Subject: AKS Pods CrashLoopBackOff After Enabling Load Balancer and HPA for WebUI Service

Description:

We are encountering an issue with our AKS workload where all pods for the WebUI service enter a CrashLoopBackOff state after configuring Horizontal Pod Autoscaler and LoadBalancer service.

Current Configuration:

1. Deployment (webui-deployment.yaml):

- replicas: `2 (min)`
- Image: `acramsacrdev.azurecr.io/ams/dev/webui:9164`
- Container port: `8080`
- Resource requests/limits:
    - CPU: `250m/500m`
    - Memory: `512Mi/1Gi`

2. Service (webui-service.yaml):

- type: LoadBalancer
- Exposes port `80`, maps to container targetPort: `8080`

3. Ingress (webui-ingress.yaml):

- Host: web.enhancehealth123.com
- TLS configured via `letsencrypt-prod`

4. HPA Setup:

- Min pods: 2
- Max pods: 5

Issue Observed:

As soon as the deployment scales up (due to HPA or manual replica increase), the newly created pods crash with the following error:

Pod State:

```
State:       Waiting

Reason:      CrashLoopBackOff

Exit Code:    139

Restart Count:   291
```

Describe Output:

```
Warning  BackOff  kubelet Back-off restarting failed container webui
```

Additional Info:

- Crash occurs consistently on all pods after scaling.
- Pod image webui:9164 is available and pulled successfully.
- The environment variables ASPNETCORE_ENVIRONMENT and env are set correctly.
- Logs indicate Exit Code 139, which typically relates to a segmentation fault or memory access violation.

Request:

We need assistance in determining:

- Root cause of the CrashLoopBackOff with Exit Code 139.
- Whether the issue is due to resource limits, image issue, or internal network/misconfiguration between service and ingress.
- Any recommendations to stabilize the pod and safely scale under HPA and LoadBalancer settings.

Files:

Deployment.yml

```
apiVersion: apps/v1
```

```
kind: Deployment
metadata:
  name: webui
  namespace: webui
spec:
  replicas: 2  # 👈 Start with min replicas
  selector:
    matchLabels:
      app: webui
  template:
    metadata:
      labels:
        app: webui
    spec:
      containers:
        - name: webui
          image: acramsacrdev.azurecr.io/ams/dev/webui:9164
          ports:
            - containerPort: 8080
          env:
            - name: ASPNETCORE_ENVIRONMENT
              value: Development
            - name: env
              value: Dev
          resources:
            requests:
              cpu: "250m"
              memory: "512Mi"
            limits:
              cpu: "500m"
              memory: "1Gi"
```

Service.yml

```
apiVersion: v1
```

```
kind: Service
metadata:
  name: webui-service
  namespace: webui
spec:
  type: LoadBalancer  # This is the key change
  selector:
    app: webui
  ports:
    - protocol: TCP
      port: 80        # Port exposed by LoadBalancer
      targetPort: 8080  # Port exposed by the container
```

Ingress.yml

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: webui-ingress
  namespace: webui
  annotations:
    kubernetes.io/ingress.class: nginx
    cert-manager.io/cluster-issuer: letsencrypt-prod
spec:
  ingressClassName: nginx
  tls:
    - hosts:
        - web.enhancehealth123.com
      secretName: enhancehealth-tls
  rules:
    - host: web.enhancehealth123.com
      http:
        paths:
          - path: /
            pathType: Prefix
            backend:
              service:
                name: webui-service
                port:
                  number: 8080
```