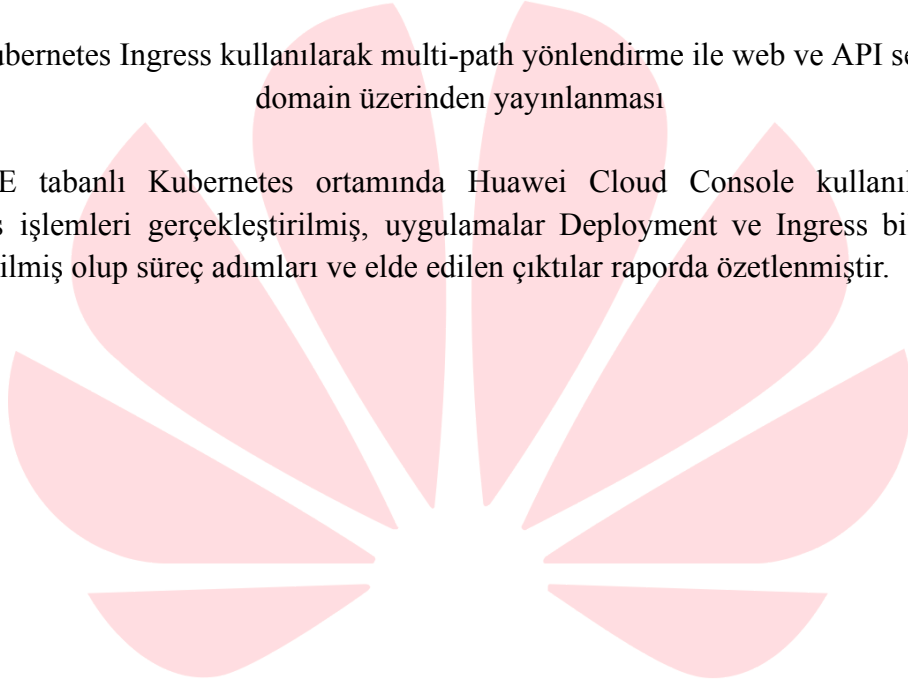


# Huawei Kubernetes CCE Bootcamp

## Final Projesi Raporu

**Konu:** Kubernetes Ingress kullanılarak multi-path yönlendirme ile web ve API servislerinin domain üzerinden yayınlanması

**Özet:** CCE tabanlı Kubernetes ortamında Huawei Cloud Console kullanılarak temel Kubernetes işlemleri gerçekleştirilmiş, uygulamalar Deployment ve Ingress bileşenleri ile orkestre edilmiş olup süreç adımları ve elde edilen çıktılar raporda özetlenmiştir.



**Hazırlayan**  
Mustafa ESEN

**2025-ANKARA**

## 1)Cluster ve Ağ Altyapısı Kurulumu

Cluster, node yönetiminin otomatik sağlanması amacıyla **Autopilot** modda oluşturulmuştur. Gerekli **VPC ve subnet** yapıları tanımlanmış ve cluster'a atanmıştır. Uygulamaların dış dünyaya erişimi ve image pull işlemlerinin sorunsuz gerçekleşmesi için **SNAT** özelliği aktif edilmiştir. Demo kapsamında ihtiyaç duyulmadığından **Alarm Service** yapılandırması yapılmamıştır.

## 2)Add-on ve Servis Yapılandırmaları

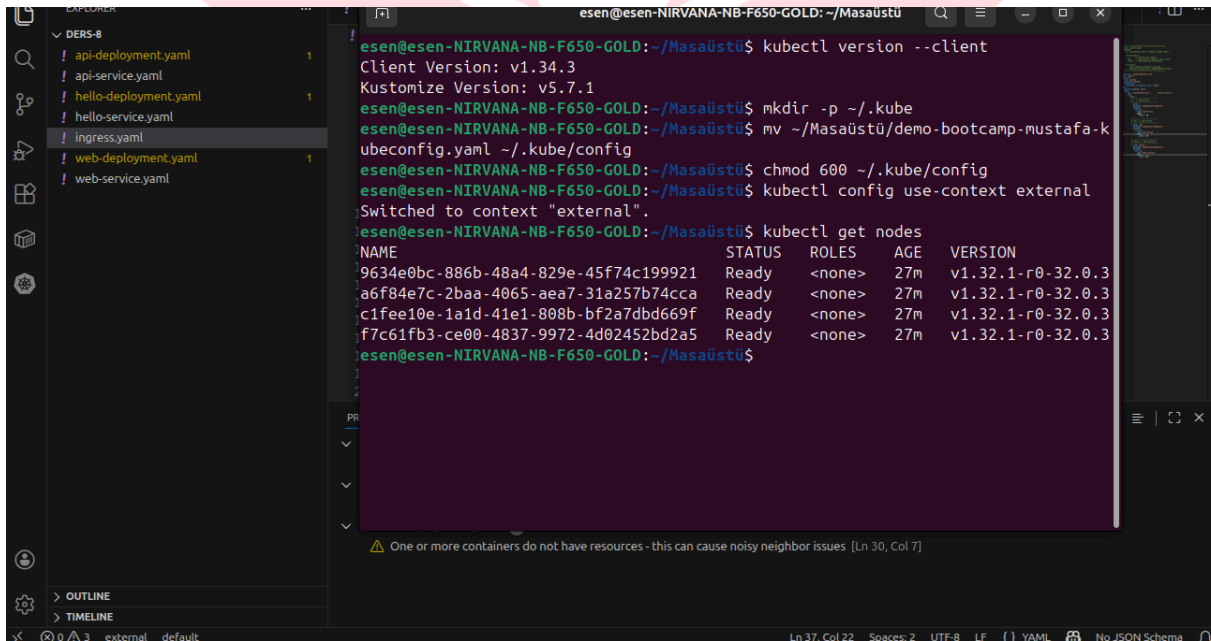
Demo senaryosunda gereksiz yük oluşturmaması adına **Monitoring ve Log Collection** add-on'ları devre dışı bırakılmıştır. Temel Kubernetes bileşenlerine odaklanılarak sade bir ortam oluşturulmuştur.

## 3)Uygulama Deployment ve Service Tanımlamaları

Uygulama ayağa kaldırma sürecinde **API, Web ve Hello** bileşenleri için ayrı ayrı **Deployment ve Service** tanımlamaları hazırlanmıştır. İlgili YAML dosyaları kullanılarak uygulamalar cluster üzerinde başarıyla deploy edilmiş ve servisler aracılığıyla erişilebilir hale getirilmiştir.

## 4)Kubernetes Cluster'a CLI Üzerinden Bağlantı ve Ortam Hazırlığı

Yerel ortamda **kubectl** kurulumu doğrulanmış, Huawei Cloud CCE tarafından sağlanan kubeconfig dosyası kullanılarak cluster context'i tanımlanmış ve aktif hale getirilmiştir. Bağlantı sonrası node'lar listelenerek cluster'ın erişilebilir ve çalışır durumda olduğu teyit edilmiştir.



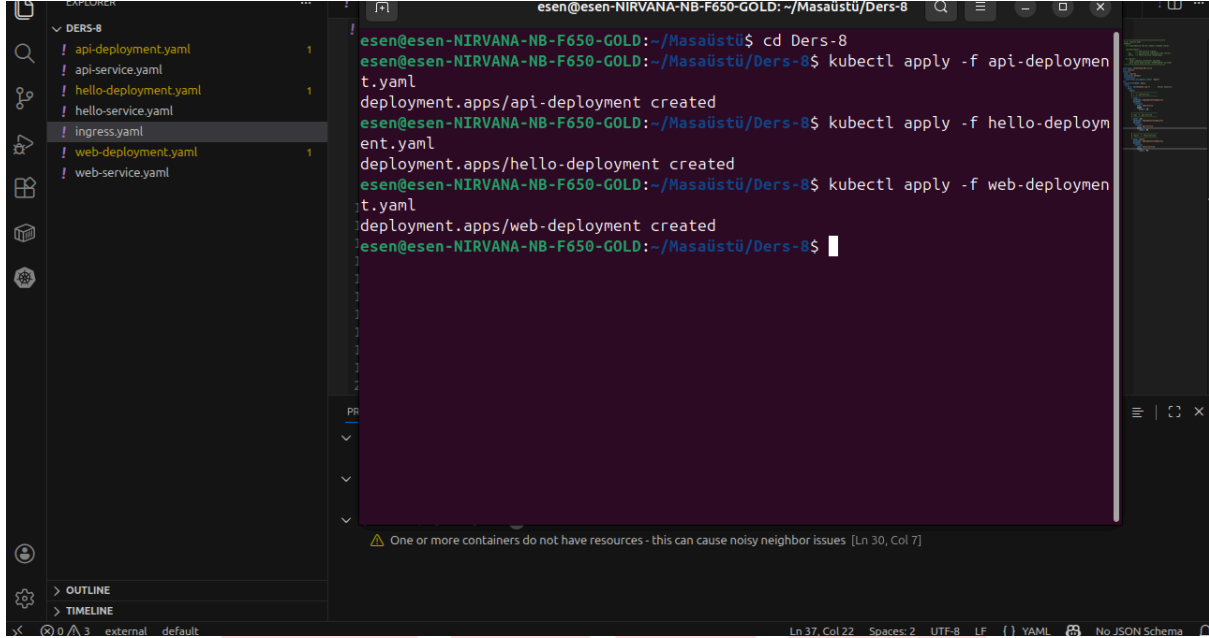
```
esen@esen-NIRVANA-NB-F650-GOLD: ~/Masaüstü$ kubectl version --client
Client Version: v1.34.3
Kustomize Version: v5.7.1
esen@esen-NIRVANA-NB-F650-GOLD: ~/Masaüstü$ mkdir -p ~/.kube
esen@esen-NIRVANA-NB-F650-GOLD: ~/Masaüstü$ mv ~/Masaüstü/demo-bootcamp-mustafa-kubeconfig.yaml ~/.kube/config
esen@esen-NIRVANA-NB-F650-GOLD: ~/Masaüstü$ chmod 600 ~/.kube/config
esen@esen-NIRVANA-NB-F650-GOLD: ~/Masaüstü$ kubectl config use-context external
Switched to context "external".
esen@esen-NIRVANA-NB-F650-GOLD: ~/Masaüstü$ kubectl get nodes
NAME                                STATUS    ROLES    AGE    VERSION
9634e0bc-886b-48a4-829e-45f74c199921 Ready    <none>   27m    v1.32.1-r0-32.0.3
a6f84e7c-2baa-4065-aea7-31a257b74cca Ready    <none>   27m    v1.32.1-r0-32.0.3
c1fee10e-1a1d-41e1-808b-bf2a7dbd669f Ready    <none>   27m    v1.32.1-r0-32.0.3
f7c61fb3-ce00-4837-9972-4d02452bd2a5 Ready    <none>   27m    v1.32.1-r0-32.0.3
esen@esen-NIRVANA-NB-F650-GOLD: ~/Masaüstü$
```

One or more containers do not have resources - this can cause noisy neighbor issues [Ln 30, Col 7]

(Görsel 1.0: Kubernetes bağlantısı.)

## 5)Uygulama Deployment'larının Cluster Üzerine Alınması

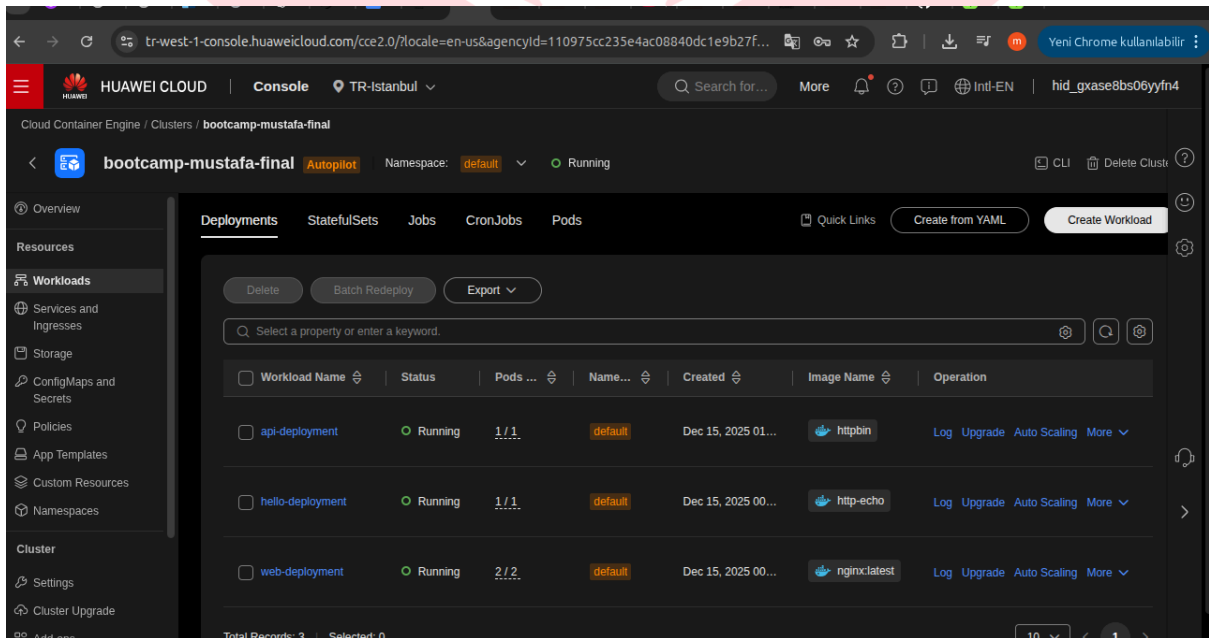
Hazırlanan YAML dosyaları kullanılarak API, Hello ve Web bileşenlerine ait Deployment tanımlamaları Kubernetes cluster üzerinde uygulanmıştır. Deployment işlemlerinin başarıyla tamamlandığı terminal çıktıları ile doğrulanmış ve uygulama pod'larının oluşturulma süreci başlatılmıştır.



```
esen@esen-NIRVANA-NB-F650-GOLD: ~/Masaüstü/Ders-8
esen@esen-NIRVANA-NB-F650-GOLD:~/Masaüstü/Ders-8$ cd Ders-8
esen@esen-NIRVANA-NB-F650-GOLD:~/Masaüstü/Ders-8$ kubectl apply -f api-deployment.yaml
deployment.apps/api-deployment created
esen@esen-NIRVANA-NB-F650-GOLD:~/Masaüstü/Ders-8$ kubectl apply -f hello-deployment.yaml
deployment.apps/hello-deployment created
esen@esen-NIRVANA-NB-F650-GOLD:~/Masaüstü/Ders-8$ kubectl apply -f web-deployment.yaml
deployment.apps/web-deployment created
esen@esen-NIRVANA-NB-F650-GOLD:~/Masaüstü/Ders-8$
```

(Görsel 1.1: Uygulama Deployment'larının Cluster'a alınması.)

Daha sonra Oluşturulan API, Hello ve Web deployment'larının Huawei Cloud CCE Console üzerinden çalışır durumda olduğu doğrulanmıştır. Pod'ların Running durumunda olduğu ve ilgili container image'lerinin başarıyla ayağa kalktığı gözlemlenmiştir.

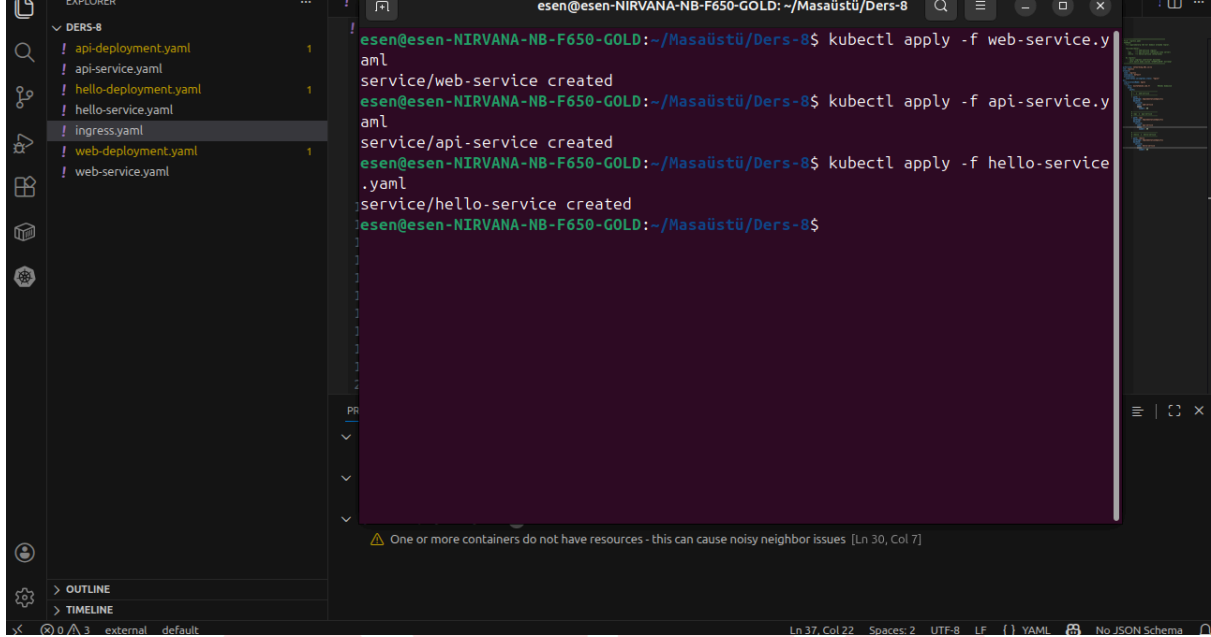


Workload Name	Status	Pods	Name	Created	Image Name	Operation
api-deployment	Running	1/1	default	Dec 15, 2025 01...	httpbin	Log Upgrade Auto Scaling More
hello-deployment	Running	1/1	default	Dec 15, 2025 00...	http-echo	Log Upgrade Auto Scaling More
web-deployment	Running	2/2	default	Dec 15, 2025 00...	nginx:latest	Log Upgrade Auto Scaling More

(Görsel 1.2: Deployment'ların console üzerinden kontrolü)

## 6)Uygulama Servislerinin Tanımlanması

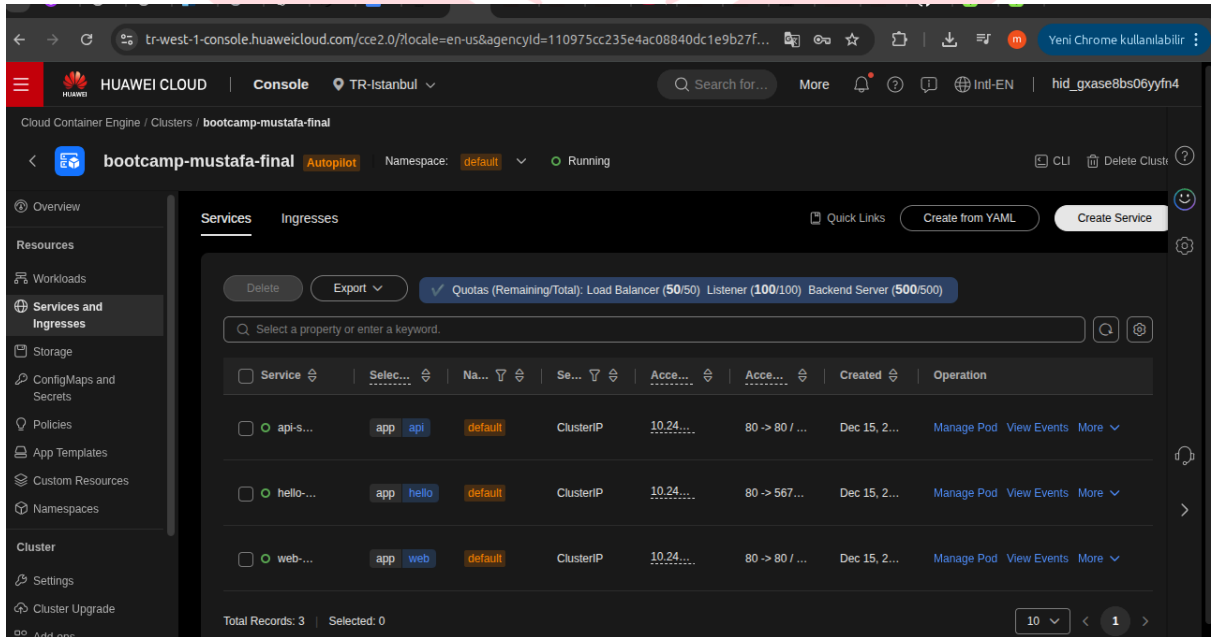
Deployment'ları oluşturulan API, Web ve Hello bileşenleri için ilgili Service tanımlamaları Kubernetes cluster üzerinde uygulanmıştır. Servisler aracılığıyla pod'lar arasında iletişim sağlanmış ve uygulama bileşenlerinin ağ üzerinden erişilebilir hale gelmesi sağlanmıştır.



```
esen@esen-NIRVANA-NB-F650-GOLD:~/Masaüstü/Ders-8$ kubectl apply -f web-service.yaml
service/web-service created
esen@esen-NIRVANA-NB-F650-GOLD:~/Masaüstü/Ders-8$ kubectl apply -f api-service.yaml
service/api-service created
esen@esen-NIRVANA-NB-F650-GOLD:~/Masaüstü/Ders-8$ kubectl apply -f hello-service.yaml
service/hello-service created
esen@esen-NIRVANA-NB-F650-GOLD:~/Masaüstü/Ders-8$
```

(Görsel 1.3: Uygulama service'lerinin tanımlanması.)

Daha sonra API, Hello ve Web uygulamalarına ait Service tanımlamalarının Huawei Cloud CCE Console üzerinden başarıyla oluşturulduğu doğrulanmıştır. Servislerin **ClusterIP** tipi ile çalıştığı ve ilgili pod'lara doğru şekilde yönlendirme yaptığı gözlemlenmiştir.

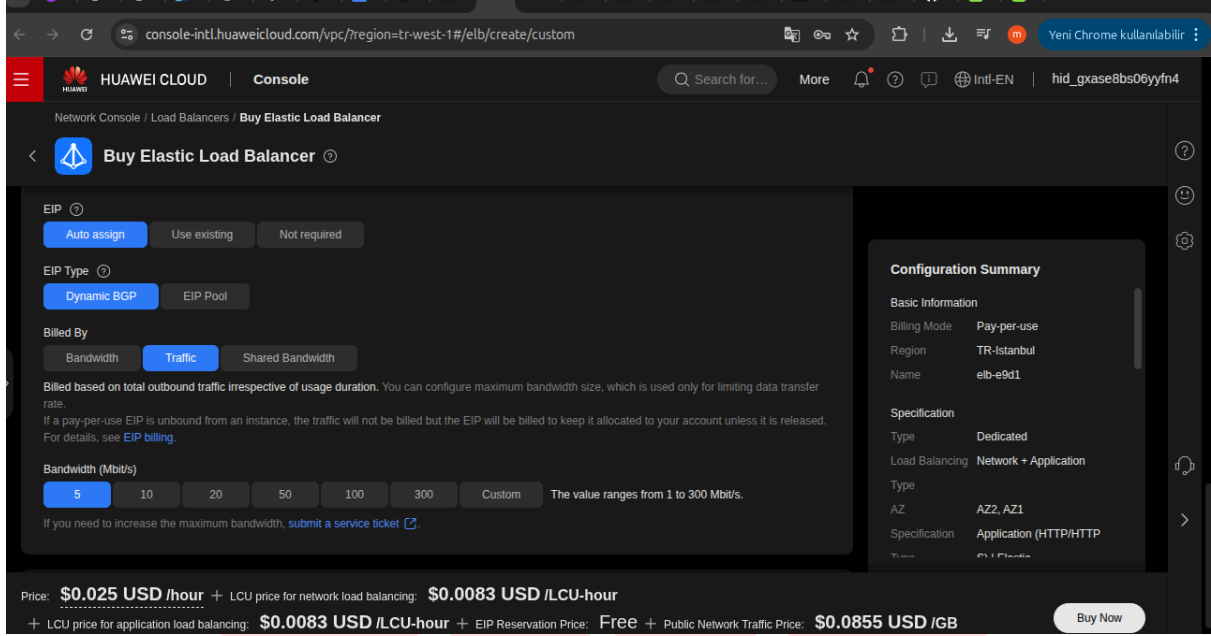


Service	Namespace	Type	IP	Port	Created	Operation
api-s...	api	ClusterIP	10.24...	80 -> 80 / ...	Dec 15, 2...	Manage Pod View Events More
hello...	hello	ClusterIP	10.24...	80 -> 567...	Dec 15, 2...	Manage Pod View Events More
web-...	web	ClusterIP	10.24...	80 -> 80 / ...	Dec 15, 2...	Manage Pod View Events More

(Görsel 1.4: Service'lerin console üzerinden kontrolü.)

## 7)Ingress Controller için Load Balancer Oluşturulması

Ingress yapılandırmasının yapılabilmesi için NGINX Ingress Controller add-on'u kullanılmıştır. Bu kapsamda, cluster'ın kurulu olduğu VPC ve node'ların bulunduğu **plane subnet** üzerinde özel bir **Elastic Load Balancer (ELB)** oluşturulmuştur. Oluşturulan Load Balancer, Ingress Controller'ın dış erişimi karşılaması amacıyla yapılandırılmıştır.



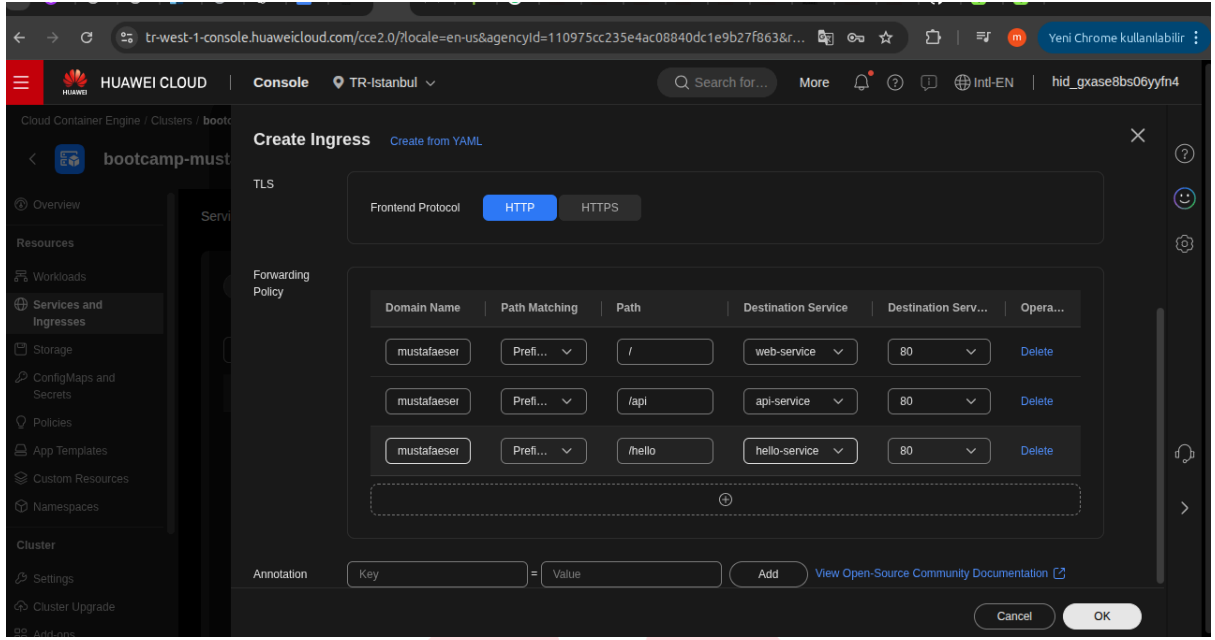
(Görsel 1.5: LoadBalancer alımı.)

## 8)Ingress Controller Kurulumu ve Domain Yönlendirme Ayarları

Ingress yapılandırmasına geçilmeden önce NGINX Ingress Controller add-on cluster üzerine kurulmuştur. Ardından, uygulamaların dış dünyadan erişilebilir olması için mustafaesen.com.tr domaininin DNS servis sağlayıcısı üzerinden, oluşturulan Load Balancer'a ait public IP adresine A kaydı tanımlanmıştır. Bu adımlar ile domain trafiğinin cluster'a yönlendirilmesi sağlanmıştır.

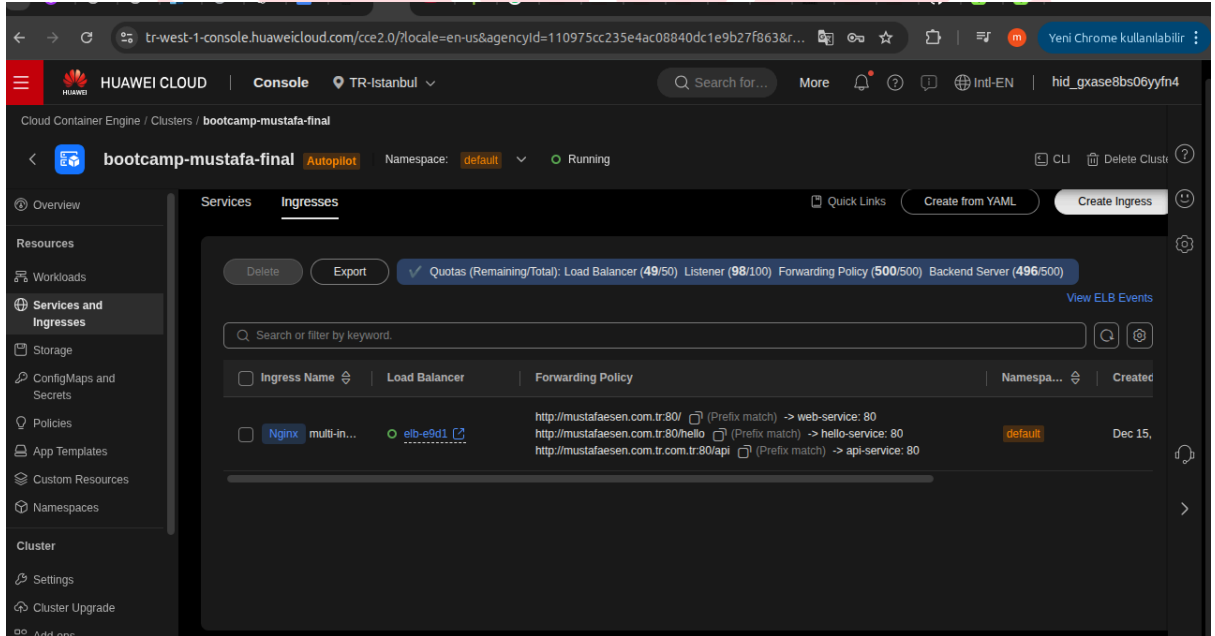
## 9)Ingress Kurallarının Tanımlanması

Ingress kuralı Huawei Cloud CCE Console arayüzü üzerinden tanımlanmıştır. Aynı domain altında multi-path yapı kullanılarak /, /api ve /hello path'leri sırasıyla **Web**, **API** ve **Hello** servislerine yönlendirilmiştir. Bu yapı sayesinde deployment ve service bileşenleri Ingress üzerinden domain aracılığıyla erişilebilir hale getirilmiştir.



(Görsel 1.6: Ingress kuralı tanımlama.)

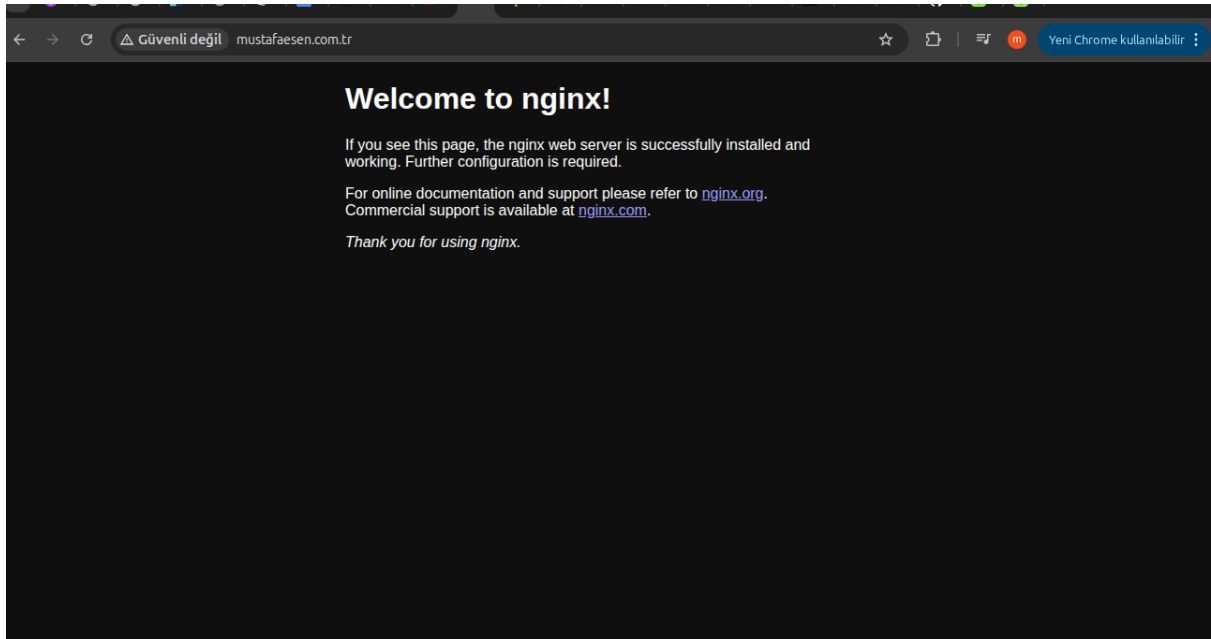
Daha sonra tanımlanan Ingress kuralının Huawei Cloud CCE Console üzerinde başarıyla oluşturulduğu doğrulanmıştır. Ingress'in ilgili Load Balancer ile ilişkilendiği ve aynı domain altında **/**, **/api** ve **/hello** path'lerinin sırasıyla **Web**, **API** ve **Hello** servislerine yönlendirildiği görülmüştür. Bu adım ile uygulamaların domain üzerinden erişilebilir olduğu teyit edilmiştir.



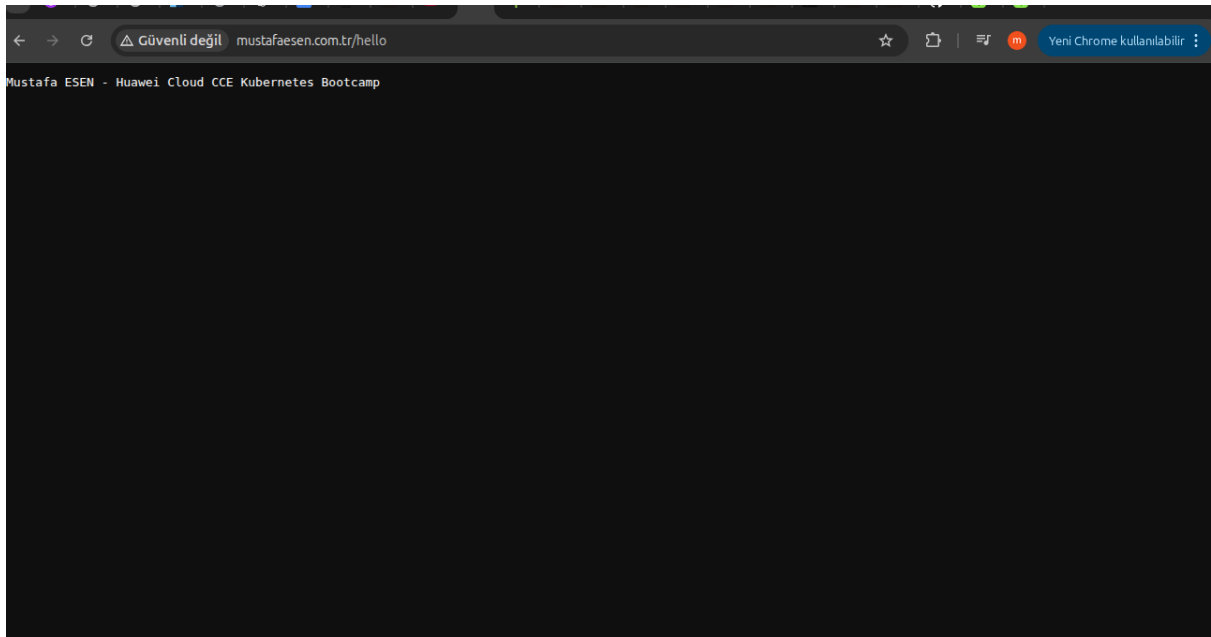
(Görsel 1.7: Ingress Kuralı doğrulama.)

## 10)Uygulamanın Domain Üzerinden Erişim Kontrolleri

Ingress yapılandırmasının tamamlanmasının ardından uygulamaların domain üzerinden erişilebilirliği sırasıyla test edilmiştir. **Ana domain** adresine yapılan erişimde **web deployment'a** ait Nginx servisinin başarıyla çalıştığı doğrulanmıştır. **/hello path'i** üzerinden yapılan istekte ilgili **Hello** servisine yönlendirme sağlanmış ve beklenen çıktı alınmıştır. **/api path'i** üzerinden yapılan istekte ise **404 Not Found** yanıtı alınmıştır. Bunun nedeni, eğitimde kullanılan **API image**'ının erişilebilir olmaması sebebiyle alternatif bir image kullanılması ve bu image'ın JSON formatında veri döndürmemesidir. Bu durum Ingress yönlendirme mekanizmasının çalışmasını etkilememiş, yalnızca uygulama içeriği nedeniyle boş yanıt oluşmuştur. Yapılan kontroller ile multi-path Ingress yönlendirme yapısının doğru şekilde çalıştığı teyit edilmiştir.



(Görsel 1.8: / ana sayfanın kontrolü.)



(Görsel 1.9: hello path kontrolü)



## 404 Not Found

nginx/1.29.4

(Görsel 2.0: api path kontrolü.)

### HTTPS ve SSL Gözlemi

HTTPS erişimi, SSL sertifikası yapılandırması yapılmadığı için yalnızca gözlemsel olarak incelenmiştir. Bu kapsamda HTTPS trafiğinin Ingress ve Load Balancer seviyesinde karşılanabildiği görülmüş, ancak sertifika tanımlaması yapılmadığından güvenli bağlantı kurulmamıştır. SSL yapılandırması demo kapsamı dışında bırakılmıştır.

### Genel Değerlendirme ve Sonuç

Bu çalışma kapsamında Huawei Cloud CCE Kubernetes Bootcamp eğitimi boyunca öğrenilen temel Kubernetes kavramları uygulamalı olarak hayata geçirilmiştir. Cluster kurulumu, ağ yapılandırmaları, deployment ve service tanımlamaları, Ingress controller kullanımı ve domain üzerinden erişim adımları uçtan uca uygulanmıştır. Gerçek bir senaryo üzerinden yapılan bu final proje ile Kubernetes bileşenlerinin birlikte çalışması ve uygulamaların Huawei Cloud altyapısı üzerinde yönetimi başarıyla gösterilmiştir.