T.C.

SAKARYA ÜNİVERSİTESİ

BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ

Court Booking Web Uygulaması – AWS Üzerine Dağıtım

Öğr.Gör. UĞUR ÖZBEK

Bulut BiliŞim

KHALED MAHYOUB

B211200551

KHALED GAMAL MAHMOUD ZORAEI MOHAMED

B211200554

1. Amaç

Bu çalışmanın amacı, yerel ortamda geliştirilen bir web uygulamasının (Court Booking Website) AWS bulut altyapısı üzerine taşınması, bir sanal sunucu (EC2) üzerinde gerekli kurulumların yapılması ve uygulamanın internet üzerinden erişilebilir şekilde çalıştırılmasıdır. Rapor; kullanılan teknoloji seçimlerini, dağıtım adımlarını, karşılaşılan sorunları ve elde edilen çıktıları adım adım belgelemektedir.

2. Projenin Hedefleri

Bu proje kapsamında hedeflenen çıktılar aşağıdaki gibidir:

- AWS üzerinde bir EC2 sanal sunucusu oluşturmak.
- Uygulama çalışma ortamını Node.js kurmak ve bağımlılıkları yüklemek.
- Uygulamayı GitHub üzerinden sunucuya taşımak (repo klonlama).
- Supabase üzerinde oluşturulan veritabanını uygulamaya bağlamak ve verilerin bulut tabanlı saklanmasını sağlamak.

3. Uygulama Seçimi ve Proje Tanımı

Dağıtımı yapılan uygulama, kullanıcıların kort/saha rezervasyonu yapabildiği bir web uygulamasıdır. Uygulama üzerinden kullanıcı kayıt/giriş işlemleri yapılabilir, uygun zaman dilimleri görüntülenebilir ve rezervasyon oluşturulabilir. Bu ödev kapsamında odak; uygulamanın fonksiyonel geliştirmesinden ziyade, bulut ortamında çalışır hale getirilmesi (deployment) ve sürecin raporlanmasıdır.

4. Kullanılan Teknolojiler ve Seçim Gerekçeleri

Bu projede kullanılan bileşenler ve seçilme gerekçeleri aşağıda özetlenmiştir.

Bileşen	**Kullanım Amacı**	**Neden Bu Seçim?**
AWS EC2	Sanal sunucu (VM) üzerinde uygulamayı çalıştırma	Gerçek bulut ortamı, yaygın kullanım, esnek kaynak seçimi ve güvenlik grubu yönetimi
Ubuntu 24.04 LTS	Sunucu işletim sistemi	Kararlı, uzun süreli güvenlik güncellemeleri, geniş topluluk ve dökümantasyon
Node.js 20 LTS & npm	Uygulama çalışma ortamı ve paket yönetimi	Uygulama Node tabanlı; LTS sürüm ile uyumluluk ve stabilite

Vite	Frontend geliştirme/build aracı	Hızlı build/serve, modern frontend projeleri için standart
GitHub & gh CLI	Versiyon kontrol ve sunucuya kod taşıma	Takım çalışması, commit geçmişi, kolay klonlama (gh)
Supabase	Yönetilen veritabanı + backend servisleri	Hızlı kurulum, bulutta yönetilen DB, kolay entegrasyon (API/SDK)

5. Bulut Platformu Seçimi ve AWS Yapılandırması

Bulut platformu olarak AWS seçilmiştir. AWS; EC2 üzerinden sanal makine oluşturma, güvenlik grupları ile port/erişim yönetimi ve kolay ölçeklenebilirlik sunduğu için tercih edilmiştir.

5.1 EC2 Instance Konfigürasyonu

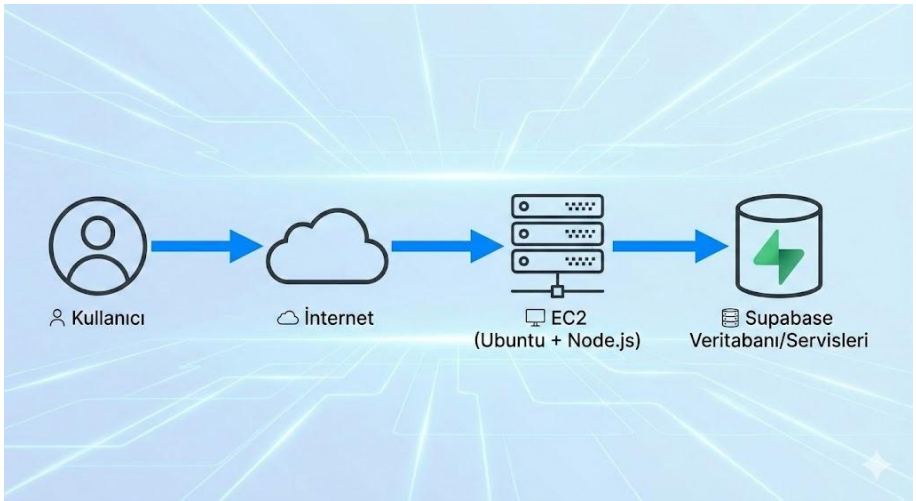
EC2 oluşturulurken kullanılan temel ayarlar aşağıdaki gibidir:

- AMI: Ubuntu Server Pro 24.04 LTS (HVM), EBS General Purpose (SSD) Volume Type.
- Instance tipi: c7i-flex.large (2 vCPU, 4 GB RAM)
- Public IPv4: Enabled
- Key Pair: SSH HTTP HTTPS anahtarları
- Depolama: EBS SSD

6. Uygulama Mimari Şeması

Projenin yüksek seviye mimarisi aşağıdaki gibidir. Kullanıcı tarayıcı üzerinden EC2 public IP adresine HTTP (port 80) ile istek gönderir. EC2 üzerinde Node.js/Vite tabanlı uygulama çalışır. Veri saklama ve bazı backend servisleri için Supabase kullanılır.

Mimari Akış: Kullanıcı → İnternet → EC2 (Ubuntu + Node.js) → Supabase Veritabanı/Servisleri



7. Dağıtım Süreci (Adım Adım)

Bu bölümde, uygulamanın AWS üzerinde çalıştırılması için izlenen adımlar komutlar ve açıklamalarla birlikte sunulmuştur.

7.1 Sistem Güncelleme

Kurulum öncesinde paket listeleri güncellenmiş ve sistem paketleri yükseltilmiştir. Bu adım, güvenlik yamaları ve bağımlılık uyumluluğu açısından önemlidir.

```
sudo apt update && sudo apt upgrade -y
```

7.2 Node.js 20 LTS Kurulumu

Uygulama Node.js ortamında çalıştığı için Node.js 20 LTS kurulmuştur. Ubuntu depolarındaki sürüm yerine NodeSource kurulumu tercih edilerek güncel ve uyumlu LTS sürüm kullanılmıştır. Ayrıca GitHub repo klonlamak için gh aracı kurulmuştur.

7.3 Port 80 Erişimi (iptables)

Uygulamanın kullanıcılar tarafından port yazmadan erişilebilmesi için HTTP standart portu olan 80 hedeflenmiştir. Security Group tarafında port 80 açık olduğundan ayrıca sunucu üzerinde firewall (iptables) kuralı eklenmiştir.

```
sudo iptables -A INPUT -p tcp --dport 80 -j ACCEPT
```

Not: iptables kuralı yeniden başlatmalarda kalıcı olmayabilir. Kalıcı yapılandırma için ufw veya iptables-persistent kullanılabilir (bkz. Bölüm 10).

7.4 Projenin GitHub’dan Sunucuya Taşınması

Kaynak kod GitHub üzerinde tutulmuştur. Sunucu tarafında repo klonlanarak aynı kod tabanı üzerinden dağıtım yapılmıştır.

```
gh repo clone khaled-t7s/Courtbookingwebsite
cd Courtbookingwebsite
ls
```

7.5 Bağımlılıkların Kurulması (npm install)

Projedeki bağımlılıklar package.json üzerinden npm ile kurulmuştur. Bu işlem node_modules klasörünü oluşturur ve uygulamanın çalışması için gerekli paketleri indirir.

```
npm install
```

7.6 Uygulamanın Çalıştırılması (Port 80)

Uygulama Vite ile çalıştırılmıştır. Port 80 gibi 1024 altındaki portlar Linux sistemlerde yönetici yetkisi gerektirdiği için uygulama sudo ile başlatılmıştır.

```
sudo npm run dev
```

Uygulama çalıştıktan sonra tarayıcı üzerinden aşağıdaki adres ile erişim test edilmiştir:

```
http://<EC2-PUBLIC-IP>/
```

8. Karşılaşılan Zorluklar ve Çözümler

Dağıtım sürecinde karşılaşılan başlıca sorunlar ve çözümleri aşağıda verilmiştir:

- Port erişimi: Uygulama dışarıdan erişilemediğinde Security Group inbound kuralları kontrol edildi ve HTTP(80) izinleri doğrulandı. Sunucu tarafında iptables ile 80 portu için kural eklendi.
- Paket yönetimi karmaşası: npm'in Node.js ile birlikte geldiği doğrulandı; bağımlılıklar npm install ile kuruldu. Apt üzerinden npm kurma girişiminde bağımlılık sorunları gözlemlendiği için NodeSource yöntemi tercih edildi.

9. Öğrenilen Dersler ve Olası İyileştirmeler

9.1 Öğrenilen Dersler

Bu çalışma sonucunda edinilen başlıca kazanımlar:

- AWS EC2 üzerinde sanal sunucu oluşturma ve temel kaynak yönetimi
- Node.js tabanlı uygulamaların Linux ortamında kurulumu ve çalıştırılması
- GitHub üzerinden versiyon kontrol ve sunucuya proje taşıma
- Supabase gibi yönetilen veritabanı servislerine entegrasyon

9.2 Olası İyileştirmeler

Proje daha üretim ortamına yakın hale getirilmek istenirse şu iyileştirmeler önerilir:

- Nginx reverse proxy kullanarak port yönlendirme ve daha stabil servis yönetimi
- HTTPS (443) ve domain ile TLS/SSL sertifikası (Let's Encrypt) ekleme
- Uygulamayı PM2 veya systemd ile arka planda sürekli çalışır hale getirme
- Firewall kurallarını kalıcı hale getirme (ufw / iptables-persistent)
- GitHub Actions ile otomatik deploy (CI/CD)
- Uygulama için özel bir domain satın alınarak EC2 public IP adresine yönlendirilmesi ve böylece daha profesyonel ve kullanıcı dostu bir erişim sağlanması.

10. Görev Dağılımı ve Katkılar

Bu proje grup çalışması olarak gerçekleştirilmiştir. Aşağıda her bir grup üyesinin projedeki görev ve sorumluluk alanları belirtilmiştir:

KHALED MAHYOUB :

- AWS hesabının oluşturulması ve EC2 sanal sunucusunun kurulması
- EC2 yapılandırmaları (AMI seçimi, instance tipi, security group ayarları)
- Mimari yapının belirlenmesi ve raporlanması

KHALED GAMAL MAHMOUD ZORAEI MOHAMED :

- Sunucu ortamının hazırlanması sistem güncellemeleri
- Node.js kurulumu ve uygulama bağımlılıklarının yüklenmesi
- Projenin GitHub üzerinden sunucuya taşınması ve çalıştırılması
- Port ve erişim ayarlarının yapılması

11. Bağlantılar

GitHub Repo: https://github.com/khaled-t7s/Courtbookingwebsite

YouTube Video: https://youtu.be/wsvYtKjCd14