

CKAD Ders Dökümanı #22 – Ingress Networking

Dil: Türkçe • CKAD Kategorisi: Services and Networking (20%) • Güncelleme: çoklu rule, çoklu path, TLS ve kubectl ile Ingress oluşturma eklendi

1) Konu Anlatımı

Ingress, dışarıdan gelen HTTP/HTTPS trafigini cluster içindeki Service'lere yönlendiren kurallar bütünüdür. Ingress'in çalışması için mutlaka bir Ingress Controller gereklidir (nginx-ingress, Traefik vb.). Ingress kaynağı tek başına trafik kabul etmez; controller bunu okuyup gerçek yönlendirmeyi gerçekleştirir.

CKAD'de tipik beklenen: host/path bazlı yönlendirme, doğru service/port referansı ve troubleshooting sırasında ingress → service → endpoints zincirini hızlı kontrol etmektir.

2) Rule, Path ve TLS Mantığı

Rule genellikle bir host (domain) ve o host altında bir veya daha fazla path tanımından oluşur. Her path, bir backend service'ye gider. Ayrıca HTTPS için TLS bölümünde host + secret eşlemesi yapılır.

2.1 Birden fazla rule (host bazlı)

Aynı Ingress içinde birden fazla host tanımlayabilirsiniz. Örneğin web.example.com ve api.example.com farklı service'lere gidebilir.

2.2 Birden fazla path (aynı host altında)

Aynı host altında / ve /api gibi path'lerle farklı backend'lere yönlendirebilirsiniz.

2.3 TLS (HTTPS) tanımı

TLS tanımı için genelde bir Secret gereklidir (type: kubernetes.io/tls). Ingress spec.tls içinde hangi host'ların hangi secret ile HTTPS servis edileceğini belirtirsiniz.

3) Örnekler (YAML)

3.1 Çoklu host + çoklu path + TLS örneği

Senaryo: • web.example.com → / path'i web-svc:80 • api.example.com → / path'i api-svc:80 • web.example.com üzerinde ayrıca /api path'i api-svc:80 • TLS: web.example.com ve api.example.com için aynı secret kullanılıyor

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: app-ing
spec:
  tls:
  - hosts:
    - web.example.com
    - api.example.com
```

```

secretName: app-tls
rules:
- host: web.example.com
  http:
    paths:
      - path: /
        pathType: Prefix
        backend:
          service:
            name: web-svc
            port:
              number: 80
      - path: /api
        pathType: Prefix
        backend:
          service:
            name: api-svc
            port:
              number: 80
- host: api.example.com
  http:
    paths:
      - path: /
        pathType: Prefix
        backend:
          service:
            name: api-svc
            port:
              number: 80

```

3.2 TLS Secret örneği (kubernetes.io/tls)

Not: Sertifika dosyaları hazırlırsa en hızlı yol kubectl create secret tls kullanmaktır.

```
kubectl create secret tls app-tls --cert=tls.crt --key=tls.key
kubectl get secret app-tls -o yaml
```

4) pathType Kısa Rehber

pathType	Ne anlama gelir?	Kısa not
Prefix	Path prefix eşleşmesi	/api → /api, /api/v1 gibi tüm alt yolları kapsar (en yaygın).
Exact	Tam eşleşme	/api sadece /api ile eşleşir, /api/v1 ile eşleşmez.
ImplementationSpecific	Spesifikler'a bağlı davranış	Nginx/Traefik'e göre değişebilir; sınavda Prefix/Exact tercih edilebilir.

5) kubectl ile Ingress Oluşturma

CKAD'de genellikle YAML ile apply kullanılır; ancak pratikte kubectl ile hızlı oluşturmak da işe yarar. Aşağıdaki komutlar kubectl sürümüne göre değişebilir; yine de en sık kullanılan yaklaşım kubectl create ingress (mevcutsa) veya kubectl apply -f'tır.

5.1 YAML ile (en güvenilir yöntem)

```
kubectl apply -f ingress.yaml  
kubectl get ingress  
kubectl describe ingress app-ing
```

5.2 kubectl create ingress (kubectl sürümüne bağlı)

Bazı kubectl sürümlerinde create ingress komutu vardır. Örnek kullanım aşağıdaki gibidir. Komut opsiyonları sürümde göre değişimden, sınavda çoğunlukla YAML tercih edilir.

```
kubectl create ingress app-ing --rule="web.example.com/*=web-svc:80"
```

5.3 Hızlı doğrulama/test

```
kubectl get ingress app-ing -o yaml  
kubectl describe ingress app-ing  
kubectl describe svc web-svc  
kubectl get endpoints web-svc
```

6) En Sık Kullanılan Komutlar ve Kullanımları

```
kubectl get ingress  
kubectl get ingress -A  
kubectl describe ingress app-ing  
kubectl get ingress app-ing -o yaml  
  
# (Varsa) ingress controller podlarını kontrol (örnek namespace)  
kubectl get pod -n ingress-nginx  
kubectl logs -n ingress-nginx deploy/ingress-nginx-controller
```

7) Troubleshooting Hızlı Rehber

ADDRESS/CLASS boş: Ingress Controller kurulu mu? (ingress-nginx/traefik podları)404 / 502: host/path eşleşmesi veya backend service/port yanlış olabilir.Endpoint yok: Service selector yanlış veya Pod Ready değil → kubectl get endpoints.TLS çalışmıyor: Secret adı yanlış veya secret type/tls key'leri eksik olabilir.