

# Geliştirici Görevi: Keycloak Olaylarının Güvenilir Yönetimi ve Kafka Üzerinden İletimi

## Problem Tanımı

Bir Kubernetes ortamında çalışan sistemde, **Keycloak** tarafından üretilen olayların eksiksiz, güvenilir ve sıralı bir şekilde bir mesajlaşma altyapısına (Kafka) iletilmesi gerekmektedir. Olayların diğer mikroservisler tarafından işlenebilmesi için bu altyapının veri kaybı olmadan ve yüksek performansla çalışması beklenmektedir. Geliştirilecek çözüm, **Quarkus** veya **Spring** framework'lerinden biri kullanılarak oluşturulmalıdır.

Sistem, **yüksek erişilebilirlik (HA)** desteklemeli, **Kafka'nın geçici olarak çalışmaması** durumunda bile işlemlerini sürdürebilmeli ve **Kubernetes üzerinde dağıtılabilir** bir yapıda olmalıdır.

## Karşılaşılan Zorluklar

### 1. Olay Yakalama ve Yönetimi

- Keycloak tarafından üretilen olaylar eksiksiz bir şekilde yakalanmalı ve güvenilir bir şekilde işlenmelidir.
- Tutarlılık ve veri kaybını önlemek bu sürecin kritik parçalarıdır.

### 2. Mesajlaşma Altyapısına İletim

- Olaylar sıralı bir şekilde ve eksiksiz olarak Kafka'ya iletilmelidir.
- Kafka'nın geçici olarak çalışmadığı durumlarda sistemin işlemleri kesintiye uğramadan çalışmaya devam etmesi gerekmektedir.

### 3. Framework Kullanımı

- Geliştirilecek çözüm için **Quarkus** veya **Spring** framework'lerinden biri kullanılmalıdır.
- Framework'ün özellikleri etkin bir şekilde kullanılmalı ve Keycloak ile entegrasyonu sağlanmalıdır.

#### 4. **Kubernetes Üzerinde Çalışabilirlik**

- Sistem Kubernetes ortamında dağıtılabılır, ölçeklenebilir ve kararlı bir şekilde çalışabilir olmalıdır.

#### 5. **Dayanıklılık ve Yüksek Erişilebilirlik (HA)**

- Sistem, yüksek trafik altında ölçeklenebilir bir yapıda olmalı ve dayanıklılığı garanti altına almalıdır.
- Olası başarısızlık durumlarında sistemin güvenilir bir şekilde toparlanması sağlanmalıdır.

## **Beklentiler**

### • **Keycloak Olay Yönetimi**

- Keycloak olayları eksiksiz bir şekilde yakalanmalıdır.

### • **Kafka'ya Güvenilir İletim**

- Olaylar sıralı ve eksiksiz bir şekilde Kafka'ya iletilmelidir.
- Kafka'nın çalışmaması durumunda olaylar güvenli bir şekilde bekletilmeli ve daha sonra gönderilmelidir.

### • **Framework Kullanımı**

- Çözüm **Quarkus** veya **Spring** framework'lerinden biri kullanılarak geliştirilmelidir.

### • **Kubernetes Uyumu**

- Sistem Kubernetes üzerinde çalışabilir olmalı ve ölçeklenebilir olmalıdır.

### • **Yüksek Erişilebilirlik (HA)**

- Sistem, her zaman erişilebilir olmalı ve başarısızlık durumlarını dayanıklı bir şekilde yönetebilmelidir.

### • **Hata Yönetimi**

- Hatalı işlemler doğru bir şekilde yönetilmeli, olaylar yeniden işlenebilir olmalıdır.

- **Test Edilebilirlik**

- Geliştirilen sistem, belirlenen senaryolarda (örneğin, login/logout olayları) eksiksiz çalışmalıdır.

## Değerlendirme Kriterleri

- **Güvenilirlik ve Tutarlılık**

- Sistemin olayları eksiksiz ve sıralı bir şekilde Kafka'ya iletip iletemediği test edilecektir.

- **Framework Kullanımı**

- Quarkus veya Spring framework'ünün doğru ve etkin kullanımı değerlendirilecektir.

- **Kubernetes Uyumluluğu ve Ölçeklenebilirlik**

- Çözümün Kubernetes üzerinde kararlı ve ölçeklenebilir bir şekilde çalışması beklenmektedir.

- **Dayanıklılık ve Hata Yönetimi**

- Kafka'nın çalışmaması gibi başarısızlık durumlarında sistemin güvenilir bir şekilde toparlanıp toparlanamadığı ve işlemlerini kesintisiz sürdürebilip sürdüremediği değerlendirilecektir.

- **Dokümantasyon**

- Geliştirilen çözümün tasarımını ve kullanılan teknolojileri açıklayan kısa ve anlaşılır bir dokümantasyon hazırlanmalıdır.

## Not

Sistem, Kubernetes üzerinde çalışabilir olmalı ve yüksek erişilebilirlik (HA) sağlamalıdır. Kafka'nın geçici olarak kullanılamadığı durumlarda bile sistemin işlevselliğini sürdürebilmesi kritik öneme sahiptir. Veri kaybını önlemek ve tutarlılığı sağlamak çözümün temel gereksinimlerindendir.