

## SLAYT 1 - 2. Gün Hedefi

Bugün Kubernetes'i  
kullanacağımız, sadece  
çalıştırmayacağız.

# SLAYT 2 – Service & Networking

## ClusterIP

Sadece cluster içinde erişilebilir

## NodePort

Her node'da port açar, dışarıdan erişim

## LoadBalancer

Bulut sağlayıcı load balancer'ı kullanır

## kube-proxy

Ağ kurallarını yönetir

# SLAYT 3 – DNS & Ingress



## Service name → IP

DNS servisi service adını IP adresine çevirir



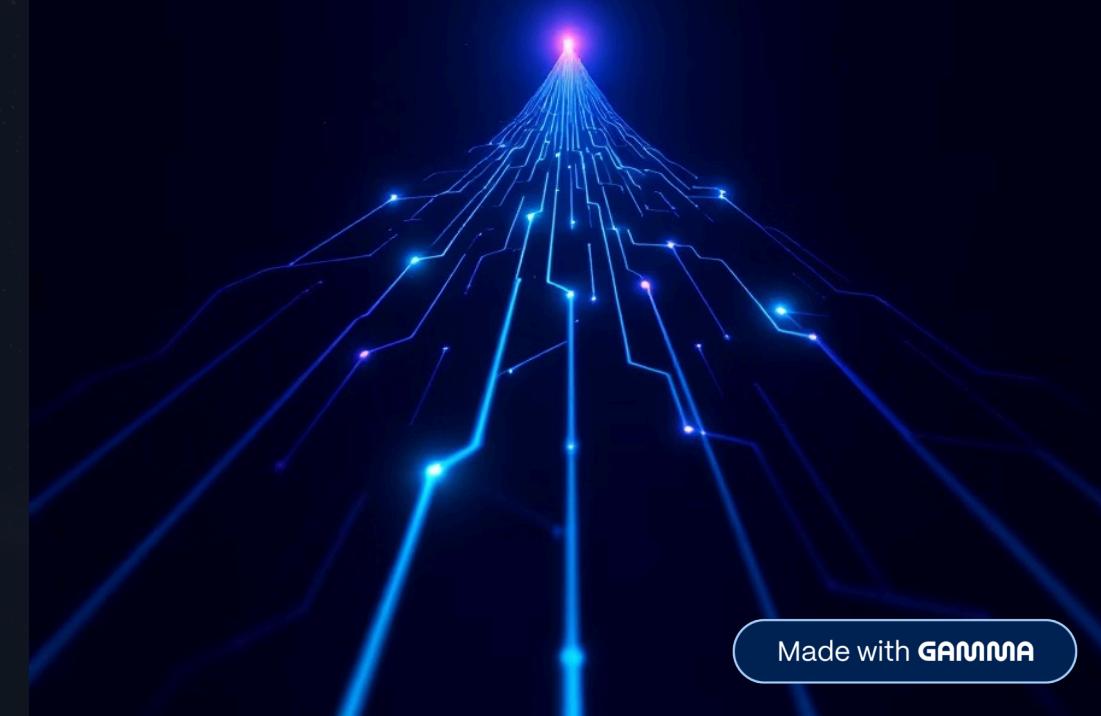
## Ingress neden var?

HTTP/HTTPS trafiğini yönetmek için gerekli

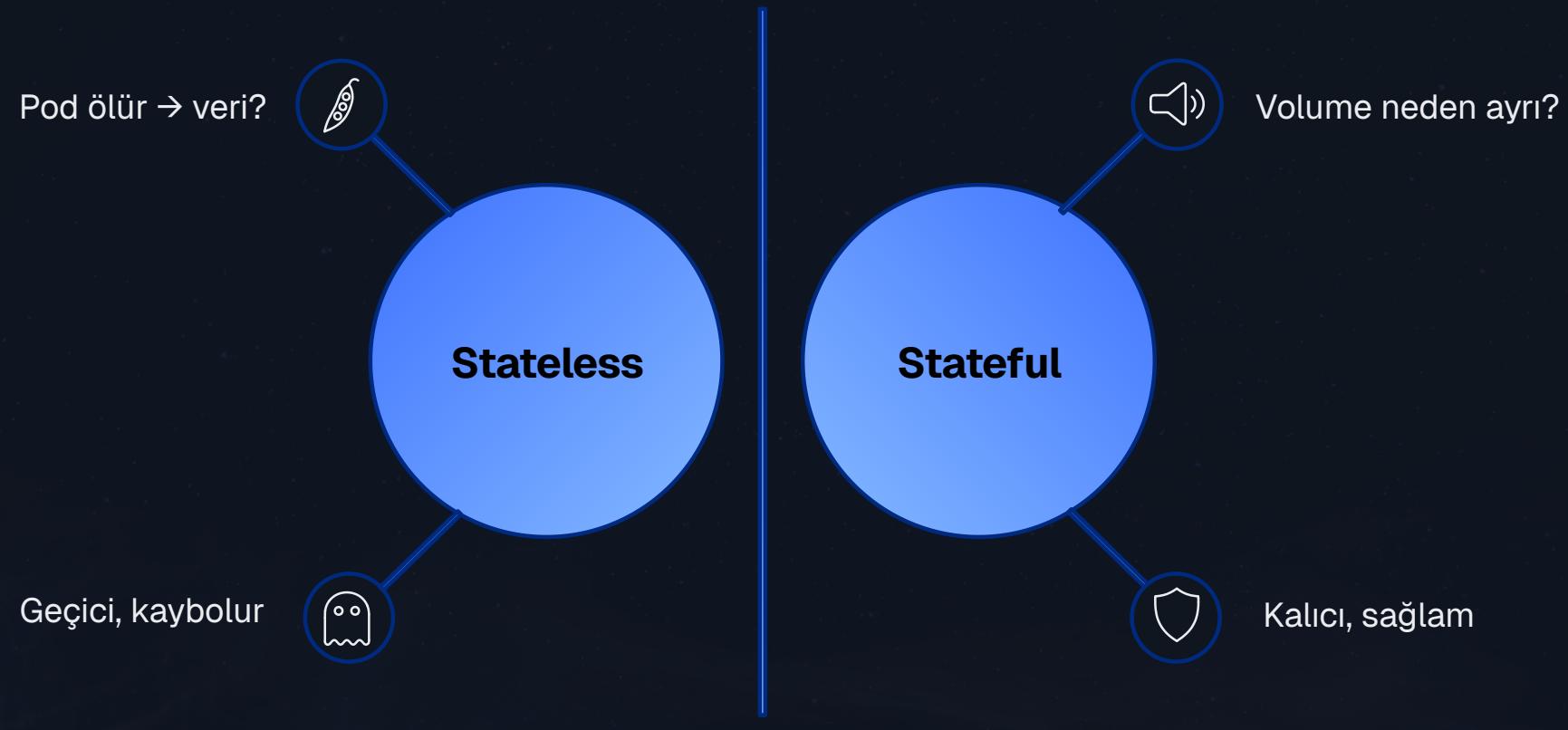


## Tek giriş noktası

Dış dünyadan cluster'a tek kapı



# SLAYT 4 – Stateless vs Stateful



# SLAYT 5 – Storage Mimarisi



## PV (Persistent Volume)

Gerçek depolama alanı olup, Kubernetes cluster'ının fiziksel olarak dışında konumlanır. NFS, AWS EBS, GCP Persistent Disk gibi çeşitli depolama türlerini destekler. Bu kaynaklar genellikle cluster yöneticisi tarafından önceden oluşturulur ve yönetilir. Pod'lar, veri kalıcılığı sağlamak için doğrudan PV'ye erişmez, bunun yerine PVC aracılığıyla dolaylı olarak bağlanır.



## PVC (Persistent Volume Claim)

Pod'un depolama talebini temsil eder ve bir PV'ye bağlanır. Geliştiriciler, ihtiyaç duydukları depolama özelliklerini (boyut, erişim modu vb.) belirten bir PVC tanımlar. Kubernetes bu talebi karşılamak için uygun bir PV bulur ve ona bağlar. Bir PVC sadece bir PV'ye bağlanabilir ve pod silinse bile PVC ve bağlı PV kalıcıdır, böylece verilerin korunması sağlanır.



## StorageClass

Dinamik PV oluşturma kurallarını tanımlar ve otomatik provisioning sağlar. Bir PVC oluşturulduğunda, StorageClass sayesinde Kubernetes otomatik olarak uygun bir PV yaratır ve PVC'ye bağlayabilir. Depolama türü, boyutu, performans özellikleri ve geri kazanım politikaları gibi ayarları içerir. AWS, GCP, Azure gibi bulut sağlayıcılarının depolama hizmetleriyle sorunsuz entegrasyon için kullanılır.

# SLAYT 6 – Resource Management

1

## requests

Pod'un minimum kaynağı. Scheduler bunu kullanarak node seçer. Garantili kaynak.

2

## limits

Pod'un maksimum kaynağı. Bunu aşarsa sistem müdahale eder. Sınır koyar.

3

## OOMKilled

Out of Memory Killed. Pod bellek limitini aştığında sistem tarafından öldürülür.



# SLAYT 7 – Scaling



## Manual scale

Pod sayısını elle artırıp azaltırsınız. Önceden bilinen yüklerle karşı kullanılır.



## HPA mantığı

Horizontal Pod Autoscaler CPU veya bellek kullanımına göre otomatik scale eder. Yük arttığında pod ekler, azaldığında siler.



## Neden her şey scale edilmez?

Veritabanı gibi stateful uygulamalar scale edilemez. Sadece stateless uygulamalar güvenli şekilde scale edilebilir.

# SLAYT 8 – Update & Rollback



## RollingUpdate

Eski pod'ları yavaş yavaş yeni pod'larla değiştirir. Hizmet hiç kesintiye uğramaz, her zaman aktif pod vardır.



## Blue-Green (kavramsal)

İki ortam vardır: mavi (eski) ve yeşil (yeni). Tüm trafik aniden yeşile geçer, sorun olursa mavi'ye geri dönülür.



## Canary (kavramsal)

Yeni versiyonu küçük bir yüzdeye gönderirsiniz. Sorun yoksa yavaş yavaş tüm trafiği yeni versiyona yönlendirirsiniz.

# SLAYT 9 – Debug & Gerçek Hayat

## CrashLoop

Pod başlıyor, hata alıyor ve kapanıyor. Logları kontrol edin, uygulama kodunda sorun var demektir.

## Pending

Pod oluşturulmaya çalışılıyor ama başlatılamıyor. Yeterli kaynak yok veya image bulunamıyor olabilir.

## ImagePull

Container image'ı indiremedi. Registry'ye erişim sorunu veya image adı yanlış olabilir.

## Panik yok

Hata mesajları normal bir parçasıdır. Logları okuyun, sorunun kaynağını bulun ve çözün.