



Kubernetes'te Observability = kubectl ile Başlar

Kubernetes cluster'ınızı izlemek ve sorunları tespit etmek için kubectl komutlarıyla başlayan temel observability rehberi. Infrastructure'dan uygulama seviyesine kadar her katmanda gözlem yapmanın yolları.

1 Cluster Sağlığı

En Üst Seviye

Ne görmek isterim?

- Cluster ayakta mı?
- Node'lar sağlıklı mı?

kubectl cluster-info

kubectl get nodes

kubectl describe node <node-adi>

Ne gözlemlersin?

- Ready / NotReady
- DiskPressure
- MemoryPressure
- NetworkUnavailable

❑ ➡ Bu, **infrastructure observability**'nin ilk adımıdır.

2 Node Seviyesi İzleme

Kaynak Kullanımı

```
kubectl top nodes
```

Görebildiklerin

- CPU kullanımı
- Memory kullanımı

Sorular

- Hangi node dolu?
- Hangi node rahat?

Monitoring aracı yokken bile "**Node boğuluyor mu?**" sorusunun cevabı burada.

3 Pod Seviyesi İzleme

```
kubectl get pods -A  
kubectl get pods -o wide  
kubectl describe pod <pod-adi>
```

Özellikle bakılan alanlar:

- Status (Running / Pending / CrashLoop)
- Restart Count
- Events

```
kubectl top pods -A
```

❑ ➡ Bu uygulama seviyesinde metric'tir.



4

Event'ler = Kubernetes'in Günlüğü

Event = Kubernetes'in kendi tuttuğu "olan biten"

```
kubectl get events -A  
kubectl get events --sort-by=.lastTimestamp
```

En sık görülenler

- FailedScheduling
- BackOff
- FailedMount
- ImagePullBackOff

"Niye olmadı?"

👉 sorusunun ilk cevabı.

5 Log = Uygulama Ne Diyor?

```
kubectl logs <pod-adi>  
kubectl logs <pod-adi> -c <container-adi>  
kubectl logs <pod-adi> --previous
```

Örnek senaryolar:

- CrashLoopBackOff → --previous
- Çok container'lı Pod → -c

👉 Bu **uygulama içi observability**.

6 Canlı İzleme

Tail, Watch

```
kubectl logs -f <pod-adi>  
kubectl get pods -w  
kubectl get events -w
```



Deploy sırasında



Pod restart olurken



Scale anında

"Bir şey yaptım, etkisini izliyorum."

7

Service & Network Gözlemei

```
kubectl get svc  
kubectl describe svc <svc-adi>  
kubectl get endpoints <svc-adi>
```



Service Pod'lara bağlı mı?



Endpoint var mı?



Trafik gidecek yer buluyor mu?

- ❑ "Service var ama çalışmıyor" durumlarının %70'i burada çözülür.

8

Deployment & Rollout İzleme

```
kubectl get deployments
```

```
kubectl rollout status deployment/<deploy-adi>
```

```
kubectl rollout history deployment/<deploy-adi>
```

Deploy ilerliyor mu?

Takıldı mı?

Eski ReplicaSet duruyor mu?

👉 Bu da **deployment observability**.

9 Kaynak Baskısı & OOM İzleme

```
kubectl describe pod <pod-adi> | grep -i oom  
kubectl describe pod <pod-adi> | grep -i -E  
"Killed|Reason"
```

```
kubectl get pod <pod-adi> -o  
jsonpath='{.status.containerStatuses[*].restartCount  
'}
```

"Uygulama niye restart oluyor?"

👉 sorusu.

10 Namespace Bazlı Gözlem (ÇOK ÖNEMLİ)

```
kubectl get pods -n <namespace>  
kubectl get events -n <namespace>  
kubectl top pods -n <namespace>
```

Production'da asla -A ile yaşamazsin.
Namespace = izleme sınırı.

