Lab 4: CS 524

In this assignment, you will learn use S3 bucket and use CDN with Cloudfront and report the results.

ANS.

S3:

Amazon Simple Storage Service (Amazon S3) is an object storage service that offers industry-leading scalability, data availability, security, and performance. This means customers of all sizes and industries can use it to store and protect any amount of data for a range of use cases, such as data lakes, websites, mobile applications, backup and restore, archive, enterprise applications, IoT devices, and big data analytics. Amazon S3 provides easy-to-use management features so you can organize your data and configure finely tuned access controls to meet your specific business, organizational, and compliance requirements. Amazon S3 is designed for 99.999999999% (11 9's) of durability, and stores data for millions of applications for companies all around the world.

Bucket Policies:

A bucket policy is a resource-based AWS Identity and Access Management (IAM) policy. You add a bucket policy to a bucket to grant other AWS accounts or IAM users access permissions for the bucket and the objects in it. Object permissions apply only to the objects that the bucket owner creates. Bucket policies use JSON-based access policy language.

Access control list (ACL):

Amazon S3 access control lists (ACLs) enable you to manage access to buckets and objects. Each bucket and object has an ACL attached to it as a subresource. It defines which AWS accounts or groups are granted access and the type of access. When a request is received against a resource, Amazon S3 checks the corresponding ACL to verify that the requester has the necessary access permissions. When you create a bucket or an object, Amazon S3 creates a default ACL that grants the resource owner full control over the resource.

CloudFront:

Amazon CloudFront is a fast content delivery network (CDN) service that securely delivers data, videos, applications, and APIs to customers globally with low latency, high transfer speeds, all within a developer-friendly environment.

CloudFront offers the most advanced security capabilities, including field level encryption and HTTPS support, seamlessly integrated with AWS Shield, AWS Web Application Firewall and Route 53 to protect against multiple types of attacks including network and application layer DDoS attacks. These services co-reside at edge networking locations – globally scaled and connected via the AWS network backbone – providing a more secure, performant, and available experience for your users.

CloudFront works seamlessly with any AWS origin, such as Amazon S3, Amazon EC2, Elastic Load Balancing, or with any custom HTTP origin.

Origin:

An origin is the location where content is stored, and from which CloudFront gets content to serve to viewers.

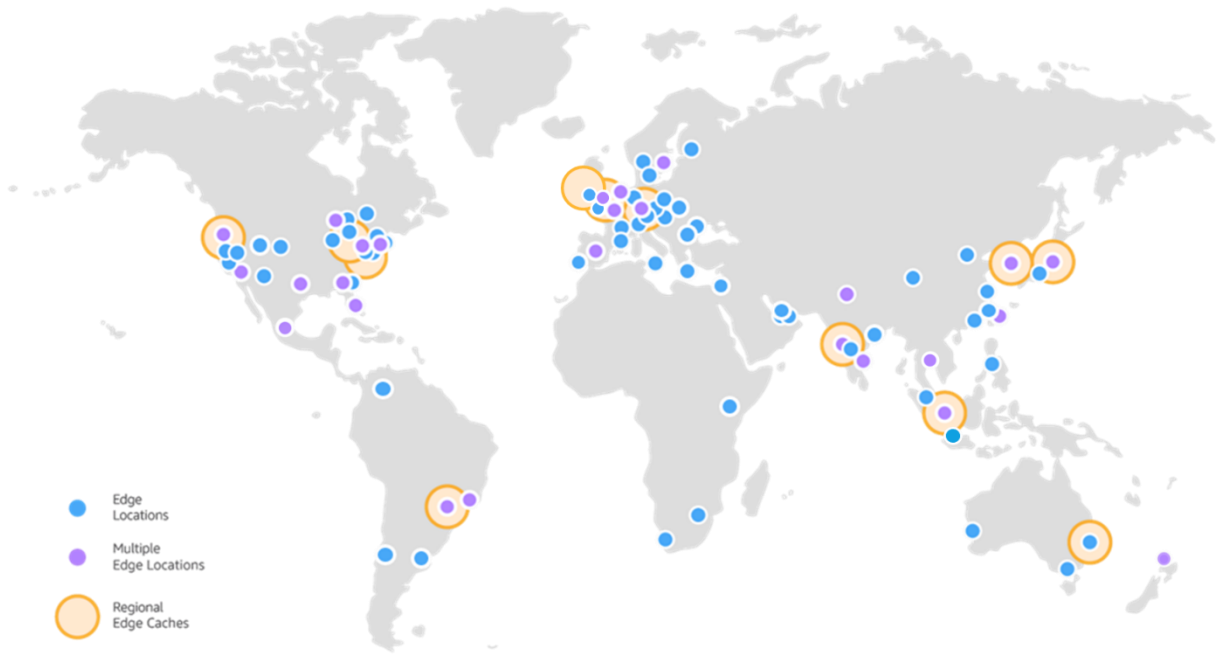
Edge:

CloudFront delivers your content through a worldwide network of data centers called edge locations. When a user requests content that you're serving with CloudFront, the request is routed to the edge location that provides the lowest latency (time delay), so that content is delivered with the best possible performance.

Location:

Amazon CloudFront peers with thousands of Tier 1/2/3 telecom carriers globally, is well connected with all major access networks for optimal performance and has hundreds of terabits of deployed capacity. CloudFront Edge locations are connected to the AWS Regions through the AWS network backbone - fully redundant, multiple 100GbE parallel fiber that circles the globe and links with tens of thousands of networks for improved origin fetches and dynamic content acceleration.

To deliver content to end users with lower latency, Amazon CloudFront uses a global network of 225+ Points of Presence (215+ Edge locations and 13 regional mid-tier caches) in 90 cities across 47 countries. Amazon CloudFront Edge locations are located in:

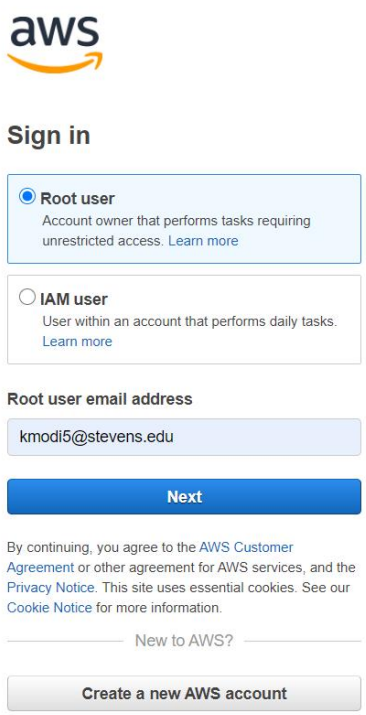


Distribution:

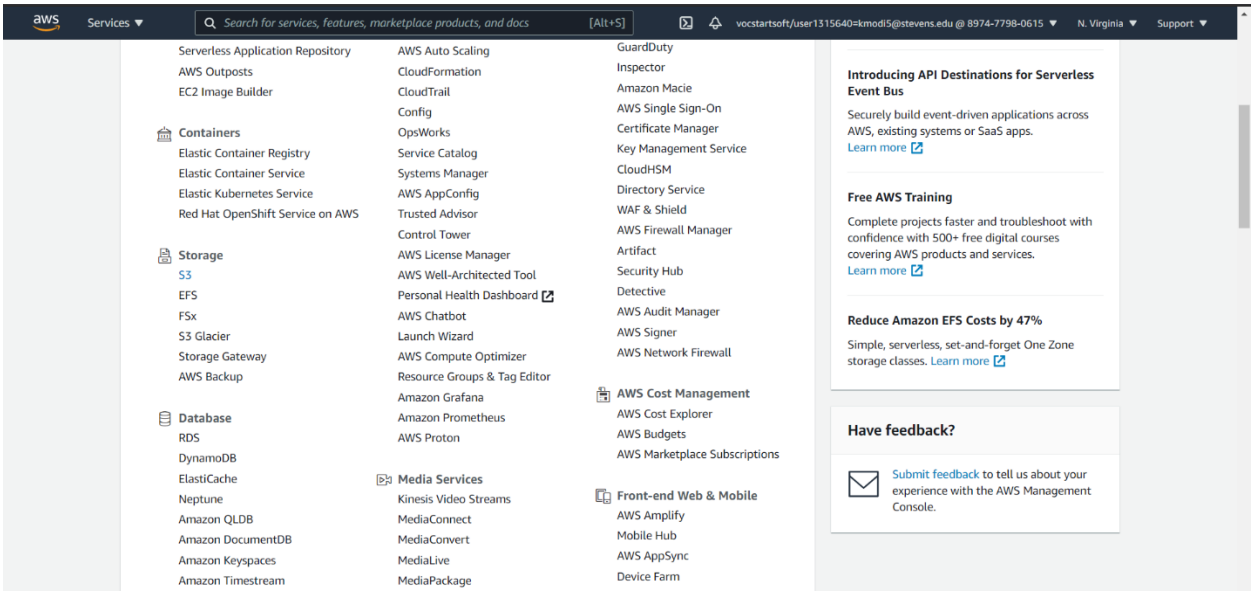
When you want to use CloudFront to distribute your content, you create a distribution and choose the configuration settings you want. For example:

- Your content origin—that is, the Amazon S3 bucket, MediaPackage channel, or HTTP server from which CloudFront gets the files to distribute. You can specify any combination of up to 25 Amazon S3 buckets, channels, and/or HTTP servers as your origins.
- Access—whether you want the files to be available to everyone or restrict access to some users.
- Security—whether you want CloudFront to require users to use HTTPS to access your content.
- Cache key—which values, if any, you want to include in the cache key. The cache key uniquely identifies each file in the cache for a given distribution.
- Origin request settings—whether you want CloudFront to include HTTP headers, cookies, or query strings in requests that it sends to your origin.
- Geo-restrictions—whether you want CloudFront to prevent users in selected countries from accessing your content.
- Access logs—whether you want CloudFront to create access logs that show viewer activity.

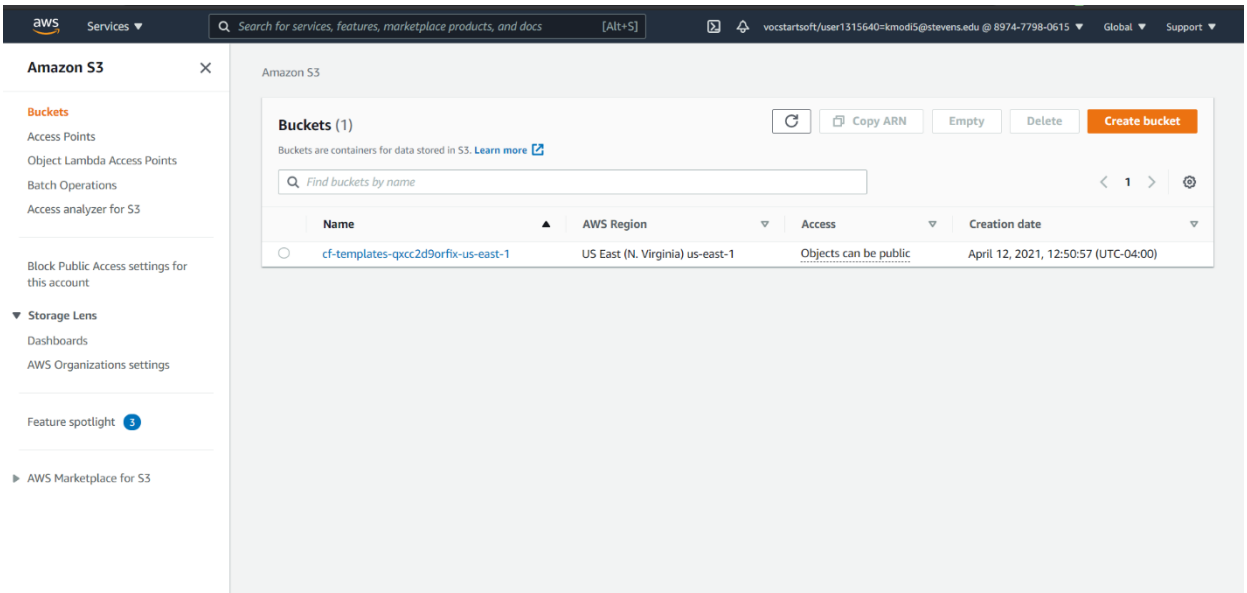
a. Once registered for AWS, now go into AWS Console



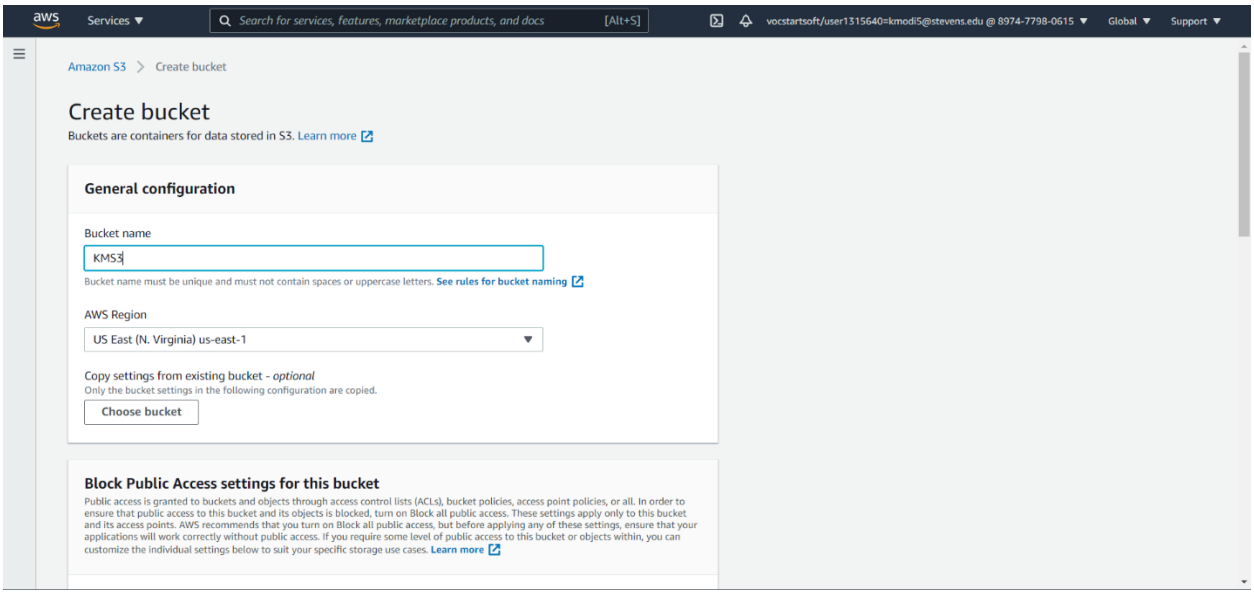
b. Go into AWS Console and Launch a S3 Bucket



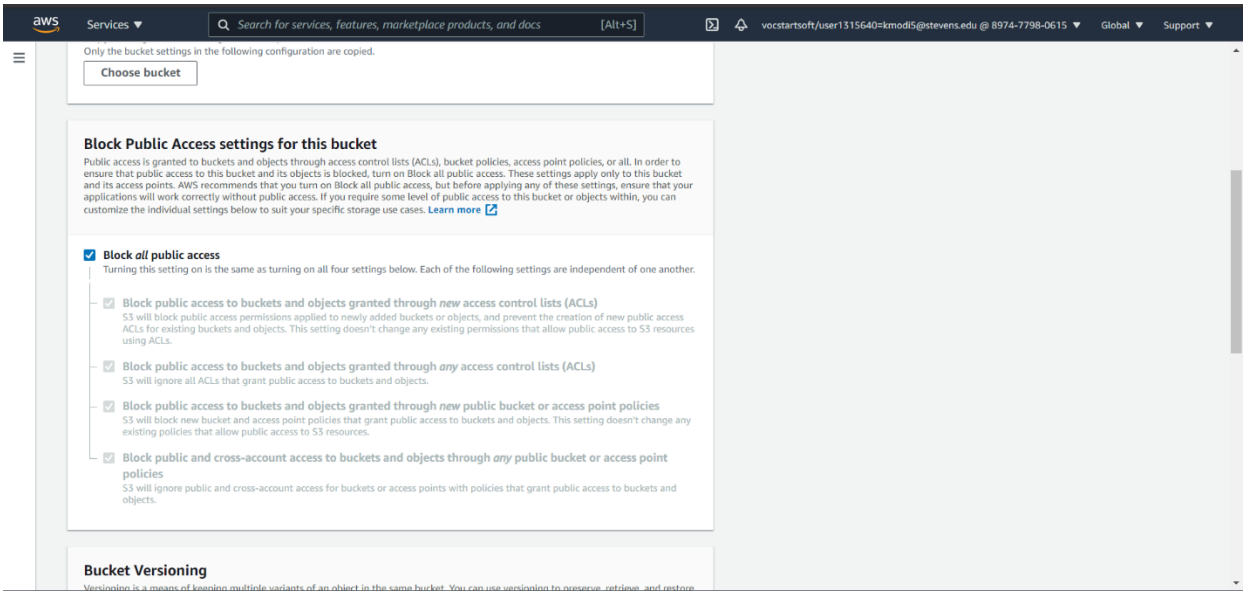
c. Create Bucket



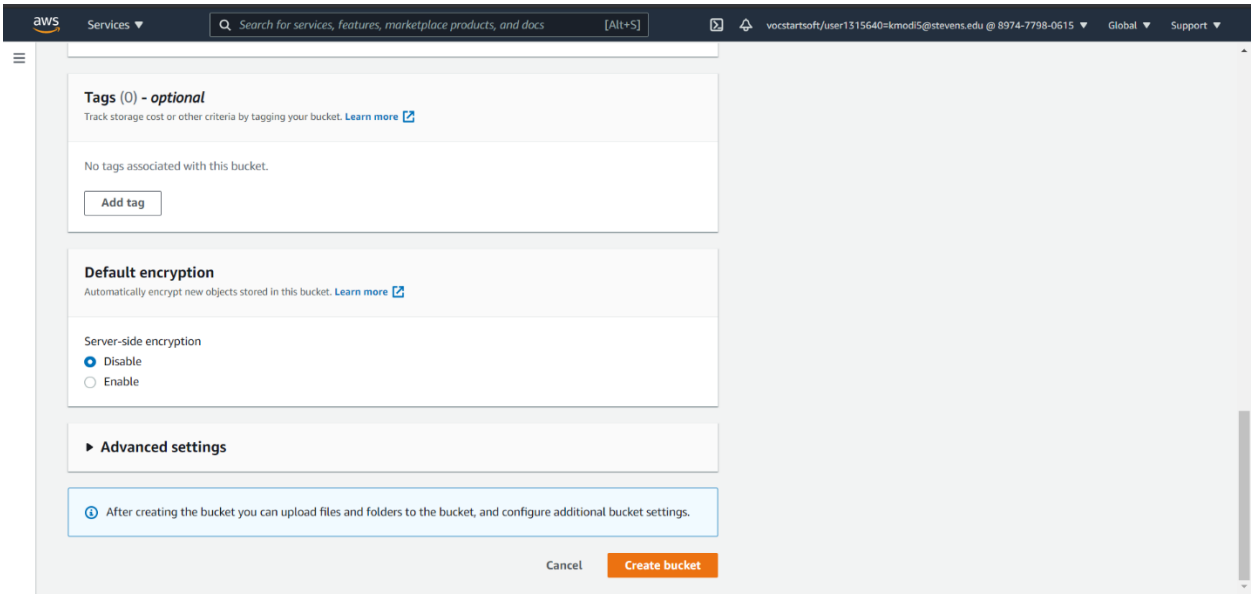
d. Fill and select the requirements



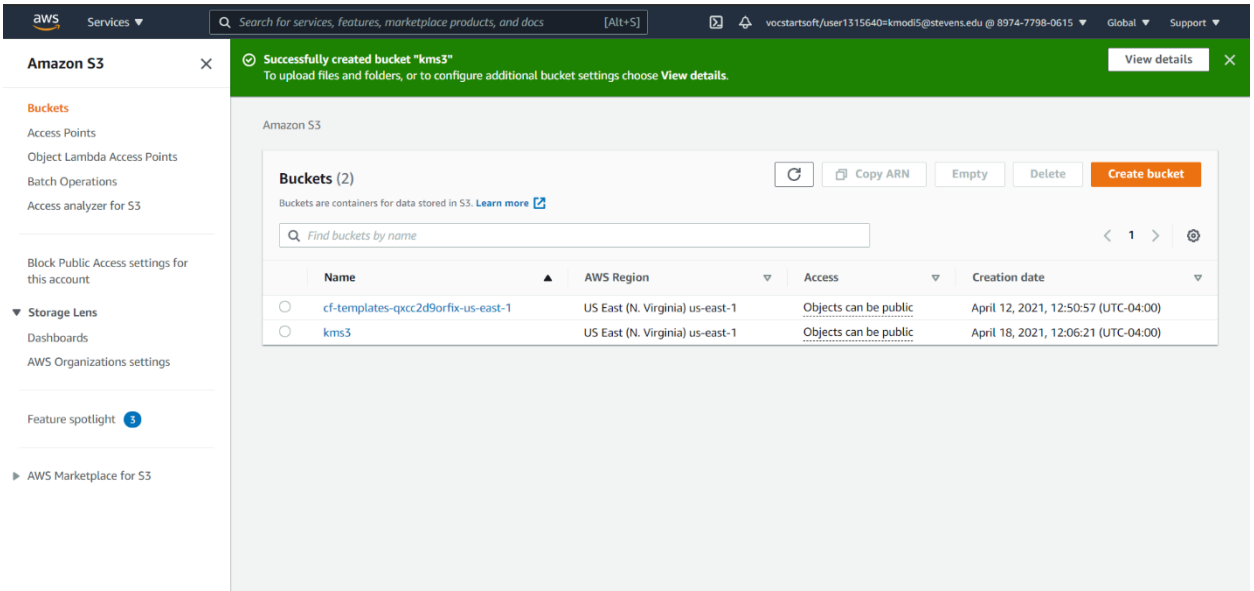
e. Unblock all public access



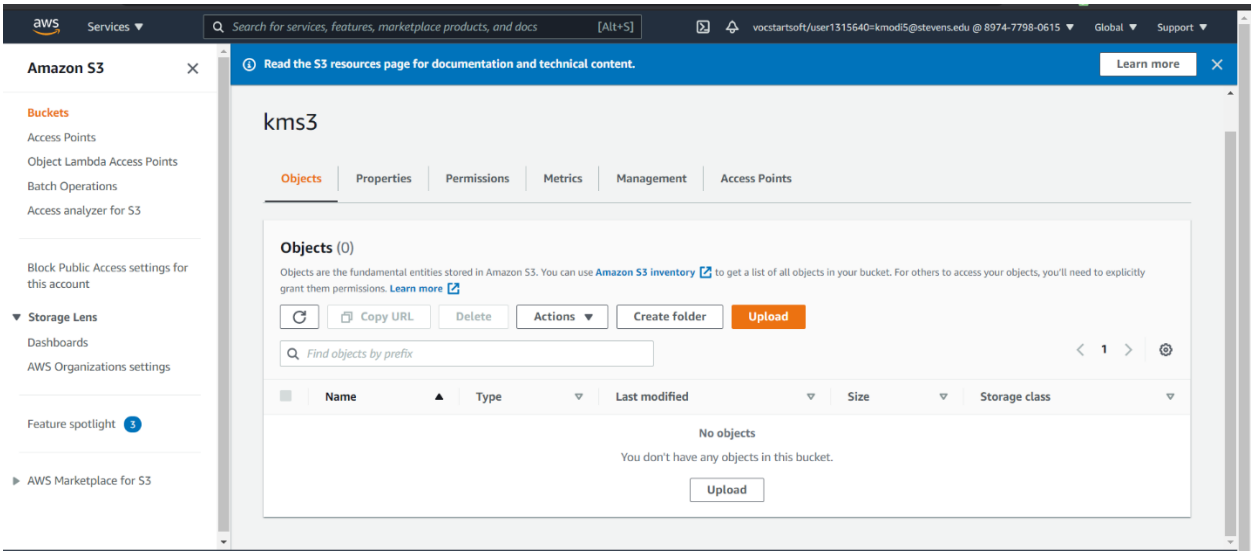
f. Create the bucket



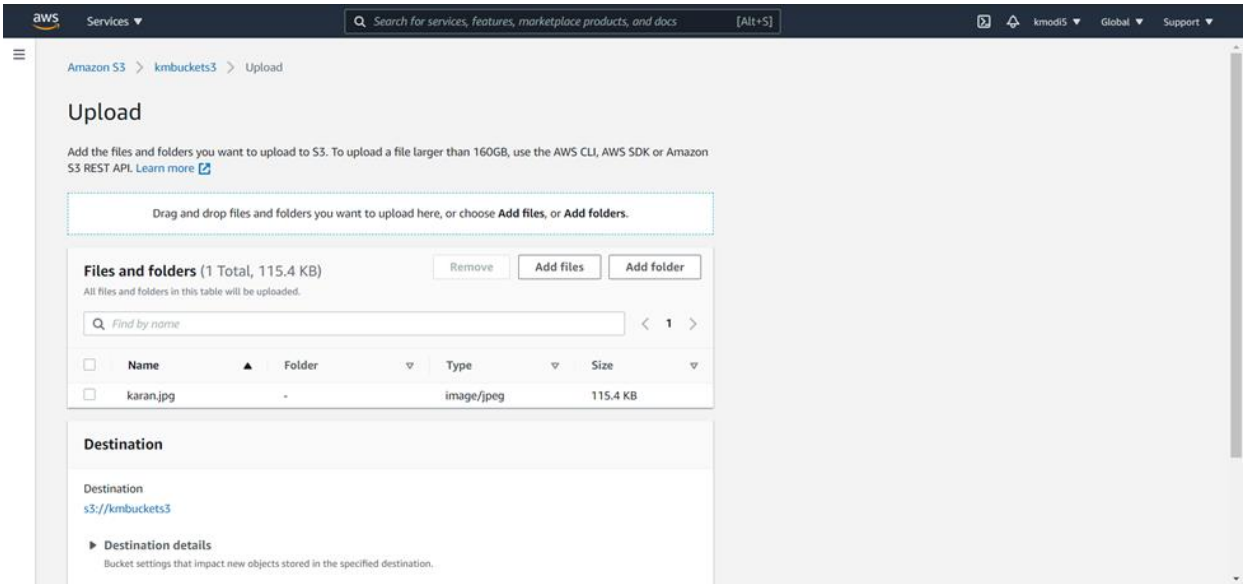
g. The bucket should be ready



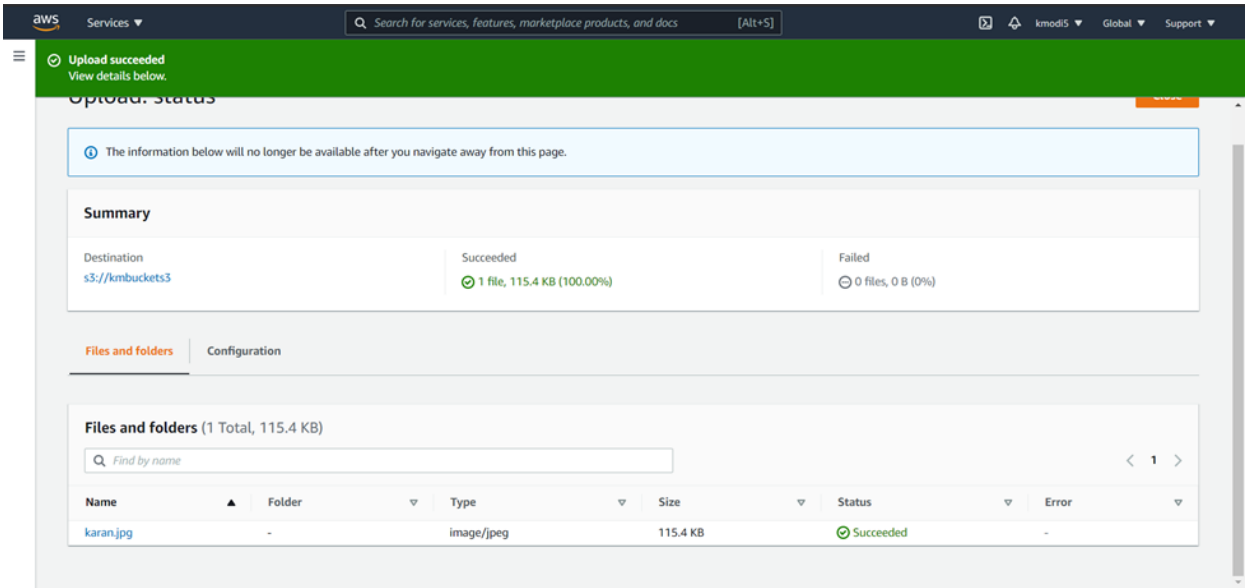
h. Upload an object



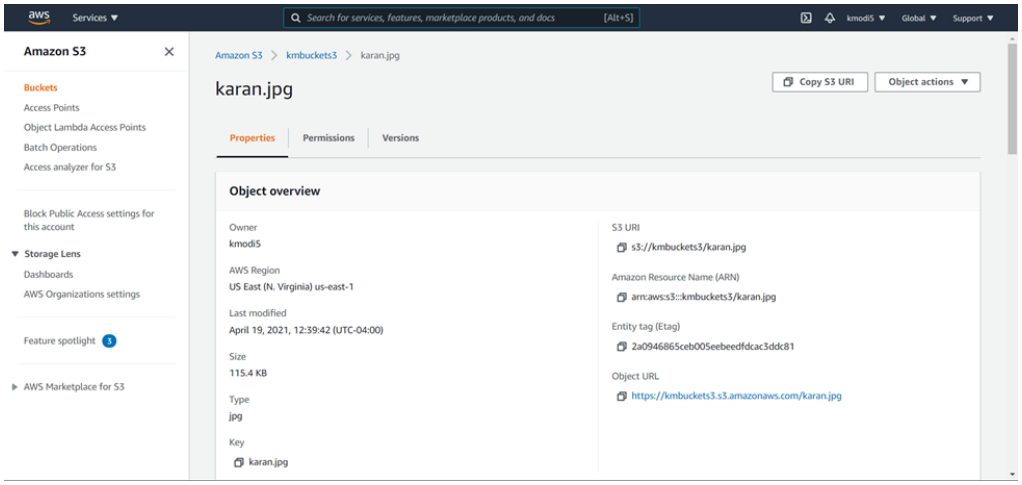
i. Upload a file in our case, an image

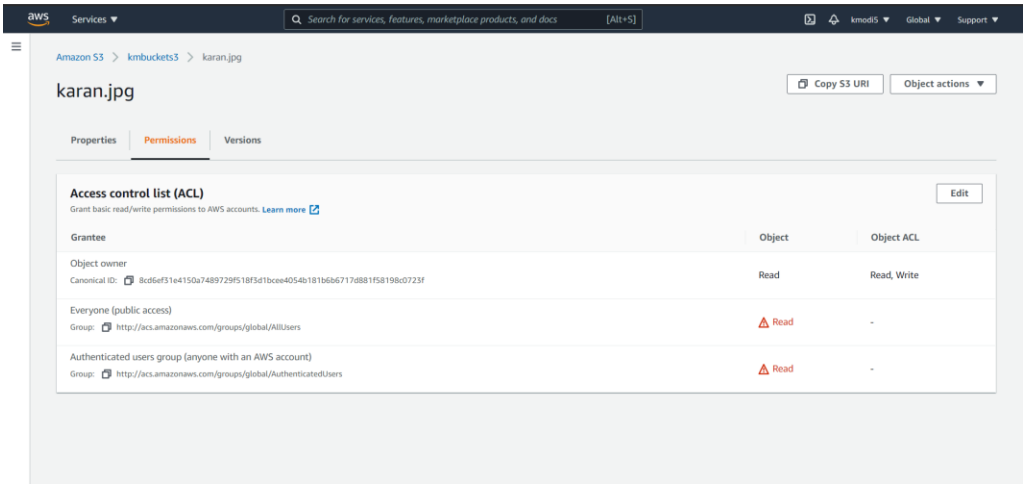


j. Check if it succeeded uploading the file

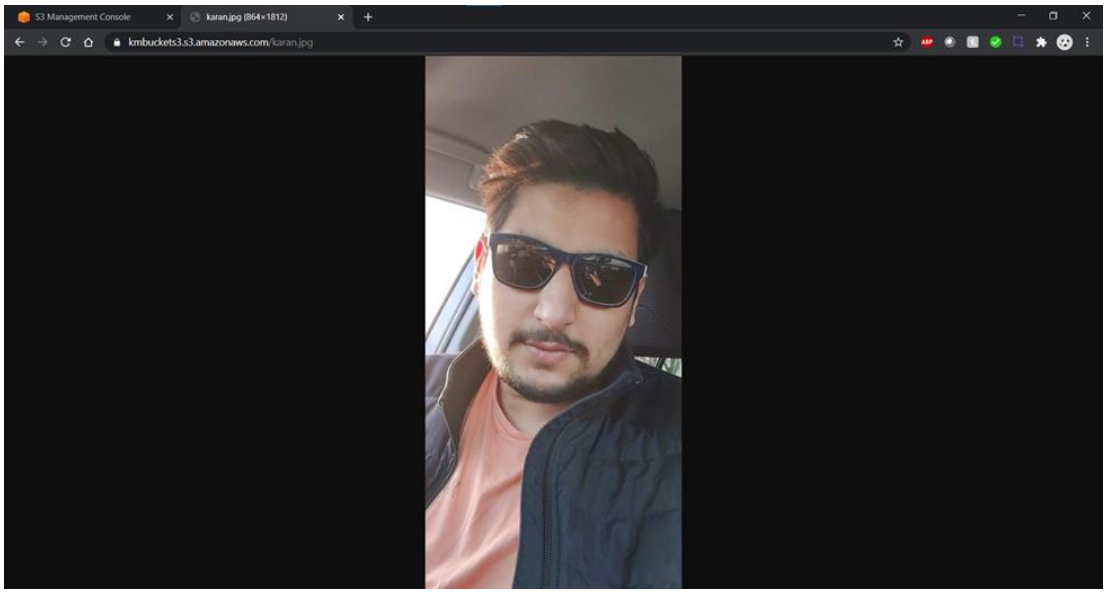


k. Give the read access for public(if required)

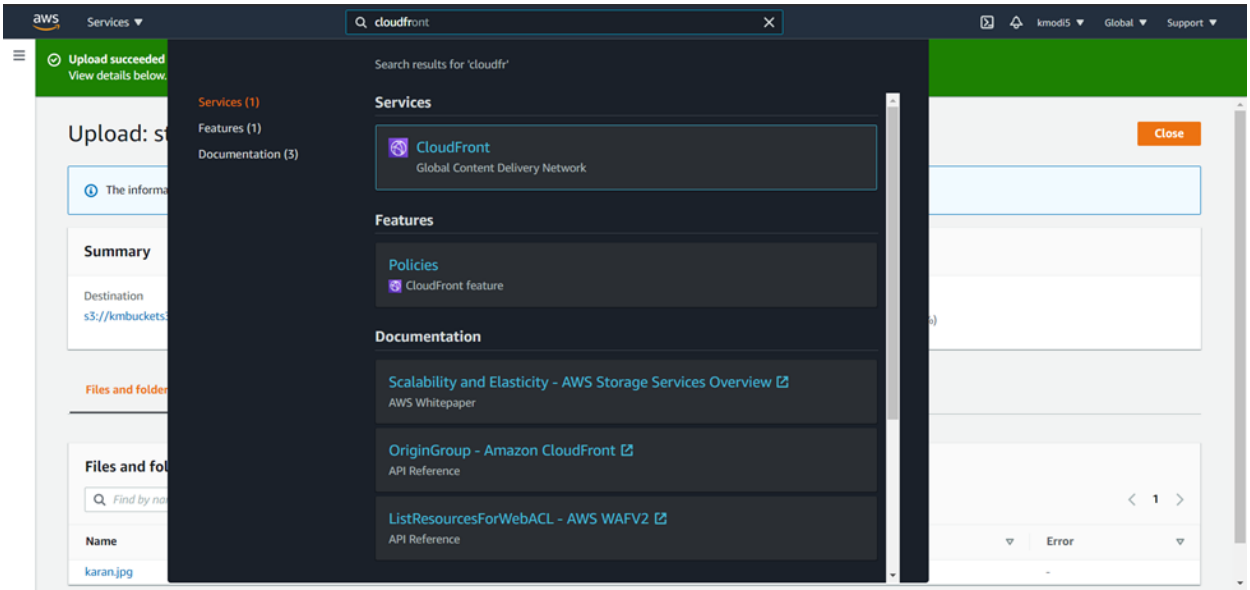




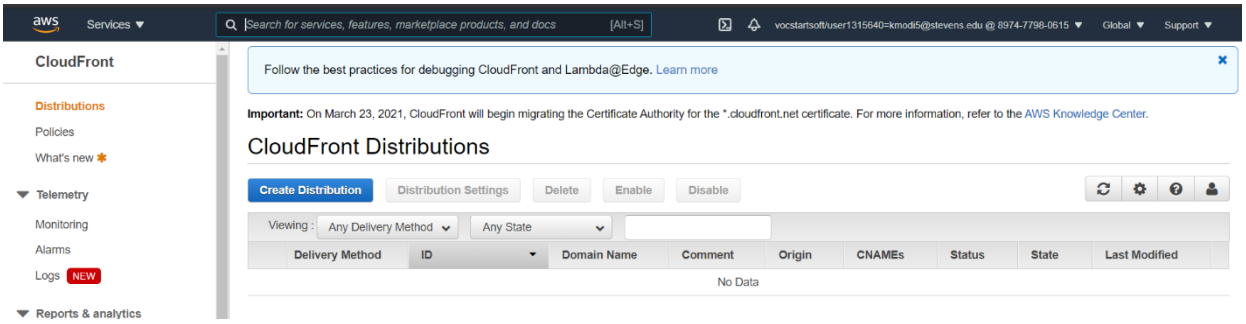
l. Access the object using the URL: https://kmbuckets3.s3.amazonaws.com/karan.jpg

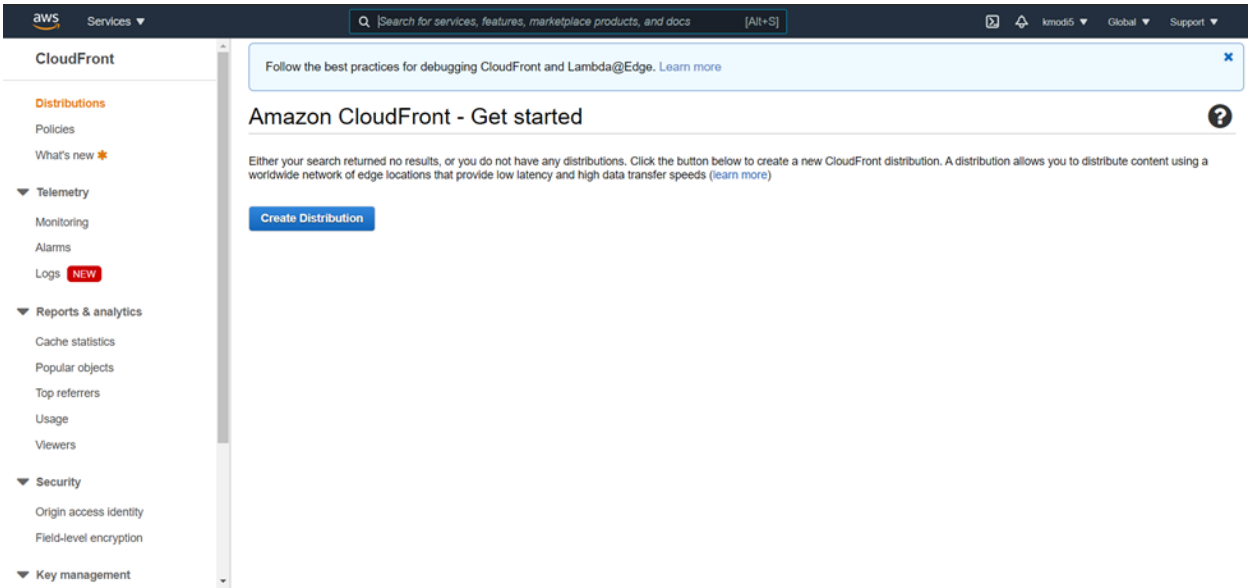


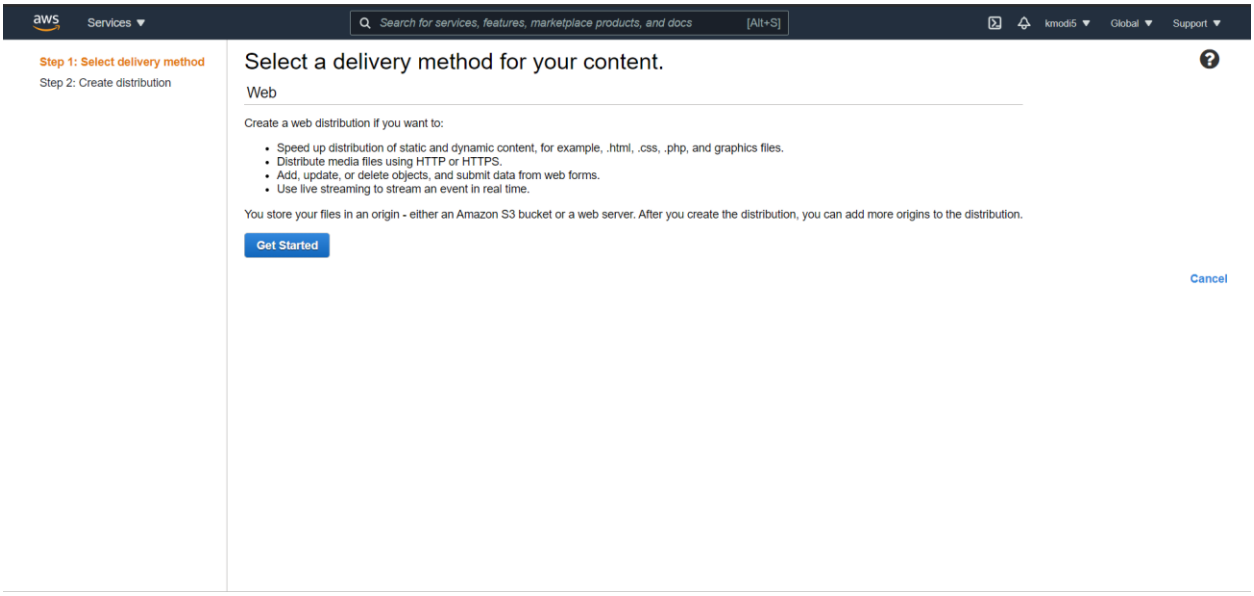
m. Launch the Cloudfront



n. Create a Distribution List







o. Select the Origin Domain Name, origin access identity and create the Distribution

aws Services ▼ Search for services, features, marketplace products, and docs [Alt+S] kmodi5 ▼ Global ▼ Support ▼

Step 1: Select delivery method
Step 2: Create distribution

Create Distribution

Origin Settings

Origin Domain Name	kmbuckets3.s3.amazonaws.com
Origin Path	/karan.jpg
Enable Origin Shield	○ Yes ● No
Origin ID	S3-kmbuckets3/karan.jpg
Restrict Bucket Access	● Yes ○ No
Origin Access Identity	● Create a New Identity ○ Use an Existing Identity
Comment	access-identity-kmbuckets3.s3.amazona
Grant Read Permissions on Bucket	○ Yes, Update Bucket Policy ● No, I Will Update Permissions
Origin Connection Attempts	3
Origin Connection Timeout	10
Origin Custom Headers	Header Name / Value

Default Cache Behavior Settings

aws Services ▼ Search for services, features, marketplace products, and docs [Alt+S] kmodi5 ▼ Global ▼ Support ▼

Step 1: Select delivery method
Step 2: Create distribution

Default Cache Behavior Settings

Path Pattern	Default (*)
Viewer Protocol Policy	○ HTTP and HTTPS ● Redirect HTTP to HTTPS ○ HTTPS Only
Allowed HTTP Methods	● GET, HEAD ○ GET, HEAD, OPTIONS ○ GET, HEAD, OPTIONS, PUT, POST, PATCH, DELETE
Field-level Encryption Config	
Cached HTTP Methods	GET, HEAD (Cached by default)
Cache and origin request settings	○ Use a cache policy and origin request policy ● Use legacy cache settings
Cache Based on Selected Request Headers	None (Improves Caching) Learn More
Object Caching	● Use Origin Cache Headers ○ Customize Learn More
Minimum TTL	0
Maximum TTL	31536000

aws Services ▼ Search for services, features, marketplace products, and docs [Alt+S] kmodi5 ▼ Global ▼ Support ▼

Step 1: Select delivery method
Step 2: Create distribution

Maximum TTL: 31536000

Default TTL: 86400

Forward Cookies: None (Improves Caching)

Query String Forwarding and Caching: None (Improves Caching)

Smooth Streaming: ○ Yes ◉ No

Restrict Viewer Access (Use Signed URLs or Signed Cookies): ○ Yes ◉ No

Compress Objects Automatically: ○ Yes ◉ No
Learn More

Lambda Function Associations

CloudFront Event	Lambda Function ARN	Include Body
Select Event Type		

Learn More

Enable Real-time Logs: ○ Yes ◉ No

Distribution Settings

Price Class: Use All Edge Locations (Best Performance)

aws Services ▼ Search for services, features, marketplace products, and docs [Alt+S] kmodi5 ▼ Global ▼ Support ▼

Step 1: Select delivery method
Step 2: Create distribution

Distribution Settings

Price Class: Use All Edge Locations (Best Performance)

AWS WAF Web ACL: None

Alternate Domain Names (CNAMEs):

SSL Certificate: ◉ Default CloudFront Certificate (*.cloudfront.net)

Choose this option if you want your users to use HTTPS or HTTP to access your content with the CloudFront domain name (such as https://d111111abcdef8.cloudfront.net/logo.jpg).
Important: If you choose this option, CloudFront requires that browsers or devices support TLSv1 or later to access your content.

○ Custom SSL Certificate (example.com):

Choose this option if you want your users to access your content by using an alternate domain name, such as https://www.example.com/logo.jpg.
You can use a certificate stored in AWS Certificate Manager (ACM) in the US East (N. Virginia) Region, or you can use a certificate stored in IAM.

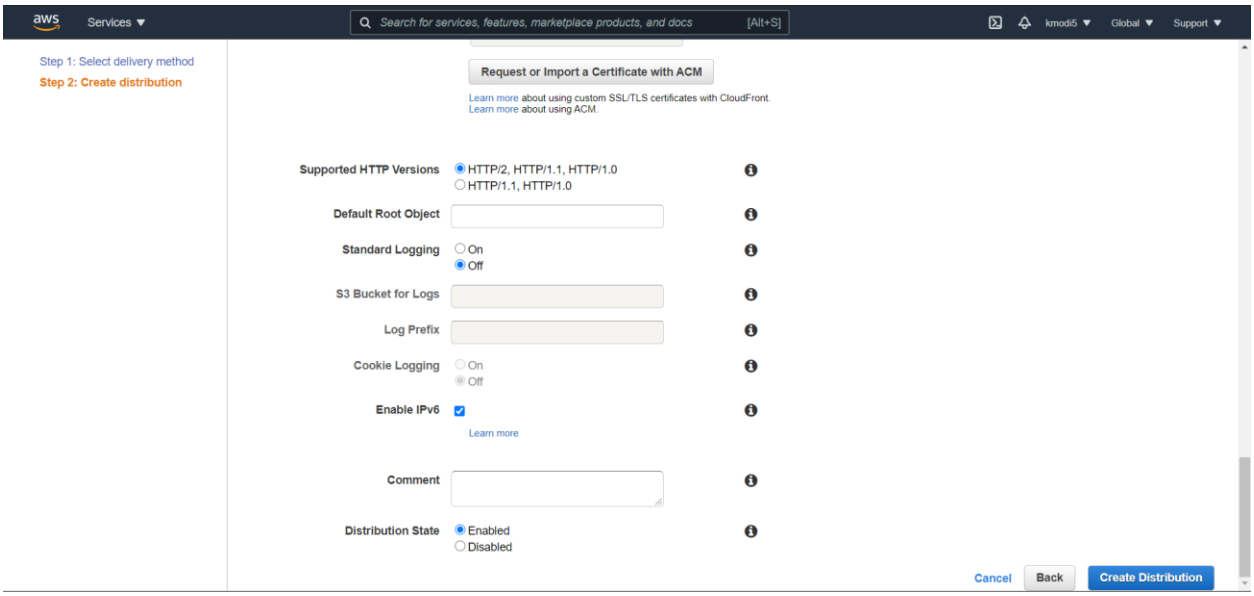
Request or Import a Certificate with ACM

Learn more about using custom SSL/TLS certificates with CloudFront.
Learn more about using ACM.

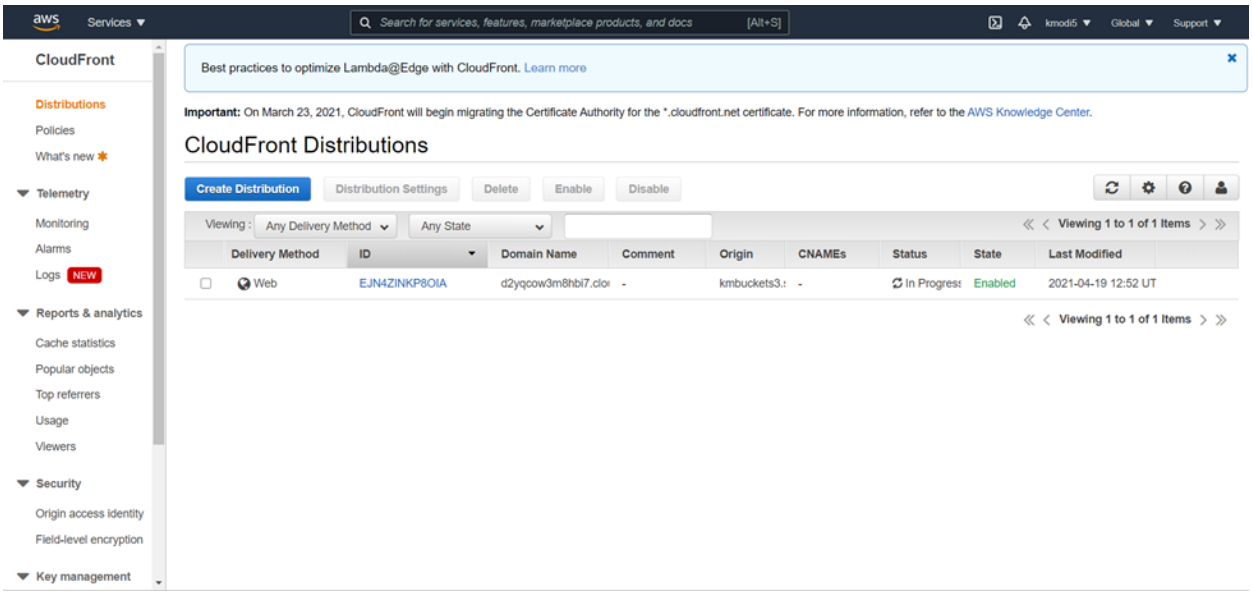
Supported HTTP Versions: ◉ HTTP/2, HTTP/1.1, HTTP/1.0 ○ HTTP/1.1, HTTP/1.0

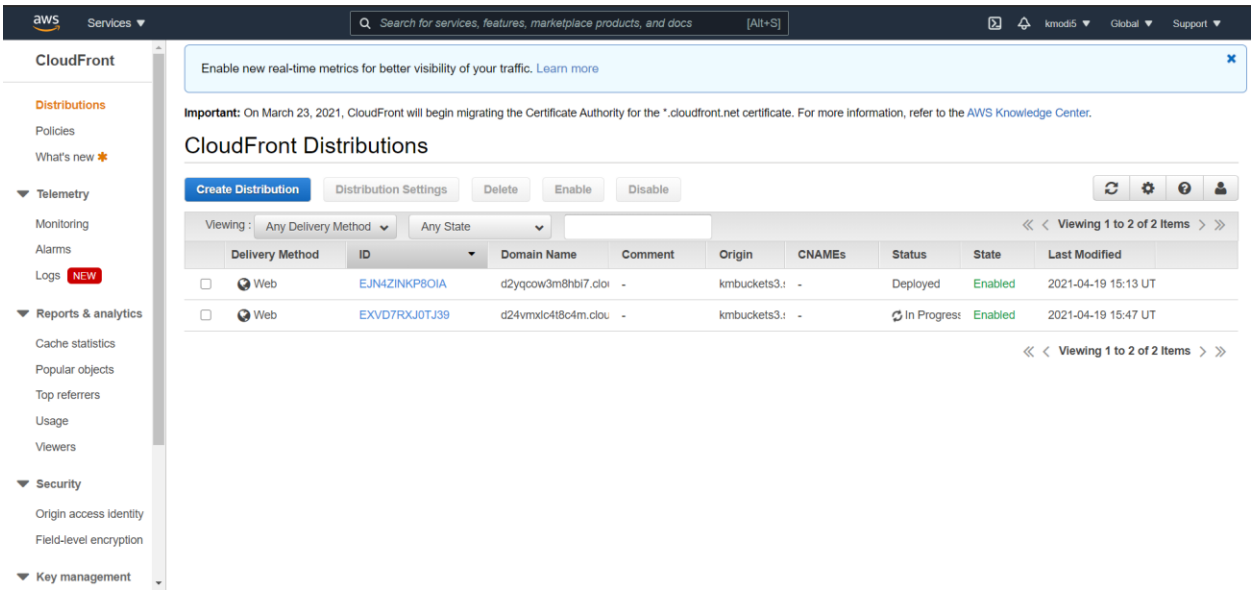
Default Root Object:

Standard Logging: ○ On

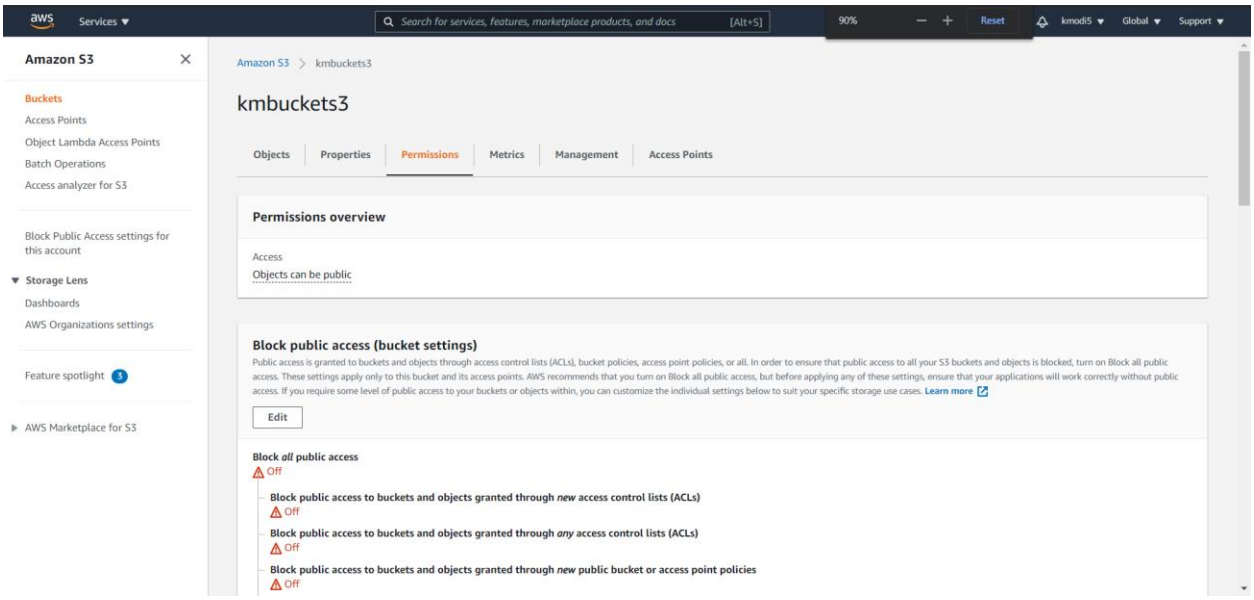


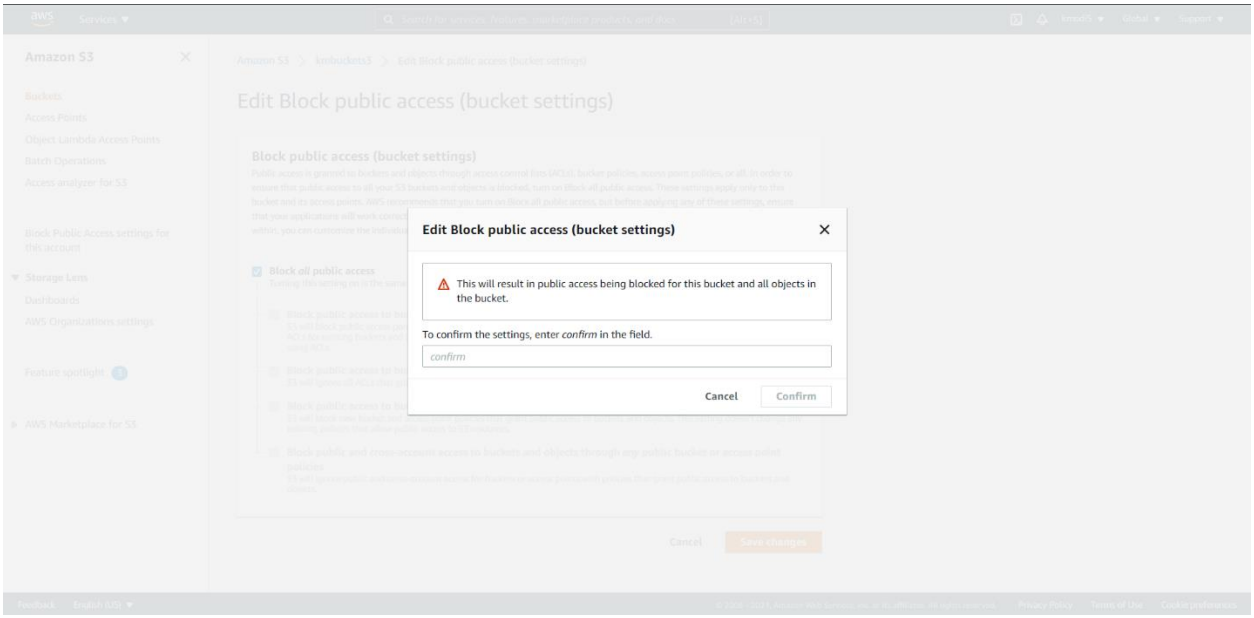
p. Now check if the cloudfront distribution is deployed





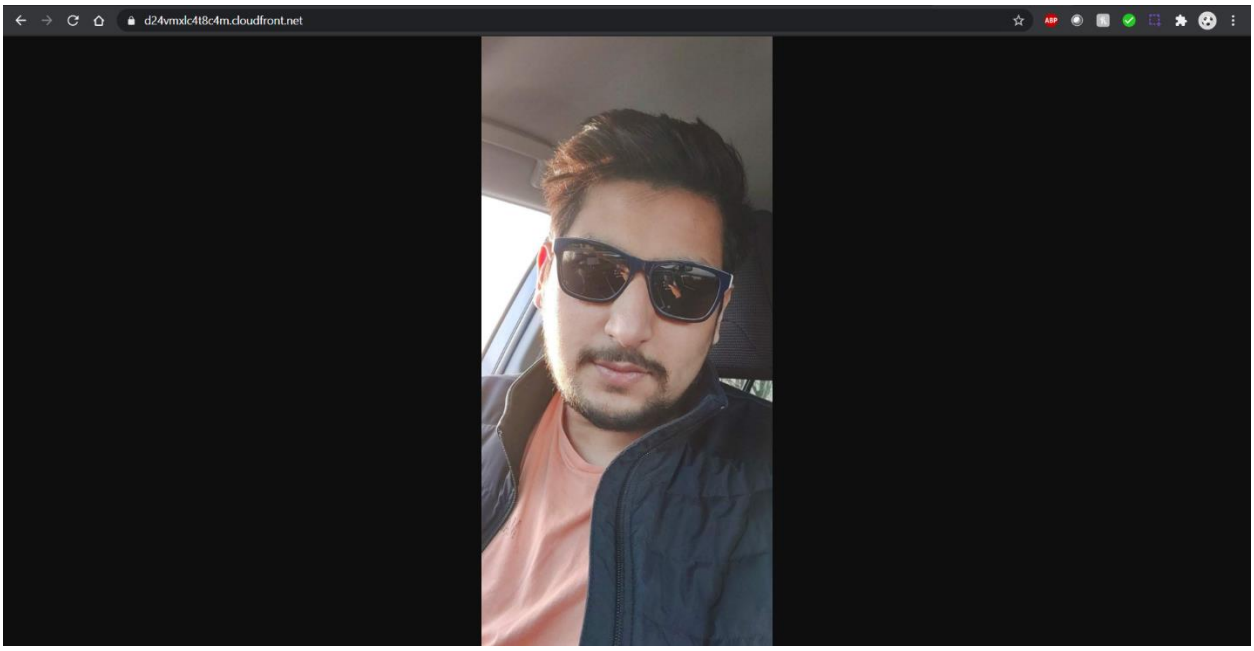
q. Disable public access to S3 bucket and allow only





r. Since we revoked public access the object can't be retrieved for public access

s. Cloud front URL: https://d24vmxlc4t8c4m.cloudfront.net/
Using cloudfront to access the bucket object which is now saved in different locations for best performance (like cache in very close to the CPU!)



We can conclude that CDN allows fast, efficient, durable, and reliable access to our files from any geographical location as seen with the above example compared to our S3 bucket object.

A content delivery network (CDN) refers to a geographically distributed group of servers which work together to provide fast delivery of Internet content.

A CDN allows for the quick transfer of assets needed for loading Internet content including HTML pages, javascript files, stylesheets, images, and videos. The popularity of CDN services continues to grow, and today the majority of web traffic is served through CDNs, including traffic from major sites like Facebook, Netflix, and Amazon.

We can see various benefits from CDN and according to my observations it would be:

a. Quick access at all times(low latency)

b. Reliable storage (distributed copies at different locations)
c. Guaranteed Service at different Geographical Locations (Asia, Americas. Europe etc)
d. Durable file service that the file isn't lost/corrupted (redundancy)'
e. Would provide better bandwidth and with it security due to the distributed nature(DDOS)
f. We can use this information at different locations gauge demand and for analytics
g.