

## Uygulamanızı İzleme ve Ölçeklendirme

Etkili izleme ve stratejik ölçeklendirme, kullanıcı talepleri ve yük değiştikçe web uygulamalarının performansını ve istikrarını korumak ve geliştirmek için hayati önem taşır. Bu uygulamalar, kritik performans ölçümlerinin sürekli izlenmesini ve değişen ihtiyaçları verimli bir şekilde ele almak için kaynakları dinamik olarak ayarlama yeteneğini içerir. Bu makalede, yetkin izleme ve ölçeklendirme için gerekli yöntemler, araçlar ve teknolojiler özetlenmektedir. uygulamaların ölçeklendirilmesi ve uygulama performansını optimize etmek için bunların nasıl uygulanacağı açıklanmaktadır.

### İzlemenin Temelleri

İzleme, bir uygulamadan performans verilerinin toplanması, analiz edilmesi ve bunlara göre hareket edilmesine yönelik sürekli bir süreçtir. Bu süreç, kullanıcı deneyimini olumsuz etkileyebilecek veya kesinti süresine neden olabilecek performans yavaşlamaları, sistem arızaları veya operasyonel anormallikler gibi potansiyel sorunları tespit etmek için çok önemlidir.

### İzlenecek Kritik Metrikler:

- 1.Trafik Seviyeleri: Kullanıcı trafiğini takip etmek, yükteki değişiklikleri öngörmeye ve bunlara hazırlanmaya yardımcı olur.
- 2.Kaynak Kullanımı: CPU, bellek ve disk kullanımı gibi metriklerin gözlemlenmesi, uygulamanın operasyonel eşikler içinde kalmasını sağlar.
- 3.Yanıt Süreleri ve Hata Oranları: Bu metriklerin izlenmesi, kullanıcının uygulama ile etkileşimini etkileyebilecek aksaklıkların tespit edilmesine yardımcı olur.
- 4.Verimlilik: Uygulamanın yönettiği işlem miktarının değerlendirilmesi, operasyonel verimliliği hakkında bilgi sağlar.
- 5.Veritabanı Performansı: Uygulama performansını önemli ölçüde etkileyebildikleri için veritabanı işlemlerinin izlenmesi çok önemlidir.

Bu metriklere dayalı uyarıların yapılandırılması, ekiplerin sorunları son kullanıcıları etkilemeden önce proaktif olarak yönetmesine olanak tanır.

### İzleme için Araçlar

Uygulama performansının derinlemesine izlenmesini kolaylaştıran çeşitli sofistike araçlar mevcuttur:

- New Relic: Gerçek zamanlı işlemleri ve geçmiş eğilimleri izlemek için ideal olan uygulama performansına ilişkin derin bilgiler sağlar.

- Datadog: Teklifler kapsamlı izleme yetenekleri Uygulamalar ve altyapı genelinde, geniş entegrasyonları ile dikkat çekiyor.
- Prometheus: Bulut tabanlı ortamlarda mükemmel olan bu araç, zaman serisi verilerini izlemek için idealdir.
- Nagios: Bilinen için onun kapsamlı izleme ve uyarı yetenekleri, sunucuları, ağları ve uygulamaları kapsar.

## Prometheus ile Örnek Kurulum:

```
# Install Prometheus
sudo apt-get install prometheus

# Configure Prometheus to monitor your application by editing the prometheus.yml
global:
  scrape_interval: 15s # Defines how frequently metrics should be scraped

scrape_configs:
  - job_name: 'my-application'
    static_configs:
      - targets: ['localhost:9090']

# Start Prometheus service to initiate monitoring
sudo systemctl start prometheus
```

## Ölçeklendirmenin Önemi

Ölçeklendirme, ek kaynaklar tahsis ederek veya iş yükünü ek sunuculara dağıtarak artan yükleri karşılamak için gereklidir.

## Ölçeklendirme Teknikleri:

- 1.Yatay Ölçeklendirme (Dışa Ölçeklendirme)Yükü etkili bir şekilde yaymak için daha fazla sunucu ekler.
- 2.Dikey Ölçeklendirme (Yukarı Ölçeklendirme)Kapasitelerini artırmak için mevcut sunucuların yeteneklerini (CPU, bellek) artırır.

## Ölçeklendirme için Araçlar

Modern araçlar ve platformlar ölçeklendirme için gelişmiş çözümler sunar:

- Kubernetes Yönetir konteynerli uygulamalar otomatik ölçeklendirme ve yük dengeleme özelliklerine sahiptir.
- AWS Otomatik Ölçeklendirme: Performansı ve maliyeti optimize etmek için bilgi işlem kaynaklarını talebe göre otomatik olarak ayarlar.
- Azure Ölçek Kümeleri: Sanal makineler ve bulut kaynakları için otomatik ölçeklendirmeyi yönetir.

Kubernetes ile Otomatik ölçeklendirme örneği:

```
# Kubernetes deployment setup
apiVersion: apps/v1
kind: Deployment
metadata:
  name: application-deployment
spec:
  replicas: 3 # Default number of replicas
  selector:
    matchLabels:
      app: application
  template:
    metadata:
      labels:
        app: application
    spec:
      containers:
        - name: application
          image: application:latest
          ports:
            - containerPort: 80
```

```
# Configuration for auto-scaling
apiVersion: autoscaling/v1
kind: HorizontalPodAutoscaler
metadata:
  name: application-scaler
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: application-deployment
  minReplicas: 3
  maxReplicas: 10
  targetCPUUtilizationPercentage: 50
```

Bu kurulum Kubernetes'e uygulamayı CPU kullanımına göre üç ila on örnek arasında otomatik olarak ölçeklendirmesi talimatını vererek kaynakların verimli kullanılmasını sağlar.

## Sonuç

Sağlam izleme ve ustaca ölçeklendirme, web uygulamalarının talepler geliştikçe adapte olabilmesini ve verimli bir şekilde çalışmasını sağlamak için vazgeçilmezdir. Gelişmiş izleme araçları ve ölçeklenebilir teknolojiler kullanılarak operasyonlar kolaylaştırılabilir, maliyetler azaltılabilir ve

kullanıcı memnuniyeti artırılabilir. Bu stratejilerin uygulanması, uygulamaların yüksek performans ve güvenilirlik standartlarını korurken operasyonel taleplere uyarlanabilir şekilde yanıt vermesini sağlar.