

# Kubernetes Ne Değildir?



## Bir uygulama değildir

Kubernetes, uygulamaların çalıştırıldığı ve yönetildiği bir orkestrasyon platformudur, uygulamanın kendisi veya bir parçası değildir.

## Bir sihir değildir

Karmaşık görünse de, Kubernetes sihirli bir çözüm değil, tanımlanmış kurallar ve mantıkla çalışan sistematik bir yapıdır.

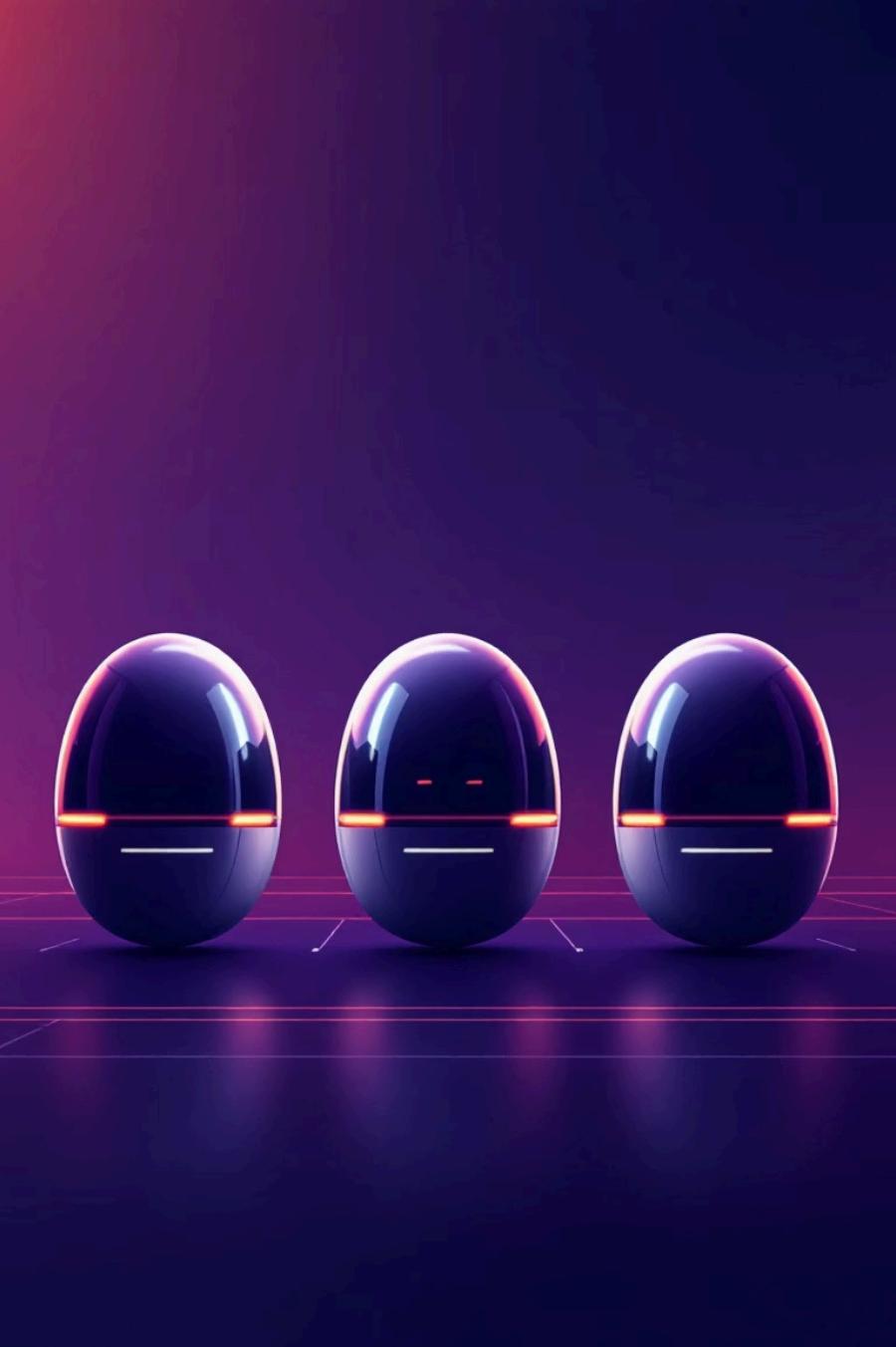
## "Bir kere kur, unut" değildir

Kubernetes, sürekli izleme, yapılandırma ve optimizasyon gerektiren dinamik bir sistemdir; bir kez kurulup bırakılamaz.

## Kubernetes = sürekli çalışan bir kontrol mekanizması

Uygulamalarınızın durumunu sürekli olarak denetleyen, istenen duruma getiren ve sürdürden aktif bir kontrol döngüsüdür.

# Kubernetes'in Temel Mantığı



# Kubernetes'in Temel Mantığı

## Desired State (İstenen Durum)

"Ben şunu istiyorum" dersin

Kubernetes de sürekli onu korumaya çalışır

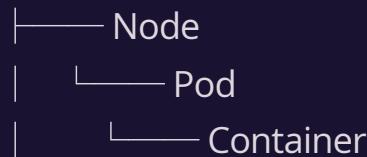
## Örnek:

3 tane pod çalışsin

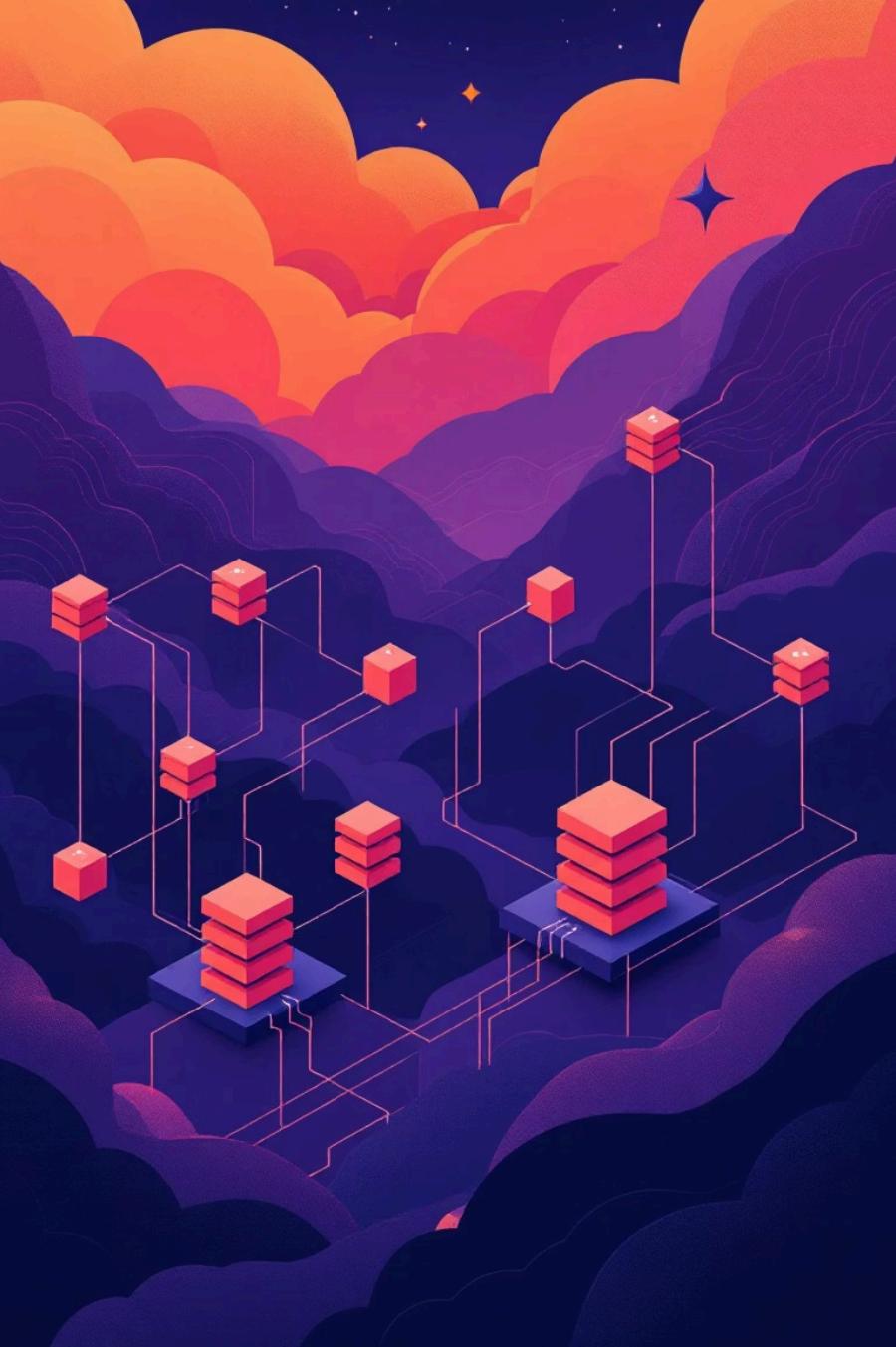
Birisi ölürse → yenisi gelsin

# Temel Kavramlar (Üstten Aşağı)

Cluster



**Kubernetes container'ı değil,**  
pod'u yönetir (çok önemli).



# Cluster Nedir?

**Kubernetes'in tamamı**

Control Plane + Worker Node'lar

**Tek bir mantıksal sistem**

Cluster = sizin "data center"ınız gibi düşünün.

# Node Nedir?



## Fiziksel ya da sanal makine

Kubernetes kümesindeki her bir Node, uygulamaları çalıştırılmak için kaynak sağlayan fiziksel bir sunucu veya bir sanal makine olabilir.



## Pod'lar burada çalışır

Pod'lar, Kubernetes'in en küçük dağıtılabilebilir birimi olup, Node'lar üzerinde uygulamaları ve hizmetleri barındırır ve çalıştırır.



## Kubernetes node'a "şunu çalıştır" der

Kubernetes Control Plane, Pod'ların ve konteynerlerin hangi Node'larda çalışacağını belirler ve Node'lara gerekli talimatları gönderir.

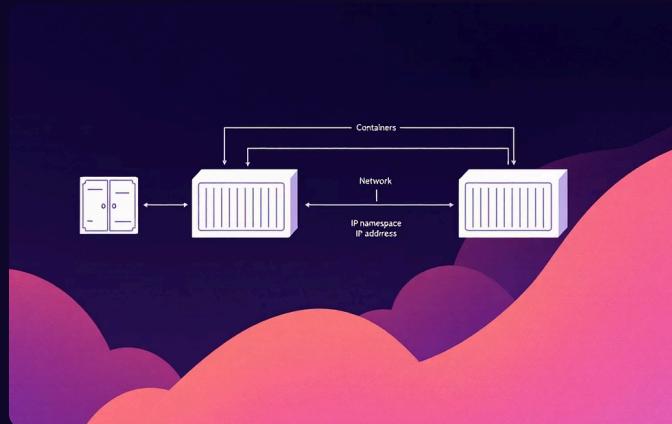
# Pod Nedir?

Kubernetes'in en küçük birimi



Bir veya birden fazla container içerir

Bir Pod, bir veya daha fazla konteyneri bir araya getirerek tek bir mantıksal birim olarak çalıştırır.



Aynı IP, aynı network namespace

Pod içindeki tüm konteynerler aynı IP adresini ve ağ alanı (network namespace) paylaşır, bu da onların kolayca iletişim kurmasını sağlar.



Pod = "birlikte yaşaması gereken container'lar"

Pod'lar, sıkı bir şekilde bağlantılı ve birlikte çalışması gereken uygulamalar için idealdir; yaşam döngüleri birlikte yönetilir.

# Kubernetes Mimarisi (Yüksek Seviye)



## Control Plane Ne Yapar?

- Karar verir
- Planlar
- Takip eder

Beyin ayrı, kaslar ayrı.



## Worker Node:

- Çalıştırır

# Control Plane Bileşenleri (Rol Odaklı)



**API Server**

Her şeyin kapısı



**etcd**

Hafıza



**Scheduler**

Nereye koyayım?



**Controller**

Düzenin mü çalışıyor?

**İsimleri ezberlemiyoruz,** ne yaptıklarını anlıyoruz.

# Kubernetes Kontrol Döngüsü



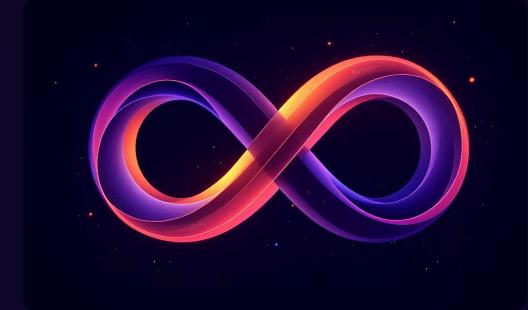
Sen bir şey istiyorsun



Kubernetes kontrol  
ediyor



Fark varsa düzeltmeye  
çalışıyor



Sonsuz döngü

Kubernetes'in gücü burada.