



**SAKARYA
ÜNİVERSİTESİ**

**BİLİŞİM SİSTEMLERİ MÜHENDİSLİĞİ TASARIMI
BULUT BİLİŞİM**

Danışman : Öğr.Gör. UĞUR ÖZBEK

Yahya Elserhan – B221200556

Video Link : <https://youtu.be/ZgGAYWBddul>

İÇİNDEKİLER

- 1. Proje Açıklaması ve Hedefleri**
- 2. Uygulama Seçimi**
- 3. Kullanılan Teknolojiler**
- 4. Bulut Platformu Seçimi**
 - 4.1. Amazon S3'nin Seçilme Nedeni**
 - 4.2. Amazon CloudFront'un Seçilme Nedeni**
- 5. Uygulama Mimari Şeması**
 - 5.1. Mimari Yapının Genel Akışı**
 - 5.2. Mimari Bileşenlerin Açıklaması**
- 6. Dağıtım Süreci (Adım Adım Uygulama Notları)**
 - 6.1. AWS Üzerinde Maliyet Kontrolü ve Bütçe Yapılandırması**
 - 6.2. Neden Amazon S3 Tercih Edildi?**
 - 6.3. Amazon S3 Bucket Oluşturulması**
 - 6.4. Uygulama Dosyalarının Amazon S3 Bucket'a Yüklenmesi**
 - 6.5. Amazon S3 Üzerinde Static Website Hosting'in Etkinleştirilmesi**
 - 6.6. Static Website Hosting Etkinleştirilmesine Rağmen Erişim Engeli (403 Forbidden)**
 - 6.7. Genel Erişim için Amazon S3 Bucket Policy Yapılandırması**
 - 6.8. Web Sitesinin AWS Üzerinde Başarıyla Yayınlanması**
 - 6.9. HTTPS Desteği Sağlama ve Amazon CloudFront'un Tercih Edilme Nedeni**
 - 6.10. Amazon CloudFront Distribution Oluşturulması**
 - 6.11. CloudFront Origin Tanımlama (S3 Bucket Bağlantısı)**
 - 6.12. CloudFront Güvenlik Yapılandırması (AWS WAF Seçilmemesi)**
 - 6.13. CloudFront Dağıtımının Oluşturulması (Review and Create)**
 - 6.14. CloudFront Dağıtımının Başarıyla Oluşturulması ve HTTPS Yapılandırmasının Tamamlanması**
- 7. Öğrenilen Dersler ve Olası İyileştirmeler**
- 8. Özeti**

1. Proje Açıklaması ve Hedefleri

Bu proje kapsamında, Bulut Bilişim dersi çerçevesinde basit bir e-ticaret web sitesinin bulut ortamında dağıtımı ve yayınlanması amaçlanmıştır. Geliştirilen uygulama; HTML, CSS ve JavaScript kullanılarak hazırlanmış, çok sayfalı (multi-page) statik bir web sitesidir. Uygulama herhangi bir sunucu tarafı (backend) bileşeni içermemekte olup yalnızca istemci tarafında çalışmaktadır.

Projenin temel amacı, modern bulut bilişim teknolojilerini kullanarak bir web uygulamasının gerçek bir bulut altyapısı üzerinde nasıl dağıtıldığını ve yayınlandığını uygulamalı olarak göstermektedir. Bu kapsamında Amazon Web Services (AWS) platformu tercih edilmiş ve AWS Free Tier imkânları kullanılarak maliyetsiz bir dağıtım gerçekleştirilmiştir.

Proje hedefleri aşağıdaki şekilde özetlenebilir:

- Statik bir web uygulamasının bulut ortamında barındırılmasını sağlamak
- Amazon S3 servisi kullanılarak web dosyalarının güvenli bir şekilde depolanması
- Amazon CloudFront servisi aracılığıyla içeriğin global olarak dağıtılması
- Web sitesinin HTTPS protokolü üzerinden güvenli erişime açılması
- Bulut bilişimde dağıtım (deployment), erişim kontrolü ve güvenlik kavramlarının pratikte uygulanması

Bu proje sayesinde, bulut tabanlı mimarilerin temel çalışma prensipleri öğrenilmiş ve gerçek bir senaryo üzerinden AWS servislerinin entegrasyonu deneyimlenmiştir.

2. Uygulama Seçimi

Bu proje kapsamında dağıtımları gerçekleştirilecek uygulama olarak **statik bir e-ticaret web sitesi** seçilmiştir. Seçilen uygulama; HTML, CSS ve JavaScript teknolojileri kullanılarak geliştirilmiş, çok sayfalı (multi-page) bir web sitesidir. Uygulama herhangi bir sunucu tarafı işlem, veritabanı veya API içermemekte olup tamamen istemci tarafında çalışmaktadır.

Uygulamanın ana giriş sayfası **online_market.html** dosyasıdır ve site birden fazla HTML sayfası arasında gezinme (navigation) desteği sunmaktadır. Bu yapı, uygulamanın statik web sitesi barındırma senaryoları için uygun olmasını sağlamaktadır.

Bu uygulamanın seçilme nedenleri aşağıda belirtilmiştir:

- Statik yapıda olması sayesinde bulut ortamında kolay ve hızlı dağıtım imkânı sunması
- Sunucu yönetimi gerektirmemesi ve maliyet açısından avantajlı olması
- Bulut bilişimde sıkça kullanılan **static website hosting** yaklaşımını uygulamalı olarak göstermeye uygun olması
- Amazon S3 ve CloudFront servisleri ile doğrudan uyumlu bir mimariye sahip olması

Bu seçim ile birlikte, bulut bilişimde statik web sitelerinin nasıl barındırıldığı, erişime açıldığı ve güvenli hale getirildiği pratik bir örnek üzerinden incelenmiştir.

3. Kullanılan Teknolojiler

Bu projede, statik bir web sitesinin bulut ortamında dağıtıımı ve güvenli bir şekilde yayınlanması amacıyla çeşitli yazılım ve bulut teknolojileri kullanılmıştır. Kullanılan teknolojiler aşağıda kategoriler hâlinde sunulmuştur.

Frontend Teknolojileri

- **HTML:** Web sayfalarının yapısal tasarımını için kullanılmıştır.
- **CSS:** Web sitesinin görsel tasarımını ve kullanıcı arayüzü düzenlemeleri için kullanılmıştır.
- **JavaScript:** Sayfalar arası etkileşim ve dinamik içeriklerin yönetimi için kullanılmıştır.

Bulut Platformu ve Servisleri

- **Amazon Web Services (AWS):** Projenin bulut altyapısı olarak kullanılmıştır.
- **Amazon S3 (Simple Storage Service):** Statik web sitesi dosyalarının depolanması ve barındırılması için kullanılmıştır.
- **Amazon CloudFront:** Web sitesinin HTTPS üzerinden güvenli bir şekilde sunulması ve içeriğin global CDN ağının üzerinden dağıtılmaması için kullanılmıştır.

Güvenlik ve Erişim

- **HTTPS (TLS/SSL):** CloudFront aracılığıyla web sitesine güvenli erişim sağlanmıştır.
- **AWS IAM ve Bucket Policy:** Kaynaklara erişim izinlerinin yönetilmesi için kullanılmıştır.

Sürüm Kontrolü

- **GitHub:** Proje kaynak kodlarının saklanması ve raporun paylaşılması amacıyla kullanılmıştır.

Bu teknolojiler bir arada kullanılarak, statik bir web sitesinin modern bulut mimarisi üzerinde güvenli, ölçeklenebilir ve düşük maliyetli bir şekilde yayınlanması sağlanmıştır.

4. Bulut Platformu Seçimi

Bu projede bulut platformu olarak **Amazon Web Services (AWS)** tercih edilmiştir. AWS, sunduğu geniş servis yelpazesi, yüksek erişilebilirlik, güvenlik olanakları ve eğitim amaçlı kullanıma uygun **Free Tier** seçenekleri sayesinde bu proje için uygun bir bulut altyapısı sunmaktadır.

AWS'nin seçilmesinin başlıca nedenleri aşağıda belirtilmiştir:

- AWS Free Tier kapsamında düşük veya sıfır maliyetle servis kullanım imkânı sunması
- Statik web siteleri için özel olarak tasarlanmış servisler (Amazon S3 ve CloudFront) sağlama
- Global altyapısı sayesinde yüksek erişilebilirlik ve performans sunması

- Güvenlik, erişim kontrolü ve maliyet yönetimi araçlarının entegre olarak sunulması

Amazon S3'nin Seçilme Nedeni

Amazon S3, statik web sitelerinin barındırılması için uygun, ölçeklenebilir ve yönetimi kolay bir depolama servisidir. Projede geliştirilen uygulama sunucu tarafı işlem içermediği için S3 Static Website Hosting özelliği kullanılarak web dosyaları doğrudan internet üzerinden erişilebilir hâle getirilmiştir.

Amazon CloudFront'un Seçilme Nedeni

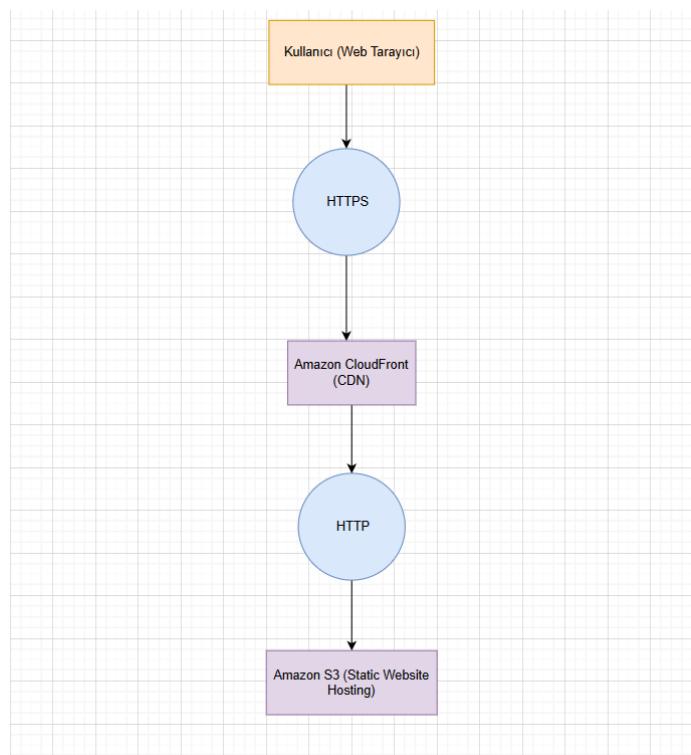
Amazon S3 Static Website Hosting varsayılan olarak HTTPS desteği sunmadığından, web sitesinin güvenli erişimini sağlamak amacıyla **Amazon CloudFront** kullanılmıştır. CloudFront, S3 üzerinde barındırılan içeriği HTTPS protokolü üzerinden sunarak TLS/SSL şifrelemesi sağlamış ve içeriği global CDN ağının üzerinden dağıtarak erişim performansını artırmıştır.

Bu yapı sayesinde web sitesi hem güvenli hem de modern bulut mimarilerine uygun bir şekilde yayınlanmıştır.

5. Uygulama Mimari Şeması

Bu projede kullanılan mimari yapı, statik bir web sitesinin bulut ortamında güvenli ve performanslı şekilde sunulmasına yönelik olarak tasarlanmıştır. Uygulama mimarisi, Amazon Web Services (AWS) tarafından sağlanan servisler kullanılarak oluşturulmuş basit ve anlaşılır bir yapıya sahiptir.

Mimari Yapının Genel Akışı



Mimari Bileşenlerin Açıklaması

- **Kullanıcı (Web Tarayıcı):**
Kullanıcılar web sitesine internet tarayıcıları üzerinden erişmektedir.
- **Amazon CloudFront:**
CloudFront, kullanıcıdan gelen istekleri karşılayan ve içeriği HTTPS protokolü üzerinden güvenli bir şekilde sunan CDN servisidir. Aynı zamanda içeriği global edge noktaları üzerinden dağıtarak erişim performansını artırmaktadır.
- **Amazon S3:**
Web sitesine ait tüm statik dosyalar (HTML, CSS, JavaScript ve görseller) Amazon S3 bucket içerisinde saklanmaktadır. S3, static website hosting özelliği sayesinde bu dosyaları web içeriği olarak sunmaktadır.

Bu mimari yapı sayesinde web sitesi, doğrudan S3 üzerinden HTTP ile erişilmek yerine CloudFront aracılığıyla HTTPS üzerinden daha güvenli, hızlı ve ölçeklenebilir bir şekilde kullanıcılarla sunulmuştur.

6. Dağıtım Süreci (Adım Adım Uygulama Notları)

6.1. AWS Üzerinde Maliyet Kontrolü ve Bütçe Yapılandırması

Bu aşamada, proje süresince beklenmeyen maliyetlerin oluşmasını engellemek amacıyla **AWS Billing and Cost Management** servisi kullanılarak maliyet kontrol mekanizmaları yapılandırılmıştır. İlk olarak hesap faturalama paneline erişilmiş ve hesabın ücretsiz plan (Free Plan) kapsamında olduğu doğrulanmıştır. Ardından **AWS Budgets** servisi üzerinden yeni bir bütçe oluşturulmuştur.

Bütçe oluşturma sürecinde, basitleştirilmiş şablonlar kullanılarak **“Zero Spend Budget”** seçeneği tercih edilmiştir. Bu bütçe türü, harcama tutarı **0.01 USD** seviyesini aşlığında kullanıcıyı anında bilgilendirecek şekilde yapılandırılmıştır. Son olarak, oluşturulan bütçenin başarıyla aktif edildiği ve sistem tarafından sağlıklı (Healthy) durumda olduğu doğrulanmıştır.

Bu yapılandırma sayesinde proje boyunca AWS servisleri güvenli bir şekilde kullanılmakta, olası maliyet riskleri erken aşamada tespit edilerek kontrol altına alınmaktadır. Bu adım, bulut ortamlarında maliyet yönetiminin önemini göstermek açısından projenin temel bileşenlerinden biridir.

Screenshot of the AWS Billing and Cost Management home page.

The page shows a message: "Your free plan account does not get charged. Credits cover your free plan costs. Your free access to AWS services will end when your free plan period expires or when you have depleted all credits." with a "View details" link and an "Upgrade plan" button.

Key sections include:

- Cost summary**: Displays Month-to-date cost, Total forecasted cost for current month, and Last month's total cost, all marked as "Data unavailable".
- Cost monitor**: Shows Budgets status, Cost anomalies status (MTD), and a note about Setup required.
- Cost breakdown**: A tabbed section showing bill details.
- Recommended actions (4)**: A list of actions with icons.

Navigation bar at the bottom includes CloudShell, Feedback, and Console Mobile App.

Screenshot of the "Choose budget type" page.

The "Budget setup" section contains two options:

- Use a template (simplified): "Use the recommended configurations. You can change some configuration options after the budget is created."
- Customize (advanced): "Customize a budget to set parameters specific to your use case. You can customize the time period, the start month, and specific accounts."

The "Billing View - optional" section allows selecting a billing view for the export.

The "Templates - new" section lists three templates:

- Zero spend budget: "Create a budget that notifies you once your spending exceeds \$0.01."
- Monthly cost budget: "Create a monthly budget that notifies you if you exceed, or are forecasted to exceed, the budget amount."
- Daily Savings Plans coverage budget: "Create a coverage budget for your Savings Plans that notifies you when you fall below the defined target."
- Daily reservation utilization budget: "Create a utilization budget for your reservations that notifies you when you fall below the defined target."

Screenshot of the "Overview" page for budgets.

A green message box says: "Your budget Zero-Spend-Alert has been created successfully. After creating a budget, it can take up to 24 hours to populate all of your spend data." with a "Submit feedback" button.

The "Budgets (1)" table shows one entry:

Name	Thresholds	Health status	Billing View	Budget	Amount ...	Forecast...
Zero-Spend-Alert	OK	Healthy	Primary View	\$1.00	-	-

Navigation bar at the bottom includes CloudShell, Feedback, and Console Mobile App.

6.2. Neden Amazon S3 Tercih Edildi?

Bu projede geliştirilen uygulama, yalnızca **HTML, CSS ve JavaScript** kullanılarak oluşturulmuş, sunucu tarafı işlem gerektirmeyen **çok sayfalı (multi-page) statik bir web sitesidir**. Bu nedenle uygulamanın barındırılması için sunucu yönetimi gerektirmeyen, ölçeklenebilir ve düşük maliyetli bir bulut servisine ihtiyaç duyulmuştur.

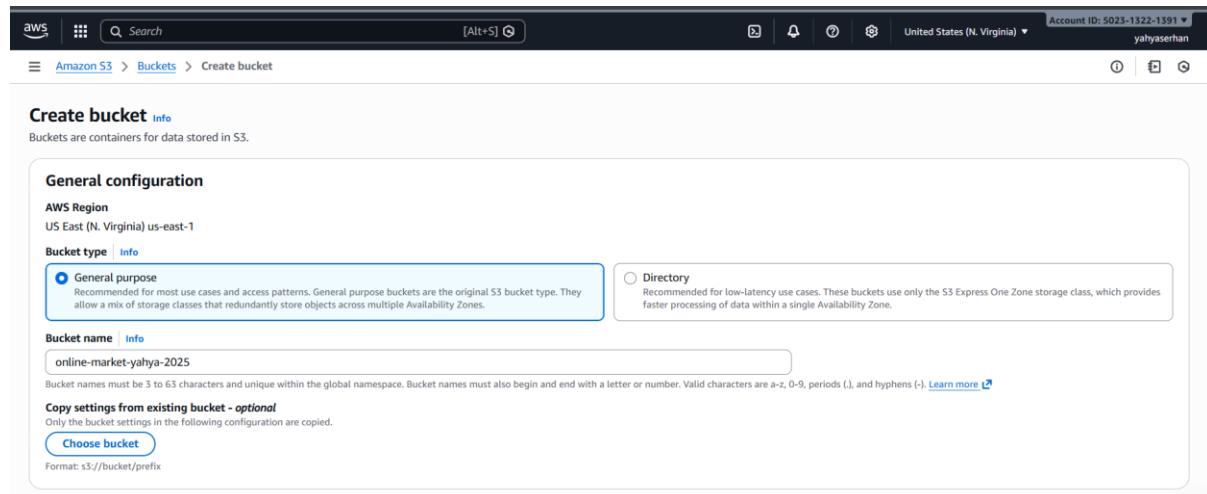
Amazon S3 (Simple Storage Service), statik web sitelerinin barındırılması için AWS tarafından önerilen bir çözümüdür. S3; yüksek erişilebilirlik, dayanıklılık (durability), kolay yapılandırma ve düşük maliyet gibi avantajlar sunar. Ayrıca AWS Free Tier kapsamında belirli kullanım sınırları dahilinde ücretsiz olarak kullanılabilmesi, proje ödevi kapsamında maliyet riskini en aza indirmiştir.

Bu nedenlerle Amazon S3, proje gereksinimlerine en uygun ve verimli bulut hizmeti olarak tercih edilmiştir.

6.3. Amazon S3 Bucket Oluşturulması

Bu aşamada, uygulama dosyalarının depolanması ve barındırılması amacıyla AWS üzerinde bir **S3 Bucket** oluşturulmuştur. Bucket, AWS Console üzerinden belirlenen bölgede (Region) oluşturulmuş ve proje dosyalarının yüklenmesine hazır hâle getirilmiştir.

Oluşturulan bucket, statik web sitesi barındırma senaryosuna uygun şekilde yapılandırılmak üzere hazırlanmıştır. Bu yapı sayesinde HTML, CSS ve JavaScript dosyaları merkezi bir depolama alanında tutulmakta ve internet üzerinden erişilebilir hâle getirilmektedir. Bucket'ın başarıyla oluşturulmasıyla birlikte, uygulamanın bulut ortamına taşınması sürecinin temel adımı tamamlanmıştır.



The screenshot shows the AWS S3 console interface. At the top, there's a green banner indicating "Successfully created bucket 'online-market-yahya-2025'". Below this, the "General purpose buckets" section lists one bucket named "online-market-yahya-2025" in the US East (N. Virginia) region. To the right, there are two informational boxes: "Account snapshot" and "External access summary - new". The bottom of the page includes standard AWS navigation links like CloudShell, Feedback, and Console Mobile App.

6.4. Uygulama Dosyalarının Amazon S3 Bucket'a Yüklenmesi

Bu adımda, projeye ait tüm **HTML, CSS, JavaScript ve görsel dosyaları** başarıyla Amazon S3 Bucket içerisine yüklenmiştir. Yükleme işlemi tamamlandıktan sonra dosyaların durumunun *Succeeded* olarak görüntülendiği doğrulanmıştır.

Bu işlem sayesinde uygulama dosyaları bulut ortamında güvenli bir şekilde depolanmış ve statik web sitesi barındırma sürecinin bir sonraki aşaması olan **Static Website Hosting yapılandırması** için hazır hâle getirilmiştir.

The screenshot shows the AWS S3 console after a file upload. A green banner at the top says "Upload succeeded". Below it, the "Files and folders" section displays a table of uploaded items. All 141 files listed have a status of "Succeeded" with a green checkmark icon. The table includes columns for Name, Folder, Type, Size, Status, and Error. The bottom of the page shows standard AWS navigation links.

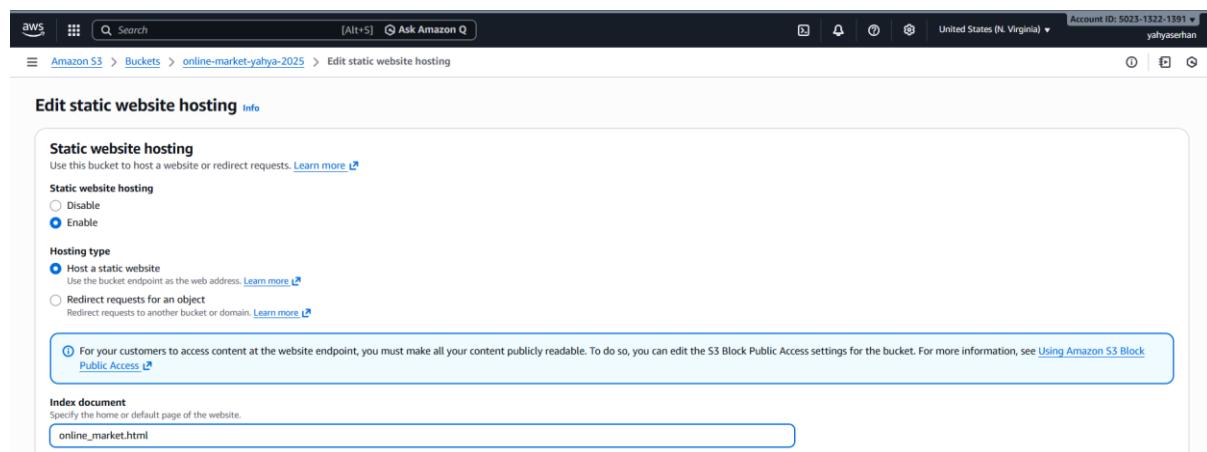
Name	Folder	Type	Size	Status	Error
Staple-Foods.html	-	text/html	15.2 KB		-
online_market.css	-	text/css	12.0 KB		-
online_market.html	-	text/html	20.8 KB		-
online_market.js	-	text/javascript	5.7 KB		-
serhan market logo.jpg	-	image/jpeg	38.2 KB		-
serhan.ico	-	image/x-icon	66.1 KB		-
used_Pictures.html	-	text/html	56.5 KB		-
ayran.jpg	dairy-products/	image/jpeg	69.0 KB		-
butter.jpg	dairy-products/	image/jpeg	10.0 KB		-

6.5. Amazon S3 Üzerinde Static Website Hosting'in Etkinleştirilmesi

Bu aşamada, oluşturulan Amazon S3 Bucket üzerinde **Static Website Hosting** özelliği etkinleştirilmiştir. Bu ayar sayesinde S3 Bucket, statik bir web sitesi barındırabilecek şekilde yapılandırılmıştır.

Yapılardırma sırasında "Host a static website" seçeneği tercih edilmiş ve ana giriş sayfası (Index document) olarak **online_market.html** tanımlanmıştır. Bu işlem ile birlikte, bucket içindeki HTML, CSS ve JavaScript dosyaları bir web sitesi olarak internet üzerinden erişilebilir hâle getirilmiştir.

Static Website Hosting'in etkinleştirilmesi, uygulamanın yerel ortamdan bulut ortamına taşınarak gerçek bir web adresi üzerinden yayınılmamasını sağlamıştır.

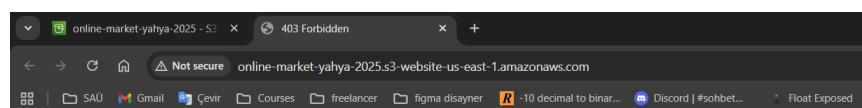


The screenshot shows the AWS S3 console with the 'Edit static website hosting' configuration for the 'online-market-yahya-2025' bucket. The 'Static website hosting' section has the 'Enable' radio button selected. Under 'Hosting type', the 'Host a static website' radio button is selected, with the 'Specify the home or default page of the website' field containing 'online_market.html'. A note at the bottom states: 'For your customers to access content at the website endpoint, you must make all your content publicly readable. To do so, you can edit the S3 Block Public Access settings for the bucket. For more information, see Using Amazon S3 Block Public Access.'

6.6. Static Website Hosting Etkinleştirilmesine Rağmen Erişim Engeli (403 Forbidden)

Static Website Hosting özelliği etkinleştirilmiş olmasına rağmen, bu aşamada web sitesine doğrudan erişim sağlanamamıştır ve tarayıcıda **403 Forbidden (Access Denied)** hatası görüntülenmiştir. Bu durum, Amazon S3 üzerinde oluşturulan bucket'ın henüz **genel erişime (public read access)** izin verecek şekilde yapılandırmamış olmasından kaynaklanmaktadır.

Bu hata, uygulamanın yanlış yapılandırıldığını değil; aksine AWS'nin varsayılan güvenlik politikaları gereği, kaynaklara izinsiz erişimi engellediğini göstermektedir. Bir sonraki adımda gerekli **Bucket Policy** tanımlanarak yalnızca okuma yetkisi verilmiş ve bu erişim problemi giderilmiştir.



The screenshot shows a web browser displaying a 403 Forbidden error page. The URL in the address bar is 'online-market-yahya-2025.s3-website-us-east-1.amazonaws.com'. The error message is '403 Forbidden'.

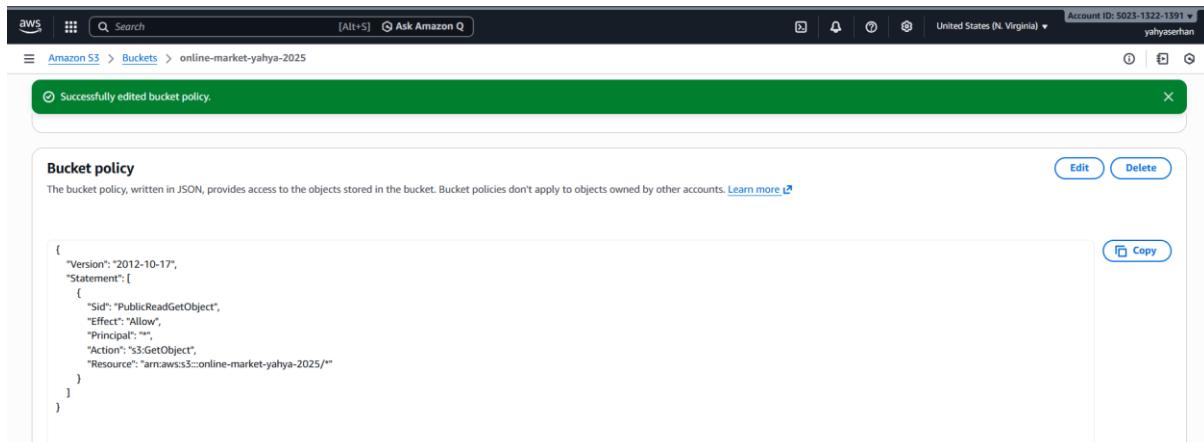
403 Forbidden

- Code: AccessDenied
- Message: Access Denied
- RequestId: 0F5WVSVWB2R7Q6T4
- HostId: wWmTu/45e7TjkBj2mrbsuoILexDF+8ybIIGuAd7RU8exTx8kJeJTA0Y1eWqv9UmE2aN5cv37MdwgjJ95rK2/RiN5YVzZC2A

6.7. Genel Erişim İçin Amazon S3 Bucket Policy Yapılandırması

Bu adımda, Amazon S3 üzerinde barındırılan statik web sitesine internet üzerinden erişim sağlanabilmesi için gerekli **Bucket Policy** yapılandırması gerçekleştirilmiştir. Tanımlanan politika ile tüm kullanıcılar yalnızca **okuma (s3:GetObject)** yetkisi verilmiş, böylece bucket içerisindeki dosyaların güvenli bir şekilde görüntülenmesi mümkün hâle getirilmiştir.

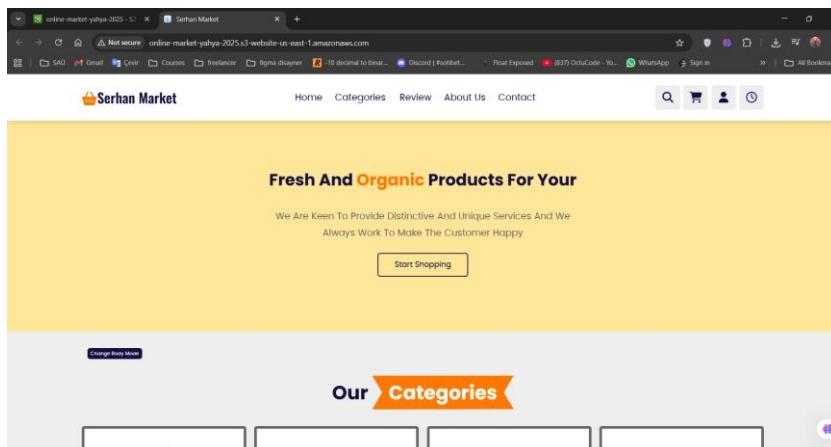
Bu yapılandırmanın uygulanmasının ardından daha önce karşılaşılan **403 Forbidden (Access Denied)** hatası giderilmiş ve web sitesi başarıyla erişilebilir duruma getirilmiştir. Bu adım, AWS'nin varsayılan güvenlik yaklaşımı korunarak kontrollü bir şekilde genel erişim sağlanması için tanımaktadır.



6.8. Web Sitesinin AWS Üzerinde Başarıyla Yayınlanması

Tüm yapılandırma adımlarının tamamlanmasının ardından, statik web sitesi Amazon S3 üzerinde başarıyla yayınlanmıştır. Web sitesine S3 tarafından sağlanan endpoint üzerinden erişim sağlanmış ve tüm sayfaların, stillerin ve görsellerin doğru şekilde yüklentiği doğrulanmıştır.

Bu aşama ile birlikte uygulama, yerel ortamdan bulut ortamına taşınmış ve gerçek bir web adresi üzerinden kullanıma sunulmuştur.



Website Link : <http://online-market-yahya-2025.s3-website-us-east-1.amazonaws.com>

6.9. HTTPS Desteği Sağlama ve Amazon CloudFront'un Tercih Edilme Nedeni

Amazon S3 Static Website Hosting servisi varsayılan olarak yalnızca **HTTP** protokolü üzerinden erişim sağlamaktadır ve doğrudan **HTTPS (SSL/TLS)** desteği sunmamaktadır. Günümüz web standartlarında HTTPS, veri güvenliği, kullanıcı gizliliği ve profesyonel bir web deneyimi açısından büyük önem taşımaktadır. Bu nedenle projenin daha güvenli ve modern bir yapıya kavuşturulması hedeflenmiştir.

Bu ihtiyacı karşılamak amacıyla **Amazon CloudFront** kullanılmıştır. CloudFront, Amazon S3 üzerinde barındırılan statik web sitelerini HTTPS protokolü üzerinden sunabilen bir **Content Delivery Network (CDN)** hizmetidir. CloudFront sayesinde SSL/TLS sertifikası sağlanmış, HTTP istekleri otomatik olarak HTTPS'e yönlendirilmiş ve web sitesi güvenli erişime açılmıştır.

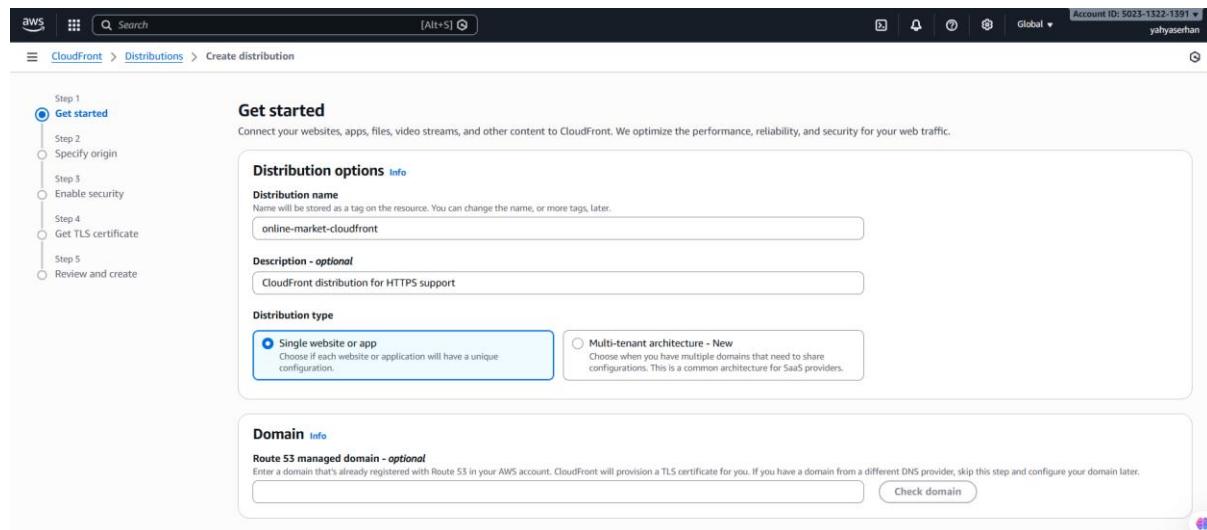
Bu yaklaşım, ek bir sunucu yapılandırmasına gerek kalmadan güvenli, ölçeklenebilir ve AWS tarafından önerilen bir çözüm sunduğu için tercih edilmiştir.

6.10. Amazon CloudFront Distribution Oluşturulması

Bu adımda, web sitesine HTTPS desteği kazandırmak amacıyla yeni bir **Amazon CloudFront Distribution** oluşturulmuştur. Dağıtım yapılandırması sırasında tek bir web sitesi için uygun olan “**Single website or app**” seçeneği tercih edilmiştir.

Dağıtımı anlamlı bir isim ve açıklama verilerek yapılandırma süreci başlatılmış, özel bir alan adı (domain) kullanılmadığı için varsayılan **CloudFront alan adı** üzerinden HTTPS erişimi sağlanması hedeflenmiştir. Bu yapılandırma, CloudFront'un S3 üzerinde barındırılan statik web sitesine güvenli bir ara katman olarak hizmet vermesini sağlamaktadır.

Bu aşama, web sitesinin HTTPS üzerinden yayınlanabilmesi için yapılan yapılandırmanın temel adımını oluşturmaktadır.



6.11. CloudFront Origin Tanımlama (S3 Bucket Bağlantısı)

Bu adımda, Amazon CloudFront dağıtıımı için **origin (kaynak)** yapılandırması yapılmıştır. Web sitesinin statik dosyaları (HTML, CSS, JavaScript ve görseller) **Amazon S3 bucket** üzerinde bulunduğu için origin type olarak **Amazon S3** seçilmiştir.

İlgili S3 bucket (online-market-yahya-2025) CloudFront'a bağlanarak, içeriklerin küresel CDN ağının üzerinden daha hızlı ve güvenli bir şekilde sunulması hedeflenmiştir. Güvenliği artırmak amacıyla **“Allow private S3 bucket access to CloudFront”** seçeneği etkinleştirilmiş ve böylece S3 içeriğine doğrudan erişim engellenip yalnızca CloudFront üzerinden erişim sağlanmıştır.

Ayrıca AWS'nin önerdiği **origin settings** ve **cache settings** kullanılarak performans, önbellekleme ve erişim yönetimi otomatik ve optimize bir şekilde yapılandırılmıştır. Bu yapı, HTTPS desteğiyle birlikte daha güvenli ve profesyonel bir statik web sitesi yayını sağlamaktadır.

The screenshot shows the 'Specify origin' step of the CloudFront configuration wizard. On the left, a vertical navigation bar lists steps: Step 1 (Get started), Step 2 (Specify origin, highlighted in blue), Step 3 (Enable security), and Step 4 (Review and create). The main area is titled 'Specify origin'.

Origin type: A section describing where content lives. It includes a note: "Your origin is where your content (such as a website or app) lives. CloudFront works with AWS-based origins and origins hosted on other cloud providers." Below are five options:

- Amazon S3** (selected): Delivers static assets like files and images, statically generated websites or single page applications (SPA).
- Elastic Load Balancer**: Delivers applications hosted behind ELB such as dynamic websites, web services, and APIs.
- API Gateway**: Delivers API endpoints for REST APIs hosted on API Gateway.
- Elemental MediaPackage**: Delivers end-to-end live events or video on demand (VOD).
- VPC origin**: Delivers applications and content hosted within private VPCs, such as EC2 instances and Application Load Balancers.
- Other**: Refers to any AWS or non-AWS origin through its publicly resolvable URL.

Origin: A section for configuring the S3 origin. It shows the selected origin as "online-market-yahya-2025.s3.us-east-1.amazonaws.com". A note says: "This S3 bucket has static web hosting enabled. If you plan to use this distribution as a website, we recommend using the S3 website endpoint rather than the bucket endpoint." A "Use website endpoint" button is available. An optional "Origin path" field contains "/path".

Settings: A section for setting origin and cache configurations. It includes:

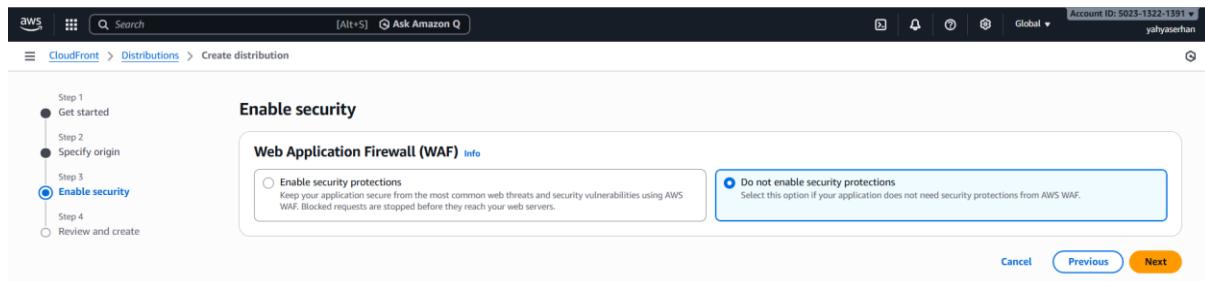
- Allow private S3 bucket access to CloudFront**: A note states: "CloudFront will update your S3 bucket policy to allow CloudFront to access your S3 bucket. The policy allows CloudFront to access the bucket only when the request is on behalf of the CloudFront distribution that contains the S3 origin." A checkbox is checked: "Allow private S3 bucket access to CloudFront - Recommended".
- Origin settings**: A note: "Origin settings control how CloudFront connects to the specified origin." A radio button is selected: "Use recommended origin settings".
- Cache settings**: A note: "Cache settings determine when CloudFront serves cached content and when it fetches new content from the origin." A radio button is selected: "Use recommended cache settings tailored to serving S3 content".

At the bottom right are buttons: "Cancel", "Previous", and "Next".

6.12. CloudFront Güvenlik Yapılandırması (AWS WAF Seçilmemesi)

Bu adımda CloudFront dağıtımını için güvenlik yapılandırması değerlendirilmiştir. AWS Web Application Firewall (WAF) hizmeti, gelişmiş web saldırılarına karşı koruma sağlasa da **ücretli bir servis** olduğu için bu projede etkinleştirilmemiştir.

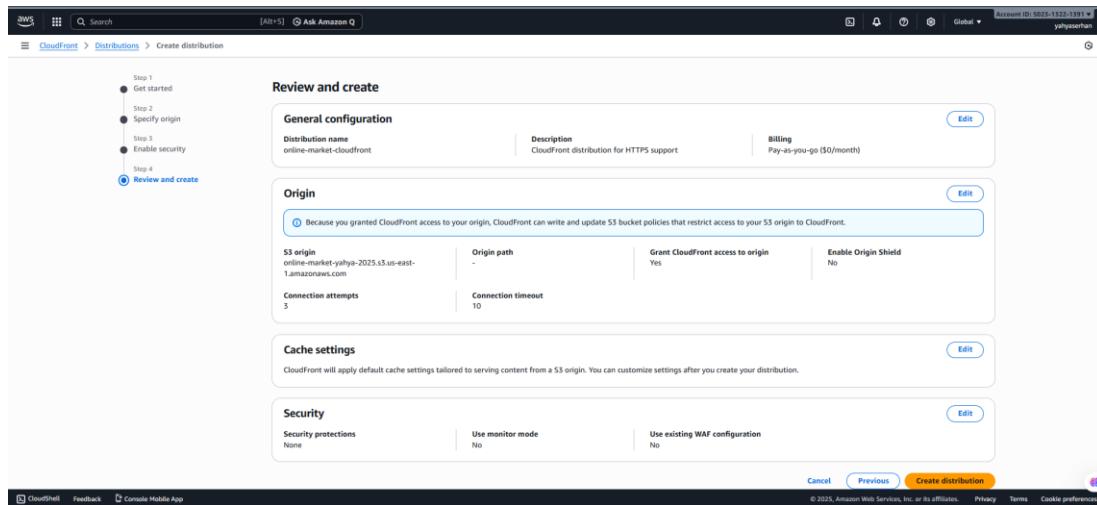
Proje kapsamında geliştirilen statik web sitesi, kullanıcı kimlik doğrulaması veya hassas veri işlemediğinden dolayı **CloudFront'un sunduğu HTTPS desteği ve AWS Shield Standard** koruması yeterli görülmüştür. Bu yaklaşım, maliyetleri sıfırda tutarken aynı zamanda temel güvenlik gereksinimlerini karşılamaktadır.



6.13. CloudFront Dağıtımının Oluşturulması (Review and Create)

Bu adımda CloudFront dağıtımına ait tüm yapılandırmalar gözden geçirilmiş ve doğrulanmıştır. Dağıtım, Amazon S3 origin kullanacak şekilde ayarlanmış, varsayılan cache ayarları korunmuş ve ek ücret gerektiren güvenlik servisleri (AWS WAF) devre dışı bırakılmıştır.

Yapılandırmaların doğruluğu kontrol edildikten sonra CloudFront dağıtım oluşturulmuş ve HTTPS üzerinden güvenli erişim sağlanması için gerekli altyapı hazırlanmıştır.

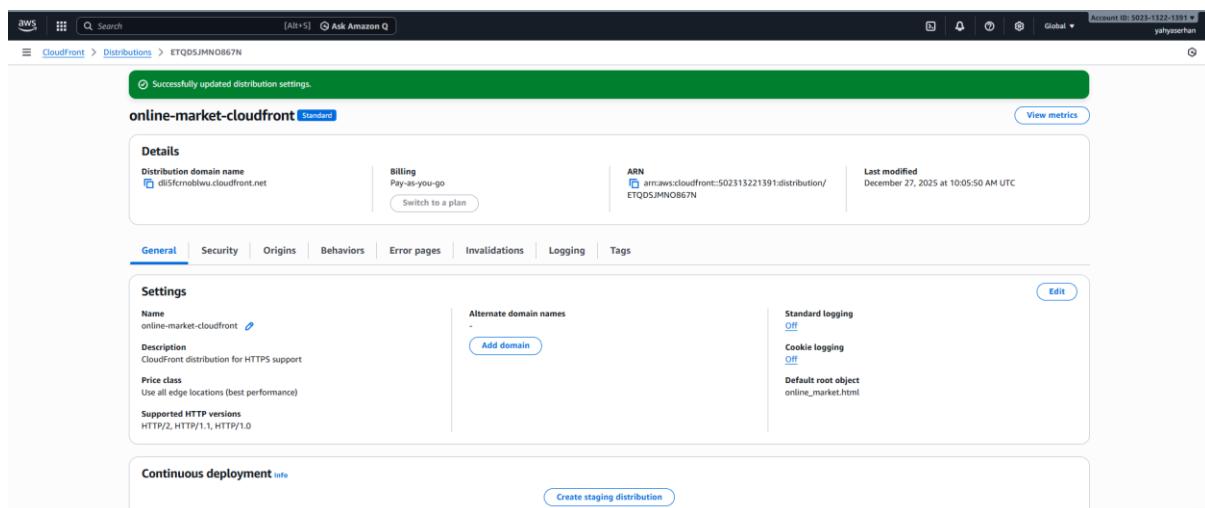


6.14. CloudFront Dağıtımının Başarıyla Oluşturulması ve HTTPS Yapılandırmasının Tamamlanması

Bu ekranında Amazon CloudFront dağıtımının başarıyla oluşturulduğu ve yapılandırma ayarlarının güncellendiği görülmektedir. Oluşturulan CloudFront dağıtımlı, Amazon S3 üzerinde barındırılan statik web sitesini origin olarak kullanmakta ve sitenin HTTPS protokolü üzerinden güvenli şekilde yayınılmamasını sağlamaktadır.

Yapılandırma kapsamında default root object olarak online_market.html tanımlanmış, böylece kullanıcılar kök URL üzerinden siteye eriştiğinde ana sayfa otomatik olarak görüntülenmektedir. CloudFront, içeriği dünya genelindeki edge lokasyonları üzerinden dağıtarak hem erişim gecikmesini azaltmada hem de veri iletimini TLS şifrelemesi ile güvence altına almaktadır.

Bu yapı sayesinde statik web sitesi, doğrudan S3 üzerinden HTTP ile yayınlanmak yerine, CloudFront aracılığıyla daha güvenli, performanslı ve ölçülebilir bir mimari ile kullanıcılar sunulmuştur.



The screenshot shows the AWS CloudFront Distribution Settings page for a distribution named "online-market-cloudfront". A green success message at the top states "Successfully updated distribution settings." Below it, the distribution details are listed: Distribution domain name (dli5fcrnoblwu.cloudfront.net), Billing plan (Pay-as-you-go), ARN (arn:aws:cloudfront:502313221391:distribution/ETQDSJMNO867N), and Last modified (December 27, 2025 at 10:05:50 AM UTC). The General tab is selected, showing the distribution's name (online-market-cloudfront), description (CloudFront distribution for HTTPS support), price class (Use all edge locations (best performance)), and supported HTTP versions (HTTP/2, HTTP/1.1, HTTP/1.0). The Settings section includes alternate domain names (Add domain) and logging options (Standard logging Off, Cookie logging Off, Default root object online_market.html). At the bottom, there is a Continuous deployment section and a "Create staging distribution" button.

Website Link: <https://dli5fcrnoblwu.cloudfront.net/>

7. Öğrenilen Dersler ve Olası İyileştirmeler

Bu proje kapsamında, bulut bilişim ortamında statik bir web sitesinin dağıtımını ve güvenli bir şekilde yayınlanması süreci uygulamalı olarak öğrenilmiştir. Amazon Web Services (AWS) servisleri kullanılarak bir uygulamanın yerel ortamdan bulut ortamına taşınmasının temel adımları deneyimlenmiştir.

Öğrenilen Dersler

- Amazon S3 kullanılarak statik web sitelerinin kolay ve düşük maliyetli bir şekilde barındırılabileceği öğrenilmiştir.
- AWS üzerinde varsayılan güvenlik ayarlarının (public access, bucket policy) önemine dair pratik bilgi kazanılmıştır.
- HTTP ve HTTPS arasındaki farklar anlaşılmış, güvenli erişim için HTTPS kullanımının gerekliliği görülmüştür.
- Amazon CloudFront'un, S3 üzerinde barındırılan içeriklere HTTPS desteği sağlayan ve performansı artıran bir CDN servisi olduğu öğrenilmiştir.
- Bulut ortamlarında maliyet kontrolünün önemi kavranmış ve AWS Budgets kullanılarak harcama takibi yapılmıştır.

Olası İyileştirmeler

- Web sitesi için özel bir alan adı (custom domain) alınarak Route 53 üzerinden yapılandırma yapılabilir.
- AWS WAF kullanılarak web sitesine yönelik güvenlik önlemleri artırılabilir.
- Dağıtım süreci, GitHub Actions veya AWS CLI kullanılarak otomatik hâle getirilebilir.
- Projeye sunucu tarafı (backend) bileşenler eklenerek daha kapsamlı bir web uygulamasına dönüştürülebilir.

Bu iyileştirmeler, projenin daha ölçüklenebilir, güvenli ve profesyonel bir yapıya kavuşmasını sağlayacaktır.

8. Özет

Bu projede statik bir e-ticaret web sitesi AWS altyapısı üzerinde güvenli ve ölçeklenebilir şekilde yayınlanmıştır. Öncelikle proje dosyaları Amazon S3 üzerine yüklenmiş ve S3, statik web sitesi barındırma (static website hosting) için yapılandırılmıştır. Hesap güvenliği ve maliyet kontrolü amacıyla AWS Billing ve Budget ayarları yapılmıştır.

Website Link : <http://online-market-yahya-2025.s3-website-us-east-1.amazonaws.com>

Daha sonra, sitenin HTTP yerine **HTTPS** üzerinden güvenli çalışması için **Amazon CloudFront** kullanılmıştır. CloudFront, S3 bucket'ı origin olarak kullanacak şekilde yapılandırılmış, TLS/SSL desteği sağlanmış ve içerik global edge noktaları üzerinden dağıtılmıştır. Böylece hem veri güvenliği artırılmış hem de performans iyileştirilmiştir. Son olarak CloudFront ayarlarında varsayılan ana sayfa (default root object) tanımlanarak sitenin sorunsuz şekilde açılması sağlanmıştır.

Website Link : <https://dli5fcrnoblwu.cloudfront.net/>