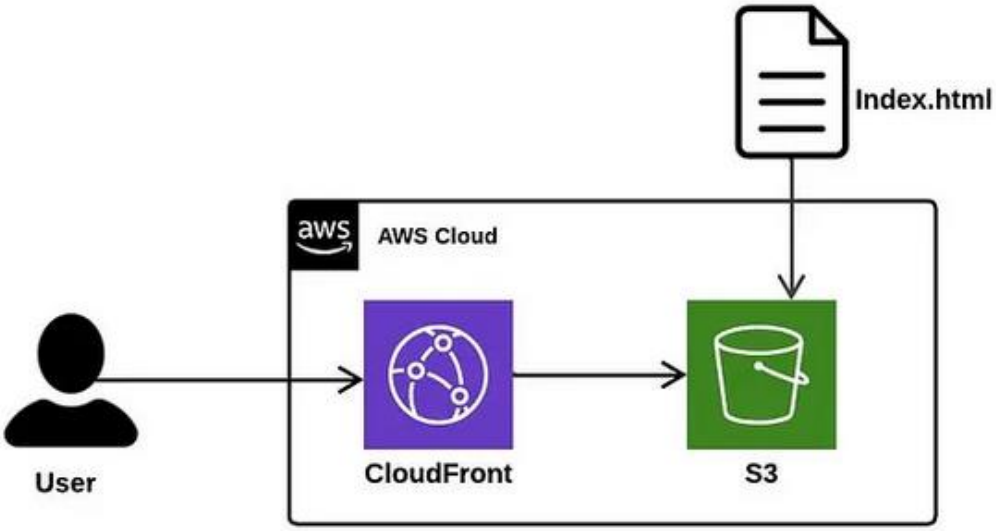
# Static Website Hosting Using Amazon S3 and CloudFront

This comprehensive project document presents a detailed, step-by-step guide to hosting a static website using **Amazon Simple Storage Service (Amazon S3)** as the core storage solution and **Amazon CloudFront** as the content delivery network (CDN). The purpose of this project is to demonstrate how developers and IT professionals can utilize AWS services to build and deploy a reliable, scalable, and globally distributed static website without the need for managing servers.

In this project, users will learn how to configure an S3 bucket to serve as a secure and accessible static web host, complete with proper permissions and static website hosting configurations. Additionally, the project will cover the setup of Amazon CloudFront to ensure high-speed content delivery by caching website files in edge locations close to the end users around the world.

The solution emphasizes a **serverless architecture**, reducing operational overhead while ensuring high availability, low latency, and cost-efficiency. By following this guide, learners will gain hands-on experience in implementing a robust static web hosting solution suitable for personal portfolios, product landing pages, documentation sites, or any informational web presence.



**Use Case: Cloud-Based Static Website Hosting with S3 and CloudFront**

**Client Background and Challenges**

My client operates a business-critical website that is currently hosted on a traditional **on-premise server infrastructure**. This setup demands continuous monitoring, hardware maintenance, and a dedicated IT operations team to manage tasks such as uptime assurance, server patching, and scalability planning. This not only increases **operational costs** but also adds significant **infrastructure complexity**.

With a steadily growing global customer base, the website has started to experience **performance degradation**, particularly in terms of loading speeds and response times. Customers accessing the site from geographically distant locations face increased **latency**, leading to a poor user experience.

Furthermore, the client has expressed **serious concerns about security**. The rise in cyber threats, such as DDoS attacks, unauthorized access attempts, and data breaches, has made it imperative for the business to shift to a more secure hosting solution. Their goal is to **protect digital assets**, **ensure data privacy**, and **comply with security best practices**.

**Client Objectives**

To address these growing challenges, the client has set forth the following goals:

1. **Migrate** the existing website to a **cloud-based solution** that is more manageable and cost-effective.
2. **Improve website performance and availability** for users across the globe.
3. **Enhance website security**, with protection against common web threats.
4. **Minimize ongoing maintenance** and reduce dependence on physical infrastructure.
5. **Lower hosting costs** while maintaining or improving site performance.

**Proposed Solution**

To meet the above requirements, we propose hosting the static website using:

- **Amazon S3**: To store and serve the static HTML, CSS, JavaScript, and image files in a secure, durable, and highly available storage service.
- **Amazon CloudFront**: To distribute the website content globally using AWS edge locations, improving content delivery speed and minimizing latency for international visitors.
- **Additional AWS Services (Optional)**: Integration with AWS WAF (Web Application Firewall) and AWS Certificate Manager (ACM) for **SSL encryption and web security** enhancements.

This serverless, cloud-native solution will offer **automatic scalability**, **global reach**, **built-in redundancy**, and **lower TCO (Total Cost of Ownership)**—perfectly aligning with the client's short-term needs and long-term digital transformation goals.

**How This Solution Benefits the Client**

To meet the client's goals of improving performance, enhancing security, and reducing costs, this solution leverages two core AWS services: **Amazon S3** and **Amazon CloudFront**. Below is a detailed explanation of how each service addresses specific business needs:

**Amazon S3: Scalable, Durable, and Cost-Effective Web Storage**

**Amazon Simple Storage Service (Amazon S3)** is a powerful object storage solution designed to offer **unlimited scalability**, **99.999999999% (11 nines) durability**, and **high availability** across multiple AWS Availability Zones. It is a popular choice for use cases such as static website hosting, data archiving, mobile apps, backup and restore, IoT storage, and analytics.

For this project, S3 serves as the **origin server** for hosting all static files (HTML, CSS, JS, images, etc.) of the website. Here's how S3 adds value for the client:

**Key Benefits of Amazon S3:**

- **Cost Savings:**
  Hosting a static website on S3 is **highly cost-efficient**. The client pays only for the storage space used and the bandwidth consumed—there's no need to provision or manage servers. This drastically cuts down the operational expenses compared to EC2 or on-premise hosting models.
- **Seamless Scalability:**
  Amazon S3 automatically scales to accommodate any traffic volume—from a few hundred visitors to millions. There's no need for manual configuration or provisioning, which makes it ideal for websites expecting unpredictable or seasonal traffic spikes.
- **High Availability & Durability:**
  With S3, your website is hosted on a **globally distributed infrastructure**. The use of multiple Availability Zones and internal data redundancy ensures the website remains up and available—even in the event of hardware failure or zone outages.
- **Reduced Management Overhead:**
  Being a **fully managed service**, Amazon S3 eliminates the need for system administration tasks such as OS patching, server health monitoring, and scaling configurations. This allows the client's IT team to focus on business innovation instead of infrastructure maintenance.

- **Data Governance & Compliance:**
  With built-in lifecycle policies, access control, and logging capabilities, S3 allows fine-grained control over data retention and access. This is particularly helpful in aligning with **industry compliance standards**.

**Amazon CloudFront: Accelerated Global Content Delivery with Edge Security**

**Amazon CloudFront** is a high-performance, secure **Content Delivery Network (CDN)** that delivers content to users from the nearest AWS edge location, reducing latency and improving website load times.

Once the static website is hosted on S3, CloudFront sits in front of it to distribute content globally with **low latency**, **high transfer speed**, and **enhanced security controls**.

**Key Benefits of Amazon CloudFront:**

**Improved Website Performance:**
CloudFront **caches content at edge locations** spread across major cities worldwide. When a user visits the site, content is delivered from the nearest edge location instead of directly from the S3 bucket, significantly reducing latency and boosting load speeds.

**Strengthened Security:**
CloudFront seamlessly integrates with **AWS WAF (Web Application Firewall)** to protect the site from common vulnerabilities like:

- SQL Injection
- Cross-Site Scripting (XSS)
- Cross-Site Request Forgery (CSRF)
  It also supports **HTTPS with SSL/TLS certificates**, protecting data-in-transit and building user trust.

- **Cost Optimization:**
  CloudFront reduces the number of direct hits to the S3 bucket by caching content closer to users. This **minimizes S3 request charges and data transfer costs**, contributing to significant savings on monthly bills.
- **High Customizability:**
  CloudFront supports **Lambda@Edge**, which allows developers to run code at AWS edge locations to customize content delivery dynamically, perform redirects, or block suspicious traffic—adding more flexibility and control.

Creation of index.html file :

<!DOCTYPE html>

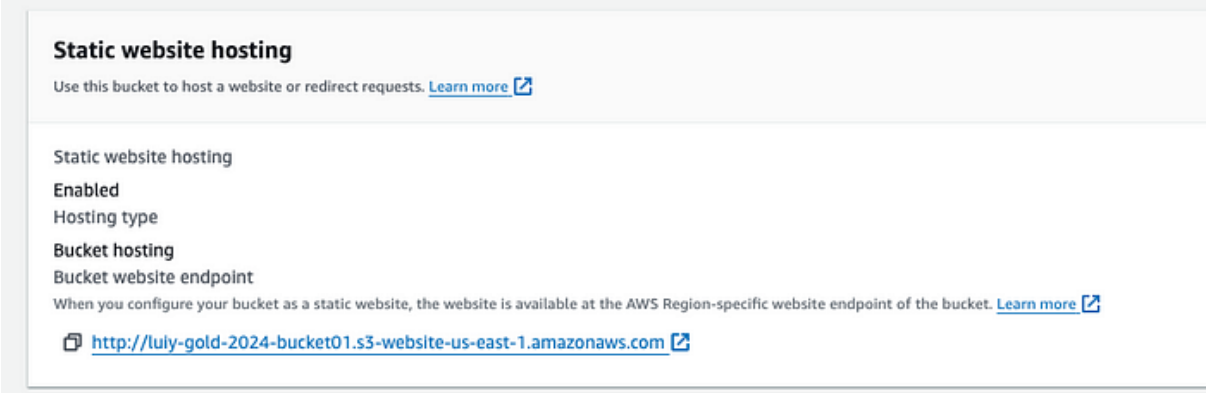
```
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Bargav - Student Profile</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      background-color: #eef3f8;
      color: #333;
      padding: 40px;
      max-width: 600px;
      margin: auto;
      border-radius: 10px;
      box-shadow: 0 4px 8px rgba(0,0,0,0.1);
    }
    h1 {
      text-align: center;
      color: #2e6da4;
    }
    p {
      font-size: 18px;
      line-height: 1.6;
    }
    .info {
      background: #fff;
      padding: 20px;
      border-radius: 8px;
    }
```

```
    </style>
</head>
<body>
    <h1>Student Profile</h1>
    <div class="info">
        <p><strong>Name:</strong> Bargav</p>
        <p><strong>College:</strong> Sri Datta Engineering College</p>
        <p><strong>Age:</strong> 18</p>
        <p><strong>Gender:</strong> Male</p>
        <p><strong>Mobile No:</strong> [Add your number here]</p>
    </div>
</body>
</html>
```

**S3 bucket Creation:**

1. Go to AWS console and select S3 service.
2. Click Create bucket. Give a unique name to the bucket, the name consists only of lowercase letters, numbers, dots (.), and hyphens (-).
3. Leave the rest of the settings as their defaults and create the bucket.
4. Click the bucket name and select the upload option to upload the HTML document. Once it is uploaded your bucket will look like this.


5. Now select the properties tab of the bucket, and scroll down to enable the static website hosting. once enabled you will get a URL like this.

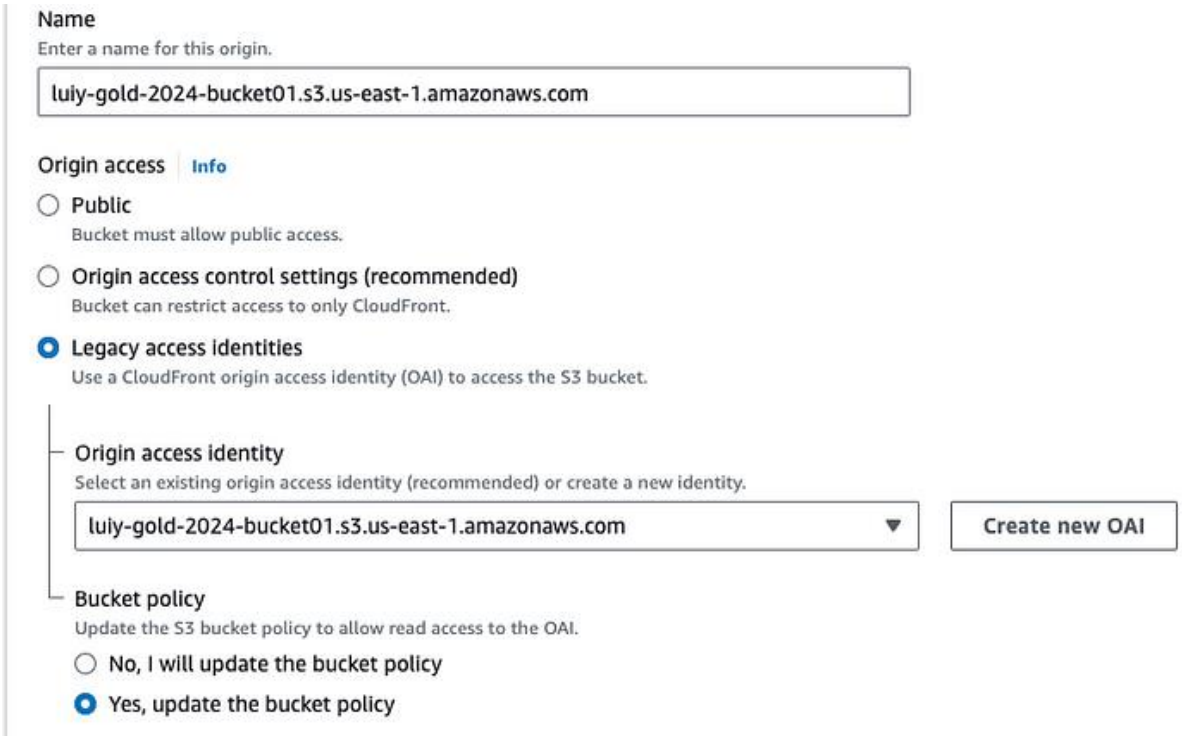



6. If you click the URL, don't expect to see the web page. This is because the bucket blocks all public access.

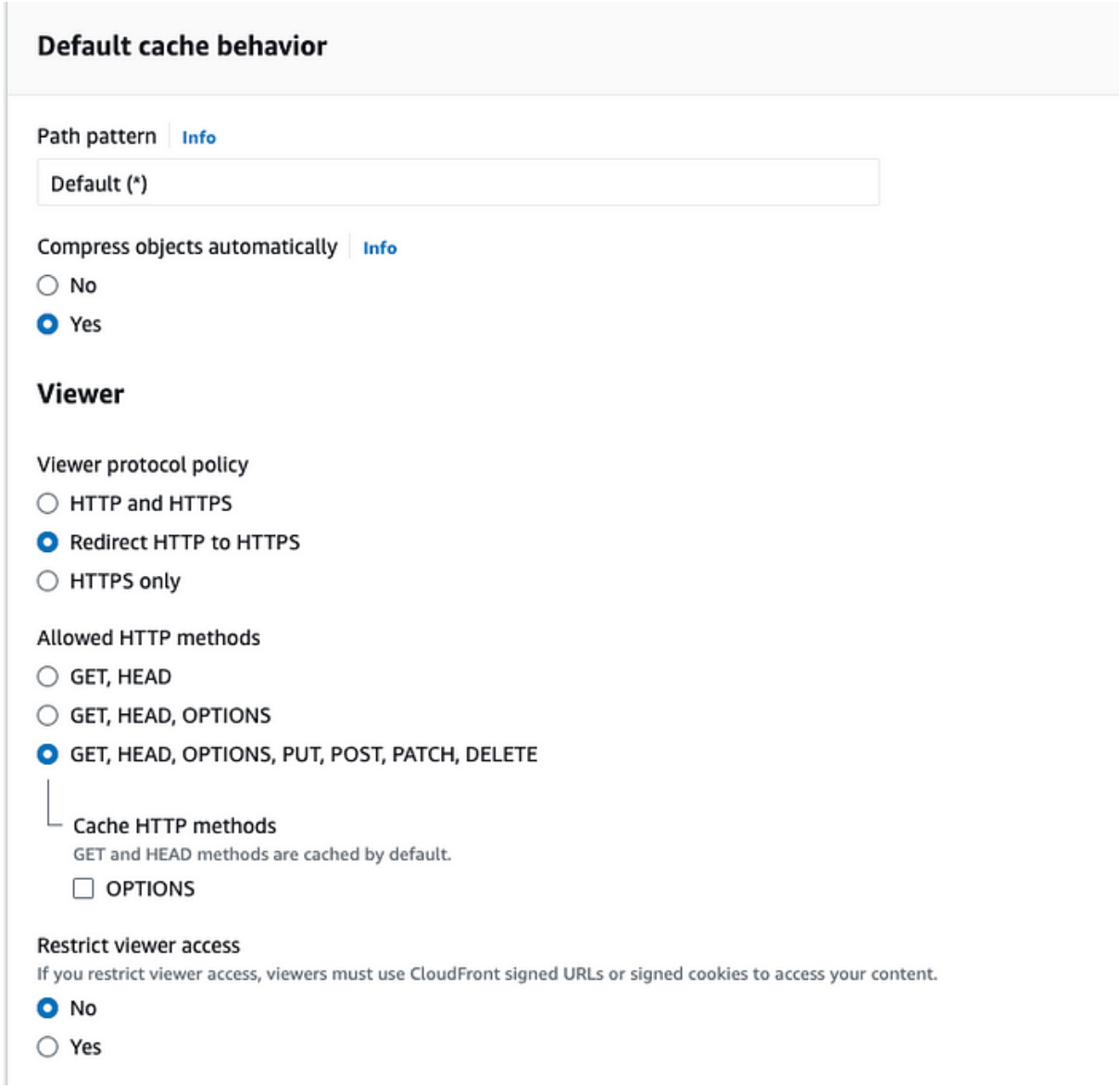

Now leave that as it is and select CloudFront from the AWS services.

**CloudFront Distribution:**

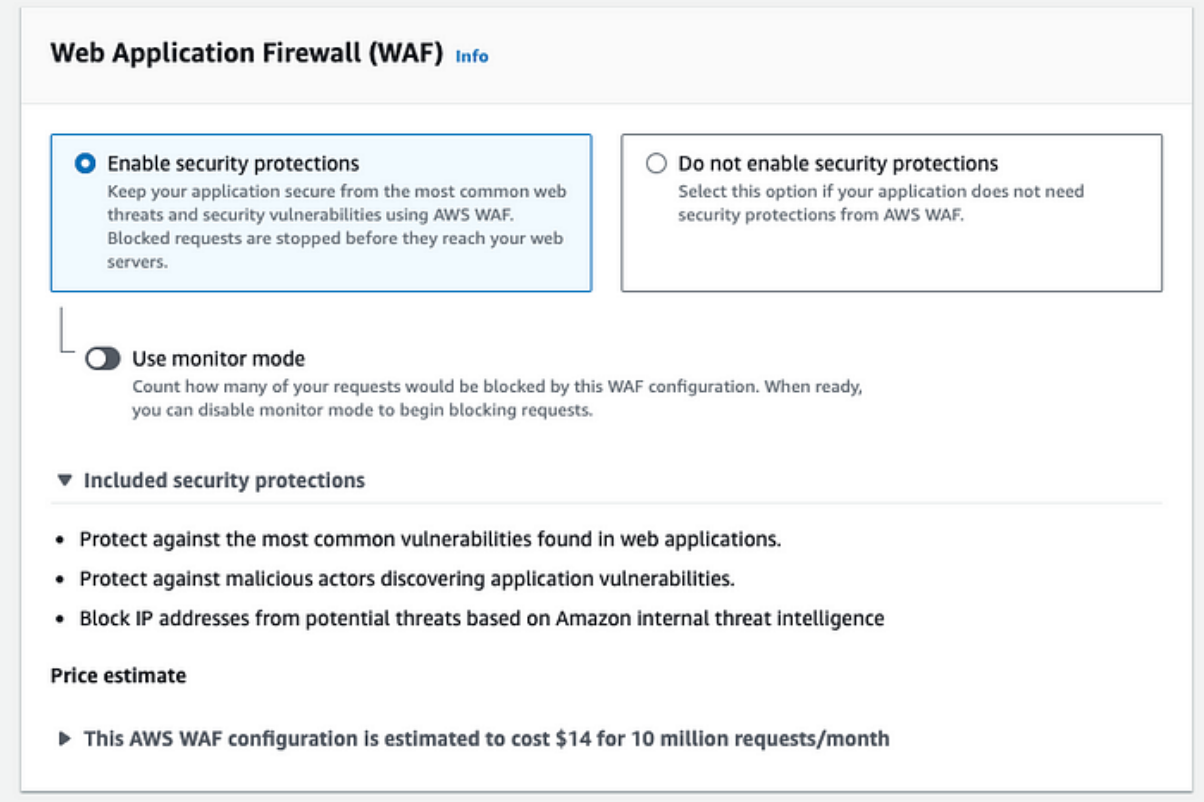
1. Select Create Distribution and follow these steps.
2. Select the Origin domain as the bucket we just created.
3. Under Origin access, go for legacy access identities and create a new OAI for our bucket. Don't forget to update the bucket policy.




4. Under Default cache behavior, select the viewer protocol policy to '**Redirect HTTP to HTTPS**' to have a secured web page as shown below.




5. Leave the rest as default and enable Security Protions to protect against most vulnerabilities found on web applications. This WAF is going to cost you some money.




**Final Step: Set Default Root Object and Create the CloudFront Distribution**

Once all the necessary settings for your CloudFront distribution are configured—including the origin (your S3 bucket), caching behavior, and access permissions—the final step is to **specify the default root object**.

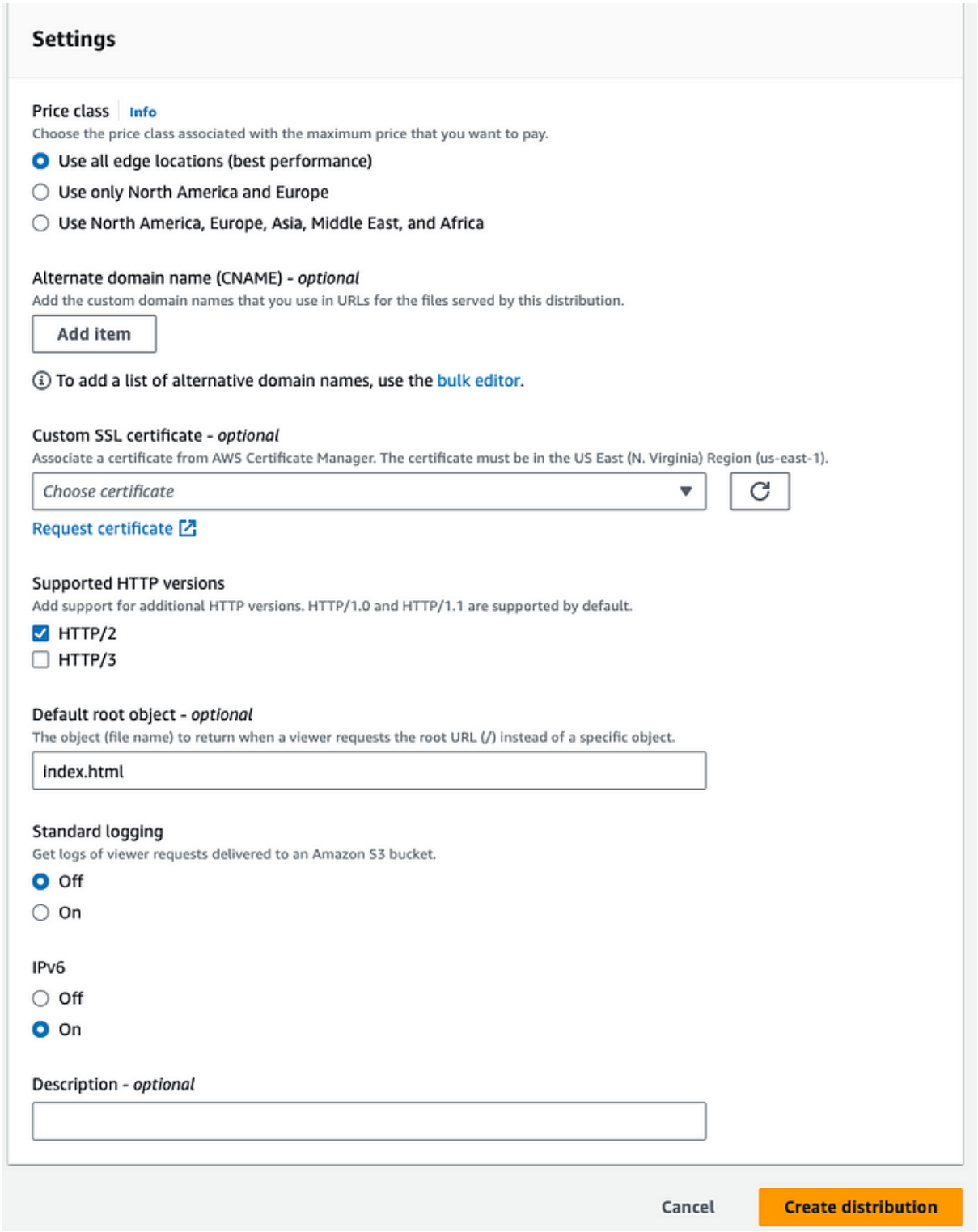
In this case, you should enter index.html as the default root object. This ensures that when users visit your domain (for example, www.example.com) **without specifying a file name**, CloudFront automatically serves the index.html file from your S3 bucket. If this setting is not defined, accessing the root domain may result in a 403 or 404 error.

**Note:**

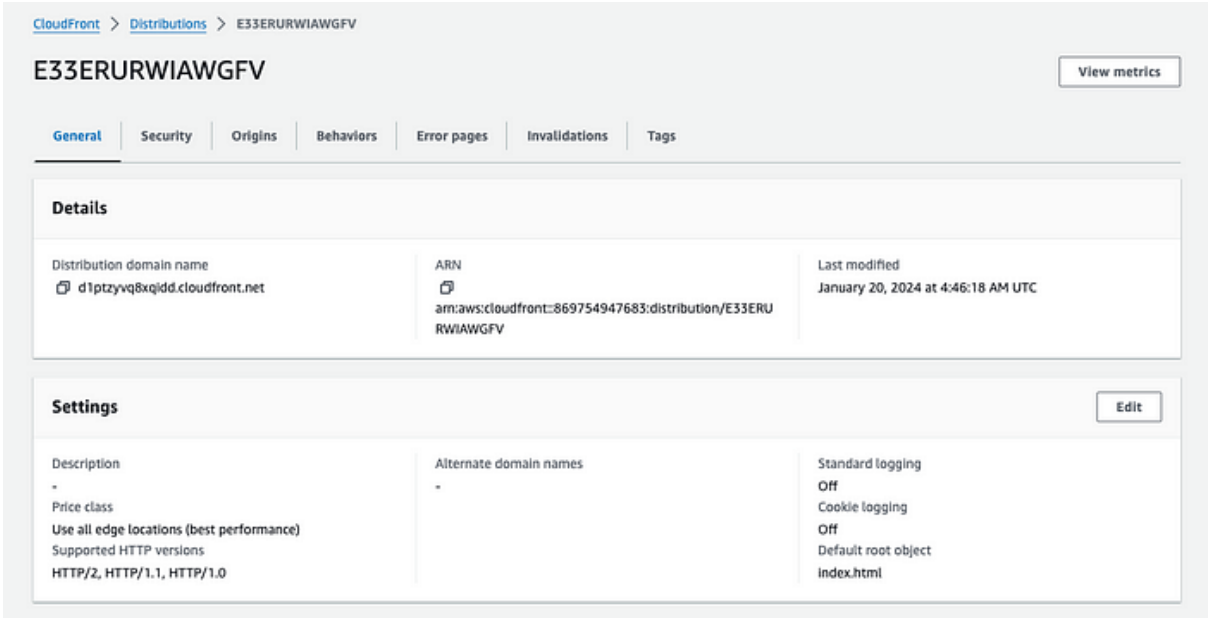
At this stage, it's recommended to add a **custom SSL/TLS certificate** (using AWS Certificate Manager) if you plan to use your own domain name and need HTTPS support. However, this guide does not cover SSL certificate configuration in detail.

Once index.html is entered in the Default Root Object field, **review all configuration settings**, and click on **"Create Distribution"**. CloudFront will now begin provisioning the distribution. This process typically takes a few minutes.

After the distribution is deployed, you will be provided with a **CloudFront URL**, which can be used to access your website globally via the CDN.




6. It will take some time say 3–5 minutes for our distribution to get deployed.




7. Once it gets deployed, click the distribution domain name,