

Kubernetes + Docker Demo Anlatımı

Sunum Öncesi Hazırlık (5 dakika önce)

1. Docker Desktop'ı Başlat

- Docker Desktop uygulamasını aç
- Sol altta Kubernetes ikonunun **yeşil** olduğundan emin ol
- Eğer yeşil değilse, 2-3 dakika bekle

2. Terminal'i Aç

- Spotlight (Cmd + Space) → "Terminal" yaz
- Terminal penceresini yan tarafa yerleştir (ekran paylaşımı için)

3. Demo Script'ini Çalıştır

```
cd ~/Desktop
bash kubernetes-demo-setup.sh
```

Script otomatik olarak:

- Eski demo'ları temizler
- 3 adet pod oluşturur
- Service'i yapılandırır
- Port numarasını gösterir

4. Tarayıcıyı Hazırla

- Chrome veya Edge'i aç
- Script'in gösterdiği URL'yi not al (örn: <http://localhost:31222>)
- Bir kez test et

Demo Sırasında Anlatım (3-5 dakika)

Giriş (30 saniye)

"Bugün size Kubernetes'in load balancing özelliğini göstereceğim. Aynı container imajından 3 farklı pod oluşturup, gelen isteklerin otomatik olarak bu podlar arasında nasıl dağıtıldığını canlı olarak izleyeceğiz."

Adım 1: Pod'ları Göster (30 saniye)

Terminal'de:

```
kubectl get pods
```

"Gördüğünüz gibi 3 adet pod çalışıyor. Her biri aynı container imajından oluşturulmuş ancak bağımsız birer instance. Hepsi 'Running' durumunda."

Adım 2: İlk Request (30 saniye)

Yeni gizli sekme aç → URL'yi gir

"Şimdi ilk isteğimizi gönderiyorum. JSON çıktısında 'os.hostname' alanına dikkat edin."

Hostname'i göster (örn: demo-app-755fb554d-7hpr5)

"Bu istek demo-app-7hpr5 pod'una gitti."

Adım 3: Farklı Pod'lara Gidiş (1 dakika)

Gizli sekmeyi **kapat**

"Şimdi yeni bir bağlantı açıyorum..."

Yeni gizli sekme aç → Aynı URL

"Ve bu sefer farklı bir pod'a gitti!"

Hostname'i göster (örn: demo-app-755fb554d-mqx8w)

"Kubernetes Service otomatik olarak isteęi **demo-app-mqx8w** pod'una yönlendirdi."

2-3 kez daha tekrarla (farklı gizli sekmeler)

Adım 4: Teknik Açıklama (1 dakika)

"Peki bu nasıl çalışıyor?"

Terminal'de mimariye bak:

```
kubectl get service demo-app
```

"Kubernetes Service, gelen tüm istekleri 3 pod arasında otomatik olarak dağıtıyor. Bu sayede:

- Yük dengeleme (load balancing) sağlıyor
- Bir pod çökse diğerleri çalışmaya devam eder
- İhtiyaç halinde pod sayısını artırabiliriz"

Adım 5: Ölçeklendirme Gösterisi (Bonus - 1 dakika)

```
kubectl scale deployment demo-app --replicas=5
```

"Şimdi pod sayısını 5'e çıkarıyorum..."

```
kubectl get pods
```

"Görüyorsunuz, 2 yeni pod oluşturuldu. Kubernetes bu kadar basit!"

Kapanış (30 saniye)

"Özetlersek: Docker ile containerları oluşturuyoruz, Kubernetes ile bu containerları yönetiyor ve ölçeklendiriyoruz. Her request farklı bir pod'a gidebiliyor ve sistem otomatik olarak dengeleniyor. Sorularınız var mı?"

Demo Sonrası Temizlik

```
bash kubernetes-demo-cleanup.sh
```

Olası Sorunlar ve Çözümler

Problem: "kubectl: command not found"

Çözüm: Docker Desktop'ta Kubernetes aktif değil

- Settings → Kubernetes → Enable Kubernetes

Problem: Pod'lar "Running" değil

Çözüm: 30 saniye bekle, tekrar kontrol et

```
kubectl get pods
```

Problem: Aynı pod'a gidiyor

Çözüm: Gizli sekme kullandığından emin ol

- Her yeni request için YENİ gizli sekme aç
- Normal sekmede HTTP Keep-Alive nedeniyle aynı pod'a gider

Problem: Port'a erişilemiyor

Çözüm: Service'in port numarasını kontrol et

```
kubectl get service demo-app
```

PORT(S) sütununda gösterilen ikinci numarayı kullan

Sunum İpuçları

- ☒ Terminal fontunu büyüt (Cmd + Plus)
- ☒ Tarayıcı zoom'unu artır (%150-200)
- ☒ Gizli sekme shortcut: Cmd + Shift + N
- ☒ Her yeni request için YENİ gizli sekme aç
- ☒ JSON'da "os.hostname" kısmını highlight et
- ☒ Sakin ve yavaş konuş, adımları açıkla
- ☒ Soru-cevap için zaman bırak

Yedek Senaryo (Eğer bir şeyler ters giderse)

Demo çalışmazsa, hazırladığım ekran görüntülerini kullan:

1. Terminal'deki pod listesi
2. Tarayıcıda farklı hostname'ler gösteren screenshot'lar

"Teknik bir sorun nedeniyle canlı gösteremiyorum ama hazırladığım kayıtlarda sürecin nasıl işlediğini görebilirsiniz."

Başarılar! ☒