

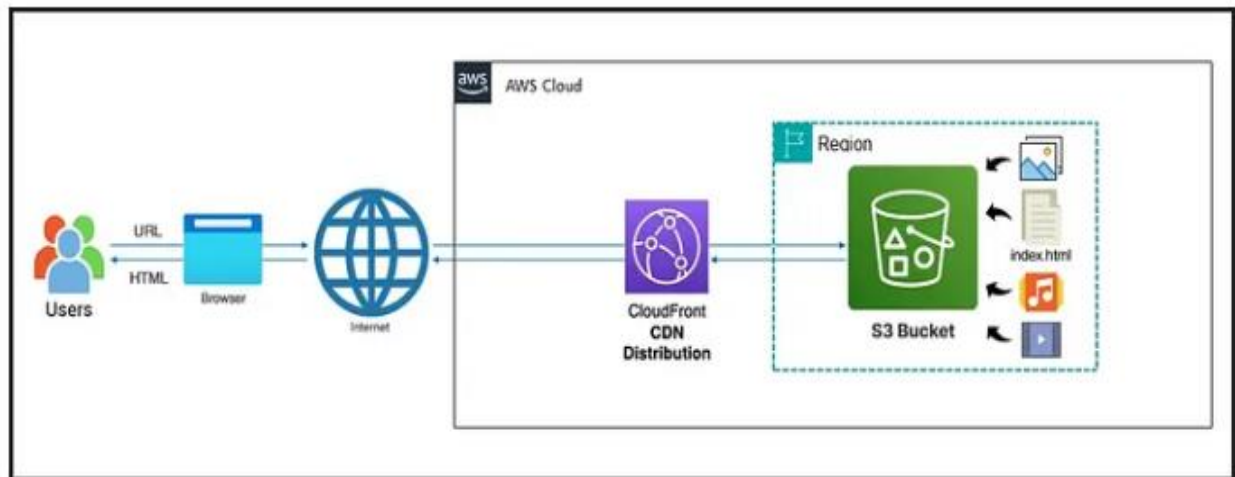
# Project: Hosting Static Website Efficiently Using AWS S3 and CloudFront

In this guide, we'll dive into the power of Amazon Web Services (AWS) to efficiently host and deliver static websites to users around the globe.

AWS S3 is a highly scalable storage service, and AWS CloudFront is a fast content delivery network (CDN), work together to make your website lightning fast and reliable.

In this project, hosting a static website using Amazon S3 and distributing it globally with CloudFront.

- Hosting a static site using S3
- Connecting it to CloudFront for faster global delivery
- Securing it using Origin Access Control (OAC)
- Enabling HTTPS with SSL for safe browsing
- Visualizing the full architecture

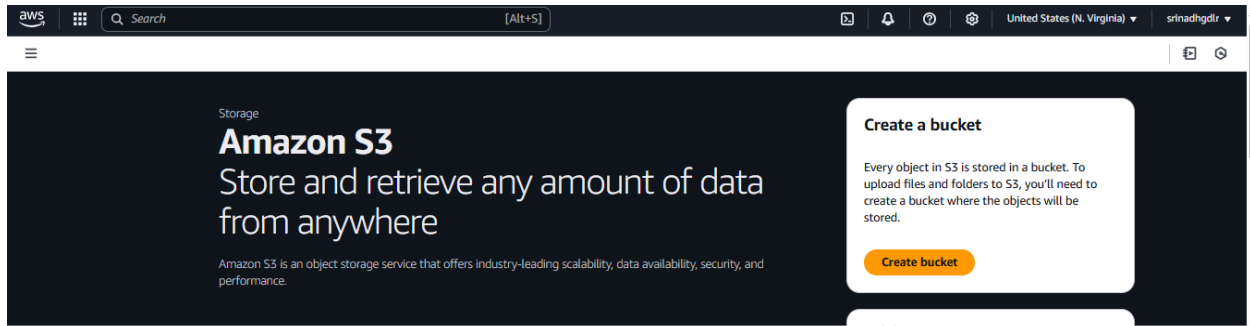


---

## Step 1: Create Amazon S3 Bucket:

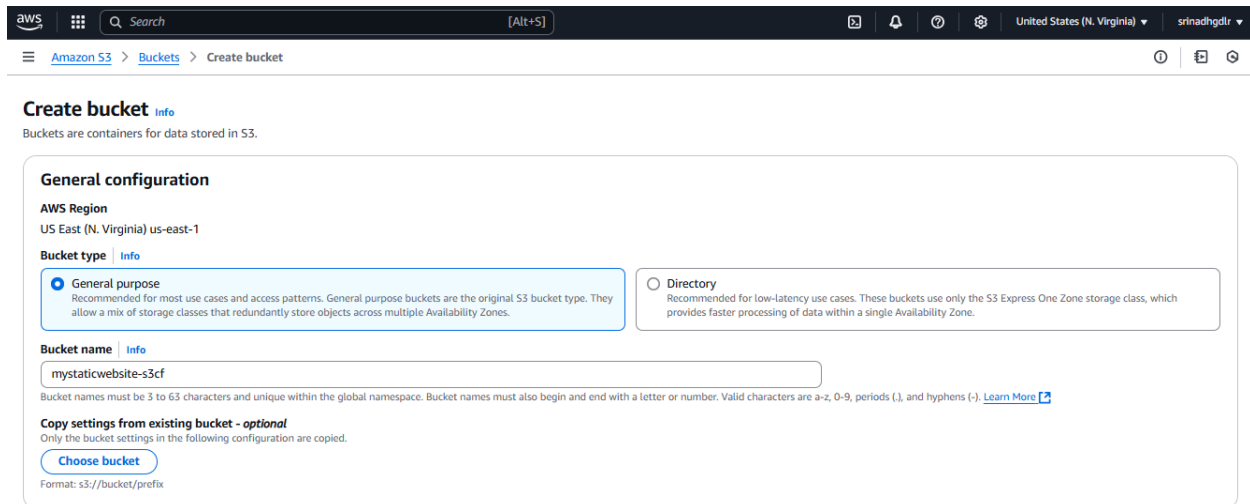
S3 bucket as a container where you will store all your website files - HTML, CSS, JavaScript, images, and other.

Search for Amazon S3 service within the AWS Management Console and Click on Create bucket button.



## Step 2: Configuring S3 Bucket:

Once you click “Create bucket,” you’ll be taken to a page where you need to configure the settings for your new bucket. This involves choosing the AWS Region, selecting the bucket type, and most importantly, giving your bucket a unique name.



Referring to above screenshot:

**AWS Region:** The first setting is to choose the AWS Region where you want your bucket to be located. It’s generally recommended to choose a region that is geographically closest to your target audience for lower latency. In this example, “US East (N. Virginia)” is selected.

**Bucket Type:** You’ll see options for “General purpose” and “Directory.” For hosting a static website, “General purpose” is the recommended and default choice, offering a balance of cost and performance. Ensure this option is selected.

**Bucket Name:** This is a crucial setting. You need to provide a globally unique name for your bucket. This name will also be part of your website’s URL later on. In the screenshot, “mystaticwebsite-s3cf” is used. Remember that bucket names have specific requirements: \* Must be between 3 and 63 characters long. \* Must be unique across all existing Amazon S3 bucket names globally. \* Must not contain uppercase letters or underscores. \* Must start and end with a letter or number. \* Can contain lowercase letters, numbers, periods (.), and hyphens (-).

## Step 3: Keeping Block all public access Enabled for Enhanced Security:

-Check the Block all public access and click Create bucket at the bottom.

**Block Public Access settings for this bucket**

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

☒ **Block all public access**  
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

- ☒ **Block public access to buckets and objects granted through new access control lists (ACLs)**  
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- ☒ **Block public access to buckets and objects granted through any access control lists (ACLs)**  
S3 will ignore all ACLs that grant public access to buckets and objects.
- ☒ **Block public access to buckets and objects granted through new public bucket or access point policies**  
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
- ☒ **Block public and cross-account access to buckets and objects through any public bucket or access point policies**  
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

**Bucket Versioning**

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. [Learn more](#)

**Bucket Versioning**

☒ Disable

☐ Enable

**Tags - optional (0)**

You can use bucket tags to track storage costs and organize buckets. [Learn more](#)

No tags associated with this bucket.

[Add tag](#)

**Default encryption** [info](#)

Server-side encryption is automatically applied to new objects stored in this bucket.

**Encryption type** [info](#)

☒ Server-side encryption with Amazon S3 managed keys (SSE-S3)

☐ Server-side encryption with AWS Key Management Service keys (SSE-KMS)

☐ Dual-layer server-side encryption with AWS Key Management Service keys (DSSE-KMS)  
Secure your objects with two separate layers of encryption. For details on pricing, see DSSE-KMS pricing on the [Storage](#) tab of the [Amazon S3 pricing page](#).

**Bucket Key**  
Using an S3 Bucket Key for SSE-KMS reduces encryption costs by lowering calls to AWS KMS. S3 Bucket Keys aren't supported for DSSE-KMS. [Learn more](#)

☐ Disable

☒ Enable

► **Advanced settings**

① After creating the bucket, you can upload files and folders to the bucket, and configure additional bucket settings.

[Cancel](#) [Create bucket](#)

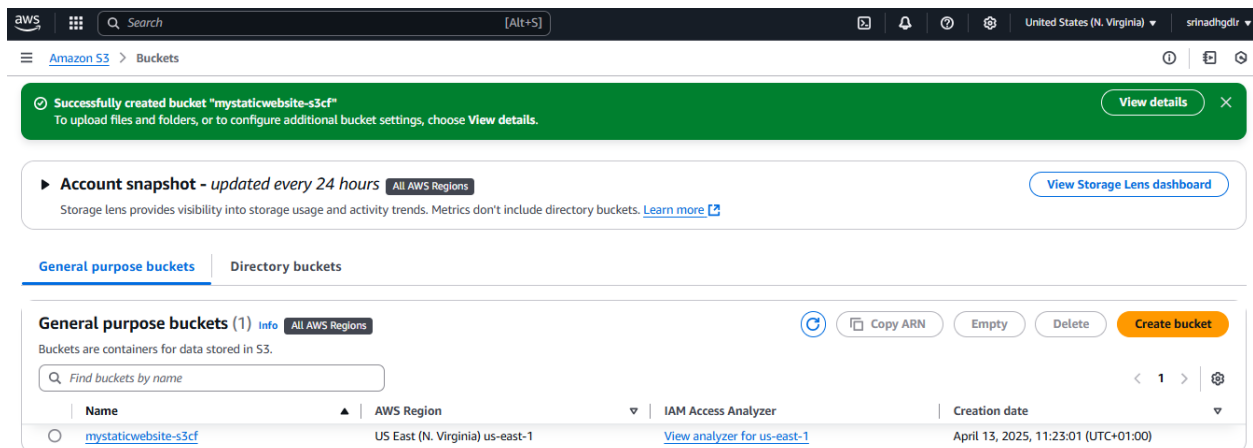
In this setup, we're taking a more secure approach by keeping the Amazon S3 bucket itself private. This means that the files stored in the bucket won't be directly accessible via a public S3 URL. Instead, we will control access through our CloudFront CDN. This method adds a layer of security by ensuring users can only access your content through the CloudFront distribution.

As shown in the screenshot : You'll find the "Block all public access" settings for this bucket section during the bucket creation process or in the bucket's permissions tab after creation.

Ensure that the “Block all public access” setting remains ON (checked). This is the recommended setting for this architecture, as it prevents direct public access to your S3 bucket.

By keeping this setting enabled, we ensure that all requests for your website content will go through CloudFront. We will later configure the CloudFront distribution to securely access the content in your private S3 bucket. This approach offers better control over who can access your content and can help prevent unauthorized access.

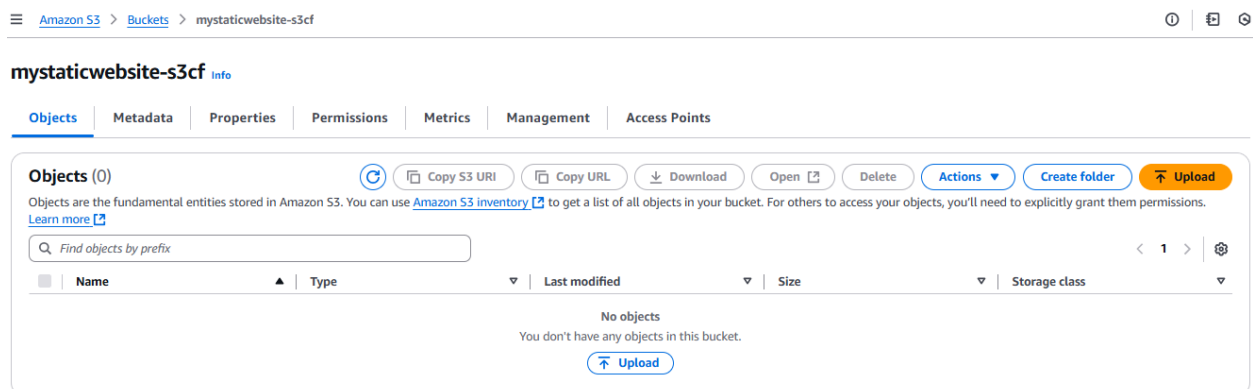
## Step 4: Amazon S3 Bucket is Ready:



Now newly created Amazon S3 bucket “mystaticwebsite-s3cf” is ready and should appear in your list of S3 buckets within the AWS Management Console.

## Step 5: Time to Upload Website Files:

Now that your S3 bucket is created and ready, it’s time to add your website content. As shown in the screenshot below, the bucket is currently empty:



click on Upload then click on Add Files

Amazon S3 > Buckets > mystaticwebsite-s3cf > Upload

Drag and drop files and folders you want to upload here, or choose **Add files** or **Add folder**.

**Files and folders** (4 total, 100.5 KB)

All files and folders in this table will be uploaded.

Find by name

<input type="checkbox"/>	Name	Folder	Type	Size
<input type="checkbox"/>	beach.jpg	-	image/jpeg	85.8 KB
<input type="checkbox"/>	index.html	-	text/html	3.3 KB
<input type="checkbox"/>	LICENSE.txt	-	text/plain	11.3 KB
<input type="checkbox"/>	README.md	-	-	41.0 B

**Destination** [info](#)

**Destination**

[s3://mystaticwebsite-s3cf](#)

**► Destination details**

Bucket settings that impact new objects stored in the specified destination.

**► Permissions**

Grant public access and access to other AWS accounts.

**► Properties**

Specify storage class, encryption settings, tags, and more.

[Cancel](#) [Upload](#)

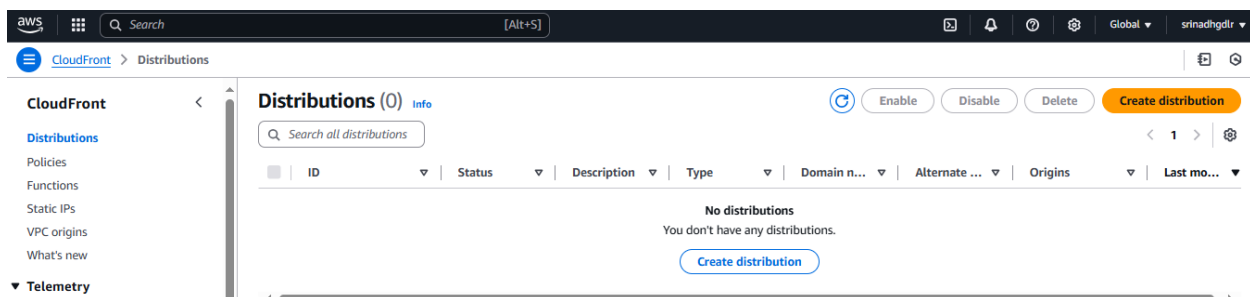
Once everything looks good, click the “Upload” button in the bottom-right corner. AWS will start uploading your content into the S3 bucket, files safely stored in your private S3 bucket.

## Step 6: Creating CloudFront Distribution:

Next crucial step is to set up AWS CloudFront. CloudFront will act as the content delivery network (CDN), caching your website content across its global network of edge locations. This ensures that your website loads quickly for users regardless of their geographic location.

Now navigate to the AWS CloudFront service page within the AWS Management Console.

On the CloudFront service page, click on “Create distribution” button to start the process of creating your CDN distribution.




## Configuring CloudFront distribution - Origin Settings:

☰ [CloudFront](#) > [Distributions](#) > Create

### Create distribution

#### Origin

##### Origin domain

Choose an AWS origin, or enter your origin's domain name. [Learn more](#) 

Q mystaticwebsite-s3cf.s3.us-east-1.amazonaws.com X

Enter a valid DNS domain name, such as an S3 bucket, HTTP server, or VPC origin ID.

##### Origin path - optional

Enter a URL path to append to the origin domain name for origin requests.

Enter the origin path

##### Name

Enter a name for this origin.

mystaticwebsite-s3cf.s3.us-east-1.amazonaws.com

##### Origin access | [Info](#)

☐ Public

Bucket must allow public access.

☒ Origin access control settings (recommended)

Bucket can restrict access to only CloudFront.

☐ Legacy access identities

Use a CloudFront origin access identity (OAI) to access the S3 bucket.

##### Origin access control

The first section is the “Origin” settings. This is where you tell CloudFront where to fetch your website content from.

Referring to the Screenshot :

Origin domain: Click in the “Origin domain” field. CloudFront will present you with a list of your S3 buckets.

Origin access: Here, you need to specify how CloudFront will access your private S3 bucket. Since we kept “Block all public access” enabled on the bucket, we need to use an Origin Access Control (OAC). As highlighted in the screenshot, select the option “Origin access control settings (recommended)”.

Origin access control settings: You’ll likely see a prompt to Create new OAC. We will configure this in the next step.

## Configuring Your CloudFront Distribution — Origin Settings (Continued)

**Origin access control**  
Select an existing origin access control (recommended) or create a new control.

Select an origin access control ▼

⊕ This field cannot be empty

Create new OAC

---

**Create new OAC** ✕

**Name**  
The name must be unique. Valid characters: letters, numbers and most special characters. Use up to 64 characters.

mystaticwebsite-s3cf.s3.us-east-1.amazonaws.com

**Description - optional**  
The description can have up to 256 characters.

Enter description

**Signing behavior**

☐ Do not sign requests

☒ Sign requests (recommended)

☐ Do not override authorization header  
Do not sign if incoming request has authorization header.

**Origin type**

S3 ▼

The origin type must be the same type as origin domain.

Cancel Create

---

**Origin access control**  
Select an existing origin access control (recommended) or create a new control.

Cloudfront-access-to-s3 ▼

Origin access control settings: As seen in the previous screenshot, you selected “Origin access control settings (recommended)”. Now, you should see a button that says either “Create control setting” or a dropdown menu if you have created OACs before. Click the button to create a new OAC.

Origin access control - Name: A dialog box titled “Create new OAC” will appear, as shown in the screenshot. Provide a descriptive name for your OAC. For example, in the screenshot, “Cloudfront-access-to-S3” is used.

Origin access control - Signing behavior: Ensure “Sign requests (recommended)” is selected, as it is in the screenshot. This ensures that CloudFront signs all requests it sends to your S3 bucket, enhancing security.

Origin type: This should automatically be set to “S3” since your origin is an S3 bucket. Confirm that it is set correctly.

Click the “Create” button. CloudFront will now create the OAC.

The screenshot shows the 'Create new OAC' dialog box in the AWS CloudFront console. At the top, the title is 'Origin access control' with a subtitle 'Select an existing origin access control (recommended) or create a new control.' Below this is a dropdown menu showing 'Cloudfront-access-to-s3' with a downward arrow. To the right of the dropdown is a button labeled 'Create new OAC'. Below the dropdown is a yellow warning box with a triangle icon and the text: 'You must update the S3 bucket policy. CloudFront will provide you with the policy statement after creating the distribution.' Below the warning box is a section titled 'Add custom header - optional' with the subtitle 'CloudFront includes this header in all requests that it sends to your origin.' and a button labeled 'Add header'. Below this is a section titled 'Enable Origin Shield' with the subtitle 'Origin shield is an additional caching layer that can help reduce the load on your origin and help protect its availability.' and two radio buttons: 'No' (selected) and 'Yes'. At the bottom is a link labeled 'Additional settings'.

## Configuring Your CloudFront Distribution — Default Cache Behavior Settings

Next, you’ll configure the “Default cache behavior settings.” These settings control how CloudFront caches your content and how it handles requests.



The screenshot shows the AWS CloudFront console interface. At the top, there's a navigation bar with the AWS logo, a search bar, and user information. Below the navigation bar, the breadcrumb trail reads 'CloudFront > Distributions > Create'. The main content area is titled 'Default cache behavior'. It includes several sections: 'Path pattern' with a text input field containing 'Default (\*)'; 'Compress objects automatically' with radio buttons for 'No' and 'Yes' (selected); 'Viewer' section containing 'Viewer protocol policy' with radio buttons for 'HTTP and HTTPS', 'Redirect HTTP to HTTPS' (selected), and 'HTTPS only'; 'Allowed HTTP methods' with radio buttons for 'GET, HEAD' (selected), 'GET, HEAD, OPTIONS', and 'GET, HEAD, OPTIONS, PUT, POST, PATCH, DELETE'; 'Restrict viewer access' with a note and radio buttons for 'No' (selected) and 'Yes'; and 'Cache key and origin requests' at the bottom.

**Compress objects automatically:** You can choose “Yes” to have CloudFront automatically compress eligible files (like text files, HTML, CSS, JavaScript) before sending them to the viewer. This can improve loading times.

**Viewer protocol policy:** Choose “Redirect HTTP to HTTPS” to ensure that all traffic to your website is secure. This will automatically redirect any HTTP requests to HTTPS.

**Allowed HTTP methods:** For static websites, select “GET, HEAD”. These are the only methods needed to retrieve and check the existence of your content. Disabling other methods adds a small layer of security.

**Cache key and origin requests:** You can typically leave this as the default setting, which is usually “Cache policy and origin request policy (recommended)”. This allows CloudFront to use its managed policies for optimal caching and request handling for most static website scenarios.

**Cache policy:** CloudFront offers managed cache policies. For static websites, the default “Managed-CachingOptimized” policy often works well. You might see this selected or be able to choose it from a dropdown.

**Origin request policy:** Similarly, for origin request policies, the default “Managed-AllViewer” policy is often suitable for static websites. This forwards all viewer headers, cookies, and query strings to the origin. For a basic static website, you might not need to forward cookies or query strings, and you could potentially select a more restrictive policy if needed for advanced use cases.

### Cache key and origin requests

We recommend using a cache policy and origin request policy to control the cache key and origin requests.

- ☒ Cache policy and origin request policy (recommended)  
☐ Legacy cache settings

#### Cache policy

Choose an existing cache policy or create a new one.

CachingOptimized  
Policy with caching enabled. Supports Gzip and Brotli compression. Recommended for S3

[Create cache policy](#) [View policy](#)

#### Origin request policy - optional

Choose an existing origin request policy or create a new one.

AllViewerAndCloudFrontHeaders-2022-06  
Policy to forward all parameters in viewer requests and all CloudFront headers as of June 2022

[Create origin request policy](#) [View policy](#)

#### Response headers policy - optional

Choose an existing response headers policy or create a new one.

Select response headers

[Create response headers policy](#)

#### ► Additional settings

**Default root object:** This is the file that CloudFront will serve when a user requests the root of your domain (e.g., /). As shown in the screenshot, typically, this will be your main HTML file, named index.html. Enter index.html in this field.

#### Default root object - optional

The object (file name) to return when a viewer requests the root URL (/) instead of a specific object.

index.html

### Web Application Firewall (WAF)

In this guide, we do not enable AWS WAF for the sake of simplicity and to keep the setup beginner-friendly, as shown in the screenshot. However, for production-level or more critical applications, it is highly recommended to enable AWS WAF. AWS WAF helps protect your application from common web exploits such as SQL injection and cross-site scripting (XSS).

#### Web Application Firewall (WAF) Info

##### ☐ Enable security protections

Keep your application secure from the most common web threats and security vulnerabilities using AWS WAF. Blocked requests are stopped before they reach your web servers.

##### ☒ Do not enable security protections

Select this option if your application does not need security protections from AWS WAF.

#### Standard logging Info

Additional charges may apply. See Info for more details.

##### Log delivery

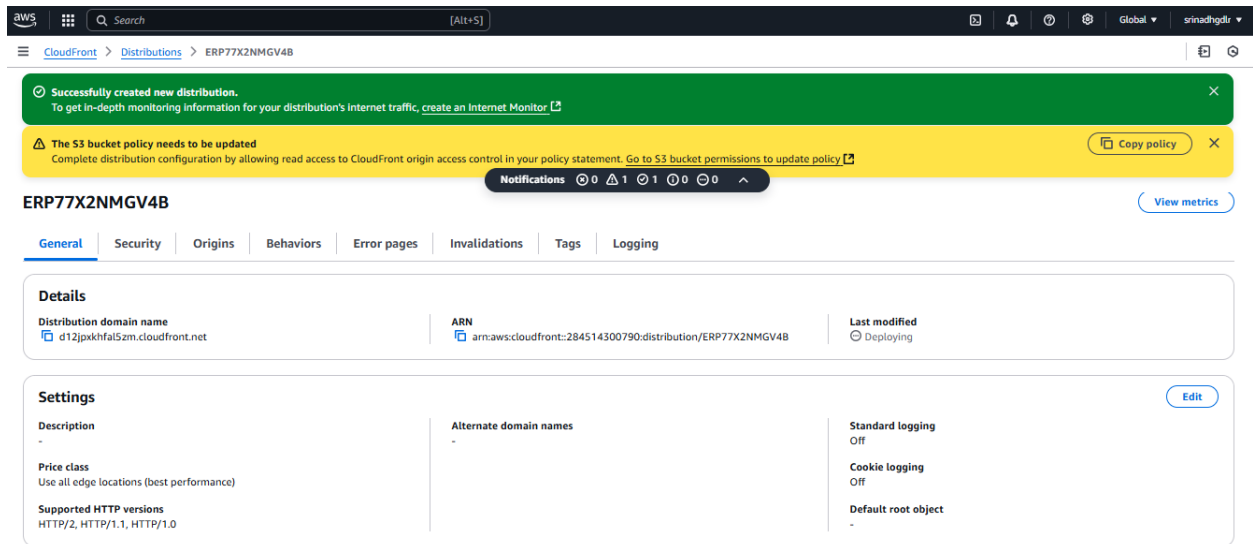
Get logs of viewer requests to CloudWatch, Amazon S3 or Firehose

- ☒ Off  
☐ On

[Cancel](#)

[Create distribution](#)

Once you have configured all the necessary settings, review them carefully and click the “Create distribution” button

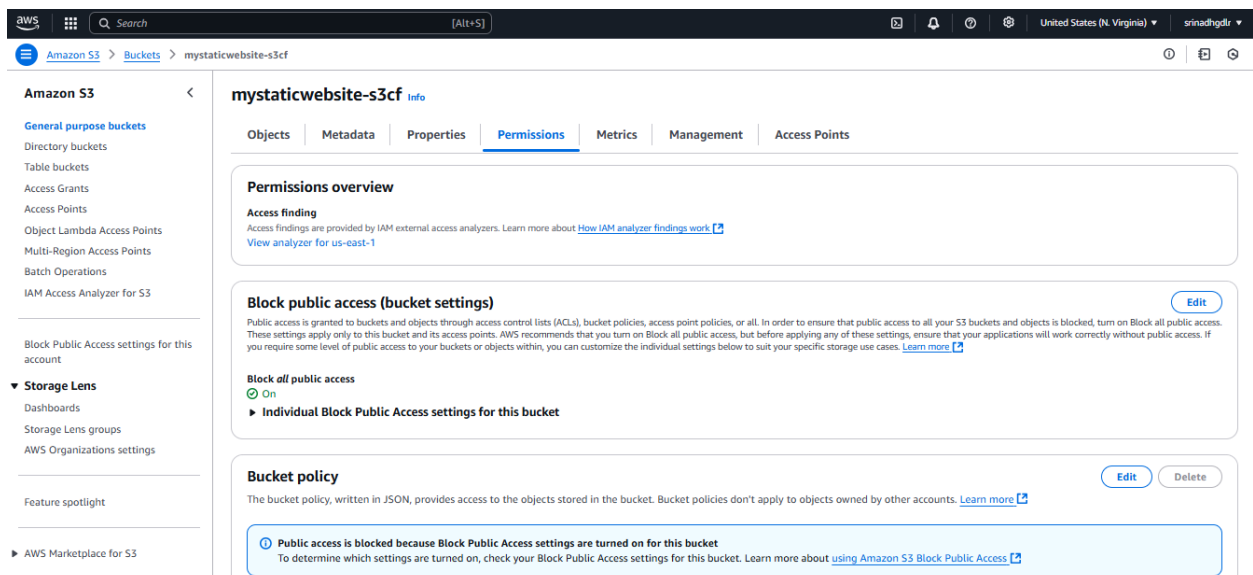


after you click the “Create distribution” button, you should see a green banner at the top indicating that the distribution was successfully created.

However, you’ll also notice a yellow banner with a warning message: “The S3 bucket policy needs to be updated.” This is a crucial step because we configured our S3 bucket to block all public access and are using Origin Access Control (OAC) to allow CloudFront to access the content.

As highlighted in the yellow banner, you need to grant CloudFront permission to access your private S3 bucket. The banner provides a “Copy policy” button. Click this button to copy the necessary bucket policy. We will now proceed to update the S3 bucket policy in the next step.

## Step 7: Updating S3 Bucket Policy:



Now that you've copied the bucket policy from the CloudFront console, you need to apply it to your S3 bucket.

Navigate back to your S3 bucket in the AWS Management Console.

Go to the "Permissions" tab of your bucket.

Scroll down to the "Bucket policy" section. You should see an "Edit" button on the right side. Click this button to open the policy editor.

The screenshot shows the AWS Management Console interface for editing a bucket policy. The breadcrumb navigation at the top reads: Amazon S3 > Buckets > mystaticwebsite-s3cf > Edit bucket policy. On the left sidebar, the 'Storage Lens' section is expanded, showing options like Dashboards, Storage Lens groups, and AWS Organizations settings. The main content area is titled 'Edit bucket policy' and includes a 'Bucket policy' section with a description and a 'Learn more' link. Below this, the 'Bucket ARN' is listed as 'arn:aws:s3::mystaticwebsite-s3cf'. The 'Policy' section contains a JSON editor with line numbers 1 through 20. The JSON code is as follows: 

```
{
  "Version": "2008-10-17",
  "Id": "PolicyForCloudFrontPrivateContent",
  "Statement": [
    {
      "Sid": "AllowCloudFrontServicePrincipal",
      "Effect": "Allow",
      "Principal": {
        "Service": "cloudfront.amazonaws.com"
      },
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3::mystaticwebsite-s3cf/*",
      "Condition": {
        "StringEquals": {
          "AWS:SourceArn": "arn:aws:cloudfront::284514380790:distribution/ERP77X2NMQV48"
        }
      }
    }
  ]
}
```

 To the right of the JSON editor is an 'Edit statement' panel with a 'Select a statement' section and an '+ Add new statement' button. At the bottom of the editor, there is a status bar showing 'JSON Ln 20, Col 7', security and error counts (all zero), and a 'Preview external access' link. At the very bottom right, there are 'Cancel' and 'Save changes' buttons.

In the policy editor, paste the policy that you copied from the CloudFront console in the previous step. This policy specifically grants the Origin Access Control (OAC) you created for your CloudFront distribution the permission to get objects from your bucket.

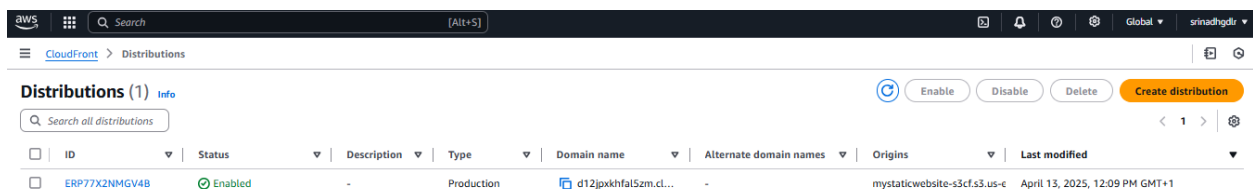
After pasting the policy, click the "Save changes" button. Your S3 bucket is now configured to allow access only from your CloudFront distribution through the OAC.

## Step 8: Accessing Your Website via CloudFront:

Once your CloudFront distribution is created, it will take some time for it to be fully deployed across all of CloudFront’s edge locations. You can monitor the status of your distribution on the CloudFront console.

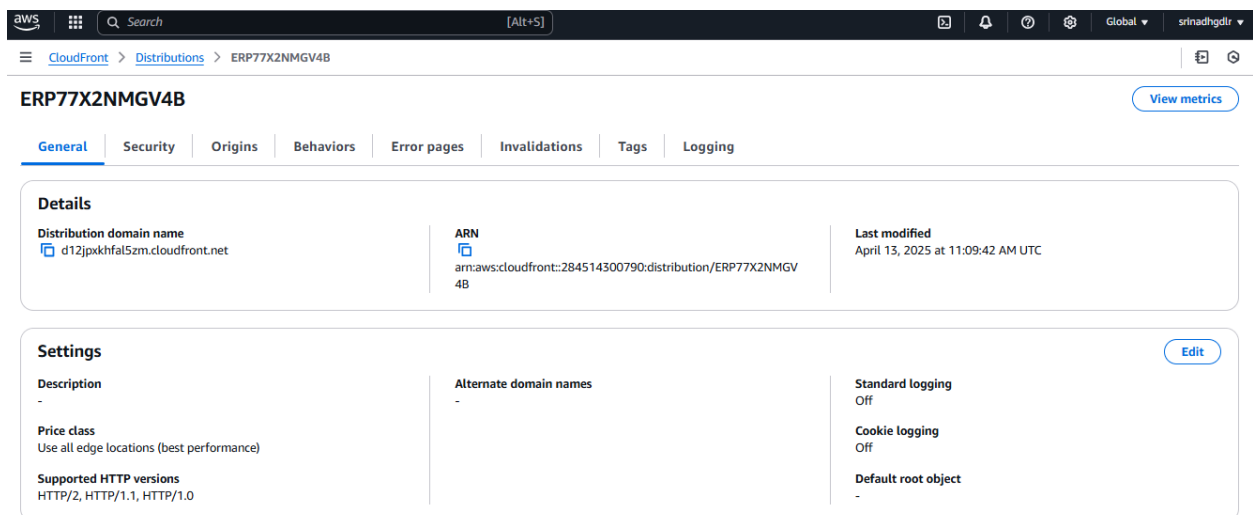
Navigate to the CloudFront service in the AWS Management Console and click on “Distributions” in the left-hand menu. You should see your newly created distribution in the list.

The “Status” column will initially show “Deploying.” This process can take several minutes, depending on the configuration. Once the deployment is complete, the status will change to “Enabled.”



The screenshot shows the AWS CloudFront console with the 'Distributions' tab selected. A table lists the distributions, with one distribution 'ERP77X2NMGV4B' shown in an 'Enabled' state.

ID	Status	Description	Type	Domain name	Alternate domain names	Origins	Last modified
ERP77X2NMGV4B	Enabled	-	Production	d12jpxkhfal5zm.cl...	-	mystaticwebsite-s3cf.s3.us-e...	April 13, 2025, 12:09 PM GMT+1



The screenshot shows the details and settings for the CloudFront distribution 'ERP77X2NMGV4B'. The 'General' tab is selected, showing details and settings.

### ERP77X2NMGV4B

[View metrics](#)

**General** | Security | Origins | Behaviors | Error pages | Invalidations | Tags | Logging

#### Details

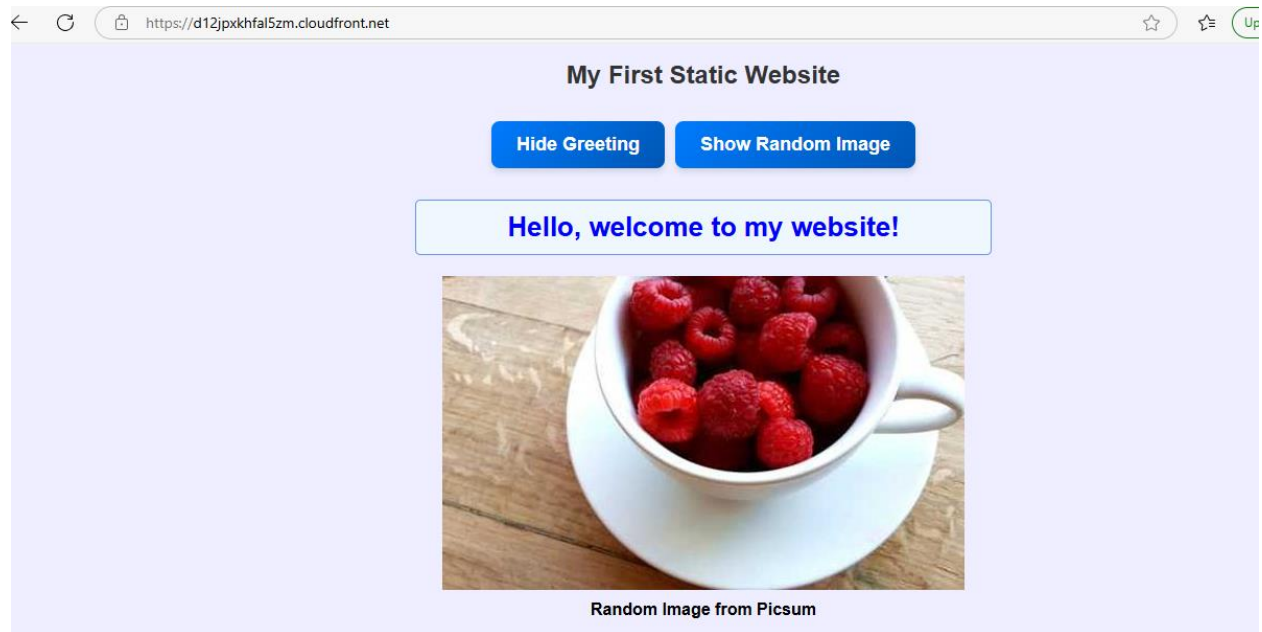
<b>Distribution domain name</b> d12jpxkhfal5zm.cloudfront.net	<b>ARN</b> arn:aws:cloudfront::284514300790:distribution/ERP77X2NMGV4B	<b>Last modified</b> April 13, 2025 at 11:09:42 AM UTC
--	---	---

#### Settings

<b>Description</b> -	<b>Alternate domain names</b> -	<b>Standard logging</b> Off
<b>Price class</b> Use all edge locations (best performance)		<b>Cookie logging</b> Off
<b>Supported HTTP versions</b> HTTP/2, HTTP/1.1, HTTP/1.0		<b>Default root object</b> -

[Edit](#)

Once it’s enabled, you can copy the Distribution domain name and paste it into your web browser to see your website live.



Congratulations on reaching this incredible step! Your static website should now be loading lightning-fast, and super efficiently through the power of AWS S3 and CloudFront!

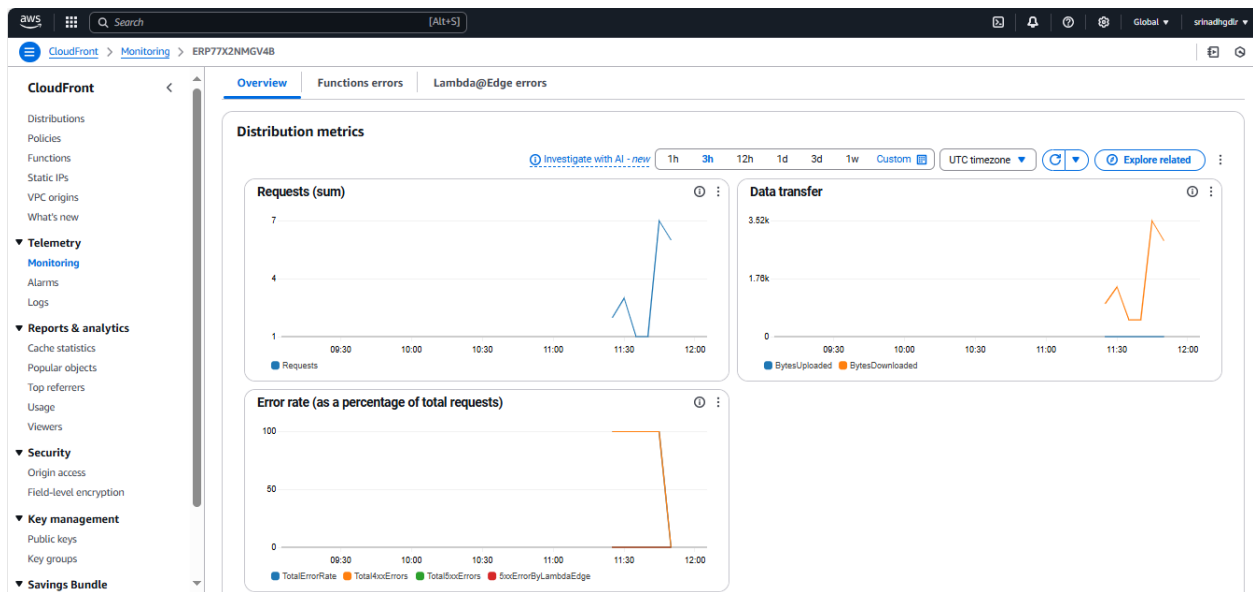
---

## Last Step: Monitoring CloudFront Distribution:

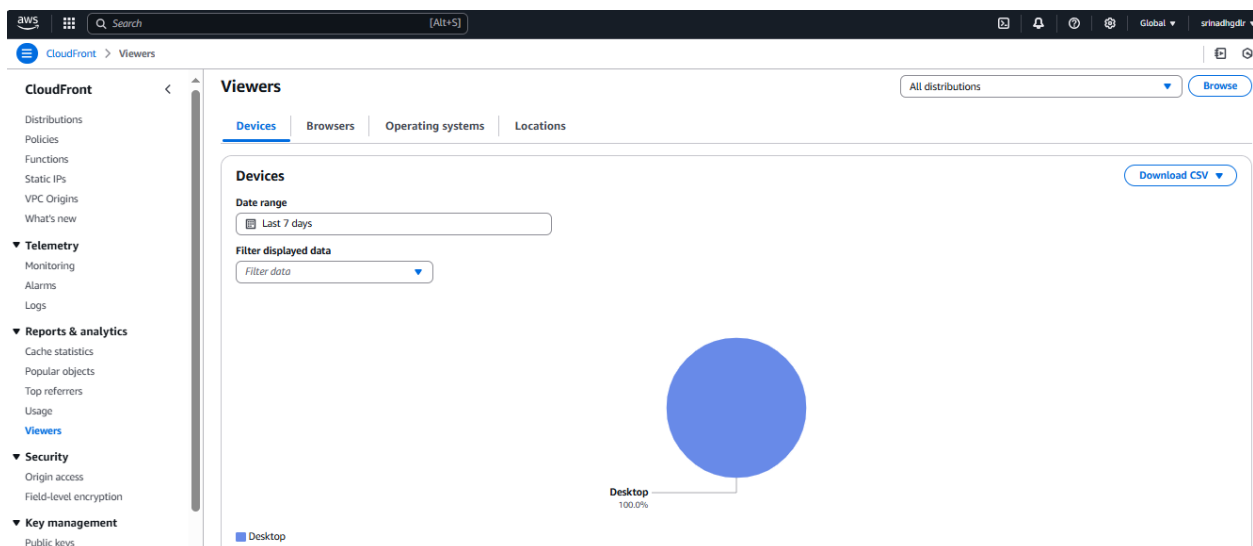
Once your static website is up and running, it's essential to monitor its performance and understand. AWS CloudFront provides built-in monitoring tools that allow you to gain insights into how your content is being delivered and accessed.

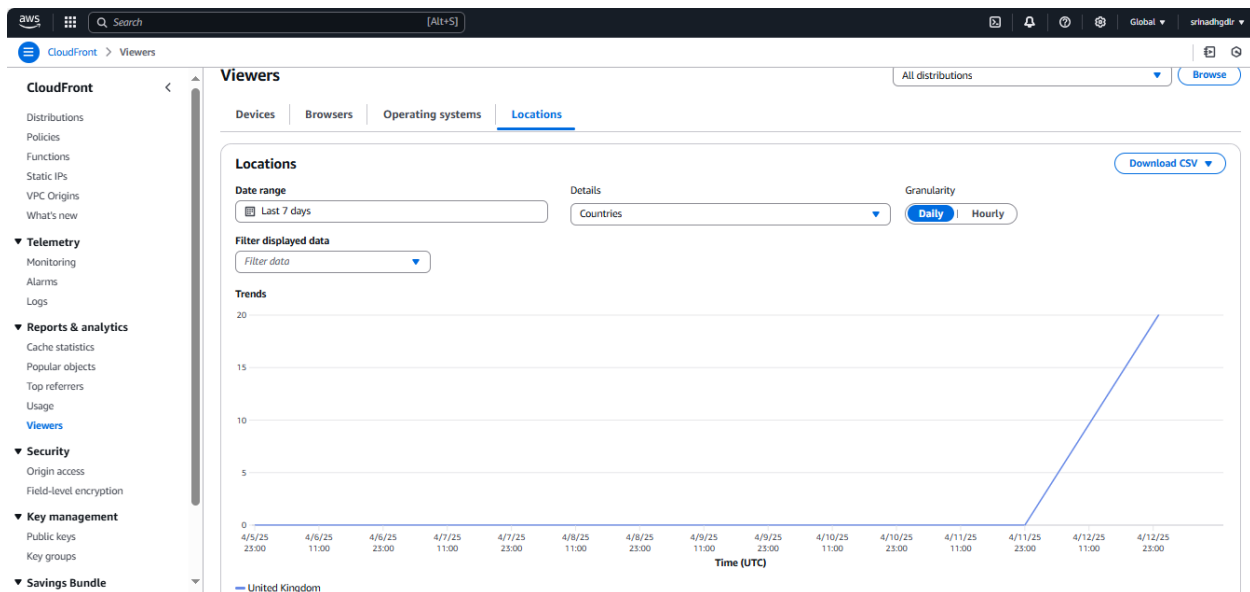
In the AWS Management Console, navigate to the CloudFront service. In the left-hand menu, under the "Telemetry" section, you'll find options like "Monitoring," "Alarms," and "Logs."

Click on "Monitoring" to get an overview of your distribution's performance, including metrics like request counts, Data transfer, error rates, and latency.



For a deeper understanding of your viewers, click on “Viewers” (also under “Telemetry”). Here, as illustrated in the screenshot, you can see valuable data such as:





Devices: The types of devices your users are using to access your website (e.g., Desktop, Mobile, Tablet).

Browsers: The web browsers your visitors are using.

Operating systems: The operating systems of your users' devices.

Locations: The geographic locations from where your website is being accessed.

These monitoring tools help you understand your audience better, identify potential issues, and optimize your website's performance over time. Regularly checking these metrics can provide valuable insights into your website's usage and help you make informed decisions about future improvements.

*You have successfully deployed your static website using Amazon S3 and CloudFront, leveraging AWS's scalable and globally distributed content delivery network (CDN) for optimal performance, security, and reliability. Your website is now accessible with low latency and high availability, ensuring a seamless experience for users worldwide.*

=====END=====