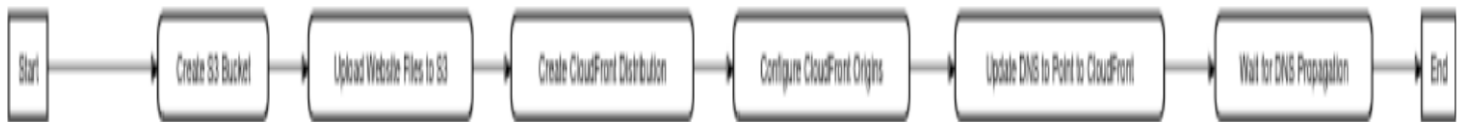


Task: Deploy Website to S3 with CloudFront.

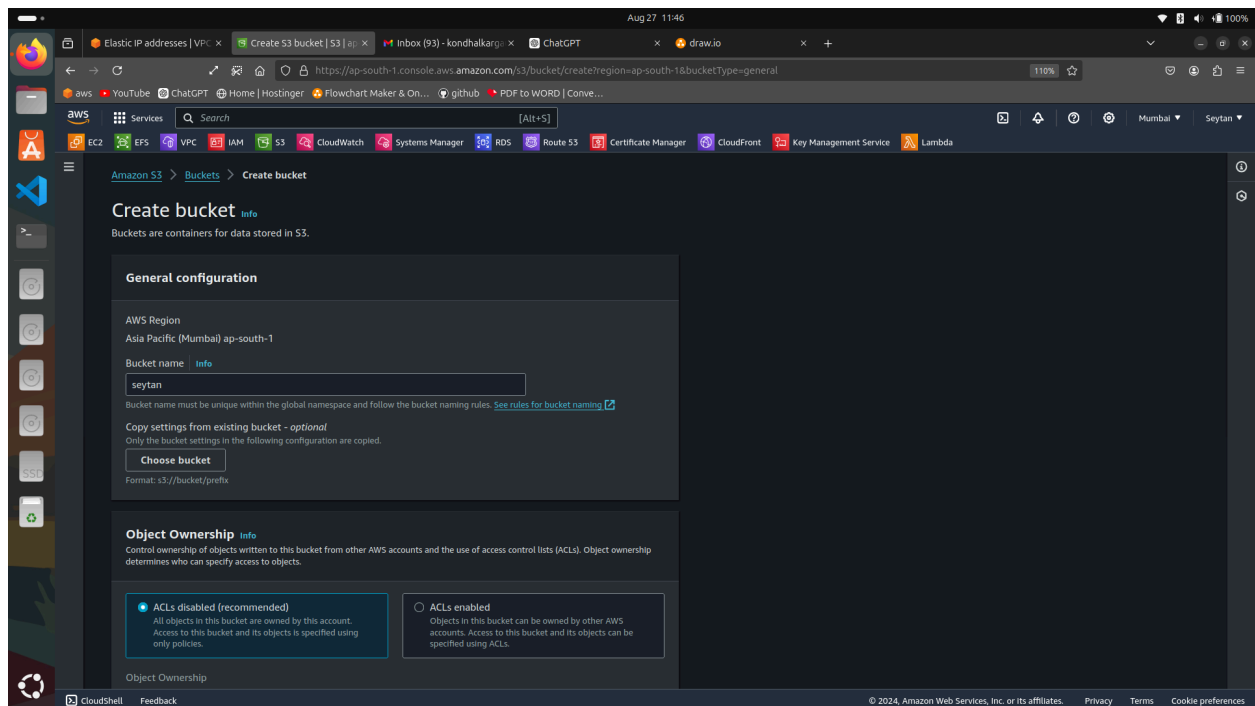
Diagrammatic Representation:



Step 1: Create an S3 Bucket.

What is S3?

→Amazon S3 (Simple Storage Service) is a scalable cloud storage service from AWS for storing and retrieving any amount of data. It provides high durability, availability, and security for data storage needs.



Step 2: Upload Webpage File i.e index.html to S3 bucket.

Drag and drop files and folders you want to upload here, or choose [Add files](#) or [Add folder](#).

Files and folders (1 Total, 295.0 B)

[Remove](#) [Add files](#) [Add folder](#)

All files and folders in this table will be uploaded.

Find by name

< 1 >

<input type="checkbox"/>	Name	Folder
<input type="checkbox"/>	index.html	-

Destination [Info](#)

Destination

s3://seytan

► 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

Step 3: Now Go to the Permission tab of S3 bucket and uncheck the block public access.

Amazon S3 > Buckets > seytan > Edit Block public access (bucket settings)

Edit Block public access (bucket settings) [Info](#)

Block public access (bucket settings)

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 all your S3 buckets and 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 your buckets 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.

[Cancel](#) [Save changes](#)

Step 4: Now in same permission tabs select the edit object ownership and make it ACLs enabled.

Amazon S3 > Buckets > seytan > Edit Object Ownership

Edit Object Ownership [Info](#)

Object Ownership

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

☐ **ACLs disabled (recommended)**
All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

☒ **ACLs enabled**
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

⚠️ We recommend disabling ACLs, unless you need to control access for each object individually or to have the object writer own the data they upload. Using a bucket policy instead of ACLs to share data with users outside of your account simplifies permissions management and auditing.

⚠️ Enabling ACLs turns off the bucket owner enforced setting for Object Ownership
Once the bucket owner enforced setting is turned off, access control lists (ACLs) and their associated permissions are restored. Access to objects that you do not own will be based on ACLs and not the bucket policy.
☐ I acknowledge that ACLs will be restored.

Object Ownership

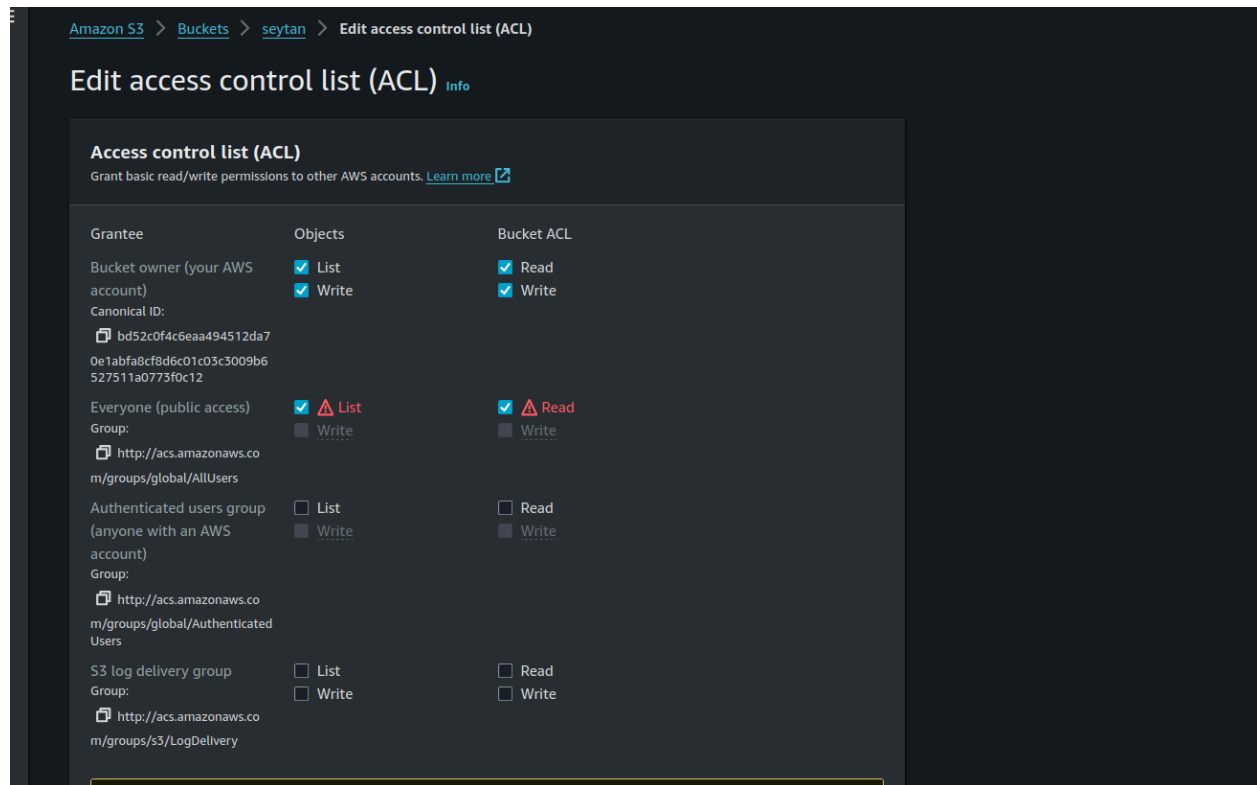
☒ **Bucket owner preferred**
If new objects written to this bucket specify the bucket-owner-full-control canned ACL, they are owned by the bucket owner. Otherwise, they are owned by the object writer.

☐ **Object writer**
The object writer remains the object owner.

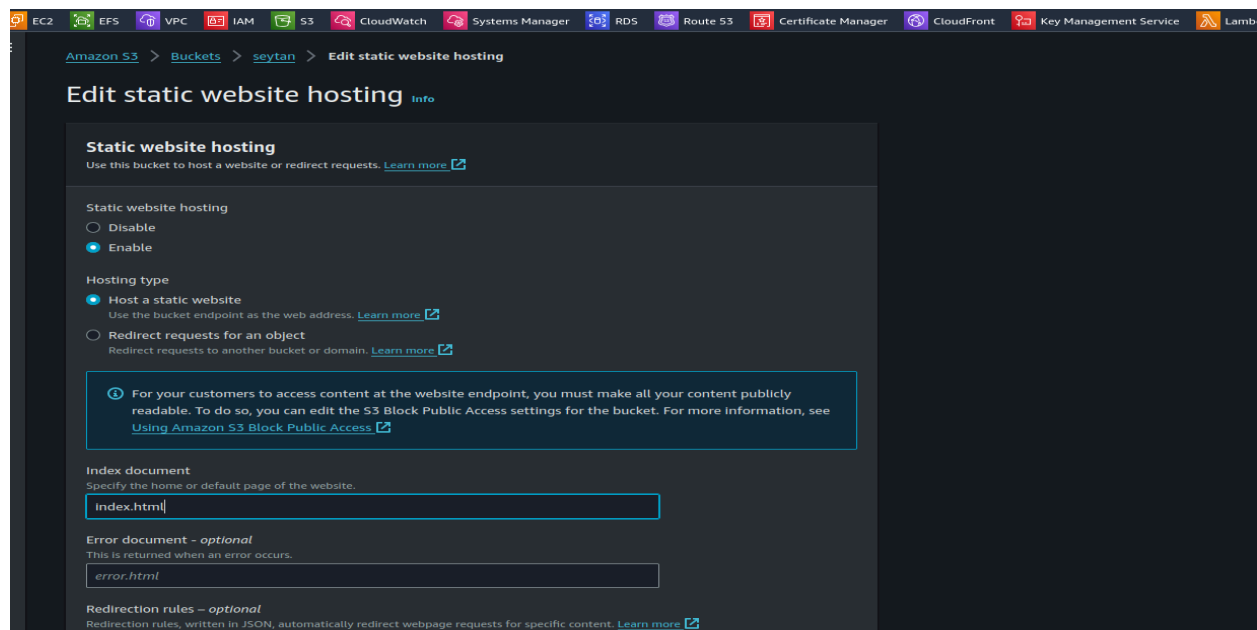
🔗 If you want to enforce object ownership for new objects only, your bucket policy must specify that the bucket-owner-full-control canned ACL is required for object uploads. [Learn more](#)

[Cancel](#) [Save changes](#)

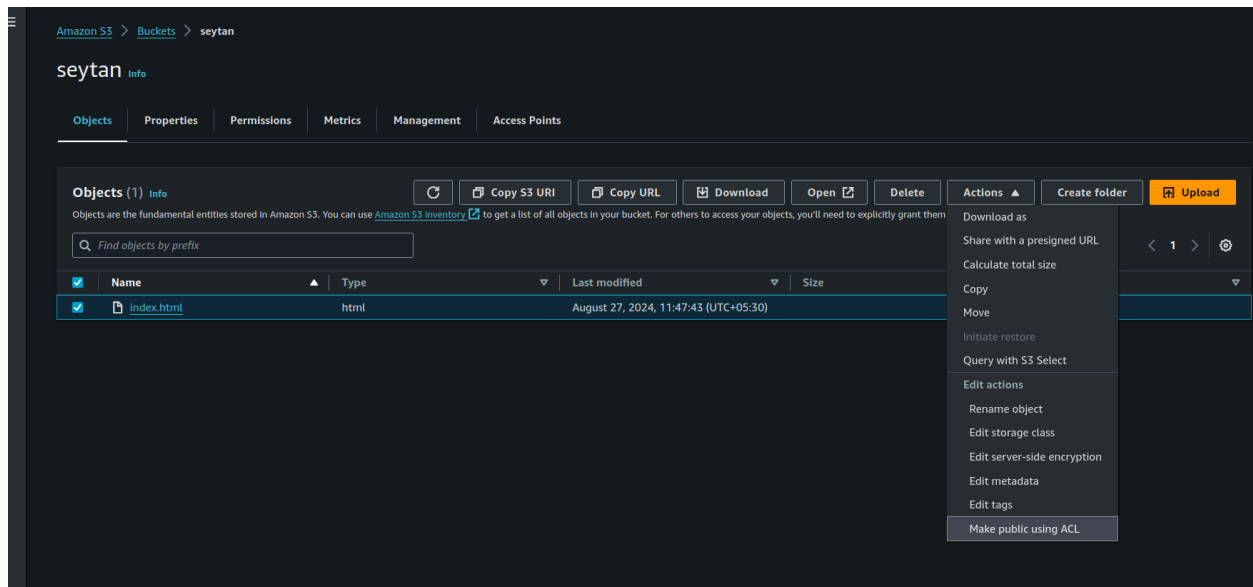
Step 5: Now change the permission for ACL. In this allow the everyone (public access to list and read).



Step 6: Now go to the Properties tab of bucket and at the end enable the static website hosting and give index.html as default page.

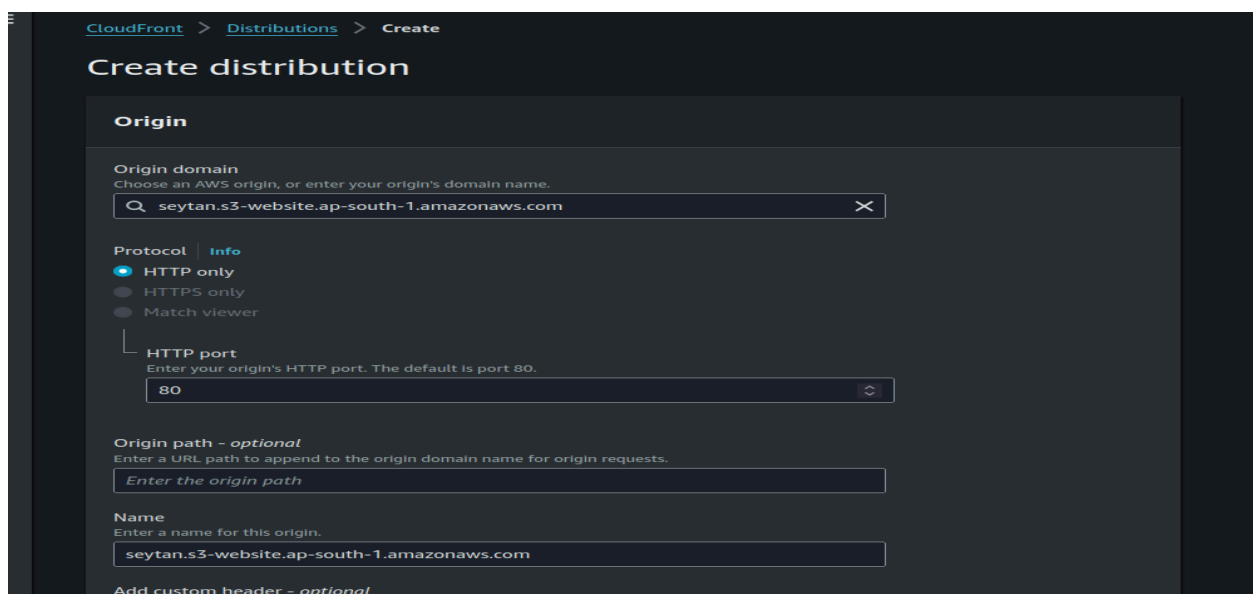


Step 7: Now select the object and click on actions tab. In this select the last option make public using ACL and enable it .

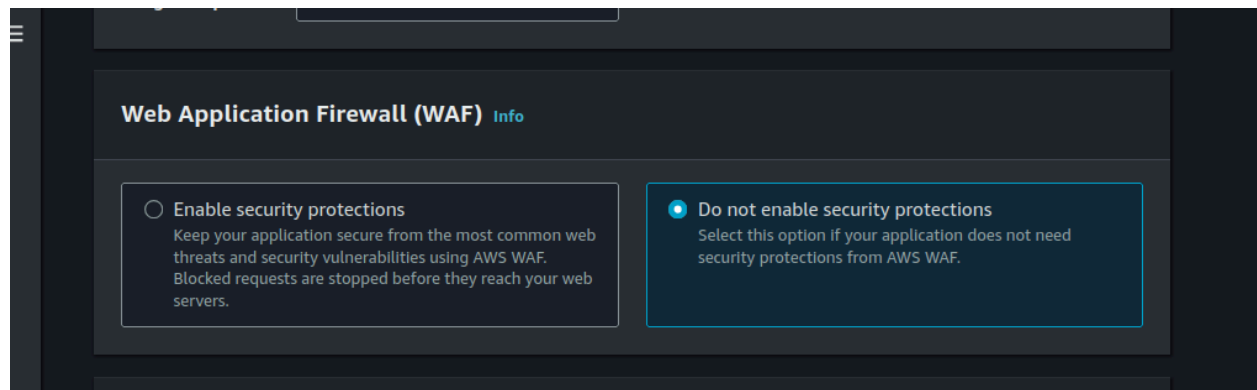


Step 8: Now create a Distribution in CloudFront using s3 as origin Domain.

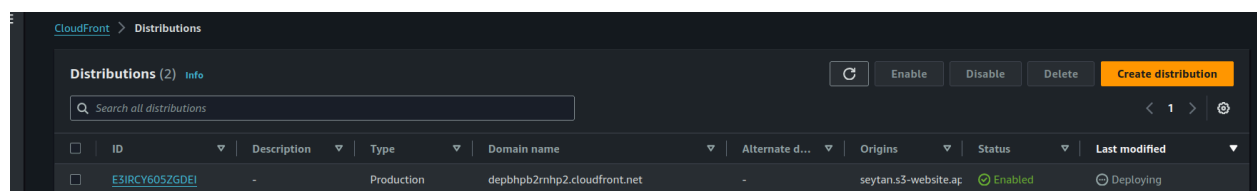
What is CloudFront ?→Amazon CloudFront is a content delivery network (CDN) service from AWS that speeds up the delivery of web content by distributing it to edge locations closer to users. It enhances performance and reduces latency for delivering websites, APIs, and media content.



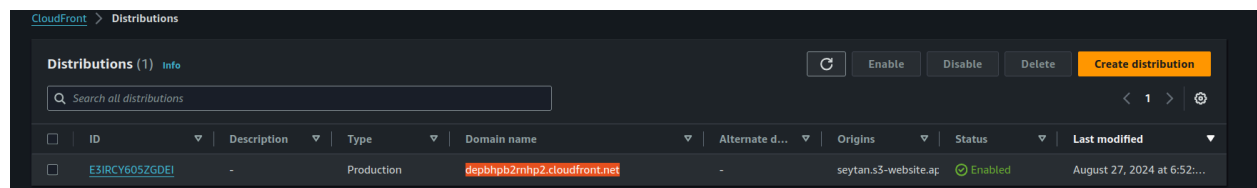
Select the WAF to do not enable as we dont need this now.



Step 9: Wait till the deploying changes from to a particular time of deployed.



Step 10: After The Last Modified Changes from deploying to a time it copy the dns and paste it in the browser.



Output:

