Website is static means it does not run any server side scripts

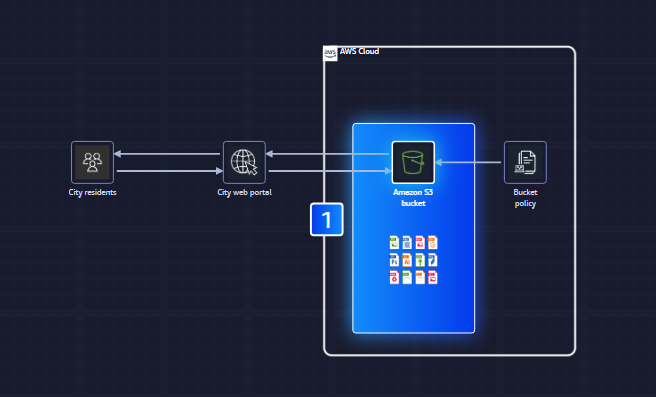
Move static web page to S3 and use static website hosting feature

It will also control access to the web page

For S3 you don’t need to deploy or manage any web server.

S3 buckets are automatically replicated across multiple AWS data centers for high resiliency

Also, any website hosted on Amazon S3 can automatically scale to handle thousands of concurrent requests.



Files and any metadata that describes a file is called as object

Objects are stored in S3 containers called buckets

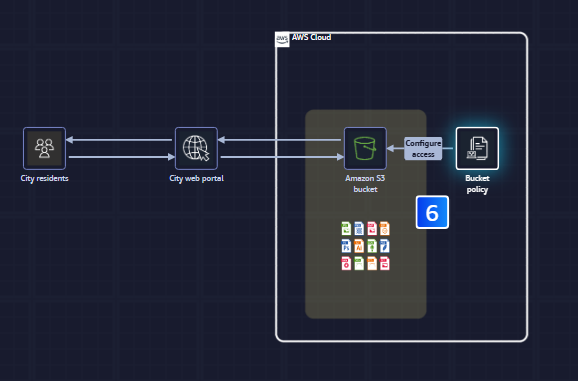
Static website will be hosted in S3 bucket

When, S3 bucket is configured for website hosting, the bucket is assigned a URL.

When requests are made to this URL, S3 returns the HTML file known as root object that was set for the bucket. This root object is index.html

This index.html can be renamed to anything like waves.html

Bucket policy can be configured to determine permissions

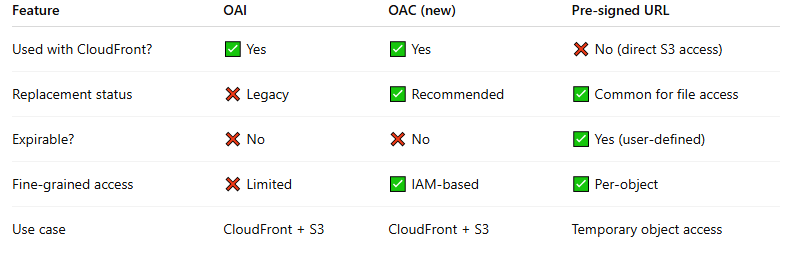


Bucket policy is written in JSON format

“Action”:[“s3:GetObject”],  
“Effect”:”Allow”,

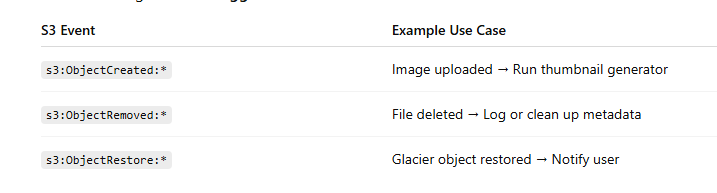
Steps

1. S3
2. Create a bucket
3. In this bucket we can have files like index.html, main.js, styles.css, text.html etc
4. We can upload upto 5GB in single PUT operation. For larger objects upto 5TB, we need multipart upload API
5. Permissions
6. Turning off "Block all public access" is necessary for static web hosting through your S3 bucket.
7. Encryption by default –SSE3 with Amazon Managed Keys
8. Static website hosting
9. index document = index.html
10. error document = error.html
11. Amazon S3 supports virtual-hosted-style URLs and path-style URLs. A virtual-hosted-style URL looks like: https://bucket-name.s3.Region.amazonaws.com/key A path-style URL looks like: <https://s3.Region.amazonaws.com/bucket-name/keyname>
12. Hosting type = Bucket hosting



OAI used cloudfront user identity

Setup Lambda on S3

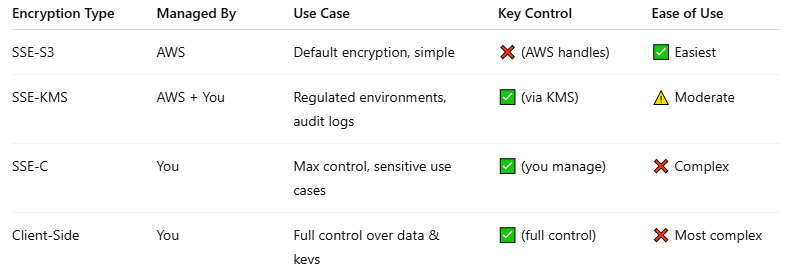
1. S3> Create bucket>Properties> Event Notification
2. 

There are two ways in which permission to S3 resources is granted

1. Identity based policy (IAM user, IAM role, IAM group)
2. Bucket Policy

S3 versioning – versioning allows uses to keep multiple variants of an object in the same S3 bucket

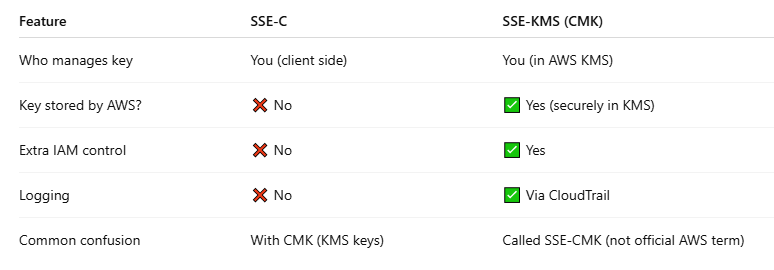
S3 object lock – write once read many (WORM)



KMS – Created, managed and used on your behalf by an AWS service integrated with AWS KMS

CMK – KMS keys that you create. Full control over these KMS keys





1. SSE-KMS/SSE-CMK
   1. kms:GenerateDatakey and kms:Decrypt in key policy
   2. x-amz-server-side-encryption: aws:kms
2. SSE-S3
   1. x-amz-server-side-encryption: AES256
3. SSE-C (Customer provided key)
   1. x-amz-server-side-encryption-customer-algorithm: AES256
   2. x-amz-server-side-encryption-customer-key: <Base64-encoded-key>
   3. x-amz-server-side-encryption-customer-key-MD5: <Base64-encoded-MD5>

S3 server Access logging – provides detailed records of the requests that are made to a bucket