Amazon S3
Another base block of AWS

Need more Space? Storage?

- The need for **storage** is increasing every day and knowing the amount of capacity you may need in the future is difficult to predict.

- You may either over-utilize it leading to an application failure because of not having sufficient space.

- you may end up buying stacks of storage which will then be underutilized.

- Keeping all these hassles in mind, Amazon came up with an internet storage service called *AWS S3*.

Amazon Simple Storage Service (S3)

- **Amazon S3** (Simple Storage Service) is a scalable, high-speed, low-cost web-based service designed for online backup and archiving of data and application programs.
- It allows to upload, store, and download any type of files up to 5 GB in size.
- This service allows the subscribers to access the same systems that Amazon uses to run its own web sites.
- The subscriber has control over the accessibility of data, i.e. privately/publicly accessible.

Amazon S3 Advantages

- **Low cost and Easy to Use** – Using S3, the user can store a large amount of data at very low charges.
- **Secure** – Amazon S3 supports data transfer over SSL and the data gets encrypted automatically once it is uploaded.
- **Scalable** – We can store as much data as we have and access it anytime.
- **Durable-** It regularly verifies the integrity of data stored using checksums e.g. if corruption in data, it is immediately repaired.
- **Higher performance** – Amazon S3 is integrated with Amazon CloudFront, that distributes content to the end users with low latency and provides high data transfer speeds without any minimum usage commitments.
- **Integrated with AWS services**

How is data organized in S3?

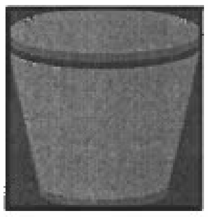
Data in S3 is organized in the form of buckets.

•A Bucket is a logical unit of storage in S3.
•A Bucket contains objects which contain the data and metadata.

Before adding any data in S3 the user has to create a bucket which will be used to store objects

AWS S3 Overview - Buckets

- Amazon S3 allows people to store objects (files) in "buckets" (directories)
- Buckets must have a globally unique name
- Buckets are defined at the region level
- Naming convention
  - No uppercase
  - No underscore
  - 3-63 characters long
  - Not an IP
  - Must start with lowercase letter or number

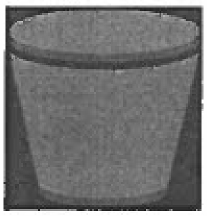


21

AWS S3 Overview - Objects

- Objects (files) have a Key.The key is the FULL path:
  - <my_bucket>/my_file.txt
  - <my_bucket>/my_folder1/another_folder/my_file.txt
- There's no concept of "directories" within buckets (although the UI will trick you to think otherwise)
- Just keys with very long names that contain slashes ("/")
- Object Values are the content of the body:
  - Max Size is 5TB
  - If uploading more than 5GB, must use "multi-part upload"
- Metadata (list of text key / value pairs – system or user metadata)
- Tags (Unicode key / value pair – up to 10) – useful for security / lifecycle
- Version ID (if versioning is enabled)

AWS S3 - Versioning



- You can version your files in AWS S3
- It is enabled at the bucket level
- Same key overwrite will increment the "version": 1, 2, 3….
- It is best practice to version your buckets
  - Protect against unintended deletes (ability to restore a version)
  - Easy roll back to previous version
- Any file that is not versioned prior to enabling versioning will have version "null"

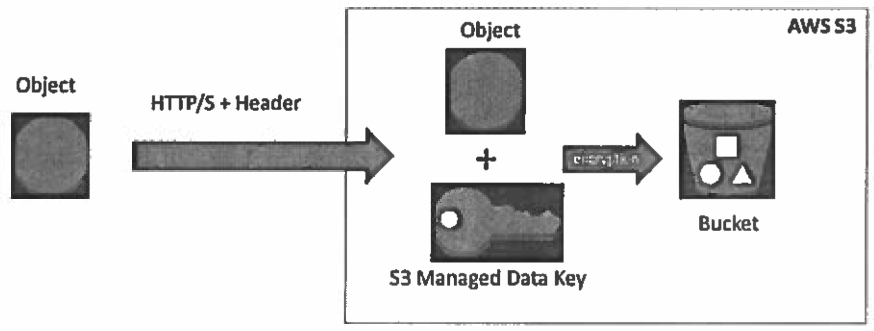
S3 Encryption for Objects

- There are 4 methods of encrypting objects in S3
  - SSE-S3: encrypts S3 objects using keys handled & managed by AWS
  - SSE-KMS: leverage AWS Key Management Service to manage encryption keys
  - SSE-C: when you want to manage your own encryption keys
  - Client Side Encryption

- It's important to understand which ones are adapted to which situation for the exam

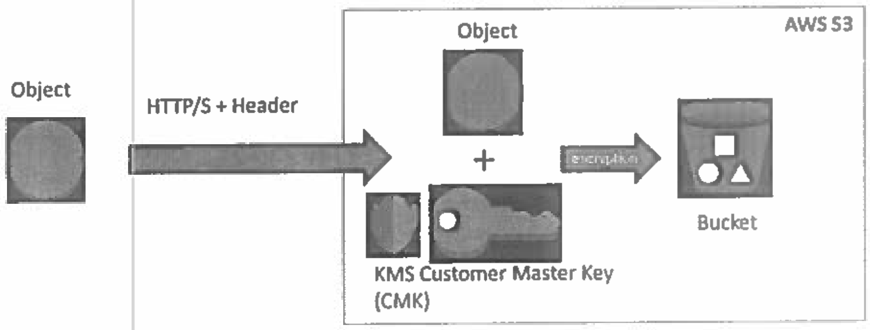
SSE-S3

- SSE-S3: encryption using keys handled & managed by AWS S3
- Object is encrypted server side
- AES-256 encryption type
- Must set header: “x-amz-server-side-encryption": "AES256"



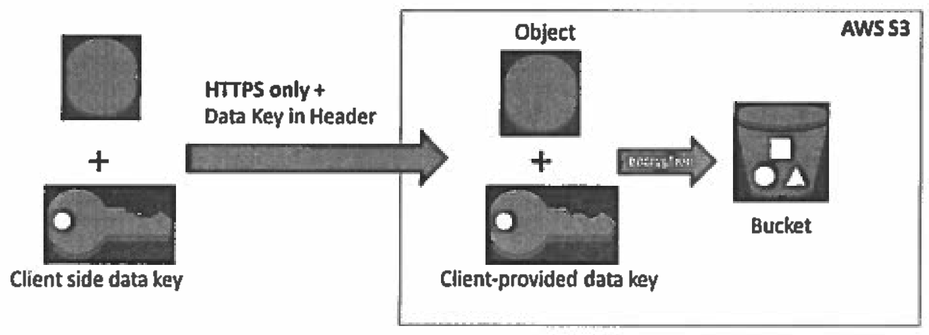
SSE-KMS

- SSE-KMS: encryption using keys handled & managed by KMS
- KMS Advantages: user control + audit trail
- Object is encrypted server side
- Must set header: 'x-amz-server-side-encryption": "aws:kms"



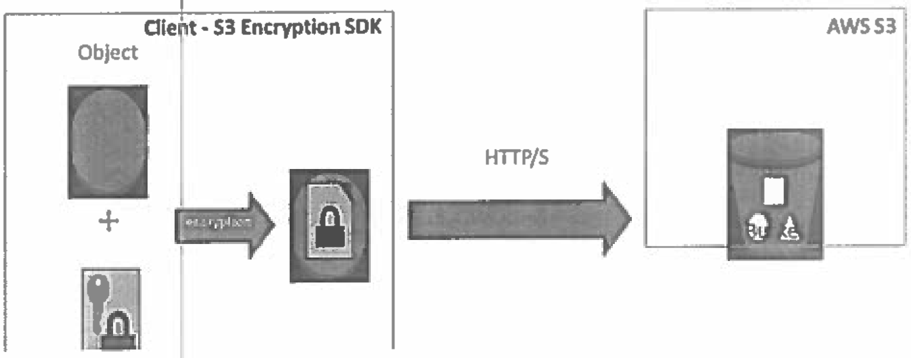
SSE-C

- Amazon S3 does not store the encryption key you provide
- HTTPS must be used
- Encryption key must provided in HTTP headers, for every HTTP request made
  Object



Client Side Encryption

- Clients must decrypt data themselves when retrieving from S3
- Customer fully manages the keys and encryption cycle



Encryption in transit (SSL)

- AWS S3 exposes:
  - HTTP endpoint: non encrypted
  - HTTPS endpoint: encryption in flight

- You're free to use the endpoint you want, but HTTPS is recommended
- HTTPS is mandatory for SSE-C
- Encryption in flight is also called SSL / TLS

AWS S3 Operations

- **Create a Bucket** – Create and name your own bucket in which to store your objects.
- **Write an Object** – Store data by creating or overwriting an object. When you write an object, you specify a unique key in the namespace of your bucket. This is also a good time to specify any access control you want on the object.
- **Read an Object** – Read data back. You can download the data via HTTP or BitTorrent.
- **Deleting an Object** – Delete some of your data.
- **Listing Keys** – List the keys contained in one of your buckets. You can filter the key list based on a prefix.

S3 Security

- User based
  - IAM policies - which API calls should be allowed for a specific user from IAM console
- Resource Based
  - Bucket Policies - bucket wide rules from the S3 console - allows cross account
  - Object Access Control List (ACL) – finer grain
  - Bucket Access Control List (ACL) – less common

S3 Bucket Policies

- JSON based policies
  - Resources: buckets and objects
  - Actions: Set of API to Allow or Deny
  - Effect: Allow / Deny
  - Principal:The account or user to apply the policy to

- Use S3 bucket for policy to:
  - Grant public access to the bucket
  - Force objects to be encrypted at upload
  - Grant access to another account (Cross Account)

32

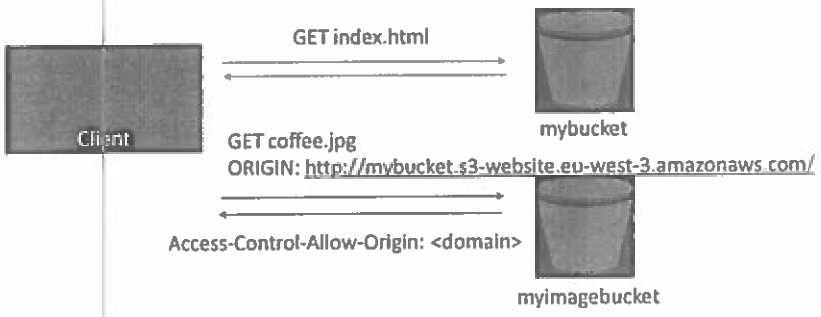
S3 Websites

- S3 can host static websites and have them accessible on the www
- The website URL will be:
  - <bucket-name>.s3-website-<AWS-region>.amazonaws.com

    OR
  - <bucket-name>.s3-website.<AWS-region>.amazonaws.com
- If you get a 403 (Forbidden) error, make sure the bucket policy allows public reads!

websites that can request your files in S3 (and limit your costs)

S3 CORS


GET index.html
mybucket
Client
GET coffee.jpg
ORIGIN: http://mybucket.s3-website.eu-west-3.amazonaws.com/
Access-Control-Allow-Origin: <domain>
myimagebucket

AWS S3 - Consistency Model

- Read after write consistency for PUTS of new objects
  - As soon as an object is written, we can retrieve it
    ex: (PUT 200 -> GET 200)
  - This is true, except if we did a GET before to see if the object existed
    ex: (GET 404 -> PUT 200 -> GET 404) – eventually consistent

- Eventual Consistency for DELETES and PUTS of existing objects
  - If we read an object after updating, we might get the older version
    ex: (PUT 200 -> PUT 200 -> GET 200 (might be older version))
  - If we delete an object, we might still be able to retrieve it for a short time ex: (DELETE 200 -> GET 200)

S3 storage classes with a "health-care" use case:

• **Amazon S3 Standard for frequent data access**

This is suitable for performance sensitive use cases where the latency should be kept low.

e.g. in a hospital, frequently accessed data will be the data of admitted patients, which should be retrieved quickly.

. **Amazon S3 Standard for infrequent data access**

This is suitable for use cases where the data is long lived and less frequently accessed, i.e for data archival but still expects high performance.

e.g. in the same hospital, people who have been discharged, their records/data will not be needed on a daily basis, but if they return with any complication, their discharge summary should be retrieved quickly.

S3 storage classes with a "health-care" use case:

Amazon Glacier:

Suitable for use cases where the data is to be archived, and high performance is not required, it has a lower cost than the other two services.

e.g. in the hospital, patients' test reports, prescriptions, MRI, X Ray, Scan docs etc. that are older than a year will not be needed in the daily run and even if it is required, lower latency is not needed.

Characteristics	Standard	Standard - Infrequent Access	Glacier
Durability	99.99%	99.99%	99.99%
Availability	99.99%	99.90%	N/A
Minimum Object Size	No limit	128KB	No limit
Minimum Storage Duration	No minimum duration	30 Days	90 Days
First Byte Latency	milliseconds	milliseconds	4 hours
Retrieval Fee	No Fee	per GB retrieved	per GB retrieved

AWS S3 Features

- **Storage Class**
- Amazon S3 offers a range of storage classes designed for different use cases.
- These include Amazon S3 STANDARD for general-purpose storage of frequently accessed data.
- Amazon S3 STANDARD_IA for long-lived, but less frequently accessed data,
- And GLACIER for long-term archive.

Lets do a small Project

Hosting a Static website on AWS S3

Project Statement

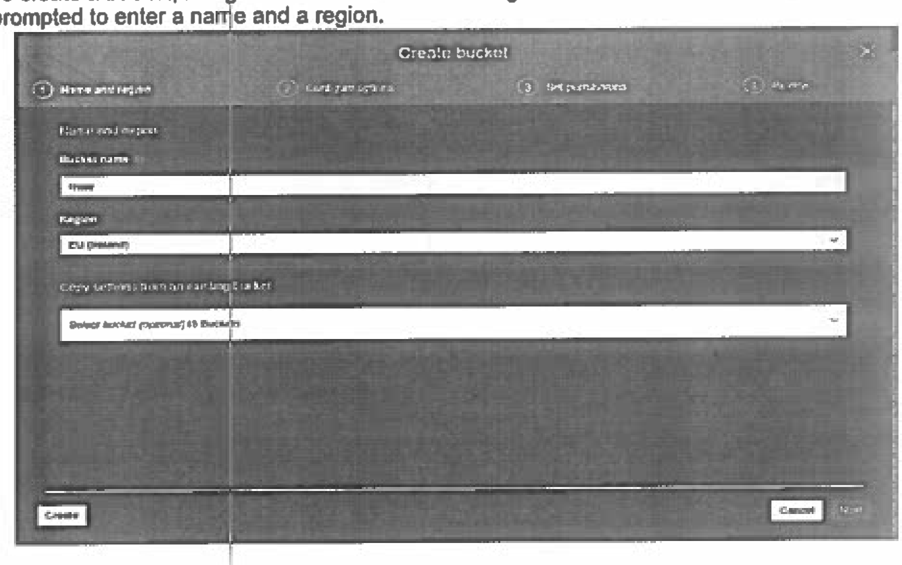
Project Statement – Hosting a Static Website on Amazon S3

- Let's first understand: What is a static website?
- In short, it's a website comprised of only HTML, CSS, and/or JavaScript. That means server-side scripts aren't supported, so if you want to host a Rails or PHP app, you'll need to look elsewhere.
- For simpler purposes, welcome to the wonderful world of hosting websites on AWS S3!

41

Step 1: Create a bucket

- To create a bucket, navigate to S3 in the AWS Management Console and hit Create Bucket. You'll be prompted to enter a name and a region.



Name and region

Bucket name: [illegible]

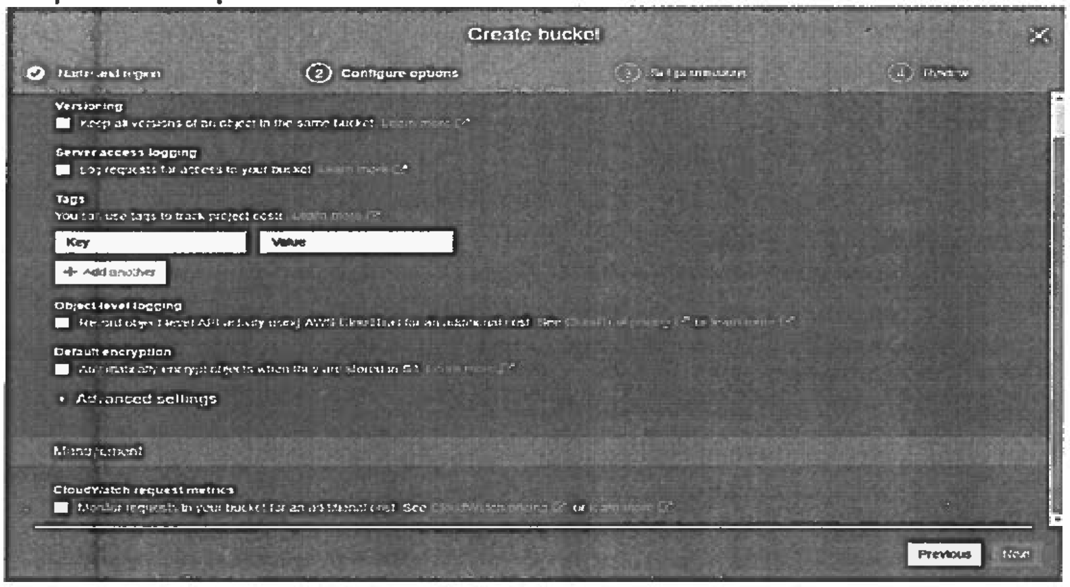
Region: EU (Ireland)

Copy settings from an existing bucket

Select bucket (optional) 49 Buckets

42

Step 2: Bucket Properties



Create bucket

Name and region | 2 Configure options | 3 Set permissions | 4 Review

Versioning

Keep all versions of an object in the same bucket. Learn more

Server access logging

Log requests for access to your bucket. Learn more

Tags

You can use tags to track project cost. Learn more

Key | Value

+ Add another

Object-level logging

Record object-level API activity using AWS CloudTrail for an additional cost. See CloudTrail pricing or learn more

Default encryption

Automatically encrypt objects when they are stored in S3. Learn more

Advanced settings

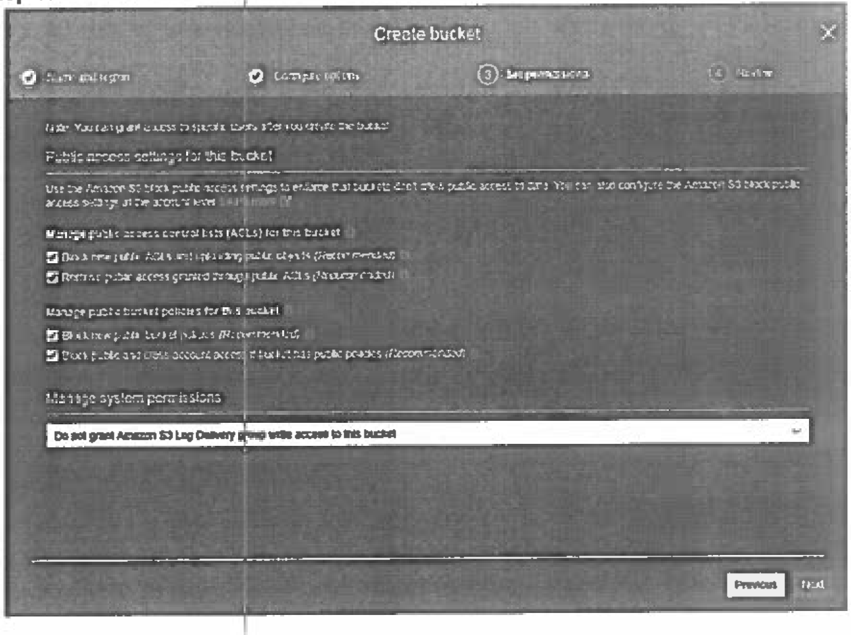
Management

CloudWatch request metrics

Monitor requests in your bucket for an additional cost. See CloudWatch pricing or learn more

Previous | Next

Step 3: Permissions



Note: You can grant access to specific users after you create the bucket.

Public access settings for this bucket

Use the Amazon S3 block public access settings to enforce that buckets don't allow public access to data. You can also configure the Amazon S3 block public access settings at the account level. Learn more

Manage public access control lists (ACLs) for this bucket

✓ Block new public ACLs and uploading public objects (Recommended)

✓ Remove public access granted through public ACLs (Recommended)

Manage public bucket policies for this bucket

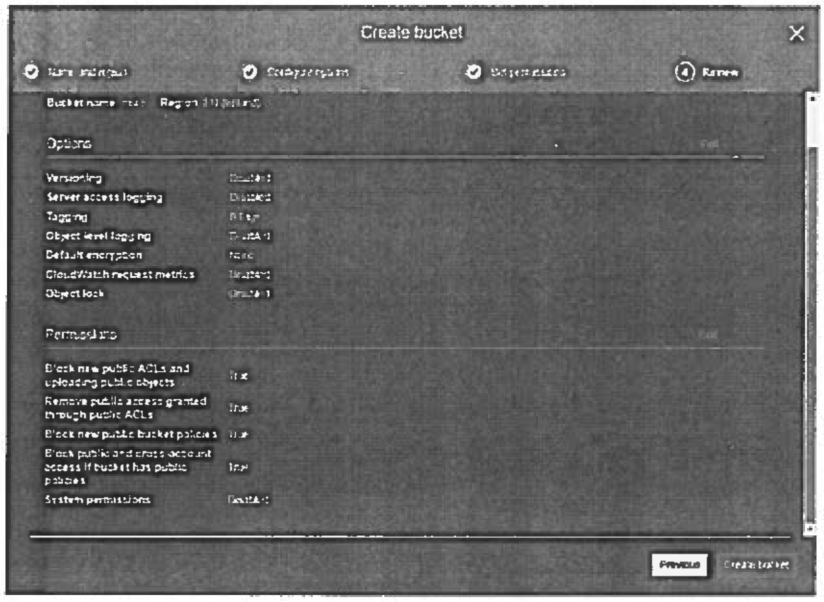
✓ Block new public bucket policies (Recommended)

✓ Block public and cross-account access if bucket has public policies (Recommended)

Manage system permissions

Do not grant Amazon S3 Log Delivery group write access to this bucket

Step 1: Create a bucket



Bucket name [illegible] Region [illegible]

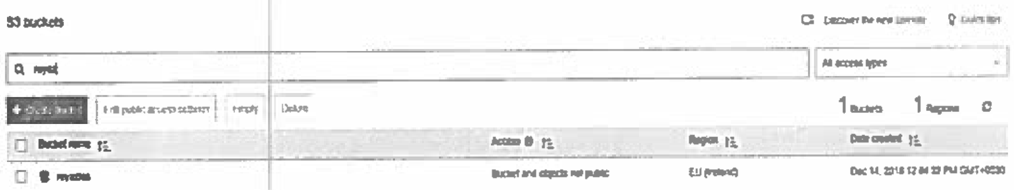
Options

Versioning	Disabled
Server access logging	Disabled
Tagging	0 Tags
Object level logging	Disabled
Default encryption	None
CloudWatch request metrics	Disabled
Object lock	Disabled

Permissions

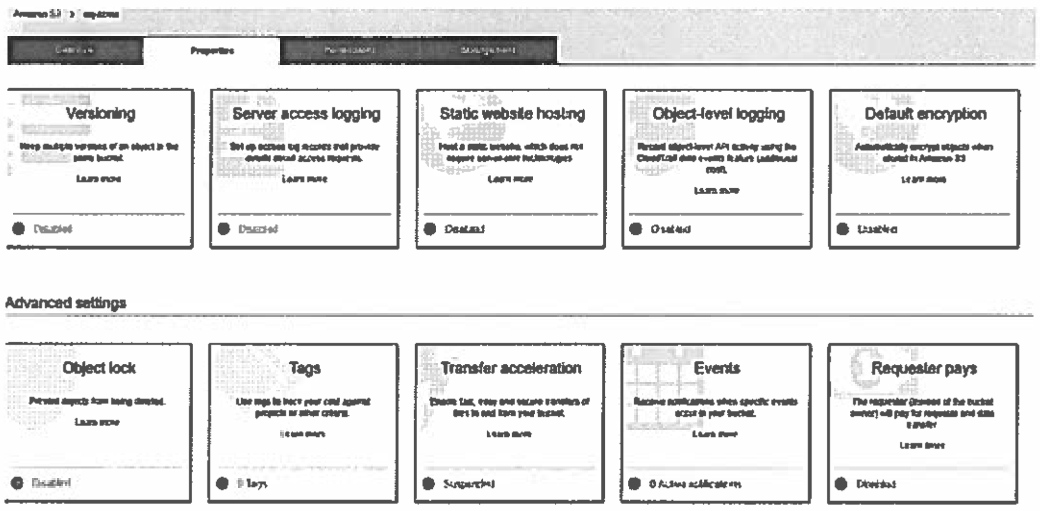
Block new public ACLs and uploading public objects	True
Remove public access granted through public ACLs	True
Block new public bucket policies	True
Block public and cross-account access if bucket has public policies	True
System permissions	Disabled

Find your bucket by searching.

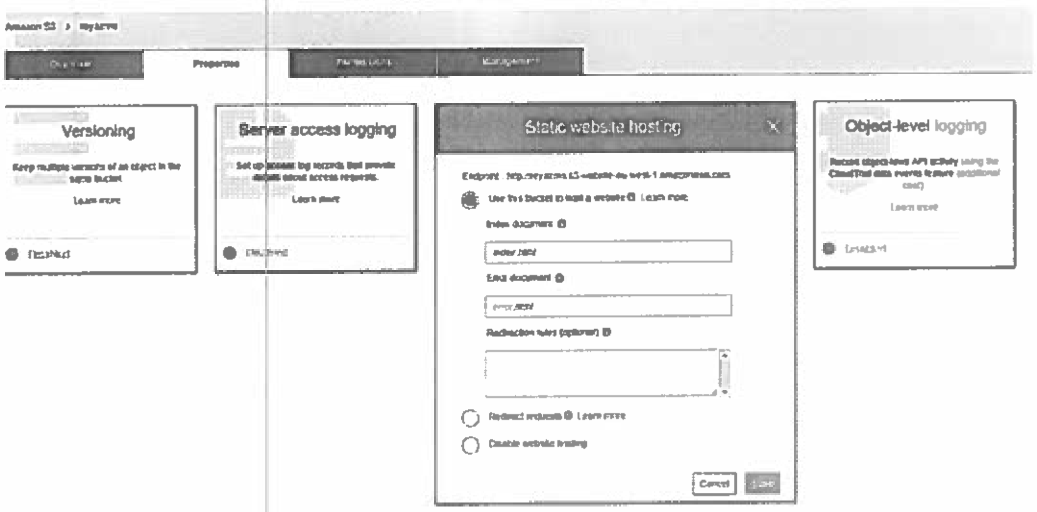


Bucket name	Access	Region	Date created
[illegible]	Bucket and objects not public	EU (Ireland)	Dec 14, 2018 12:44:33 PM GMT+0530

Find your bucket and click on it and you will see the below screen



Click on Static website hosting and select "Use this bucket to host a website"



Amazon S3 > [illegible]

Overview | Properties | Permissions | Management

Versioning

Keep multiple versions of an object in the same bucket

Learn more

● Disabled

Server access logging

Set up access log records that provide details about access requests.

Learn more

● Disabled

Static website hosting

Endpoint : [illegible]

● Use this bucket to host a website Learn more

Index document

index.html

Error document

error.html

Redirection rules (optional)

○ Redirect requests Learn more

○ Disable website hosting

Cancel | Save

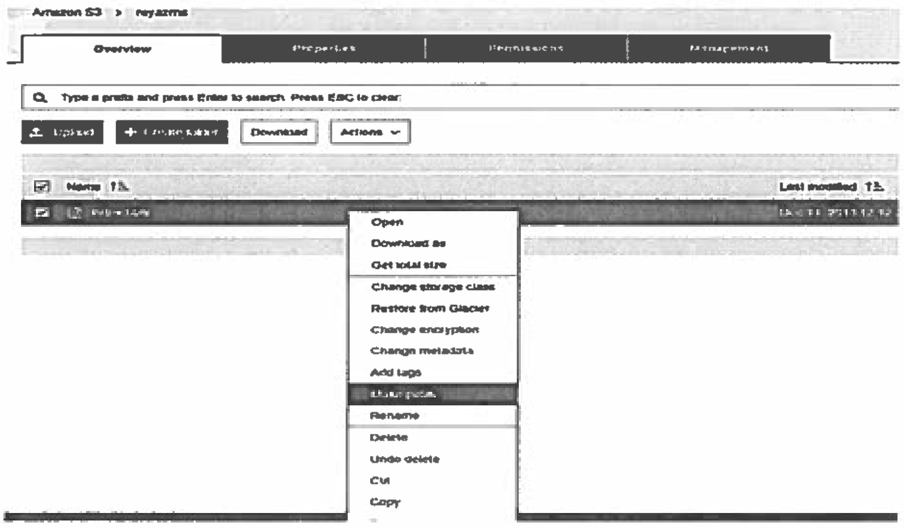
Object-level logging

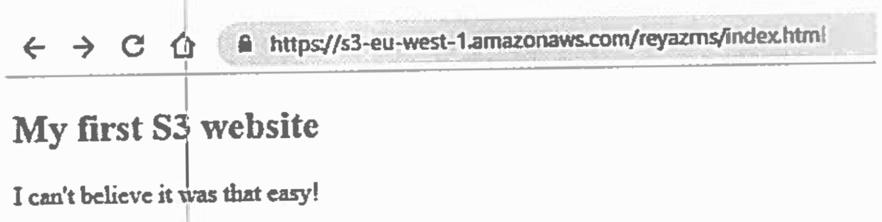
Record object-level API activity using the CloudTrail data events feature (additional cost)

Learn more

● Disabled

You don't have index.html. Create a simple html file with name index.html and upload to the bucket and Make public.
Access the bucket link from the properties





My first S3 website

I can't believe it was that easy!