Deploying an Online Movie Watching Application on Cloud

Description

You are working in an online entertainment provider company. As you have knowledge of cloud computing, you are asked **to deploy the company's website on cloud**.

Background of the problem statement:

You work for Binge Watch Online, an online entertainment provider company.

You have created a website for the company and used a public **cloud to deploy** the website. After deploying it on cloud, users are complaining about the reloading speed of the pages. The website is **getting global traffic** and **static** assets like pages that are **served from a single server**. You need to make sure that the traffic coming to the website from different parts of the world **is load balanced at the DNS level.**

You can use either Azure or AWS platforms to design the solution using IaaS OR PaaS or **SaaS.**

You must use the following tools:

- **AWS: Route 53, S3 Bucket, CloudFront, EC2**

You have been asked to:

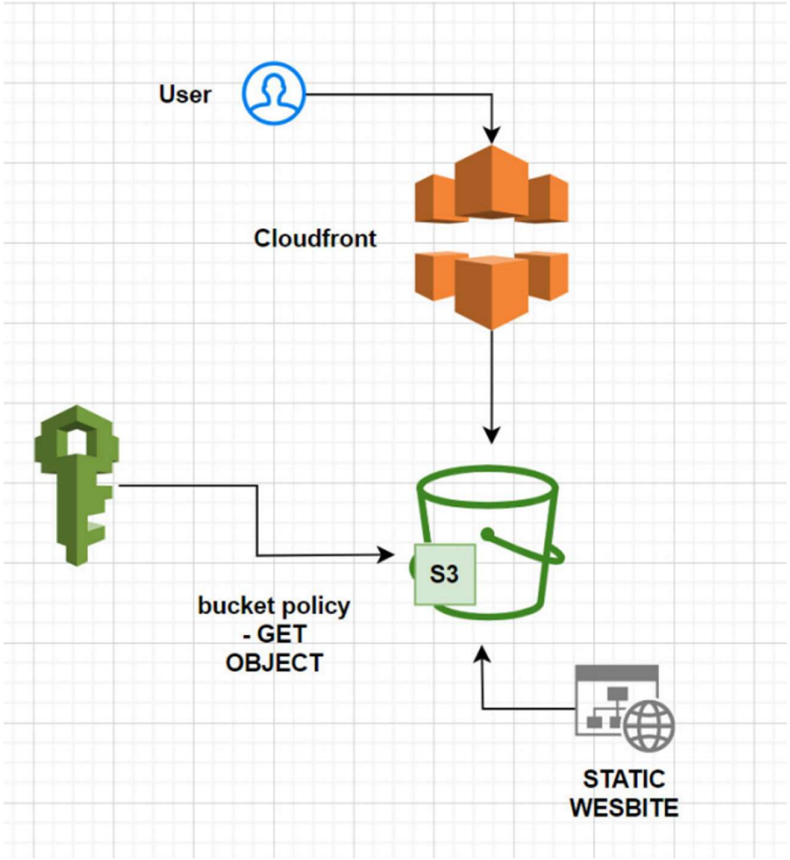
1. Suggest an appropriate solution so that your company can make use of the cloud while keeping the requirements mentioned above for your company in mind.
2. Provide an approach to:

   a. Govern all the resources being used for development, testing, and production of the company's website.

   b. Keep a separate track of the billing life cycle and cost management of all the services being used for hosting the company's website on Cloud.
3. Upload all static content of your web site to cloud.
4. Create a CDN endpoint and configure it to serve the static files you have uploaded.
5. Use storage service and upload files for your teammates to share.
6. Connect a Windows or Linux VM to the Storage service.

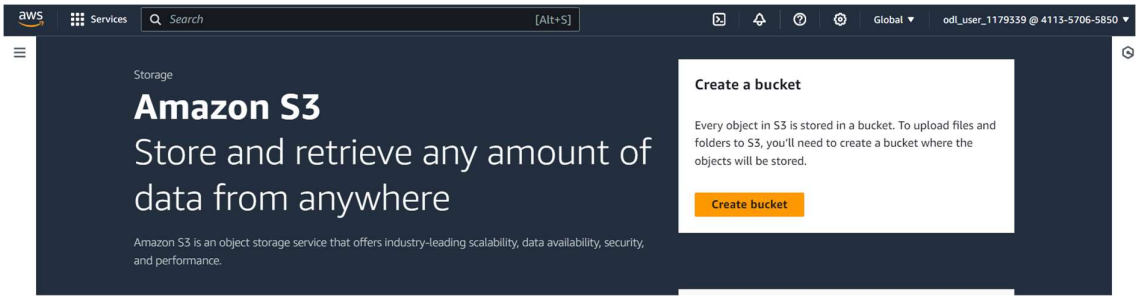
Solution:

Entertainment IN SAAS Deployment (W/o using Infrastructure provisioning):



STEPS:

1. create **S3 bucket** and **upload** entertainment app files to the bucket
2. enable **static website hosting** in the bucket
3. unblock public access for the bucket and add **get bucket policy - *** to the bucket
4. make the website is accessible using static DNS of s3 bucket



Choose region and give bucket name and leave everything default!

Amazon S3 > Buckets > Create bucket

Create bucket Info

Buckets are containers for data stored in S3. Learn more

General configuration

AWS Region

US East (N. Virginia) us-east-1

Bucket type Info

- General purpose
  Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability Zones.
- Directory - *New*
  Recommended for low-latency use cases. These buckets use only the S3 Express One Zone storage class, which provides faster processing of data within a single Availability Zone.

Bucket name Info

vd-entainment-capstone1

Bucket name must be unique within the global namespace and follow the bucket naming rules. See rules for bucket naming

Copy settings from existing bucket - *optional*

Only the bucket settings in the following configuration are copied.

Choose bucket

Upload files which you have

Amazon S3 > Buckets > vd-entainment-capstone1 > Upload

Upload Info

Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. Learn more

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

Files and folders (2 Total, 2.5 KB)

Remove | Add files | Add folder

All files and folders in this table will be uploaded.

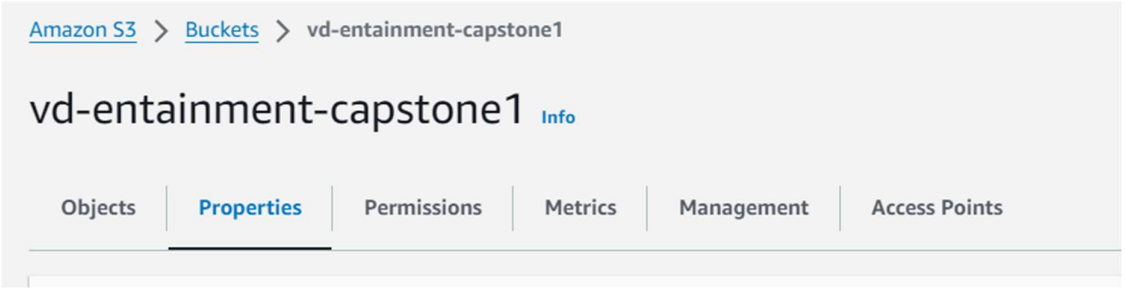
Find by name

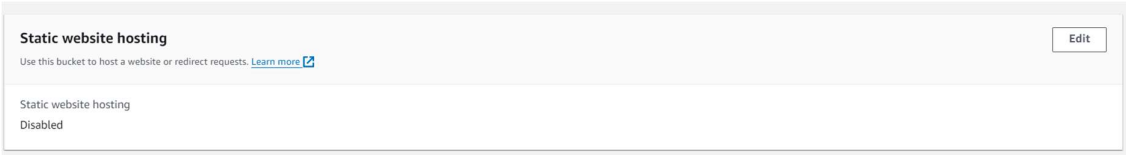
1

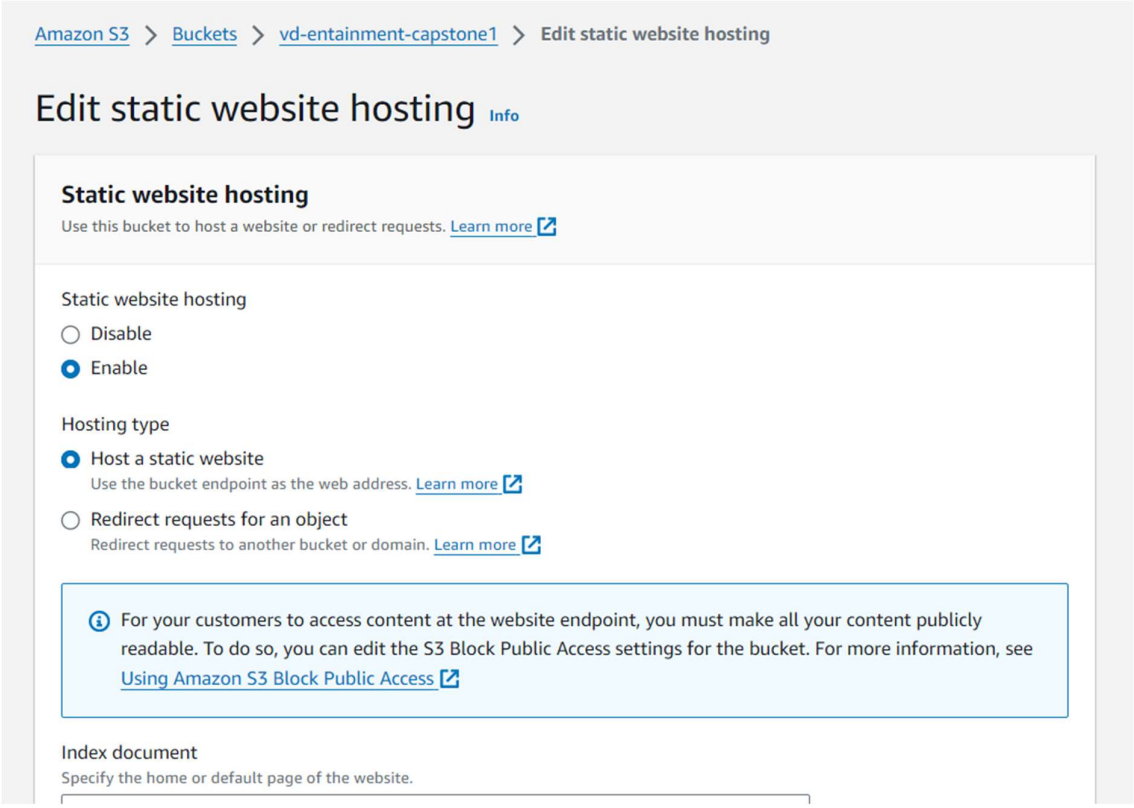
Name	Folder	Type	Size
index.html	-	text/html	2.2 KB
thankyou.html	-	text/html	293.0 B

Click upload!

Enable static web site hosting in the properties of the bucket section!







Give document index.html name which you uploaded!

Index document

Specify the home or default page of the website.

index.html

Error document - *optional*

This is returned when an error occurs.

error.html

Redirection rules – *optional*

Redirection rules, written in JSON, automatically redirect webpage requests for specific content. Learn more

1

And save changes.

And now try to access the endpoint of static web site hosting, it will give 403 forbidden error.

Static website hosting

Use this bucket to host a website or redirect requests. Learn more

Static website hosting

Enabled

Hosting type

Bucket hosting

Bucket website endpoint

When you configure your bucket as a static website, the website is available at the AWS Region-specific website endpoint of the bucket. Learn more

http://vd-entainment-capstone1.s3-website-us-east-1.amazonaws.com

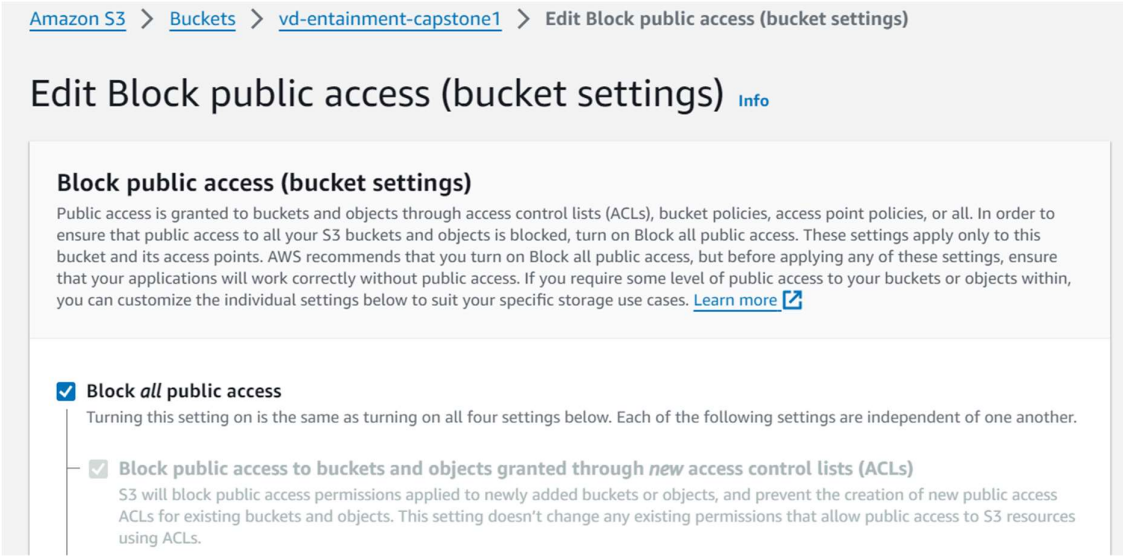
Not secure vd-entainment-capstone1.s3-website-us-east-1.amazonaws.com

Keywords for Resu...

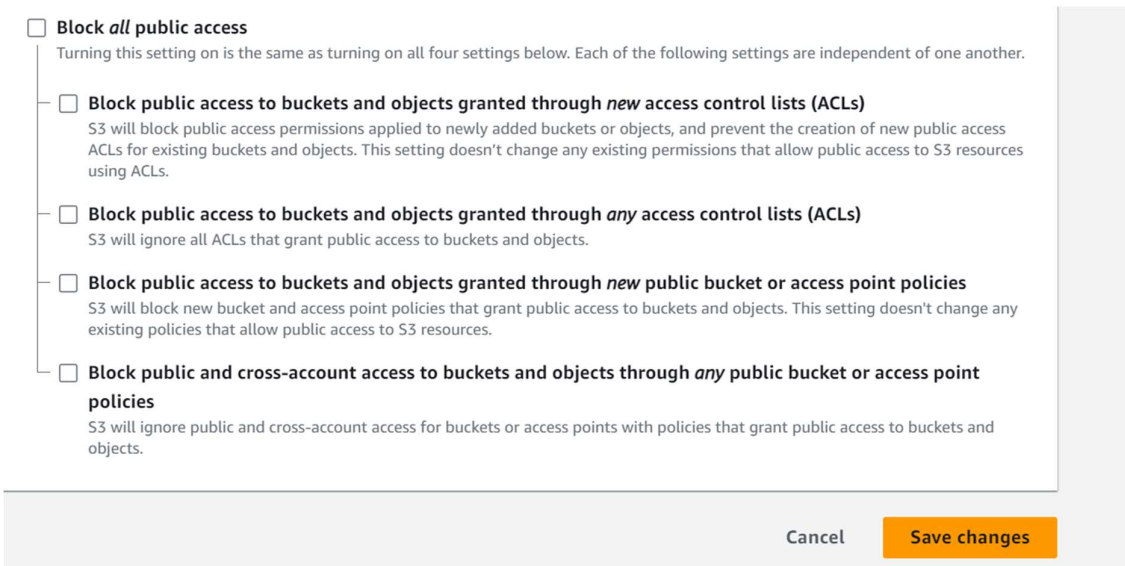
403 Forbidden

- Code: AccessDenied
- Message: Access Denied
- RequestId: 57HHKYJ3CEK96PFS
- HostId: 4hOYcIjISbQUkEf7wxti+3rOdatIqKnfrnQSE6wEacoC1+M3D4ZpoumsPe/Jj9dwnq8u49wKJU8=

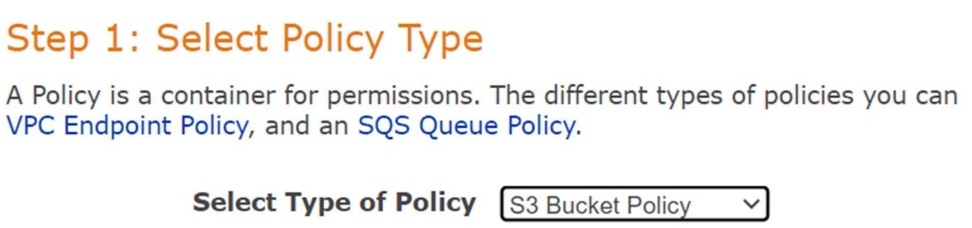
The reason is we haven't enabled the public access for the bucket,



so now enable the public access and add a bucket policy to access the content of files.



click on bucket edit policy and go for policy generator.



Amazon Resource Name: arn:aws:s3:::vd-entainment-capstone1

In policy after the ARN, put**" /*"**

action: Get Object

add statement and click generate policy.

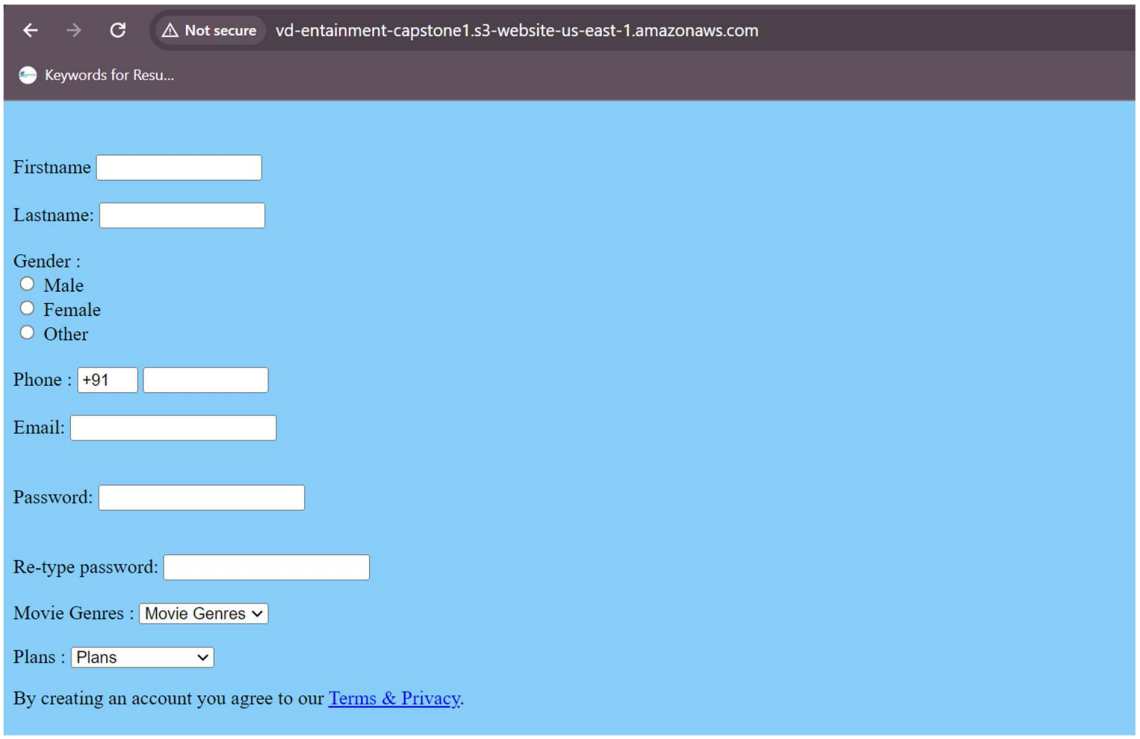
Policy JSON Document

Click below to edit. To save the policy, copy the text below to a text editor.
Changes made below will **not be reflected in the policy generator tool.**

```
{
  "Id": "Policy1703229669565",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1703229641014",
      "Action": [
        "s3:GetObject"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::vd-entainment-capstone1/*",
      "Principal": "*"
    }
  ]
}
```

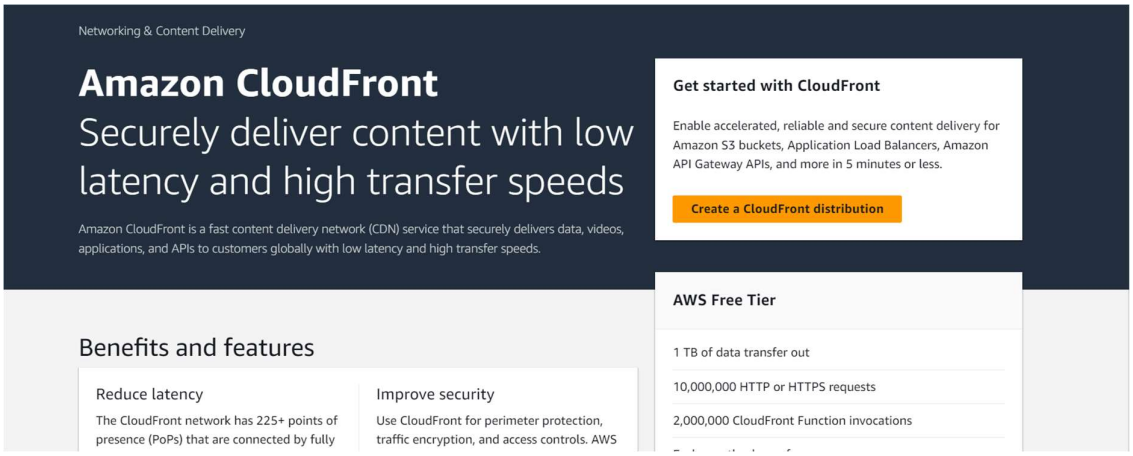

```
{
  "Id": "Policy1703229669565",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1703229641014",
      "Action": [
        "s3:GetObject"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::vd-entainment-capstone1/*",
      "Principal": "*"
    }
  ]
}
```

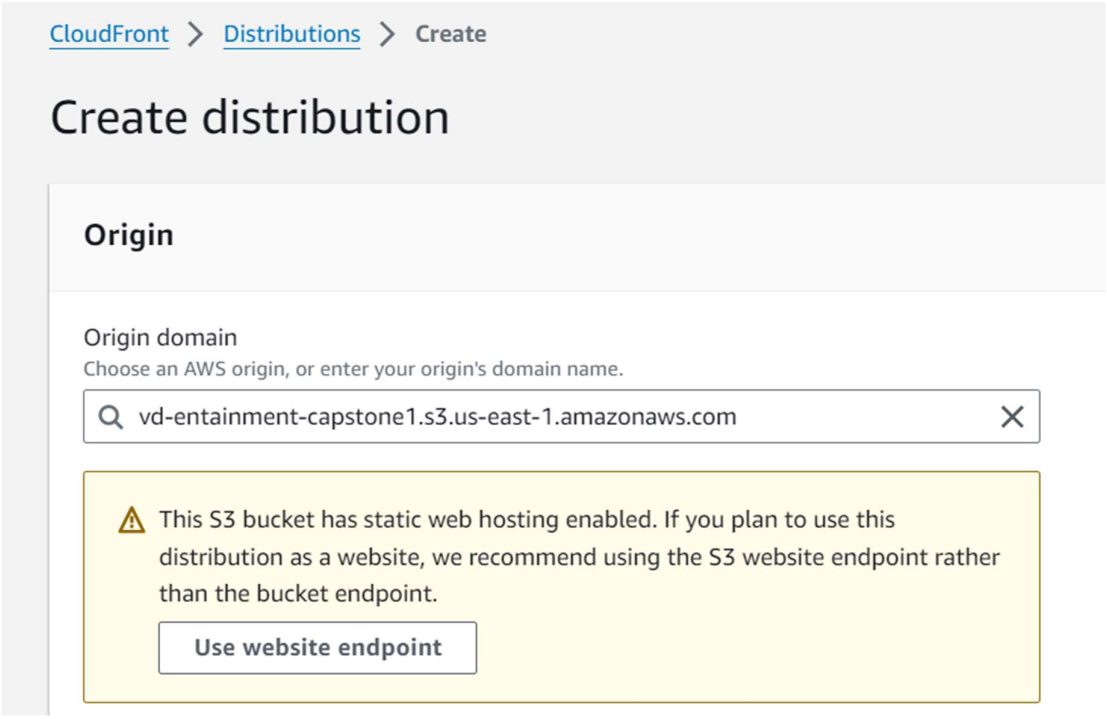

Paste the policy in the bucket edit and save changes, now if you try to access your application would be available.



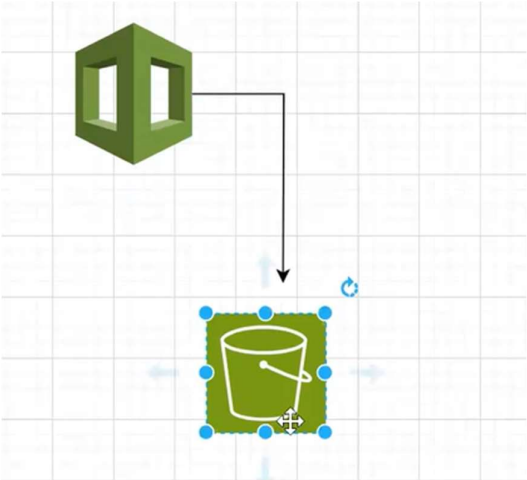
I made sure now my application is highly available, and I need to take care of single server setup. I take care of availability, the one part which I am missing is load balancing at the DNS level.

For global traffic should be handled at the DNS level, for that I need to create CloudFront.





Give your OAI, is like plugin that sit between CloudFront and s3 bucket. And it takes care of the authentication. To access s3 from CloudFront, OAI will be sitting in front of s3.



Enter a name for this origin.

vd-entainment-capstone1.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

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

demo-202312101910-mk1.s3.us-east-1.amazonaws.com Origin type: S3
temp

Create control setting

Bucket policy

Policy must allow access to CloudFront IAM service principal role.

◉ I will manually update the policy

⚠ **You must update the S3 bucket policy**
CloudFront will provide you with the policy statement after creating the distribution.

Since in our requirement is across the globe, we will select all edge locations and default file name id index.html. Create the distribution (leave other by default)

Settings

Price class | Info

Choose the price class associated with the maximum price that you want to pay.

- ◉ Use all edge locations (best performance)
- ○ Use only North America and Europe
- ○ Use North America, Europe, Asia, Middle East, and Africa

Alternate domain name (CNAME) - *optional*

Add the custom domain names that you use in URLs for the files served by this distribution.

Add item

ⓘ To add a list of alternative domain names, use the bulk editor.

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.

[ ] Use monitor mode
Count how many of your requests would be blocked by this WAF configuration. When ready, you can disable monitor mode to begin blocking requests.

▼ **Included security protections**

- Protect against the most common vulnerabilities found in web applications.
- Protect against malicious actors discovering application vulnerabilities.
- Block IP addresses from potential threats based on Amazon internal threat intelligence

Price estimate

▶ **This AWS WAF configuration is estimated to cost $14 for 10 million requests/month**

Supported HTTP versions

Add support for additional HTTP versions. HTTP/1.0 and HTTP/1.1 are supported by default.

- [x] HTTP/2
- [ ] HTTP/3

Default root object - *optional*

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

index.html

Standard logging

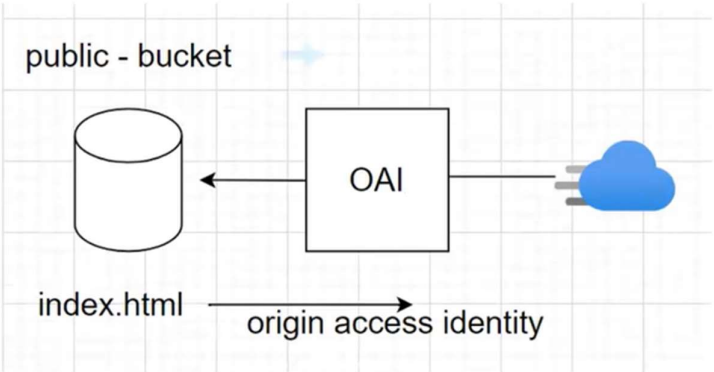
Get logs of viewer requests delivered to an Amazon S3 bucket.

- (●) Off
- ( ) On

IPv6

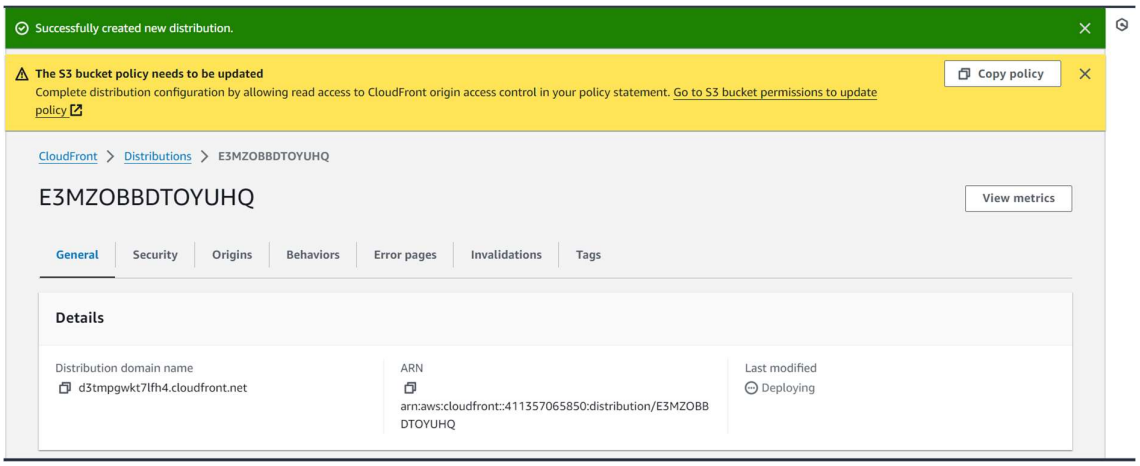
- ( ) Off
- (●) On

You also need to select ACM certificate and WAF. (currently not needed)



We got our DNS; the deployment is happening in the backened.in few minutes we will be able to access the endpoint.

If you observe the s3 policy new generated policy by cloud front will need to be added in s3 bucket→permissions; by removing the older one, as we(user) want to only from CloudFront bucket needs to be accessed with s3 bucket , not by public access of the s3 bucket any more.

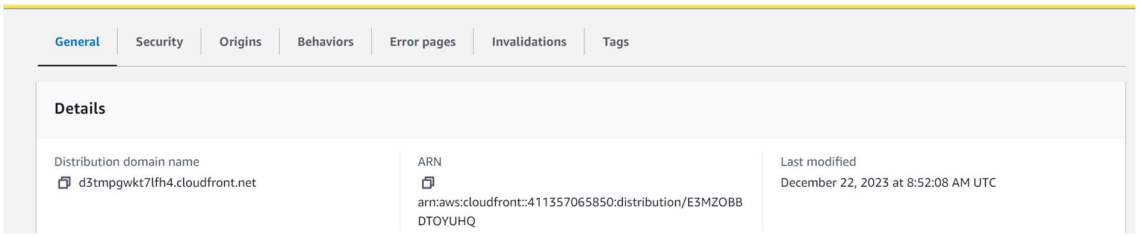


Click save changes!

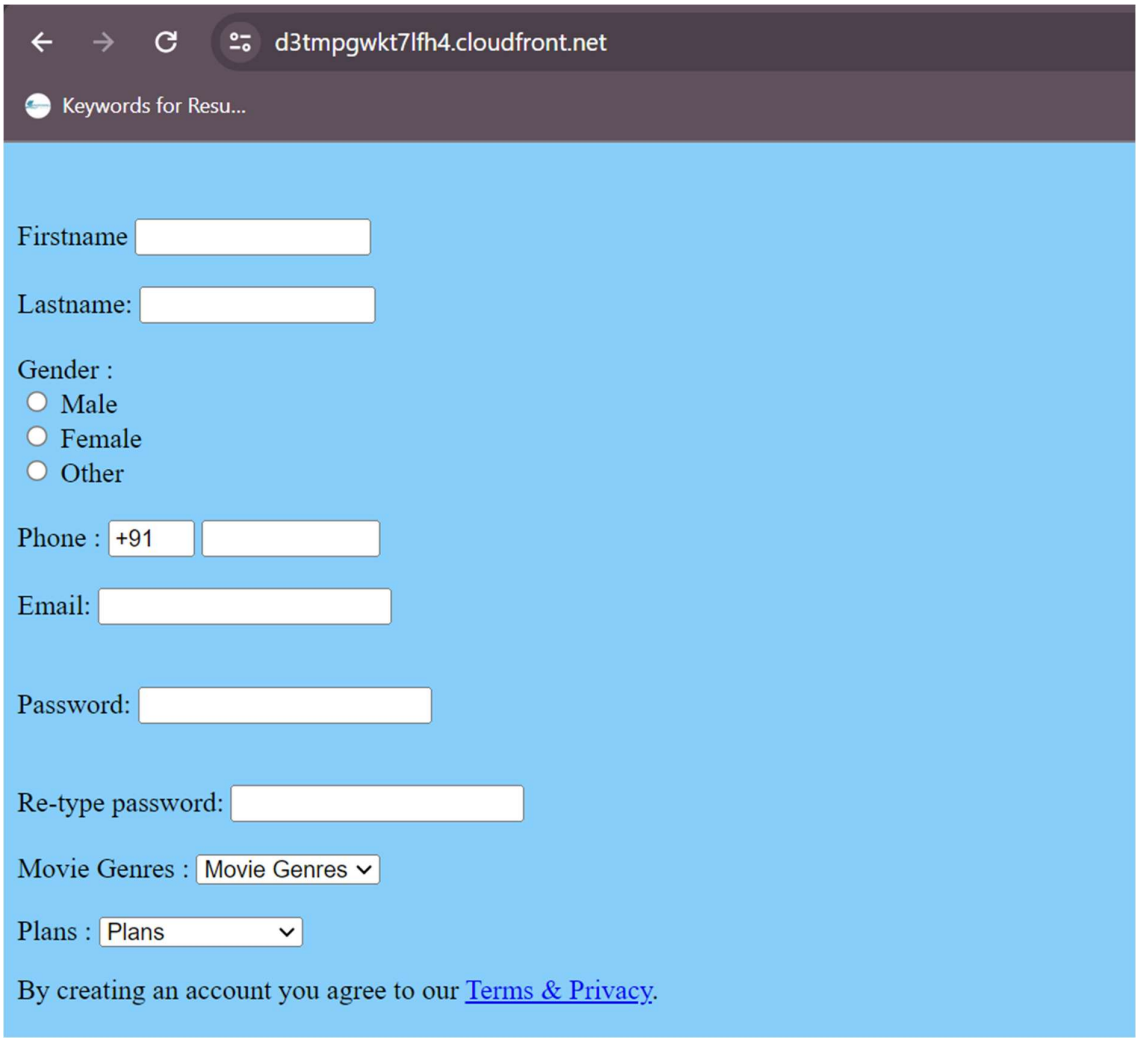

```
{
    "Version": "2008-10-17",
    "Id": "PolicyForCloudFrontPrivateContent",
    "Statement": [
      {
        "Sid": "AllowCloudFrontServicePrincipal",
        "Effect": "Allow",
        "Principal": {
          "Service": "cloudfront.amazonaws.com"
        },
        "Action": "s3:GetObject",
        "Resource": "arn:aws:s3:::vd-entainment-capstone1/*",
        "Condition": {
          "StringEquals": {
            "AWS:SourceArn":
"arn:aws:cloudfront::411357065850:distribution/E3MZOBBDTOYUHQ"
          }
        }
```

```
        }
    ]
}
```

Now copy the cloudfront distribution domain name and paste it in the browser

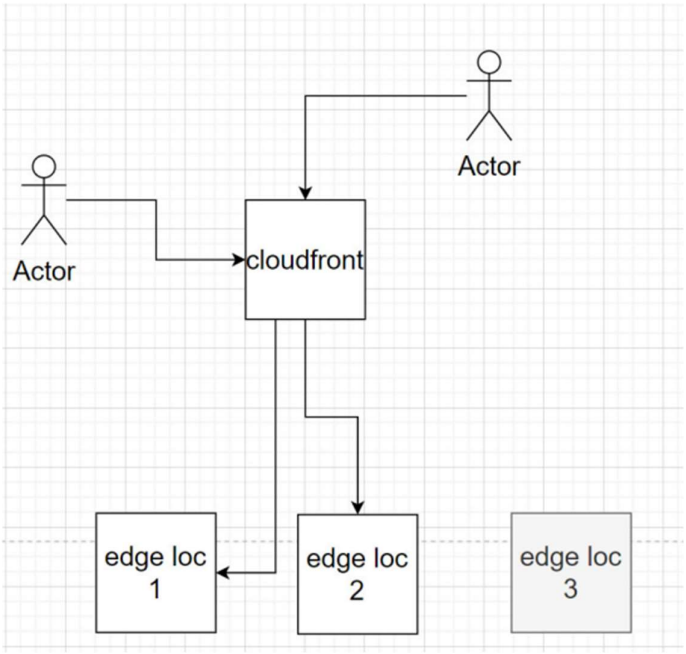


https://d3tmpgwkt7lfh4.cloudfront.net



Its is accessible, now the S3 endpoint access will be disabled because we removed the public access in the policy(updated)

CloudFront already does the job of routing traffic to the edge locations, so I don't have a purpose of DNS explicitly to be deployed here.



5. create CDN and attach S3 as the source

6. create OAI so that CDN can get objects from S3 bucket

7. attach edge locations so that data is distributed across all the servers across the globe

8. access entertainment webpage using CDN, and it will be accessible

9. thus solved the speed issue as S3 scales automatically based on traffic and load is globally balanced using CDN

Do we need a DNS now to distribute traffic at a global level?

10. in case if we are looking for custom DNS name then we can create route53 and attach CloudFront distribution to it as A record.