

ISE 465 - Bulut Bilişim Proje Raporu

AWS Üzerinde Konteyner Tabanlı Bulut Depolama Sistemi (Nextcloud)

HAMZA AKBAŞ – B211200056

EREN PORSUK – B211200010

OSMAN CAN – B211200053

YouTube video link : <https://youtu.be/b0V2qGZtoJ4>

Uygulama link : <http://18.195.52.139:8080/login?clear=1>

Github link : <https://github.com/hmzakbss/Sakarya-ISE465-Cloud-Project-Nextcloud.git>



**SAKARYA
ÜNİVERSİTESİ**

İçindekiler

1. Proje Açıklaması ve Hedefleri.....	3
2. Uygulama ve Bulut Platformu Seçimi.....	3
Uygulama Seçimi: Nextcloud.....	3
Bulut Platformu: Amazon AWS	3
3. Uygulama Mimari Şeması.....	3
4. Dağıtım Süreci.....	4
4.1: AWS Ortamının Hazırlanması.....	4
4.2: Sunucu Konfigürasyonu	4
4.3: Konteyner Orkestrasyonu.....	5
4.4: Servisin Başlatılması	5
5. Otomasyon için Kod Parçacıkları.....	5
6. Karşılaşılan Zorluklar ve Çözümler	6
7. Öğrenilen Dersler ve Olası İyileştirmeler.....	6
8. Grup Üyeleri ve Görev Dağılımı.....	7
KAYNAKÇA	8

1. Proje Açıklaması ve Hedefleri

Bu projenin temel amacı, üçüncü taraf veri depolama hizmetlerine (Google Drive, Dropbox vb.) alternatif olarak, kontrolü tamamen kullanıcıya ait olan, güvenli ve ölçeklenebilir bir **Kişisel Bulut Depolama (SaaS)** çözümü oluşturmaktır.

Proje kapsamında, **Amazon Web Services (AWS)** altyapısı kullanılarak sanal bir sunucu (**EC2**)ayağa kaldırılmış ve uygulama dağıtımını modern **Konteynerizasyon (Docker)** teknolojileri ile gerçekleştirmiştir. Hedeflenen kazanımlar şunlardır:

- **IaaS** (Hizmet Olarak Altyapı) katmanında sunucu yönetimi deneyimi kazanmak.
- Mikroservis mimarisini anlayarak uygulama ve veritabanını izole konteynerlerde çalışırmak.
- Bulut güvenliği (Security Groups) ve ağ yapılandırması üzerinde pratik yapmak.

2. Uygulama ve Bulut Platformu Seçimi

Uygulama Seçimi: Nextcloud

Proje için açık kaynak kodlu **Nextcloud** platformu seçilmiştir. Seçim nedenleri:

- **Veri Egemenliği:** Verilerin nerede saklandığı kullanıcı kontrolündedir.
- **Modüler Yapı:** Dosya saklamanın yanı sıra takvim, rehber ve iş birliği araçları sunar.
- **Grup Çalışmasına Uygunluk:** Çoklu kullanıcı desteği sayesinde grup üyeleri ortak klasörlerde çalışabilir.



Bulut Platformu: Amazon AWS

Altyapı sağlayıcısı olarak **AWS (Amazon Web Services)** tercih edilmiştir.



- **EC2 (Elastic Compute Cloud):** Esnek ve ölçeklenebilir işlem gücü sunduğu için kullanılmıştır.
- **Maliyet Etkinliği:** t2.micro instance tipi kullanılarak "Free Tier" (Ücretsiz Katman) sınırları içinde kalınmıştır.

3. Uygulama Mimari Şeması

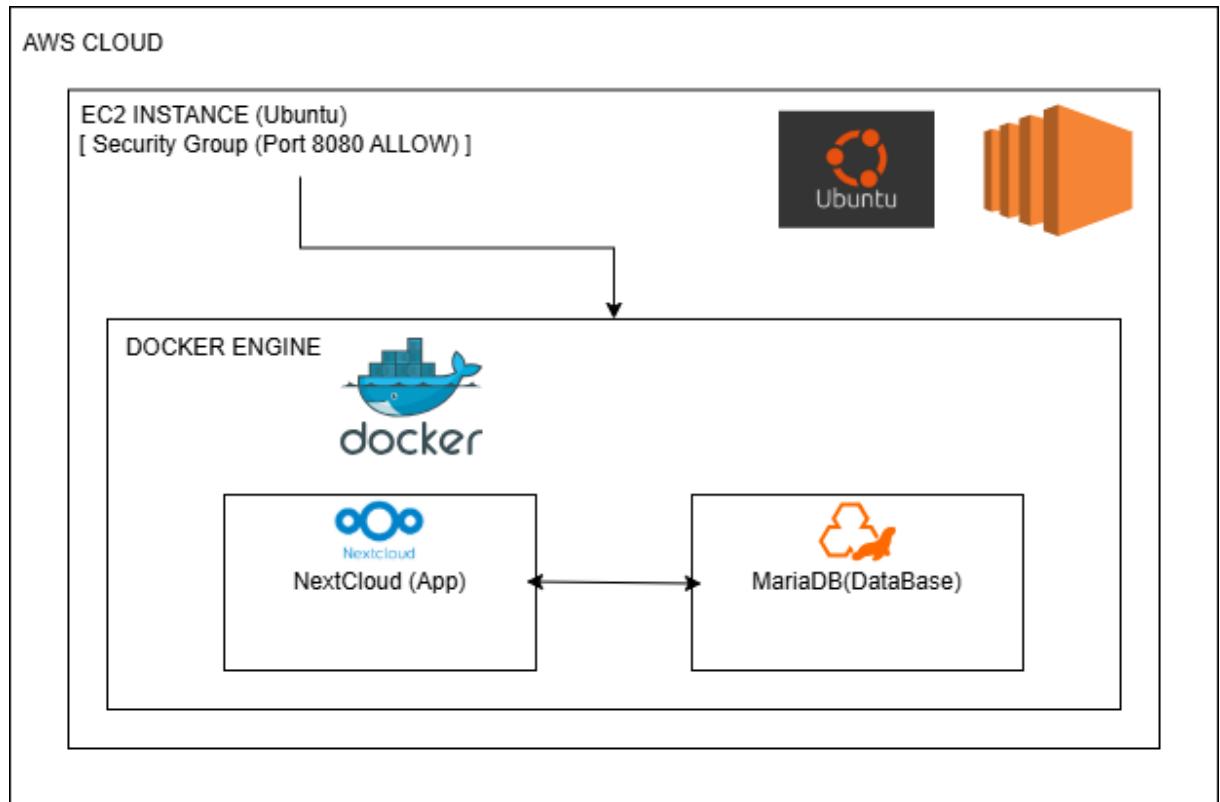
Projemiz, **İki Katmanlı (Two-Tier)** bir mimariye sahiptir.

- 1.1. **İstemci (Client):** Web tarayıcılarından HTTP isteği gönderir.
- 1.2. **AWS Security Group:** Firewall görevi görerek sadece 22 (SSH) ve 8080 (Web) portlarına izin verir.
- 1.3. **EC2 Instance (Ubuntu):** Tüm sistemin çalıştığı sanal sunucudur.

1.4. **Docker Engine:** Uygulama konteynerlerini yöneten platformdur.

1.5. **Konteynerler:**

- **App Container:** Nextcloud web arayüzü ve uygulama mantığı.
- **DB Container:** MariaDB veritabanı. Bu iki konteyner kendi aralarında özel bir sanal ağ üzerinden haberleşir.



4. Dağıtım Süreci

4.1: AWS Ortamının Hazırlanması

- AWS Konsolu üzerinden **Ubuntu Server 24.04 LTS** işletim sistemine sahip bir EC2 sunucusu başlatıldı.
- SSH bağlantısı için .pem uzantılı bir anahtar çifti (Key Pair) oluşturuldu.
- **Security Group** ayarlarından 8080 numaralı port dış dünyaya (0.0.0.0/0) açıldı.

NOT : EC2 instance özellikleri ile ilgili bilgilendirme videoda yapılacaktır.

4.2: Sunucu Konfigürasyonu

SSH protokolü ile sunucuya bağlanıldı ve gerekli bağımlılıklar yüklandı:

```
sudo apt update  
sudo apt install docker.io docker-compose -y
```

4.3: Konteyner Orkestrasyonu

Uygulamanın ve veritabanının birbiriyle uyumlu çalışması için docker-compose.yml dosyası hazırlandı. Bu dosya sayesinde manuel kurulum yerine "Infrastructure as Code" (Kod olarak Altyapı) prensibi uygulandı.

4.4: Servisin Başlatılması

Hazırlanan otomasyon dosyası tek bir komutla çalıştırıldı:

```
sudo docker-compose up -d
```

5. Otomasyon için Kod Parçacıkları

Projede kullanılan docker-compose.yml dosyası aşağıdadır. Bu kod, Nextcloud ve MariaDB servislerini tanımlar, birbirine bağlar ve verilerin kalıcı olması için "Volume" yapılandırmasını içerir.

```
version: '2'

services:
  db:
    image: mariadb:10.5
    restart: always
    command: --transaction-isolation=READ-COMMITTED --binlog-format=ROW
    volumes:
      - db_data:/var/lib/mysql
    environment:
      - MYSQL_ROOT_PASSWORD=admin123
      - MYSQL_PASSWORD=admin123
      - MYSQL_DATABASE=nextcloud
      - MYSQL_USER=nextcloud
  app:
    image: nextcloud
    restart: always
    ports:
      - 8080:80
    links:
      - db
    volumes:
      - nextcloud_data:/var/www/html
    environment:
      - MYSQL_PASSWORD=admin123
      - MYSQL_DATABASE=nextcloud
      - MYSQL_USER=nextcloud
      - MYSQL_HOST=db
```

volumes:

db_data:

nextcloud_data:

6. Karşılaşılan Zorluklar ve Çözümler

Proje geliştirme sürecinde aşağıdaki teknik aksaklıklar yaşanmış ve çözülmüştür:

1. **SSH İzin Hatası:** Windows ortamında .pem anahtar dosyası kullanılırken "*Permissions are too open*" hatası alındı.
 - o *Çözüm:* Dosya özelliklerinden güvenlik izinleri düzenlenerek, dosya sahibinden başka tüm kullanıcıların erişim yetkisi kaldırıldı.
2. **Erişim Problemi:** Konteynerler çalışmasına rağmen tarayıcıdan siteye erişilemedi.
 - o *Çözüm:* AWS Security Group ayarlarında 8080 portunun kapalı olduğu fark edildi ve "Inbound Rules" güncellenerek port trafiğe açıldı.

7. Öğrenilen Dersler ve Olası İyileştirmeler

Öğrenilenler:

- Bulut bilişimde güvenlik (Firewall/Security Group) ayarlarının, en az sunucu kurulumu kadar kritik olduğu anlaşıldı.
- Docker Compose kullanmanın, uygulamayı taşınabilir hale getirdiği ve kurulum süresini dakikalara indirdiği deneyimlendi.

Olası İyileştirmeler:

- **SSL Sertifikası:** Şu an HTTP üzerinden çalışan sisteme Let's Encrypt ile HTTPS desteği eklenebilir.
- **S3 Entegrasyonu:** Dosyaların EC2 diski yerine Amazon S3 bucket'larında tutulması sağlanarak sınırsız depolama alanına ulaşılabilir.

8. Grup Üyeleri ve Görev Dağılımı

Bu proje üç kişilik bir grup çalışması olarak gerçekleştirilmiştir. Her bir grup üyesi proje sürecinin farklı aşamalarında aktif rol almıştır.



1. HAMZA AKBAŞ

- AWS EC2 sunucusunun kurulumu ve yapılandırılması.
- Security Group (Güvenlik Duvarı) kurallarının yönetimi (Port 8080/22).
- IP yapılandırması ve ağ optimizasyonu.
- Video sunumunun kurgusu ve proje teslim süreci.



2. EREN PORSUK

- Docker ve Docker Compose ortamının sunucuya kurulması.
- docker-compose.yml otomasyon dosyasının kodlanması.
- Mikroservis mimarisinin (App + Database) kurgulanması.
- Konteynerlerin performansı ve kaynak tüketiminin (CPU/RAM) takibi.



3. OSMAN CAN

- Nextcloud uygulama içi ayarların yapılması (Admin paneli).
- SSH anahtar yönetimi ve sunucu erişim güvenliğinin sağlanması.
- Kullanıcı senaryolarının (Dosya yükleme, kullanıcı açma) test edilmesi.
- Proje raporunun teknik dokümantasyonu ve mimari şemanın çizimi.

KAYNAKÇA

Amazon Web Services. (2025). *What is Amazon EC2?* Amazon Web Services Documentation. Erişim tarihi: 28 Aralık 2025, <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html>

Docker Inc. (2025). *Get started with Docker.* Docker Documentation. Erişim tarihi: 28 Aralık 2025, <https://docs.docker.com/get-started/>

MariaDB Foundation. (2025). *MariaDB Server* (Versiyon 10.5) [Docker Image]. Docker Hub. Erişim tarihi: 28 Aralık 2025, https://hub.docker.com/_/mariadb

Nextcloud GmbH. (2025). *Installation on Docker.* Nextcloud Server Administration Guide. Erişim tarihi: 28 Aralık 2025, https://docs.nextcloud.com/server/latest/admin_manual/installation/docker/