## Yüksek Kullanılabilirlik ve Olay Kurtarma:

**Proje:** Bu proje, bir web uygulamasının kesintisiz çalışmasını sağlamak ve veri kaybı riskini azaltmak için yüksek kullanılabilirlik ve olay kurtarma önlemleri uygulamayı hedeflemektedir.

**Proje Açıklaması:** Bu proje, yüksek kullanılabilirlik ve olay kurtarma sağlamak için Amazon Web Services (AWS) kullanılarak bir web uygulamasının altyapısını oluşturmayı amaçlamaktadır. Projede, çoklu bölgelerde çalışan bir Amazon RDS veritabanı, S3 ile dağıtılan statik içerik, CloudFront ile içerik dağıtım ağı yapılandırması, Elastic Beanstalk veya Lambda ile uygulama sunucusu dağıtımı ve veri senkronizasyonu ve yedekleme için hizmetler kullanılacaktır.

#### **Projenin Temel Adımları:**

- Multi-AZ yapılandırmasına sahip Amazon RDS veritabanı örneği oluşturun.
- S3 kullanarak web uygulamasının statik içeriğini dağıtılması.
- AWS CloudFront ile içerik dağıtım ağı (CDN) yapılandırması yapılması.
- AWS Elastic Beanstalk veya AWS Lambda gibi hizmetlerle uygulama sunucusunu dağıtılması.

## Proje İçin Kullanılacak Amazon Hizmetleri

- Amazon RDS
- Amazon S3
- AWS CloudFront
- AWS Elastic Beanstalk
- AWS Lambda

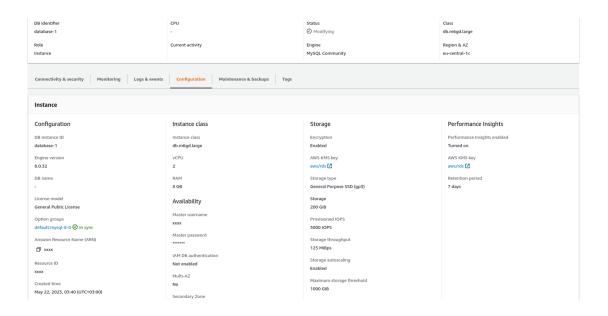
#### **Amazon Database**

AWS tarafından sunulan çeşitli veritabanı hizmetlerini içeren bir koleksiyondur.

- 1. Amazon RDS (Relational Database Service): İlişkisel veritabanları için sunulan hizmettir. MySQL, PostgreSQL, Oracle, SQL Server ve Amazon Aurora gibi popüler veritabanı motorlarını destekler.
- 2. Amazon DynamoDB: Tamamen yönetilen, hızlı ve ölçeklenebilir bir NoSQL veritabanıdır. Yüksek performans, otomatik ölçeklendirme ve esnek veri modelleme özellikleri sunar.
- 3. Amazon ElastiCache: İşlem hızını artırmak için yönetilen bir önbellek hizmetidir. Memcached ve Redis gibi popüler önbellek motorlarını destekler.
- 4. Amazon Neptune: Grafları temel alan yönetilen bir graf veritabanıdır. İlişkisel bağımlılıkları olan verileri depolamak ve sorgulamak için idealdir. Amazon Redshift: Veri ambarı hizmetidir. Büyük miktarda veriyi depolamak, işlemek ve analiz etmek için kullanılır. Paralel sorgulama yetenekleri ve kolay entegrasyon özellikleri sunar.
- 5. Amazon Neptune: Grafları temel alan yönetilen bir graf veritabanıdır. İlişkisel bağımlılıkları olan verileri depolamak ve sorgulamak için idealdir.Amazon Redshift: Veri ambarı hizmetidir. Büyük miktarda veriyi depolamak, işlemek ve analiz etmek için kullanılır. Paralel sorgulama yetenekleri ve kolay entegrasyon özellikleri sunar.

#### Multi-AZ Yapılandırmasına Sahip Amazon RDS Veritabanı Örneği Oluşturulması

Amazon Management System üzerinde RDS olarak arama yapılıp yeni bir database oluşturmak için gerekli adımlar oluşturulur. Veritabanı içerisinde istenilen veritabanı seçilir, kullanıcı adı – parola gibi gereksinimler girildikten sonra "Availability & durability" alanında "Multi-AZ DB Instance" seçeneği seçilir. Multi-AZ DB Cluster, Amazon Aurora gibi dağıtık veritabanı motorlarında kullanılan bir yapılandırmadır. Bu yapılandırmada, veritabanı birincil (primary) ve yedek (replica) olmak üzere en az iki Availability Zone'da (AZ) bulunan veritabanı örneklerinden oluşur. Multi-AZ DB Instance ise Amazon RDS'in diğer veritabanı motorları için sunulan bir yüksek kullanılabilirlik seçeneğidir. Bu yapılandırmada, birincil veritabanı örneği ve yedek örneği birbirinden farklı Availability Zone'larında yer alır. Birincil örnek üzerindeki veri ve yapılandırma otomatik olarak yedek örneğe aktarılır ve yedek örnek, birincil örnek hizmet dışı kalırsa otomatik olarak devreye alınır.





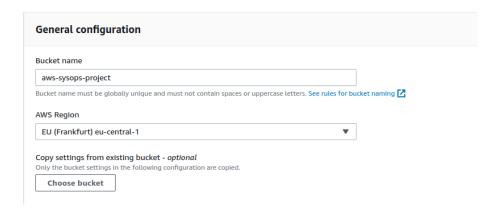
#### **AMAZON S3 (Simple Storage Service)**

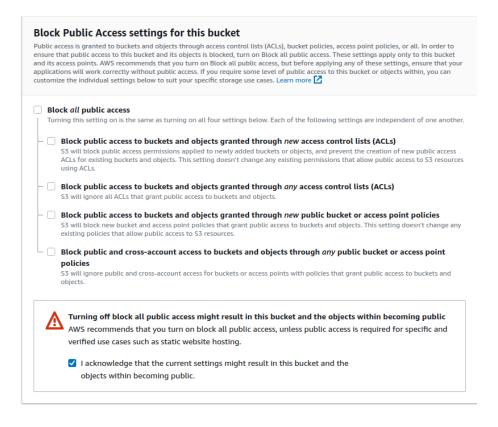
Amazon S3, bulut tabanlı bir depolama sistemidir. Kullanıcıların verileri depolamasına, yönetmesine ve bu verilere güvenli şekilde erişebilmesine olanak tanır.

S3 tanımlaması yaparken default olarak, dış ağdan gelen bütün istekler bloklanır.

- Block public access to buckets and objects granted through new access control lists (ACLs): Bu
  seçenek, yeni eklenen kovalar veya nesneler üzerinden verilen erişim kontrol listeleri (ACL'ler) ile
  halka açık erişimi engeller. Bu ayar, mevcut ACL'ler aracılığıyla S3 kaynaklarına halka açık erişim
  izni veren herhangi bir mevcut izni değiştirmez.
- Block public access to buckets and objects granted through any access control lists (ACLs): Bu
  seçenek, kovalar ve nesneler üzerinde halka açık erişim izni veren tüm ACL'leri görmezden gelir.
- Block public access to buckets and objects granted through new public bucket or access point
  policies: Bu seçenek, yeni kova ve erişim noktası politikaları aracılığıyla kovalar ve nesneler
  üzerinde halka açık erişim izni verilmesini engeller. Bu ayar, S3 kaynaklarına halka açık erişim izni
  veren mevcut politikaları değiştirmez.
- Block public and cross-account access to buckets and objects through any public bucket or access point policies: Bu seçenek, kovalar veya erişim noktaları üzerinde halka açık erişim izni veren politikaları görmezden gelir ve bu politikalar aracılığıyla yapılan hesaplar arası erişimi de engeller.

Amazon S3 hizmetinden create bucket seçeneği seçilerek yeni bir bucket oluşturulması sağlanır. Bucket için isim seçilir, "*Block* all *public access*" seçenekleri disable edilir ve diğer tüm ayarlar default olarak bırakılır.





Seçilen bucketa gidilerek istenilen dosya bucketa yüklenir. Dosyalar yüklendikten sonra ilgili bucket adresine gidildiği zaman "Access Denied" hatası verecektir. Bunun birkaç sebebi olabilir fakat bu projede daha bucket policy eklenmediği için "Access Denied" hatası alırız. Bucket policy, S3 bucket'ına gelen istekleri nasıl işleyeceğini ve kimlerin hangi eylemleri gerçekleştirebileceğini kontrol etmek için kullanılır. Bucket policy eklemek için ilgili bucket sekmesinden "Permissions" tabında "Bucket Policy" seçeneğinde "Edit" seçilir ve altta verilen kod parçacığı bucket policy'e eklenir.

Version: JSON politika belirtimi için kullanılan versiyon numarasını belirtir.

Sid (Statement ID): Deklarasyonun benzersiz bir kimliğidir.

Effect: İzinlerin nasıl uygulanacağını belirler.

Principal: Eylemi gerçekleştiren kimliği belirler. Örneğin, belirli bir IAM kullanıcısı, IAM rolü veya AWS hesabı olabilir.

Action: İzin verilen veya reddedilen eylemleri belirtir.

Resource: Erişim izninin uygulanacağı İlgili bucket kaynağını belirtir.

İlgili işlemler yapıldıktan sonra S3 bucketa URL ile erişilebilinir.

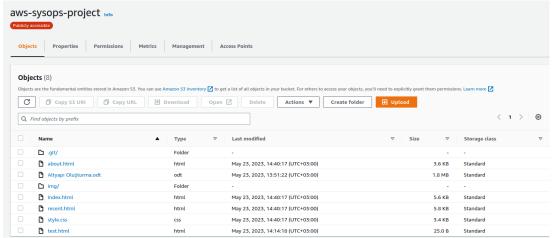
## Amazon EC2 Instance'ın S3 Bucketa Bağlanması

EC2 instance'ı S3 bucketa bağlamak için EC2 instance'ına ssh ile bağlanıp awscli kurmak ve aws config ayarlarına access key ve secret access keylerini eklemek gerekmektedir.

SSH ile bağlandıktan sonra "yum install awscli" komutu ile awscli aracı kurulur. "aws configure" komutu ile gerekli bilgiler doldurulur. "aws s3 cp /path/to/local/file s3://your-bucket-name/" ile instance içerisindeki dosyalar s3 bucket'ına aktarımı sağlanır.

```
[root@ip-i0-0-1-218 git]E aws s3 cp./Responsive_Blog_Mebsite_si://aws-sysops-project --recursive upload: Responsive_Blog_Mebsite_sit/description to s3://aws-sysops-project_sit/config upload: Responsive_Blog_Mebsite_sit/config to s3://aws-sysops-project_sit/config upload: Responsive_Blog_Mebsite_sit/config upload: Responsive_Blog_Mebsite_sit/refab_to s3://aws-sysops-project_sit/config upload: Responsive_Blog_Mebsite_sit/refab_to s3://aws-sysops-project_sit/fab_upload: Responsive_Blog_Mebsite_sit/conks/pre-commit.sample to s3://aws-sysops-project_sit/fab_upload: Responsive_Blog_Mebsite_sit/conks/pre-push.sample to s3://aws-sysops-project_sit/fab_upload: Responsive_Blog_Mebsite_sit/conks/pre-push.sample to s3://aws-sysops-project_sit/fab_upload: Responsive_Blog_Mebsite_sit/conks/pre-push.sample to s3://aws-sysops-project_sit/conks/pre-push.sample upload: Responsive_Blog_Mebsite_sit/conks/pre-push.sample to s3://aws-sysops-project_sit/conks/pre-push.sample upload: Responsive_Blog_Mebsite_sit/conks/pre-nerge-commit.sample to s3://aws-sysops-project_sit/conks/pre-nerge-commit.sample upload: Responsive_Blog_Mebsite_sit/conks/pre-nerge-commit.sample to s3://aws-sysops-project_sit/conks/pre-nerge-commit.sample upload: Responsive_Blog_Mebsite_sit/conks/pre-nerge-commit.sample to s3://aws-sysops-project_sit/conks/pre-nerge-commit.sample upload: Responsive_Blog_Mebsite_sit/conks/pre-nerge-commit.sample to s3://aws-sysops-project_sit/conks/pre-nerge-commit.sample upload: Responsive_Blog_Mebsite_sit/conks/prepare-commit-nsg.sample upload: Responsive_Blog_Mebsite_sit/conks/prepare-commit-nsg.sample to s3://aws-sysops-project_sit/conks/prepare-commit-nsg.sample upload: Responsive_Blog_Mebsite_sit/conks/prepare-commit-nsg.sample upload: Responsive_Blog_Mebsite_sit/conks/prepare-commit-nsg.sample upload: Responsive_Blog_Mebsite_sit/conks/prepare-commit-nsg.sample upload: Responsive_Blog_Mebsite_sit/conks/prepare-commit-nsg.sample upload: Responsive_Blog_Mebsite_sit/conks/prepare-commit-nsg.sit/aws-sysops-project_sit/conks/prepar
```

EC2 instance'a bir blog post indirdim ve bu blog postu s3 bucketa yükledim.



# Living The Social Life

Home About me Recent posts



July 23, 2019 | 3 comments

#### Finding simplicity in life

Life can get complicated really quickly, but it doesn't have to be! There are many ways to simplify your life, a few of which we've explored in the past. This week we're taking a bit of a approach through, in how you can find simplicity in the life you already leaving.

CONTINUE READING

Keeping cooking simple



I find life better, and I'm happier, when things are nice and simple.



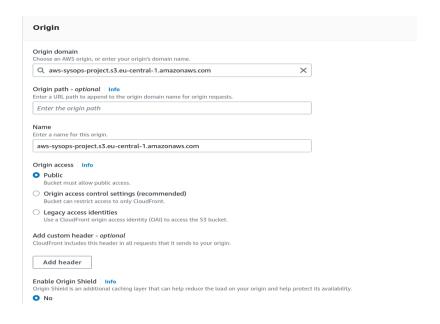


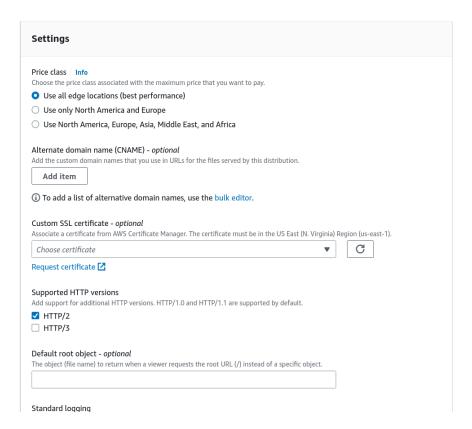


## Amazon CloudFront ile İçeriğin Dağıtılması

CloudFront, kullanıcıların isteklerini en yakın yerde bulunan önbellek sunucularına yönlendirerek içeriği daha hızlı teslim eder.

CloudFront sayfasında "Create Distrubition" seçeneği seçilerek CloudFront oluşturulur. Origin name seçeneğinde ilgili s3 bucketı seçilir. Diğer gerekli şekilde yapılandırılır. Price class seçeneğinde "Use all edge location" seçeneği seçilir.Price class, CloudFront'un fiyatlandırma seviyesini belirlemek için kullanılan bir seçenektir. Bu seçenek, CloudFront'un hizmet verdiği edge konumlarının kapsamını ve buna bağlı olarak fiyatlandırmayı etkiler. Use all edge locations (best performance): Bu seçenek, CloudFront'un tüm edge konumlarını kullanmasını ve böylece içeriği dünya genelinde en hızlı şekilde dağıtmasını sağlar. Bu seçenek, en yüksek performansı sunar, ancak maliyetler daha yüksek olabilir.





#### AWS Elastic Beanstalk veya AWS Lambda Gibi Hizmetlerle Uygulama Sunucusunu Dağıtılması.

#### **AWS Elastic Beanstalk**

AWS Elastic Beanstalk, uygulama geliştirme sürecini hızlandıran ve kolaylaştıran bir hizmettir. Elastic Beanstalk, geliştiricilere uygulama kodlarını yüklemeleri, yapılandırmaları ve ölçeklemeleri için otomatikleştirilmiş bir platform sağlar. AWS Lambda ise olay tetiklemeli bir hesaplama hizmetidir. Lambda, geliştiricilere kodlarını çalıştırmak için bir altyapı sağlar, böylece geliştiriciler sadece kodlarını yükler ve Lambda'nın olay tetiklemesine yanıt olarak kodu çalıştırmasını sağlar.

AWS Elastic Beanstalk ile uygulama sunucusu dağıtmak için aşağıdaki adımları izleyebilirsiniz:

- 1. Elastic Beanstalk konsoluna giriş yapın veya AWS CLI kullanarak Elastic Beanstalk uygulamanızı oluşturun.
- 2. Uygulama kodunuzu Elastic Beanstalk'e yükleyin.
- 3. Elastic Beanstalk, otomatik olarak uygulama kodunu altyapıya dağıtır ve yönetir.
- 4. Elastic Beanstalk, uygulamanızı ölçeklendirir ve yüksek kullanılabilirlik sağlar.
- 5. Elastic Beanstalk konsolu veya AWS CLI üzerinden uygulamanızı yönetebilir ve izleyebilirsiniz.

AWS Lambda ile uygulama sunucusu dağıtmak için aşağıdaki adımları izleyebilirsiniz:

- 1. AWS Lambda konsoluna giriş yapın veya AWS CLI kullanarak Lambda fonksiyonunuzu oluşturun.
- 2. Kodunuzu Lambda fonksiyonuna yükleyin ve tetikleme olayını belirleyin.
- 3. Lambda, tetikleme olayı gerçekleştiğinde kodunuzu otomatik olarak çalıştırır.
- 4. Lambda, işlevin ölçeklenmesini ve yönetimini otomatik olarak sağlar.
- 5. Lambda fonksiyonunuzun sonuçlarını takip edebilir ve izleyebilirsiniz.