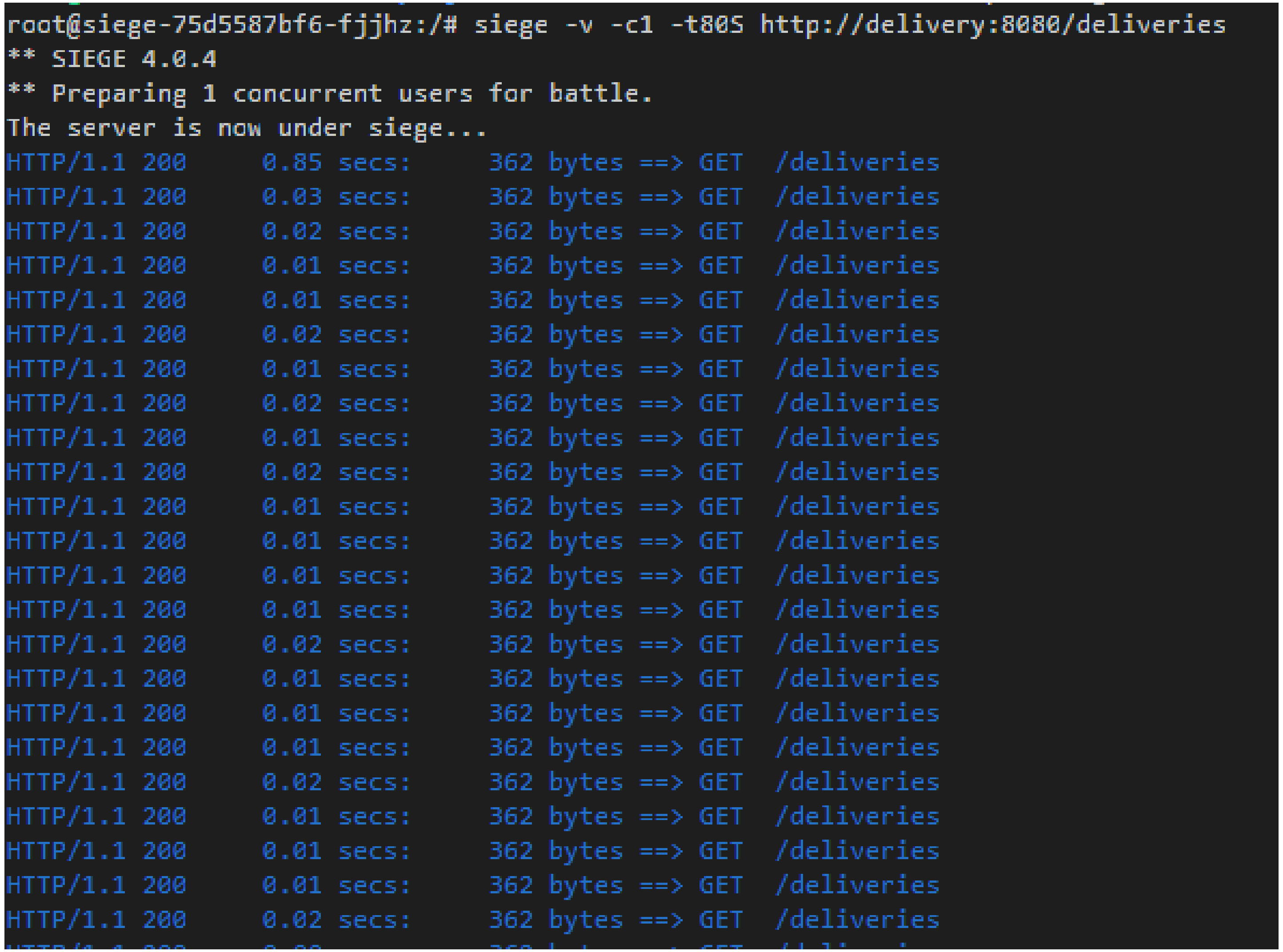
```
Every 1.0s: kubectl get pod

NAME                        READY   STATUS             RESTARTS   AGE
delivery-d5fd6f476-cqzmg    1/1     Running            0          4m3s
liveness-exec               0/1     CrashLoopBackOff   7          16m
order-5f88dff96d-rmws4      1/1     Running            1          13m
siege-75d5587bf6-fjjhz      1/1     Running            0          39s
```

```
root@labs-910775232:/home/project/Kubernetes-basic# kubectl create deploy siege --image=ghcr.io/acmexii/siege-nginx:latest
deployment.apps/siege created
root@labs-910775232:/home/project/Kubernetes-basic# kubectl exec pod/siege-75d5587bf6-fjjhz -it -- /bin/bash
root@siege-75d5587bf6-fjjhz:/# 
```



```
root@labs-910775232:/home/project/container-orchestration/yaml/liveness# kubectl get deploy -o wide
NAME       READY   UP-TO-DATE   AVAILABLE   AGE     CONTAINERS    IMAGES                                   SELECTOR
delivery   1/1     1            1           4m22s   delivery      ghcr.io/acmexii/delivery-rediness:v1     app=delivery
order      1/1     1            1           13m     order         ghcr.io/acmexii/order-liveness:latest    app=order
siege      1/1     1            1           58s     siege-nginx   ghcr.io/acmexii/siege-nginx:latest       app=siege
```

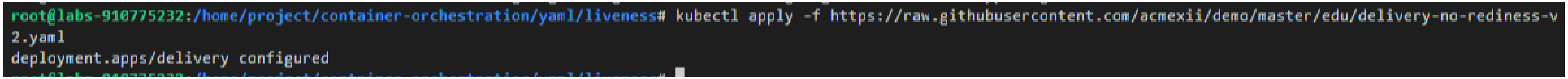
Delivery 서비스에 대해 Siege로 부하 발생



```
root@siege-75d5587bf6-fjjhz:/# siege -v -c1 -t80S http://delivery:8080/deliveries
** SIEGE 4.0.4
** Preparing 1 concurrent users for battle.
The server is now under siege...
HTTP/1.1 200     0.85 secs:     362 bytes ==> GET  /deliveries
HTTP/1.1 200     0.03 secs:     362 bytes ==> GET  /deliveries
HTTP/1.1 200     0.02 secs:     362 bytes ==> GET  /deliveries
HTTP/1.1 200     0.01 secs:     362 bytes ==> GET  /deliveries
HTTP/1.1 200     0.01 secs:     362 bytes ==> GET  /deliveries
HTTP/1.1 200     0.02 secs:     362 bytes ==> GET  /deliveries
HTTP/1.1 200     0.01 secs:     362 bytes ==> GET  /deliveries
HTTP/1.1 200     0.02 secs:     362 bytes ==> GET  /deliveries
HTTP/1.1 200     0.01 secs:     362 bytes ==> GET  /deliveries
HTTP/1.1 200     0.02 secs:     362 bytes ==> GET  /deliveries
HTTP/1.1 200     0.01 secs:     362 bytes ==> GET  /deliveries
HTTP/1.1 200     0.01 secs:     362 bytes ==> GET  /deliveries
HTTP/1.1 200     0.01 secs:     362 bytes ==> GET  /deliveries
HTTP/1.1 200     0.01 secs:     362 bytes ==> GET  /deliveries
HTTP/1.1 200     0.02 secs:     362 bytes ==> GET  /deliveries
HTTP/1.1 200     0.01 secs:     362 bytes ==> GET  /deliveries
HTTP/1.1 200     0.01 secs:     362 bytes ==> GET  /deliveries
HTTP/1.1 200     0.01 secs:     362 bytes ==> GET  /deliveries
HTTP/1.1 200     0.02 secs:     362 bytes ==> GET  /deliveries
HTTP/1.1 200     0.01 secs:     362 bytes ==> GET  /deliveries
HTTP/1.1 200     0.01 secs:     362 bytes ==> GET  /deliveries
HTTP/1.1 200     0.01 secs:     362 bytes ==> GET  /deliveries
HTTP/1.1 200     0.02 secs:     362 bytes ==> GET  /deliveries
```

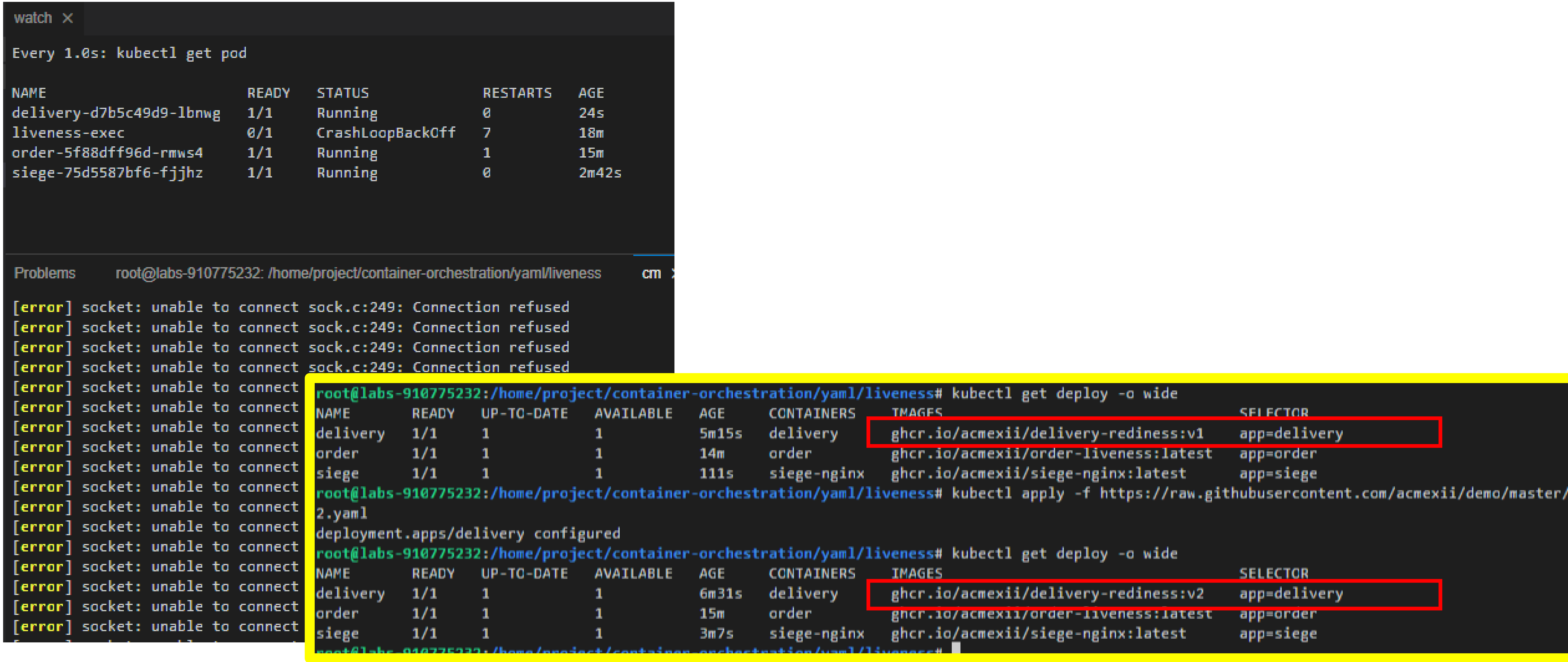
80초 동안 부하 발생

Siege로 부하가 걸린 상태에서 Delivery 서비스를 v2로 Rollout



```
root@labs-910775232:/home/project/container-orchestration/yaml/liveness# kubectl apply -f https://raw.githubusercontent.com/acmexii/demo/master/edu/delivery-no-rediness-v
2.yaml
deployment.apps/delivery configured
root@labs-910775232:/home/project/container-orchestration/yaml/liveness#
```

신규 버전으로 Rollout되는 과정에서 네트워크 장애 발생



watch ×
Problems
root@labs-910775232: /home/project/container-orchestration/yaml/liveness
cm

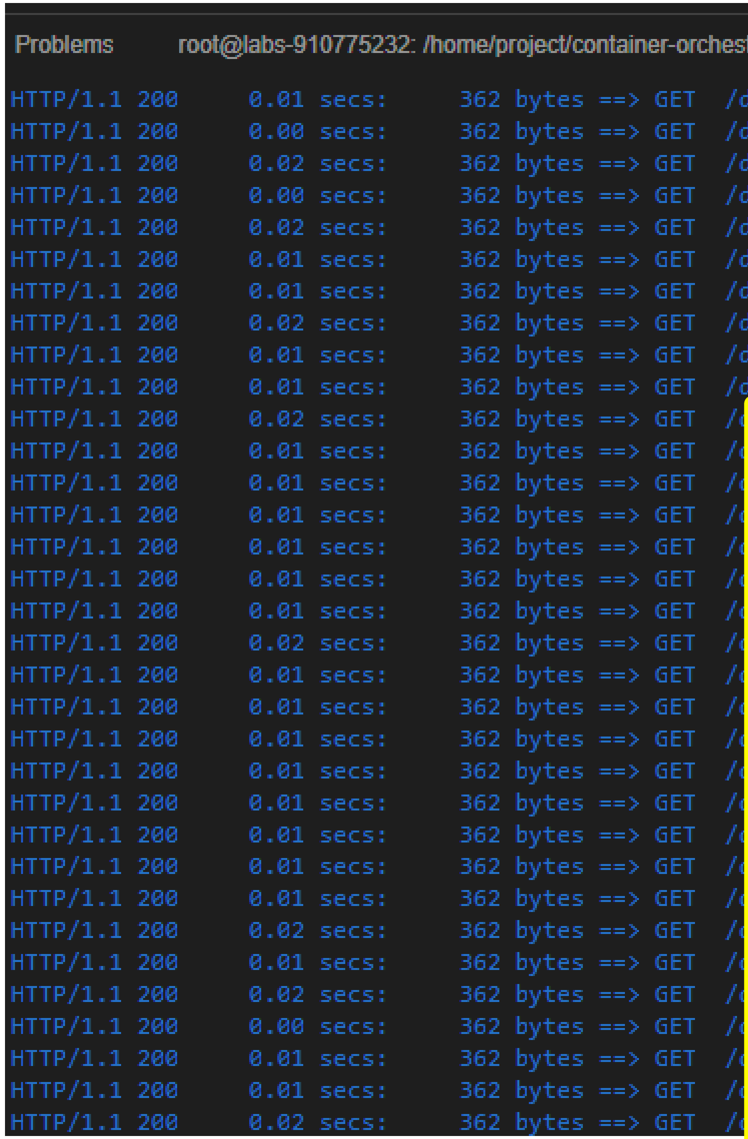
```
Every 1.0s: kubectl get pod

NAME                        READY   STATUS             RESTARTS   AGE
delivery-d7b5c49d9-lbnwg    1/1     Running            0          24s
liveness-exec               0/1     CrashLoopBackOff   7          18m
order-5f88dff96d-rmws4      1/1     Running            1          15m
siege-75d5587bf6-fjjhz      1/1     Running            0          2m42s
```

```
[error] socket: unable to connect sock.c:249: Connection refused
[error] socket: unable to connect sock.c:249: Connection refused
[error] socket: unable to connect sock.c:249: Connection refused
[error] socket: unable to connect sock.c:249: Connection refused
[error] socket: unable to connect
[error] socket: unable to connect
[error] socket: unable to connect
[error] socket: unable to connect
[error] socket: unable to connect
[error] socket: unable to connect
[error] socket: unable to connect
[error] socket: unable to connect
[error] socket: unable to connect
[error] socket: unable to connect
[error] socket: unable to connect
[error] socket: unable to connect
[error] socket: unable to connect
```

```
root@labs-910775232:/home/project/container-orchestration/yaml/liveness# kubectl get deploy -o wide
NAME       READY   UP-TO-DATE   AVAILABLE   AGE     CONTAINERS     IMAGES                                  SELECTOR
delivery   1/1     1            1           5m15s   delivery       ghcr.io/acmexii/delivery-rediness:v1    app=delivery
order      1/1     1            1           14m     order          ghcr.io/acmexii/order-liveness:latest   app=order
siege      1/1     1            1           111s    siege-nginx    ghcr.io/acmexii/siege-nginx:latest      app=siege
root@labs-910775232:/home/project/container-orchestration/yaml/liveness# kubectl apply -f https://raw.githubusercontent.com/acmexii/demo/master/
2.yaml
deployment.apps/delivery configured
root@labs-910775232:/home/project/container-orchestration/yaml/liveness# kubectl get deploy -o wide
NAME       READY   UP-TO-DATE   AVAILABLE   AGE     CONTAINERS     IMAGES                                  SELECTOR
delivery   1/1     1            1           6m31s   delivery       ghcr.io/acmexii/delivery-rediness:v2    app=delivery
order      1/1     1            1           15m     order          ghcr.io/acmexii/order-liveness:latest   app=order
siege      1/1     1            1           3m7s    siege-nginx    ghcr.io/acmexii/siege-nginx:latest      app=siege
root@labs-910775232:/home/project/container-orchestration/yaml/liveness#
```

Siege로 부하가 걸린 상태에서 Delivery 서비스를 v3로 Rollout

siege -v -c1 -t60S http://delivery:8080/deliveries

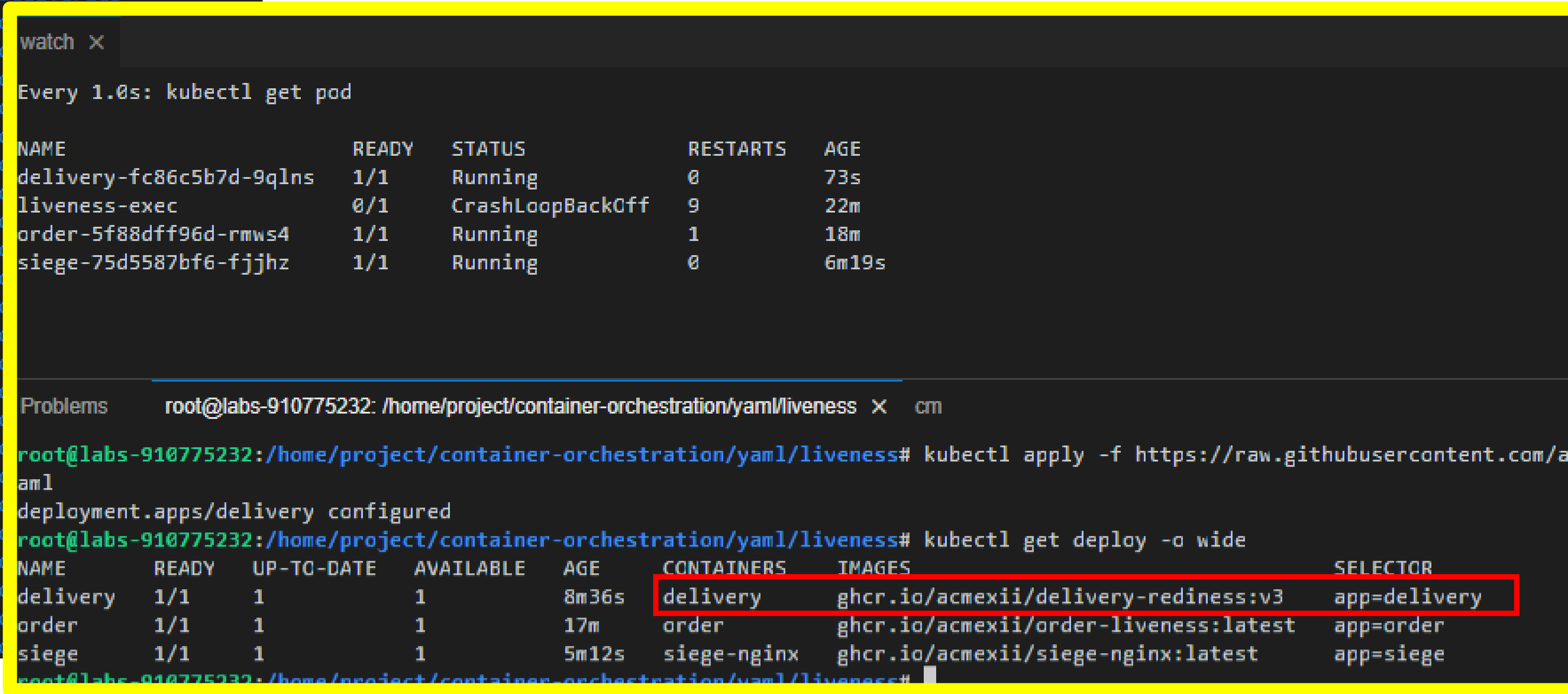


Problems
root@labs-910775232: /home/project/container-orchestration/yaml/liveness

```
HTTP/1.1 200     0.01 secs:     362 bytes ==> GET  /deliveries
HTTP/1.1 200     0.00 secs:     362 bytes ==> GET  /deliveries
HTTP/1.1 200     0.02 secs:     362 bytes ==> GET  /deliveries
HTTP/1.1 200     0.00 secs:     362 bytes ==> GET  /deliveries
HTTP/1.1 200     0.02 secs:     362 bytes ==> GET  /deliveries
HTTP/1.1 200     0.01 secs:     362 bytes ==> GET  /deliveries
HTTP/1.1 200     0.01 secs:     362 bytes ==> GET  /deliveries
HTTP/1.1 200     0.02 secs:     362 bytes ==> GET  /deliveries
HTTP/1.1 200     0.01 secs:     362 bytes ==> GET  /deliveries
HTTP/1.1 200     0.01 secs:     362 bytes ==> GET  /deliveries
HTTP/1.1 200     0.02 secs:     362 bytes ==> GET  /
HTTP/1.1 200     0.01 secs:     362 bytes ==> GET  /
HTTP/1.1 200     0.01 secs:     362 bytes ==> GET  /
HTTP/1.1 200     0.01 secs:     362 bytes ==> GET  /
HTTP/1.1 200     0.01 secs:     362 bytes ==> GET  /
HTTP/1.1 200     0.01 secs:     362 bytes ==> GET  /
HTTP/1.1 200     0.01 secs:     362 bytes ==> GET  /
HTTP/1.1 200     0.02 secs:     362 bytes ==> GET  /
HTTP/1.1 200     0.01 secs:     362 bytes ==> GET  /
HTTP/1.1 200     0.01 secs:     362 bytes ==> GET  /
HTTP/1.1 200     0.01 secs:     362 bytes ==> GET  /
HTTP/1.1 200     0.01 secs:     362 bytes ==> GET  /
HTTP/1.1 200     0.01 secs:     362 bytes ==> GET  /
HTTP/1.1 200     0.01 secs:     362 bytes ==> GET  /
HTTP/1.1 200     0.01 secs:     362 bytes ==> GET  /
HTTP/1.1 200     0.01 secs:     362 bytes ==> GET  /
HTTP/1.1 200     0.02 secs:     362 bytes ==> GET  /
HTTP/1.1 200     0.01 secs:     362 bytes ==> GET  /
HTTP/1.1 200     0.02 secs:     362 bytes ==> GET  /
HTTP/1.1 200     0.00 secs:     362 bytes ==> GET  /
HTTP/1.1 200     0.01 secs:     362 bytes ==> GET  /
HTTP/1.1 200     0.01 secs:     362 bytes ==> GET  /
HTTP/1.1 200     0.02 secs:     362 bytes ==> GET  /
```

부하가 있는 상태에서 Readiness 설정 되어있는 새버전 배포

kubectl apply -f https://raw.githubusercontent.com/acmexii/demo/master/edu/delivery-rediness-v3.yaml

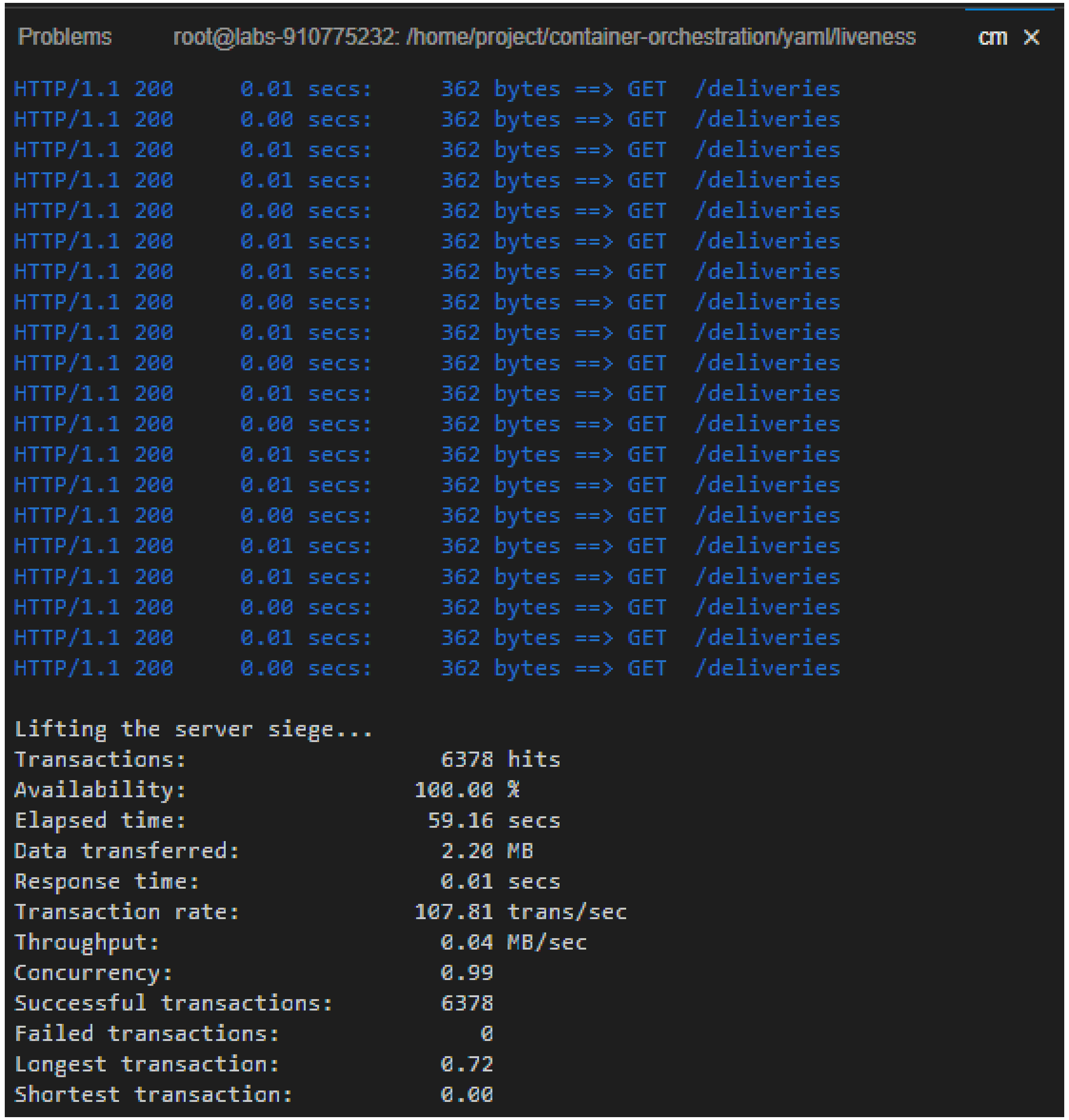


```
Every 1.0s: kubectl get pod

NAME                        READY   STATUS             RESTARTS   AGE
delivery-fc86c5b7d-9qlns    1/1     Running            0          73s
liveness-exec               0/1     CrashLoopBackOff   9          22m
order-5f88dff96d-rmws4      1/1     Running            1          18m
siege-75d5587bf6-fjjhz      1/1     Running            0          6m19s
```

Problems
root@labs-910775232: /home/project/container-orchestration/yaml/liveness ×
cm

```
root@labs-910775232:/home/project/container-orchestration/yaml/liveness# kubectl apply -f https://raw.githubusercontent.com/a
aml
deployment.apps/delivery configured
root@labs-910775232:/home/project/container-orchestration/yaml/liveness# kubectl get deploy -o wide
NAME       READY   UP-TO-DATE   AVAILABLE   AGE     CONTAINERS     IMAGES                                  SELECTOR
delivery   1/1     1            1           8m36s   delivery       ghcr.io/acmexii/delivery-rediness:v3    app=delivery
order      1/1     1            1           17m     order          ghcr.io/acmexii/order-liveness:latest   app=order
siege      1/1     1            1           5m12s   siege-nginx    ghcr.io/acmexii/siege-nginx:latest      app=siege
root@labs-910775232:/home/project/container-orchestration/yaml/liveness#
```

네트워크 장애없이 배포 완료 (Availability 100 % 보장)



Problems
root@labs-910775232: /home/project/container-orchestration/yaml/liveness
cm ✕

```
HTTP/1.1 200     0.01 secs:     362 bytes ==> GET  /deliveries
HTTP/1.1 200     0.00 secs:     362 bytes ==> GET  /deliveries
HTTP/1.1 200     0.01 secs:     362 bytes ==> GET  /deliveries
HTTP/1.1 200     0.01 secs:     362 bytes ==> GET  /deliveries
HTTP/1.1 200     0.00 secs:     362 bytes ==> GET  /deliveries
HTTP/1.1 200     0.01 secs:     362 bytes ==> GET  /deliveries
HTTP/1.1 200     0.01 secs:     362 bytes ==> GET  /deliveries
HTTP/1.1 200     0.00 secs:     362 bytes ==> GET  /deliveries
HTTP/1.1 200     0.01 secs:     362 bytes ==> GET  /deliveries
HTTP/1.1 200     0.00 secs:     362 bytes ==> GET  /deliveries
HTTP/1.1 200     0.01 secs:     362 bytes ==> GET  /deliveries
HTTP/1.1 200     0.00 secs:     362 bytes ==> GET  /deliveries
HTTP/1.1 200     0.01 secs:     362 bytes ==> GET  /deliveries
HTTP/1.1 200     0.01 secs:     362 bytes ==> GET  /deliveries
HTTP/1.1 200     0.00 secs:     362 bytes ==> GET  /deliveries
HTTP/1.1 200     0.01 secs:     362 bytes ==> GET  /deliveries
HTTP/1.1 200     0.01 secs:     362 bytes ==> GET  /deliveries
HTTP/1.1 200     0.00 secs:     362 bytes ==> GET  /deliveries
HTTP/1.1 200     0.01 secs:     362 bytes ==> GET  /deliveries
HTTP/1.1 200     0.00 secs:     362 bytes ==> GET  /deliveries

Lifting the server siege...
Transactions:                   6378 hits
Availability:                 100.00 %
Elapsed time:                  59.16 secs
Data transferred:               2.20 MB
Response time:                  0.01 secs
Transaction rate:             107.81 trans/sec
Throughput:                     0.04 MB/sec
Concurrency:                    0.99
Successful transactions:        6378
Failed transactions:               0
Longest transaction:            0.72
Shortest transaction:           0.00
```